

Serviceguard 버전 A.11.16 관리

제11판

제2판



i n v e n t

제품 제조 번호: B3936-90082

2004년 5월

2005년 6월 제2판 발행

알림

© Copyright 1995-2005 Hewlett-Packard Development Company, L.P.

기밀 컴퓨터 소프트웨어입니다. 소유하거나 사용하거나 복사하려면 HP에서 유효한 라이선스를 얻어야 합니다. FAR 12.1211 및 12.212에 따라 상업용 품목의 기술 데이터, 상업용 컴퓨터 소프트웨어 및 컴퓨터 소프트웨어 설명서는 공급업체의 표준 상업용 라이선스에 의거하여 미국 정부에 사용이 허가되었습니다.

이 문서의 내용은 예고 없이 변경될 수 있습니다.

HP 제품과 서비스에 대한 보증은 오직 제품 및 서비스와 함께 제공되는 명시적 보증서만을 근거로 합니다. 이 설명서의 어떤 내용도 추가 보증 제정으로 해석할 수 없습니다. HP는 이 문서에 포함된 기술적 오류나 편집상의 오류에 대해 책임을 지지 않습니다.

HP 본사 위치:

*Hewlett-Packard Co.
3000 Hanover St.
Palo Alto, CA 94304*

UNIX는 The Open Group의 등록 상표입니다. Microsoft Windows는 Microsoft Corporation의 미국 등록 상표입니다. Linux Red Hat Advanced Server 2.1 Red Hat Enterprise Linux는 Red Hat의 미국 등록 상표입니다. SuSE Linux는 SuSE의 미국 등록 상표입니다. Microsoft 및 Windows는 Microsoft Corporation의 미국 등록 상표입니다.

1. Serviceguard 소개

Serviceguard의 개념 24

 장애 조치 25

Serviceguard Manager 사용 27

 Serviceguard Manager를 사용한 모니터링 27

 Serviceguard Manager를 사용한 관리 28

 Serviceguard Manager를 사용한 구성 29

 Serviceguard Manager 도움말 30

 Serviceguard Manager의 작동 방식 31

클러스터 및 패키지 구성 정보 33

2. Serviceguard 하드웨어 구성 이해

클러스터 구성 요소의 중복 36

중복 네트워크 구성 요소 38

 중복 이더넷 구성 38

 중복 FDDI 연결 제공 40

 이중 접속 FDDI 스테이션 사용 41

 직렬(RS232) 하트비트 회선 사용 41

 장애가 발생한 네트워크 카드의 교체 43

중복 디스크 저장 장치 44

 지원되는 디스크 인터페이스 44

 데이터 보호 45

 Event Monitoring Service를 통한 디스크 모니터링 46

 장애가 발생한 디스크 메커니즘의 교체 46

 장애가 발생한 입출력 카드의 교체 46

 SCSI 디스크 구성 예제 47

 파이버 채널 디스크 구성 예제 49

 공유 SCSI 버스에서 루트 디스크 제한 50

중복 전원 공급 장치 53

큰 규모의 클러스터 54

 활성/대기 모델 54

 저장 장치에 지점 간 연결 55

3. Serviceguard 소프트웨어 구성 요소 이해

Serviceguard 아키텍처 58

목차

| | |
|--|----|
| Serviceguard 데몬 | 58 |
| 클러스터 관리자의 작동 방식 | 62 |
| 클러스터 구성 | 62 |
| 하트비트 메시지 | 62 |
| 전체 클러스터의 수동 시작 | 64 |
| 클러스터 자동 시작 | 64 |
| 동적 클러스터 재편성 | 64 |
| 좌우 분열 현상을 방지하는 클러스터 쿼럼 | 65 |
| 클러스터 잠금 | 65 |
| 클러스터 잠금으로 LVM 잠금 디스크 사용 | 66 |
| 클러스터 잠금으로 쿼럼 서버 사용 | 68 |
| 클러스터 잠금 없음 | 69 |
| 패키지 관리자의 작동 방식 | 71 |
| 패키지 종류 | 71 |
| 장애 조치 패키지 | 71 |
| 패키지 구성 | 72 |
| 패키지 실행/중지 시기 및 위치 결정 | 72 |
| 패키지 전환 | 72 |
| 장애 조치 정책 | 75 |
| 장애 복구 정책 | 78 |
| 이전 버전의 패키지 구성 파일 사용 | 81 |
| Event Monitoring Service 사용 | 81 |
| EMS HA 모니터 사용 | 82 |
| 패키지 장애 조치 동작 선택 | 82 |
| 패키지 제어 스크립트의 작동 방식 | 85 |
| 패키지 실행을 제어하는 요소 | 85 |
| 제어 스크립트 시작 전 | 86 |
| 실행 스크립트 실행 중 | 87 |
| 실행 스크립트의 정상 종료와 비정상 종료 | 88 |
| cmrunserv를 사용한 서비스 시작 | 89 |
| 서비스 실행 중 | 89 |
| 서비스, 서브넷 또는 모니터링되는 리소스에서 장애가 발생할 때 | 90 |
| 명령에 의해 패키지가 중지될 때 | 91 |
| 중지 스크립트 실행 중 | 91 |
| 중지 스크립트의 정상 종료와 비정상 종료 | 92 |
| 네트워크 관리자의 작동 방식 | 96 |

| | |
|---|-----|
| 고정 IP 주소와 변동 가능 IP 주소 | 96 |
| 변동 가능 IP 주소 추가 및 삭제 | 97 |
| LAN 인터페이스 모니터링 및 장애 감지 | 97 |
| 자동 포트 통합 | 103 |
| VLAN 구성 | 105 |
| 데이터 저장 장치의 볼륨 관리자 | 108 |
| 중복 저장 장치의 유형 | 108 |
| 미러링된 저장 장치의 예 | 108 |
| 디스크 어레이의 저장 장치 예 | 110 |
| 볼륨 관리자의 유형 | 112 |
| HP-UX Logical Volume Manager(LVM) | 113 |
| VERITAS Volume Manager(VxVM) | 113 |
| VERITAS Cluster Volume Manager(CVM) | 114 |
| 볼륨 관리자 비교 | 116 |
| 장애에 대한 대응 | 119 |
| 노드에 장애가 발생할 때의 TOC(제어권 전달) | 119 |
| 하드웨어 장애에 대한 대응 | 120 |
| 패키지 및 서비스 장애에 대한 대응 | 120 |
| 서비스 재시작 | 121 |
| 네트워크 통신 장애 | 121 |

4. HA 클러스터 계획 및 문서화

| | |
|--------------------------------------|-----|
| 일반 계획 | 125 |
| Serviceguard 메모리 요구 사항 | 125 |
| 확장을 고려한 계획 | 126 |
| 하드웨어 계획 | 127 |
| SPU 정보 | 128 |
| 네트워크 정보 | 128 |
| 예상되는 최대 클러스터 크기에 맞는 SCSI 주소 설정 | 132 |
| 디스크 입출력 정보 | 133 |
| 하드웨어 구성 워크시트 | 134 |
| 전원 공급 계획 | 136 |
| 전원 공급 구성 워크시트 | 137 |
| 쿼럼 서버 계획 | 139 |
| 쿼럼 서버 워크시트 | 140 |
| LVM 계획 | 141 |

목차

| | |
|-------------------------|-----|
| LVM 워크시트 | 142 |
| CVM 및 VxVM 계획 | 144 |
| CVM 및 VxVM 워크시트 | 144 |
| 클러스터 구성 계획 | 146 |
| 하트비트 서브넷과 재편성 시간 | 147 |
| 클러스터 잠금 정보 | 147 |
| 클러스터 구성 매개 변수 | 148 |
| 클러스터 구성 워크시트 | 155 |
| 패키지 구성 계획 | 157 |
| 논리 볼륨 및 파일 시스템 계획 | 157 |
| EMS 리소스 구성 매개 변수 | 159 |
| 확장을 고려한 계획 | 160 |
| 전환 및 장애 조치 동작 선택 | 160 |
| 패키지 구성 파일 매개 변수 | 160 |
| 패키지 제어 스크립트 변수 | 170 |

5. HA 클러스터 구성 만들기

| | |
|--------------------------------------|-----|
| 시스템 준비 | 180 |
| 파일 위치 이해 | 180 |
| 보안 파일 편집 | 182 |
| 액세스 역할 | 185 |
| 이름 확인 서비스 정의 | 188 |
| 루트 논리 볼륨의 미리 만들기 | 190 |
| 클러스터 잠금 디스크 선택 | 192 |
| 커널 구성의 일관성 유지 | 193 |
| 네트워크 시간 프로토콜 사용 | 193 |
| 네트워크 및 커널 매개 변수 조정 | 193 |
| 클러스터 크기 변경 준비 | 194 |
| 쿼럼 서버 설치 | 196 |
| 쿼럼 서버 설치 | 196 |
| 쿼럼 서버 실행 | 197 |
| Serviceguard 설치 | 198 |
| LVM을 사용하여 저장 장치 하부 구조 만들기 | 199 |
| 미러링되는 개별 데이터 디스크의 볼륨 그룹 만들기 | 199 |
| PV 링크를 사용하는 디스크 어레이의 볼륨 그룹 만들기 | 203 |
| 볼륨 그룹을 다른 노드에 배포 | 205 |

| | |
|---|-----|
| 추가 볼륨 그룹 만들기 | 208 |
| VxVM을 사용하여 저장 장치 하부 구조 만들기 | 209 |
| VERITAS Volume Manager 초기화 | 209 |
| LVM에서 VxVM으로 디스크 변환 | 210 |
| VxVM용 디스크 초기화 | 210 |
| 이전에 LVM에서 사용된 디스크 초기화 | 210 |
| 디스크 그룹 만들기 | 211 |
| 볼륨 만들기 | 211 |
| 파일 시스템 만들기 | 212 |
| 디스크 그룹 내보내기 | 212 |
| 디스크 그룹 다시 가져오기 | 213 |
| 시스템 재부팅 시 Clearimport | 213 |
| 클러스터 구성 | 214 |
| Serviceguard Manager 를 사용하여 클러스터 구성 | 214 |
| Serviceguard 명령을 사용하여 클러스터 구성 | 215 |
| 클러스터 구성 확인 | 225 |
| 이진 구성 파일 배포 | 227 |
| CVM을 사용하여 저장 장치 하부 구조 만들기 | 230 |
| VERITAS Volume Manager 초기화 | 230 |
| 클러스터를 CVM과 함께 사용할 수 있도록 준비 | 231 |
| 클러스터 시작 및 마스터 노드 확인 | 232 |
| CVM용 디스크 초기화 | 233 |
| 디스크 그룹 만들기 | 233 |
| 볼륨 만들기 | 233 |
| 파일 시스템 만들기 | 234 |
| 패키지 구성에 디스크 그룹 추가 | 235 |
| 실행 중인 클러스터 관리 | 236 |
| Serviceguard Manager 를 사용하여 클러스터 작동 검사 | 236 |
| Serviceguard 명령을 사용하여 클러스터 작동 검사 | 236 |
| 볼륨 그룹의 자동 활성화 방지 | 238 |
| 자동 시작 기능 설정 | 239 |
| 시스템 메시지 변경 | 239 |
| 단일 노드 클러스터 관리 | 240 |
| 클러스터 구성 삭제 | 241 |

6. 패키지 및 서비스 구성

목차

| | |
|---|-----|
| 패키지 구성 만들기 | 244 |
| Serviceguard Manager 를 사용하여 패키지 구성 | 244 |
| Serviceguard 명령을 사용하여 패키지 구성 | 245 |
| 실행 중인 클러스터에 패키지 추가 또는 제거..... | 255 |
| 패키지 제어 스크립트 작성..... | 256 |
| Serviceguard Manager 를 사용하여 패키지 제어 스크립트 작성 | 256 |
| 명령을 사용하여 패키지 제어 스크립트 작성 | 256 |
| 데이터베이스 제품의 패키지 만들기..... | 257 |
| 패키지 제어 스크립트 사용자 정의..... | 257 |
| 다수의 저장 장치를 위한 최적화..... | 260 |
| 패키지 제어 스크립트 템플릿 파일..... | 260 |
| 패키지 제어 스크립트에 사용자 정의 함수 추가 | 269 |
| 추가 제품 지원 | 271 |
| 패키지 구성 확인 | 272 |
| 구성 배포 | 273 |
| Serviceguard Manager 를 사용하여 구성 파일 및 제어 스크립트 배포 | 273 |
| HP-UX 명령을 사용하여 패키지 제어 스크립트 복사..... | 273 |
| HP-UX 명령을 사용하여 이진 클러스터 구성 파일 배포 | 273 |
| 클러스터 및 패키지 작동 테스트..... | 274 |

7. 클러스터 및 패키지 유지 관리

| | |
|---|-----|
| 클러스터 및 패키지 상태 확인 | 276 |
| Serviceguard Manger 를 사용하여 클러스터 및 패키지 상태 확인..... | 276 |
| cmviewcl 명령을 사용하여 클러스터 및 패키지 상태 확인..... | 277 |
| 클러스터 및 노드 관리 | 292 |
| 모든 노드가 중지되었을 때 클러스터 시작 | 292 |
| 실행 중인 클러스터에 이미 구성된 노드 추가..... | 294 |
| 실행 중인 클러스터에서 노드 제거..... | 294 |
| 전체 클러스터 중지 | 295 |
| 클러스터 자동 재시작 | 296 |
| 패키지 및 서비스 관리 | 297 |
| 패키지 시작 | 297 |
| 패키지 중지 | 298 |
| 패키지 이동 | 298 |
| 패키지 전환 동작 변경 | 299 |
| 클러스터 재구성..... | 302 |

| | |
|-----------------------------|-----|
| 중지된 클러스터 재구성 | 303 |
| 실행 중인 클러스터 재구성 | 304 |
| 패키지 재구성 | 310 |
| 중지된 클러스터에서 패키지 재구성 | 310 |
| 실행 중인 클러스터에서 패키지 재구성 | 311 |
| 실행 중인 클러스터에 패키지 추가 | 311 |
| 실행 중인 클러스터에서 패키지 삭제 | 312 |
| 서비스 재시작 카운터 재설정 | 312 |
| 재구성 중 허용되는 패키지 상태 | 313 |
| 클러스터 이벤트에 대응 | 316 |
| 시스템에서 Serviceguard 제거 | 317 |

8. 클러스터 문제 해결

| | |
|---|-----|
| 클러스터 작동 테스트 | 320 |
| Serviceguard Manager를 사용하여 클러스터 시작 | 320 |
| 패키지 관리자 테스트 | 320 |
| 클러스터 관리자 테스트 | 321 |
| 네트워크 관리자 테스트 | 321 |
| 하드웨어 모니터링 | 323 |
| Event Monitoring Service 사용 | 323 |
| EMS 하드웨어 모니터 사용 | 323 |
| 하드웨어 모니터 및 연속성 요청 | 324 |
| HP Predictive 모니터링 사용 | 324 |
| 디스크 교체 | 325 |
| 고장난 어레이 메커니즘 교체 | 325 |
| HA 엔클로저의 고장난 메커니즘 교체 | 325 |
| 잠금 디스크 교체 | 326 |
| 인라인 SCSI 접속 단자를 사용하여 온라인 하드웨어 유지 관리 | 326 |
| 입출력 카드 교체 | 330 |
| LAN 카드 교체 | 331 |
| 오프라인 교체 | 331 |
| 온라인 교체 | 331 |
| 카드 교체 이후 | 332 |
| 잘못된 퀴럼 서버 시스템 교체 | 333 |
| 문제 해결 방법 | 335 |
| 패키지 IP 주소 확인 | 335 |

목차

| | |
|--|-----|
| 시스템 로그 파일 확인 | 336 |
| 객체 관리자 로그 파일 확인 | 338 |
| Serviceguard Manager 로그 파일 확인 | 338 |
| 구성 파일 확인 | 338 |
| 패키지 제어 스크립트 확인 | 338 |
| cmcheckconf 명령 사용 | 339 |
| cmscancl 명령 사용 | 339 |
| cmviewconf 명령 사용 | 340 |
| LAN 구성 확인 | 340 |
| 문제 해결 | 342 |
| Serviceguard 명령 중지 | 342 |
| 클러스터 재편성 | 343 |
| 시스템 관리 오류 | 343 |
| VxVM 디스크 그룹 문제 | 346 |
| 패키지 이동 오류 | 348 |
| 노드 및 네트워크 장애 | 348 |
| 쿼럼 서버 문제 해결 | 349 |

A. Serviceguard 명령

B. Enterprise Cluster Master Toolkit

C. 고가용성 클러스터 응용 프로그램 설계

| | |
|----------------------------|-----|
| 응용 프로그램 작동 자동화 | 366 |
| 사용자 개입 최소화 | 366 |
| 응용 프로그램 시작 및 종료 정의 | 367 |
| 응용 프로그램의 장애 조치 속도 제어 | 368 |
| 비 데이터 파일 시스템 복제 | 368 |
| 원시 볼륨 사용 | 369 |
| JFS 사용 검토 | 369 |
| 데이터 손실 최소화 | 369 |
| 재시작 가능한 트랜잭션 사용 | 370 |
| 검사점 사용 | 371 |
| 다중 서버의 설계 | 372 |
| 복제된 데이터 지점의 설계 | 372 |

| | |
|-----------------------------------|-----|
| 다중 시스템에서 실행할 응용 프로그램 설계 | 373 |
| 노드에 특정한 정보 최소화 | 373 |
| SPU ID 또는 MAC 주소 사용 최소화 | 375 |
| 응용 프로그램에 고유 이름 지정 | 375 |
| uname(2) 사용 시 주의점 | 376 |
| 고정 포트에 바인딩 | 377 |
| 변동 가능 IP 주소에 바인딩 | 377 |
| 응용 프로그램마다 고유한 볼륨 그룹 부여 | 378 |
| SNA 응용 프로그램을 위한 복수 목적지 사용 | 379 |
| 파일 잠금 방지 | 379 |
| 클라이언트 연결 복원 | 380 |
| 응용 프로그램 장애 처리 | 382 |
| 장애를 극복할 수 있는 응용 프로그램 만들기 | 382 |
| 응용 프로그램 모니터링 가능 | 383 |
| 계획된 중단 시간 최소화 | 384 |
| 응용 프로그램 업그레이드/패치에 필요한 시간 단축 | 384 |
| 온라인 응용 프로그램 재구성 제공 | 386 |
| 유지 관리 작업 문서화 | 386 |

D. HA 응용 프로그램과 Serviceguard 통합

| | |
|---------------------------------|-----|
| HA 응용 프로그램을 통합하기 위한 검사 목록 | 388 |
| 단일 시스템의 기존 응용 프로그램 동작 정의 | 388 |
| 다중 시스템에 HA 응용 프로그램 통합 | 388 |
| 클러스터 테스트 | 389 |

E. 롤링 소프트웨어 업그레이드

| | |
|-----------------------|-----|
| 롤링 업그레이드 단계 | 392 |
| 커널 일관성 유지 | 393 |
| 롤링 업그레이드 예제 | 394 |
| 단계 1 | 394 |
| 단계 2 | 395 |
| 단계 3 | 395 |
| 단계 4 | 396 |
| 단계 5 | 396 |
| 롤링 업그레이드의 제한 사항 | 398 |

목차

F. 계획 워크시트 양식

| | |
|------------------------------|-----|
| 하드웨어 계획 워크시트..... | 400 |
| 전원 공급 장치 워크시트 | 402 |
| 쿼텀 서버 워크시트 | 403 |
| LVM 볼륨 그룹 및 물리 볼륨 워크시트 | 404 |
| VxVM 디스크 그룹 및 디스크 워크시트 | 406 |
| 클러스터 구성 워크시트..... | 408 |
| 패키지 구성 워크시트..... | 410 |
| 패키지 제어 스크립트 워크시트..... | 411 |

G. LVM에서 VxVM 데이터 저장 장치로 마이그레이션

| | |
|----------------------|-----|
| VxVM 로드..... | 414 |
| 볼륨 그룹 마이그레이션..... | 415 |
| VxVM 패키지 사용자 정의..... | 417 |
| CVM 패키지 사용자 정의..... | 419 |
| LVM 볼륨 그룹 제거..... | 422 |

H. IPv6 네트워크 지원

| | |
|-----------------------------------|-----|
| IPv6 주소 유형..... | 424 |
| IPv6 주소의 텍스트 표현 | 424 |
| IPv6 주소 접두부 | 425 |
| 유니캐스트 주소 | 426 |
| IPv4 및 IPv6 호환성..... | 426 |
| 네트워크 구성 제한 사항 | 430 |
| IPv6 변동 가능 주소 및 중복 주소 감지 기능 | 431 |
| 로컬 기본/대기 LAN 패턴..... | 433 |
| 구성 예제 | 434 |
| 중복 주소 감지 기능 | 436 |

| | |
|--|-----|
| 표 1. | 17 |
| 표 3-1. 패키지 구성 데이터..... | 76 |
| 표 3-2. 샘플 클러스터의 노드 목록..... | 79 |
| 표 3-3. 패키지 장애 조치 동작..... | 83 |
| 표 3-4. 오류 조건 및 패키지 이동..... | 93 |
| 표 3-5. Serviceguard에 있는 볼륨 관리자의 장점 및 단점..... | 117 |
| 표 4-1. 클러스터 구성에서 SCSI 주소 지정..... | 132 |
| 표 5-1. cmclnodelist 예제..... | 187 |
| 표 7-1. 영구적인 클러스터 구성 변경의 종류..... | 302 |
| 표 7-2. 패키지의 변경 종류..... | 313 |
| 표 8-1. cmscancl 명령으로 표시되는 데이터..... | 340 |
| 표 A-1. MC/Serviceguard 명령..... | 351 |
| 표 H-1. IPv6 주소 유형..... | 424 |
| 표 H-2. | 426 |
| 표 H-3. | 426 |
| 표 H-4. | 427 |
| 표 H-5. | 427 |
| 표 H-6. | 428 |
| 표 H-7. | 428 |
| 표 H-8. | 428 |

| | |
|--|-----|
| 그림 1-1. 일반적인 클러스터 구성 | 23 |
| 그림 1-2. 장애 조치 후의 일반적인 클러스터 | 25 |
| 그림 1-3. Serviceguard Manager를 사용한 모니터링 | 28 |
| 그림 1-4. Serviceguard Manager 패키지 관리 | 29 |
| 그림 1-5. Serviceguard Manager를 사용한 구성 | 30 |
| 그림 1-6. Serviceguard 클러스터 구성 작업 | 33 |
| 그림 2-1. 중복 LAN | 39 |
| 그림 2-2. 중복 FDDI 구성 | 40 |
| 그림 2-3. 이중 접속 FDDI 스테이션으로 구성 | 41 |
| 그림 2-4. 직렬(RS232) 하트비트 회선 | 42 |
| 그림 2-5. 고가용성을 위해 연결된 미러링된 디스크 | 48 |
| 그림 2-6. 고가용성 디스크 어레이를 사용하는 클러스터 | 49 |
| 그림 2-7. 파이버 채널 스위치로 연결된 디스크 어레이를 사용하는 클러스터 | 50 |
| 그림 2-8. 다른 공유 버스의 루트 디스크 | 51 |
| 그림 2-9. 다른 공유 버스의 기본 디스크 및 미러 | 52 |
| 그림 2-10. 8개의 노드로 구성된 활성/대기 클러스터 | 55 |
| 그림 2-11. XP 또는 EMC 디스크 어레이를 사용한 8개의 노드로 구성된 클러스터 | 56 |
| 그림 3-1. Serviceguard 소프트웨어 구성 요소 | 58 |
| 그림 3-2. 잠금 디스크 작동 | 67 |
| 그림 3-3. 쿼럼 서버의 작동 방식 | 69 |
| 그림 3-4. 장애 조치 중 패키지 이동 | 72 |
| 그림 3-5. 패키지 전환 전 | 74 |
| 그림 3-6. 패키지 전환 후 | 75 |
| 그림 3-7. 장애 조치 이전의 순환 대기 구성 | 77 |
| 그림 3-8. 장애 조치 이후의 순환 대기 구성 | 77 |
| 그림 3-9. 장애 조치 후의 CONFIGURED_NODE 정책 패키지 | 78 |
| 그림 3-10. 장애 조치 이전의 자동 장애 복구 구성 | 79 |
| 그림 3-11. 장애 조치 이후의 자동 장애 복구 구성 | 80 |
| 그림 3-12. 노드 1이 다시 시작된 후의 자동 장애 복구 구성 | 80 |
| 그림 3-13. 중요 사건별로 본 패키지 작동 과정 | 86 |
| 그림 3-14. 실행 스크립트 실행 중의 패키지 동작 과정 | 87 |
| 그림 3-15. 중지 스크립트 실행 중의 패키지 동작 과정 | 92 |
| 그림 3-16. 로컬 네트워크 전환 이전의 클러스터 | 100 |
| 그림 3-17. 로컬 네트워크 전환 이후의 클러스터 | 101 |
| 그림 3-18. 케이블 장애 후의 로컬 전환 | 102 |
| 그림 3-19. 통합된 네트워크 포트 | 104 |

그림

| | |
|---|-----|
| 그림 3-20. 공유 저장 장치의 물리 디스크 | 109 |
| 그림 3-21. 미러링된 물리 디스크 | 109 |
| 그림 3-22. 볼륨 그룹에 구성된 여러 장치 | 110 |
| 그림 3-23. LUN에 결합된 물리 디스크 | 110 |
| 그림 3-24. LUN에 대한 다중 경로 | 111 |
| 그림 3-25. 볼륨 그룹의 다중 경로 | 112 |
| 그림 4-1. 클러스터 구성 예 | 127 |
| 그림 7-1. 상태 확인: Serviceguard Manager 맵 | 276 |
| 그림 7-2. 상태 확인: Serviceguard Manager 속성 시트 | 277 |
| 그림 8-1. 인라인 단자가 있는 F/W SCSI-2 버스 | 328 |
| 그림 E-1. 롤링 업그레이드 전의 클러스터 실행 | 394 |
| 그림 E-2. 노드 2로 이동한 패키지를 포함하는 클러스터 실행 | 395 |
| 그림 E-3. 노드 1을 HP-UX 11.00으로 업그레이드 | 395 |
| 그림 E-4. 노드 1이 클러스터에 다시 참가 | 396 |
| 그림 E-5. 노드 1로 이동한 패키지를 포함하는 클러스터 실행 | 396 |
| 그림 E-6. 업그레이드 후의 클러스터 실행 | 397 |
| 그림 H-1. 예제 1: 대기 구성의 IPv4 및 IPv6 주소 | 434 |
| 그림 H-2. 예제 1: 대기 LAN으로 장애 조치된 후의 IPv4 및 IPv6 주소 | 434 |
| 그림 H-3. 예제 2: 대기 구성의 IPv4 및 IPv6 주소 | 435 |
| 그림 H-4. 예제 2: 대기 LAN으로 장애 조치된 후의 IPv4 및 IPv6 주소 | 435 |

발행 정보

표 1

| 발행 날짜 | 제품 번호 | 버전 |
|------------------------|-------------|---------------------------------|
| 1995년 1월 | B3936-90001 | 제1판 |
| 1995년 6월 | B3936-90003 | 제2판 |
| 1995년 12월 | B3936-90005 | 제3판 |
| 1997년 8월 | B3936-90019 | 제4판 |
| 1998년 1월 | B3936-90024 | 제5판 |
| 1998년 10월 | B3936-90026 | 제6판 |
| 2000년 12월 | B3936-90043 | 제7판 |
| 2001년 9월 | B3936-90056 | 제8판 |
| 2002년 3월 | B3936-90068 | 제9판 |
| 2003년 6월 | B3936-90070 | 제10판 |
| 2004년 6월 | B3936-90076 | 제11판 |
| 2004년 6월(2005년 6월 제2쇄) | B3936-90076 | 제11판, 제2쇄. 주로 보안 및 액세스 부분을 업데이트 |

마지막 발행 날짜와 제품 번호는 현재 버전을 나타내며, Serviceguard의 최신 버전은 A.11.16입니다.

새로운 버전의 설명서가 발행되면 발행 날짜가 변경됩니다. 미미한 수정 및 업데이트로 인한 재발행 시에는 발행 날짜가 변경되지 않습니다. 포괄적인 기술 변경 사항이 있으면 제품 번호가 개정됩니다.

새 발행본은 이전 발행본 이래로 업데이트된 모든 내용을 담고 있습니다.

HP 출판 담당 부서:

*Infrastructure Solutions Division
Hewlett-Packard Co.
19111 Pruneridge Ave.
Cupertino, CA 95014*

머리말

제2쇄에서는 제11판의 제1쇄에 주로 182페이지의 “보안 파일 편집”에 관한 새로운 내용을 추가하였습니다.

이 설명서에서는 HP-UX 운영 체제의 HP 9000 시리즈 800이나 HP Integrity 서버에서 실행될 수 있도록 Serviceguard를 구성하는 방법을 설명합니다. 그 내용은 다음과 같습니다.

- 1장, "Serviceguard 소개"에서는 Serviceguard 클러스터에 대해 설명하고 이 설명서를 사용하기 위한 정보를 제공합니다.
- 2장, "Serviceguard 하드웨어 구성 이해"에서는 Serviceguard에서 사용되는 하드웨어 구성에 대한 일반적인 관점을 제공합니다.
- 3장, "Serviceguard 소프트웨어 구성 요소 이해"에서는 Serviceguard의 소프트웨어 구성 요소를 설명하고 HP-UX 운영 체제에서 해당 구성 요소의 작동하는 방식을 보여 줍니다.
- 4장, "HA 클러스터 계획 및 문서화"에서는 계획 과정을 단계적으로 살펴하며 클러스터에 대한 정보를 구성하는 일련의 워크시트를 제공합니다.
- 5장, "HA 클러스터 구성 만들기"에서는 클러스터 구성 만들기를 설명합니다. 제11판의 제2쇄에는 182페이지의 “보안 파일 편집”에 대한 새로운 정보가 포함됩니다.
- 6장, "패키지 및 서비스 구성"에서는고가용성 패키지 및 이와 관련된 제어 스크립트 만드는 방법에 대해 설명합니다.
- 7장, "클러스터 및 패키지 유지 관리"에는 기본 클러스터 관리 작업이 나와 있습니다.
- 8장, 클러스터 문제 해결"에서는 클러스터 테스트 및 문제 해결 방법을 설명합니다.
- 부록 A, "Serviceguard 명령"는 Serviceguard에서 사용하는 명령을 나열하고 각 맨페이지의 요약 정보를 다시 수록합니다.
- 부록 B, "Enterprise Cluster Master Toolkit"에서는 특별한 클러스터 구성을 만드는 데 사용할 수 있는 도구 그룹을 설명합니다.

- 부록 C, "고가용성 클러스터 응용 프로그램 설계"에서는 **Serviceguard** 환경에서 최적의 성능을 제공할 수 있는 클러스터 인식 응용 프로그램을 만들기 위한 지침을 제공합니다.
- 부록 D, "HA 응용 프로그램과 **Serviceguard** 통합"에서는 기존 응용 프로그램을 **Serviceguard**와 통합하는 방법을 제공합니다.
- 부록 E, "롤링 소프트웨어 업그레이드"에서는 응용 프로그램을 종료하지 않고 한 **Serviceguard**나 **HP-UX** 릴리즈에서 다른 **Serviceguard**나 **HP-UX** 릴리즈로 이동하는 방법을 보여 줍니다.
- 부록 F, "계획 워크시트 양식"에는 **Serviceguard** 구성을 준비할 수 있는 일련의 빈 워크시트가 들어 있습니다.
- 부록 G, "LVM에서 **VxVM** 데이터 저장 장치로 마이그레이션"에서는 **LVM** 데이터 저장소를 **VxVM** 데이터 저장소로 변환하는 방법을 설명합니다.
- 부록 H, "IPv6 네트워크 지원"에서는 **IPv6** 주소 지정 체계 및 지원되는 기본 인터페이스 및 대기 인터페이스 구성에 대해 설명합니다.

관련 서적

다음 문서에서 유용한 정보를 추가로 얻으실 수 있습니다.

- <http://www.docs.hp.com> High Availability에서:
 - *Clusters for High Availability: a Primer of HP Solutions*. Hewlett-Packard Professional Books: Prentice Hall PTR, 2001 (ISBN 0-13-089355-2)
 - *Managing HP Serviceguard for Linux, Fifth Edition*, (B9903-90046) 2005년 5월
 - *Designing Disaster Tolerant High Availability Clusters* (B7660-90009)
 - **Serviceguard OPS Edition**으로 OPS 클러스터 구성(B5158-90043)
 - *HP Serviceguard Extension for Faster Failover, Version A.01.00, Release Notes*, 2004년 6월(T2389-90001)
 - *Managing Serviceguard Extension for SAP* 2004년 12월(T2357-90007)
 - **Enterprise Cluster Master Toolkit 버전 B.02.20 릴리즈 노트**(HP-UX 11iv1) 2004년 12월(T1909-90021)

— **Enterprise Cluster Master Toolkit 버전 B.02.21 릴리즈 노트**(HP-UX 11iv2) 2004년 12월(T1909-90026)

— *Using High Availability Monitors* (B5736-90042)

— *Using the Event Monitoring Service* (B7609-90022)

— *Managing Highly Available NFS* (B5140-90017)

— *HP Serviceguard Quorum Server Release Notes*(B8467-90026)

<http://www.docs.hp.com> Networking and Communications에서:

— *HP Auto Port Aggregation (APA) Support Guide*

— *HP Auto Port Aggregation Release Notes* 및 기타 Auto Port Aggregation documents

<http://www.docs.hp.com> HP-UX Operating Environments에서:

— **시스템 및 작업 그룹 관리**(5990-8178)

<http://www.docs.hp.com/en/hpux11.0.html>에서:

— *Using Advanced Tape Services*(B3936-90032)

Serviceguard에서 VxVM 저장소를 사용하기 전에 다음을 참조하십시오.

- VERITAS Volume Manager Administrator's Guide. VERITAS 용어를 해설합니다.
- VERITAS Volume Manager Storage Administrator Administrator's Guide
- VERITAS Volume Manager Reference Guide
- VERITAS Volume Manager Migration Guide
- VERITAS Volume Manager for HP-UX Release Notes

다음 URL을 통해 HP 고가용성 웹 페이지에 액세스할 수 있습니다.

- <http://www.hp.com/go/ha>

다음 URL을 통해 다양한 HP-UX 설명서에 액세스할 수 있습니다.

- <http://docs.hp.com>(영문 설명서)
- <http://docs.hp.com/ko>(한글 설명서)

문제점 보고

소프트웨어나 설명서에 문제가 있을 경우 **HP** 지사나 고객 서비스 센터에 문의하십시오.

1 Serviceguard 소개

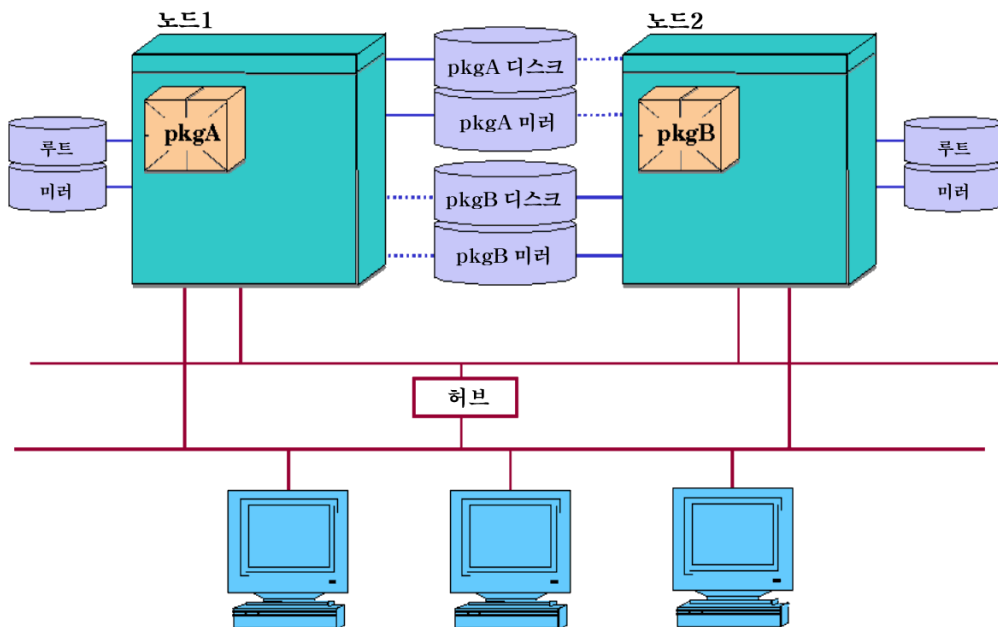
이 장에서는 HP-UX용 Serviceguard에 대해 소개하고 이 설명서에서 다른 정보를 찾을 수 있는 위치를 설명합니다. 이 장에 포함된 내용은 다음과 같습니다.

- Serviceguard의 개념
- Serviceguard Manager 사용
- 클러스터 및 패키지 구성 정보

Serviceguard 클러스터의 설정을 시작할 준비가 되었으면 이 장을 건너뛰고 4장 “HA 클러스터 계획 및 문서화”의 내용을 읽으십시오. 자세한 설정 단계는 5장 “HA 클러스터 구성 만들기”를 참조하십시오.

그림 1-1에서는 두 개의 노드로 구성된 일반적인 Serviceguard 클러스터를 보여 줍니다.

그림 1-1 일반적인 클러스터 구성



Serviceguard의 개념

Serviceguard를 사용하여 **HP 9000** 또는 **HP Integrity** 서버의 고가용성 클러스터를 만들 수 있습니다. **고가용성** 컴퓨터 시스템에서는 하드웨어나 소프트웨어에 장애가 발생해도 응용 프로그램 서비스를 계속 실행할 수 있습니다. 고가용성 시스템은 소프트웨어 장애뿐만 아니라 **SPU(System Processing Unit)**, 디스크 또는 **LAN(Local Area Network)** 구성 요소의 장애로부터 사용자를 보호합니다. 구성 요소 중 하나에서 장애가 발생하면 중복 구성 요소가 대신 작업을 수행합니다. **Serviceguard**와 다른 고가용성 하위 시스템에서 구성 요소 간의 전달을 조정합니다.

Serviceguard 클러스터는 네트워크로 연결된 **HP 9000** 또는 **HP Integrity** 서버(노드라고 하는 호스트 시스템)의 그룹입니다. 이 클러스터에는 소프트웨어와 하드웨어가 충분히 중복되어 있으므로 **단순 장애 요인(Single Point of Failure)**이 발생해도 서비스에 심각한 장애를 일으키지 않습니다. 응용 프로그램 서비스(개별 **HP-UX** 프로세스)는 **패키지** 단위로 그룹화됩니다. 단일 서비스, 노드, 네트워크 또는 다른 리소스에서 장애가 발생해도 **Serviceguard**에서 패키지의 제어권을 클러스터 내의 다른 노드에 자동으로 전달할 수 있으므로 중단을 최소화하면서 서비스의 가용성을 유지할 수 있습니다.

그림 1-1에서 노드 1(두 개의 **SPU** 중 하나)은 패키지 **A**를 실행하고 노드 2는 패키지 **B**를 실행합니다. 각 패키지에는 별도의 디스크 그룹이 연관되어 있으며 이러한 디스크에는 패키지의 응용 프로그램에서 필요로 하는 데이터와 이 데이터의 미러 사본이 들어 있습니다. 두 노드는 모두 미러링되는 디스크의 두 그룹에 물리적으로 연결되어 있습니다. 그러나 한 번에 하나의 노드만이 주어진 디스크 그룹의 데이터에 액세스할 수 있습니다. 그림에서 노드 1은 위쪽에 있는 두 디스크에 단독으로 액세스하고 있으며(실선) 노드 2는 위쪽의 디스크에 액세스하지 않고 연결되어 있습니다(점선). 마찬가지로 노드 2는 아래쪽에 있는 두 디스크에 단독으로 액세스하고 있으며(실선) 노드 1은 아래쪽의 디스크에 액세스하지 않고 연결되어 있습니다(점선).

데이터의 미러 사본은 디스크 장애를 대비한 중복 기능을 제공합니다. 이밖에도 노드 1과 노드 2에 연결된 디스크의 데이터 버스는 총 4개입니다. 이 구성에서는 각 패키지가 서로 다른 버스를 사용하므로 최대의 중복성과 최적의 입출력 성능을 얻을 수 있습니다.

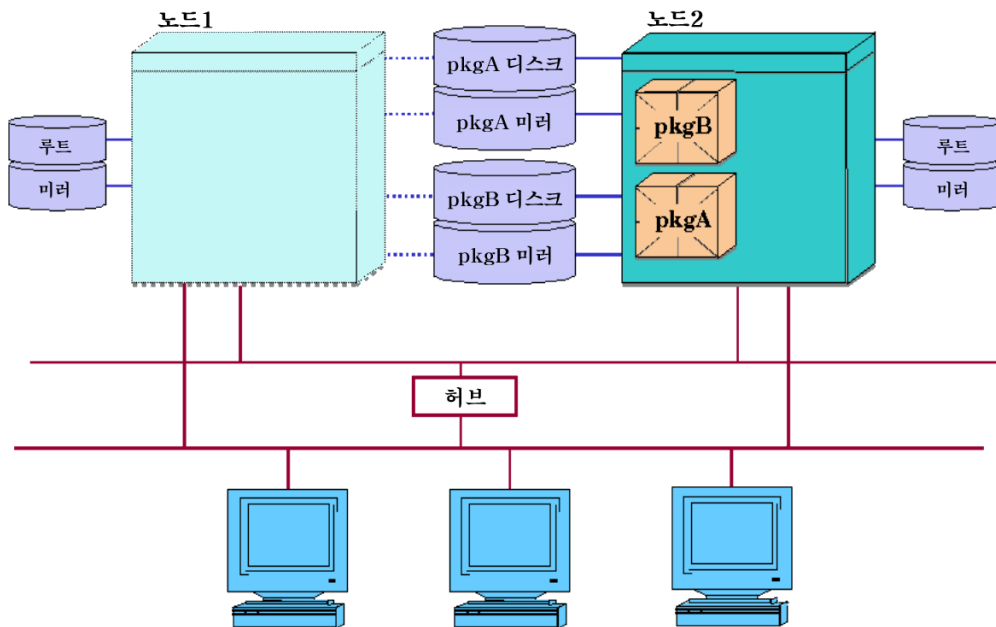
네트워크 하드웨어는 각 노드에서 중복 **LAN** 인터페이스를 제공하도록 연결되어 있습니다. **Serviceguard**에서는 클러스터 노드 간의 안정적인 통신을 위해 **TCP/IP** 네트워크 서비스를 사용합니다. 노드 간의 통신에는 각 작동 노드가 보내는 신호이며 클러스

터 운영의 핵심 역할을 하는 **하트비트 메시지**의 전송도 포함됩니다. TCP/IP 서비스는 다른 종류의 노드 간 통신에도 사용됩니다. 하트비트에 대한 자세한 내용은 “Serviceguard 소프트웨어 이해” 장을 참조하십시오.

장애 조치

일반적인 조건에서 완전히 운영되고 있는 Serviceguard 클러스터는 각각의 노드에서 패키지가 실행되는 동안 클러스터 구성 요소의 상태를 모니터링하는 역할만 합니다. Serviceguard 클러스터에서 실행되는 호스트 시스템을 **활성 노드**라고 합니다. 패키지를 만들 때는 하나의 **기본 노드**와 하나 이상의 **대체 노드**를 지정합니다. 노드나 노드의 네트워크 통신에 장애가 발생하면 Serviceguard에서는 패키지의 제어권을 다음으로 사용 가능한 대체 노드에 전달할 수 있습니다. 그림 1-2에서는 이러한 경우를 보여 줍니다.

그림 1-2 장애 조치 후의 일반적인 클러스터



제어권이 전달된 후 패키지는 대체 노드가 계속 실행되는 동안 대체 노드에 남아 있습니다. 그러나 필요하다면 기본 노드가 다시 온라인 상태로 되는 즉시 기본 노드로 되돌아가도록 패키지를 구성할 수 있습니다. 또는 적절한 때에 수동으로 패키지의 제어권을 기본 노드에 다시 반환할 수 있습니다.

그림 1-2에서는 클러스터의 전원 연결이 표시되지 않았지만 전원 연결도 중요한 요소입니다. 클러스터에서 단순 장애 요인을 모두 제거하려면 노드, 디스크 및 디스크 미러의 단순 장애 요인을 방지하는 데 필요한 만큼의 개별 전원 회로를 포함해야 하며, 각 전원 회로는 무정전 전원으로 보호해야 합니다. 자세한 내용은 4장 “HA 클러스터 계획 및 문서화”의 “전원 공급 계획” 절을 참조하십시오.

Serviceguard는 다른고가용성 제품과 함께 사용할 수 있도록 설계되었습니다. 이러한 제품에는 디스크 하위 시스템의 단순 장애 요인을 제거할 수 있도록 디스크 중복 기능을 제공하는 MirrorDisk/UX나 VERITAS Volume Manager, Serviceguard에서 직접 처리하지 않는 장애를 모니터링하여 찾아낼 수 있는 Event Monitoring Service(EMS), 데이터 보호를 위해 다양한 RAID 수준을 사용하는 디스크 어레이, HP PowerTrust와 같이 정전과 관련된 장애를 제거하는 HP 지원 UPS(무정전 전원 공급 장치) 등이 있습니다. 가용성을 최대한 높이려면 Serviceguard와 이러한 제품을 함께 사용하는 것이 좋습니다.

Serviceguard Manager 사용

Serviceguard Manager는 Serviceguard의 그래픽 사용자 인터페이스입니다.

Serviceguard Manager 관리 스테이션은 HP-UX, Linux 및 Windows 시스템일 수 있으며, 여기에서 HP-UX 또는 Linux의 Serviceguard 클러스터를 모니터링하고 관리하고 구성할 수 있습니다.

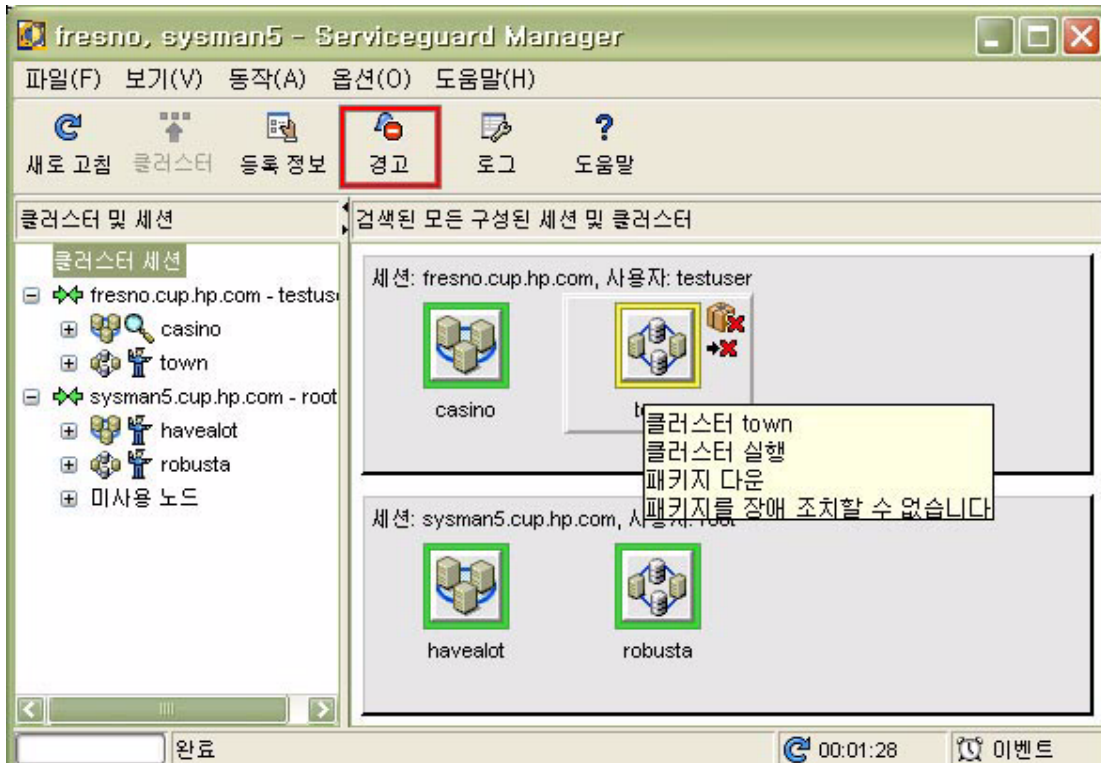
- 모니터링: 서버넷의 Serviceguard 객체에 대한 정보를 볼 수 있습니다. 이 객체는 그래픽 맵에서 계층 구조 트리로 표시됩니다. 자세한 내용은 객체의 속성에서 볼 수 있습니다.
- 관리: Serviceguard Manager를 통해 클러스터 및 패키지의 실행이나 중지 같은 관리 작업을 수행할 수 있습니다. 이 클러스터에는 Serviceguard 버전 A.11.12 이상이 설치되어 있어야 합니다.
- 구성: 클러스터 및 패키지 구성을 만들거나 수정할 수 있습니다. 이 클러스터에는 Serviceguard 버전 A.11.16 이상이 설치되어 있어야 합니다.

Serviceguard Manager에 대한 자세한 내용은 <http://docs.hp.com> -> high availability -> Serviceguard에 게시된 최신 **Serviceguard Manager 릴리즈 노트**를 참조하십시오. Serviceguard Manager는 Serviceguard 디스크와 함께 제공되는 Serviceguard Distributed Components 디스크나 <http://software.hp.com>에서 무료로 다운로드할 수 있습니다. 한글 매뉴얼은 다음 웹 사이트에서 찾을 수 있습니다. <http://docs.hp.com/ko/allproducts.html>

Serviceguard Manager를 사용한 모니터링

관리 스테이션에서 세션 서버에 연결한 다음 세션 서버에서 연결할 수 있는 모든 클러스터를 표시하도록 요청하거나 특정 클러스터를 나열할 수 있으며, 서버넷의 모든 미사용 노드, 즉 현재 어떤 클러스터 구성에도 포함되지 않은 노드를 표시하도록 요청할 수도 있습니다.

그림 1-3 Serviceguard Manager를 사용한 모니터링

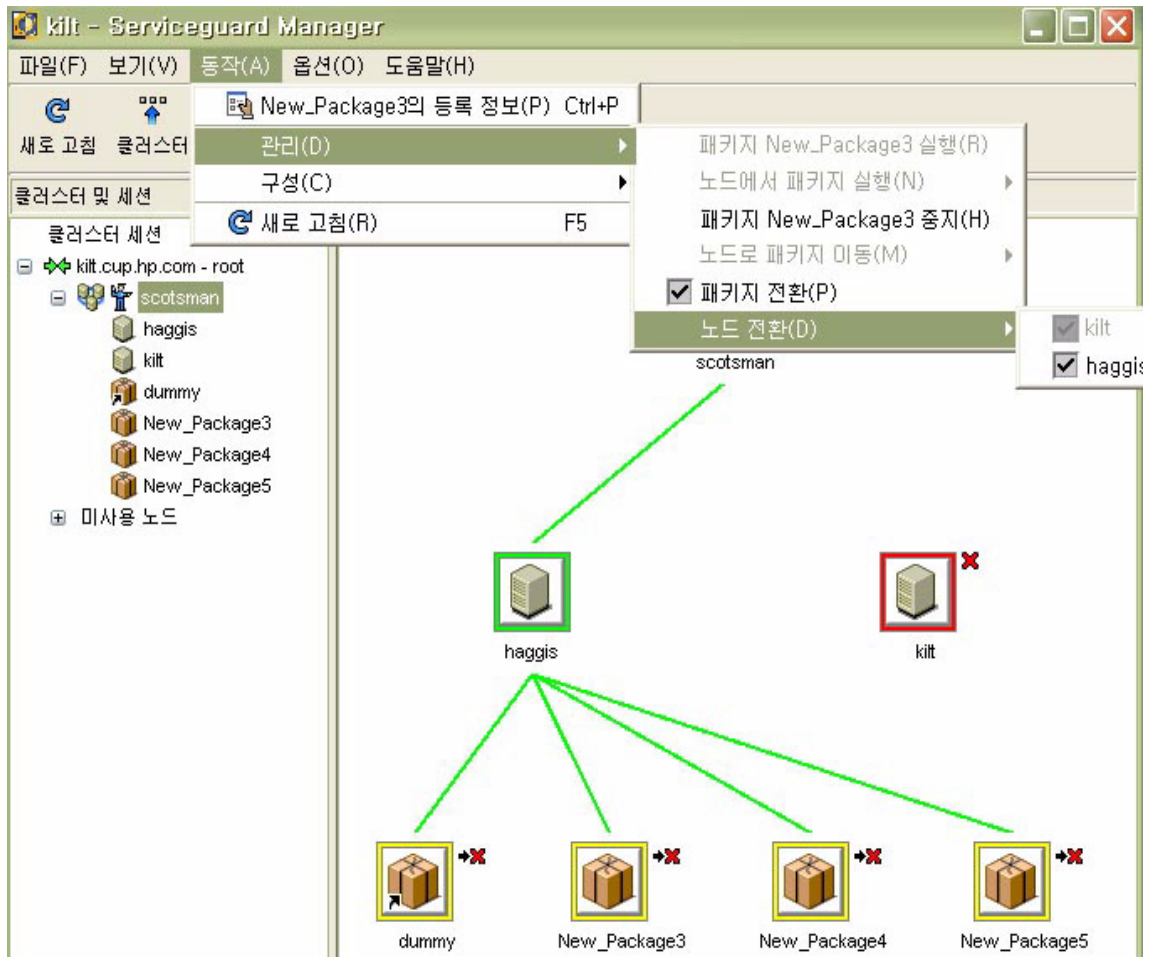


Serviceguard Manager를 사용한 관리

적절한 액세스 권한(Serviceguard A.11.14 및 A.11.15) 또는 액세스 제어 정책 (Serviceguard A.11.16)이 있는 경우 클러스터, 노드 및 패키지를 관리할 수 있습니다.

- 클러스터: 중지, 실행
- 클러스터 노드: 중지, 실행
- 패키지: 중지, 실행, 노드 간 이동, 노드 전환 및 패키지 전환 플래그 재설정

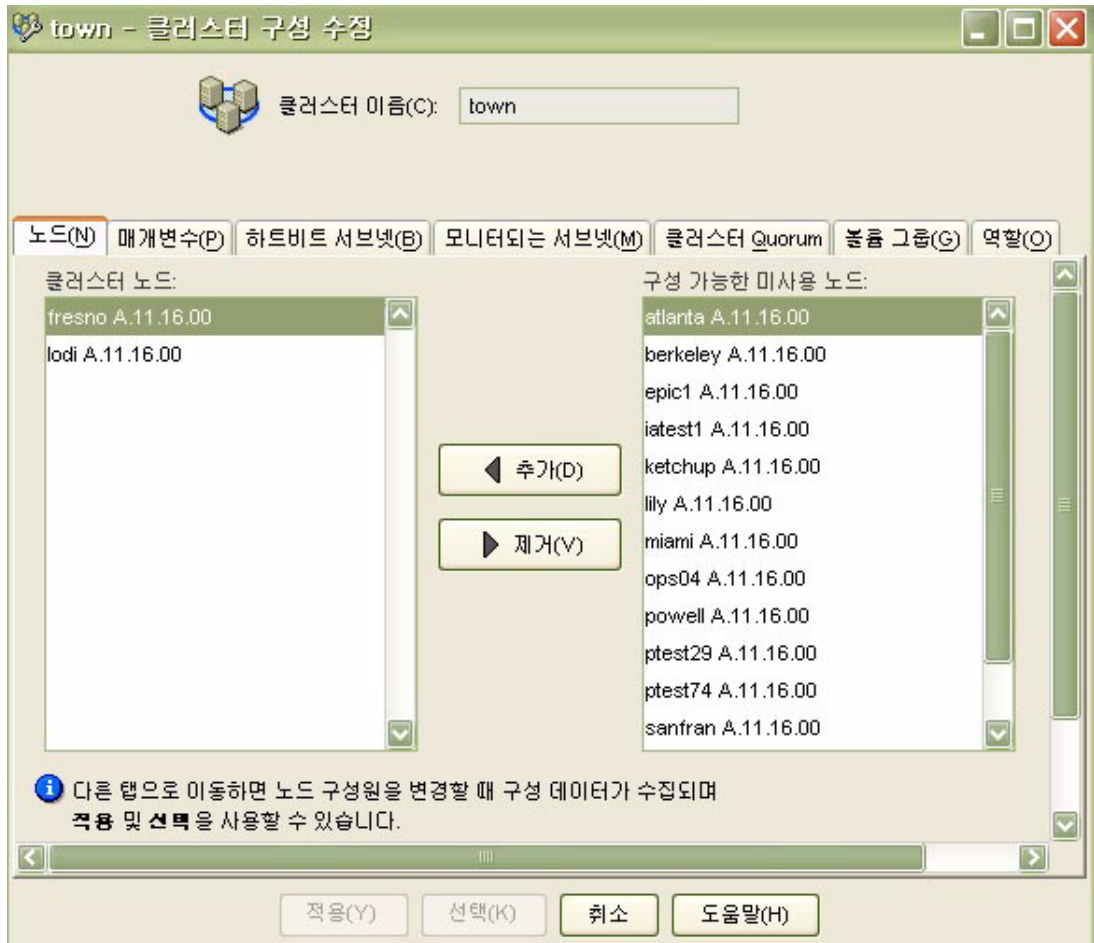
그림 1-4 Serviceguard Manager 패키지 관리



Serviceguard Manager를 사용한 구성

Serviceguard 버전 A.11.16을 사용하는 경우 클러스터와 패키지를 구성할 수도 있습니다. 단, 서버 노드와 대상 클러스터 모두에 Serviceguard 버전 A.11.16이 설치되어 있고 클러스터 노드에 대한 루트(UID=0) 로그인 권한이 있어야 합니다.

그림 1-5 Serviceguard Manager를 사용한 구성



Serviceguard Manager 도움말

온라인 도움말을 보려면 화면 위쪽에 있는 “도움말” 메뉴 항목을 클릭하십시오.

“Serviceguard Manager 사용” 아래에 있는 다음 도움말 항목은 이 인터페이스를 처음 사용하는 사용자에게 특히 유용합니다.

- “메뉴 및 도구 모음 명령”
- “Serviceguard Manager 탐색”
- “맵 범례”

Serviceguard Manager의 작동 방식

Unix 또는 Linux 관리 스테이션에서 Serviceguard Manager를 시작하려면 `sgmgr` 명령을 입력합니다. 명령줄에서 옵션을 입력하거나 인터페이스가 열린 후 대화 상자에서 옵션을 입력할 수 있습니다. 명령 구문 및 옵션에 대한 내용을 보려면 “온라인 도움말” -> “문제 해결”을 참조하거나 명령줄에 `man sgmgr`를 입력하십시오.

Windows 관리 스테이션에서 Serviceguard Manager를 시작하려면 바탕 화면에서 해당 아이콘을 두 번 클릭합니다. 실제로 사용되는 명령을 보거나 변경하려면 해당 아이콘을 마우스 오른쪽 단추로 클릭하고 등록 정보(또는 속성)를 선택합니다. 명령 구문 및 옵션에 대한 내용은 “온라인 도움말” -> “문제 해결”을 참조하십시오.

저장된 “스냅샷” 클러스터 파일을 열려면 `sgm` 확장자를 포함한 파일 이름을 지정합니다. 단, 해당 파일 및 디렉토리를 볼 수 있는 권한이 있어야 합니다. 예제 파일을 보면 Serviceguard Manager에 쉽게 익숙해질 수 있습니다.

“작동 중인” 클러스터를 보려면 Serviceguard 노드의 COM(Cluster Object Manager) 데몬에 연결합니다. COM은 Serviceguard와 함께 자동으로 설치됩니다. 이 노드는 세션 서버가 되며 서버넷을 통해 다른 Serviceguard 노드의 COM과 연결할 수 있습니다. 세션 서버는 관리 스테이션의 명령을 대상 노드에 중계하고 대상 노드의 구성 및 상태 데이터를 다시 관리 스테이션에 중계합니다. 또한 대상 노드의 작업 메시지도 관리 스테이션에 중계합니다.

연결하려면 세션 서버의 `/etc/passwd` 파일에서 올바른 사용자 이름과 암호를 지정해야 합니다. 그런 다음 보려는 클러스터를 나열합니다. 현재 클러스터에 구성되지 않았지만 Serviceguard가 설치되어 있는 노드를 보려면 “미사용 노드”를 클릭합니다.

세션 서버가 클러스터에서 정보를 얻기 위해서는 대상 클러스터에서 세션 서버의 액세스를 허용해야 합니다. 대상 노드는 `/etc/hosts` 또는 DNS를 통해 세션 서버의 호스트 이름을 확인합니다. Serviceguard 버전 A.11.16에서는 액세스 방법 및 비 루트 역할이 다음과 같이 변경되었습니다.

- **Serviceguard** 버전 **A.11.16** 클러스터에서 클러스터 구성 파일 또는 패키지 구성 파일에는 액세스 제어 정책이 있어야 하고 이 정책에는 의도한 사용자 및 **COM** 서버 호스트 이름과 최소한 모니터링 역할이 포함되어야 합니다.
- 이전 버전의 **Serviceguard**에서는 `/etc/cmcluster/cmclnodelist/` 파일에 **COM** 호스트 노드와 사용자 루트 쌍이 있어야 합니다. 이전 버전의 **Serviceguard**에 대한 자세한 내용은 <http://docs.hp.com> (영문 설명서) 및 <http://docs.hp.com/ko> (한글 설명서)에 게시된 해당 설명서 및 릴리즈 노트를 참조하십시오.

액세스 정책에 대한 자세한 내용은 182페이지의 “보안 파일 편집”을 참조하십시오.

Serviceguard 버전 **A.11.16**에서는 클러스터의 노드에 루트(**UID=0**) 암호를 지정한 경우 **Serviceguard Manager**를 사용하여 클러스터 및 클러스터에 대한 패키지 구성을 만들거나 수정할 수 있습니다.

클러스터 및 패키지 구성 정보

이 설명서에서는 Serviceguard를 사용하여 HA 클러스터를 만들기 위해 수행해야 하는 작업에 대해 설명합니다. 그림 1-6에서는 이러한 작업을 보여 줍니다.

그림 1-6 Serviceguard 클러스터 구성 작업

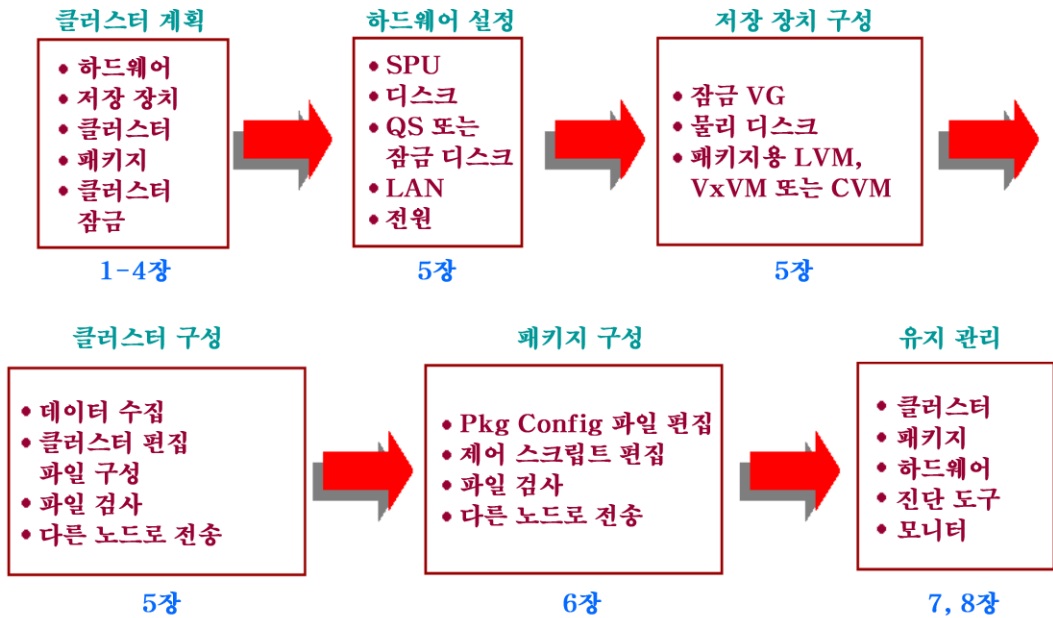


그림 1-6에 표시된 작업은 4장~7장에 단계별로 자세히 설명되어 있습니다. **시작하기 전에** 구성에 필요한 모든 데이터를 수집하는 것이 좋습니다. 데이터 수집 방법에 대한 자세한 내용은 4장 “HA 클러스터 계획 및 문서화”를 참조하십시오.

Serviceguard 소개

클러스터 및 패키지 구성 정보

2 Serviceguard 하드웨어 구성 이해

이 장에서는 Serviceguard 하드웨어 구성 요소의 동작 방식에 대해 간단히 설명합니다. 이 장에 포함된 내용은 다음과 같습니다.

- 클러스터 구성 요소의 중복
- 중복 네트워크 구성 요소
- 중복 디스크 저장 장치
- 중복 전원 공급 장치
- 큰 규모의 클러스터

Serviceguard 소프트웨어 구성 요소에 대한 내용은 다음 장을 참조하십시오.

클러스터 구성 요소의 중복

일반적인 클러스터에서는 높은 수준의 가용성을 제공하기 위해 두 개 이상의 SPU 및 두 개 이상의 독립된 디스크와 같은 중복 시스템 구성 요소를 사용합니다. 이러한 중복 시스템 구성 요소는 단순 장애 요인(Single Point of Failure)을 없애 줍니다. 일반적으로 중복 시스템 구성 요소가 많을수록 장애가 발생할 때 액세스할 수 있는 응용 프로그램, 데이터 및 지원 서비스가 늘어납니다.

하드웨어 중복 외에, 장애가 발생할 때 응용 프로그램을 다른 SPU나 네트워크로 전달할 수 있도록 하고 이를 제어하는 소프트웨어 지원도 필요합니다. Serviceguard에서는 다음과 같은 소프트웨어 지원을 제공합니다.

- LAN에 장애가 발생하면 Serviceguard는 대기 LAN으로 전환되거나 영향을 받은 패킷지를 대기 노드로 이동시킵니다.
- SPU에 장애가 발생하면 응용 프로그램은 지체 없이 장애가 발생한 SPU에서 사용 가능한 SPU로 자동 전달됩니다.
- 디스크 인터페이스와 같이 모니터링되는 다른 리소스에 장애가 발생하면 패킷지를 다른 노드로 이동할 수 있습니다.
- 소프트웨어 장애가 발생하면 작업 방해를 최소화하면서 같은 노드나 다른 노드에서 응용 프로그램을 다시 시작할 수 있습니다.

또한 Serviceguard에서는 시스템 관리, 유지 보수, 버전 업그레이드 등의 이유로 SPU를 중지해야 할 때 다른 SPU로 응용 프로그램의 제어권을 쉽게 전달할 수 있습니다.

현재 Serviceguard 클러스터에서 지원되는 최대 노드 수는 16개입니다. Fast/Wide SCSI 디스크 또는 디스크 어레이는 공유(다중 초기자) 버스에서 동시에 최대 4개의 노드에 연결될 수 있습니다. HP SureStore XP 시리즈 및 EMC Symmetrix와 같이 파이버 채널을 사용하지만 공유 버스를 사용하지 않는 디스크 어레이는 동시에 16개의 노드 모두에 연결될 수 있습니다.

패킷지 장애 조치에 관한 지침은 클러스터의 디스크 기술 유형에 따라 달라집니다. 예를 들어, Fast/Wide SCSI 디스크 또는 디스크 어레이의 데이터에 액세스하는 패키지에서는 최대 4개의 노드로 장애 조치가 수행될 수 있습니다. 패키지가 클러스터에 있는 디스크의 데이터에 액세스할 때 이 클러스터가 파이버 채널이나 HP SureStore XP 또는 EMC Symmetrix 디스크 기술을 사용한다면 16개의 노드에서 장애 조치를 수행하도록 패키지를 구성할 수 있습니다.

공유 버스에 있는 디스크의 데이터에 액세스하지 **않는** 패키지는 디스크 기술에 관계없이 클러스터에서 구성된 모든 노드에서 장애 조치를 수행하도록 구성될 수 있습니다. 예를 들어, 로컬 실행 파일만 실행하는 패키지는 디스크의 연결 유형에 관계없이 해당 실행 파일의 로컬 사본이 있는 클러스터의 모든 노드에서 장애 조치를 수행하도록 구성될 수 있습니다.

중복 네트워크 구성 요소

네트워크에서 단순 장애 요인을 없애려면 클러스터 노드가 액세스하는 각 서버넷에 중복 네트워크 인터페이스가 있어야 하며, 케이블 장애가 발생했을 때 보호하기 위해 중복 케이블도 있어야 합니다. 각 인터페이스 카드는 서로 다른 케이블로 연결되고 각 케이블은 허브나 브리지 등의 구성 요소로 연결됩니다. FDDI 네트워크의 경우 각 인터페이스 카드는 케이블을 통해 다른 집중 장치에 연결됩니다. 브리지나 집중 장치 또는 스위치를 통해 서로 연결된 이러한 물리적 케이블 정렬을 **브리지 연결 네트워크**라고 합니다.

IP 주소는 브리지 연결 네트워크의 인터페이스와 연관될 수 있습니다. 연관된 IP 주소가 있는 인터페이스를 **기본 인터페이스**라고 하고 연관된 IP 주소가 없는 인터페이스는 **대기 인터페이스**라고 합니다. 대기 인터페이스는 기본 인터페이스에 장애가 발생했을 때 Serviceguard에서 전환용으로 사용할 수 있는 인터페이스입니다. Serviceguard는 기본 인터페이스에서 장애를 발견하면 장애가 발생한 인터페이스 카드의 IP 주소와 관련 연결을 사용 가능한 대기 인터페이스 카드로 전환합니다.

네트워크 구성에 대한 선택 옵션은 다음 절에서 자세하게 설명합니다. 지원되는 네트워크의 전체 목록은 사용 중인 제품 버전의 **Serviceguard 릴리즈 노트**를 참조하십시오.

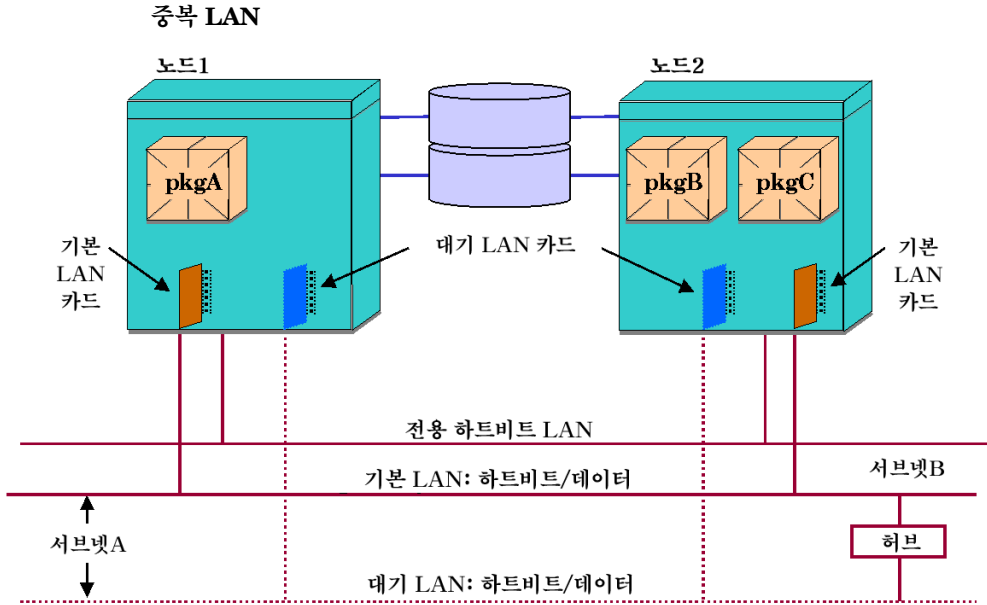
주

파이버 채널은 더 이상 하트비트나 데이터 LAN으로 지원되지 않습니다.

중복 이더넷 구성

그림 2-1에서는 이더넷으로 구성된 중복 네트워크 구성 요소의 사용 방법을 보여 줍니다. 토론 링도 비슷한 방식으로 구성됩니다.

그림 2-1



이 그림에서 두 개의 노드로 구성된 Serviceguard 클러스터에는 데이터/하트비트 서브넷(서브넷 A)을 위해 기본 및 대기 LAN 카드를 사용하여 구성한 하나의 브리지 연결 네트워크가 있습니다. 다른 LAN 카드는 선택적 전용 하트비트 LAN을 제공합니다. 기본 및 대기 LAN 세그먼트는 허브로 연결되어 중복 데이터/하트비트 서브넷을 제공합니다. 각 노드는 이 서브넷의 자체 IP 주소를 갖습니다. 데이터/하트비트 서브넷의 기본 LAN 카드에 장애가 발생하면 Serviceguard는 동일한 노드에 있는 대기 LAN 카드로 로컬 전환을 수행합니다.

중복 하트비트는 모두 하트비트를 전달하는 기본 LAN과 전용 LAN에서 제공됩니다. 그림 2-1에서 보는 것처럼, 전용 하트비트 LAN에는 이미 다른 서브넷을 통해 중복되는 경로가 있으므로 로컬 전환이 필요하지 않습니다. 기본 LAN에서 데이터가 정체되는 경우 전용 하트비트 LAN은 하트비트 장애의 오진단을 방지합니다. 각 노드는 전용 하트비트 LAN의 자체 IP 주소를 갖습니다.

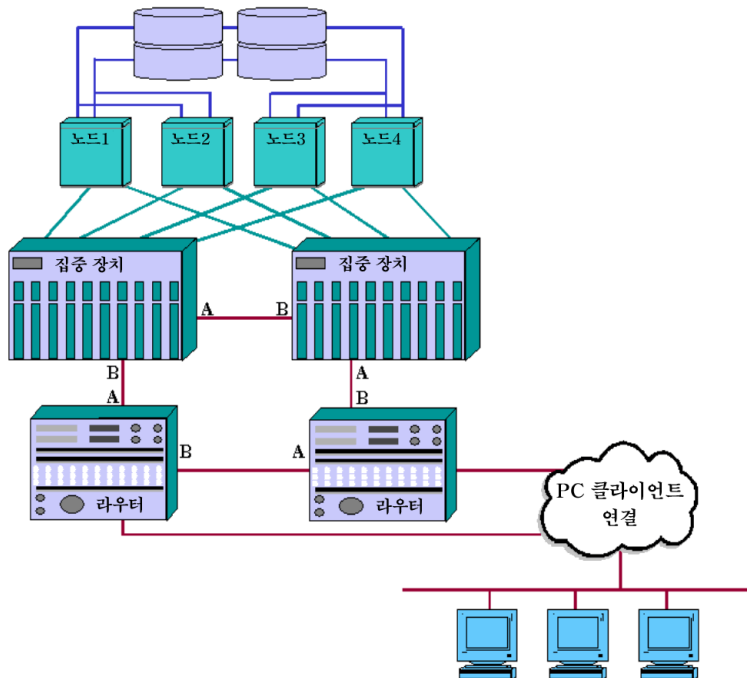
주 하트비트/데이터 LAN에서 네트워크 트래픽이 너무 많지는 않은지 확인해야 합니다. 트래픽이 너무 많으면 이 LAN은 전용 하트비트 LAN에 장애가 발생할 경우 하트비트 전송을 제대로 수행하지 못할 수 있습니다.

중복 FDDI 연결 제공

FDDI는 고속 파이버 상호 연결 매체입니다. FDDI를 사용하는 경우 스타 토폴로지를 사용하여 모든 노드를 두 개의 집중 장치에 연결함으로써 중복 구성을 만들 수 있습니다. 이때 두 개의 집중 장치는 클러스터 외부와 통신하는 두 개의 라우터와 연결됩니다. 이 경우 각 노드에서 두 개의 FDDI 카드를 사용하게 됩니다. 그림 2-2에서는 이러한 구성을 보여 줍니다. 포트 A에서 포트 B로 교차 연결된 이중 케이블을 통해 집중 장치가 서로 연결되어 있습니다. 라우터는 두 집중 장치에 모든 패킷을 보내도록 구성되어야 합니다.

그림 2-2

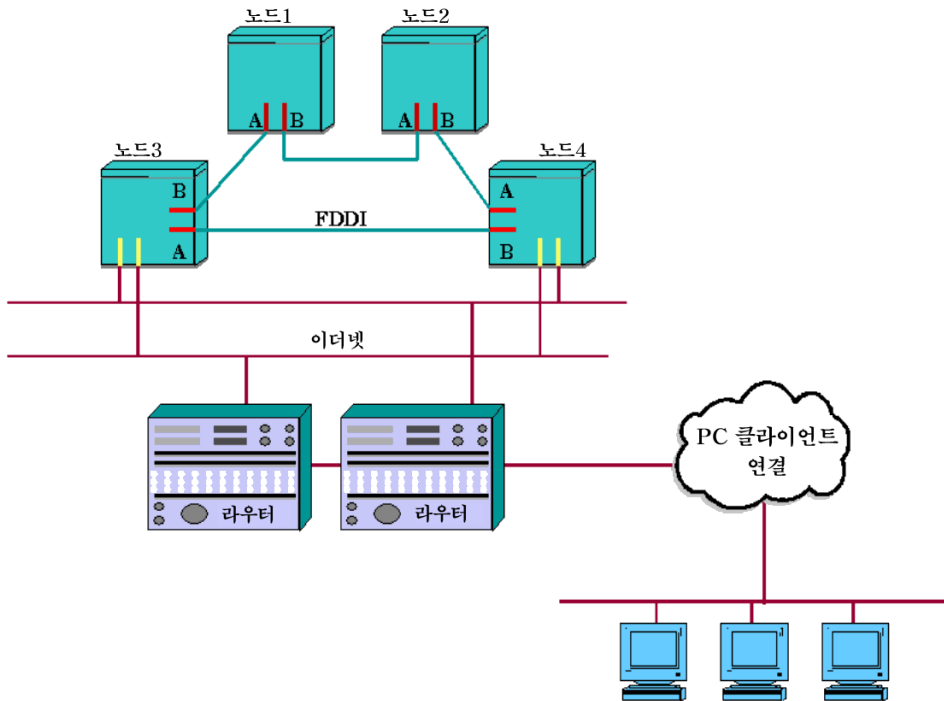
중복 FDDI 구성



이중 접속 FDDI 스테이션 사용

중복 FDDI 연결을 구성하는 또 다른 방법은 그림 2-3에 나와 있는 것처럼 각 노드의 이중 접속 스테이션을 구성하여 FDDI 링을 만드는 것입니다. 이 구성은 시스템 카드 케이스에서 슬롯이 한 개만 사용된다는 이점이 있습니다. 그림 2-3에서는 노드 3과 4도 이더넷을 사용하여 클러스터 외부와의 연결을 제공합니다.

그림 2-3 이중 접속 FDDI 스테이션으로 구성



이중 접속 카드를 사용하면 케이블과 커넥터에 장애가 발생했을 때는 보호를 받을 수 있으나 카드에 장애가 발생했을 때는 보호를 받지 못합니다. LAN 카드 장애가 발생하면 패키지가 다른 노드로 전환됩니다.

직렬(RS232) 하트비트 회선 사용

Serviceguard는 하트비트 전용으로 직렬(RS232) 통신을 사용하는 노드가 두 개인 구성을 지원합니다. 이 구성을 대체 하트비트 인터페이스로 선택하여 중복 하트비트 데이터를 제공할 수 있습니다.

주

직렬(RS232) 하트비트 회선은 노드가 두 개인 클러스터 구성에서만 사용할 수 있습니다. 직렬 하트비트 회선은 하트비트 LAN이 하나뿐이고 두 개의 노드로 구성된 클러스터에서 필수적입니다. 하트비트 LAN이 둘 이상이거나 하트비트 LAN이 하나이고 대기 LAN이 하나인 경우에는 직렬(RS232) 하트비트를 사용하지 않아야 합니다.

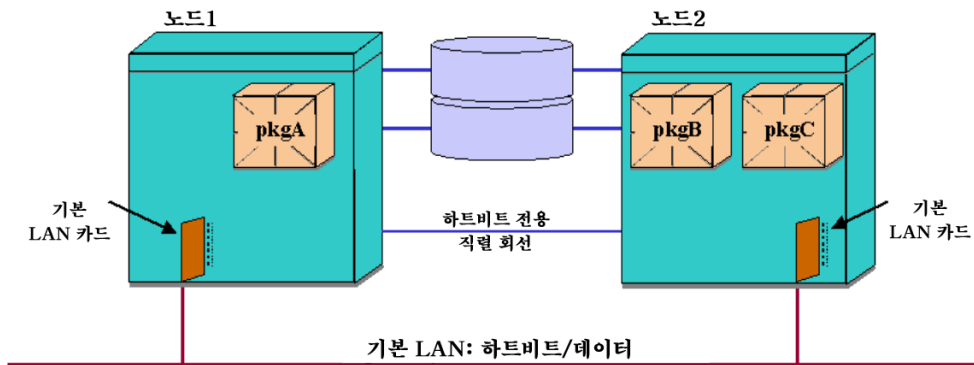
한 노드에서 하트비트 네트워크 카드에 장애가 발생하는 경우, 직렬 회선 하트비트가 있으면 정상 노드가 계속 작동하여 모든 패키지를 실행하는 동안 클러스터가 LAN 컨트롤러 카드 상태를 감지하고 네트워크 연결이 잘못된 노드를 중단시킬 수 있습니다.

중복 하트비트용으로 구성된 직렬(RS232) 회선이 있는 경우에도 하트비트 신호를 전달하는 데 하나의 LAN이 여전히 필요합니다. Serviceguard에서는 클러스터 구성원 사이의 통신에 TCP/IP가 필요하므로 직렬 회선 하트비트는 네트워크 포화 상태에서 보호책이 되지만 네트워크 장애 시에는 그렇지 않습니다.

직렬(RS232) 회선은 이더넷이나 FDDI 같은 TCP/IP 프로토콜을 실행하는 네트워크 카드에 비해 기본적으로 신뢰성이 떨어집니다. TCP/IP와 달리, 직렬 회선 프로토콜에는 오류 수정이나 재시도 메커니즘이 없습니다. 또한 직렬 회선은 표준이 없어서 구성하기가 복잡하고 어려울 수 있습니다.

그림 2-4에서는 직렬(RS232) 하트비트 회선을 보여 줍니다.

그림 2-4 직렬(RS232) 하트비트 회선



장애가 발생한 네트워크 카드의 교체

일부 시스템 구성에서는 장애가 발생한 네트워크 카드를 클러스터 실행 중에 교체할 수 있습니다. 교체 방법에 대해서는 “클러스터 문제 해결” 장의 “LAN 카드 교체” 절을 참조하십시오.

중복 디스크 저장 장치

클러스터의 각 노드는 자체 루트 디스크를 가질 뿐 아니라 여러 개의 다른 디스크에도 물리적으로 연결됩니다. 이때 두 개 이상의 노드가 데이터 및 해당 데이터의 패키지와 연관된 프로그램에 액세스할 수 있는 방식으로 연결됩니다. 이 액세스는 **Logical Volume Manager(LVM)**, **VERITAS Volume Manager(VxVM)** 또는 **VERITAS Cluster Volume Manager(CVM)**와 같은 저장 장치 관리자에서 제공됩니다. 디스크 저장 장치 그룹은 한 번에 하나의 노드에 의해서만 활성화될 수 있지만 패키지를 이동할 때는 대체 노드를 통해 저장 장치 그룹을 활성화할 수 있습니다. 패키지가 소유하는 저장 장치 그룹의 모든 디스크가 원래의 노드와 해당 패키지에서 사용 가능한 모든 대체 노드에 연결되어야 합니다. 디스크 저장 장치는 **RAID**나 소프트웨어 미러링을 사용하여 중복 구성됩니다.

지원되는 디스크 인터페이스

둘 이상의 노드에 연결되는 디스크(공유 데이터 디스크)를 위해 **Serviceguard**에서 지원하는 인터페이스의 종류는 다음과 같습니다.

- 단일 종단 SCSI
- Fast/Wide SCSI
- 파이버 채널

모든 SCSI 디스크가 지원되는 것은 아닙니다. 현재 지원되는 디스크의 목록은 *HP Unix Servers Configuration Guide*를 참조하십시오. 이 설명서는 HP 담당자를 통해 구할 수 있습니다.

주

PCI SCSI 어댑터를 사용하는 시스템이 포함된 클러스터에서는 PCI 및 NIO SCSI 어댑터를 같은 공유 SCSI 버스에 동시에 연결할 수 없습니다.

공유 버스에서 디스크를 사용할 경우에는 외부 공유 **Fast/Wide SCSI** 버스에 인라인 단자를 장착해야 합니다. 자세한 내용은 “문제 해결” 장을 참조하십시오.

SCSI 버스 우선 순위를 계획하고 할당할 때는 공유 버스에 있는 각 노드의 컨트롤러에 할당되는 SCSI 주소에 따라 여러 노드가 공유하는 버스를 한 노드에서 제어할 수 있습니다. 모든 인터페이스 카드의 주소를 비롯한 모든 SCSI 주소는 공유 버스의 모든 장치에서 고유해야 합니다. SCSI 버스의 주소 지정 및 우선 순위에 대한 내용은 **주변 장치를 위한 HP-UX 구성** 설명서를 참조하십시오.

데이터 보호

다음 방법 중 하나를 사용하여고가용성 시스템의 데이터를 보호해야 합니다.

- 디스크 미러링
- RAID 수준 및 다중 데이터 경로를 사용하는 디스크 어레이

디스크 미러링

디스크 미러링은 데이터를 보호하는 방법 중 하나입니다. Serviceguard 패키지에서 사용되는 논리 볼륨은 미러링되어야 합니다. Serviceguard는 디스크에 있는 데이터를 보호하지 않습니다. 이 기능은 HP의 MirrorDisk/UX 제품이 있는 LVM 저장 장치와 VERITAS Volume Manager(B9116AA)가 있는 CVM 및 VxVM에 제공됩니다. 소프트웨어 미러링을 사용하여 논리 볼륨을 구성하면 미러링된 각 세트의 볼륨에는 완전히 똑같은 데이터가 포함됩니다. 디스크 중 하나에 장애가 발생하면 저장 장치 관리자가 자동으로 다른 디스크 미러에 액세스하여 데이터의 가용성을 유지합니다. 온라인 백업이나 더 높은 수준의 가용성을 얻기 위해 LVM(또는 VxVM의 추가 plex)에서 3-웨이 미러링을 사용할 수도 있습니다.

파이버 채널이나 SCSI 버스 장애로부터 보호하려면 별도의 버스에서 데이터의 각 사본에 액세스해야 합니다. 즉, 디스크 드라이브에 있는 데이터의 모든 사본이 동일한 버스에 연결되게 하면 안 됩니다.

고가용성을 확보하려면 데이터 디스크와 루트 디스크를 모두 미러링해야 합니다. 데이터 디스크를 미러링하지 않은 상태에서 디스크 장애가 발생하면 디스크가 대체되어 데이터가 다시 로드될 때까지 클러스터의 모든 노드에서 응용 프로그램을 실행할 수 없게 됩니다. 루트 디스크에 장애가 발생하면 데이터가 공유되기 때문에 클러스터의 다른 노드에서 응용 프로그램을 실행할 수 있습니다. 하지만 루트 디스크에 장애가 발생할 때의 시스템 동작은 예측할 수 없으며, 시스템이 계속 실행되고 있을 때 응용 프로그램이 정지되어 장애가 발생한 노드가 중지될 때까지 다른 노드에서 해당 응용 프로그램이 시작되지 못할 수도 있습니다. 루트 디스크를 미러링하면 루트 디스크에 장애가 발생해도 시스템이 계속해서 정상적으로 작동할 수 있으므로 이러한 시스템 중단 시간을 없애는 데 도움이 됩니다.

RAID 수준 및 다중 데이터 경로를 사용하는 디스크 어레이

데이터를 보호하는 또 다른 방법은 RAID 수준 1이나 RAID 수준 5와 같이 데이터 중복을 제공하는 하드웨어 RAID 수준을 사용한 디스크 어레이를 채택하는 것입니다. 어레이는 디스크의 데이터를 중복 구성합니다. 이러한 보호는 각 노드와 어레이 사이에 중복 호스트 버스 인터페이스(SCSI 또는 파이버 채널)를 사용하여 보완되어야 합니다. LVM의 PV 링크 기능이나 VxVM의 DMP(Dynamic Multipathing)를 통해 구성된 중복 인터페이스를 사용하면 입출력 채널의 단순 장애 요인으로부터 보호할 수 있으며, RAID 1 또는 5 구성은 저장 장치 미디어에 중복성을 제공합니다. PV 링크는 LVM에서 대체 링크로도 알려져 있고 VxVM에서는 다중 경로로도 알려져 있습니다.

DMP는 VxVM의 별도 구성 요소로 사용할 수 있습니다. 능동/능동 장치의 DMP에는 B9116AA가 필요하지만 능동/수동 장치의 DMP는 기본 제품인 Base-VxVM 없이 사용할 수 있습니다.

Event Monitoring Service를 통한 디스크 모니터링

LVM을 사용하고 있으면 별도의 제품(B5736DA)으로 사용할 수 있는 EMS HA 모니터의 디스크 모니터 기능을 사용하여 장애가 발생한 메커니즘을 감지하도록 디스크 모니터링을 구성할 수 있습니다. 패키지 장애 조치를 트리거하거나 Serviceguard, 다른 응용 프로그램 또는 전자 메일로 디스크 장애 이벤트를 보고하도록 모니터링을 설정할 수 있습니다. 자세한 내용은 <http://docs.hp.com> -> High Availability에 있는 설명서 *Using High Availability Monitors(B5736-90042)*를 참조하십시오.

장애가 발생한 디스크 메커니즘의 교체

미러링으로 데이터는 보호되지만 디스크 장애가 발생한 후에는 장애가 발생한 디스크를 교체해야 합니다. 일반 디스크의 경우 클러스터를 중지한 다음 장치를 교체합니다. 특수한 HA 디스크 엔클로저를 사용하는 디스크 어레이의 경우에는 클러스터가 작동 중이고 응용 프로그램이 온라인인 상태에서 디스크를 교체할 수 있습니다. 교체 방법에 대해서는 “클러스터 문제 해결” 장의 “디스크 교체” 절을 참조하십시오.

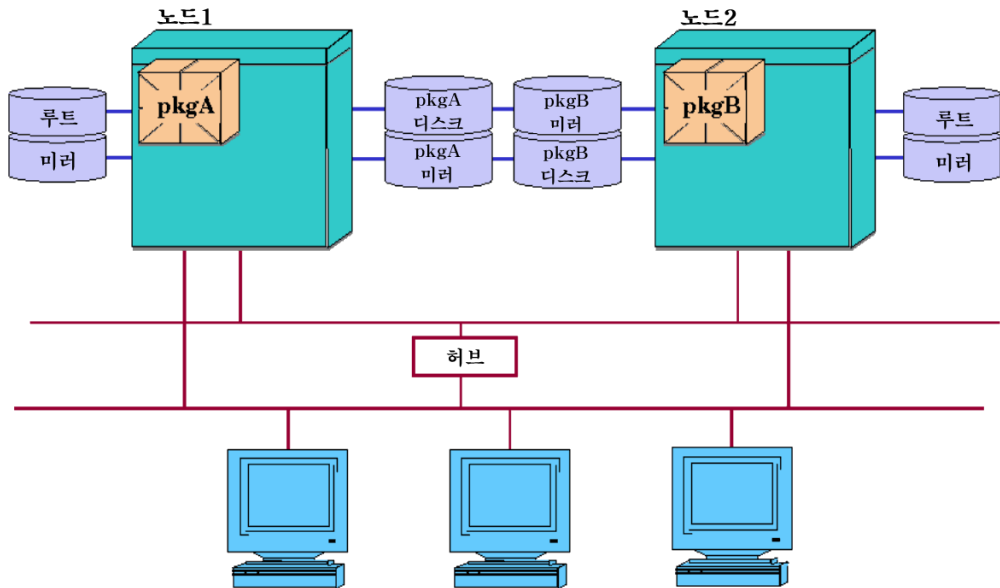
장애가 발생한 입출력 카드의 교체

일부 시스템 구성에서는 시스템이 온라인 상태일 때 장애가 발생한 디스크 입출력 카드를 교체할 수 있습니다. 교체 방법에 대해서는 “클러스터 문제 해결” 장의 “I/O 카드 교체” 절을 참조하십시오.

SCSI 디스크 구성 예제

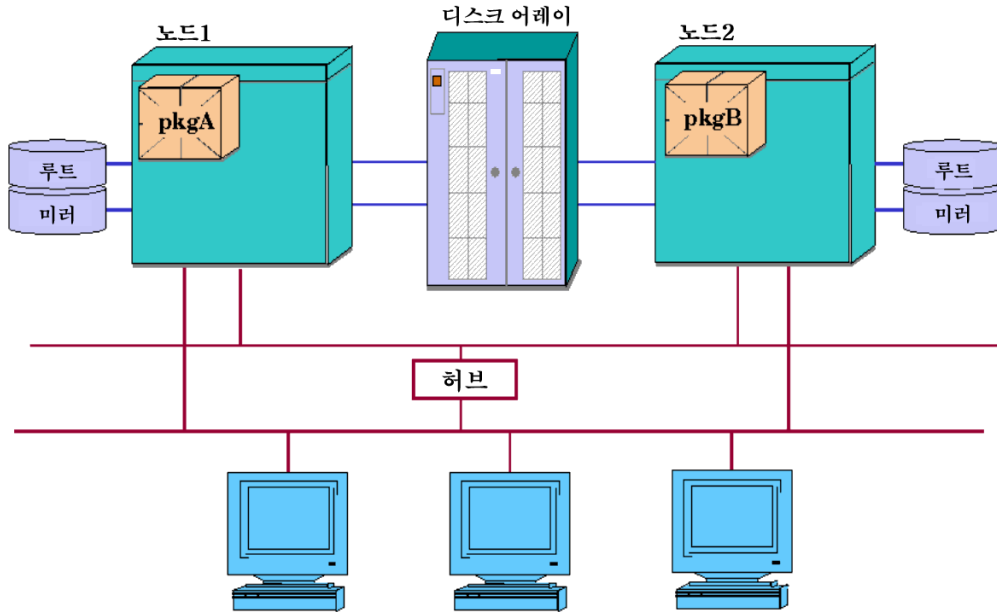
그림 2-5에서는 두 개의 노드로 구성된 클러스터를 보여 줍니다. 각 노드에는 미러링된 루트 디스크 하나와 기본 노드로서 패키지 하나가 포함되어 있습니다. 다른 노드에서 패키지를 받을 수 있도록 각 노드에는 리소스가 할당되어 있습니다. 각 패키지에는 디스크 볼륨 그룹이 하나 할당되어 있으며 이 볼륨 그룹에 있는 논리 볼륨들은 미러링되어 있습니다. 패키지 A의 디스크와 패키지 B의 디스크 미러가 한 인터페이스에 있지만 패키지 B의 디스크와 패키지 A의 디스크 미러는 별도의 버스에 있습니다. 이러한 배치로 인해 단순 장애 요인이 제거되며 버스 중 하나에서 장애가 발생할 경우 디스크나 디스크 미러를 사용할 수 있게 됩니다.

그림 2-5 고가용성을 위해 연결된 미러링된 디스크



아래의 그림 2-6에서는 두 입출력 채널에서 각 노드에 하나의 디스크 어레이가 연결되어 있는 유사한 클러스터를 보여 줍니다. 이러한 종류의 구성에서는 LVM의 PV 링크를 사용하거나 VERITAS DMP(Dynamic Multipath) 또는 EMC PowerPath와 같은 다른 다중 경로 소프트웨어를 사용할 수 있습니다.

그림 2-6 고가용성 디스크 어레이를 사용하는 클러스터



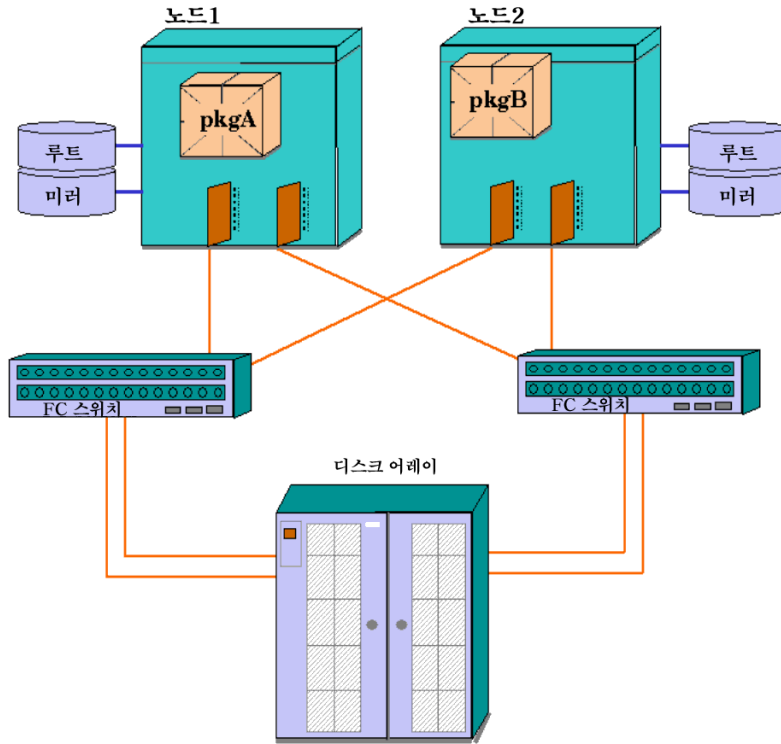
PV 링크를 비롯한 Serviceguard의 논리 볼륨 구성에 대한 자세한 내용은 “HA 클러스터 구성 만들기” 장을 참조하십시오.

파이버 채널 디스크 구성 예제

아래의 그림 2-7에서는 간단한 미러링을 사용하는 루트 디스크를 보여 줍니다. 여기에서는 공유 저장 장치가 디스크 어레이에 연결된 중복 파이버 채널 스위치를 통해 액세스됩니다. 각 노드가 두 개의 스위치 모두에 연결되도록 케이블이 구성되어 있으며, 두 개의 스위치는 중복 링크를 통해 디스크 어레이에 연결되어 있습니다.

그림 2-7

파이버 채널 스위치로 연결된 디스크 어레이를 사용하는 클러스터



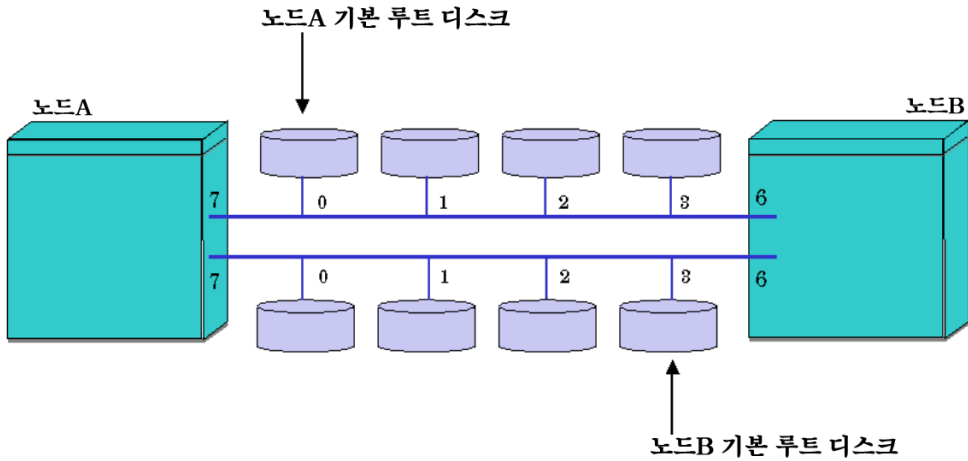
이러한 종류의 구성에서는 LVM의 PV 링크를 사용하거나 VERITAS DMP(Dynamic Multipath) 또는 EMC PowerPath와 같은 다른 다중 경로 소프트웨어도 사용할 수 있습니다.

공유 SCSI 버스에서 루트 디스크 제한

IODC 펌웨어는 같은 SCSI 버스에서 동시에 두 개 이상의 노드 부팅을 지원하지 않습니다. 따라서 클러스터당 둘 이상의 루트 디스크를 단일 SCSI 버스에 연결하지 않아야 합니다.

예를 들어, 그림 2-8에서는 두 개의 노드가 외부 SCSI 버스를 공유하고 노드 A의 기본 루트 디스크는 해당 버스에 연결되어 있으나 노드 B의 기본 루트 디스크는 다른 버스에 연결되어 있는 구성을 보여 줍니다. 번호 0~3과 6, 7은 다른 버스에 있는 SCSI 주소입니다.

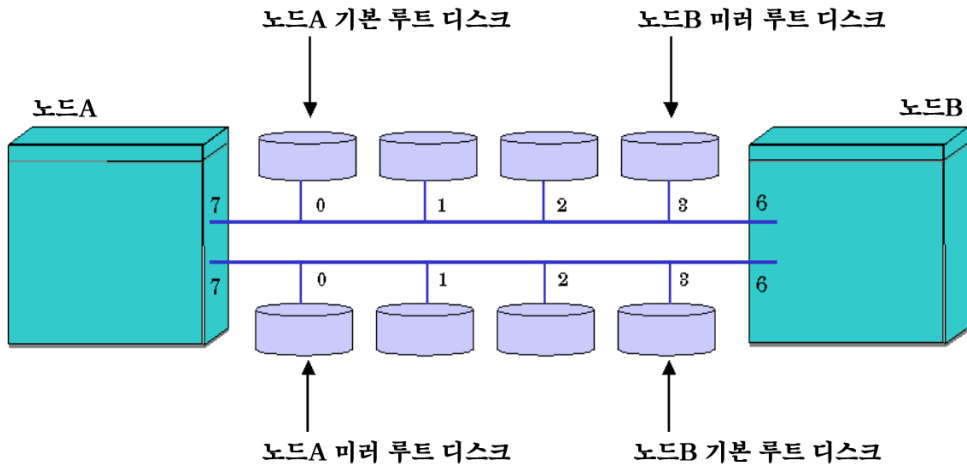
그림 2-8 다른 공유 버스의 루트 디스크



두 노드의 기본 루트 디스크가 같은 버스에 연결되는 구성은 지원되지 않습니다.

노드 A의 기본 루트 디스크가 있는 SCSI 버스에 노드 B의 루트 디스크 미러 사본을 둘 수 있습니다. 이 경우 두 시스템을 동시에 부팅하면 세 가지 장애가 발생하지만 이러한 위험은 허용할 만한 것입니다. 이 시나리오에서는 노드 B가 기본 루트 디스크를 손실하여 재부팅되고, 노드 B가 재부팅될 때 노드 A가 동시에 재부팅되는 경우에만 IODC 펌웨어에 문제가 발생하게 됩니다. 이 구성은 그림 2-9에서 보여 줍니다.

그림 2-9 다른 공유 버스의 기본 디스크 및 미러



디스크 어레이가 공유 버스에 있으면 어레이에 있는 디스크를 루트 디스크로 사용할 수 없습니다.

중복 전원 공급 장치

노드와 디스크에 배터리 백업을 제공하여 하드웨어의 가용성을 확장할 수 있습니다. HP PowerTrust와 같이 HP가 지원하는 UPS(무정전 전원 공급 장치)를 사용하면 일시적인 정전 피해를 입지 않을 수 있습니다.

미러 사본을 여러 전원에 연결하는 방식으로 디스크를 전원 회로에 연결해야 합니다. 부트 디스크는 해당 노드와 같은 회로에서 전원을 공급 받아야 합니다.

특히 클러스터 잠금 디스크(클러스터를 재편성할 때 타이 브레이커로 사용됨)에는 중복 전원 공급 장치가 있어야 합니다. 이 디스크는 클러스터의 노드와 다른 전원 공급 장치에서 전원을 공급 받을 수도 있습니다. HP 담당자는 클러스터의 전원 공급 장치, 디스크 및 LAN 하드웨어의 레이아웃에 대해 자세한 정보를 제공할 수 있습니다.

많은 디스크 어레이와 랙이 장착된 시스템에는 입력 전원이 여러 개 있으며, 장치의 여러 입력 전원은 별도의 전원 회선에 연결되도록 구성되어야 합니다. 일반적으로 입력 전원이 두세 개인 장치는 전원 회선 하나에만 장애가 발생할 경우 정상적으로 계속 작동할 수 있습니다. 따라서 클러스터의 모든 하드웨어에 입력 전원이 두세 개 있는 경우, 클러스터의 전원 회선 설계에서 단순 장애 요인을 없애려면 별도의 전원 회선이 적어도 세 개는 필요합니다.

큰 규모의 클러스터

Serviceguard에서는 최대 16개의 노드로 구성된 클러스터를 만들 수 있습니다. 16개의 노드로 구성된 클러스터는 이더넷을 통해 개별 SPU를 연결하여 만들 수 있으며, 최대 8개의 시스템을 FDDI 네트워킹을 사용하여 Serviceguard 클러스터로 구성할 수 있습니다.

16개의 노드로 구성되는 클러스터를 구성할 수 있다고 해서 모든 종류의 클러스터 구성이 16개의 노드 구성과 동일한 방식으로 이루어지는 것은 아닙니다. 예를 들어, 공유 F/W SCSI 버스의 경우 버스 부하와 케이블 길이 제한으로 인해 동일한 공유 버스에 연결될 수 있는 실질적인 노드 수는 4개로 제한됩니다. 그러나 이 경우에도 16개의 노드를 관리 단위로 설정하고 여러 대용량 저장 장치에 연결된 여러 SCSI 버스에 4개의 노드로 구성되는 하위 그룹을 설정할 수 있습니다.

XP 시리즈나 EMC 디스크 어레이에 대한 비공유 F/W SCSI 연결의 경우에는 4개의 노드 제한이 적용되지 않습니다. 각 노드는 PV 링크를 지원하는 두 개의 F/W SCSI 버스를 통해 XP나 EMC에 직접 연결될 수 있습니다. 패키지는 16개의 노드 모두에서 장애 조치를 수행하도록 구성될 수 있습니다. 이 구성 방식에 대한 자세한 내용은 다음에 나오는 “저장 장치에 지점 간 연결”을 참조하십시오.

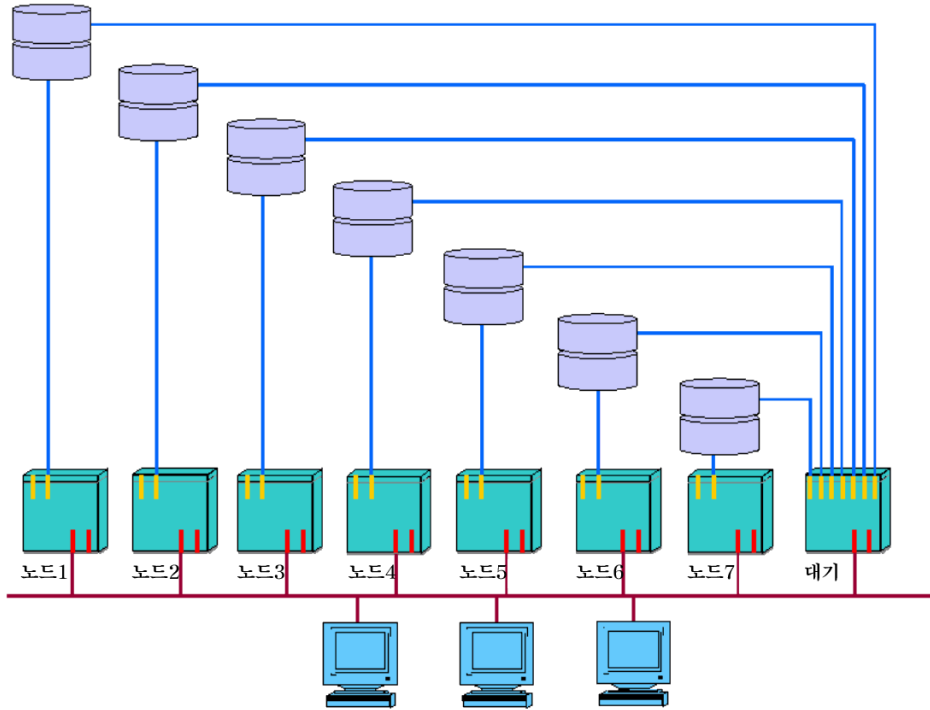
주

큰 규모의 클러스터를 구성할 때는 클러스터 및 패키지 구성 시간뿐만 아니라 cmviewcl 같은 명령의 실행 시간도 길어집니다. 일부 명령 옵션은 이 문제를 해결하는데 유용합니다. 자세한 내용은 cmquerycl에 대한 맨페지를 참조하십시오.

활성/대기 모델

대기 노드가 포함된 클러스터를 만들 수도 있습니다. 예를 들어, 8개의 노드를 사용하는 구성에서 한 노드가 다른 7개의 노드에 대해 대기 노드 역할을 하도록 설정하려면 각 활성 노드에 별도의 연결을 할 수 있는 7개의 공유 버스를 사용하여 백업 노드를 구성하면 됩니다. 이 구성은 그림 2-10에서 보여 줍니다.

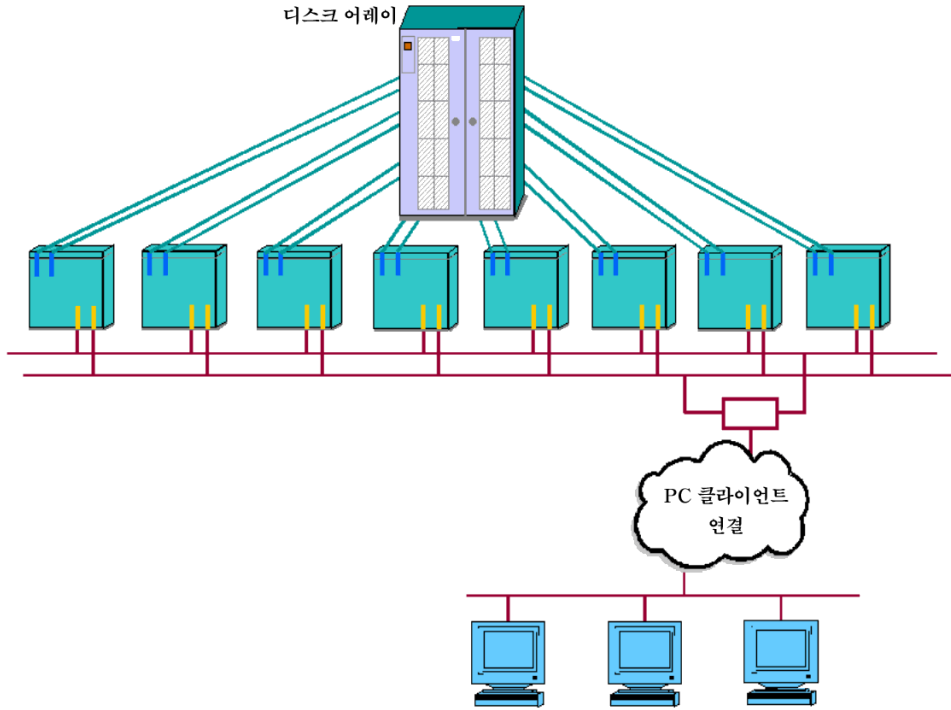
그림 2-10 8개의 노드로 구성된 활성/대기 클러스터



저장 장치에 지점 간 연결

일부 저장 장치에서는 공유 SCSI 버스를 사용하지 않고도 다수의 호스트 노드로 지점 간 연결을 할 수 있습니다. 그림 2-11에서는 파이버 채널로 연결된 8개의 노드로 구성된 클러스터의 예제를 보여 줍니다. 여기에서 클라이언트 연결은 이더넷을 통해 제공됩니다. 노드는 16개의 입출력 포트가 구성된 XP 256 또는 EMC 디스크 어레이에 있는 공유 데이터에 액세스합니다. 각 노드는 PV 링크로 구성된 두 개의 분리된 F/W SCSI 채널을 사용하여 어레이에 연결됩니다. 각 채널은 전용 버스이며 테이지 체인으로 연결되지 않습니다.

그림 2-11 XP 또는 EMC 디스크 어레이를 사용한 8개의 노드로 구성된 클러스터



파이버 채널 스위치 구성은 조정 루프 또는 섬유 로그인 토폴로지를 사용하여 지원됩니다. 지원되는 클러스터 구성에 대한 자세한 내용은 *HP Unix Servers Configuration Guide*를 참조하십시오. 이 설명서는 HP 담당자를 통해 구할 수 있습니다.

3 Serviceguard 소프트웨어 구성 요소 이해

이 장에서는 Serviceguard 소프트웨어 구성 요소의 동작 방식에 대해 간단히 설명합니다. 이 장에 포함된 내용은 다음과 같습니다.

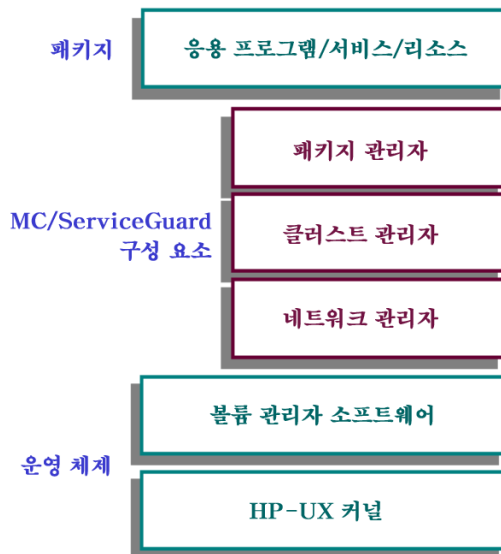
- Serviceguard 아키텍처
- 클러스터 관리자의 작동 방식
- 패키지 관리자의 작동 방식
- 패키지 제어 스크립트의 작동 방식
- 네트워크 관리자의 작동 방식
- 데이터 저장 장치의 볼륨 관리자
- 장애에 대한 대응

Serviceguard 클러스터 설정을 시작할 준비가 되었으면 이 장을 건너뛰고 4장 “HA 클러스터 계획 및 문서화”의 내용을 읽으십시오.

Serviceguard 아키텍처

다음 그림에서는 Serviceguard의 주요 소프트웨어 구성 요소를 보여 줍니다. 이 장에서는 이러한 구성 요소에 대해 자세히 설명합니다.

그림 3-1 Serviceguard 소프트웨어 구성 요소



Serviceguard 데몬

Serviceguard와 관련된 데몬 프로세스로는 다음의 아홉 가지가 있습니다.

- /usr/sbin/cmclconfd — Serviceguard 구성 데몬
- /usr/sbin/cmclld — Serviceguard 클러스터 데몬
- /usr/sbin/cmlogd — Serviceguard Syslog 로그 데몬
- /usr/sbin/cmlvmd — 클러스터 논리 볼륨 관리자 데몬
- /opt/cmom/sbin/cmomd — 클러스터 객체 관리자 데몬
- /usr/sbin/cmsnmpd — 클러스터 SNMP 서브 에이전트(선택적으로 실행)

- /usr/sbin/cmsrvassistd — Serviceguard 서비스 보조 데몬
- /usr/sbin/cmtaped — Serviceguard 공유 테이프 데몬
- /usr/sbin/qs — Serviceguard 퀘럼 서버 데몬

이러한 데몬은 모두 /var/adm/syslog/syslog.log 파일에 로그를 기록합니다.

단, 표준 출력 파일에 로그를 기록하는 퀘럼 서버 데몬과 /var/opt/cmom/cmomd.log 에 로그를 기록하는 /opt/cmom/sbin/cmomd 데몬은 예외입니다. 퀘럼 서버 데몬의 경우에는 출력 경로를 /var/adm/qs/qs.log라는 파일로 다시 지정하는 것이 좋습니다.

구성 데몬: cmclconfd

이 데몬은 Serviceguard 명령이 클러스터 내의 모든 노드로부터 정보를 수집할 때 사용됩니다. 이 데몬은 네트워크 및 볼륨 그룹에 대한 정보와 같은 구성 정보를 수집하며 이진 클러스터 구성 파일을 클러스터의 모든 노드에 배포하기도 합니다. 이 데몬은 inetd(1M)에 의해 시작되며 관련 항목은 /etc/inetd.conf 파일에 있습니다.

클러스터 데몬: cmcld

이 데몬은 Serviceguard 클러스터 내의 다른 노드에 있는 다른 cmcld 데몬에 하트비트 메시지를 보내 클러스터 구성원을 결정하는 데 사용됩니다. 이 데몬은 실시간 우선 순위로 실행되며 메모리에서 잠겨 있습니다. cmcld 데몬은 커널에서 커널 중지 상태를 감지하는 데 사용되는 **안전 타이머**를 설정합니다. cmcld 데몬이 이 타이머를 주기적으로 재설정하지 않으면 해당 커널에서 시스템 **TOC**(제어권 전달)를 발생시킵니다. 이렇게 되면 CPU가 재설정되고 크래시 덤프 파일이 생성됩니다. cmcld가 대다수의 클러스터 구성원과 통신할 수 없는 경우나, cmcld가 예기치 않게 종료 또는 중단되거나 상당 시간 동안 실행 불가능 상태가 되어 커널 타이머를 업데이트할 수 없게 됨으로써 커널이 중지되는 경우에 이 문제가 발생할 수 있습니다. 안전 타이머의 만료로 인해 **TOC**가 발생할 때는 그 전에 /var/adm/syslog/syslog.log와 커널의 메시지 버퍼에 메시지가 기록됩니다.

또한 cmcld 데몬은 시스템에서 네트워크 상태를 감지하고 로컬 LAN 장애 조치를 수행합니다. Serviceguard 패키지의 관리를 맡아 패키지의 실행 위치와 시작 시기를 결정하기도 합니다.

주

Serviceguard의 세 가지 주요 구성 요소인 패키지 관리자, 클러스터 관리자 및 네트워크 관리자는 cmcld 데몬의 일부로 실행됩니다. 이 데몬은 모든 클러스터 노드에서 우선 순위 **20**으로 실행됩니다. 사용자 프로세스의 우선 순위는 **20**보다 낮아야 합니다. 그렇지 않으면 Serviceguard에서 커널 안전 타이머를 업데이트하지 못하게 될 수 있으며 이 경우 해당 노드에서 **TOC**가 발생할 수 있습니다.

Syslog 로그 데몬: cmlogd

cmlogd는 cmc1d에서 syslog에 메시지를 기록할 때 사용됩니다. cmc1d에서 syslog에 기록하는 모든 메시지는 cmlogd를 통해 기록됩니다. 이는 cmc1d에서 직접 syslog에 메시지를 기록할 경우 발생할 수 있는 시간 지연을 방지하기 위한 것입니다.

클러스터 논리 볼륨 관리자 데몬: cmlvmd

이 데몬은 클러스터를 인식하도록 준비된 모든 볼륨 그룹을 관리합니다. 볼륨 그룹이 클러스터를 인식할 수 있으면 클러스터 노드는 단독 모드로 볼륨 그룹을 활성화할 수 있습니다. 따라서 동시에 둘 이상의 노드에서 볼륨 그룹을 쓰기 모드로 활성화할 수는 없습니다.

클러스터 객체 관리자 데몬: cmomd

이 데몬은 클라이언트, 즉 Serviceguard Manager와 같이 클러스터 객체의 상태를 알고 있어야 하는 외부 제품 또는 도구에 클러스터에 대한 정보를 제공합니다. 클라이언트는 객체 관리자에 쿼리를 보내고 객체 관리자로부터 쿼리에 대한 응답을 받습니다. 이 데몬은 객체 관리자의 클라이언트에서만 사용하므로 클러스터 시스템에서 실행되고 있지 않을 수도 있습니다.

cmomd는 클라이언트의 연결을 승인하고 쿼리를 조사합니다. 쿼리는 여러 개의 범주 또는 클래스로 나뉘지고 각 범주 또는 클래스는 다양한 서비스 공급자에 의해 처리됩니다. 공급자는 연결된 모든 노드의 cmc1confd 데몬을 비롯한 다양한 소스에서 데이터를 수집하여 중앙의 수집 위치로 보냅니다. 수집된 데이터는 이 곳에서 해당 클라이언트 쿼리의 요구에 맞게 필터링됩니다. 이 데몬은 inetd(1M)에 의해 시작되며 관련 항목은 /etc/inetd.conf 파일에 있습니다.

클러스터 SNMP 에이전트 데몬: cmsnmpd

이 데몬은 Serviceguard의 그래픽 인터페이스인 Serviceguard Manager에서 클러스터에 대한 정보를 수집할 때 사용됩니다. 이 데몬은 클러스터 MIB을 만듭니다. 클러스터 MIB에 대한 자세한 내용은 URL <http://docs.hp.com/hpux/ha>의 Frequently Asked Questions 부분에서 볼 수 있습니다. 이 데몬은 /etc/rc.config.d/cmsnmpagt 파일을 편집하여 이 하위 에이전트가 자동으로 시작되도록 한 경우에만 실행됩니다. 올바르게 실행되려면 Serviceguard 클러스터가 가동되기 전에 cmsnmpd가 시작되어야 합니다.

서비스 보조 데몬: cmsrvassistd

이 데몬은 클러스터 데몬인 cmcld가 필요로 하는 스크립트나 프로세스를 분기하고 실행합니다. 이 데몬은 다음 두 종류의 분기를 수행합니다.

- 패키지 실행 및 중지 스크립트 실행
- 서비스 시작

서비스의 경우, cmcld는 서비스 프로세스를 모니터링하고, 서비스 재시도 횟수에 따라 `cmsrvassistd`를 통해 서비스를 다시 시작하거나 패키지를 중지시킨 다음 사용 가능한 다른 노드로 옮깁니다.

공유 테이프 데몬: cmtaped

공유 테이프 데몬은 클러스터에 포함된 모든 공유 테이프 장치를 관리합니다. 공유 테이프 장치는 `stapplyconf` 명령을 사용하여 구성할 수 있습니다.

쿼럼 서버 데몬: qs

쿼럼 서버 데몬은 클러스터를 재편성할 때 필요하면 타이 브레이크 서비스를 제공합니다. 쿼럼 서버는 클러스터 외부의 시스템에서 실행되며 **Serviceguard**가 아니라 사용자에 의해 시작됩니다. 대개는 `/etc/inittab`에서 시작되는데 이는 쿼럼 서버에 장애가 발생하거나 쿼럼 서버가 강제 종료될 경우 자동으로 다시 실행된다는 것을 의미합니다.

클러스터의 모든 구성원은 쿼럼 서버와의 연결을 시작하고 이를 유지합니다. 쿼럼 서버가 종료되면 **Serviceguard** 노드는 이를 감지한 다음 연결이 복구될 때까지 주기적으로 쿼럼 서버에 재연결을 시도합니다. 쿼럼 서버가 종료되어 있는 동안 클러스터가 다시 구성되고 타이 브레이크 서비스를 필요로 하는 클러스터에 파티션이 있는 경우에는 재구성된 내용이 적용되지 않습니다.

클러스터 관리자의 작동 방식

클러스터 관리자는 클러스터를 초기화하고, 클러스터의 상태를 모니터링하고, 노드에 장애가 발생할 경우 이를 인식하고, 클러스터에 참가하거나 클러스터에서 나가는 노드가 있을 때 클러스터의 재편성을 조정하는 데 사용됩니다. 클러스터 관리자는 각 노드에서 실행되는 데몬 프로세스로 작동합니다. 클러스터를 시작하고 재편성할 때 하나의 노드가 **클러스터 코디네이터** 역할을 하도록 선택됩니다. 모든 노드가 클러스터 관리 기능을 일부 수행하기는 하지만 노드 간 통신의 중심점이 되는 것은 클러스터 코디네이터입니다.

클러스터 구성

시스템 관리자는 클러스터 구성 매개 변수를 설정하고 초기 클러스터 시작 작업을 수행합니다. 그 다음에 클러스터는 정상 작동 상태에서 수동 간섭 없이 자체적으로 조정합니다. 클러스터 구성 매개 변수에는 클러스터 이름과 노드, 클러스터 하트비트에 대한 네트워킹 매개 변수, 클러스터 잠금 정보, 타이밍 매개 변수 등이 있습니다(“계획” 장 참조). 클러스터 매개 변수는 **Serviceguard Manager**를 사용하거나 **클러스터 ASCII 구성 파일**(5장 참조)을 편집하여 입력할 수 있습니다. 입력된 매개 변수는 클러스터의 모든 노드로 전파되는 이진 구성 파일을 작성하는 데 사용됩니다. 이 이진 클러스터 구성 파일은 클러스터의 모든 노드에서 동일해야 합니다.

하트비트 메시지

클러스터 관리자의 핵심 작업은 클러스터의 노드 간에 **하트비트 메시지**를 주고 받는 것입니다. 클러스터의 각 노드는 해당 노드의 하트비트 장치로 구성된 **RS232** 직렬 회선이나 모니터링되는 **TCP/IP** 네트워크를 통해 클러스터 코디네이터와 하트비트 메시지를 교환합니다. LAN 모니터링에 대한 내용은 뒷부분의 “LAN 인터페이스 모니터링 및 장애 감지” 절에서 자세히 설명합니다.

지정된 시간 안에 다른 모든 클러스터 노드로부터 하트비트 메시지를 받지 못하는 클러스터 노드가 하나라도 있으면 클러스터 재편성이 시작됩니다. 재편성이 끝나 새로운 노드 집합으로 클러스터가 구성되면 이 정보는 **패키지 코디네이터**에 전달됩니다(다음 절 “패키지 관리자의 작동 방식” 참조). 새 클러스터에서 제외된 노드에서 실행되고 있던 패키지는 대체 노드로 전달됩니다. 순간적인 하트비트 손실이 발생하면 클러스터는 전과 동일한 노드로 재편성될 수 있습니다. 이러한 경우 재편성 과정 중에 응용 프로그램의 성능이 약간 저하되어도 패키지는 중지되거나 전환되지 않습니다.

하트비트와 데이터가 같은 LAN 서브넷을 통해 전송되는 경우 데이터 정체로 인해 하트비트 제한 시간이 초과되면 Serviceguard가 하트비트를 잘못 인식하여 불필요한 클러스터 재편성을 시작할 수 있습니다. 이런 상황을 방지하려면 전용 하트비트가 있는 것이 좋으며 하트비트를 데이터 네트워크를 통해 구성하거나 직렬(RS232) 회선을 통해 실행하는 것이 좋습니다. 전용 LAN은 필요하지 않지만 네트워크 분석 결과 클러스터에 하트비트 손실 가능성이 있는 것으로 나타나면 이를 사용할 수도 있습니다.

주

직렬 회선만으로 하트비트를 처리할 수는 없습니다. Serviceguard에서의 직렬 회선 사용에 대한 자세한 내용은 2장의 “직렬(RS232) 하트비트 회선 사용” 절을 참조하십시오.

중요

다중 하트비트는 병렬로 전송됩니다. 일반적으로는 클러스터 노드를 서로 연결하는 모든 서브넷을 하트비트 네트워크로 구성하는 것이 좋습니다. 이렇게 하면 추가 비용 없이 여러 장애에 대한 보호 성능을 향상시킬 수 있습니다. 그러나 VERITAS Cluster Volume Manager(CVM)를 사용하는 경우에는 하트비트 서브넷을 하나만 사용할 수 있습니다. 이 경우 하트비트는 대기 LAN을 사용하여 구성하거나 통합된 포트 그룹으로 구성해야 합니다. 자세한 내용은 뒷부분의 “CVM에서 필요한 단일 하트비트 서브넷”을 참조하십시오.

각 노드는 클러스터 하트비트 간격으로 지정된 속도로 하트비트 메시지를 보냅니다. 클러스터 하트비트 간격은 클러스터를 구성할 때 만드는 클러스터 구성 파일에서 설정합니다. 자세한 내용은 “HA 클러스터 구성 만들기” 장을 참조하십시오.

전체 클러스터의 수동 시작

수동 시작에서는 클러스터 구성에 포함된 모든 노드로 클러스터를 구성합니다. 수동 시작은 일반적으로 클러스터를 처음으로 작동할 때, 클러스터 전반에 걸친 유지 관리 또는 업그레이드 작업이 끝난 다음, 또는 클러스터를 재구성한 다음에 수행됩니다.

시작하기 전에 동일한 내용의 이전 클러스터 구성 파일이 클러스터 내의 모든 노드에 있어야 합니다. 시스템 관리자는 **Serviceguard Manager**에서 클러스터를 시작하거나 한 노드에서 `cmrunc1` 명령을 실행하여 클러스터를 시작합니다. `cmrunc1` 명령은 클러스터가 실행되지 않을 때, 즉 `cmcl1d` 데몬을 실행하는 노드가 없을 때만 사용할 수 있습니다.

클러스터를 시작할 때 클러스터 관리자 소프트웨어는 시작 명령에서 지정된 모든 노드가 클러스터의 유효한 구성원인지, 시작되어 실행 중인지, 클러스터를 구성하려고 하는지, 그리고 서로 통신할 수 있는 여부를 확인합니다. 이러한 조건이 모두 충족되면 클러스터 관리자는 클러스터를 만듭니다.

클러스터 자동 시작

클러스터 자동 시작은 노드가 다시 부팅되어 클러스터에 참여할 때마다 수행됩니다. 즉, 개별 노드가 다시 부팅되었을 때도 수행될 수 있고, 심한 정전으로 인해 모든 **SPU**가 작동이 중지되는 경우와 같이 클러스터의 모든 노드에 장애가 발생했을 때도 수행될 수 있습니다.

클러스터 자동 시작은 `/etc/rc.config.d/cmcluster` 파일에서 `AUTOSTART_CMCLD` 플래그가 **1**로 설정되어 있는 경우에 수행됩니다. 이 매개 변수가 **1**로 설정되어 있을 때 노드가 다시 부팅되면 해당 노드는 기존 클러스터에 다시 참가하거나, 클러스터가 없을 경우 새로운 클러스터를 구성하려고 시도합니다.

동적 클러스터 재편성

동적 재편성이란 실행 중인 클러스터에 참가하거나 클러스터에서 나가는 노드가 있을 때 일어나는 클러스터 구성원의 일시적 변경을 말합니다. 재편성은 구성 파일의 영구적 수정을 의미하는 재구성과는 다릅니다. 클러스터 재편성은 다음과 같은 경우에 발생하며 이외에도 여러 가지 경우가 있습니다.

- 활성화 노드에서 **SPU** 또는 네트워크 장애가 발견된 경우
- 비활성 노드가 클러스터에 참가하려고 하며 클러스터 관리자 데몬이 해당 노드에서 시작된 경우
- 클러스터 구성에서 노드가 추가되거나 삭제된 경우
- 시스템 관리자가 노드를 중지한 경우

- 노드가 패키지 장애로 인해 중지된 경우
- 노드가 서비스 장애로 인해 중지된 경우
- 네트워크 트래픽 과다로 인해 클러스터에서 하트비트 신호를 받을 수 없는 경우
- 하트비트 네트워크의 작동이 중지되었고 하트비트를 전송할 다른 네트워크가 구성되지 않은 경우

재편성을 하면 일반적으로 다른 구성의 클러스터가 생깁니다. 새 클러스터에는 이전 클러스터 구현에서보다 더 적거나 많은 수의 노드가 포함될 수 있습니다.

좌우 분열 현상을 방지하는 클러스터 쿼럼

일반적으로 클러스터 재편성 알고리즘에는 이전에 실행되던 노드 중 과반수(즉, 50% 이상)의 클러스터 쿼럼이 필요합니다. 이전에 실행되던 클러스터의 양 절반(정확히 50%) 모두에 클러스터 재편성이 허용되는 경우에는 동일한 클러스터의 인스턴스가 두 개 실행되는 좌우 분열(split-brain) 상황이 발생합니다. 좌우 분열 상황에서는 같은 응용 프로그램이 서로 다른 클러스터에서 실행되어 동시에 동일한 디스크에 액세스하게 될 수 있습니다. 한 쪽에서는 복구 작업을 시작하고 다른 쪽에서는 디스크 상태를 변경할 수도 있을 것입니다. Serviceguard의 쿼럼 요구 사항은 이러한 좌우 분열 상황을 방지하기 위한 것입니다.

클러스터 잠금

일반적으로는 50%를 넘는 클러스터 쿼럼이 필요하지만 이전에 실행되던 노드 중 나머지 50%가 재편성을 하지 않는 경우에는 정확히 50%의 노드로 새 클러스터를 재편성할 수도 있습니다. 타이 브레이커를 사용하여 동일한 크기의 두 노드 그룹 가운데 하나를 선택한 후 한 그룹에는 클러스터 구성을 허용하고 다른 그룹은 강제로 종료시키면 됩니다. 이 타이 브레이커를 클러스터 잠금이라고 합니다. 클러스터 잠금은 잠금 디스크나 쿼럼 서버를 통해 구현됩니다.

클러스터 잠금은 실행 중인 클러스터에서 장애가 발생했고 Serviceguard가 새 클러스터를 구성하려고 하는데 이전 클러스터가 크기가 같은 두 개의 하위 클러스터로 나뉘어 있는 경우에만 타이 브레이커로 사용됩니다. 이 경우 두 개의 하위 클러스터는 모두 클러스터 잠금을 얻으려고 합니다. 클러스터 잠금을 얻은 하위 클러스터만 새 클러스터를 구성하게 되므로 두 개의 하위 클러스터가 동시에 실행될 수 없습니다. 두 하위 클러스터의 크기가 같지 않은 경우에는 50% 이상의 노드를 포함하는 하위 클러스터가 새 클러스터를 구성하고 클러스터 잠금은 사용되지 않습니다.

두 노드로 구성된 클러스터의 경우에는 클러스터 잠금을 구성해야 합니다. 두 노드 간에 통신이 끊어지면 클러스터 잠금을 얻은 노드가 클러스터를 차지하고 다른 노드는 중지되거나 TOC를 수행하게 됩니다. 클러스터 잠금이 없는 경우 클러스터에 있는 노드 중 하나에서 장애가 발생하면 다른 노드도 중지되므로 클러스터 전체가 중지됩니다. 클러스터 잠금을 얻으려고 하는 중에 클러스터 잠금에 문제가 생겨도 클러스터가 중지됩니다.

잠금 요구 사항

노드가 하나인 클러스터에는 클러스터 잠금이 필요하지 않지만 노드가 두 개인 클러스터에는 클러스터 잠금이 **필요합니다**. 노드가 네 개 이상인 클러스터에서는 클러스터 잠금을 반드시 사용하는 것이 좋습니다. 다섯 개 이상의 노드가 있는 클러스터에서는 클러스터 잠금 디스크를 사용할 수 없지만 퀵림 서버는 사용할 수 있습니다.

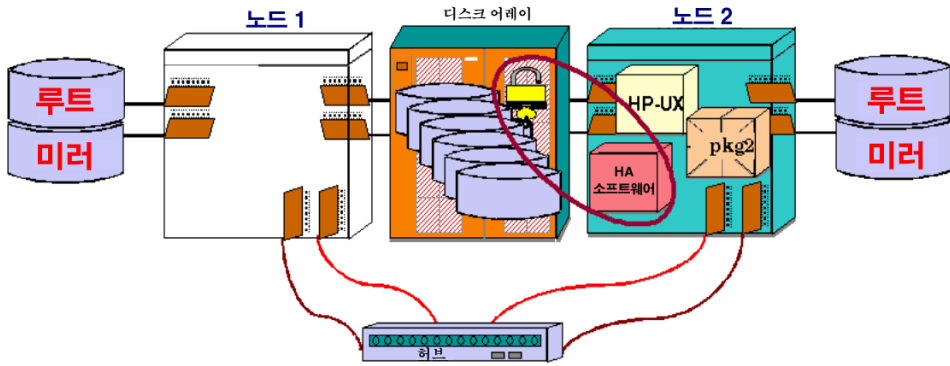
클러스터 잠금으로 LVM 잠금 디스크 사용

잠금 디스크는 최대 네 개의 노드로 구성된 클러스터에 사용할 수 있습니다. 클러스터 잠금 디스크는 클러스터의 모든 노드에서 공유할 수 있는 볼륨 그룹에 있는 LVM의 특수 영역입니다. 한 노드가 클러스터 잠금을 얻으면 이 영역은 다른 노드에서 이 사실을 인식할 수 있도록 표시됩니다.

잠금 디스크는 클러스터 잠금용으로만 사용되는 것이 아니라 사용자 데이터를 저장하는 일반적인 볼륨 그룹의 일부로도 사용될 수 있습니다. 클러스터 잠금 볼륨 그룹과 물리 볼륨 이름은 클러스터 구성 파일에서 식별됩니다.

그림 3-2에서는 잠금 디스크의 작동 방식을 보여 줍니다.

그림 3-2 잠금 디스크 작동



Serviceguard에서는 주기적으로 잠금 디스크의 상태를 확인하고 이상이 발견되면 syslog 파일에 메시지를 기록합니다. 잠금 디스크의 문제를 조기에 발견하려면 이 파일을 모니터링해야 합니다.

만들고 있는 고가용성 구성의 종류에 따라 두 가지 잠금 디스크 옵션인 단일 잠금 디스크와 이중 잠금 디스크 중에서 선택할 수 있습니다. **가능하면 단일 잠금 디스크를 선택하는 것이 좋습니다.** 그러나 잠금 디스크 옵션을 선택할 때는 한 노드의 전원 회로에 장애가 발생하더라도 클러스터 잠금을 계속 사용할 수 있어야 한다는 점을 고려해야 합니다. 따라서 잠금 구성을 선택할 때는 사용 가능한 전원 회로의 수도 고려해야 합니다. 어떤 옵션을 선택하든지 클러스터의 모든 노드가 클러스터 잠금에 액세스하여 고가용성을 유지해야 합니다.

단일 잠금 디스크

단일 잠금 디스크를 사용하는 것이 좋습니다. 단일 잠금 디스크는 클러스터에 있는 모든 노드의 전원 회로와는 다른 회로에서 구성되어야 합니다. 예를 들어, 두 개의 노드로 구성되고 별도로 전원이 공급되는 클러스터 잠금 디스크 하나를 갖춘 클러스터의 경우에는 세 개의 전원 회로를 사용하는 것이 좋습니다. 노드 두 개로 구성된 클러스터의 경우 이 단일 잠금 디스크는 어느 노드와도 전원 회로를 공유할 수 없으므로 외부 디스크여야 합니다. 노드가 3~4개인 클러스터의 경우 이 디스크는 50% 이상의 노드와 전원 회로를 공유할 수 없습니다.

이중 잠금 디스크

클러스터 노드와 동일한 캐비닛에 내부적으로 마운트된 디스크를 사용하는 경우 캐비닛에 잠금 디스크가 있는 노드의 전원이 중단되면 클러스터 잠금도 사용할 수 없게 되므로 이런 유형의 클러스터에서 단일 잠금 디스크는 단순 장애 요인(Single Point of Failure)이 됩니다. 마찬가지로, 두 개의 개별 데이터 센터에서 실행되는 노드가 포함된 캠퍼스 클러스터에서 단일 잠금 디스크는 해당 디스크가 있는 데이터 센터에 심각한 장애가 발생할 경우 단순 장애 요인이 됩니다. **이와 같은 두 경우에만** 별도로 전원이 공급되는 클러스터 디스크로 구성된 이중 클러스터 잠금을 사용하여 잠금 디스크가 단순 장애 요인이 되지 않도록 해야 합니다. 이중 클러스터 잠금의 경우 디스크는 다른 디스크와 전원 회로나 노드 새시를 공유할 수 없습니다. 이 경우 하나의 노드와 디스크에 영향을 주는 전원 장애가 발생해도 다른 노드와 디스크는 계속 사용할 수 있으므로 나머지 노드에서 클러스터 재편성이 수행될 수 있습니다. 캠퍼스 클러스터의 경우에는 각 데이터 센터에 잠금 디스크가 하나씩 있어야 하며 모든 노드가 양쪽 잠금 디스크에 액세스할 수 있어야 합니다. 한 데이터 센터에서 장애가 발생해도 다른 데이터 센터에 있는 노드가 로컬 잠금 디스크를 얻을 수 있으므로 성공적으로 새로운 클러스터를 재편성할 수 있게 됩니다.

주

이중 잠금 디스크는 중복 클러스터 잠금을 제공하지 않습니다. 사실상 이중 잠금은 복합 잠금이므로 클러스터를 구성할 때 두 개의 디스크를 사용할 수 있어야 합니다. 반면 단일 잠금 디스크의 경우에는 디스크가 하나만 필요합니다. 따라서 이중 클러스터 잠금은 정확히 절반의 클러스터 노드에서 장애가 발생할 때 단일 클러스터 잠금을 격리시킬 수 없는 **경우에만 사용하는 것이 좋습니다.**

이중 잠금 디스크 중 하나에서 장애가 발생하면 Serviceguard는 정기 검사를 수행할 때 이를 감지하고 syslog 파일에 메시지를 기록합니다. 잠금 디스크 중 하나가 손실된 후 클러스터 노드에 장애가 발생하면 클러스터 전체의 작동이 중지될 수 있습니다.

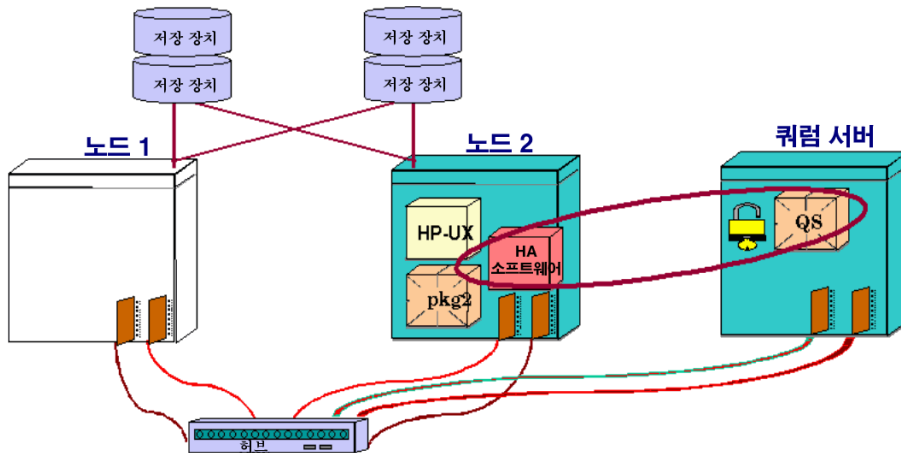
클러스터 잠금으로 퀴럼 서버 사용

퀴럼 서버는 클러스터 크기에 상관없이 사용할 수 있습니다. 퀴럼 서버 프로세스는 해당 프로세스에서 Quorum 서비스를 제공하고 있는 대상 클러스터의 외부 컴퓨터에서 실행됩니다. 퀴럼 서버는 알려진 포트에서 Serviceguard 노드의 연결 요청을 수신합니다. 퀴럼 서버는 메모리에 각 클러스터를 위한 특별 영역을 유지 관리하면서 한 노드가 클러스터 잠금을 얻으면 이 영역에 표시하여 다른 노드에 잠금이 “설정”되었음을 알립니다.

니다. 크기가 같은 두 노드 그룹 사이에서 통신이 끊어지면 퀴럼 서버로부터 잠금을 얻은 그룹이 클러스터를 차지하고 다른 노드에서는 TOC를 수행합니다. 클러스터 잠금이 없는 경우 노드 그룹 중 어느 한 쪽에서 장애가 발생하면 다른 그룹도 중지되므로 클러스터 전체가 중지됩니다. 퀴럼 서버에 액세스하려고 할 때 퀴럼 서버를 사용할 수 없으면 클러스터는 중지됩니다.

그림 3-3에서는 퀴럼 서버의 작동 방식을 보여 줍니다. 노드 1과 노드 2 사이의 통신이 끊어지면 퀴럼 서버는 하나의 노드(이 예제에서는 노드 2)를 선택하여 클러스터를 계속 실행합니다. 다른 노드는 중지됩니다.

그림 3-3 퀴럼 서버의 작동 방식



퀴럼 서버는 별도의 시스템에서 실행되며 여러 클러스터에 Quorum 서비스를 제공할 수 있습니다.

클러스터 잠금 없음

일반적으로 클러스터 잠금 없이 세 개 이하의 노드가 포함된 클러스터를 구성하면 안 됩니다. 두 개의 노드로 구성된 클러스터에서는 클러스터 잠금이 필요합니다. 노드가 세 개 이상인 구성에서는 클러스터 잠금을 사용하지 않을 수도 있습니다. 단, 클러스터에 타이 브레이크 서비스가 필요할 수 있다는 것을 고려하여 결정해야 합니다. 예를 들어, 세 개의 노드로 구성된 클러스터에서 노드 하나가 유지 관리 목적으로 제거되면 클러스터는 두 개의 노드를 포함하는 클러스터로 재편성됩니다. 이 경우 나중에 노드 또는 통신 장애로 인해 타이 브레이크 상황이 발생하면 전체 클러스터를 사용할 수 없게 됩니다.

노드가 네 개 이상인 클러스터에서는 클러스터가 동일한 크기로 둘로 나뉠 가능성이 매우 적으므로 클러스터 잠금을 사용하지 않아도 됩니다. 그러나 한 번에 정확히 절반의 노드에서 장애가 발생하지 않도록 클러스터를 구성해야 합니다. 예를 들어, 동일한 수의 노드 사이를 연결하는 단일 LAN과 같이 단순 장애 요인이 될 수 있는 요인이 없어야 하며 정확히 절반의 노드가 단일 전원 회로를 사용하지 않도록 해야 합니다.

패키지 관리자의 작동 방식

클러스터의 각 노드는 패키지 관리자의 인스턴스를 실행합니다. 클러스터 코디네이터에 있는 패키지 관리자를 **패키지 코디네이터**라고 합니다.

패키지 코디네이터는 다음 작업을 수행합니다.

- 패키지를 실행하거나 중지하거나 이동할 시기와 위치를 결정합니다.

모든 노드에서 패키지 관리자는 다음 작업을 수행합니다.

- 사용자 정의 제어 스크립트를 실행하여 패키지 및 패키지 서비스를 실행하거나 중지합니다.
- 모니터링되는 리소스의 상태 변화에 대응합니다.

패키지 종류

클러스터에서 실행할 수 있는 패키지 종류로는 **장애 조치 패키지**와 **시스템 다중 노드 패키지**의 두가지가 있습니다. 시스템 다중 노드 패키지는 **VERITAS Cluster Volume Manager(CVM)**를 저장 장치 관리자로 사용하는 시스템에서만 사용됩니다.

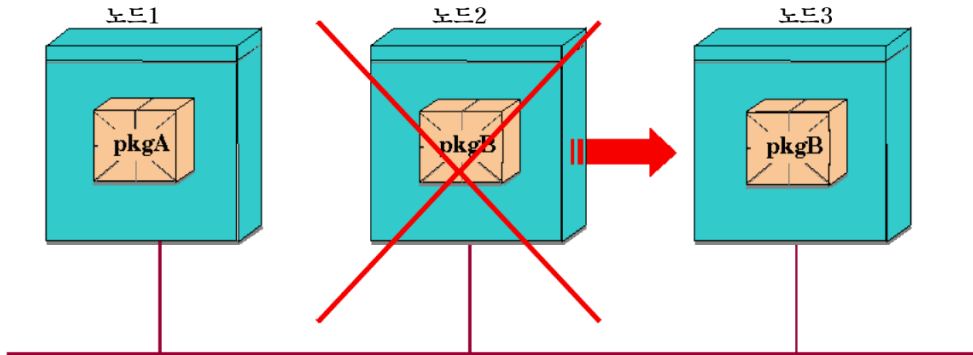
`VxVM-CVM-pkg`라고 하는 이 패키지는 클러스터 내의 모든 활성 노드에서 실행되며 볼륨 관리자 소프트웨어에 클러스터 구성원 정보를 제공합니다. 이 패키지 종류는 저장 장치 관리에 **CVM**을 사용하는 경우에만 구성되고 사용됩니다. **CVM**용 시스템 다중 노드 패키지를 만드는 방법에 대해서는 **5장**을 참조하십시오. 이 절의 나머지 부분에서는 표준 장애 조치 패키지에 대해 설명합니다.

장애 조치 패키지

장애 조치 패키지는 클러스터가 시작되면 해당 노드에서 시작됩니다. 패키지 **장애 조치**는 패키지 코디네이터가 새 노드에서 패키지를 시작할 때 수행됩니다. 패키지 장애 조치는 서비스, 네트워크 또는 리소스 장애가 발생할 경우 기존의 패키지를 중지시키고 패키지의 새 인스턴스를 시작하는 과정입니다.

다음 그림에서는 장애 조치 방식을 보여 줍니다.

그림 3-4 장애 조치 중 패키지 이동



패키지 구성

각 패키지는 개별적으로 구성됩니다. **Serviceguard Manager**를 사용하거나 **패키지 ASCII 구성 파일**(6장 참조)을 편집하여 패키지를 만듭니다. 그런 다음 `cmapplyconf` 명령을 사용하여 패키지를 확인하고 클러스터 구성 데이터베이스에 적용합니다. 또한 패키지 서비스의 실행을 관리하는 **패키지 제어 스크립트**를 만듭니다. 그러면 패키지의 실행 준비가 끝납니다.

패키지 실행/중지 시기 및 위치 결정

패키지 구성 파일은 패키지에 이름을 지정하고 패키지가 실행할 수 있는 노드를 우선 순위별로 나열(즉, 우선 순위가 가장 높은 노드를 먼저 표시)한 목록을 포함합니다. 또한 이 파일에는 장애 조치 동작을 결정하는 세 개의 매개 변수가 포함됩니다. 이 매개 변수는 `AUTO_RUN`, `FAILOVER_POLICY` 및 `FAILBACK_POLICY`입니다.

패키지 전환

`AUTO_RUN` 매개 변수(이전 버전의 **Serviceguard**에서는 `PKG_SWITCHING_ENABLED` 매개 변수)는 클러스터가 시작될 때 패키지에 적용되는 기본 글로벌 전환 속성을 정의합니다. 즉, 장애 시 해당 패키지를 새 노드에서 자동으로 다시 시작할지 여부와 클러스터가 시작될 때 해당 패키지를 자동으로 시작할지 여부를 정의합니다. 클러스터가 실행된 후에는 `cmmodpkg` 명령을 사용하여 각 패키지의 패키지 전환 속성을 설정할 수 있습니다.

이 매개 변수는 패키지 ASCII 구성 파일에 다음 코드로 기록됩니다.

```
# The default for AUTO_RUN is YES. In the event of a  
# failure, this permits the cluster software to transfer the  
package  
# to an adoptive node. Adjust as necessary.  
  
AUTO_RUN    YES
```

패키지 전환이란 패키지와 패키지에 연관된 IP 주소를 새 시스템으로 이동하는 과정입니다. 새 시스템에는 이미 동일한 서브넷이 구성되어 제대로 작동하고 있어야 합니다. 그렇지 않으면 패키지가 시작되지 않습니다. 패키지 장애 조치 시 TCP 연결은 끊어집니다. TCP 응용 프로그램은 자동으로 다시 연결되지 않으므로 수동으로 다시 연결해야 합니다. 패키지가 여러 서브넷에 종속되어 있으면 패키지가 시작되기 전에 대상 노드에서 모든 서브넷을 사용할 수 있어야 합니다.

그림 3-5와 그림 3-6에서는 변동 가능한 IP 주소의 전환 과정을 보여 줍니다. 그림 3-5에서는 두 개의 노드로 구성된 클러스터의 전환 전 상태를 보여 줍니다. 여기서 패키지 1은 노드 1에서 실행되고 패키지 2는 노드 2에서 실행됩니다. 사용자는 사용하려는 패키지의 IP 주소를 사용하여 노드에 연결합니다. 각 노드에는 고정 ID 주소가 연관되어 있고 각 패키지에는 IP 주소가 연관되어 있습니다.

그림 3-5 패키지 전환 전

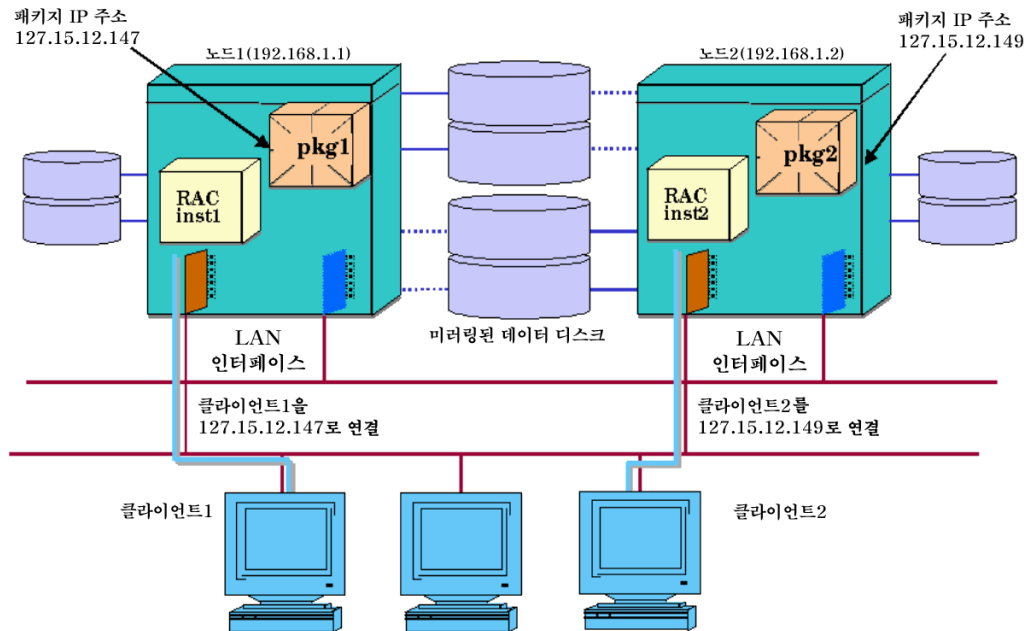
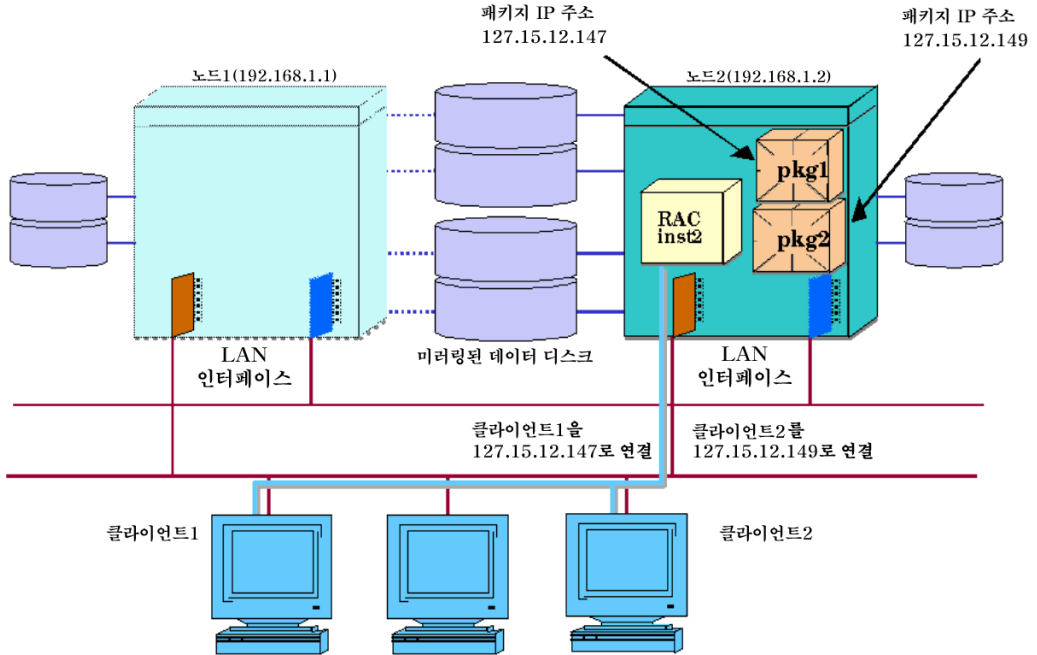


그림 3-6에서는 노드 1에 장애가 발생하여 패키지 1이 노드 2로 이동한 상태를 보여 줍니다. 패키지 1의 IP 주소는 해당 패키지와 함께 노드 2로 이동되었습니다. 이제 패키지 1은 노드 2에서 실행되며 계속 사용할 수 있습니다. 또한 노드 2에서는 패키지 1의 디스크와 패키지 2의 디스크에 모두 액세스할 수 있습니다.

그림 3-6 패키지 전환 후



장애 조치 정책

패키지 관리자는 패키지 구성 파일에 포함된 우선 순위 목록과 이 파일에서 코딩되거나 **Serviceguard Manager**를 사용하여 설정된 `FAILOVER_POLICY` 매개 변수를 기준으로 패키지를 실행할 노드를 선택합니다. 특정 노드를 지정하지 않은 상태에서 패키지를 시작해야 할 때 패키지 관리자는 이 장애 조치 정책에 따라 패키지를 실행할 노드를 선택합니다. 이 정책은 장애 조치 과정에만 적용되는 것이 아니라 초기 시작을 포함한 패키지 시작 과정에도 적용됩니다. 두 가지 장애 조치 정책은 `CONFIGURED_NODE`(기본값)와 `MIN_PACKAGE_NODE`입니다. 이 매개 변수는 패키지 ASCII 구성 파일에 다음 코드로 기록됩니다.

```
# Enter the failover policy for this package. This policy will
be used
# to select an adoptive node whenever the package needs to be
started.
# The default policy unless otherwise specified is
```

패키지 관리자의 작동 방식

```
CONFIGURED_NODE.
# This policy will select nodes in priority order from the list
of
# NODE_NAME entries specified below.

# The alternative policy is MIN_PACKAGE_NODE. This policy will
select
# the node, from the list of NODE_NAME entries below, which is
# running the least number of packages at the time of failover.

#FAILOVER_POLICY          CONFIGURED_NODE
```

장애 조치 정책 값으로 CONFIGURED_NODE를 사용하면 노드 목록의 사용 가능한 노드 중 우선 순위가 가장 높은 노드에서 패키지가 시작됩니다. 장애 조치가 발생하면 패키지는 목록의 사용 가능한 노드 중 다음으로 우선 순위가 높은 노드로 이동합니다.

장애 조치 정책 값으로 MIN_PACKAGE_NODE를 사용하는 경우에는 현재 실행하고 있는 패키지의 수가 가장 적은 노드에서 패키지가 시작됩니다. 실행 중인 패키지 수가 가장 적다는 것이 로드가 가장 작다는 것을 의미하지는 않습니다. 이 과정에서 확인하는 것은 단지 노드에서 현재 실행되고 있는 패키지의 수일 뿐입니다.

자동 순환 대기

MIN_PACKAGE_NODE 장애 조치 정책을 사용하면 노드 하나를 클러스터의 **자동 순환 대기** 노드로 사용하는 것이 가능한 클러스터를 구성할 수 있습니다. 예를 들어, 네 개의 노드가 포함된 클러스터의 경우 다음 패키지 구성을 사용할 수 있습니다. 모든 패키지는 모든 노드에서 실행될 수 있으며 동일한 NODE_NAME 목록을 갖습니다. 이 예제에서는 노드 이름이 패키지마다 다른 순서로 나타나지만 꼭 그래야 하는 것은 아닙니다.

표 3-1

패키지 구성 데이터

| 패키지 이름 | NODE_NAME 목록 | FAILOVER_POLICY |
|--------|---------------------------|------------------|
| pkgA | 노드 1, 노드 2, 노드 3, 노드 4 | MIN_PACKAGE_NODE |
| pkgB | 노드 2, 노드 3, 노드 4, 노드 1 | MIN_PACKAGE_NODE |
| pkgC | 노드 3, 노드 4, 노드 1, 노드 2 | MIN_PACKAGE_NODE |

클러스터가 시작되면 각 패키지는 그림 3-7에서와 같이 시작됩니다.

그림 3-7 장애 조치 이전의 순환 대기 구성

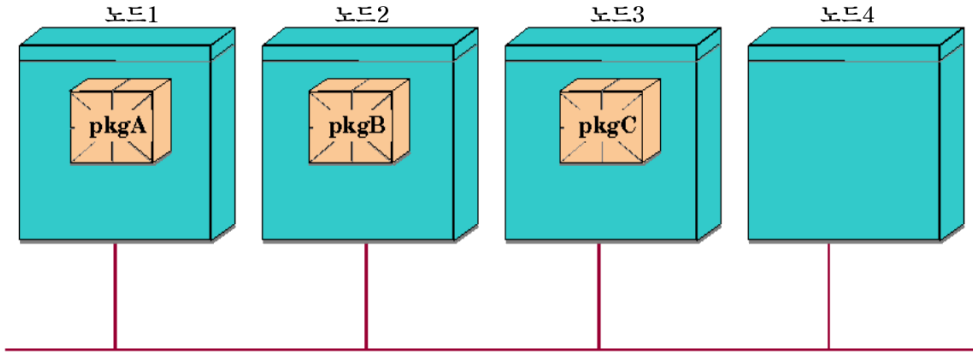
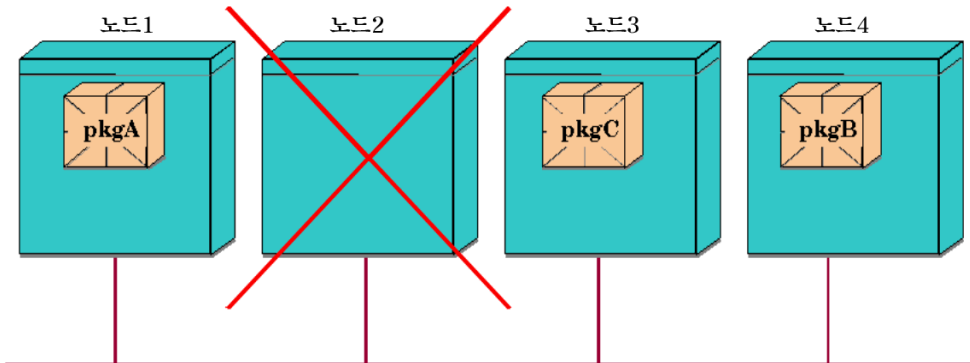


그림 3-8에서처럼 노드 2에서 장애가 발생하면 장애 조치를 위해 해당 노드의 패키지는 실행 중인 패키지 수가 가장 적은 노드로 이동됩니다.

그림 3-8 장애 조치 이후의 순환 대기 구성

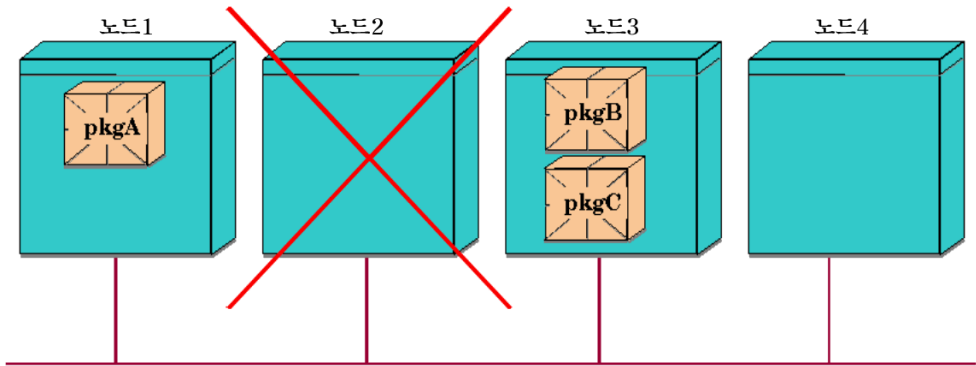


주

MIN_PACKAGE_NODE 정책을 사용할 경우 노드 2가 복구되어 다시 클러스터에 참가하면 이 노드가 가장 적은 수의 패키지를 실행하게 되므로 새로운 대기 노드가 됩니다.

CONFIGURED_NODE 장애 조치 정책을 사용하여 이러한 패키지를 설정한 경우 처음에는 패키지가 그림 3-7과 같이 시작되지만 노드 2에서 장애가 발생하면 해당 패키지는 그림 3-9와 같이 노드 3에서 시작됩니다.

그림 3-9 장애 조치 후의 CONFIGURED_NODE 정책 패키지



장애 조치 정책 값으로 CONFIGURED_NODE를 사용하면 해당 클러스터의 구성원으로 실행 중인 노드 중에서 우선 순위가 가장 높은 노드에서 패키지가 시작됩니다. 장애 조치가 발생하면 패키지는 목록의 사용 가능한 노드 중 다음으로 우선 순위가 높은 노드로 이동합니다.

장애 복구 정책

FAILBACK_POLICY 매개 변수를 사용하면 패키지가 기본 노드에서 실행되고 있지 않을 때 기본 노드를 사용할 수 있게 되면 패키지를 해당 기본 노드로 다시 이동할지 여부를 지정할 수 있습니다. 구성된 기본 노드는 패키지의 노드 목록에 첫 번째로 나열되는 노드입니다.

이 정책에 사용할 수 있는 두 개의 값은 AUTOMATIC과 MANUAL입니다. 이 매개 변수는 패키지 ASCII 구성 파일에 다음 코드로 기록됩니다.

```
# Enter the failback policy for this package. This policy will
be used
# to determine what action to take during failover when a
package
# is not running on its primary node and its primary node is
capable
# of running the package. Default is MANUAL which means no
attempt
# will be made to move the package back to it primary node when
it is
# running on an alternate node. The alternate policy is
AUTOMATIC which
```

```
# means the package will be moved back to its primary node
whenever the
# primary node is capable of running the package.
```

```
#FAILBACK_POLICY          MANUAL
```

예를 들어, 네 개의 노드가 포함된 다음 구성에서는 FAILOVER_POLICY가 CONFIGURED_NODE로 설정되어 있고 FAILBACK_POLICY가 AUTOMATIC으로 설정되어 있습니다.

그림 3-10 장애 조치 이전의 자동 장애 복구 구성

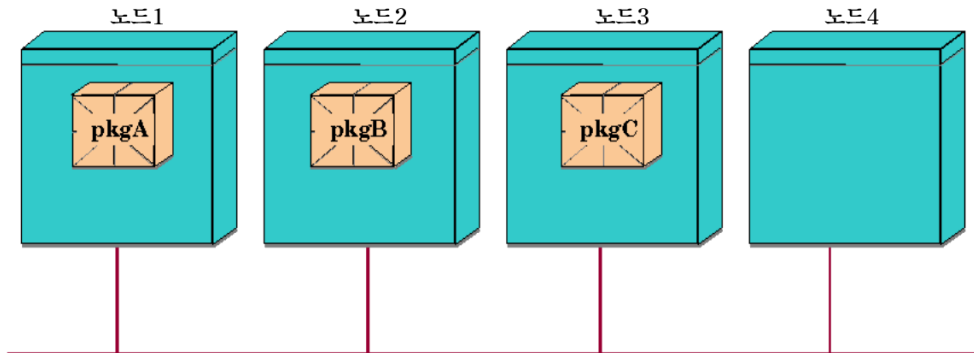
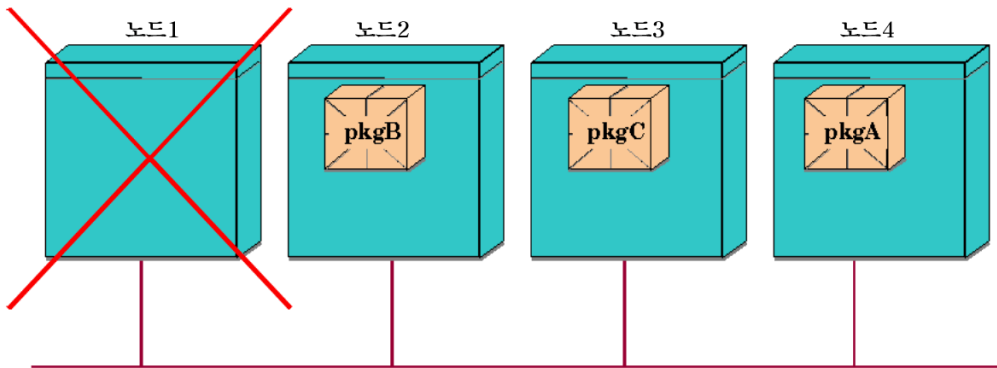


표 3-2 샘플 클러스터의 노드 목록

| 패키지 이름 | NODE_NAME 목록 | 장애 조치 정책 | 장애 복구 정책 |
|--------|--------------|-----------------|-----------|
| pkgA | 노드 1, 노드 4 | CONFIGURED_NODE | AUTOMATIC |
| pkgB | 노드 2, 노드 4 | CONFIGURED_NODE | AUTOMATIC |
| pkgC | 노드 3, 노드 4 | CONFIGURED_NODE | AUTOMATIC |

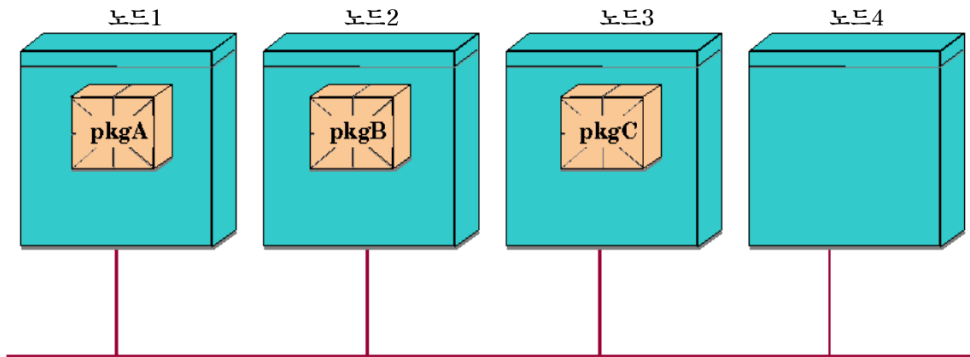
다음과 같이 노드 1에서 장애가 발생하면 클러스터 재편성 후 pkgA는 노드 4에서 실행됩니다.

그림 3-11 장애 조치 이후의 자동 장애 복구 구성



노드 1은 재부팅된 후에 클러스터에 다시 참가합니다. 이때 pkgA는 자동으로 노드 4에서 중지되고 노드 1에서 다시 시작됩니다.

그림 3-12 노드 1이 다시 시작된 후의 자동 장애 복구 구성



주

FAILBACK_POLICY를 AUTOMATIC으로 설정하면 중요한 업무 시간에 패키지 장애 복구가 수행되어 응용 프로그램이 중단될 수 있습니다. 자동 장애 복구를 사용하는 경우에는 패키지를 기본 노드로 다시 전환하기 위해 패키지 서비스를 일시적으로 중단해도 문제되지 않을 때까지 패키지의 기본 노드를 클러스터에 다시 추가하지 않을 수 있습니다.

장애 조치 정책과 장애 복구 정책의 결합

FAILOVER_POLICY를 MIN_PACKAGE_NODE로 설정하고 FAILBACK_POLICY를 AUTOMATIC으로 설정하여 함께 사용할 경우, 가장 적은 수의 패키지를 실행하는 노드는 장애 조치가 발생할 때마다 다를 수 있으므로 예상하지 못한 노드에서 패키지가 실행될 수 있습니다.

이전 버전의 패키지 구성 파일 사용

이전 버전의 Serviceguard를 사용하여 만든 패키지 구성 파일을 사용할 경우, FAILOVER_POLICY는 기본 패키지 동작인 CONFIGURED_NODE가 되고 FAILBACK_POLICY는 기본 패키지 동작인 MANUAL이 됩니다. 이러한 정책을 변경하려면 패키지 구성 파일을 편집하여 매개 변수를 추가하거나 `cmmakepkg`를 사용하여 새 패키지 구성 파일을 만듭니다.

Serviceguard 버전 A.11.12부터 `PKG_SWITCHING_ENABLED` 매개 변수는 `AUTO_RUN`으로 이름이 변경되고 `NET_SWITCHING_ENABLED` 매개 변수는 `LOCAL_LAN_FAILOVER_ALLOWED`로 이름이 변경되었습니다. 이전 이름을 사용해도 구성 파일에서 여전히 동일한 효과를 나타내지만 가능하면 키워드를 변경하는 것이 좋습니다.

Event Monitoring Service 사용

기본 패키지 리소스에는 클러스터 노드, LAN 인터페이스, 그리고 한 응용 프로그램에 속한 개별 프로세스들인 서비스가 포함됩니다. 이러한 리소스는 모두 Serviceguard에서 직접 모니터링합니다. 이 외에 추가 모니터를 구성하는 데 사용할 수 있는 **Event Monitoring Service** 레지스트리도 사용할 수 있습니다. 이 레지스트리를 사용하면 다른 소프트웨어 구성 요소에서 해당 구성 요소의 리소스에 대한 모니터링 기능을 Serviceguard에 제공할 수 있게 됩니다. 현재 다른 소프트웨어 제품과 함께 제공되는 모니터로는 **EMS** 고가용성 모니터와 **ATM** 모니터가 있습니다.

등록된 리소스가 패키지에 구성되면 패키지 관리자는 리소스 등록자를 호출하여 리소스에 대한 외부 모니터를 시작합니다. 노드가 클러스터에 참가할 때나 패키지가 시작된 후 즉시 리소스가 시작되도록 구성할 수 있습니다. 그러면 모니터는 패키지를 시작하기 전에 리소스를 사용할 수 있는지 확인하는 메시지를 Serviceguard로 보냅니다. 또한 패키지 관리자는 패키지가 시작된 후 리소스를 사용할 수 없게 되면 다른 노드로 패키지를 이동하거나 다른 조치를 취할 수 있습니다.

Serviceguard Manager 패키지 구성 영역에 표시된 사용 가능한 리소스 목록에서 리소스를 선택하여 패키지의 등록된 리소스를 지정할 수 있습니다. Serviceguard Manager에서 표시하는 목록의 크기는 시스템에 등록된 리소스 모니터에 따라 달라집니다. 또는 `/opt/resmon/bin/resls` 명령을 사용하여 시스템에 등록된 리소스에 대한 정보를 얻을 수 있습니다. 자세한 내용은 `resls(1m)`에 대한 맨페이지를 참조하십시오.

EMS HA 모니터 사용

별도의 제품으로 사용할 수 있는 EMS HA 모니터(B5736DA)는 디스크 및 기타 리소스 모니터링을 패키지 종속 항목으로 설정하는 데 사용할 수 있습니다. 다음은 EMS를 사용하여 모니터링할 수 있는 리소스 속성의 예입니다.

- 논리 볼륨 상태
- 물리 볼륨 상태
- 시스템 로드
- 사용자 수
- 파일 시스템 활용도
- LAN 상태

모니터를 패키지 종속 항목으로 구성하면 해당 모니터는 리소스가 중지되었음을 나타내는 이벤트가 발생할 때 이를 패키지 관리자에게 알립니다. 그러면 패키지는 장애 조치를 위해 대체 노드로 이동될 수 있습니다.

EMS HA 모니터는 모니터링된 이벤트를 그래픽 표시 또는 운영자 알림을 위한 OpenView IT/Operations와 같은 대상 응용 프로그램에 보고할 때도 사용할 수 있습니다. 자세한 내용은 *Using High Availability Monitors (B5736-90022)*를 참조하십시오.

패키지 장애 조치 동작 선택

패키지 장애 조치 동작을 결정하기 위해 실행되고 있지 않은 패키지를 자동으로 실행할 노드를 결정하는 패키지 장애 조치 정책을 정의할 수 있습니다. 또한 기본 노드를 사용할 수 있게 되었을 때 패키지를 기본 노드로 자동 전환할지 여부를 결정하는 장애 복구 정책도 정의할 수 있습니다.

다음 표에서는 여러 가지 장애 조치 동작과 Serviceguard Manager 또는 ASCII 패키지 구성 파일에서 각 동작을 결정하는 설정에 대해 설명합니다.

표 3-3 **패키지 장애 조치 동작**

| 전환 동작 | Serviceguard Manager의 옵션 | ASCII 파일의 매개 변수 |
|--|---|---|
| <p>일반적으로 서비스, 네트워크 또는 EMS 장애가 감지되면 패키지가 다른 노드로 전환됩니다. 전환이 수행되기 전에 중지 스크립트가 실행됩니다(기본값).</p> | <ul style="list-style-type: none"> • 패키지 빠른 장애 조치를 사용 불가능으로 설정합니다(기본값). • 모든 서비스에 대해 서비스 빠른 장애 조치를 사용 불가능으로 설정합니다(기본값). • 패키지에 대해 자동 전환을 사용 가능으로 설정합니다(기본값). | <ul style="list-style-type: none"> • NODE_FAIL_FAST_ENABLED를 NO로 설정합니다(기본값). • SERVICE_FAIL_FAST_ENABLED를 NO로 설정합니다(기본값). • 패키지에 대해 AUTO_RUN을 YES로 설정합니다(기본값). |
| <p>패키지가 장애 조치를 위해 활성 패키지 수가 가장 적은 노드로 이동됩니다.</p> | <ul style="list-style-type: none"> • 장애 조치 정책을 최소 패키지 노드로 설정합니다. | <ul style="list-style-type: none"> • FAILOVER_POLICY를 MIN_PACKAGE_NODE로 설정합니다. |
| <p>패키지가 장애 조치를 위해 노드 목록의 다음 노드로 이동됩니다(기본값).</p> | <ul style="list-style-type: none"> • 장애 조치 정책을 구성 노드로 설정합니다(기본값). | <ul style="list-style-type: none"> • FAILOVER_POLICY를 CONFIGURED_NODE로 설정합니다(기본값). |
| <p>패키지가 기본 노드가 아닌 다른 노드에서 실행되고 있을 때 기본 노드를 사용할 수 있게 되면 자동으로 패키지를 중지했다가 기본 노드에서 다시 시작합니다.</p> | <ul style="list-style-type: none"> • 장애 복구 정책을 자동으로 설정합니다. | <ul style="list-style-type: none"> • FAILBACK_POLICY를 AUTOMATIC으로 설정합니다. |

표 3-3 패키지 장애 조치 동작 (계속)

| 전환 동작 | Serviceguard Manager의 옵션 | ASCII 파일의 매개 변수 |
|---|--|--|
| <p>패키지가 기본 노드가 아닌 다른 노드에서 실행되고 있을 때 필요하면 수동으로 패키지를 기본 노드로 전환 시켜야 합니다.</p> | <ul style="list-style-type: none"> 장애 복구 정책을 수동으로 설정합니다(기본값). 장애 조치 정책을 구성된 노드로 설정합니다(기본값). | <ul style="list-style-type: none"> FAILBACK_POLICY를 MANUAL로 설정합니다(기본값). FAILOVER_POLICY를 CONFIGURED_NODE로 설정합니다(기본값). |
| <p>특정 서비스에서 장애가 발생하면 노드에서 TOC(제어권 전달) 즉, 정상적인 종료 절차를 따르지 않는 즉각적 중지가 일어난 후 모든 패키지가 전환됩니다. TOC 이전에 먼저 시스템 재부팅이 시도됩니다. 중지 스크립트는 실행되지 않습니다.</p> | <ul style="list-style-type: none"> 특정 서비스에 대해 서비스 빠른 장애 조치를 사용 가능으로 설정합니다. 모든 패키지에 대해 자동 전환을 사용 가능으로 설정합니다. | <ul style="list-style-type: none"> 특정 서비스에 대해 SERVICE_FAIL_FAST_ENABLED를 YES로 설정합니다. 모든 패키지에 대해 AUTO_RUN을 YES로 설정합니다. |
| <p>서비스 중 어느 하나에서라도 장애가 발생하면 TOC가 일어난 후 모든 패키지가 전환됩니다. TOC 이전에 먼저 시스템 재부팅이 시도됩니다.</p> | <ul style="list-style-type: none"> 모든 서비스에 대해 서비스 빠른 장애 조치를 사용 가능으로 설정합니다. 모든 패키지에 대해 자동 전환을 사용 가능으로 설정합니다. | <ul style="list-style-type: none"> 모든 서비스에 대해 SERVICE_FAIL_FAST_ENABLED를 YES로 설정합니다. 모든 패키지에 대해 AUTO_RUN을 YES로 설정합니다. |

패키지 제어 스크립트의 작동 방식

패키지는 Serviceguard가 구성된 응용 프로그램을 시작하고 중지하는 수단입니다. 패키지는 Serviceguard에서 장애 조치가 수행되는 단위이기도 합니다. 패키지는 고가용성을 위해 Serviceguard에서 관리하는 서비스, 디스크 볼륨, IP 주소 등의 모음입니다. 하나의 클러스터에 최대 150개의 패키지와 총 900개의 서비스가 있을 수 있습니다.

패키지 실행을 제어하는 요소

현재 실행되고 있지 않은 패키지가 클러스터 내의 적합한 노드에서 실행 가능하도록 구성되어 있는 것을 패키지 관리자가 인식하면 해당 패키지가 시작됩니다. 패키지를 실행할 수 있는 노드가 여러 개인 경우 패키지 관리자는 장애 조치 정책에 따라 패키지를 시작할 노드를 결정합니다. 따라서 사용자가 직접 `cmrunpkg` 명령을 사용할 필요가 없습니다. 대개는 패키지를 하나 이상의 노드에서 실행 가능하도록 지정하는 `cmmodpkg` 명령을 사용하여 패키지를 시작하는 것이 가장 좋습니다.

패키지 관리자는 모든 노드에서 패키지를 실행할 수 없는 경우가 아니면 패키지가 계속 실행 상태를 유지하게 합니다. 패키지가 실행될 수 없는 가장 일반적인 경우는 특정 노드에서 해당 패키지에 대해 `AUTO_RUN`이 비활성화되어 있거나 `NODE_SWITCHING`이 비활성화되어 있을 때입니다. 패키지가 한 노드에서 장애로 중단될 경우 해당 패키지를 실행할 수 있도록 지정된 다른 노드가 있으면 이 패키지는 자동으로 새 위치에서 시작됩니다. 이 프로세스를 **패키지 전환** 또는 **원격 전환**이라고 합니다.

패키지를 만들 때는 해당 패키지를 실행할 수 있는 노드 목록을 지정합니다. 표준 패키지는 한 번에 하나의 노드에서만 실행될 수 있으며 노드 목록의 다음으로 사용 가능한 노드에서 실행될 수 있습니다. `AUTO_RUN` 매개 변수를 `YES`로 설정하면 클러스터가 시작될 때 패키지가 자동으로 시작될 수 있습니다. 그러나 `AUTO_RUN`이 `NO`로 설정된 패키지는 클러스터가 시작되어도 자동으로 시작되지 않습니다. 따라서 이 종류의 패키지는 `cmmodpkg` 명령을 사용하여 명시적으로 활성화해야 합니다.

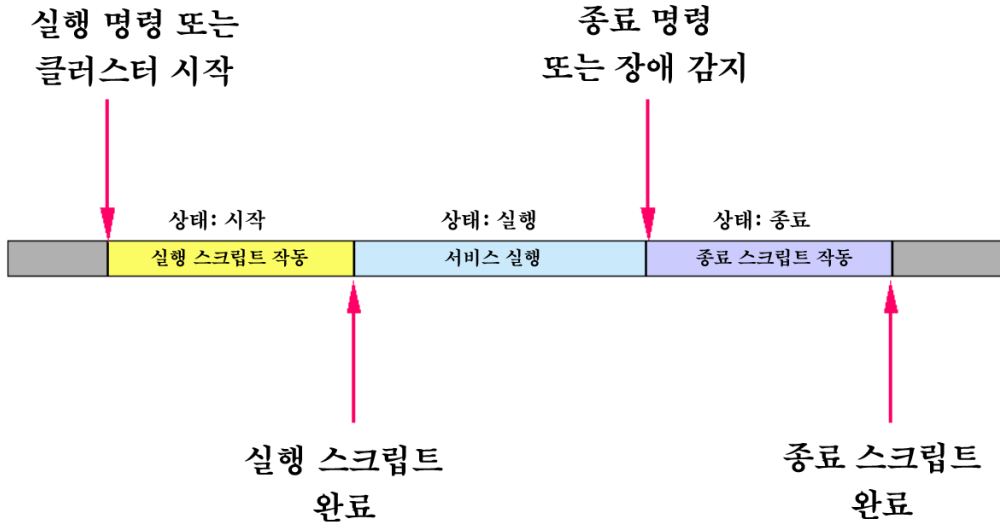
주

클러스터가 실행되고 있을 때 패키지를 구성한 경우에는 `cmapplyconf` 명령이 완료된 후 즉시 패키지가 시작되지 않습니다. 클러스터를 중지한 후 다시 시작하지 않고 패키지를 시작하려면 `cmrunpkg` 또는 `cmmodpkg` 명령을 실행해야 합니다.

패키지 제어 스크립트의 작동 방식

패키지가 시작되는 방식과 패키지 실행 중의 동작을 알고 있는 것이 좋습니다. 그림 3-13에서는 패키지의 작동 과정을 일부 보여 줍니다.

그림 3-13 중요 사건별로 본 패키지 작동 과정



다음은 패키지 작동 과정에서 가장 중요한 시점들입니다.

1. 제어 스크립트 시작 전
2. 실행 스크립트 실행 중
3. 서비스 실행 중
4. 서비스, 서버넷 또는 모니터링되는 리소스에서 장애가 발생할 때
5. 중지 스크립트 실행 중
6. 명령에 의해 패키지 또는 노드가 중지될 때
7. 노드에 장애가 발생할 때

제어 스크립트 시작 전

먼저 노드가 선택됩니다. 이 노드는 패키지의 노드 목록에 포함되어 있어야 하고 패키지의 장애 조치 정책에 맞아야 합니다. 또한 패키지에 필요한 모든 리소스를 선택된 노드에서 사용할 수 있어야 합니다. 이러한 리소스 중 하나가 패키지에 대해 모니터링되는 서버넷입니다. 서버넷을 사용할 수 없으면 해당 노드에서 패키지를 시작할 수 없습

니다. 또 다른 유형의 리소스는 모니터링되는 외부 리소스에 대한 의존성입니다. 모니터링 결과 구성된 리소스의 값 중 허용된 범위를 벗어나는 것이 있으면 패키지를 시작할 수 없습니다.

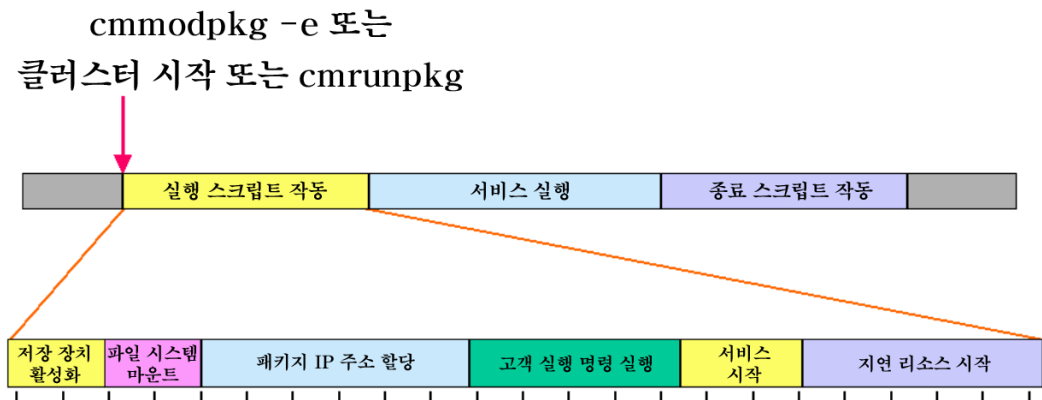
노드가 선택된 후에는 해당 노드에서 패키지를 시작할 수 있는지 확인하기 위한 검사가 수행됩니다. 그런 다음 선택된 노드에서 제어 스크립트에 의해 패키지 서비스가 시작됩니다. 엄격하게 말하면 선택된 노드에 있는 실행 스크립트를 사용하여 패키지가 시작되는 것입니다.

실행 스크립트 실행 중

패키지 관리자는 특정 노드에서 패키지를 시작할 수 있는지 확인한 후 실행 스크립트 ('start' 매개 변수를 사용하여 실행된 제어 스크립트)를 시작합니다. 이 스크립트는 다음 단계를 수행합니다(그림 3-14 참조).

1. 볼륨 그룹 또는 디스크 그룹을 활성화합니다.
2. 파일 시스템을 마운트합니다.
3. 노드의 LAN 카드에 패키지의 IP 주소를 할당합니다.
4. 사용자 정의 실행 명령을 실행합니다.
5. 각 패키지 서비스를 시작합니다.
6. 패키지가 필요로 하는 리소스 중에서 지연 시작으로 특별히 설정된 리소스를 시작합니다.
7. 종료 코드 0으로 종료합니다.

그림 3-14 실행 스크립트 실행 중의 패키지 동작 과정



어느 한 단계에서라도 오류가 발생하면 스크립트가 비정상적으로 종료됩니다(종료 코드 1). 예를 들어, 패키지 서비스를 시작할 수 없을 때는 제어 스크립트가 오류와 함께 종료됩니다.

또한 RUN_SCRIPT_TIMEOUT에 지정된 시간 안에 종료 스크립트의 실행이 완료되지 않으면 패키지 관리자가 스크립트를 강제 종료합니다. 실행 스크립트가 실행되는 동안에는 실행 스크립트와 같은 디렉토리에 있는 로그 파일에 메시지가 기록됩니다. 이 로그는 실행 스크립트와 이름이 같으며 확장자가 .log입니다. 정상적인 시작도 패키지 시작과 관련된 오류 메시지나 경고와 함께 기록됩니다.

주

패키지 실행 스크립트는 작업을 마친 후 종료됩니다. 즉, 패키지가 정상적으로 실행되고 나면 이 스크립트는 더 이상 실행되지 않습니다. 스크립트가 종료된 후 패키지 관리자는 스크립트에 의해 시작된 서비스의 PID를 직접 모니터링합니다. 모니터링하고 있는 서비스가 종료되면 패키지 관리자는 패키지 중지 스크립트를 실행하거나, SERVICE_FAILFAST_ENABLED가 YES로 설정되어 있는 경우에는 패키지가 실행되고 있는 노드를 중지합니다. 패키지 제어 스크립트에서 서비스의 Restarts 횟수가 지정되어 있으면 이 횟수 범위 내에서는 패키지 실행 스크립트를 다시 실행하지 않고 서비스가 다시 시작될 수 있습니다.

실행 스크립트의 정상 종료와 비정상 종료

실행 스크립트가 종료될 때의 종료 코드에 따라 패키지에 대한 다음 동작이 결정됩니다. 정상 종료는 패키지 시작이 성공적이었음을 의미하지만 그 외의 다른 모든 종료는 시작 작업이 성공적으로 완료되지 않았음을 의미합니다.

- 0 — 정상 종료. 패키지가 정상적으로 시작되었으므로 해당 노드에서 모든 서비스가 실행됩니다.
- 1 — 비정상 종료. NO_RESTART 종료라고도 합니다. 패키지가 모든 시작 단계를 정상적으로 완료되지 못했습니다. 이 경우 서비스가 강제 종료되고 패키지는 장애 조치를 위해 다른 노드로 이동하지 못하게 됩니다.
- 2 — 대체 종료. RESTART 종료라고도 합니다. 오류가 있기는 하지만 다른 노드에서 패키지가 시작될 수 있습니다. 오류가 있지만 다른 노드에서는 성공적으로 시작할 수 있는 패키지인 경우에는 사용자 정의 프로시저에서 이 유형의 종료를 사용할

수 있습니다.

RESTART 방식으로 종료된 패키지는 로컬 노드에서는 실행되지 못하지만 다른 노드에서는 실행될 수 있습니다.

- **Timeout** — `RUN_SCRIPT_TIMEOUT`이 초과될 때 발생하는 종료 유형. 이 경우에는 패키지가 강제로 종료되고 전체 노드에서 실행할 수 없게 됩니다. 그러나 현재 노드에서는 실행할 수 있습니다.

cmrunserv를 사용한 서비스 시작

패키지 제어 스크립트에서 `cmrunserv` 명령은 개별 서비스를 시작합니다. 이 명령은 파일에 기록된 각 서비스에 대해 한 번씩 실행됩니다. 각 서비스에는 재시작 횟수가 연관되어 있습니다. `cmrunserv` 명령은 이 횟수를 패키지 관리자에게 전달하고 패키지 관리자는 이에 따라 서비스 실행 실패 시 해당 횟수만큼 서비스를 다시 시작합니다. 일반적인 설정은 다음과 같습니다.

```
SERVICE_RESTART[0]=" " ; do not restart
SERVICE_RESTART[0]="-r <n>" ; restart as many as <n> times
SERVICE_RESTART[0]="-R" ; restart indefinitely
```

주

재시작 횟수를 `<n>`으로 설정하고 `SERVICE_FAILFAST_ENABLED`를 YES로 설정한 경우 재시작 시도가 `<n>`번 실패하고 나면 빠른 장애 조치가 발생합니다. 서비스에 대해 `SERVICE_RESTART`를 "-R"로 설정하고 동시에 `SERVICE_FAILFAST_ENABLED`를 YES로 설정하는 것은 좋지 않습니다.

서비스 실행 중

클러스터 서비스가 정상적으로 실행되는 동안 패키지 관리자는 다음 사항을 지속적으로 모니터링합니다.

- 서비스의 프로세스 ID
- 패키지 구성 파일에서 모니터링 대상으로 구성된 서브넷
- 패키지가 종속되어 있는 구성된 리소스

일부 장애는 로컬 전환을 발생시키기도 합니다. 예를 들어, 특정 LAN 카드에 장애가 발생한 경우 해당 서버넷에 대기 LAN 카드가 구성되어 있으면 네트워크 관리자는 정상인 LAN 카드로 전환시킵니다. 서비스에 장애가 발생했는데 해당 서비스의 RESTART 매개 변수가 0보다 큰 값으로 설정된 경우에는 패키지를 중지하지 않고 구성된 횟수만큼 서비스가 다시 시작됩니다.

구성된 EMS 리소스 의존성이 있고 이벤트를 발생시키는 트리거가 있는 경우 패키지는 중지됩니다.

모든 서비스가 실행되는 정상 작동 상태에서는 `cmviewcl` 명령의 출력 결과 중 “Script Parameters” 섹션에서 서비스 상태를 확인할 수 있습니다.

서비스, 서버넷 또는 모니터링되는 리소스에서 장애가 발생할 때

장애가 발생한 경우 패키지 제어 스크립트의 작동 과정은 다음과 같습니다. 서비스에서 장애가 발생했는데 더 이상 재시작을 시도할 수 없거나, 서버넷에서 장애가 발생했는데 대기 LAN이 없거나, 구성된 리소스에서 장애가 발생한 경우, 패키지는 현재 노드에서 종료되며 패키지 전환 플래그의 설정에 따라 다른 노드에서 다시 시작될 수 있습니다.

패키지가 정상적으로 중지된다는 것은 패키지 중지 스크립트가 실행되는 것을 의미합니다(다음 절 참조). 그러나 장애가 발생한 서비스에 대해 `SERVICE_FAILFAST_ENABLED`가 YES로 설정된 경우에는 장애가 감지되는 즉시 노드가 중지됩니다. 이 플래그가 설정되지 않은 경우에는 서비스에 장애가 발생해도 중지 스크립트를 실행하여 패키지를 정상적으로 중지하게 됩니다.

`AUTO_RUN`이 YES로 설정되어 있는 경우에는 시작에 필요한 모든 조건이 충족되면 적합한 다른 노드에서 패키지가 시작됩니다. `AUTO_RUN`이 NO로 설정되어 있으면 패키지는 단지 중지만 되고 다른 어느 노드에서도 시작되지 않습니다.

주

패키지가 서버넷에 종속된 경우 기본 노드의 서버넷에 장애가 발생하면 패키지 종료가 시작됩니다. 대체 노드에서 패키지가 다시 시작되기 전에 즉시 서버넷이 복구되면 기본 노드에서 패키지가 다시 시작될 수 있습니다. 따라서 이 경우 패키지는 클러스터의 다른 노드로 전환되지 않습니다.

명령에 의해 패키지가 중지될 때

Serviceguard `cmhaltpkg` 명령을 실행하면 특정 패키지에 대해 실행 중인 서비스를 중지하는 패키지 중지 스크립트를 실행하는 것과 같은 결과가 나타납니다. 이 명령은 패키지를 정상적으로 종료한 후 자동 패키지 시작(AUTO_RUN)을 비활성화합니다.

주

`-n <nodename>` 옵션과 함께 `cmhaltpkg` 명령을 실행하면 해당 노드에서 패키지가 실행 중인 경우에만 패키지가 중지됩니다.

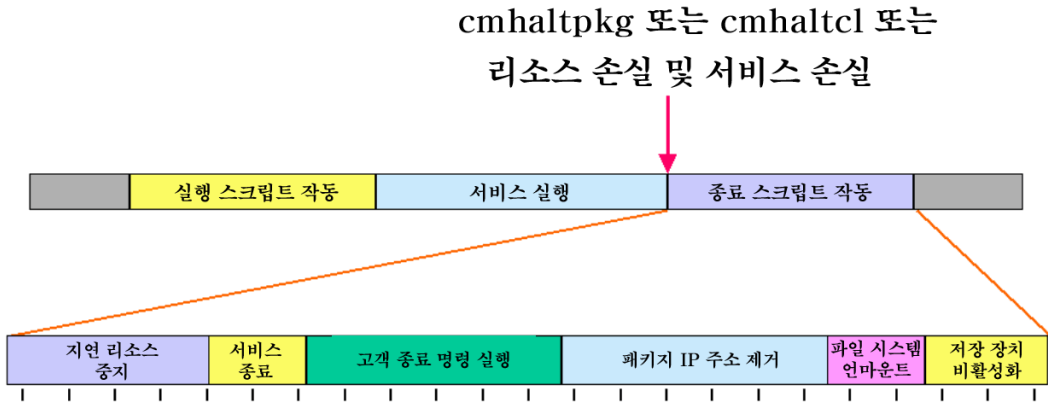
`cmmodpkg` 명령은 패키지를 중지하는 데는 사용될 수 없지만 패키지가 특정 노드나 모든 노드로 전환되지 못하게 할 수 있습니다. 전환이 불가능하게 설정되어도 패키지는 계속 실행됩니다. 하지만 패키지가 현재 노드에서 실행 중지된 경우에는 다른 노드에서 시작될 수 없습니다.

중지 스크립트 실행 중

패키지 관리자가 서비스 장애를 감지한 경우나 특정 패키지에 대해 `cmhaltpkg` 명령이 실행된 경우 중지 스크립트(즉, 'halt' 매개 변수를 통해 실행된 제어 스크립트)가 시작됩니다. 이 스크립트는 다음 단계를 수행합니다(그림 3-15 참조).

1. 이미 시작된 지연 시작 리소스를 중지합니다.
2. 모든 패키지 서비스를 중지합니다.
3. 사용자 정의 중지 명령을 실행합니다.
4. 노드의 LAN 카드에서 패키지 IP 주소를 제거합니다.
5. 파일 시스템의 마운트를 해제합니다.
6. 볼륨 그룹을 비활성화합니다.
7. 종료 코드 0으로 종료합니다.

그림 3-15 중지 스크립트 실행 중의 패키지 동작 과정



어느 한 단계에서라도 오류가 발생하면 스크립트가 비정상적으로 종료됩니다(종료 코드 1). 또한 HALT_SCRIPT_TIMEOUT에서 지정된 시간 안에 중지 스크립트의 실행이 완료되지 않으면 패키지 관리자가 스크립트를 강제 종료합니다. 중지 스크립트가 실행되는 동안에는 중지 스크립트와 같은 디렉토리에 있는 로그 파일에 메시지가 기록됩니다. 이 로그는 중지 스크립트와 이름이 같으며 확장자가 .log입니다. 정상적인 시작도 패키지 중지과 관련된 오류 메시지나 경고와 함께 로그에 기록됩니다.

중지 스크립트의 정상 종료와 비정상 종료

패키지가 다른 노드로 이동할 수 있는지 여부는 중지 스크립트가 종료될 때의 종료 코드에 따라 결정됩니다. 가능한 종료 코드는 다음과 같습니다.

- 0 — 정상 종료. 패키지가 정상적으로 중지되었으므로 해당 노드에서 모든 서비스가 중지됩니다.
- 1 — 비정상 종료. NO_RESTART 종료라고도 합니다. 패키지가 정상적으로 중단되지 않았습니다. 이 경우 서비스가 강제 종료되고 패키지는 전체 노드에서 실행할 수 없게 됩니다. 그러나 현재 노드에서는 실행할 수 있습니다.
- Timeout — HALT_SCRIPT_TIMEOUT이 초과될 때 발생하는 종료 유형. 이 경우에는 패키지가 강제로 종료되고 전체 노드에서 실행할 수 없게 됩니다. 그러나 현재 노드에서는 실행할 수 있습니다.

패키지 제어 스크립트 오류 및 종료 조건

표 3-4는 오류 조건, 빠른 장애 조치 설정 및 장애 조치 패키지의 패키지 이동에 대한 가능한 조합을 나타낸 표입니다.

표 3-4 오류 조건 및 패키지 이동

| 패키지 오류 조건 | | | 결과 | | | |
|---------------|-------------------|--------------------|---------------------|-----------------------|------------------------|-------------------|
| 오류 또는 종료 코드 | 노드 빠른 장애 조치 사용 가능 | 서비스 빠른 장애 조치 사용 가능 | 오류 후 기본노드의 HP-UX 상태 | 오류 또는 종료 후 중지 스크립트 실행 | 오류 후 기본 노드에서 패키지 실행 허용 | 대체 노드에서 패키지 실행 허용 |
| 서비스 장애 | 예 | 예 | TOC | 아니요 | 해당 없음(TOC) | 예 |
| 서비스 장애 | 아니요 | 예 | TOC | 아니요 | 해당 없음(TOC) | 예 |
| 서비스 장애 | 예 | 아니요 | 계속 실행 | 예 | 아니요 | 예 |
| 서비스 장애 | 아니요 | 아니요 | 계속 실행 | 예 | 아니요 | 예 |
| 실행 스크립트 종료 1 | 모든 설정 | 모든 설정 | 계속 실행 | 아니요 | 변화 없음 | 아니요 |
| 실행 스크립트 종료 2 | 예 | 모든 설정 | TOC | 아니요 | 해당 없음(TOC) | 예 |
| 실행 스크립트 종료 2 | 아니요 | 모든 설정 | 계속 실행 | 아니요 | 아니요 | 예 |
| 실행 스크립트 시간 종료 | 예 | 모든 설정 | TOC | 아니요 | 해당 없음(TOC) | 예 |
| 실행 스크립트 시간 종료 | 아니요 | 모든 설정 | 계속 실행 | 아니요 | 변화 없음 | 아니요 |
| 중지 스크립트 종료 1 | 예 | 모든 설정 | 계속 실행 | 해당 없음 | 예 | 아니요 |
| 중지 스크립트 종료 1 | 아니요 | 모든 설정 | 계속 실행 | 해당 없음 | 예 | 아니요 |

표 3-4 오류 조건 및 패키지 이동(계속)

| 패키지 오류 조건 | | | 결과 | | | |
|---------------|-------------------|--------------------|---------------------|-----------------------|------------------------|--|
| 오류 또는 종료 코드 | 노드 빠른 장애 조치 사용 가능 | 서비스 빠른 장애 조치 사용 가능 | 오류 후 기본노드의 HP-UX 상태 | 오류 또는 종료 후 중지 스크립트 실행 | 오류 후 기본 노드에서 패키지 실행 허용 | 대체 노드에서 패키지 실행 허용 |
| 중지 스크립트 시간 종료 | 예 | 모든 설정 | TOC | 해당 없음 | 해당 없음(TOC) | 예 (cmhaltpkg 명령이 실행된 후에 시간종료가 발생한 경우가 아닐 때) |
| 중지 스크립트 시간 종료 | 아니요 | 모든 설정 | 계속 실행 | 해당 없음 | 예 | 아니요 |
| 서비스 장애 | 모든 설정 | 예 | TOC | 아니요 | 해당 없음(TOC) | 예 |
| 서비스 장애 | 모든 설정 | 아니요 | 계속 실행 | 예 | 아니요 | 예 |
| 네트워크 손실 | 예 | 모든 설정 | TOC | 아니요 | 해당 없음(TOC) | 예 |
| 네트워크 손실 | 아니요 | 모든 설정 | 계속 실행 | 예 | 예 | 예 |
| 모니터링되는 리소스 손실 | 예 | 모든 설정 | TOC | 아니요 | 해당 없음(TOC) | 예 |

표 3-4 오류 조건 및 패키지 이동(계속)

| 패키지 오류 조건 | | | 결과 | | | |
|---------------|-------------------|--------------------|---------------------|-----------------------|--|-------------------|
| 오류 또는 종료 코드 | 노드 빠른 장애 조치 사용 가능 | 서비스 빠른 장애 조치 사용 가능 | 오류 후 기본노드의 HP-UX 상태 | 오류 또는 종료 후 중지 스크립트 실행 | 오류 후 기본 노드에서 패키지 실행 허용 | 대체 노드에서 패키지 실행 허용 |
| 모니터링되는 리소스 손실 | 아니오 | 모든 설정 | 계속 실행 | 예 | 예(지연 시작 리소스가 아닌 경우)/아니오(지연 시작 리소스인 경우) | 예 |

네트워크 관리자의 작동 방식

네트워크 관리자의 목적은 네트워크 카드 및 케이블 장애를 감지하고 복구하여 클라이언트가 네트워크 서비스를 계속 사용할 수 있게 유지하는 것입니다. 실제로 이는 각 패키지의 IP 주소를 패키지가 실행 중인 노드의 주 LAN 인터페이스 카드에 할당하고 모든 인터페이스의 상태를 모니터링하여 필요할 때 전환하는 것을 의미합니다.

고정 IP 주소와 변동 가능 IP 주소

각 노드(호스트 시스템)에는 각각의 활성 네트워크 인터페이스에 대한 IP 주소가 적어도 하나 있어야 합니다. **고정 IP 주소**라고 하는 이 주소는 노드의 `/etc/rc.config.d/netconf` 파일이나 `/etc/rc.config.d/netconf-ipv6` 파일에서 구성됩니다. 고정 IP 주소는 다른 노드로 전달될 수 없지만 대기 LAN 인터페이스 카드로는 전달될 수 있습니다. 고정 IP 주소는 패키지와 **연관되지 않습니다**. 고정 IP 주소는 하트비트 메시지(앞에서 설명한 “클러스터 관리자의 작동 방식” 절 참조)와 기타 데이터를 전송하는 데 사용됩니다.

일반적으로 고정 IP 주소 외에도 각 패키지에 고유한 IP 주소를 하나 이상 할당합니다. 패키지 IP 주소는 패키지가 시작될 때 패키지 제어 스크립트에 있는 `cmmodnet` 명령으로 주 LAN 인터페이스 카드에 할당됩니다. 패키지와 연관된 IP 주소는 실제로 한 클러스터 노드에서 다른 노드로 이동할 수 있으므로 **변동 가능 IP 주소**라고 합니다(**패키지 IP 주소** 또는 **유동 IP 주소**라고도 함). 한 클러스터에서 150개의 패키지에 걸쳐 최대 200개의 변동 가능 IP 주소를 사용할 수 있습니다. 이 주소는 IPv4와 IPv6 주소의 조합이 될 수 있습니다.

변동 가능 IP 주소는 패키지에 할당되는 가상 호스트 IP 주소와 같습니다. DNS(도메인 이름 시스템)를 통해 각 패키지 이름을 구성하는 것이 좋습니다. 그러면 프로그램에서는 패키지의 변동 가능 IP 주소를 반환하는 `gethostbyname()`을 호출할 때 입력 값으로 사용하는 호스트 이름처럼 패키지 이름을 사용할 수 있습니다.

고정 IP 주소와 변동 가능 IP 주소는 모두 LAN 카드에 장애가 발생할 경우 대기 LAN 인터페이스로 전환됩니다. 또한 패키지의 제어권이 전달되면 대체 노드에서 변동 가능 주소(고정 주소가 아님)를 사용할 수 있습니다. 즉, 응용 프로그램이 현재 패키지가 있는 노드를 몰라도 변동 가능 주소를 통해 패키지에 액세스할 수 있습니다.

IP 주소의 유형

Serviceguard에서는 IPv4 및 IPv6 주소 유형을 모두 지원합니다. IPv4 주소는 “n.n.n.n” 형식의 전통적인 주소로, 여기서 ‘n’은 0부터 255까지의 십진수입니다. IPv6 주소는 “x:x:x:x:x:x:x” 형식으로, 여기서 ‘x’는 128비트 주소를 구성하는 8개의 16비트 단위 각각에 대한 16진수 값입니다. IPv4 주소만 하트비트 주소로 사용할 수 있지만 IPv4 및 IPv6 주소는 다양한 조합을 포함하여 모두 클러스터에서 고정 IP로 정의될 수 있습니다. IPv4와 IPv6 주소는 모두 변동 가능(패키지) IP 주소로도 사용될 수 있습니다.

변동 가능 IP 주소 추가 및 삭제

패키지가 시작되면 변동 가능 IP 주소를 지정된 IP 서브넷에 추가할 수 있습니다. 패키지가 중지되면 변동 가능 IP 주소는 지정된 서브넷에서 삭제됩니다. 변동 가능 IP 주소의 추가 및 삭제는 패키지 제어 스크립트에서 `cmmodnet` 명령을 통해 처리됩니다. 자세한 내용은 “패키지 및 패키지 서비스 구성” 장을 참조하십시오.

IP 주소는 각각의 주 네트워크 인터페이스 카드에서만 구성됩니다. 대기 카드는 IP 주소로 구성되지 않습니다. 동일한 네트워크 카드에 있는 여러 IPv4 주소는 동일한 IP 서브넷에 속해야 합니다.

로드 공유

하나의 패키지에는 같은 IP 주소와 연관된 서비스가 여러 개 있을 수 있습니다. 한 서비스가 새 시스템으로 이동하면 해당 IP 주소를 사용하는 다른 서비스도 이동합니다. 각 서비스에서 자체 패키지를 만들고 이 패키지에 고유 IP 주소를 할당하여 로드를 공유할 수 있습니다. 이렇게 하면 관리자는 선택한 서비스를 로드가 보다 적은 시스템으로 이동할 수 있습니다.

LAN 인터페이스 모니터링 및 장애 감지

Serviceguard에서는 주기적으로 클러스터 구성 파일에 지정된 모든 네트워크 인터페이스 카드를 폴링합니다. 각 노드에서 다음과 같은 방식으로 네트워크 장애가 감지됩니다. 노드 상의 한 인터페이스가 폴러로 지정됩니다. 폴러는 해당 노드에서 자신이 속한 브리지 연결 네트워크에 있는 다른 기본 인터페이스와 대기 인터페이스를 폴링하여 상태가 양호한지 확인합니다. 일반적으로 폴러는 대기 인터페이스입니다. 대기 인터페이스가 없는 브리지 연결 네트워크에서는 기본 인터페이스에 폴링 작업이 할당됩니다. 브리지 연결 네트워크에 대해서는 2장의 “중복 네트워크 구성 요소” 절을 참조하십시오.

폴링 인터페이스는 자신이 속한 브리지 연결 네트워크에 있는 노드의 다른 모든 인터페이스로 LAN 패킷을 보내고 응답 패킷을 받습니다.

LAN 드라이버에서 오류를 보고할 때마다 Serviceguard는 즉시 해당 카드에 오류가 있음을 선언하고 해당되는 경우 로컬 전환을 수행합니다. 예를 들어, 카드에 전송 장애가 있으면 Serviceguard는 즉시 오류 알림을 받고 해당 카드가 작동 중지된 것으로 표시합니다.

또한 Serviceguard 네트워크 관리자는 인터페이스에서 보내고 받은 패킷 수를 조사하여 카드에 문제가 있는지를 확인합니다. Serviceguard에서는 두 가지 방법으로 송수신 패킷 수를 처리할 수 있습니다. 클러스터 구성 파일에서는 NETWORK_FAILURE_DETECTION 매개 변수에 대해 다음 두 값 중 하나를 선택합니다.

- **INOUT**: 미리 지정된 시간 동안 들어오고 나가는 패킷의 수가 모두 증가하지 않으면 Serviceguard에서는 카드에 오류가 있는 것으로 선언합니다. 들어오는 패킷이 나가는 패킷 중 어느 한 쪽만 증가하지 않는 경우에는 카드에 오류가 있는 것으로 선언하지 않습니다. 즉, **두 패킷 수가 모두** 증가하지 않아야 합니다. 이것이 기본값입니다.
- **INONLY_OR_INOUT**: 이 옵션도 들어오고 나가는 패킷 수가 증가하지 않으면 카드에 오류가 있는 것으로 선언합니다. 그러나 들어오는 패킷 수만 증가하지 않는 경우에도 카드에 오류가 있는 것으로 선언한다는 점이 다릅니다.

이 옵션은 Serviceguard A.11.16에서 새로 추가된 옵션으로, 모든 환경에 적합하지는 않습니다. 이 옵션을 선택하려면 먼저 다음 조건이 충족되어야 합니다.

- 클러스터의 모든 브리지 연결 네트워크에 각각 셋 이상의 인터페이스가 있어야 합니다.
- 각각의 기본 인터페이스에는 하나 이상의 대기 인터페이스가 있어야 하고 기본 인터페이스는 대기 스위치에 연결되어야 합니다.
- 주 스위치는 대기 인터페이스에 직접 연결되어야 합니다.
- 모든 브리지 연결 네트워크에 단순 장애 요인(Single Point of Failure)이 없어야 합니다.

로컬 전환

로컬 네트워크 전환에는 로컬 네트워크 인터페이스 장애를 감지하는 과정과 장애 조치를 위해 로컬 백업 LAN 카드로 이동하는 과정이 포함됩니다. 백업 LAN 카드에는 IP 주소가 구성되어 있지 않아야 합니다.

로컬 네트워크 전환의 경우 이더넷의 TCP/IP 연결은 손실되지 않지만 IEEE 802.3 연결은 손실됩니다. IPv4의 경우 이더넷, 토큰 링 및 FDDI는 ARP 프로토콜을 사용하고 HP-UX는 임의의 ARP를 보내 MAC(링크 수준) 주소와 IP 수준 주소 간의 주소 매핑을 원격 시스템에 알립니다. IEEE 802.3에는 *rearp* 기능이 없습니다.

IPv6에서는 ARP 대신 NDP(Neighbor Discovery Protocol)를 사용합니다. NDP 프로토콜은 호스트와 라우터에서 다음을 수행하는 데 사용됩니다.

- 같은 링크에 있는 인접 장치의 링크 계층 주소를 확인하고 무효화된 캐시된 값을 신속하게 제거합니다.
- 대신 패킷을 전달할 인접 라우터를 찾습니다.
- 연결할 수 있는 인접 장치와 연결할 수 없는 인접 장치를 적극적으로 추적하고 변경된 링크 계층 주소를 감지합니다.
- 라우터 경로에 장애가 발생하면 정상적으로 작동하는 대체 라우터를 찾습니다.

다음과 같은 이더넷 계열에서는 로컬 전환 구성이 지원됩니다.

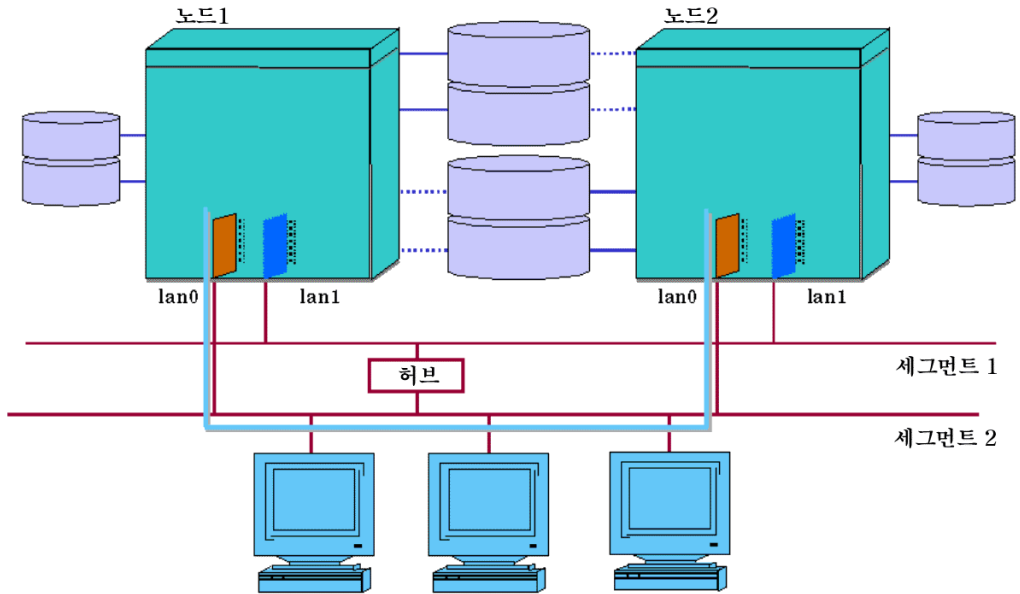
- 1000Base-SX 및 1000Base-T
- 1000Base-T 또는 1000BaseSX와 100Base-T

그러나 HP-UX 11i에서는 100Base-T 카드가 점보 프레임 지원하지 않기 때문에 점보 프레임을 사용하면 안 됩니다. 1000Base-T나 1000Base-SX를 사용하는 네트워크 인터페이스 카드는 10BaseT로 로컬 장애 조치를 수행할 수 없습니다.

전송 중에 IP 패킷은 손실되지만 TCP(전송 제어 프로토콜)는 패킷을 다시 전송합니다. UDP(사용자 데이터그램 프로토콜)의 경우 패킷은 프로토콜에 의해 자동으로 다시 전송되지 않습니다. 그러나 UDP는 불안정한 서비스이므로 UDP 응용 프로그램은 네트워크 패킷 손실과 복구 작업을 적절하게 처리할 준비가 되어 있어야 합니다. 로컬 전환은 동일한 유형의 두 LAN 사이에서만 지원됩니다. 예를 들어, 이더넷과 FDDI 인터페이스 사이의 로컬 전환은 지원되지 않지만 10BT 이더넷과 100BT 이더넷 사이의 로컬 전환은 지원됩니다.

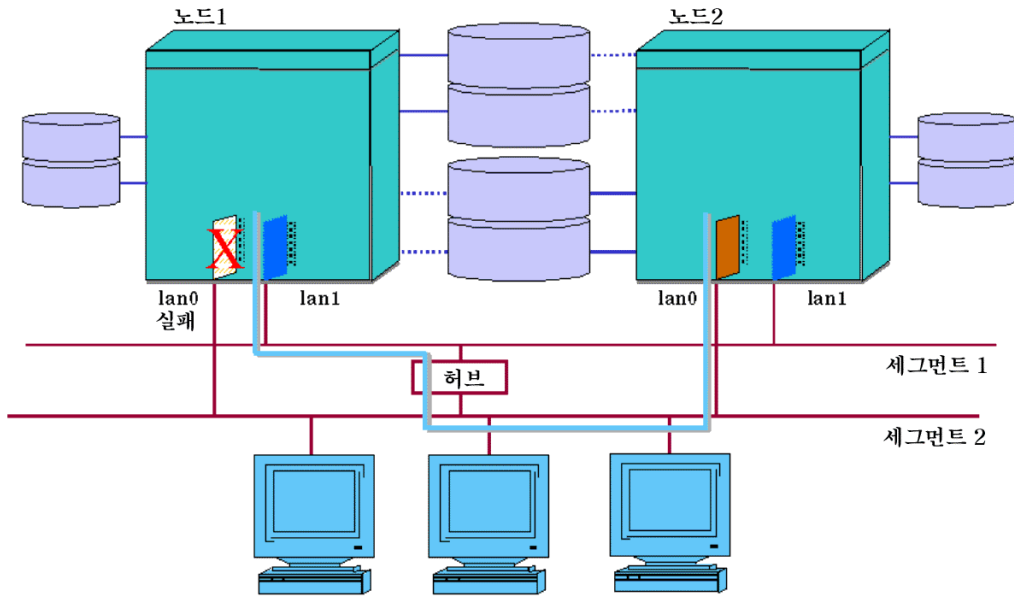
그림 3-16에서는 하나의 브리지 연결 네트워크에서 연결된 두 노드를 보여 줍니다. LAN 세그먼트 1과 2는 허브로 연결되어 있습니다.

그림 3-16 로컬 네트워크 전환 이전의 클러스터



노드 1과 노드 2는 LAN 세그먼트 2를 통해 통신하며 LAN 세그먼트 1은 대기용입니다. 그림 3-17에서는 노드 1의 LAN 세그먼트 2 네트워크 인터페이스 카드에 장애가 발생한 경우를 보여 줍니다.

그림 3-17 로컬 네트워크 전환 이후의 클러스터



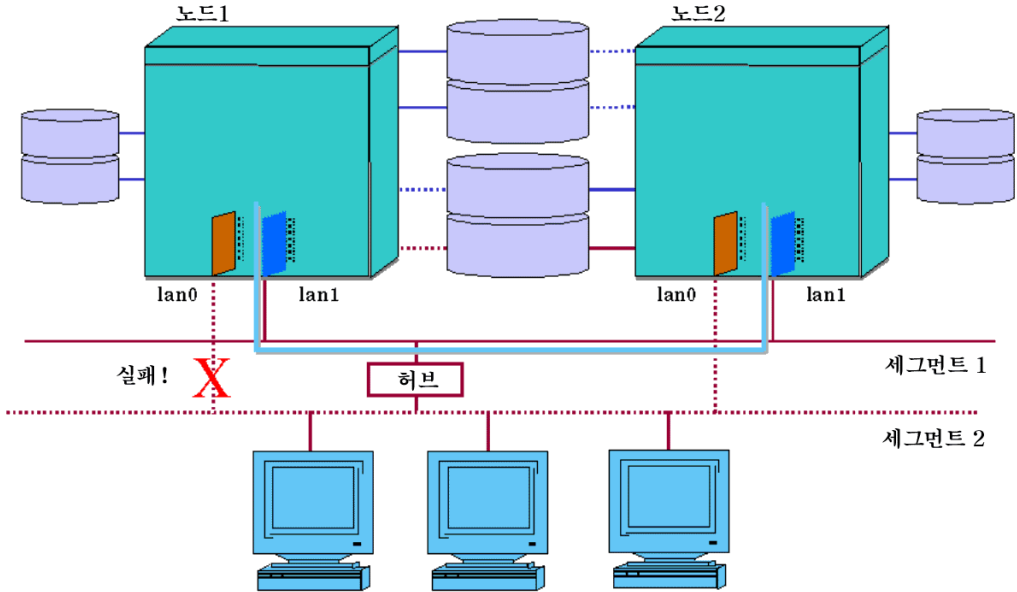
대기 인터페이스가 인계를 받으면 IP 주소는 대기 인터페이스와 연관된 하드웨어 경로로 전환됩니다. 전환은 TCP/IP 수준에서 투명하게 이루어집니다. 모든 응용 프로그램은 원래 노드에서 계속 실행됩니다. 이때 노드 1의 IP 트래픽은 전송되는 동안 지연됩니다. 그러나 TCP/IP 연결은 계속 유지되고 응용 프로그램은 계속 실행됩니다. 노드 1의 패키지 제어는 영향을 받지 않습니다.

주

이더넷 네트워크에서 Serviceguard는 “이더넷 프로토콜”로 구성된 네트워크 인터페이스나 “IEEE 802.3 프로토콜에서 캡슐화된 SNAP”로 구성된 네트워크 인터페이스 사이의 로컬 장애 조치를 지원합니다. 한 인터페이스에서 두 프로토콜을 모두 사용할 수는 없으며 서로 다른 프로토콜을 사용하는 인터페이스 사이에서는 로컬 장애 조치를 수행할 수 없습니다.

그림 3-18에서는 로컬 전환의 또 다른 예를 보여 줍니다. 이 경우 세그먼트 2에 영향을 주는 장애가 발생하여 양쪽 노드 모두 세그먼트 1에 연결된 LAN 카드로 전환됩니다.

그림 3-18 케이블 장애 후의 로컬 전환



로컬 네트워크 전환은 노드가 하나 이상 포함된 클러스터에만 적용됩니다. 노드가 하나만 필요하고 더 복잡한 클러스터를 설정하고 싶지 않을 때 이 로컬 네트워크 전환 기능을 사용하려면 단일 노드 클러스터를 설계하면 됩니다.

로컬 전환 후 주 LAN 인터페이스로의 복귀 전환

노드가 중지될 때마다 클러스터 데몬(cmc1d)은 Serviceguard가 구성하여 대기 인터페이스에서 실행 중인 서브넷을 기본 인터페이스로 전환하려고 합니다. 이것은 기본 인터페이스의 링크 상태에 관계없이 수행됩니다. 이렇게 복귀시키는 이유는 클러스터가 시작되기 전의 원래 네트워크 구성을 보존하기 위한 것입니다. cmhaltnode 명령이 실행되면 특정 노드에 대해 복귀 전환이 일어나며 cmhaltc1 명령이 실행되면 클러스터의 모든 노드에 대해 복귀 전환이 일어납니다.

원격 전환

원격 전환(즉, 패키지 전환)은 패키지와 관련 IP 주소를 새로운 시스템으로 옮기는 과정입니다. 새 시스템에는 이미 동일한 서브넷이 구성되어 제대로 작동하고 있어야 합니다. 그렇지 않으면 패키지가 시작되지 않습니다. 원격 전환에서 TCP 연결은 손실됩니다.

TCP 응용 프로그램은 자동으로 다시 연결되지 않으므로 수동으로 다시 연결해야 합니다. 패키지가 여러 서브넷에 종속되어 있으면 패키지가 시작되기 전에 대상 노드에서 모든 서브넷을 사용할 수 있어야 합니다.

원격 전환은 동일한 유형의 LAN 사이에서만 지원됩니다. 예를 들어, 한 시스템의 이더넷과 장애 조치 시스템의 FDDI 인터페이스 사이의 원격 전환은 지원되지 않습니다. 변동 가능 IP 주소의 원격 전환은 앞에 나온 그림 3-5와 그림 3-6을 참조하십시오.

전환 후의 주소 확인 메시지

유동 IPv4 주소가 로컬이나 원격에서 새 인터페이스로 이동하면 IP 주소와 링크 계층 주소 간의 새 매핑을 나타내기 위해 ARP 메시지가 브로드캐스팅됩니다. 이동된 각 IPv4 주소에 대해 ARP 메시지가 전송됩니다. 브로드캐스트를 받는 모든 시스템에서는 연관된 ARP 캐시 항목을 업데이트하여 변경 사항을 반영해야 합니다. 현재 ARP 메시지는 IP 주소가 새 시스템에 추가될 때 전송됩니다. ARP 메시지는 ARP 요청의 형태로 전송됩니다. ARP 요청 메시지의 송신자 및 수신자 프로토콜 주소 필드는 모두 동일한 유동 IP 주소로 설정됩니다. 이렇게 하면 메시지를 받는 노드에서 응답을 보내지 않습니다.

IPv4와 달리 IPv6 주소는 NDP 메시지를 사용하여 인접 장치의 링크 계층 주소를 확인합니다.

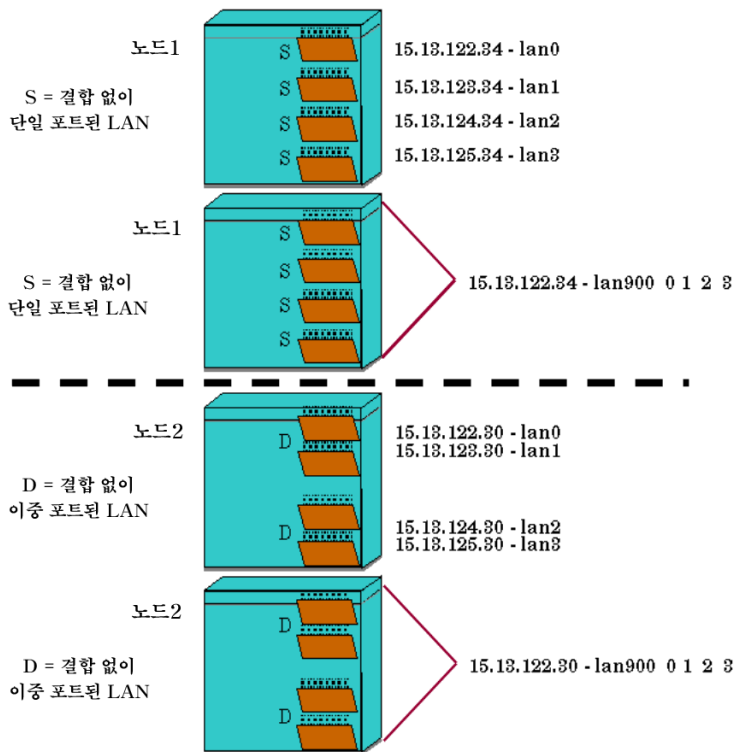
자동 포트 통합

Serviceguard에서는 HP-APA(Auto-Port Aggregation, HP 제품 번호 J4240AA)를 통해 자동 포트 통합의 사용을 지원합니다. HP-APA는 물리적인 여러 개의 고속 이더넷이나 기가비트 이더넷 포트를 하나의 논리 링크 집합으로 통합하는 네트워킹 기술입니다. HP-APA를 사용하면 여러 개의 100Mbps 고속 이더넷 링크나 1Gbps 이더넷 링크(또는 각각 200Mbps와 2Gbps의 전이중 이더넷 링크)를 기초로 유연하고 확장 가능한 대역폭을 얻을 수 있습니다. 그 밖의 이점으로는 물리적 링크 간의 로드 균형 조정, 자동 결함 감지 및 고가용성이 필요한 환경에 대한 복구 등이 있습니다. 포트 통합 기능은 링크 통합 또는 트렁킹이라고도 합니다. APA는 이중 스택 커널에서도 지원됩니다.

각 링크 집합을 사용할 수 있게 되면 IP 및 MAC 주소가 하나뿐인 여러 물리 포트의 단일 논리 링크로 볼 수 있습니다. HP-APA는 최대 4개의 물리 포트를 하나의 링크 집합으로 통합할 수 있습니다. 시스템당 허용되는 링크 집합의 수는 50개입니다. 빈 링크 집합에는 MAC 주소가 없습니다.

다중 포트 네트워크 카드에서 포트를 통합할 수 있습니다. 현재 최대 4개의 포트가 있는 카드를 사용할 수 있습니다. 또는 다른 카드에서 포트를 결합할 수 있습니다. 그림 3-19에서는 두 가지 예를 보여 줍니다.

그림 3-19 통합된 네트워크 포트



통합되지 않은 구성의 단일 및 이중 포트 LAN에는 네 개의 LAN 카드가 있습니다. 각 카드는 통합되지 않은 별도의 IP 주소 및 MAC 주소와 연관되어 있고 고유한 LAN 이름(lan0, lan1, lan2, lan3)을 갖고 있습니다. 이러한 포트가 통합되면 네 개의 포트 모두가 하나의 IP 주소 및 MAC 주소와 연관됩니다. 이 예제에서 통합된 포트 집합의 이름은 lan900이며 이 이름은 HP-UX 11i에서 이 집합의 이름이 됩니다. HP-UX 11.0에서는 집합 이름이 lan100으로 시작합니다.

이더넷 카드 종류(단일 또는 이중 포트)와 통합 그룹을 다양하게 조합할 수 있습니다. 그러나 하트비트 연결 시 단순 장애 요인을 없애려면 APA를 조합할 때 두 개 이상의 실제 카드를 사용해야 합니다. HP-APA는 현재 링크 집합의 자동 및 수동 구성을 모두 지원합니다.

Serviceguard를 사용하여 APA를 구현하는 방법에 대한 자세한 내용은 <http://docs.hp.com>의 Networking and Communications 모음에서 *HP Auto Port Aggregation (APA) Support Guide* 및 기타 APA 설명서를 참조하십시오.

VLAN 구성

현재 Serviceguard 클러스터에서는 HP-UX VLAN 소프트웨어를 사용한 가상 LAN 구성이 지원됩니다. VLAN은 이중 스택 커널에서도 지원됩니다.

VLAN의 개념

가상 LAN(또는 VLAN)은 물리적 위치에 상관없이 연관 규칙에 따라 네트워크 노드를 그룹화하는 네트워킹 기술입니다. 특히 VLAN을 사용하면 물리 LAN을 여러 개의 논리 LAN 세그먼트나 브로드캐스트 도메인으로 나눌 수 있습니다. 논리 LAN을 구성할 때 각 인터페이스에는 태그 ID가 지정됩니다. 동일한 태그 ID를 공유하는 VLAN 인터페이스는 동일한 물리 네트워크에 있는 것처럼 서로 통신할 수 있습니다. 가상 LAN을 만들면 브로드캐스트 트래픽이 감소하고 네트워크 성능 및 보안이 향상되며 관리 기능이 개선됩니다. HP-UX에서 초기 VLAN 연관 규칙은 포트, IP 및 프로토콜을 기반으로 합니다. 하나의 물리 LAN 인터페이스에서 여러 개의 VLAN 인터페이스를 구성할 수 있습니다. 이렇게 구성된 VLAN 인터페이스는 응용 프로그램에서 일반 네트워크 인터페이스로 인식됩니다. 이러한 VLAN 인터페이스에 IP 주소를 할당하여 자체 서브넷을 형성할 수 있습니다. VLAN 인터페이스 구성 방법에 대한 자세한 내용은 *Using HP-UX VLAN(T1453-90001)* 설명서를 참조하십시오.

HP-UX VLAN 지원

HP-UX 11i의 Serviceguard A.11.14부터 VLAN이 지원됩니다.

VLAN 지원은 다른 링크 기술과 비슷합니다. VLAN 인터페이스는 클러스터의 데이터 네트워크뿐 아니라 하트비트로도 사용될 수 있습니다. 네트워크 관리자는 클러스터에 구성된 VLAN 인터페이스의 상태를 모니터링하고, 장애가 감지되면 VLAN 인터페이스에 대한 로컬 및 원격 장애 조치를 수행합니다. VLAN 인터페이스에 대한 장애 조치는 일반적으로 해당 인터페이스가 만들어진 물리 NIC 포트에 장애가 발생할 경우 수행됩니다.

구성 제한 사항

HP-UX에서는 물리 NIC 포트 하나당 최대 1024개의 VLAN을 만들 수 있습니다. 물론 이러한 구성을 제대로 사용하기 위해서는 많은 양의 시스템 리소스가 필요합니다. VLAN 기술을 사용할 수 있게 됨에 따라 각 클러스터 노드에 많은 수의 네트워크 인터페이스가 구성된 경우에는 Serviceguard에서 잠재적 성능 저하, CPU 사용량 증가 및 메모리 부족 문제가 발생할 수 있습니다. VLAN 네트워크에 충분한 유연성을 제공하려면 Serviceguard 솔루션은 다음의 VLAN 및 일반 네트워크 구성 요구 사항을 준수해야 합니다.

- 클러스터 ASCII 파일에서 노드당 최대 30개의 네트워크 인터페이스를 지원해야 합니다. 인터페이스는 물리 NIC 포트, VLAN 인터페이스, APA 집합 또는 이들의 조합이 될 수 있습니다.
- Serviceguard 하트비트 요구 사항을 충족하려면 VLAN 인터페이스가 만들어진 물리 LAN 인터페이스가 VLAN 인터페이스와 함께 ASCII 파일에 구성되어 있어야 합니다.
- 클러스터 노드에 VLAN과 함께 두 개 이상의 물리 NIC 포트가 구성된 경우 NIC 포트당 최대 14개의 VLAN 인터페이스가 지원되어야 합니다. 물리 NIC 포트가 VLAN과 함께 구성된 유일한 NIC 포트인 경우 이 제한은 28개의 VLAN 인터페이스까지 증가할 수 있습니다. 이러한 제한 사항은 노드당 30개의 네트워크 인터페이스로 제한됩니다. 이러한 구성을 보여 줄 수 있는 한 가지 예로 하트비트 네트워크로 구성된 두 개의 단일 포트 NIC 카드를 갖춘 클러스터 노드가 있습니다. 각 NIC 카드당 최대 14개의 VLAN 인터페이스가 있을 수 있으며 노드당 총 30개의 네트워크 인터페이스가 있을 수 있습니다.
- VLAN에 대한 로컬 장애 조치는 동일한 링크 유형에서 이루어져야 합니다. 주 VLAN과 대기 VLAN의 VLAN ID(또는 태그 ID)는 동일해야 합니다.
- VLAN 구성은 HP-UX 11i에서만 지원됩니다.

- 포트 기반 및 IP 서브넷 기반 VLAN만 지원됩니다. Serviceguard에서는 TCP/IP 이외의 전송 프로토콜을 지원하지 않으므로 프로토콜 기반 VLAN은 지원되지 않습니다.
- 각 VLAN 인터페이스는 주 VLAN 인터페이스의 대기 인터페이스로 사용되는 경우가 아니면 고유 서브넷에서 IP 주소를 할당 받아야 제대로 작동합니다.
- APA 집합 상의 VLAN 인터페이스는 지원되지 않습니다.
- 장애 조치를 위해 물리 LAN 인터페이스에서 VLAN 인터페이스로 이동하거나 그 반대로 이동하는 것은 VLAN 소프트웨어의 제약 때문에 지원되지 않습니다.
- WAN(Wide Area Network) 클러스터에서는 VLAN을 사용할 수 없습니다.

고급 하트비트 요구 사항

VLAN 기술은 엔터프라이즈 네트워크 구성에서 더 큰 유연성을 제공합니다.

Serviceguard가 이러한 동적 환경에서 안정성과 가용성을 동시에 유지하면서 성공적으로 실행되도록 하려면 클러스터에 VLAN 인터페이스가 있을 때 다음과 같이 기존 하트비트 규칙에 제한 사항을 더 추가하여 계속 적용해야 합니다.

1. 기존의 하트비트 최소 요구 사항은 변경되지 않지만, 단순 장애 요인을 없애기 위해 VLAN 인터페이스를 포함할 수 있는 모든 LAN 인터페이스 대신 물리 네트워크에 대해 이 요구 사항을 확인합니다.
2. VLAN을 비롯한 모든 네트워크에서 여전히 하트비트를 사용하는 것이 좋습니다.
3. VLAN이 하트비트 네트워크에 적합하지 않을 경우 클러스터 ASCII 파일에 있는 모든 나머지 물리 네트워크에 하트비트를 사용하는 것이 좋습니다.

데이터 저장 장치의 볼륨 관리자

볼륨 관리자는 저장 장치 그룹이라는 디스크 저장 장치 단위를 만드는 데 사용할 수 있는 도구입니다. 저장 장치 그룹에는 단일 시스템과 고가용성 클러스터에서 사용되는 논리 볼륨이 포함되어 있습니다. Serviceguard 클러스터에서 저장 장치 그룹은 패키지 제어 스크립트에 의해 활성화됩니다.

중복 저장 장치의 유형

Serviceguard에서는 두 가지 유형의 공유 데이터 저장 장치가 지원됩니다. 이러한 저장 장치는 하드웨어에 중복 저장 장치를 구성하는 외부 **디스크 어레이**와 미러링된 **개별 디스크(JBOD, "Just a Bunch of Disks"**라고도 함)입니다. 미러링을 **RAID1**이라고 부르고 디스크 어레이에서 사용 가능한 중복 장치는 **RAID5** 등으로 부릅니다. 두 저장 방법에는 다음과 같이 약간의 차이가 있습니다.

- JBOD를 사용하는 경우 저장 장치의 기본 요소는 개별 디스크입니다. 이 디스크는 미러(**RAID1**)를 만들기 위해 다른 디스크와 쌍을 이루어야 합니다. 일반적으로 Serviceguard 구성에서는 다른 저장 장치에 별도의 미러가 있습니다.
- 디스크 어레이를 사용하는 경우 저장 장치의 기본 요소는 이미 **RAID5**를 통해 중복 저장 장치를 제공하는 **LUN**입니다.

미러링된 저장 장치의 예

그림 3-20에서는 HA 저장 장치 랙을 사용하여 미러링된 저장 장치의 예를 보여 줍니다. 이 예에서 노드 1과 노드 2는 병렬 구성으로 연결되어 있으며 각 노드에는 두 개의 공유 저장 장치와 연결되는 중복 경로가 있습니다. 두 노드에는 각각 루트 파일 시스템, 스왑 등을 위해 사용되는 두 개의 공유되지 않는 내부 디스크도 있습니다. 각 공유 저장 장치에는 세 개의 디스크가 있습니다. 두 저장 장치에 있는 세 디스크의 장치 파일 이름은 각각 c0t0d0, c0t1d0, c0t2d0과 c1t0d0, c1t1d0, c1t2d0입니다.

그림 3-20 공유 저장 장치의 물리 디스크

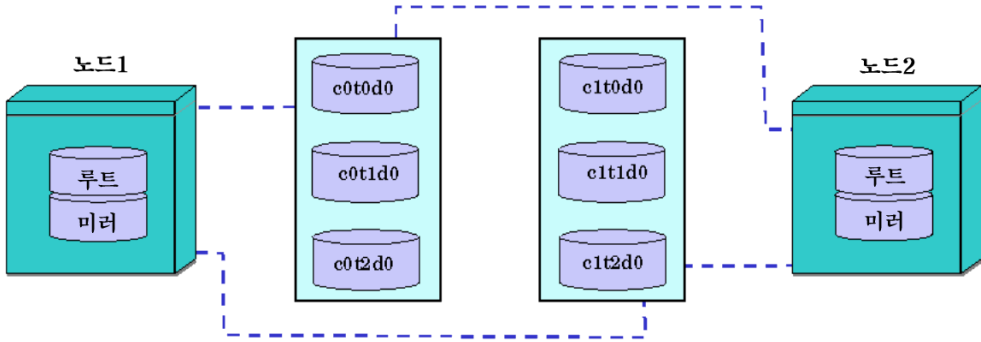


그림 3-21에서는 다중 디스크 미러링 구성에 결합된 개별 디스크를 보여 줍니다.

그림 3-21 미러링된 물리 디스크

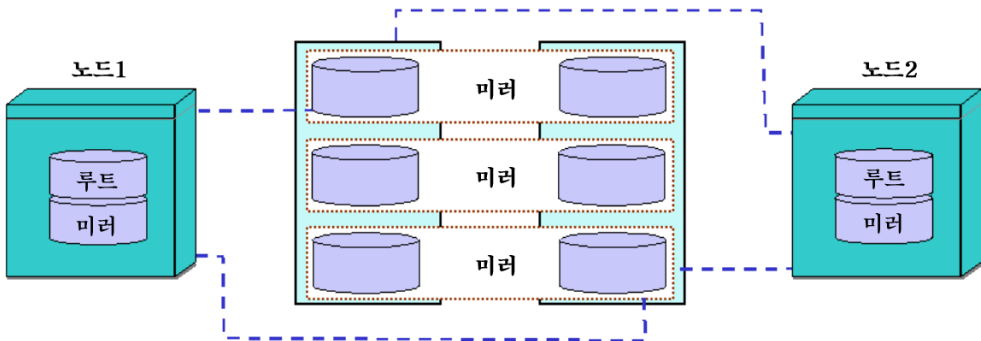
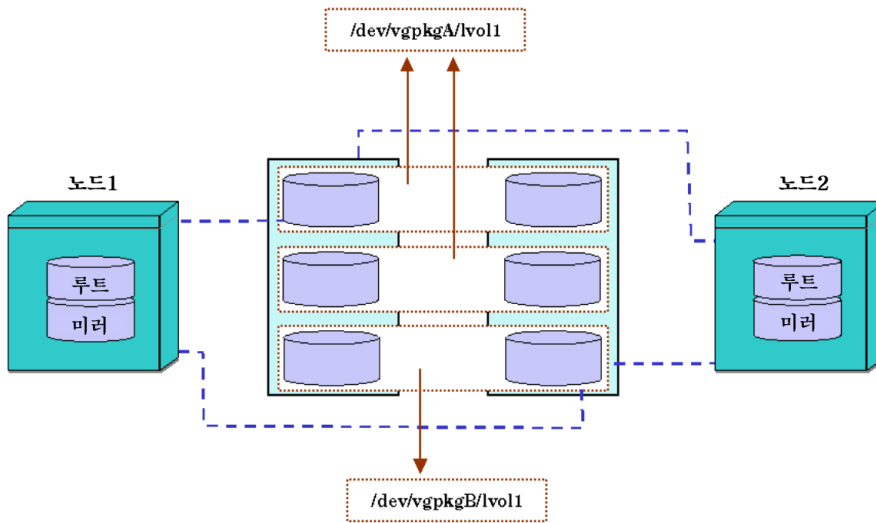


그림 3-22에서는 LVM 볼륨 그룹으로 구성된 미러인 `/dev/vgpkgA`와 `/dev/vgpkgB`를 보여 줍니다. 볼륨 그룹은 가용성이 높은 응용 프로그램에서 사용되도록 Serviceguard 패키지에 의해 활성화됩니다.

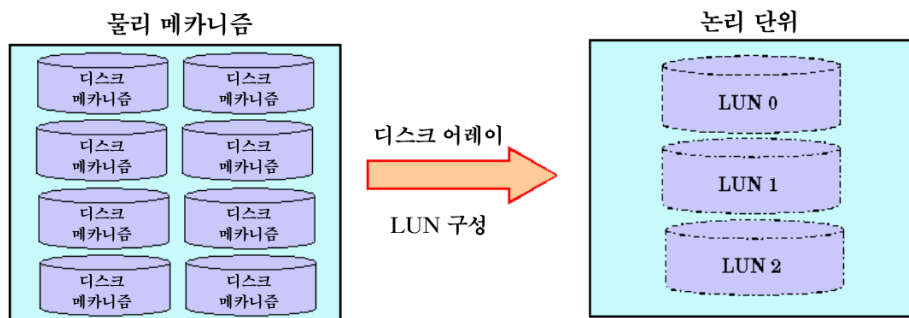
그림 3-22 볼륨 그룹에 구성된 여러 장치



디스크 어레이의 저장 장치 예

그림 3-23에서는 디스크 어레이에 구성된 저장 장치의 예를 보여 줍니다. 물리 디스크는 어레이 유틸리티 프로그램에 의해 논리 장치 또는 LUN으로 구성되어 운영 체제에서 인식할 수 있습니다.

그림 3-23 LUN에 결합된 물리 디스크

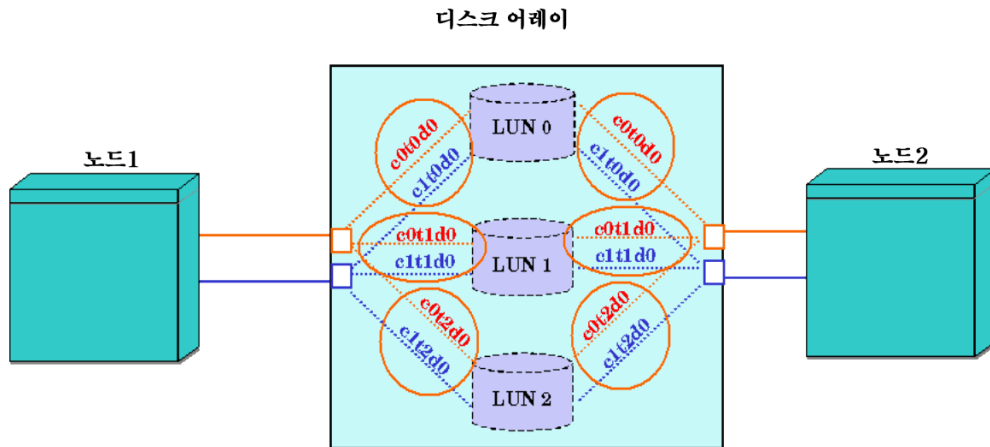


주

일반적으로 LUN은 디스크 어레이 제조업체에서 제공하는 유틸리티 프로그램을 사용하여 정의됩니다. 어레이는 매우 다양하므로 저장 장치와 함께 제공되는 설명서를 참조해야 합니다.

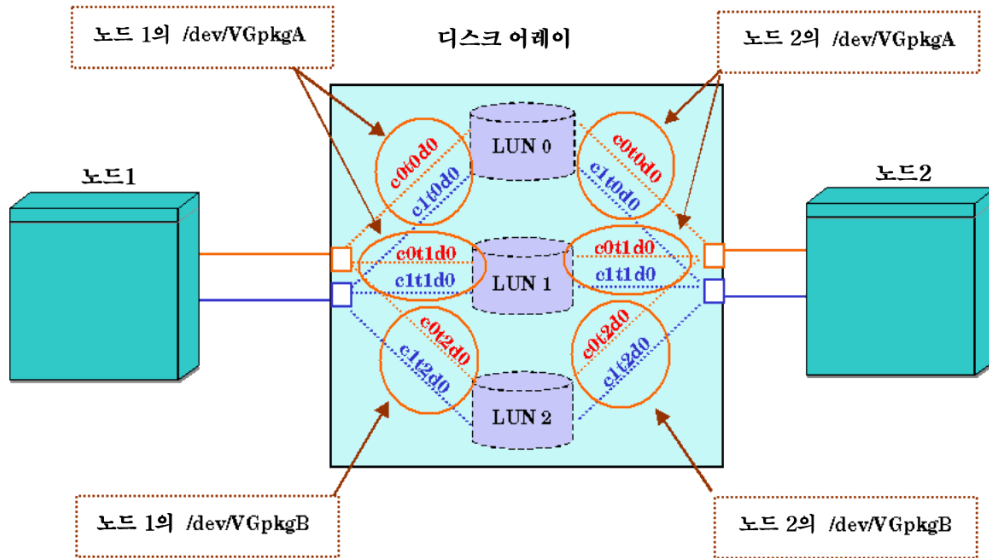
그림 3-24에서는 여러 경로(링크)로 구성되어 데이터에 대한 중복 경로를 제공하는 LUN을 보여 줍니다.

그림 3-24 LUN에 대한 다중 경로



마지막으로, 다중 경로는 그림 3-25에서와 같이 볼륨 그룹으로 구성됩니다.

그림 3-25 볼륨 그룹의 다중 경로



볼륨 관리자의 유형

Serviceguard에서는 데이터 저장 장치의 볼륨 관리자를 선택할 수 있습니다.

- HP-UX Logical Volume Manager(LVM) 및 MirrorDisk/UX(선택적)
- HP-UX용 VERITAS Volume Manager(VxVM) — 기본 및 추가 제품
- HP-UX용 VERITAS Cluster Volume Manager(CVM)

이러한 볼륨 관리자를 모두 사용하여 클러스터 저장 장치를 구성하는 방법에 대해서는 5장과 6장에 별도의 절로 설명합니다. 이 절의 나머지 부분에서는 사용 가능한 볼륨 관리자의 몇 가지 차이를 설명하고 사용 중인 클러스터 환경에 적절한 선택을 하는 데 유용한 지침을 제공합니다.

주

HP-UX Logical Volume Manager에 대한 내용은 **시스템 및 작업 그룹 관리**를 참조하십시오. VERITAS 볼륨 관리 제품에 대한 전체 설명은 *VERITAS Volume Manager for HP-UX Release Notes*를 참조하십시오.

HP-UX Logical Volume Manager(LVM)

Logical Volume Manager(LVM)는 HP-UX의 기존 저장 장치 관리 제품입니다. LVM은 운영 체제에 포함되며 모든 클러스터 노드에서 사용할 수 있습니다. LVM은 MirrorDisk/UX의 사용을 지원합니다. MirrorDisk/UX는 총 세 개의 데이터 사본에 대한 미러링을 두 개까지 갖는 디스크 미러링을 허용하는 추가 제품입니다.

현재 HP-UX 루트 디스크를 LVM 볼륨 그룹으로 구성할 수 있습니다. 이 경우 HP-UX 루트 디스크는 VERITAS 루트 디스크 그룹인 rootdg와 같지 않습니다. rootdg는 VERITAS Volume Manager 제품을 사용하는 모든 노드에서 HP-UX 루트 디스크 외에 추가로 구성되어야 합니다. Serviceguard 클러스터 잠금 디스크도 LVM 볼륨 그룹에서 구성된 디스크를 사용하여 구성됩니다.

LVM은 HP-UX 단일 시스템과 Serviceguard 클러스터에서 계속해서 지원됩니다.

VERITAS Volume Manager(VxVM)

HP-UX용 기본 VERITAS Volume Manager(Base-VXVM)는 추가 비용 없이 HP-UX 11i와 함께 제공됩니다. 여기에는 vmsa라는 Java 기반 GUI를 비롯한 기본 볼륨 관리자 기능이 포함되어 있습니다. Base-VXVM만으로도 Serviceguard의 클러스터 저장 장치를 구성할 수 있습니다. 그러나 제한된 기능 집합만 사용할 수 있습니다.

추가 제품인 HP-UX용 VERITAS Volume Manager(B9116AA)는 기본 볼륨 관리뿐 아니라 개선된 볼륨 관리자 기능을 모두 제공합니다. 여기에는 미러링, 활성 장치 및 활성 저장 장치에 대한 동적 다중 경로 지정, 자동 디스크 재배치(Hot Relocation) 등의 기능이 있습니다. HP-UX용 VERITAS FastResync 옵션(B9118AA)은 또 다른 추가 제품으로, 추가 비용을 지불하면 사용할 수 있습니다. 이 제품은 분할된 미러를 볼륨으로 다시 동기화하는 데 필요한 시간을 줄여 줍니다. 이 제품을 사용하려면 HP-UX용 VERITAS Volume Manager(B9116AA)를 구매해야 합니다.

다음과 같은 클러스터에서 VxVM을 사용할 수 있습니다.

- 노드가 16개 이하인 경우(크기는 상관 없음)
- 클러스터 시작 시간이 짧아야 하는 경우
- 공유 저장 장치 그룹 활성화가 필요하지 않은 경우
- 모든 노드가 모든 디스크에 연결되어 있지는 않은 경우
- RAID 5나 스트라이프 미러링을 사용해야 하는 경우
- 여러 개의 하트비트 서브넷이 구성되어 있는 경우

VxVM에서 디스크 그룹 전파

VxVM을 사용하면 클러스터 실행 여부에 관계없이 모든 노드에서 디스크 그룹을 만들 수 있습니다. 그런 다음 사용자는 각 노드에서 디스크 그룹 가져오기를 시도하여 디스크 그룹을 확인해야 합니다. 따라서 VxVM을 사용하면 CVM을 사용할 때보다 디스크 그룹 전파에 필요한 단계가 더 많아지지만 모든 노드에서 디스크 그룹을 만들 수 있다는 장점이 있습니다.

VxVM에서의 패키지 시작 시간

VxVM을 사용하면 디스크 그룹을 사용하는 패키지 제어 스크립트에서 각 디스크 그룹을 가져옵니다. 따라서 클러스터 시작 시간이 영향을 받지 않지만 VxVM은 패키지가 시작될 때 디스크 그룹을 가져오므로 개별 패키지 시작 시간이 늘어날 수 있습니다.

VERITAS Cluster Volume Manager(CVM)

VERITAS Volume Manager(VxVM) 대신에 VERITAS Cluster Volume Manager(CVM)를 사용하여 클러스터 저장 장치를 구성할 수도 있습니다. Base-VXVM은 Serviceguard가 설치될 때 기본 클러스터 기능을 일부 제공합니다. 노드는 네 개까지 지원되지만 소프트웨어 미러링, 활성 장치/활성 저장 장치에 대한 동적 다중 경로 지정 또는 추가 라이선스가 필요한 다른 많은 기능은 지원되지 않습니다. VERITAS Cluster Volume Manager(CVM, B9117AA)는 클러스터 사용을 위해 특별히 설계된 VxVM 볼륨 관리자의 향상된 버전입니다. VERITAS Volume Manager(B9116AA)와 함께 설치되는 경우 CVM 추가 제품은 클러스터 환경에서 개선된 VxVM 기능의 대부분을 제공합니다. CVM은 Serviceguard에서 클러스터 구성원에 대한 정보를 직접 얻어 클러스터를 정확하게 인식합니다. 클러스터 정보는 클러스터의 모든 노드에서 실행되는

VxVM-CVM-pkg라는 특별한 시스템 다중 노드 패키지를 통해 제공됩니다. CVM과 함께 사용할 수 있도록 VxVM 디스크 그룹을 구성하려면 클러스터가 작동되고 있어야 하고 이 패키지를 실행하고 있어야 합니다.

CVM을 사용하면 한 번에 한 노드에서 저장 장치를 활성화할 수 있으며 한 노드에서는 쓰기 활성화를 수행하고 다른 노드에서는 읽기 활성화를 동시에 수행할 수 있습니다(예: 백업 허용). CVM은 클러스터에 완전한 미러링과 DMP(동적 다중 경로 지정)를 제공합니다.

다음과 같은 클러스터에서 CVM을 사용할 수 있습니다.

- 패키지 장애 조치 후 빠른 디스크 그룹 활성화가 필요한 응용 프로그램을 실행할 경우
- 한 번에 두 노드 이상에서 활성화가 필요한 경우. 예를 들면, 한 노드에서 백업을 수행하는 동안 볼륨을 사용하는 패키지는 다른 노드에서 활성 상태여야 하는 경우입니다. 이 경우 디스크 그룹을 사용하는 패키지는 단독 쓰기 모드로 디스크 그룹을 활성화하고 백업을 수행하는 노드는 공유 읽기 모드로 디스크 그룹을 활성화하게 됩니다.
- 하나의 하트비트 서브넷으로만 구성된 경우

이번 릴리즈에서는 노드가 4개 이하일 때 CVM을 사용할 수 있습니다. 공유 저장 장치는 노드가 장치의 데이터에 액세스하는지 여부에 상관없이 클러스터의 모든 노드에 연결되어야 합니다.

CVM에서의 클러스터 시작 시간

CVM을 사용하면 VxVM-CVM-pkg 제어 스크립트가 완료될 때 모든 공유 DG(디스크 그룹)를 가져옵니다. DG 수, 노드 수 및 구성(디스크, 볼륨 등의 수)에 따라 가져오는 데 시간이 좀 걸릴 수 있습니다. 이 패키지의 현재 시간 종료 값은 3분이지만 규모가 더 큰 구성의 경우에는 이 값을 늘려야 할 수도 있습니다. CVM DG를 사용하는 장애 조치 패키지는 VxVM-CVM-pkg가 작동될 때까지 시작되지 않습니다. 이러한 지연은 패키지 장애 조치 시간에 영향을 주지 않습니다. 이것은 클러스터 시작 시 한 번 발생하는 오버헤드 비용입니다.

CVM에서 디스크 그룹 전파

CVM을 사용하면 CVM 마스터 노드라는 하나의 클러스터 노드에서 디스크 그룹이 만들어집니다. 공유 디스크 그룹을 만들려면 이 클러스터가 실행되고 있어야 합니다. CVM은 각 노드가 각 디스크를 인식할 수 있는지 확인하고 유효하지 않은 DG가 만들어지지 않도록 합니다.

CVM에서 필요한 단일 하트비트 서버넷

일반적으로는 클러스터 노드를 서로 연결하는 모든 서버넷을 하트비트 네트워크로 구성하는 것이 좋습니다. 이렇게 하면 추가 비용 없이 여러 장에 대한 보호 성능을 향상시킬 수 있습니다. 그러나 VERITAS Cluster Volume Manager(CVM)를 사용하는 경우에는 하트비트 서버넷을 하나만 사용할 수 있습니다. VxVM-CVM-pkg가 추가되면 (5장 참조) cmcheckconf 및 cmapplyconf가 하트비트 서버넷이 하나만 구성되었는지 확인합니다. 서버넷에 더 많은 하트비트가 구성되어 있으면 cmcheckconf 및 cmapplyconf에 실패하게 됩니다.

이러한 제한 사항은 VxVM-CVM-pkg에서 시작되는 VERITAS 클러스터 데몬 (vxclustd)의 설계 때문입니다. 이 데몬은 단일 서버넷의 단일 IP 연결에서만 클러스터 노드에 대한 정보를 교환할 수 있습니다. 또한 클러스터의 노드 간 통신은 Serviceguard 클러스터 데몬인 cmcld의 경우와 VERITAS 클러스터 데몬의 경우에서 동일해야 합니다. 따라서 cmcld도 단일 하트비트 서버넷의 사용만 허용됩니다.

단일 서버넷만 사용하는 데 따른 문제도 있습니다. NODE_TIMEOUT에 지정된 시간이 종료될 때까지 하트비트가 수신되지 않으면 클러스터는 재편성됩니다. 초과된 시간에 따라 하나 또는 모든 클러스터 노드에서 TOC가 발생할 수 있습니다. 이것은 하트비트 패킷이 전송되지 못할 정도로 서버넷에 트래픽이 많은 경우에 가장 일반적으로 볼 수 있는 현상입니다. 이 현상은 쉽게 복구되지 않지만 Serviceguard나 APA에서 감지할 수 없는 네트워크 구성 문제나 네트워크 하드웨어 문제가 있는 경우에도 볼 수 있습니다. 이러한 상황에서 문제를 방지하는 가장 일반적인 방법은 여러 개의 하트비트 네트워크를 사용하는 것입니다.

CVM의 경우 단일 하트비트 서버넷은 시스템이나 클러스터의 장애 가능성을 최소화할 수 있도록 대기 LAN과 함께 구성되거나 통합 포트 그룹으로 구성되어야 합니다.

볼륨 관리자 비교

다음 표는 현재 사용 가능한 볼륨 관리자의 장점과 단점을 요약한 것입니다.

표 3-5 Serviceguard에 있는 볼륨 관리자의 장점 및 단점

| 제품 | 장점 | 단점 |
|-----------------------------|--|---|
| Logical Volume Manager(LVM) | <ul style="list-style-type: none"> • 기존 시스템이 견고하고 HP-UX 사용자에게 잘 알려져 있음 • 기존 패키지를 변경할 필요 없음 • 클러스터당 최대 16개의 노드 지원 • 다중 데이터 경로에 PV 링크 사용 지원 • 다중 노드에서 읽기 전용 활성화뿐 아니라 단독 활성화도 지원 • 클러스터 잠금 디스크를 구성하는 데 사용할 수 있음 • 다중 하트비트 서브넷 지원 | <ul style="list-style-type: none"> • 다른 볼륨 관리자가 가진 유연성과 확장 기능이 없음 • PV 링크는 활성/대기 상태로만 있을 수 있으며 한 번에 하나의 링크만 활성화 |
| Base-VxVM | <ul style="list-style-type: none"> • HP-UX 11i 이상에서 소프트웨어 무료 • 간단한 기본 기능 제공 • 다중 하트비트 서브넷 지원 | <ul style="list-style-type: none"> • 제한된 기능 집합 • 클러스터 잠금 장치로 사용할 수 없음 |

표 3-5 Serviceguard에 있는 볼륨 관리자의 장점 및 단점(계속)

| 제품 | 장점 | 단점 |
|---|--|---|
| VERITAS Volume Manager — B9116AA(VxVM) | <ul style="list-style-type: none"> 모든 노드에서의 디스크 그룹 구성 소프트웨어 미러링, RAID 0/1, RAID 5 및 활성화 장치/활성 저장 장치에 대한 동적 다중 경로 지정 등의 고급 볼륨 관리 기능 집합 CVM에서보다 클러스터 시작 시간이 빠름 클러스터당 최대 16개의 노드 지원 다중 하트비트 서브넷 지원 | <ul style="list-style-type: none"> 공유 모드나 읽기 전용 모드에서는 다중 노드에서의 활성화를 지원하지 않음 vxdg 가져오기 시간이 많이 걸리므로 패키지 시작 시 지연될 수 있음 고급 기능은 추가 라이선스를 구입해야 함 클러스터 잠금 장치로 사용할 수 없음 |
| VERITAS Cluster Volume Manager — B9117AA(CVM) | <ul style="list-style-type: none"> 소프트웨어 미러링 및 RAID 0/1을 비롯한 고급 볼륨 관리 기능 집합 VxVM에서보다 패키지 시작 시간이 빠름 단독 활성화 지원 동시에 여러 노드에서 서로 다른 모드로 활성화 지원 | <ul style="list-style-type: none"> 디스크 그룹은 마스터 노드에서 구성되어야 함 최대 4개의 노드로 구성된 클러스터에서만 사용할 수 있음 VxVM에서보다 클러스터 시작 시 더 오래 걸릴 수 있음 추가 라이선스를 구입해야 함 스트라이프 미러나 RAID 5를 지원하지 않음 단일 하트비트 서브넷만 지원 |

장애에 대한 대응

Serviceguard는 장애의 종류에 따라 서로 다른 방식으로 대응합니다. 대부분의 하드웨어 장애에 대한 대응은 사용자가 구성할 수 없지만 패키지 및 서비스 장애인 경우에는 사용자가 시스템의 대응을 어느 정도 설정할 수 있습니다.

노드에 장애가 발생할 때의 TOC(제어권 전달)

Serviceguard 클러스터에서 가장 중요한 장애 대응 방식은 정상적인 종료 없이 SPU를 즉시 중지하는 HP-UX TOC(제어권 전달)입니다. 이 TOC는 데이터의 무결성을 보호하기 위해 수행됩니다.

클러스터 노드가 지정된 시간 동안 클러스터 구성원 과반수와 통신할 수 없거나, 커널 중지, 커널 스핀, 런어웨이 실시간 프로세스가 발생하거나, Serviceguard 클러스터 데몬인 `cmclld`에 장애가 발생한 경우에 TOC가 수행됩니다. TOC가 수행되는 동안 시스템 덤프가 수행되고 다음과 같은 메시지가 콘솔로 전송됩니다.

```
Serviceguard: Unable to maintain contact with cmclld daemon.
Performing TOC to ensure data integrity.
```

특정 상황에서는 Serviceguard가 TOC를 시작하기도 합니다. 패키지 구성 파일에서 서비스 빠른 장애 조치 매개 변수가 설정되어 있으면 해당 서비스에 장애가 발생할 때마다 TOC 수행과 함께 전체 노드의 작동이 중지됩니다. 패키지 구성 파일에서 `NODE_FAIL_FAST_ENABLED`가 YES로 설정되어 있으면 패키지 제어 스크립트를 0이나 1 이외의 값으로 종료시키는 장애가 발생하거나 시간이 종료될 때마다 전체 노드는 TOC를 수행하고 작동이 중단됩니다. 또한 패키지 및 서비스와 관계없이 이벤트에 의해 노드 수준 장애가 발생할 수 있습니다. 하트비트가 손실되거나 클러스터 데몬(`cmclld`) 또는 기타 중요한 데몬이 손실되면 패키지와 패키지 서비스가 정상적으로 작동되어도 노드의 작동은 중단됩니다.

드문 경우지만 TOC 전에 먼저 시스템이 다시 부팅되기도 합니다. 안전 타이머가 만료되기 전에 재부팅을 완료할 수 있으면 TOC는 수행되지 않습니다. 어떤 경우이든 패키지는 다른 노드로 신속하게 이동할 수 있습니다.

하드웨어 장애에 대한 대응

시스템이 중지되거나 SPU 회로가 물리적으로 손상되는 등의 심각한 문제가 발생하면 Serviceguard에서는 노드 장애를 인식하고 현재 해당 노드에서 실행 중인 패키지를 클러스터의 다른 대체 노드로 이동합니다. 각 패키지의 새 위치는 패키지의 기본 노드와 대체 노드가 설정되어 있는 패키지 구성 파일에 의해 결정됩니다. 다른 노드로 패키지를 이동할 때 프로그램 카운터는 이동되지 않습니다. 이동된 패키지에 있는 프로세스는 처음부터 다시 시작됩니다. 응용 프로그램이 장애 발생 후에 즉시 다시 시작되기 위해서는 “고장 극복” 능력을 갖고 있어야 합니다. 즉, 패키지에 있는 모든 프로세스가 이러한 재시작을 인식할 수 있도록 작성되어 있어야 합니다. 이것은 일반적인 시스템 고장 후 다시 시작해야 하는 응용 프로그램의 설계와 같습니다.

LAN 인터페이스 장애 시 대기 LAN 인터페이스가 있으면 대기 LAN 인터페이스로 로컬 전환이 수행됩니다. 하트비트 LAN 인터페이스에 장애가 발생한 경우 대기 하트비트나 중복 하트비트가 구성되어 있지 않으면 노드는 TOC를 수행하고 작동이 중단됩니다. 대기 인터페이스가 없는 상태에서 모니터링되는 데이터 LAN 인터페이스에서 장애가 발생하면 해당 패키지에 대해 NODE_FAILFAST_ENABLED(“계획” 장의 “패키지 구성 계획” 절 참조)가 YES로 설정된 경우에만 TOC 수행과 함께 노드 작동이 중단됩니다.

LVM의 MirrorDisk/UX나 VxVM 및 CVM의 VERITAS 미러링과 같은 별도의 제품으로 디스크를 보호할 수 있습니다. 또한 별도로 제공되는 EMS 디스크 모니터를 통해 잠금 디스크 장애 등과 같은 특정 장애가 발생할 때 운영자에게 알릴 수 있습니다. 자세한 내용은 *Using High Availability Monitors* 설명서(HP 제품 번호 B5736-90042)를 참조하십시오.

Serviceguard에서는 전원 장애에 직접적으로 대응하지 않지만 개별 클러스터 구성 요소의 전원에 장애가 발생할 경우 해당 구성 요소의 장애로 잘못 인식되어 적절한 전환 동작이 수행될 수 있습니다. HP에서 지원하는 HP PowerTrust 등의 UPS(무정전 전원 공급 장치)를 사용하여 전원 장애로부터 보호할 수 있습니다.

패키지 및 서비스 장애에 대한 대응

기본적으로 패키지나 패키지 서비스에서 장애가 발생하면 ‘stop’ 매개 변수를 사용한 제어 스크립트가 실행되어 패키지를 중지한 다음 대체 노드에서 패키지가 다시 시작됩니다. 구성된 리소스 충족성이 충족되지 않았음을 나타내는 EMS 모니터 이벤트 보고서를 패키지 관리자가 받으면 패키지의 작동이 중단되고 대체 노드에서 재시작이 시도됩니다.

원하는 경우 전송 전에 노드가 비정상 종료(TOC)되도록 지정하여 이 기본 동작을 수정할 수 있습니다. 드문 경우이지만 **Serviceguard**에서는 이 동작이 지정되면 **TOC** 전에 시스템을 다시 부팅하려고 합니다. 버퍼 캐시에 있는 버퍼를 플러시할 수 있을 만큼 시간이 충분하면 성공적으로 다시 부팅되고 **TOC**는 수행되지 않습니다. 어떤 경우이든 미리 정의된 시간 안에 시스템의 작동이 중단됩니다.

패키지 종료가 중단되고 노드가 알 수 없는 상태에 빠지는 경우 빠른 장애 조치 옵션을 사용하면 신속하게 장애 조치를 할 수 있습니다. 그러면 노드가 다시 부팅되어 깨끗한 상태가 됩니다. 그러나 노드에서 고장이 발생하면 노드의 **모든** 패키지가 갑자기 중지됩니다.

패키지 구성 중에 노드 및 서비스 빠른 장애 조치 매개 변수를 설정하면 장애가 발생할 때의 패키지 및 노드 동작이 정확하게 결정됩니다. 적절한 장애 조치 동작을 선택하는 방법에 대해서는 “계획” 장의 “패키지 구성 매개 변수” 절을 참조하십시오.

서비스 재시작

장애가 발생한 후 서비스가 다시 로컬로 시작되도록 할 수 있습니다. 이렇게 하려면 패키지 제어 스크립트에서 각 서비스의 재시작 횟수를 지정해야 합니다. 서비스가 시작되면 서비스 환경에서 **RESTART_COUNT** 변수가 설정됩니다. 서비스는 실행될 때 이 변수를 조사하여 장애 후에 다시 시작되었는지 여부를 확인할 수 있으며, 장애 후 다시 시작된 경우에는 정리와 같은 적절한 작업을 수행할 수 있습니다.

네트워크 통신 장애

클러스터의 중요한 요소 중 하나는 네트워크 자체의 상태입니다. 각 노드는 계속 클러스터를 모니터링하면서 다른 노드의 하트비트 메시지를 수신하여 모든 노드가 서로 통신할 수 있는지를 확인합니다. 지정된 시간 동안 노드가 하트비트 메시지를 받지 못하면 노드 시간 종료 발생하고 클러스터 재편성이 일어나며, 이후에도 계속해서 하트비트 메시지를 받지 못하면 **TOC**가 수행됩니다. 두 개의 노드로 구성된 클러스터에서 **RS-232** 회선을 사용하면 네트워크 포화에 따른 **LAN**의 일시적 하트비트 손실로 **TOC**가 발생하는 것을 방지할 수 있습니다. **RS232** 회선은 네트워크 장애가 발생했을 때 이를 신속하게 감지하는 데도 도움이 됩니다.

Serviceguard 소프트웨어 구성 요소 이해
장애에 대한 대응

4 HA 클러스터 계획 및 문서화

Serviceguard 클러스터를 만드는 작업은 클러스터 구성의 모든 하드웨어 및 소프트웨어 구성 요소에 대한 정보를 수집하여 기록하는 계획 단계부터 시작합니다. 계획은 하드웨어 및 네트워크 구성 요소의 간단한 목록을 만드는 것으로 시작합니다. 설치와 구성이 계속 진행될수록 목록이 확장되고 내용도 점점 구체적으로 됩니다. 하드웨어를 설치한 다음에는 그래픽 사용자 인터페이스인 **Serviceguard Manager**나 다양한 **HP-UX** 명령을 사용하여 현재 구성에 대한 정보를 얻을 수 있습니다. 이 정보는 이 장에서 제공하는 워크시트에 기록됩니다. 클러스터를 만드는 과정에서 이 계획 워크시트를 참고하여 **Serviceguard Manager**에 값을 입력하거나 ASCII 구성 파일과 제어 스크립트의 값을 편집합니다.

Serviceguard 버전 11.16에서는 **SAM** 고가용성 도구를 더 이상 사용할 수 없습니다.

Serviceguard 버전 11.16 이상에서는 클러스터를 구성할 때 버전 **A.04.00** 이상의 **Serviceguard Manager**를 사용합니다. 또한 **Serviceguard** 버전 11.12 이상에서 클러스터를 관리할 때는 버전 11.12 이상의 **Serviceguard Manager**를 사용합니다.

Serviceguard Manager에 대한 자세한 내용은 7장을 참조하십시오. 이전 버전에서 **SAM**을 사용하는 방법에 대한 자세한 내용은 이 설명서의 이전 버전을 참조하십시오.

구성을 마친 후에는 **Serviceguard Manager**를 사용하여 모든 구성 데이터를 **a.sgm** 파일에 저장하고, 나중에 **Serviceguard Manager**에서 이 정보를 볼 수 있습니다.

Serviceguard Manager에서 구성된 클러스터와 패키지는 **Serviceguard Manager**에서 수정할 수 있습니다. 7장의 “**Serviceguard Manager** 사용” 절을 참조하십시오.

이 장의 내용은 다음 사항을 계획하는 데 유용합니다.

- 일반 계획
- 하드웨어 계획
- 전원 공급 계획
- 퀘럼 서버 계획
- LVM 계획

- CVM 및 VxVM 계획
- 클러스터 구성 계획
- 패키지 구성 계획

이 장에서 설명하는 각 계획 단계에서는 성공적인 설치 및 유지 관리를 위해 매개 변수와 기타 관련 데이터를 선택적으로 기록할 수 있는 워크시트를 사용할 수 있습니다. 각 단계를 진행하면서 중요한 구성 세부 사항을 모두 기록하여 운영 시스템을 문서화하십시오. 클러스터를 실제로 구성할 때 이 워크시트의 정보를 참조하십시오. 전체 워크시트 양식은 부록 G에 수록되어 있습니다.

주

계획 및 설치 과정은 상당 부분이 중복되므로 실제로 구성을 진행하기 전까지는 워크시트의 내용을 완성하기가 힘듭니다. 워크시트를 완성하지 못한 경우에는 구성을 진행하면서 누락된 요소를 채워 시스템을 문서화하십시오.

이후 장에서는 구성과 유지 관리 작업에 대해 자세히 설명합니다.

일반 계획

고가용성 목표를 분명히 이해하고 있으면 하드웨어 요구 사항을 정의하고 시스템을 설계하는 데 많은 도움이 됩니다. 다음 질문을 일반 계획의 지침으로 사용하십시오.

1. 장애가 발생해도 계속해서 사용할 수 있어야 하는 응용 프로그램은 무엇입니까?
2. 이 응용 프로그램을 지원하는 데 필요한 시스템 리소스(처리 능력, 네트워크, SPU, 메모리, 디스크 공간)는 무엇입니까?
3. 정상 작동 상태에서 이 리소스는 클러스터의 노드 간에 어떤 식으로 분산됩니까?
4. 발생할 수 있는 모든 장애 상황에서, 특히 노드에 장애가 발생할 때 이 리소스는 클러스터의 노드 간에 어떤 식으로 분산됩니까?
5. 클러스터의 일상적인 유지 관리 과정에서는 리소스가 어떻게 분산됩니까?
6. 네트워크 요구 사항은 무엇입니까? 모든 네트워크와 서버넷을 사용할 수 있습니까?
7. 다음과 같은 단순 장애 요인(Single Point of Failure)을 모두 제거했습니까?
 - 네트워크 장애 요인
 - 디스크 장애 요인
 - 전원 장애 요인
 - 응용 프로그램 장애 요인

Serviceguard 메모리 요구 사항

Serviceguard에서 필요로 하는 잠금 가능한 메모리의 총량은 클러스터에 구성된 패키지 수에 따라 달라집니다. 다음은 필요한 메모리의 대략적인 양을 계산하기 위한 공식입니다.

$$\text{총 메모리} = 6\text{MB} + 100\text{KB} / \text{클러스터 내 패키지 수}$$

이 총 메모리 양은 주어진 패키지가 해당 노드에 있는지 여부와 관계없이 클러스터의 모든 노드에 필요합니다.

확장을 고려한 계획

클러스터를 처음 구성할 때는 노드 집합을 선택하고 초기 구성에 사용할 패키지 그룹을 정의하게 됩니다. 나중에 필요에 따라 노드와 패키지를 추가하거나 공유 데이터를 저장하는 데 사용할 디스크 하드웨어를 추가할 수 있습니다. **클러스터를 실행 중지하지 않고도** 확장할 수 있게 하려면 다음 지침에 따라 초기 구성을 주의 깊게 계획해야 합니다.

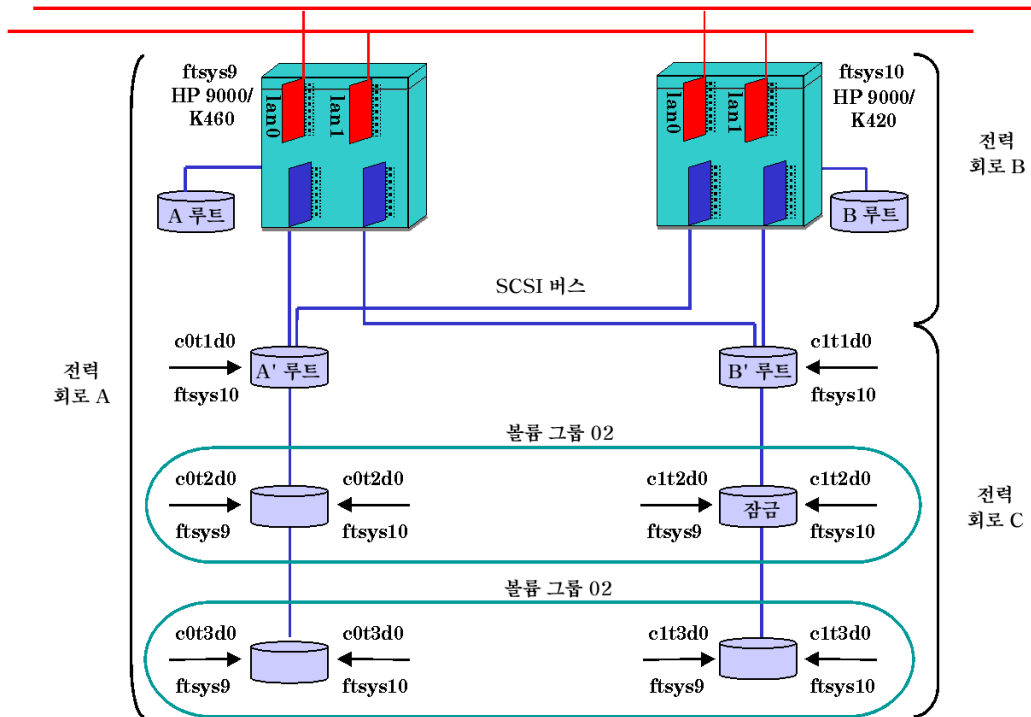
- **Maximum Configured Packages** 매개 변수(이 장 뒷부분의 “클러스터 구성 계획” 절 참조)를 설정할 때는 나중에 패키지를 추가할 수 있도록 값을 충분히 높게 설정합니다. 패키지 용량을 늘릴수록 더 많은 메모리 리소스(패키지당 100KB)가 사용 됩니다.
- **SCSI** 버스 케이블 연결 방법을 계획할 때는 필요할 경우 공유 데이터 저장용으로 디스크 하드웨어를 추가할 수 있도록 합니다. 추가 장치용으로 사용하려는 **SCSI** 포트에는 인라인 **SCSI** 단자 케이블을 연결해야 합니다. 이렇게 하면 버스가 활성화된 상태에서 장치를 추가할 수 있습니다.
- 확장을 고려할 때는 클러스터 잠금 규칙에 주의해야 합니다. 노드가 하나인 클러스터에는 클러스터 잠금이 필요하지 않지만 노드가 두 개인 클러스터에는 클러스터 잠금이 필요합니다. 노드가 네 개 이상인 클러스터에서는 클러스터 잠금을 사용하는 것이 매우 좋습니다. 노드가 다섯 개 이상인 클러스터의 경우에는 쿼럼 서버를 사용할 수 있지만 클러스터 잠금 디스크는 사용할 수 없습니다.
- 나중에 클러스터 실행 상태에서 패키지를 추가할 때 필요하게 될 네트워크는 미리 클러스터 구성에 포함시켜야 합니다.
- **EMS**에서 모니터링되는 리소스로서, 나중에 클러스터 실행 상태에서 패키지를 추가할 때 필요하게 될 리소스는 미리 클러스터 구성에 포함시켜야 합니다. 클러스터의 모든 패키지에 대해 이미 구성된 리소스 의존성은 나중에 추가되는 패키지에서도 항상 사용할 수 있습니다. 그러나 한 번도 구성된 적이 없는 리소스는 패키지 실행 상태에서 패키지에 추가할 수 없습니다.

클러스터 실행 상태에서 동적으로 클러스터 구성을 변경하는 방법에 대한 자세한 내용은 “클러스터 및 패키지 유지 관리” 장을 참조하십시오.

하드웨어 계획

하드웨어 계획 단계에서는 실제 하드웨어를 검사해야 합니다. 하드웨어 구성을 다이어그램으로 그려 어댑터 카드, 버스, 케이블 연결, 디스크 및 주변 장치를 표시하면 유용합니다. 그림 4-1은 두 개의 노드로 구성된 클러스터의 다이어그램 예입니다.

그림 4-1 클러스터 구성 예



계획하고 있는 클러스터를 위와 같이 그려 보고 그 내용을 하드웨어 워크시트에 기록하십시오. 각 장치 어댑터를 설치할 슬롯을 지정하고 각 어댑터의 버스 주소를 지정하십시오. 그리고 클러스터를 구성하면서 이 내용을 세부적으로 업데이트하십시오(5장 참조). **SPU마다 하나의 양식을 사용하십시오.** 양식에는 다음 세 부분이 포함되어 있습니다.

- SPU 정보

- 네트워크 정보
- 디스크 입출력 정보

SPU 정보

SPU 정보에는 클러스터에서 사용하는 시스템의 기본 특성이 포함됩니다. 여러 가지 모델의 컴퓨터를 동일한 클러스터에서 함께 사용할 수 있습니다. 이 구성 모델은 **HP Integrity** 서버에도 적용됩니다. **700** 시리즈 워크스테이션은 **Serviceguard**에서 지원되지 않습니다.

노드마다 하나의 워크시트를 사용하여 다음 항목을 기록하십시오.

Server Number

시리즈 번호를 입력합니다(예: rp8400 또는 rx8620-32).

Host Name

시스템에서 호스트 이름으로 사용할 이름을 입력합니다.

Memory Capacity

메모리를 MB 단위로 입력합니다.

Number of I/O slots

슬롯 수를 지정합니다.

네트워크 정보

Serviceguard에서는 LAN 인터페이스뿐만 아니라 클러스터 하트비트 전송 전용으로 구성된 직렬(**RS232**) 회선도 모니터링합니다.

주

Serviceguard에서는 같은 클러스터의 노드 간에 라우터를 통한 통신을 지원하지 않습니다.

Serviceguard 통신은 클러스터 노드 간에 이루어지는 데이터 링크 계층의 **DLPI(Data Link Provider Interface)** 트래픽과 전송 계층의 **UDP/TCP(User Datagram Protocol/Transmission Control Protocol)** 트래픽의 교환에 의존합니다.

LAN 정보

서브넷마다 LAN 인터페이스가 적어도 하나는 필요하지만 네트워크의 단순 장애 요인을 없애기 위해서는 둘 이상의 LAN 인터페이스가 필요합니다. 이 중 하나는 주 LAN 인터페이스이고 나머지는 대기 LAN 인터페이스여야 합니다.

클라이언트 데이터에 사용될 하트비트를 포함해서 모든 서브넷에 하트비트를 구성하는 것이 좋습니다. 워크시트에서 각 LAN 인터페이스에 대해 다음을 입력합니다.

Subnet Name

서브넷의 IP 주소 마스크를 입력합니다. 하트비트 IP 주소는 노드마다 같은 서브넷에 있어야 합니다.

Interface Name

이 노드에서 서브넷에 액세스하는 데 사용될 LAN 카드의 이름을 입력합니다. 카드를 설치한 후에 lanscan을 사용하면 이 이름이 표시됩니다.

IP Address

해당 인터페이스에 사용될 해당 노드의 호스트 IP 주소를 입력합니다. IP 주소가 없는 대기 인터페이스의 경우에는 ‘Standby’라고 입력합니다.

IPv4 주소는 소수점으로 구분된 네 자리 문자열이며 형식은 다음과 같습니다.

nnn.nnn.nnn.nnn

IPv6 주소는 콜론으로 구분된 여덟 자리 16진수 값 문자열이며 형식은 다음과 같습니다.

xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx.

IPv6 주소 형식에 대한 자세한 내용은 부록, “IPv6 주소 유형”을 참조하십시오.

NETWORK_FAILURE_DETECTION

주 네트워크 카드와 대기 네트워크 카드가 있는 경우

Serviceguard에서는 카드에 장애가 발생했을 때 이를 판별하고 장애 조치를 위해 트래픽을 다른 카드로 전달할지 여부를 알아야 합니다. Serviceguard 네트워크 관리자는 장애를 감지하기 위해 들어오는 트래픽과 나가는 트래픽을 모두 모니터링하고, 일정 시

간 동안 네트워크 트래픽이 없으면 해당 카드를 중지됨 상태로 표시한 후 장애 조치를 시도하기 시작합니다. **Serviceguard**에서 시간을 계산하는 방식은 LAN 카드의 종류에 따라 다릅니다.

구성 파일에서는 네트워크 인터페이스 카드에 장애가 발생한 경우 이를 확인하기 위한 방법을 다음 중 하나로 지정합니다.

- INOUT - 기본적으로 사용되는 방법으로, 폴링으로 보낸 패킷 수를 계산하고 들어오는 패킷 수와 나가는 패킷 수가 **모두** 증가하지 않을 때만 해당 카드를 작동 중지된 것으로 선언합니다.
- INONLY_OR_INOUT - 이 옵션에는 INOUT 방법이 포함됩니다. 그러나 이 옵션은 들어오는 패킷 수만 증가하지 않는 경우에도 카드가 작동 중지된 것으로 선언합니다. 이 방법을 사용할 경우 **Serviceguard**에서는 추가 원격 폴링을 수행하여 수신 장애 보고서의 유효성을 확인합니다. 추가 원격 폴링은 **Serviceguard** 버전 A.11.16에 새로 추가된 기능입니다.

기본값은 INOUT입니다.

새 옵션이 적합한지 여부는 주로 네트워크 구성에 따라 달라집니다. 자세한 내용은 97페이지의 “LAN 인터페이스 모니터링 및 장애 감지”를 참조하십시오. [http://docs.hp.com/hpux/ha->Serviceguard의 White Papers](http://docs.hp.com/hpux/ha->Serviceguard의WhitePapers) 절에 있는 “Inbound Failure Detection Enhancement”도 참조하십시오.

Kind of LAN Traffic

서브넷의 용도를 다음 중 하나로 지정합니다.

- 하트비트
- 클라이언트 트래픽
- 대기

목록에 레이블을 붙여 브리지 연결 네트워크에 속하는 서브넷을 표시하십시오.

지금까지 설명한 워크시트 정보는 클러스터 관리자 및 패키지 관리자가 수행하는 구성 단계에서 서브넷 그룹을 만들고 IP 주소를 지정하기 위해 사용됩니다.

RS232 정보

직렬(RS232) 회선을 구성하려면 각 노드의 직렬 포트에 대응되는 직렬 장치 파일을 결정해야 합니다.

1. MUX 패널을 사용할 경우, MUX에 대응되는 시스템 슬롯 번호를 적고 패널에서 선택된 포트 다음에 나타나는 포트 번호도 적어 둡니다.
2. 각 노드에 대해 `ioscan -fnC tty`를 사용하여 하드웨어 주소 및 장치 파일 이름을 표시합니다. 예를 들면 다음과 같습니다.

```
# ioscan -fnC tty
```

이렇게 하면 특정 노드의 각 RS232 장치와 연관된 장치 파일이 모두 나열됩니다.

3. 장치 파일을 확인한 후에는 다음과 같이 연결을 확인합니다. 이때 노드 1은 `/dev/tty0p0`을 사용하고 노드 2는 `/dev/tty0p6`을 사용한다고 가정합니다.

- 노드 1의 터미널에서 다음 명령을 실행합니다.

```
# cat < /dev/tty0p0
```

- 노드 2의 터미널에서 다음 명령을 실행합니다.

```
# cat /etc/passwd > /dev/tty0p6
```

암호 파일의 내용이 노드 1의 터미널에 표시되어야 합니다.

4. 워크시트에 다음을 입력합니다.

RS232 Device File

이 노드에 사용될 직렬 인터페이스에 대응되는 장치 파일 이름을 입력합니다. ASCII 구성 파일에서 이 매개 변수는 `SERIAL_DEVICE_FILE`입니다.

Second Node Name

RS232를 통해 이 노드에 연결될 노드의 이름을 입력하십시오. 노드 이름은 31바이트까지 지정할 수 있습니다.

예상되는 최대 클러스터 크기에 맞는 SCSI 주소 설정

SCSI 표준에서는 SCSI 주소에 따라 우선 순위를 정의합니다. 컨트롤러가 SPU에서 고갈되지 않도록 하려면 SCSI 인터페이스 카드를 최고 우선 순위로 구성해야 합니다. 따라서 가용성이 높은 클러스터를 구성할 때는 노드에 최고 우선 순위의 SCSI 주소를 부여하고 디스크에는 보다 낮은 우선 순위의 주소를 부여해야 합니다.

SCSI의 경우 높은 우선 순위는 7에서 시작하여 0까지 낮아진 다음 다시 15에서 8까지 분포됩니다. 따라서 최고 우선 순위는 7이고 최저 우선 순위는 8입니다. 예를 들어, 클러스터에 최대 네 개의 노드가 있고 네 대의 시스템 모두 디스크 문자열을 공유한다면 SCSI 주소는 네 시스템 모두의 인터페이스 카드에서 고유하게 설정되고 우선 순위가 높은 주소여야 합니다. 따라서 시스템과 디스크의 주소는 다음과 같이 지정해야 합니다.

표 4-1 클러스터 구성에서 SCSI 주소 지정

| 시스템 또는 디스크 | 호스트 인터페이스 SCSI 주소 |
|------------|-------------------|
| 기본 시스템 A | 7 |
| 기본 시스템 B | 6 |
| 기본 시스템 C | 5 |
| 기본 시스템 D | 4 |
| 디스크 #1 | 3 |
| 디스크 #2 | 2 |
| 디스크 #3 | 1 |
| 디스크 #4 | 0 |
| 디스크 #5 | 15 |
| 디스크 #6 | 14 |
| 기타 | 13 - 8 |

주

공유 SCSI 버스에서 부트/루트 디스크를 낮은 우선 순위의 주소로 구성하면 부트/루트 장치에 액세스할 때 시간이 종료되어 시스템 장애가 발생할 수 있습니다. 이러한 현상은 같은 버스에 노드와 디스크가 많이 구성된 클러스터에서 발생할 수 있습니다.

SCSI 주소를 할당하는 올바른 방법은 클러스터 노드의 인터페이스 카드에 가장 높은 우선 순위의 SCSI 주소를 할당하고, 그 다음 공유 버스의 부트/루트 디스크, 공유 버스의 나머지 디스크 순서로 높은 우선 순위의 SCSI 주소를 할당하는 것입니다.

디스크 입출력 정보

워크시트의 이 부분에서는 디스크 장치 어댑터가 설치되는 위치를 지정할 수 있습니다. 노드의 각 디스크 장치 어댑터에 연결된 각 디스크에 대해 워크시트에 다음 항목을 입력합니다.

| | |
|-------------------------|--|
| <i>Bus Type</i> | 버스의 종류를 지정합니다. 지원되는 버스는 파이버 채널, 단일 중단 SCSI, F/W SCSI, Ultra2 및 Ultra160 SCSI입니다. |
| <i>Slot Number</i> | 컴퓨터 뒷면에서 인터페이스 카드가 삽입된 슬롯 번호를 지정합니다. |
| <i>Address</i> | 나중에 <code>ioscan</code> 을 사용하여 하드웨어를 표시할 때 시스템에 표시될 버스 하드웨어 경로 번호를 입력합니다. |
| <i>Disk Device File</i> | 디스크 장치 파일 이름을 입력합니다. 장치 파일 이름을 표시하려면 <code>ioscan -fnC disk</code> 명령을 사용합니다. |

이 부분의 워크시트 정보는 논리 볼륨 관리자를 사용하여 미러링된 디스크 구성을 만들 때 사용됩니다. 또한 디스크 구성에 대한 정보를 가능한 한 많이 수집하는 것이 유용합니다. 사용 가능한 디스크에 대한 정보는 다음 명령을 사용하여 얻을 수 있습니다.

- `diskinfo`
- `ioscan -fnC disk`
- `lssfs /dev/*dsk/c*`
- `bdf`
- `mount`
- `swapinfo`

- vgdisplay -v
- lvdisplay -v
- lvlnboot -v
- vxdg list(VxVM 및 CVM)
- vxprint(VxVM 및 CVM)

이러한 명령은 표준 HP-UX 명령입니다. 각 명령의 용도에 대해서는 해당 맨페이지를 참조하십시오. 하드웨어를 설치하고 시스템을 다시 부팅한 후 **모든 노드**에서 이 명령을 실행해야 합니다. 이 정보는 저장 장치 그룹 및 클러스터 구성을 수행할 때 유용합니다. lssf 명령의 출력을 인쇄하고, 인쇄된 내용을 이용하여 디스크를 할당할 물리 볼륨 그룹을 표시할 수 있습니다.

하드웨어 구성 워크시트

다음 워크시트는 클러스터 하드웨어 구성을 정리하고 기록하는 데 유용합니다. 이 워크시트를 필요한 만큼 복사하여 사용하십시오. 워크시트를 다 채운 후에는 나중에 참조할 수 있도록 잘 보관하십시오.

SPU Information:

Host Name ____ftsys9_____ Series No _____rp8400_____

Memory Capacity ____128 MB _____ Number of I/O Slots ____12_____

LAN Information:

| Name of Subnet | Name of Interface | Node IP Addr | Traffic Type |
|----------------|-------------------|-----------------|--------------|
| __Blue__ | __lan0__ | __35.12.16.10__ | __HB__ |

| Name of Subnet | Name of Interface | Node IP Addr | Traffic Type |
|----------------|-------------------|--------------|--------------|
| __Blue__ | __lan2__ | _____ | __standby__ |

| Name of Subnet | Name of Interface | Node IP Addr | Traffic Type |
|----------------|-------------------|-----------------|----------------|
| __Red__ | __lan1__ | __35.12.15.12__ | __HB, client__ |

Network Failure Dections: ____INOUT__

Serial (RS232) Heartbeat Interface Information:

RS232 Device File ____/dev/tty0p0__

Second Node Name ____ftsys10_____

=====

Disk I/O Information for Shared Disks:

Bus Type _SCSI_ Slot Number _4_ Address _16_ Disk Device File __c0t1d0_

Bus Type _SCSI_ Slot Number _6_ Address _24_ Disk Device File __c0t2d0_

Bus Type _____ Slot Number ____ Address _____ Disk Device File _____

Attach a printout of the output from the `ioscan -fnC disk` command after installing disk hardware and rebooting the system. Mark this printout to indicate which physical volume group each disk belongs to.

전원 공급 계획

클러스터를 설계할 때는 전선 전원과 UPS(무정전 전원)라는 두 가지 전원을 고려해야 합니다. 전원 회로에 문제가 생겨도 클러스터의 작동이 중단되지 않아야 합니다.

서버, 대용량 저장 장치 및 기타 하드웨어는 두세 개의 전원 공급 장치를 별도로 포함하고 있는 경우가 많으므로 하나 이상의 전원 공급 장치나 전원 회로에서 전원 문제가 발생해도 작동이 중단되지 않습니다. 클러스터에 중복 전원 공급 장치를 갖춘 장치가 포함되어 있는 경우에는 각 장치의 전원 공급 장치를 별도의 전원 회로에 연결합니다. 이렇게 하면 단일 전원 회로에 장애가 발생해도 클러스터에 포함된 주요 장치에 전체적인 장애가 발생하지는 않게 됩니다. 예를 들어, 클러스터의 모든 장치에 세 개의 전원 공급 장치가 있는 경우 전원이 클러스터의 단순 장애 요인이 되지 않도록 하려면 최소한 세 개의 별도 전원 회로가 필요합니다.

전원 공급 장치를 하나만 사용하는 서비스의 경우에는 노드의 절반 이상이 하나의 전원 공급 장치에 연결되어서는 안 됩니다. 노드의 정확히 절반이 하나의 전원 공급 장치로부터 전원을 공급 받는 경우, 클러스터 잠금 디스크 또는 쿼럼 서버는 다른 전원 공급 장치로부터 전원을 공급 받도록 해야 합니다. 그렇지 않으면 장애가 발생한 후에 클러스터를 재편성할 수 없게 됩니다. 자세한 내용은 “클러스터 구성 계획”의 클러스터 잠금 절을 참조하십시오.

전원 장애가 발생했을 때 가용성을 높이려면 적어도 각 노드의 SPU와 클러스터 잠금 디스크에는 별도의 UPS를 사용해야 합니다. 쿼럼 서버 노드에서는 해당 노드가 서비스를 제공하는 모든 클러스터와는 다른 전원을 사용해야 합니다. 별도의 물리 볼륨 그룹의 구성원으로 구성될 클러스터 잠금 디스크나 쿼럼 서버는 각각 다른 전원 공급 장치에 연결해야 합니다. 마지막 규칙은 서로 다른 전원 공급 장치에 연결되어 서로 다른 입출력 버스를 사용하는 물리 디스크 사이에서 디스크 미러링이 수행되도록 하기 위한 것입니다.

혼동을 피하기 위해 각 하드웨어 장치와 전원 공급 장치에 서로 다른 장치 번호로 레이블을 붙여 구분하는 것이 좋습니다. 전원 공급 워크시트에서 사용할 특정 하드웨어 장치와 해당 장치에 연결할 전원 공급 장치를 지정하십시오. 워크시트에 다음 레이블 정보를 입력하십시오.

| | |
|------------------|-------------------------------|
| <i>Host Name</i> | 각 SPU의 호스트 이름을 입력합니다. |
| <i>Disk Unit</i> | 각 디스크의 디스크 드라이브 장치 번호를 입력합니다. |
| <i>Tape Unit</i> | 각 백업 장치의 테이프 장치 번호를 입력합니다. |

Other Unit 기타 장치의 번호를 입력합니다.

Power Supply 호스트나 기타 장치가 연결된 UPS의 전원 공급 장치 번호를 입력합니다.

SPU 전원 한계뿐만 아니라 UPS 및 캐비닛 전원 한계도 지켜야 합니다.

전원 공급 구성 워크시트

다음 워크시트는 전원 공급 장치 구성을 정리하고 기록하는 데 유용합니다. 이 워크시트를 필요한 만큼 복사하여 사용하십시오. 워크시트를 다 채운 후에는 나중에 참조할 수 있도록 잘 보관하십시오.

```
=====
SPU Power:

Host Name ____ftsys9_____ Power Supply ____1_____
Host Name ____ftsys10_____ Power Supply ____2_____

=====
Disk Power:

Disk Unit _____1_____ Power Supply ____3_____
Disk Unit _____2_____ Power Supply ____4_____
Disk Unit _____ Power Supply _____
Disk Unit _____ Power Supply _____
Disk Unit _____ Power Supply _____
Disk Unit _____ Power Supply _____

=====
Tape Backup Power:

Tape Unit _____ Power Supply _____
Tape Unit _____ Power Supply _____

=====
Other Power:
```

HA 클러스터 계획 및 문서화

전원 공급 계획

Unit Name _____ Power Supply _____

Unit Name _____ Power Supply _____

쿼럼 서버 계획

QS(Quorum Server)는 클러스터에 타이 브레이크 서비스를 제공합니다. QS에 대한 자세한 내용은 3장의 “클러스터 잠금으로 쿼럼 서버 사용”에서 설명합니다.

쿼럼 서버는 다음과 같은 특징이 있습니다.

- 최대 50개의 클러스터와 함께 사용할 수 있지만 총 노드 수는 100개를 초과할 수 없습니다.
- 지원되는 수의 노드를 포함한 클러스터를 지원할 수 있습니다.

주

쿼럼 서버가 실행되는 노드와 쿼럼 서버의 서비스를 받는 클러스터는 동일한 서브넷에 있는 것이 좋습니다. 이렇게 하면 쿼럼 서버 작동에 영향을 줄 수 있는 네트워크 지연을 방지할 수 있습니다. 서로 다른 서브넷을 사용할 경우에는 쿼럼 서버 시간 종료를 일으킬 수 있는 네트워크 지연이 발생할 수도 있습니다. 이러한 시간 종료 현상이 발생하지 않도록 하려면 클러스터 ASCII 파일에서 QS_TIMEOUT_EXTENSION 매개 변수를 사용하여 쿼럼 서버의 시간 종료 간격을 늘리면 됩니다.

쿼럼 서버에 연결하는 데 사용되는 네트워크가 클러스터 하트비트 네트워크인 경우에는 하트비트 네트워크 역할을 하는 다른 네트워크가 적어도 하나는 있어야 두 네트워크에서 동시에 장애를 일으키는 것을 방지할 수 있습니다.

쿼럼 서버 워크시트를 사용하여 하나 이상의 클러스터에서 사용할 쿼럼 서버를 지정합니다. 또한 클러스터 구성 워크시트에 쿼럼 서버 호스트 및 타이밍 매개 변수를 입력해야 합니다.

QS 워크시트에 입력할 내용은 다음과 같습니다.

Quorum Server Host

쿼럼 서버의 호스트 이름을 입력합니다.

IP Address

쿼럼 서버에 액세스하는 데 사용할 IP 주소를 입력합니다. 쿼럼 서버는 IPv4 네트워크에 구성해야 합니다. IPv6 주소는 지원되지 않습니다.

Supported Node Names

해당 쿼럼 서버에서 지원할 전체 클러스터 노드의 이름을 입력(31
바이트 이하)합니다. 이러한 항목은 쿼럼 서버 프로세스가 실행
되는 시스템의 `qs_authfile`에 입력됩니다.

쿼럼 서버 워크시트

다음 워크시트는 쿼럼 서버 하드웨어 구성을 정리하고 기록하는 데 유용합니다.
이 워크시트를 필요한 만큼 복사하여 사용하십시오. 워크시트를 다 채운 후에는 나중에
참조할 수 있도록 잘 보관하십시오.

Quorum Server Data:

=====

QS Hostname: _____ IP Address: _____

=====

Quorum Services are Provided for:

Cluster Name: _____

Host Names _____

Host Names _____

Cluster Name: _____

Host Names _____

Host Names _____

LVM 계획

HP-UX Logical Volume Manager(LVM)나 다음 절에 설명된 VERITAS VxVM 및 CVM 소프트웨어를 사용하여 저장 장치 그룹을 만들 수 있습니다.

LVM을 사용하여 디스크 레이아웃을 설계할 때는 다음 사항을 고려해야 합니다.

- 루트 디스크는 자신의 볼륨 그룹에 속해야 합니다.
- 고가용성 응용 프로그램, 서비스 또는 데이터를 포함하는 볼륨 그룹은 기본 노드 및 모든 대체 노드에서 액세스할 수 있는 버스에 있어야 합니다.
- 고가용성 응용 프로그램, 서비스 및 데이터는 고가용성을 지원하지 않는 응용 프로그램, 서비스 및 데이터와는 다른 볼륨 그룹에 위치해야 합니다.
- 제어권을 함께 전달해야 하는 고가용성 응용 프로그램, 서비스 및 데이터는 그룹으로 묶어 하나의 볼륨 그룹 또는 연속된 볼륨 그룹에 위치시켜야 합니다.
- 제어권을 독립적으로 전달해야 하는 서로 다른 두 개의 고가용성 응용 프로그램, 서비스 또는 데이터를 같은 볼륨 그룹에 위치시키면 안 됩니다.
- 루트 디스크는 다른 노드에서 활성화할 수 있는 볼륨 그룹에 속하면 안 됩니다.
- 기본 볼륨 그룹 이름(vg01, vg02 등) 이외의 볼륨 그룹 이름을 사용하는 것이 좋습니다. 연관된 고가용성 응용 프로그램을 나타내는 볼륨 그룹 이름(예: /dev/vegetables)을 선택하면 클러스터를 쉽게 관리할 수 있습니다.

EMS HA 디스크 모니터를 사용하려면 *Using High Availability Monitors(B5736-90025)* 설명서에서 “Rules for Using EMS Disk Monitor with Serviceguard” 절을 참조하십시오.

LVM 워크시트

다음 워크시트는 물리 디스크 구성을 정리하고 기록하는 데 유용합니다. 이 워크시트를 필요한 만큼 복사하여 사용하십시오. 워크시트를 다 채운 후에는 나중에 참조할 수 있도록 잘 보관하십시오. 이 워크시트에는 볼륨 그룹과 물리 볼륨만 포함됩니다. 패키지 구성 워크시트(이 장의 뒷부분에서 설명)에는 각 볼륨 그룹을 구성하는 논리 볼륨과 파일 시스템에 대한 정보를 기록할 수 있는 항목이 있습니다.

=====

Volume Group Name: _____/dev/vg01_____

Name of First Physical Volume Group: _____bus0_____

Physical Volume Name:
_____/dev/dsk/c1t2d0_____

Physical Volume Name:
_____/dev/dsk/c2t2d0_____

Physical Volume Name:
_____/dev/dsk/c3t2d0_____

Physical Volume Name:

Physical Volume Name:

Physical Volume Name:

Physical Volume Name:

Name of Second Physical Volume Group: _____bus1_____

Physical Volume Name:
_____/dev/dsk/c4t2d0_____

Physical Volume Name:
_____/dev/dsk/c5t2d0_____

Physical Volume Name:

_____/dev/dsk/c6t2d0_____

Physical Volume Name:

Physical Volume Name:

Physical Volume Name:

Physical Volume Name:

Physical Volume Name:

CVM 및 VxVM 계획

이전 절에서 설명한 HP-UX Logical Volume Manager(LVM)나 VERITAS VxVM 및 CVM 소프트웨어를 사용하여 저장 장치 그룹을 만들 수 있습니다.

CVM 또는 VxVM 디스크 그룹을 사용하여 저장 장치 구성을 설계할 때는 다음을 고려해야 합니다.

- VxVM 저장 장치를 사용할 각 클러스터 노드에 *rootdg* 디스크 그룹을 만들어야 합니다. **LVM 볼륨 그룹이 사용되는 경우 이 디스크 그룹은 HP-UX 루트 디스크와는 다릅니다.** VxVM 루트 디스크 그룹은 해당 그룹이 만들어진 노드에서 가져올 수만 있습니다. 이 디스크 그룹은 각 클러스터 노드에서 **한 번**만 만들어집니다.
- CVM 디스크 그룹은 클러스터를 구성한 후에 만드는 반면 VxVM 디스크 그룹은 원하는 경우 클러스터 구성 전에 만들 수 있습니다.
- 고가용성 응용 프로그램, 서비스 및 데이터는 고가용성을 지원하지 않는 응용 프로그램, 서비스 및 데이터와는 다른 디스크 그룹에 위치해야 합니다.
- 제어권을 독립적으로 전달해야 하는 서로 다른 두 개의 고가용성 응용 프로그램, 서비스 또는 데이터를 동일한 디스크 그룹에 위치시키면 안 됩니다.
- HP-UX 루트 디스크는 클러스터 노드 간에 공유되지 **않는** LVM 또는 VxVM 볼륨 그룹(VxVM 3.5 이상)에 속할 수 있습니다.
- 클러스터 잠금 디스크는 LVM 볼륨 그룹과 함께 구성할 수만 있습니다.
- VxVM 디스크 그룹 이름은 클러스터 구성 ASCII 파일에 입력할 수 없습니다. 이 이름은 *cmqueryc1*을 통해 클러스터 구성 ASCII 파일에 삽입되지 않습니다.

CVM 및 VxVM 워크시트

다음 워크시트는 물리 디스크 구성을 정리하고 기록하는 데 유용합니다. 이 워크시트를 필요한 만큼 복사하여 사용하십시오. 워크시트를 다 채운 후에는 나중에 참조할 수 있도록 잘 보관하십시오. 이 워크시트에는 볼륨 그룹과 물리 볼륨만 포함됩니다. 패키지 구성 워크시트(이 장의 뒷부분에서 설명)에는 각 볼륨 그룹을 구성하는 논리 볼륨과 파일 시스템에 대한 정보를 기록할 수 있는 항목이 있습니다.

=====

Disk Group Name: _____ dg01 _____

Disk Name: _____ c1t2d0 _____

Disk Name: _____ c2t2d0 _____

Disk Name: _____ c3t2d0 _____

Disk Name: _____

Disk Name: _____

Disk Name: _____

Disk Name: _____

Disk Group Name: _____ dg02 _____

Disk Name: _____ c1t3d0 _____

Disk Name: _____ c2t3d0 _____

Disk Name: _____ c3t3d0 _____

Disk Name: _____

Disk Name: _____

Disk Name: _____

Disk Name: _____

클러스터 구성 계획

클러스터는 장애가 발생했을 때 최대한 빠르게 복구될 수 있도록 설계해야 합니다. 실제로 복구하는 데 걸리는 시간은 다음과 같은 여러 요인에 따라 달라집니다.

- 클러스터 하트비트 간격과 노드 제한 시간의 길이. 이 값은 가능한 한 짧게 설정해야 하지만 각각 1000000(1초)과 2000000(2초)보다 짧아서는 안 됩니다. 하트비트 주기 값은 1000000(1초)으로 설정하고 노드 시간 종료 값은 5~8초 (5000000~8000000) 범위 내에서 설정하는 것이 좋습니다.
- 패키지 제어 스크립트에서 실행 및 중지 명령의 설계. 이 명령은 빠른 실행을 위해 작성해야 합니다.
- 원시 디스크 액세스의 가용성. 원시 디스크 액세스를 사용하는 응용 프로그램은 시스템 중단 복구 서비스로 설계해야 합니다.
- 응용 프로그램 및 데이터베이스 복구 시간. 최단 복구 시간이 되도록 설계해야 합니다.

또한 다음과 같이 클러스터 간에 일관성을 유지해야 합니다.

- 모든 노드에서 사용자 이름이 같아야 합니다.
- 모든 노드에서 UID가 같아야 합니다.
- 모든 노드에서 GID가 같아야 합니다.
- 모든 노드에서 시스템 영역의 응용 프로그램이 같아야 합니다.
- 시스템 시간은 클러스터 전체에서 같아야 합니다.
- /usr 파일과 같이 둘 이상의 노드에서 사용할 수 있는 파일은 모든 노드에서 동일해야 합니다.

Serviceguard Extension for Faster Failover는 두 개의 노드로 구성된 클러스터에서 장애 조치 시간을 최적화할 수 있는 별도 구매 제품입니다. 클러스터는 몇 가지 요구 사항에 맞게 구성해야 합니다. 설치가 완료되면 클러스터 구성 파일에서 매개 변수를 사용하여 이 제품을 활성화할 수 있습니다. 이 제품의 릴리즈 노트는

<http://docs.hp.com> -> high availability에 게시되어 있습니다.

한글 매뉴얼은 다음 웹 사이트에서 찾을 수 있습니다.

<http://docs.hp.com/ko/allproducts.html>

하트비트 서브넷과 재편성 시간

클러스터 재편성 속도는 사용되는 하트비트 네트워크의 종류에 따라 부분적으로 달라집니다. 이더넷이 다른 종류에 비해 장애 조치 속도가 더 느립니다. 하트비트 서브넷을 둘 이상 사용하는 경우에는 장애 조치 시간이 가장 빠른 서브넷이 사용됩니다.

클러스터 잠금 정보

클러스터 잠금의 목적은 이미 클러스터된 노드의 정확히 절반이 새 클러스터를 구성하려고 할 때 새 클러스터가 하나만 구성되도록 하는 것입니다. 새 클러스터는 하나만 구성되고 이 클러스터가 해당 패키지에 지정된 디스크에 단독으로 액세스할 수 있어야 합니다. 잠금 디스크나 퀴럼 서버를 클러스터 잠금으로 지정할 수 있습니다.

노드가 하나인 클러스터에는 잠금이 필요하지 않습니다. 노드가 두 개인 클러스터에는 클러스터 잠금을 사용해야 하고 규모가 더 큰 클러스터에도 잠금을 사용하는 것이 좋습니다. 노드가 다섯 개 이상인 클러스터에서는 퀴럼 서버만 클러스터 잠금으로 사용할 수 있습니다.

클러스터 잠금 디스크와 재편성 시간

잠금 디스크를 사용하는 경우 클러스터 잠금 디스크를 얻는 데 걸리는 시간은 사용되는 디스크 입출력 인터페이스에 따라 달라집니다. 모든 디스크 하드웨어를 구성한 후에는 클러스터를 구성하기에 앞서 클러스터의 모든 노드를 지정하는 `cmquerycl` 명령을 사용하여 사용 가능한 디스크 목록과 각 디스크에 연관된 재편성 시간을 볼 수 있습니다. 예를 들면 다음과 같습니다.

```
# cmquerycl -v -n ftsys9 -n ftsys10
```

또는 SAM을 사용하여 클러스터 잠금 물리 볼륨의 목록과 재편성 시간을 표시할 수 있습니다.

기본적으로 Serviceguard에서는 재편성 시간이 가장 빠른 디스크가 선택됩니다. 그러나 전원 상황을 고려하여 다른 디스크를 선택해야 하는 경우도 있습니다. 가능하면 클러스터 잠금 디스크에는 별도로 전원을 공급해야 합니다.

클러스터 잠금 디스크 및 확장 계획

클러스터가 시작되어 실행 중일 때 클러스터 노드를 추가할 수 있습니다. 그러나 클러스터를 작동 중지시키지 않고 클러스터 노드를 추가하려면 몇 가지 규칙을 따라야 합니다. 노드가 다섯 개 이상인 클러스터에서는 잠금 디스크를 사용할 수 없습니다. 따라서 노드를 새로 추가함으로써 총 노드 수가 다섯 개 이상이 될 때는 퀴럼 서버를 사용해야 합니다.

클러스터 구성 매개 변수

클러스터 관리자의 작동을 위해서는 일련의 클러스터 매개 변수를 정의해야 합니다. 이 매개 변수는 클러스터의 모든 노드에 있는 이진 클러스터 구성 파일에 저장됩니다. 이 매개 변수는 `cmquerycl` 명령을 실행할 때 생성되는 클러스터 구성 템플릿을 편집하여 입력할 수 있습니다(“HA 클러스터 구성 만들기” 참조). 다음 매개 변수 이름은 클러스터 ASCII 구성 파일에서 사용하는 이름입니다.

다음 매개 변수를 지정해야 합니다.

CLUSTER_NAME `cmviewcl` 및 다른 명령의 출력과 클러스터 구성 파일에 나타나는 클러스터의 이름입니다.

클러스터 이름에는 공백, 슬래시(/), 백슬래시(\) 및 별표(*)를 사용할 수 없습니다. 다른 문자는 모두 사용할 수 있습니다. 클러스터 이름은 39자까지 가능합니다.

QS_HOST 현재 클러스터 외부에서 퀴럼 서버 기능을 제공하는 호스트 시스템의 이름 또는 IP 주소입니다. 이 매개 변수는 클러스터의 타이브레이크 서비스를 위해 퀴럼 서버를 사용하는 경우에만 사용됩니다.

QS_POLLING_INTERVAL 퀴럼 서버가 실행 중인지 확인하기 위해 퀴럼 서버와 연락을 시도하는 시간 간격(마이크로초)입니다. 기본값은 300,000,000마이크로초(5분)입니다.

QS_TIMEOUT_EXTENSION

쿼럼 서버 시간 종료는 쿼럼 서버가 클러스터와 통신하지 않는 동안의 시간으로, 이 시간이 지나면 클러스터에서는 쿼럼 서버를 **중지됨**으로 표시합니다. 이 시간은 **Serviceguard** 매개 변수를 기초로 계산되지만 확장 시간을 마이크로초 수로 추가하여 시간을 늘릴 수 있습니다.

`QS_TIMEOUT_EXTENSION`은 선택적 매개 변수입니다.

`FIRST_CLUSTER_LOCK_VG`, `SECOND_CLUSTER_LOCK_VG`

클러스터 잠금이 기록된 물리 디스크 볼륨이 포함된 볼륨 그룹입니다. 두 개의 노드로 구성된 클러스터에서는 클러스터 잠금 볼륨 그룹을 반드시 지정해야 합니다. 두 개의 클러스터 잠금을 만드는 경우에는 각 잠금 모두의 볼륨 그룹 이름을 입력해야 합니다. 이 매개 변수는 클러스터의 타이 브레이크 서비스를 위해 잠금 디스크를 사용하는 경우에만 사용됩니다.

첫 번째 잠금 볼륨 그룹에는 `FIRST_CLUSTER_LOCK_VG`를 사용합니다. 두 번째 잠금 볼륨 그룹이 있으면 매개 변수 `SECOND_CLUSTER_LOCK_VG`가 파일의 다른 줄에 포함됩니다.

주

잠금 볼륨 그룹은 클러스터 **ASCII** 구성 파일에서 `VOLUME_GROUP` 매개 변수로도 정의되어야 합니다.

`NODE_NAME`

클러스터의 노드가 될 각 시스템의 호스트 이름입니다. 노드 이름의 길이는 최대 31바이트까지 가능하며 전체 도메인 이름을 포함하면 안됩니다. 예를 들어, `ftsys9.cup.hp.com`이 아닌 `ftsys9`를 입력합니다.

`NETWORK_INTERFACE`

하트비트 또는 사용자 데이터용으로 사용될 각 **LAN**의 이름입니다. 예를 들면, `lan0`과 같이 입력합니다.

`HEARTBEAT_IP`

클러스터 하트비트를 전달하는 서브넷을 나타내는 IP 표시법입니다. 하트비트 IP 주소는 노드마다 같은 서브넷에 있어야 합니다. 하트비트 IP 주소는 IPv4 주소만 될 수 있습니다.

VERITAS CVM 디스크 그룹을 저장 장치로 사용하는 경우에는 단일 하트비트 서브넷만 사용할 수 있습니다. 이 경우 하트비트는 대기 LAN을 사용하여 구성하거나 통합된 포트 그룹으로 구성해야 합니다.

주

RPC(Remote Procedure Call) 프로토콜 및 서비스를 사용하려는 경우에는 개인 하트비트 네트워크를 사용하지 않는 것이 좋습니다. RPC에서는 각 네트워크 어댑터 장치 또는 입출력 카드가 라우팅 가능한 네트워크에 연결되어 있다고 가정합니다. 개인 또는 격리된 하트비트 LAN은 라우팅할 수 없으므로, 이러한 LAN으로 경로가 지정된 RPC 요청 응답은 시간 종료로 서비스를 받지 못하게 될 수 있습니다.

HP-UX에서 자주 사용되는 RPC 기반 응용 프로그램으로는 NFS, NIS 및 NIS+, CDE가 있습니다. 타사 및 국내에서 개발된 응용 프로그램도 RPC API 라이브러리를 통해 직접적으로 RPC 서비스를 이용할 수 있습니다. RPC 사용에 관한 자세한 정보는 응용 프로그램 공급업체에 문의하십시오.

STATIONARY_IP

클러스터 하트비트를 전달하지 않는 모니터링되는 각 서브넷의 IP 주소입니다. 모니터링될 서브넷은 수에 제한 없이 지정할 수 있습니다. 응용 프로그램 데이터와 하트비트 메시지를 분리하려면 이 항목에서 모니터링되는 비 하트비트 서브넷을 정의합니다.

고정 IP 주소는 IPv4 또는 IPv6 주소가 될 수 있습니다. IPv6 주소 형식에 대한 자세한 내용은 부록, 424페이지의 “IPv6 주소 유형”을 참조하십시오.

FIRST_CLUSTER_LOCK_PV, SECOND_CLUSTER_LOCK_PV

잠금 볼륨 그룹에서 클러스터 잠금이 기록될 물리 볼륨의 이름입니다. 이 매개 변수는 첫 번째 물리 잠금 볼륨에 대해

FIRST_CLUSTER_LOCK_PV로 지정하고 두 번째 물리 잠금 볼륨에

대해서는 `SECOND_CLUSTER_LOCK_PV`로 지정합니다. 두 번째 물리 잠금 볼륨이 있으면 매개 변수 `SECOND_CLUSTER_LOCK_PV`가 파일의 다른 줄에 포함됩니다. 이러한 매개 변수는 클러스터의 타미 브레이크 서비스를 위해 잠금 디스크를 사용하는 경우에만 사용됩니다.

클러스터의 두 노드에 표시되는 것과 동일한 물리 볼륨 이름을 입력합니다(각 노드에서 동일한 물리 볼륨이 서로 다른 이름일 수 있음). 클러스터 잠금을 두 개 만들 경우 두 개의 잠금 모두에 대한 물리 볼륨 이름을 입력합니다. 물리 볼륨 그룹 식별자는 39자까지 가능합니다.

`SERIAL_DEVICE_FILE`

각 노드에서 선택한 직렬(**RS232**) 포트에 해당되는 장치 파일의 이름입니다. **RS232**를 하트비트 회선으로 사용할 경우 이 매개 변수를 지정합니다.

ASCII 클러스터 구성 파일에서는 이 매개 변수가 `SERIAL_DEVICE_FILE`입니다. 장치 파일 이름은 39자까지 가능합니다.

`HEARTBEAT_INTERVAL`

클러스터의 한 노드에서 다른 노드로 하트비트 메시지를 전송할 때의 일반적인 시간 간격입니다. 값은 초 단위로 입력합니다.

ASCII 클러스터 구성 파일에서는 이 매개 변수가 `HEARTBEAT_INTERVAL`이고 값은 마이크로초 단위로 입력됩니다. 기본값은 1,000,000마이크로초입니다. 이 매개 변수를 기본값보다 작은 값으로 설정하는 것은 **좋지 않습니다**.

가능하면 기본값을 사용해야 합니다. 권장되는 최대값은 15초이고 지원되는 최대값은 30초입니다. 이 값은 *Node Timeout* (아래 참조) 값의 1/2 이상이어야 합니다.

`NODE_TIMEOUT`

한 노드에서 다른 노드를 사용 불가능하다고 결정하고 클러스터 재편성을 시작하기 위해 경과되어야 하는 시간입니다. 이 매개 변수는 마이크로초 단위로 입력합니다.

기본값은 ASCII 파일에서 2,000,000마이크로초입니다. 최소값은 2 * (하트비트 간격)입니다. 권장되는 최대값은 ASCII 파일에서 30,000,000이고 Serviceguard Manager에서는 30초입니다.

기본값을 사용할 경우에 가장 빠르게 클러스터를 재편성할 수 있습니다. 그러나 기본값을 사용하게 되면 일시적인 시스템 정지나 네트워크 과부하로 인해 비정상적인 재편성이 일어날 가능성이 증가할 수 있습니다. 대부분의 경우에는

5,000,000~8,000,000(5~8초)으로 설정하는 것이 적절합니다.

권장되는 최대값은 30초이고 지원되는 최대값은 60초입니다.

AUTO_START_TIMEOUT

자동으로 클러스터가 시작되는 동안 노드에서는 클러스터에 참가하려는 시도를 하다가 이 시간이 지나면 참가 시도를 중지합니다.

ASCII 클러스터 구성 파일에서는 이 매개 변수가

AUTO_START_TIMEOUT입니다. 모든 노드에서는 클러스터가 작업을 완료하기 전에 다른 노드가 시작될 수 있도록 이 매개 변수로 지정된 시간 동안 기다려야 합니다. 이 시간을 선택할 때는 클러스터에서 가장 느린 부팅 시간을 기준으로 해야 합니다. 가장 느린 부팅 노드의 부팅 시간에서 가장 빠른 부팅 노드의 부팅 시간을 빼고 여기에 600초(10분)를 더한 값을 입력합니다.

기본값은 ASCII 파일에서 600,000,000마이크로초이고 Serviceguard Manager에서는 600초입니다.

NETWORK_POLLING_INTERVAL

Serviceguard에 구성된 네트워크를 검사하는 주기입니다. ASCII 클러스터 구성 파일에서는 이 매개 변수가

NETWORK_POLLING_INTERVAL입니다.

기본값은 ASCII 파일에서 2,000,000마이크로초이고

Serviceguard Manager에서는 2초입니다. 따라서 클러스터 관리자 2초마다 각 네트워크 인터페이스를 폴링하여 여전히 정보를 보내고 받을 수 있는지 확인합니다. 이 값을 변경하면 네트워크 장

애를 감지하는 속도가 달라질 수 있습니다. 최소값은 1,000,000 (1초)입니다. 권장되는 최대값은 15초이고 지원되는 최대값은 30초입니다.

MAX_CONFIGURED_PACKAGES

이 매개 변수는 클러스터에 구성될 수 있는 최대 패키지 수를 설정합니다. ASCII 클러스터 구성 파일에서는 이 매개 변수가 MAX_CONFIGURED_PACKAGES입니다.

기본값은 0입니다. 따라서 패키지를 사용하려면 이 매개 변수를 변경해야 합니다. 최소값은 0이고 최대값은 150입니다. 나중에 패키지를 추가할 때 이 매개 변수를 다시 설정하기 위해 클러스터를 작동 중지시켜야 할 필요가 없도록 하려면 값을 충분히 크게 설정합니다. 그러나 너무 높게 설정하면 메모리가 낭비되므로 좋지 않습니다. 패키지를 사용하려면 모든 클러스터 노드에 대해 6MB와 100KB 정도의 잠금 가능 메모리가 필요합니다. CVM 디스크 저장 장치를 사용하는 경우에는 VxVM-CVM-pkg용으로 하나를 추가합니다.

주

각 노드에서 HP-UX 커널 매개 변수를 조정하여 해당 노드에서 동시에 실행되는 가장 큰 수의 패키지에 적절하게 설정해야 합니다.

VOLUME_GROUP

클러스터의 노드 두 개 이상에 연결된 디스크가 포함된 LVM 볼륨 그룹의 이름입니다. 이러한 디스크를 클러스터 인식 디스크라고 합니다. ASCII 클러스터 구성 파일에서는 이 매개 변수가 VOLUME_GROUP입니다. 볼륨 그룹 이름은 39자까지 가능합니다.

Access Control Policies

각 정책에 대해 `USER_NAME`, `USER_HOST` 및 `USER_ROLE`의 세 가지 요소를 지정합니다. `Serviceguard Manager`의 경우 `USER_HOST`는 세션 노드의 이름이어야 합니다. 클러스터와 클러스터 패키지의 구성 파일에 설정된 정책은 충돌하거나 중복되지 않아야 합니다. 자세한 내용은 182페이지의 “보안 파일 편집”을 참조하십시오.

FAILOVER_OPTIMIZATION

별도 구입 제품인 `Serviceguard Extension for Faster Failover`를 설치한 경우에만 이 매개 변수를 볼 수 있습니다. 이 매개 변수를 `TWO_NODE`로 설정하여 제품을 활성화할 수 있습니다. 기본값은 `NONE`이므로 사용되지 않습니다. 이 제품과 클러스터 구성 요구 사항에 대한 자세한 내용을 보려면 <http://www.docs.hp.com/hpux/ha>에서 `Serviceguard Extension for Faster Failover`를 클릭하십시오.

NETWORK_FAILURE_DETECTION

주 네트워크 카드와 대기 네트워크 카드가 있는 경우 `Serviceguard`에서는 카드에 장애가 발생했을 때 이를 판별하고 장애 조치를 위해 트래픽을 다른 카드로 전달할지 여부를 알아야 합니다. `Serviceguard` 네트워크 관리자는 장애를 감지하기 위해 들어오는 트래픽과 나가는 트래픽을 모두 모니터링하고, 일정 시간 동안 네트워크 트래픽이 없으면 해당 카드를 중지된 상태로 표시한 후 장애 조치를 시도하기 시작합니다. `Serviceguard`에서 시간을 계산하는 방식은 LAN 카드의 종류에 따라 다릅니다.

구성 파일에서는 네트워크 인터페이스 카드에 장애가 발생한 경우 이를 확인하기 위한 방법을 다음 중 하나로 지정합니다.

- `INOUT` - 기본적으로 사용되는 방법으로, 수신 장애 수와 송신 장애 수를 따로 계산하고 두 수가 모두 중요 수준에 도달한 경우에만 해당 카드를 작동 중지된 것으로 선언합니다.
- `INONLY_OR_INOUT` - 이 옵션은 수신 장애와 송신 장애를 결합한 것으로 소스에 상관없이 총 장애 횟수가 일정 수준 이상으로 커지면 해당 카드가 작동 중지된 것으로 선언합니다. 이 방법을 사용할 경우 `Serviceguard`에서는 추가 원격 폴링을 수행하여 수신 장애 보고서의 유효성을 확인합니다.

기본값은 `INOUT`입니다.

이 옵션이 적합한지 여부는 주로 네트워크 구성에 따라 달라집니다. 새 **INONLY_OR_INOUT** 옵션이 현재 네트워크 구성에 가장 적합한지 여부에 대한 자세한 내용을 보려면

<http://docs.hp.com/hpux/ha> -> **Serviceguard White Papers**에서 “**Inbound Failure Detection Enhancement**”를 참조하십시오.

클러스터 구성 워크시트

다음 워크시트는 클러스터 하드웨어 구성을 정리하고 기록하는 데 유용합니다.

Name and Nodes:

=====

Cluster Name: ourcluster

Node Names: node1 node2

Maximum Configured Packages: 12

=====

Quorum Server Data:

=====

Quorum Server Host Name or IP Address: lp_qs

Quorum Server Polling Interval: 300000000 microseconds

Quorum Server Timeout Extension: microseconds

=====

Subnets:

=====

Heartbeat Subnet: 15.13.168.0

Monitored Non-heartbeat Subnet: 15.12.172.0

Monitored Non-heartbeat Subnet:

=====

Cluster Lock Volume Groups and Volumes:

=====

First Lock Volume Group:

Physical Volume: |

Name on Node 1:

Name on Node 2:

Disk Unit No:

```

|
| Power Supply No: _____
=====
Timing Parameters:
=====
Heartbeat Interval: _1 sec_
=====
Node Timeout: _2 sec_
=====
Network Polling Interval: _2 sec_
Network Monitor _INOUT_
=====
Autostart Delay: _10 min___
=====
Cluster Aware LVM Volume Groups _____
=====
Access Policies
  User: __ ANY_USER
  Host: __ ftsys9__
  Role: __ full_admin__

  User: __ sara itgrp lee __
  Host: __ ftsys10__
  Role: __ package_admin__
=====
```

패키지 구성 계획

패키지 계획에는 고가용성 서비스의 각 그룹에 대한 정보를 모으는 작업이 포함됩니다. 이 정보 가운데 일부는 패키지 구성 파일을 작성하는 데 사용되고 일부는 패키지 제어 스크립트를 편집하는 데 사용됩니다.

주

패키지에서 활성화할 LVM 볼륨 그룹은 클러스터 구성 파일에서도 클러스터 인식 볼륨 그룹으로 정의되어야 합니다. 자세한 내용은 앞의 “클러스터 구성 계획” 절을 참조하십시오. 패키지에서 활성화할 VERITAS 디스크 그룹은 패키지 구성 ASCII 파일에서 다음과 같이 정의되어야 합니다.

논리 볼륨 및 파일 시스템 계획

볼륨 그룹의 논리 볼륨을 클러스터에서 패키지 작동을 위한 하부 구조의 일부로 사용해야 하는 경우가 있습니다. 패키지가 한 노드에서 다른 노드로 이동되는 경우 이전 노드에서와 같은 디스크에 있는 데이터에 액세스할 수 있어야 합니다. 이렇게 하려면 볼륨 그룹을 활성화하고 해당 볼륨 그룹에 상주하는 파일 시스템을 마운트합니다.

Serviceguard에서 고가용성 응용 프로그램, 서비스 및 데이터는 공유 버스상의 볼륨 그룹에 위치합니다. 노드에서 장애가 발생하면 이 응용 프로그램, 서비스 및 데이터를 포함하는 볼륨 그룹은 장애 노드에서 비활성화되고 대체 노드에서 다시 활성화됩니다. 이렇게 하려면 볼륨 그룹을 장애 노드에서 대체 노드로 이동할 수 있도록 구성해야 합니다.

패키지 구성을 계획할 때는 다음을 결정해야 합니다.

- 어떤 볼륨 그룹이 필요합니까?
- 디스크 공간이 어느 정도 필요하며 이것을 논리 볼륨에 어떻게 할당해야 합니까?
- 각 패키지에 어떤 파일 시스템이 마운트되어야 합니까?
- 각 논리 볼륨 구성을 가져오는 데 어떤 노드가 필요합니까?

- 패키지가 대체 노드로 이동되면 성능에 어떤 효과가 있습니까?

볼륨 그룹, 논리 볼륨 및 파일 시스템 패키지별로 목록을 만드십시오. 그런 다음 서로 다른 시간에 공통 파일 시스템에 액세스해야 하는 노드를 기록하십시오.

기본 논리 볼륨 이름(lvol1, lvol2 등)과는 다른 사용자 지정 논리 볼륨 이름을 사용하는 것이 좋습니다. 연관된 고가용성 응용 프로그램을 나타내는 논리 볼륨 이름(예: lvoldatabase)을 선택하면 클러스터를 쉽게 관리할 수 있습니다.

/etc/fstab 파일에 *commented* 행을 추가하여 각 노드의 패키지 관련 볼륨 그룹, 논리 볼륨, 파일 시스템을 보다 자세히 문서화할 수 있습니다. 다음은 데이터베이스 응용 프로그램용으로 구성된 예제입니다.

```
# /dev/vg01/lvoldb1 /applic1 vxfs defaults 0 1 # These six e
ntries are
# /dev/vg01/lvoldb2 /applic2 vxfs defaults 0 1 # for informa
tion purposes
# /dev/vg01/lvoldb3 raw_tables ignore ignore 0 0 # only. They
record the
# /dev/vg01/lvoldb4 /general vxfs defaults 0 2 # logical vol
umes that
# /dev/vg01/lvoldb5 raw_free ignore ignore 0 0 # exist for S
erviceguard's
# /dev/vg01/lvoldb6 raw_free ignore ignore 0 0 # HA package.
Do not uncomment.
```

각 논리 볼륨에 파일 시스템 또는 원시 장치에 대한 사용을 나타내는 항목을 만듭니다.

주의

Serviceguard 패키지에서 사용하는 파일 시스템을 마운트할 때는 /etc/fstab를 사용하지 **마십시오**.

Serviceguard에서 볼륨 그룹을 만들고, 내보내고, 가져오는 방법에 대한 자세한 내용은 “HA 클러스터 구성 만들기” 장을 참조하십시오.

EMS 리소스 구성 매개 변수

Serviceguard에서는 EMS 리소스 구성을 위한 일련의 매개 변수를 제공합니다. 이러한 매개 변수에는 RESOURCE_NAME, RESOURCE_POLLING_INTERVAL, RESOURCE_START 및 RESOURCE_UP_VALUE 등이 포함됩니다. 패키지가 의존하는 각각의 리소스에 대해 이 매개 변수를 패키지 구성 파일에 입력하십시오. RESOURCE_START 가 DEFERRED로 설정된 리소스가 있으면 패키지 제어 스크립트에 해당 리소스에 대한 DEFERRED_RESOURCE_NAME 항목을 추가합니다.

RESOURCE_START 옵션은 Serviceguard에서 EMS 리소스에 대한 리소스 모니터링을 시작해야 할 시점을 결정하는 데 사용됩니다. RESOURCE_START 옵션은 AUTOMATIC 또는 DEFERRED로 설정할 수 있습니다.

AUTOMATIC을 지정할 경우 Serviceguard 클러스터 데몬이 노드에서 시작되면 Serviceguard에서는 자동으로 이 리소스에 대한 리소스 모니터링을 시작합니다. 리소스가 AUTOMATIC으로 구성되어 있으면 패키지 제어 스크립트에서 DEFERRED_RESOURCE_NAME을 정의할 필요가 없습니다.

DEFERRED를 선택하면 Serviceguard에서는 노드가 시작될 때 이 DEFERRED 리소스에 대해 리소스 모니터링을 시작하려고 시도하지 않습니다. 그러나 이 DEFERRED 리소스는 패키지 제어 스크립트에서 DEFERRED_RESOURCE_NAME 매개 변수로 지정하여 패키지가 실행될 때 패키지 제어 스크립트에 의해 시작되도록 해야 합니다.

다음 예제에서는 DEFERRED 및 AUTOMATIC 리소스를 구성하는 방법을 보여 줍니다. 패키지 구성 파일에서 다음과 같이 리소스를 지정하십시오.

```
RESOURCE_NAME           /net/interfaces/lan/status/lan0
RESOURCE_POLLING_INTERVAL 60
RESOURCE_START           DEFERRED
RESOURCE_UP_VALUE        = UP
```

```
RESOURCE_NAME           /net/interfaces/lan/status/lan1
RESOURCE_POLLING_INTERVAL 60
RESOURCE_START           DEFERRED
RESOURCE_UP_VALUE        = UP
```

```
RESOURCE_NAME           /net/interfaces/lan/status/lan2
RESOURCE_POLLING_INTERVAL 60
```

```
RESOURCE_START          AUTOMATIC
RESOURCE_UP_VALUE       = UP
```

패키지 제어 스크립트에서는 다음과 같이 DEFERRED_RESOURCE_NAME 매개 변수를 사용하여 지연 시작 리소스만 지정합니다.

```
DEFERRED_RESOURCE_NAME[0]="/net/interfaces/lan/status/lan0"
DEFERRED_RESOURCE_NAME[1]="/net/interfaces/lan/status/lan1"
```

확장을 고려한 계획

클러스터가 작동 중일 때 패키지를 추가할 수 있습니다. 자세한 방법은 “클러스터 및 패키지 관리” 장에서 설명합니다. 패키지를 추가할 때는 클러스터 구성 파일에 정의되어 있는 MAX_CONFIGURED_PACKAGES 값을 초과하지 않도록 주의해야 합니다.

패키지 수를 늘릴 예정인 경우에는 패키지를 사용하려면 각 클러스터 노드에서 6MB 외에 100KB 정도의 잠금 가능 메모리가 필요하다는 것을 기억해야 합니다.

전환 및 장애 조치 동작 선택

동일한 실제 서버넷의 LAN 카드에서 장애가 발생할 때 대기 LAN 카드로 IP 주소를 전환하는 동작은 Serviceguard Manager에서 자동 전환이 사용 가능으로 설정된 경우 (ASCII 패키지 구성 파일에서는 LOCAL_LAN_FAILOVER_ALLOWED가 YES로 설정된 경우)에 일어납니다. 자동 전환 사용 기능이 기본값입니다.

패키지 장애 조치 동작을 결정하기 위해 실행되고 있지 않은 패키지를 자동으로 실행할 노드를 결정하는 패키지 장애 조치 정책을 정의할 수 있습니다. 또한 기본 노드를 사용할 수 있게 되었을 때 패키지를 기본 노드로 자동 전환할지 여부를 결정하는 장애 복구 정책도 정의할 수 있습니다.

이전 장의 표 3-3에서는 장애 조치 동작의 종류와, 각 동작을 결정하는 Serviceguard Manager 또는 ASCII 패키지 구성 파일의 설정에 대해 설명합니다.

패키지 구성 파일 매개 변수

패키지 구성 파일을 생성하기 전에 다음 패키지 구성 데이터를 조합하십시오. 아래 주어진 매개 변수 이름은 Serviceguard Manager에 나타나는 이름입니다. ASCII 클러스터 구성 파일의 코딩 이름은 각 항목의 끝에 표시되어 있습니다. **각 패키지에 대해** 다음 매개 변수를 지정하고 워크시트에 입력해야 합니다.

Package name

패키지의 이름입니다. 패키지 이름은 클러스터 내에서 고유해야 합니다. 패키지를 시작, 중지, 수정하고 볼 때 이 이름을 사용합니다. **ASCII** 패키지 구성 파일에서는 이 매개 변수가 `PACKAGE_NAME`입니다.

패키지 **ASCII** 파일에서 패키지 이름에는 **1~39**자를 사용할 수 있습니다. 패키지 이름에는 공백, 슬래시(/), 백슬래시(\), 별표(*)를 사용할 수 없습니다. 다른 문자는 모두 사용할 수 있습니다.

Serviceguard Manager에서는 영숫자 이외의 문자를 사용할 수 없습니다. 자세한 내용은 온라인 도움말 항목의 “패키지 구성”을 참조하십시오.

PACKAGE_TYPE

패키지 유형입니다. 이 매개 변수는 패키지가 한 번에 한 노드에서 실행되는지, 아니면 한 번에 여러 노드에서 실행되는지를 나타냅니다. 유효한 유형은 `FAILOVER` 및 `SYSTEM_MULTI_NODE`입니다. 기본값은 `FAILOVER`입니다.

`SYSTEM_MULTI_NODE` 유형은 모든 클러스터 노드 간에 공유 **CVM** 데이터 저장 장치를 지원하는 **VxVM-CVM-pkg** 패키지를 정의하는 경우에만 사용됩니다. `SYSTEM_MULTI_NODE` 유형으로는 사용자 정의 패키지를 만들 수 없습니다.

Failover policy

패키지가 자동으로 시작되어야 할 노드를 선택하기 위해 패키지 관리자가 사용하는 정책입니다. **ASCII** 패키지 구성 파일에서 이 매개 변수는 `FAILOVER_POLICY`입니다.

기본값은 `CONFIGURED_NODE`로, 이 경우에는 해당 패키지에 대한 노드 이름 목록에서 다음으로 사용할 수 있는 노드를 선택합니다. 기본 설정 순서는 패키지의 구성 파일에서 노드 이름 항목이 나열된 순서입니다.

대체 정책은 `MIN_PACKAGE_NODE`로, 이 경우에는 해당 패키지를 실행할 수 있는 각 노드에서 실행 중인 패키지의 목록을 만들고 가장 적은 수의 패키지를 실행하고 있는 노드를 선택하게 됩니다.

Failback policy

기본 노드에서 실행할 수 있는 패키지인데 기본 노드에서 실행되고 있지 않은 경우에 패키지 관리자가 취해야 하는 조치를 결정하는데 사용되는 정책입니다. `ASCII` 패키지 구성 파일에서 이 매개 변수는 `FAILBACK_POLICY`입니다.

기본값은 `MANUAL`로, 이 경우에는 패키지가 대체 노드에서 실행 중일 때 패키지를 기본 노드로 다시 이동하기 위한 시도를 하지 않습니다. 이 동작은 이전 버전의 `Serviceguard`에서와 동일합니다. 대체 정책은 `AUTOMATIC`입니다. 이 값을 설정하면 패키지 장애 조치 정책으로 `MIN_PACKAGE_NODE`가 선택되어 있고 패키지의 기본 노드가 현재 노드보다 적은 수의 패키지를 실행하고 있을 경우 기본 노드에서 패키지를 실행할 수 있게 되는 즉시 패키지가 중지되었다가 기본 노드에서 다시 시작됩니다.

Package nodes

패키지의 기본 노드 및 대체 노드의 이름입니다. `ASCII` 구성 파일에서는 이 매개 변수가 `NODE_NAME`입니다. 이 매개 변수에는 패키지를 실행할 수 있는 각 노드의 노드 이름을 입력합니다.

노드 이름을 지정하는 순서는 중요합니다. 먼저 기본 노드 이름을 선택한 다음 첫 번째 대체 노드 이름, 그 다음에 두 번째 대체 노드 이름 순으로 선택하십시오. 장애 조치 시 패키지 제어권은 패키지 구성 파일에 나열된 다음 대체 노드로 전달됩니다.

클러스터에 있는 모든 노드를 사용해야 하고 사용 순서가 중요하지 않은 경우에는 `NODE_NAME *`으로 지정할 수 있습니다.

노드 이름의 길이는 31바이트까지 가능합니다.

Package auto run

ASCII 패키지 구성 파일에서는 이 매개 변수가 `AUTO_RUN`이고 이전에는 `PKG_SWITCHING_ENABLED`였습니다. 이 매개 변수로는 패키지가 시작되는 방법을 지정합니다. 자동 전환을 사용 가능으로 설정(`AUTO_RUN`을 YES로 설정)하면 패키지는 적합한 노드를 사용할 수 있을 때 해당 노드에서 자동으로 시작되고 장애가 발생할 때 자동으로 다른 노드로 이동됩니다. 자동 전환을 사용 불가(`AUTO_RUN`을 NO로 설정)하면 클러스터가 시작될 때 패키지가 자동으로 시작되지 않으며 장애가 발생해도 다른 노드로 이동되지 않습니다.

ASCII 파일에서 `AUTO_RUN`의 기본값은 YES로, 이 경우 패키지는 사용 가능한 노드에서 정상적으로 시작될 수 있습니다.

Serviceguard Manager에서는 해당 확인란을 선택하여 자동 실행 기능을 활성화하고 ASCII 파일에서는 `AUTO_RUN` YES 또는 NO를 입력합니다.

Local LAN failover

Enabled 또는 **Disabled**를 입력합니다. 이 매개 변수는 장애가 발생할 경우 **Serviceguard**에서 로컬로 LAN을 전환할 수 있도록 허용합니다. 즉, 장애가 발생하면 패키지 IP 주소를 대기 LAN 카드로 전달할 수 있습니다. 기본값은 **Enabled**입니다. ASCII 패키지 구성 파일에서는 이 매개 변수가 `LOCAL_LAN_FAILOVER_ALLOWED`이며 사용할 수 있는 값은 YES와 NO입니다.

Serviceguard Manger에서는 기본값이 미리 선택되어 있고 ASCII 파일에서는 기본값이 YES입니다.

Node fail fast

제어 스크립트 자체에서 장애가 발생하거나 서버넷에 장애가 발생하거나 리소스가 작동 중지되었음을 나타내는 **EMS** 모니터 이벤트가 보고될 경우, 이 매개 변수가 **Enabled**로 설정되어 있으면 **Serviceguard**는 제어 스크립트 장애가 발생한 노드에서 **TOC**를 실행합니다. ASCII 패키지 구성 파일에서 이 매개 변수는 `NODE_FAIL_FAST_ENABLED`이며 사용할 수 있는 값은 YES와 NO입니다.

기본값은 **Serviceguard Manger**에서는 사용 불가(선택되어 있지 않음)이고 ASCII 파일에서는 NO입니다.

NODE_FAIL_FAST_ENABLED를 YES로 설정하면 다음 장애 중 하나가 발생할 경우 패키지의 실행 노드가 중지됩니다.

- 패키지 서브넷에 장애가 발생했으며 백업 네트워크를 사용할 수 없는 경우
- EMS 리소스에 장애가 발생한 경우
- 중지 스크립트가 없는 경우
- Serviceguard에서 중지 스크립트를 실행할 수 없는 경우
- 중지 스크립트 또는 실행 스크립트가 시간 종료된 경우

그러나 패키지 중지 스크립트가 “exit 1”과 함께 실패하면 Serviceguard에서는 노드를 중지하지는 않지만 해당 패키지에 대해 NO_RESTART를 설정합니다. 이렇게 하면 패키지 전환 (AUTO_RUN)이 비활성화되므로 대체 노드에서 패키지가 시작되지 않습니다.

Controlscript pathname

Serviceguard Manager의 기본값인 유도 모드를 사용할 경우 Serviceguard Manager에서는 제어 스크립트를 자동으로 만들고 유지 관리합니다. 필요하면 사용자가 직접 제어 스크립트를 편집하고 경로의 이름을 지정할 수 있습니다. 그러나 유도 모드 이외의 값을 사용할 때는 Serviceguard Manager에서 스크립트를 업데이트할 수 없습니다.

패키지 제어 스크립트의 전체 경로 이름을 입력합니다. 스크립트는 “cmcluster”라는 문자열이 포함된 디렉토리에 있어야 합니다. 스크립트가 실행 가능한지 확인하십시오. ASCII 패키지 구성 파일에서 이 매개 변수는 RUN_SCRIPT 및 HALT_SCRIPT라는 두 개의 개별 매개 변수에 매핑됩니다.

실행 및 중지 스크립트와 동일한 스크립트를 사용하는 것이 좋습니다. 이 스크립트에는 패키지 실행 명령과 패키지 중지 명령이 모두 포함됩니다. 패키지 실행 명령과 패키지 중지 명령을 별도의 스크립트로 나누려면 패키지 구성 파일을 사용하여 두 개의 개별 스크립트 이름을 지정하면 됩니다. 그러나 대부분의 경우에는 ASCII 파일에서 RUN_SCRIPT와 HALT_SCRIPT의 이름으로 하나의 제어 스크립트 이름을 사용하는 것이 더 간단합니다. 패키지가

시작되면 실행 스크립트가 실행되고 매개 변수 “start”가 전달되며, 패키지가 중지되면 중지 스크립트가 실행되고 매개 변수 “stop”이 전달됩니다.

별도의 패키지 실행 및 중지 스크립트를 작성하려면 두 스크립트 모두에 동일한 구성 정보(노드 이름, IP 주소 등)를 포함해야 합니다.

Run script timeout 및 Halt script timeout

스크립트가 지정된 제한 시간까지 완료되지 않으면 Serviceguard에서 스크립트를 종료시킵니다. ASCII 구성 파일에서 이 매개 변수는 RUN_SCRIPT_TIMEOUT과 HALT_SCRIPT_TIMEOUT입니다. 초 단위로 값을 입력합니다.

기본값은 0으로 제한 시간이 없음을 의미합니다. 최소값은 10초이지만 HALT_SCRIPT_TIMEOUT 최소값은 모든 Service Halt Timeout 값의 합보다 커야 합니다. 절대 최대값은 HP-UX 매개 변수 ULONG_MAX에 의해서만 제한되며, 이 값은 4,294초입니다.

제한 시간이 초과되면 다음 결과가 발생합니다.

- 패키지 제어권이 전달되지 않습니다.
- 실행 또는 중지 명령이 실행되지 않습니다.
- 글로벌 전환을 할 수 없게 됩니다.
- 현재 노드에서 패키지를 실행할 수 없게 됩니다.
- 제어 스크립트가 상태 1로 종료됩니다.

중지 스크립트의 제한 시간이 초과되면 수동 정리를 수행해야 할 수 있습니다. 자세한 내용은 8장의 “패키지 제어 스크립트 중지 또는 실패”를 참조하십시오.

주

VxVM 디스크 그룹은 패키지가 실행될 때 가져오고 패키지가 중지될 때 내보냅니다. 패키지에서 **VxVM** 디스크를 많이 사용하고 있는 경우에는 모든 디스크에서 가져오거나 내보내기를 마칠 수 있도록 제한 시간을 충분히 높게 설정해야 합니다.

CVM diskgroups

이 매개 변수는 **CVM** 디스크 그룹에 사용됩니다. 패키지에서 사용할 모든 **CVM** 디스크 그룹의 이름을 입력합니다.

ASCII 패키지 구성 파일에서는 이 매개 변수가 `STORAGE_GROUP`입니다.

주

패키지 **ASCII** 구성 파일에 **LVM** 볼륨 그룹 또는 **VxVM** 디스크 그룹의 이름을 입력하면 안 됩니다.

Service name

각 서비스의 고유 이름을 입력합니다. **ASCII** 패키지 구성 파일에서는 이 매개 변수가 `SERVICE_NAME`입니다.

각 서비스에 대해 하나의 `SERVICE_NAME` 항목을 정의하십시오. 패키지당 최대 30개의 서비스와 클러스터당 최대 900개의 서비스를 구성할 수 있습니다. 서비스 이름에는 공백, 슬래시(/), 백슬래시(\), 별표(*)를 사용할 수 없습니다. 다른 문자는 모두 사용할 수 있습니다. 서비스 이름은 39자까지 가능합니다.

Service fail fast

각 서비스에 대해 **Enabled** 또는 **Disabled**를 입력합니다. 이 매개 변수는 서비스에 장애가 발생할 경우 노드에도 장애가 발생하는지 여부를 나타냅니다. 이 매개 변수를 **Enabled**로 설정하면 **Serviceguard**에서는 서비스에 장애가 발생할 때 **TOC**를 통해 서비스가 실행되고 있는 노드를 중지시킵니다. **TOC**를 수행하기 전에는 노드를 다시 부팅하기 위한 시도가 이루어집니다. 기본값은 사용 불가능합니다.

ASCII 패키지 구성 파일에서는 이 매개 변수가 `SERVICE_FAIL_FAST_ENABLED`이며 사용 가능한 값은 YES와 NO입니다. 기본값은 NO입니다. 각 서비스에 대해 하나의 `SERVICE_FAIL_FAST_ENABLED` 항목을 정의합니다.

Service halt timeout

서비스가 중지되면 **Serviceguard**에서는 먼저 **SIGTERM** 신호를 보내 서비스를 종료합니다. 프로세스가 종료되지 않으면 지정된 제한 시간 동안 대기한 다음 **SIGKILL** 신호를 보내 프로세스를 강제로 종료합니다. ASCII 패키지 구성 파일에서는 이 매개 변수가 `SERVICE_HALT_TIMEOUT`입니다.

값을 지정하지 않으면 **Serviceguard**에서 시간 제한을 허용하지 않게 됩니다(0초). 최대값은 HP-UX 매개 변수 `ULONG_MAX`에 의해서만 제한되며, 이 값은 4,294초입니다.

각 서비스에 대해 하나의 `SERVICE_HALT_TIMEOUT` 항목을 정의합니다.

Subnet

패키지에 대해 모니터링될 IP 서브넷을 입력합니다.

ASCII 패키지 구성 파일에서는 이 매개 변수가 `SUBNET`입니다.

EMS resource

패키지에서 의존하는 리소스로서 **Serviceguard**에서 모니터링할 리소스의 이름입니다. ASCII 패키지 구성 파일에서는 이 매개 변수가 `RESOURCE_NAME`입니다.

리소스 이름은 특정 시스템 리소스의 중요 속성 이름입니다. 리소스 이름에는 시스템에서 해당 리소스가 포함되는 리소스 클래스와 하위 클래스의 계층 구조 전체가 포함됩니다. **Serviceguard Manager**의 EMS 탭에서 찾아보기 단추를 클릭하면 표시되는 목록("사용 가능한 EMS 리소스")이나 리소스 모니터와 함께 제공되는 설명서에서 리소스 이름을 찾을 수 있습니다.

클러스터당 최대 60개의 리소스를 정의할 수 있습니다. 아래에 설명된 Resource Up Values의 한계도 참조하십시오.

리소스 이름 문자열의 최대 길이는 1024자입니다.

Resource polling interval

구성된 패키지 리소스를 모니터링하는 주기입니다. ASCII 패키지 구성 파일에서는 이 매개 변수가 RESOURCE_POLLING_INTERVAL입니다.

기본값은 60초입니다. 최소값은 1이며 최대값은 기능적으로 제한되지 않습니다.

Serviceguard Manger 패키지 구성 화면의 EMS 리소스 탭에 있는 찾아보기 단추를 클릭하면 표시되는 목록(“선택한 EMS 리소스 설명”)이나 리소스 모니터와 함께 제공되는 설명서에서 리소스 폴링 간격을 볼 수 있습니다.

Resource start

패키지가 시작되기 전이나 후에 리소스를 시작해야 하는지 여부를 결정하는 리소스 속성입니다. ASCII 패키지 구성 파일에서는 이 매개 변수가 RESOURCE_START입니다.

기본 설정은 AUTOMATIC으로, 이 경우 노드가 클러스터에 참가하면 리소스가 시작됩니다. 사용 가능한 다른 설정은 DEFERRED로, 이 경우에는 리소스가 시작되기 전에 패키지 서비스가 시작됩니다. 리소스가 DEFERRED 시작으로 구성된 경우에는 해당 리소스 이름을 제어 스크립트의 DEFERRED_RESOURCE_NAME 매개 변수에 추가해야 합니다. Serviceguard Manger의 유도 모드를 사용하여 제어 스크립트를 만드는 경우에는 이 리소스 이름이 자동으로 입력됩니다.

Resource UP criteria

추가 패키지 리소스에 장애가 발생했는지 여부를 판단하는 기준입니다. ASCII 패키지 구성 파일에서는 이 매개 변수가 RESOURCE_UP_VALUE입니다. Resource Up Value는 Serviceguard Manger의 EMS 찾아보기 단추를 클릭하면 표시되는 “선택한 EMS 리소스 설명” 목록이나 리소스 모니터와 함께 제공되는 설명서에서 리소스 작동 값을 볼 수 있습니다.

패키지당 총 15개의 Resource Up Value를 구성할 수 있습니다. 예를 들어, 패키지에 리소스가 하나만 있으면 Resource Up Value를 15개까지 정의할 수 있습니다. 두 개의 Resource Name이 정의되어 있고 그 중 한 리소스에 10개의 Resource Up Value가 정의되어 있으면 다른 Resource Name은 5개의 Resource Up Value만 가질 수 있습니다.

Resource Up Value 문자열의 최대 길이는 1024자입니다.

Access control policies

해당 패키지에 대한 패키지 관리 액세스를 구성할 수 있습니다. 정책이 클러스터 구성 파일에 정의된 액세스 정책과 충돌하거나 중복되지 않는지 확인합니다. 자세한 내용은 182페이지의 “보안 파일 편집”을 참조하십시오.

패키지 구성 워크시트

다음 예제와 같이 패키지 구성 데이터를 각 패키지에 대해 별도의 워크시트로 정리하십시오.

Package Configuration File Data:

=====

Package Name: _____pkg11_____

Failover Policy: _____CONFIGURED_NODE_____

Failback Policy: _____AUTOMATIC_____

Primary Node: _____ftsys9_____

First Failover Node:_____ftsys10_____

Additional Failover Nodes:_____

Package Run Script: `___/etc/cmcluster/pkg1/control.sh__`Timeout: `__NO_TIMEOUT__`

Package Halt Script: `___/etc/cmcluster/pkg1/control.sh__`Timeout: `__NO_TIMEOUT__`

Package AutoRun Enabled? `__YES__` Local LAN Failover Allowed? `__YES__`

Node Failfast Enabled? `__NO__`

CVM Storage Groups:

Additional Package Resource:

Resource Name:_____ Polling Interval_____ Resource UP Value_____

Access Policies:

User: `__any_user__` From node: `__ftsys9__` Role: `__package_admin__`

User: `__lee ron admn__` From node: `__ftsys10__` Role: `__package_admin__`

패키지 제어 스크립트 변수

각 패키지와 함께 제공되는 제어 스크립트를 편집하여 변수 집합에 값을 할당해야 합니다. 다음 변수를 설정할 수 있습니다.

PATH

스크립트에서 사용할 경로를 지정합니다.

VGCHANGE

LVM 볼륨 그룹의 활성화 방법을 지정합니다. 볼륨 그룹을 단독 모드로 활성화하려면 기본값(`VGCHANGE="vgchange -a e"`)을 그대로 둡니다. 이렇게 하면 볼륨 그룹을 만들 때 `vgchange -c y`를 사용하여 볼륨 그룹을 초기화한 것으로 가정됩니다.

디스크가 별도의 실제 경로에 미러링된 경우에는 `VGCHANGE="vgchange -a e -q n"`를 사용합니다.

디스크가 별도의 실제 경로에 미러링된 경우 패키지 시작과 동시에 미리 재동기화가 시작되도록 하려면 `VGCHANGE="vgchange -a e -q n -s"`를 사용합니다.

단독 활성화 모드 이외의 모드를 사용하려면
VGCHANGE="vgchange -a y"를 사용합니다. 단일 노드 클러스터
구성에서는 단독 활성화를 사용하지 않아야 합니다.

CVM_ACTIVATION_CMD

VERITAS CVM 디스크 그룹의 활성화 명령을 지정합니다.

디스크 그룹을 단독 쓰기 모드로 활성화하려면 기본값인
CVM_ACTIVATION_CMD="vxdg -g \\${DiskGroup} set
activation=exclusivewrite"를 사용합니다.

디스크 그룹을 읽기 전용 모드로 활성화하려면
CVM_ACTIVATION_CMD="vxdg -g \\${DiskGroup} set
activation=readonly"를 사용합니다.

디스크 그룹을 공유 읽기 모드로 활성화하려면
CVM_ACTIVATION_CMD="vxdg -g \\${DiskGroup} set
activation=sharedread"를 사용합니다.

디스크 그룹을 공유 쓰기 모드로 활성화하려면
CVM_ACTIVATION_CMD="vxdg -g \\${DiskGroup} set
activation=sharedwrite"를 사용합니다.

주

VxVM 디스크 그룹에서는 특정 활성화 명령을 선택할 수 없습니
다. VxVM 디스크 그룹 활성화에는 항상 같은 명령이 사용됩니다.

VXVOL

미러링된 VxVM 볼륨에 대한 미러 복구 방법을 제어합니다.

복구가 완료될 때까지 패키지 제어 스크립트가 대기하도록 하려
면 기본값인 VXVOL="vxvol -g \\${DiskGroup} startall"을 사
용합니다.

패키지 시작과 동시에 미러 재동기화가 수행되도록 하려면
VXVOL="vxvol -g \\${DiskGroup} -o bg startall"을 사용합
니다.

Volume Groups

이 배열 매개 변수에는 패키지에서 활성화할 **LVM** 볼륨 그룹의 목록이 포함됩니다. 별도의 줄에 각 **VG**를 입력합니다.

CVM Disk Groups

이 배열 매개 변수에는 패키지에서 사용할 **VERITAS CVM** 디스크 그룹의 목록이 포함됩니다. 별도의 줄에 각 디스크 그룹을 입력합니다.

VxVM Disk Groups

이 배열 매개 변수에는 패키지에서 활성화할 **VERITAS VxVM** 디스크 그룹의 목록이 포함됩니다. 별도의 줄에 각 디스크 그룹을 입력합니다.

Logical Volumes, File Systems and Mount Options

이러한 배열 매개 변수는 배열 변수에 세 개씩 함께 입력됩니다. 각 패키지에서 사용되는 파일 시스템, 파일 시스템에 대한 논리 볼륨 및 마운트 옵션 문자열을 지정합니다. 패키지 제어 스크립트 파일에서는 이 변수가 **LV**, **FS** 및 **FS_MOUNT_OPT**입니다.

패키지가 시작되면 스크립트에서 하나 이상의 저장 장치 그룹을 활성화하고 논리 볼륨을 파일 시스템에 마운트합니다. 패키지가 중지될 때는 스크립트에서 파일 시스템의 마운트를 해제하고 각 저장 장치 그룹을 비활성화합니다. 모든 저장 장치 그룹은 각 대상 노드에 액세스할 수 있어야 합니다. **CVM** 디스크 그룹은 클러스터의 모든 노드에 액세스할 수 있어야 합니다. 각 파일 시스템(**FS**)에 대해 하나의 논리 볼륨(**LV**)을 지정해야 합니다.

논리 볼륨은 **LVM** 볼륨 그룹, **VERITAS CVM** 디스크 그룹 또는 **VERITAS VxVM** 디스크 그룹에서 만들 수 있습니다. **LV**는 사용되는 저장 장치 그룹 유형에 상관없이 임의의 순서로 입력할 수 있습니다.

Filesystem Unmount Count

패키지가 종료될 때 각 파일 시스템에 대해 허용되는 마운트 해제 횟수입니다. 기본값은 1입니다.

Filesystem Mount Retry Count

각 파일 시스템의 마운트 재시도 횟수입니다. 기본값은 0입니다. 패키지가 시작될 때 마운트 지점이 사용 중이고 FS_MOUNT_RETRY_COUNT가 0이면, 패키지 시작에 실패하고 스크립트는 코드 1과 함께 종료됩니다. 마운트 지점이 사용 중이고 FS_MOUNT_RETRY_COUNT가 0보다 크면 스크립트는 FS_MOUNT_RETRY_COUNT에 지정된 횟수만큼 사용 중인 마운트 지점에 대한 사용자를 강제 종료시키려고 시도합니다. 지정된 횟수만큼 재시도한 후에도 마운트에 실패하면 스크립트가 코드 1과 함께 종료됩니다.

CONCURRENT_VGCHANGE_OPERATIONS

패키지를 시작하거나 종료하는 동안 동시에 활성화하거나 비활성화할 수 있는 볼륨 그룹의 수를 지정합니다. 기본값은 1입니다. 많은 수의 볼륨 그룹을 사용하는 경우 이 변수를 더 높은 값으로 설정하면 성능이 향상될 수 있습니다. 1보다 작은 값을 지정하면 스크립트에서 변수를 기본값 1로 설정하고 패키지 제어 스크립트 로그 파일에 경고 메시지를 기록합니다.

CONCURRENT_DISKGROUP_OPERATIONS

패키지를 시작하거나 종료하는 동안 동시에 가져오거나 내보낼 수 있는 VxVM 디스크 그룹의 수를 지정합니다. 기본값은 1입니다. 많은 수의 디스크 그룹을 사용하는 경우 이 변수를 더 높은 값으로 설정하면 성능이 향상될 수 있습니다. 1보다 작은 값을 지정하면 스크립트에서 변수를 기본값 1로 설정하고 패키지 제어 스크립트 로그 파일에 경고 메시지를 기록합니다.

CONCURRENT_FSCK_OPERATIONS

패키지를 시작하는 동안 동시에 실행할 수 있는 fsck 명령의 수를 지정합니다. 기본값은 1입니다. 많은 수의 파일 시스템을 검사하는 경우 이 변수를 더 높은 값으로 설정하면 성능이 향상될 수 있습니다. 1보다 작은 값을 지정하면 스크립트에서 변수를 기본값 1로 설정하고 패키지 제어 스크립트 로그 파일에 경고 메시지를 기록합니다.

CONCURRENT_MOUNT_OPERATIONS

패키지를 시작하거나 종료하는 동안 수행할 수 있는 마운트 및 마운트 해제 작업의 수를 지정합니다. 기본값은 1입니다. 많은 수의 파일 시스템을 마운트하거나 마운트 해제하는 경우 이 변수를 더 높은 값으로 설정하면 성능이 향상될 수 있습니다. 1보다 작은 값을 지정하면 스크립트에서 변수를 기본값 1로 설정하고 패키지 제어 스크립트 로그 파일에 경고 메시지를 기록합니다.

IP Addresses 및 *SUBNET*

이 변수는 패키지를 LAN 카드에 매핑할 때 사용되는 IP 주소입니다. 인터페이스 카드에 추가할 IP 주소와 각 IP 주소의 서브넷을 지정합니다. 패키지 IP 주소는 IPv4 주소이거나 IPv6 주소일 수 있습니다. IPv6 주소 형식에 대한 자세한 내용은 부록, 424페이지의 “IPv6 주소 유형”을 참조하십시오.

패키지 제어 스크립트 파일에서 이 변수는 쌍으로 입력됩니다.

예를 들어, IPv4에서는 IP[0]=192.10.25.12 및

SUBNET[0]=192.10.25.0과 같이 입력합니다. 이 경우 서브넷 마스크는 255.255.255.0입니다. IPv6에서는 IP[0]=2001::3 및

SUBNET[0]=2001::/64와 같이 입력합니다. 이 경우 서브넷 마스크는 ffff:ffff:ffff:ffff::입니다.

Service Name

패키지 내의 특정 서비스 각각에 대한 고유한 이름을 입력합니다. 모든 서비스는 Serviceguard에서 모니터링합니다. 패키지당 최대 30개의 서비스를 지정할 수 있습니다. 각 이름은 클러스터 내에서 고유해야 합니다. 서비스 이름은 패키지 제어 스크립트에서 cmrunserv 및 cmhaltserv에 사용되는 이름입니다. 이 이름은 패키지 ASCII 구성 파일에서 지정된 이름과 동일해야 합니다.

패키지 제어 스크립트 파일에서는 SERVICE_NAME이라는 배열에 값을 입력합니다. 서비스마다 하나의 서비스 이름을 입력합니다. SERVICE_NAME, SERVICE_CMD, SERVICE_RESTART 매개 변수는 패키지 제어 스크립트에서 세 개가 한 그룹으로 설정됩니다.

서비스 이름에는 공백, 슬래시(/), 백슬래시(\), 별표(*)를 사용할 수 없습니다. 다른 문자는 모두 사용할 수 있습니다. 서비스 이름은 39자까지 가능합니다.

Service Command

이름을 지정한 각각의 서비스에 대해 **Service Command**를 입력합니다. 이 명령은 제어 스크립트의 `cmrunserv` 명령을 통해 실행됩니다.

패키지 제어 스크립트 파일에서는 `SERVICE_CMD`라는 배열에 값을 입력합니다. 서비스마다 하나의 서비스 명령을 입력합니다. `SERVICE_NAME`, `SERVICE_CMD`, `SERVICE_RESTART` 매개 변수는 패키지 제어 스크립트에서 세 개가 한 그룹으로 설정됩니다.

Service Restart Parameter

재시작 횟수를 입력합니다. 이 매개 변수의 유효한 형식 중 하나는 “-r n”입니다. 여기서 n은 재시도 횟수입니다. “-r 0” 값은 재시도가 없다는 것을 의미합니다. “-R” 값은 계속해서 재시도한다는 것을 의미합니다. 기본값은 0으로, 재시작을 하지 않습니다.

패키지 제어 스크립트 파일에서는 `SERVICE_RESTART`라는 배열에 값을 입력합니다. 서비스마다 하나의 재시작 값을 입력합니다. `SERVICE_NAME`, `SERVICE_CMD`, `SERVICE_RESTART` 매개 변수는 패키지 제어 스크립트에서 세 개가 한 그룹으로 설정됩니다.

Deferred Resource Names

패키지 구성 **ASCII** 파일에서 지정한 각 지연 시작 리소스에 대해 제어 스크립트의 이 배열에 리소스 이름을 입력해야 합니다. 이 이름은 패키지 **ASCII** 구성 파일의 `RESOURCE_NAME` 매개 변수와 정확하게 일치해야 합니다.

패키지 제어 스크립트 파일에서는 `DEFERRED_RESOURCE_NAME`이라는 배열에 값을 입력합니다. 이 이름은 패키지 **ASCII** 구성 파일에서 `RESOURCE_START` 매개 변수로 나열된 리소스 이름과 일치해야 합니다.

DTC Manager Data

각 DTC의 DTC 이름을 입력합니다. Serviceguard에서 DTC를 사용하는 방법에 대한 자세한 내용은 *Using the HP DTC Manager/UX* 설명서의 “Configuring DTC Manager for Operation with Serviceguard” 장을 참조하십시오.

패키지 제어 스크립트는 환경을 정리하고 오류가 발행할 때 작업을 취소합니다. 자세한 내용은 3장의 “패키지 제어 스크립트의 작동 방식” 절을 참조하십시오.

제어 스크립트 워크시트

다음 예제와 같이 패키지 제어 스크립트 데이터를 각 패키지에 대해 별도의 워크시트로 정리하십시오.

LVM Volume Groups:

VG[0] _____ VG[1] _____ VG[2] _____

VGCHANGE: _____

CVM Disk Groups:

CVM_DG[0] ___/dev/vx/dg01___ CVM_DG[1] _____ CVM_DG[2] _____

CVM_ACTIVATION_CMD: _____

VxVM Disk Groups:

VXVM_DG[0] ___/dev/vx/dg01___ VXVM_DG[1] _____ VXVM_DG[2] _____

=====

Logical Volumes and File Systems:

LV[0] ___/dev/vg01/lv011___ FS[0] ___/mnt1___ FS_MOUNT_OPT[0] _____

LV[1] _____ FS[1] _____ FS_MOUNT_OPT[1] _____

LV[2] _____ FS[2] _____ FS_MOUNT_OPT[2] _____

FS Umount Count: _____ FS Mount Retry Count: _____

CONCURRENT VCHANGE OPERATIONS: _____ -

CONCURRENT DISKGROUP OPERATIONS: _____ -

CONCURRENT MOUNT/UMOUNT OPERATIONS: _____

CONCURRENT FSCK OPERATIONS: _____

=====
Network Information:

IP[0] ____15.13.171.14____ SUBNET ____15.13.168_____

IP[1] _____ SUBNET _____

=====
Service Name: __svc1____ Command: ___/usr/bin/MySvc -f__ Restart: __-r 2__

Service Name: _____ Command: _____ Restart: __

Deferred Resource Name _____

5 HA 클러스터 구성 만들기

5장과 6장에서는 **Serviceguard** 클러스터를 설정하는 데 필요한 구성 작업들을 수행합니다. 이 작업들은 **구성 노드**라고 하는 하나의 노드에서 수행되고 결과로 생기는 이진 파일이 **Serviceguard**에 의해 클러스터의 모든 노드에 배포됩니다. 앞으로 다루게 될 예에서는 구성 노드를 *ftsyst9*라고 하고 샘플 대상 노드는 *ftsyst10*이라고 합니다. 이 장에서는 다음 **클러스터 구성** 작업을 설명합니다.

- 시스템 준비
- 쿼럼 서버 설치
- **Serviceguard** 설치
- LVM을 사용하여 저장 장치 하부 구조 만들기
- VxVM을 사용하여 저장 장치 하부 구조 만들기
- 클러스터 구성
- CVM을 사용하여 저장 장치 하부 구조 만들기
- 실행 중인 클러스터 관리

구성 패키지는 다음 장에서 설명합니다.

Serviceguard 버전 11.16 이상에서는 그래픽 사용자 인터페이스인 **Serviceguard Manager**를 사용하여 클러스터를 구성할 수 있습니다. 클러스터를 만들려면 **Serviceguard A.11.16** 이상이 설치된 세션 서버에 연결하고, 맵이나 트리에서 **Serviceguard A.11.16** 이상이 설치된 클러스터 또는 미사용 노드를 선택한 다음 동작 메뉴로 이동합니다. 그러면 여러 개의 탭이 있는 화면이 표시됩니다. 각 탭에서는 관련 정보를 표시하고 여러 가지 선택 사항에 대해 발견된 옵션을 표시합니다. 필요한 사항을 결정하는 데 유용한 온라인 도움말도 볼 수 있습니다.

Serviceguard 명령을 사용하여 클러스터와 패키지를 구성하는 경우 각 명령의 맨 페이지를 사용하여 구문 및 사용법에 대한 정보를 얻을 수 있습니다.

시스템 준비

클러스터를 구성하기 전에 모든 클러스터 노드에 적절한 보안 파일, 커널 구성, NTP (네트워크 시간 프로토콜) 구성이 있는지 확인합니다.

파일 위치 이해

Serviceguard는 특수한 파일인 /etc/cmcluster.conf를 사용하여 HP-UX 파일 시스템 내에서 구성 및 로그 파일의 위치를 정의합니다. 파일에서 다음과 같이 위치가 정의됩니다.

```
##### cmcluster.conf#####  
#  
# Highly Available Cluster file locations  
#  
# This file must not be edited  
#####  
  
SGCONF=/etc/cmcluster  
SGSBIN=/usr/sbin  
SGLBIN=/usr/sbin  
SGLIB=/usr/lib  
SGCMOM=/opt/cmom  
SGRUN=/var/adm/cmcluster  
SGAUTOSTART=/etc/rc.config.d/cmcluster  
SGCMOMLOG=/var/adm/syslog/cmom
```

주

이러한 변수가 시스템에 정의되지 않은 경우 **root** 사용자에게 대한 로그인 프로필에 /etc/cmcluster.conf 파일을 포함시킵니다.

이 설명서 전체에서 시스템 파일 이름은 일반적으로 이러한 위치 접두사 중 하나가 붙습니다. 따라서 이 파일에서 찾을 수 있는 접두사 정의를 사용하여 \$SGCONF/<FileName>을 참조할 수 있습니다. 예를 들어, SGCONF가 /etc/cmcluster/conf로 정의되면 \$SGCONF/cmclconfig 파일에 대한 전체 경로 이름은 /etc/cmcluster/conf/cmclconfig가 됩니다.

주

/etc/cmcluster.conf 구성 파일은 편집하지 **마십시오**.

보안 파일 편집

Serviceguard 데몬은 수신되는 호스트 이름 및 사용자 이름을 정의되어 있는 액세스 제어 정책과 비교하여 명령에 대한 액세스 권한을 부여합니다. 관리자가 이러한 정책을 올바르게 구성하는 방법을 이해하기 위해서는 Serviceguard가 호스트 이름, IP 주소, 사용자 이름 및 관련 구성 파일을 처리하는 방법을 이해해야 합니다.

Serviceguard는 중복을 위해 사용 가능한 모든 IPv4 네트워크를 통신에 활용합니다. Serviceguard 노드가 해당 인터페이스에 있는 다른 노드와 통신할 수 있는 경우 액세스 제어 정책은 해당 인터페이스의 기본 IP 주소를 포함해야 합니다.

IP 주소 확인

Serviceguard에 대한 액세스 제어 정책은 이름을 기반으로 합니다. 수신 연결에 대한 IP 주소는 액세스 제어 정책과 비교하기 위해 호스트 이름으로 확인해야 합니다.

두 Serviceguard 노드 사이의 통신은 공유 네트워크를 통해 받을 수 있으므로 그러한 각 네트워크에 있는 모든 기본 주소가 식별되어야 합니다.

Serviceguard는 별명 사용을 지원합니다. IP 주소는 여러 호스트 이름으로 확인될 수 있는데 그 중 하나는 액세스 제어 정책에 정의된 이름과 일치해야 합니다.

IP 주소 확인 구성

Serviceguard는 이름 확인 서비스에 구축된 운영 체제를 사용합니다. 클러스터의 원활한 작동을 위해서는 이름 확인을 DNS나 NIS 서비스에 의존하는 것보다는 먼저 노드의 /etc/hosts 파일에 정의하는 것이 좋습니다.

예를 들어, 두 개의 개인 서버넷과 한 개의 공용 서버넷이 있는 2노드 클러스터(gryf 및 sly)의 경우 개인 서버넷을 공유하지 않는 비 클러스터 노드(비트)에 대해 사용 권한을 부여하게 됩니다. 양쪽 클러스터 노드에 있는 /etc/hosts 파일은 다음을 포함해야 합니다.

```
15.145.162.131 gryf.uksr.hp.com gryf
10.8.0.131 gryf.uksr.hp.comgryf
10.8.1.131 gryf.uksr.hp.comgryf
15.145.162.132 sly.uksr.hp.comsly
10.8.0.132 sly.uksr.hp.com sly
```

```
10.8.1.132 sly.uksr.hp.com sly
15.145.162.150 bit.uksr.hp.com bit
```

주

정규화된 도메인 이름(FQDN)을 사용하는 경우 **Serviceguard**는 호스트 이름 부분만 인식하게 됩니다. 예를 들어, 두 개의 노드 **gryf.uksr.hp.com**과 **gryf.cup.hp.com**은 둘 다 같은 호스트 **gryf**로 취급되므로 같은 클러스터에 있을 수 없습니다.

Serviceguard는 도메인 이름 별명도 지원합니다. 다른 응용 프로그램에서 고유한 기본 호스트 이름을 갖는 다른 인터페이스가 필요한 경우 **Serviceguard** 호스트 이름은 별명 중 하나가 될 수 있습니다. 예를 들면 다음과 같습니다.

```
15.145.162.131 gryf.uksr.hp.com gryf node1
10.8.0.131 gryf.uksr.hp.com gryf
10.8.1.131 gryf.uksr.hp.com gryf
15.145.162.132 sly.uksr.hp.com sly node2
10.8.0.132 sly.uksr.hp.com sly
10.8.1.132 sly.uksr.hp.com sly
```

이 구성에서 개인 서브넷의 기본 이름은 고유합니다. 별명을 제공함으로써 **Serviceguard**는 여전히 이 IP 주소를 적절한 노드에 연결하여 액세스 제어 정책에서 일치시킬 수 있습니다.

DNS, NIS 또는 LDAP와 같은 기타 소스보다 먼저 /etc/hosts 파일을 확인하도록 이름 서비스 전환 정책을 구성해야 합니다. 모든 클러스터 노드의 /etc/nsswitch.conf 파일에 먼저 '파일'이 나열된 다음 다른 서비스가 나열되는지 확인합니다. 예를 들면 다음과 같습니다.

DNS의 경우 다음(한 줄)을 입력합니다.

```
hosts: files [NOTFOUND=continue UNAVAIL=continue] dns
[NOTFOUND=return UNAVAIL=return]
```

NIS의 경우 다음(한 줄)을 입력합니다.

```
hosts: files [NOTFOUND=continue UNAVAIL=continue] nis
[NOTFOUND=return UNAVAIL=return]
```

사용자 이름 검증

Serviceguard는 클라이언트 노드의 `ident` 서비스에 의존하여 수신 네트워크 연결의 사용자 이름을 확인합니다. Serviceguard 데몬이 클라이언트의 `ident` 데몬에 연결할 수 없는 경우 사용 권한이 거부됩니다.

노드의 루트는 UID가 0인 사용자로 정의됩니다. 특정 사용자를 원격 시스템의 루트로 식별하려면 로컬 시스템에 대한 `/etc/passwd`의 "root" 사용자 항목이 UID 0일 수도 있는 다른 사용자보다 먼저 와야 합니다. `ident` 데몬은 처음으로 일치하는 UID에 대한 사용자 이름을 반환합니다. Serviceguard가 원격 사용자를 원격 노드의 루트 사용자로 간주하게 하려면 `ident` 서비스가 사용자 이름을 "root"로 반환해야 합니다.

Serviceguard가 `ident` 서비스를 사용하지 않게 구성할 수 있지만 이러한 구성은 좋지 않습니다. 자세한 내용은 백서의 "Securing Serviceguard"를 참조하십시오.

`identd` 사용을 비활성화하려면 `-i` 옵션을 `tcp hacl-cfg` 및 `hacl-probe inetd` 구성에 추가하십시오.

예를 들어, Serviceguard A.11.16을 사용하는 HP-UX의 경우

1. `/etc/inetd.conf`의 `cmclconfd` 항목이 `hacl-cfg stream tcp nowait root /usr/sbin/cmclconfd \ cmclconfd -c -i`로 표시되게 변경합니다.
2. `/etc/inetd.conf`의 `cmomd` 항목이 `hacl-probe stream tcp nowait root \ /opt/cmom/sbin/cmomd /opt/cmom/sbin/cmomd -i -f \ /var/opt/cmom/cmomd.log -r /var/opt/cmom`으로 표시되게 변경합니다.
3. `inetd: /etc/init.d/inetd restart`를 다시 시작합니다.

액세스 역할

Serviceguard에는 다음 두 가지 액세스 수준이 있습니다.

- 루트 액세스: 루트 액세스 권한이 부여된 사용자는 클러스터 및 패키지의 구성을 완전히 제어합니다.
- 루트 권한 없는 액세스: 루트 권한 없는 사용자는 다음 네 가지 역할 중 하나가 지정될 수 있습니다.
 - 모니터: 이러한 사용자는 클러스터 및 해당 패키지에 대해 읽기 전용 권한을 갖습니다. 명령줄 사용자는 `cmviewcl`, `cmquerycl`, `cmgetconf` 및 `cmviewconf` 명령을 실행할 수 있습니다. **Serviceguard Manager** 사용자는 맵, 트리 및 속성에서 상태 및 구성 정보를 볼 수 있습니다.
 - (한) 패키지 관리: 특정 패키지에 대해서만 적용됩니다. 명령줄에서 이러한 사용자는 지정된 패키지에 대해 `cmrunpkg`, `cmhaltpkg`, `cmmodnet`, `cmrunserv`, `cmhaltserv`, `cmstartres`, `cmstopres` 및 `cmmodpkg` 명령을 실행할 수 있습니다. **Serviceguard Manager** 사용자는 특정 패키지에 대한 Admin 메뉴 옵션인 Run Package, Halt Package, Move Package 및 Enable 또는 Disable Switching을 볼 수 있습니다. 패키지 관리에서는 패키지를 구성하거나 만들 수 없습니다. 패키지 관리에는 모니터 역할에 대한 권한이 포함됩니다.
 - (모든) 패키지 관리: 클러스터의 모든 패키지에 적용됩니다. 명령은 위의 역할과 같습니다. 패키지 관리에는 모니터 역할에 대한 권한이 포함됩니다.
 - 전체 관리: 이러한 사용자는 클러스터를 관리할 수 있습니다. 명령줄에서 이러한 사용자는 자신의 클러스터에서 `cmruncl`, `cmhaltcl`, `cmrunnode` 및 `cmhaltnode` 명령을 실행할 수 있습니다. 전체 관리에서는 클러스터를 구성하거나 만들 수 없습니다. **Serviceguard Manager**에서 전체 관리 사용자는 자신의 클러스터에 있는 자신의 클러스터와 패키지에 대한 Admin 메뉴를 볼 수 있습니다. 전체 관리에는 패키지 관리 역할에 대한 권한이 포함됩니다.

클러스터를 버전 11.16으로 업그레이드하면 `cmclnodelist` 항목이 클러스터 구성 파일에서 자동으로 액세스 제어 정책으로 업그레이드됩니다. 모든 루트 권한 없는 사용자와 호스트 이름 쌍에 모니터 역할(보기 전용) 역할이 부여됩니다.

액세스 제어 정책을 설정할 때 노드의 상태에 따라 다른 메커니즘이 사용됩니다. 클러스터에 구성되지 않은 노드는 클러스터에 있는 노드와는 다른 보안 구성을 사용합니다. 다음 두 절에서는 이러한 액세스 제어 정책을 구성하는 방법을 설명합니다.

구성되지 않은 노드에 대한 제어 설정

Serviceguard 액세스 제어 정책은 원격 노드가 로컬 노드에 할 수 있는 작업을 정의합니다. 새로 **Serviceguard**를 설치한 경우에는 액세스 제어 정책이 정의되어 있지 않습니다. 이 노드가 클러스터에 포함되게 하려면 다른 잠재적 클러스터 노드의 루트에 대한 액세스를 허용하도록 정책이 정의되어야 합니다. **Serviceguard Manager**의 경우 원격 **COM** 서버가 노드를 모니터링하거나 구성할 수 있게 정책이 정의되어야 합니다. 이러한 정책은 노드가 클러스터에 구성되어 있지 않은 동안에만 적용됩니다.

구성되지 않은 노드는 원격 사용자에게 두 가지 액세스 수준(루트 및 루트 권한 없는 수준)의 권한을 부여합니다. 루트 권한의 사용자는 모든 클러스터 구성 명령을 사용할 수 있습니다. 루트 권한이 없는 사용자에게는 노드 구성에 대해 읽기 전용 액세스만 부여하는 모니터 역할이 할당됩니다.

Serviceguard 노드가 클러스터에 구성되어 있지 않은 경우 원격 사용자의 권한을 부여할 때 두 가지 가능한 보안 메커니즘 중 하나에 의존합니다.

- `$SGCONF/cmclnodelist` 파일이 있는 경우 **Serviceguard**는 그 파일의 내용을 사용하여 원격 사용자의 권한을 부여합니다.
- 호스트 동일성 파일은 `r-commands`, `~/.rhosts` 및 `/etc/hosts.equiv` (`hostsequiv`)에서 사용되었습니다.

`cmclnodelist`를 사용하는 것이 가장 좋습니다.

Serviceguard는 `hostsequiv`에 액세스하기 전에 `$SGCONF/cmclnodelist`가 있는지 확인합니다. 이 파일이 있으면 **Serviceguard**는 다른 권한 부여 메커니즘을 확인하지 않습니다. **Serviceguard**에서는 `cmclnodelist`를 사용하든 `hostsequiv`를 사용하든 동일한 보안 수준을 제공합니다. 관리자는 `r-command` 액세스를 제한하려는 설치에서 `hostsequiv` 대신 `cmclnodelist` 파일 사용을 선택할 수 있습니다.

구성되지 않은 상태의 노드는 이전 버전과의 호환성을 위해 IP 주소를 기반으로 액세스 제어 정책을 정의할 수 있습니다. 위에서 지정한 대로 이름 서비스가 구성되어 있지 않은 경우 각 인터페이스 **Serviceguard**가 통신에 사용하는 기본 IP 주소는 고유 정책을 갖고 있어야 합니다. 일단 노드가 클러스터에 구성되면 이러한 정책에 대해 IP 주소를 더 이상 사용할 수 없습니다.

cmclnodelist 파일 사용

새로 설치할 때 cmclnodelist 파일은 기본적으로 만들어지지 않습니다. 이러한 "부트스트랩" 파일을 만들려면 관리자가 다음과 같은 주석을 추가해야 합니다.

```
#####
# Do Not Edit This File
# This is only a temporary file to bootstrap an unconfigured
# node with Serviceguard version A.11.16
# Once a cluster is created, Serviceguard will not consult
# this file.
#####
```

cmclnodelist 파일에 있는 항목의 형식은 다음과 같습니다.

```
[hostname or ip address] [user] [#Comment]
```

예를 들면 다음과 같습니다.

표 5-1 cmclnodelist 예제

| | | |
|------|-------|---------------------------------|
| gryf | root | # Cluster 1,Node 1 |
| gryf | user1 | # Cluster 1, Node 1 |
| sly | root | # Cluster 1, Node 2 |
| sly | user1 | # Cluster 1, Node 2 |
| bit | root | # Administration /COM Server |

이 예제에서 gryf, sly 및 bit 노드에 있는 루트는 모두 이 파일이 있는 노드에 대해 루트 액세스 권한을 갖습니다. 루트 권한이 없는 사용자 "user1"은 gryf 및 sly 노드에서 모니터 역할을 가집니다.

또한 Serviceguard는 cmclnodelist 파일에서 "+" 사용을 허용하며 이는 특정 노드에서 특정 루트 사용자가 이 노드를 구성할 수 있고 특정 루트 권한 없는 사용자는 모니터 역할을 갖는다는 것을 나타냅니다.

동일한 호스트 사용

hostsequiv를 사용하려는 설치의 경우 클러스터의 각 노드에 대한 기본 IP 주소나 호스트 이름에 권한을 부여해야 합니다. hostsequiv 사용에 대한 자세한 내용은 hosts.equiv(4) 또는 HP-UX 설명서 "시스템 및 작업 그룹 관리"를 참조하십시오.

hostsequiv에서는 루트와 동일한 모든 노드에서 모든 사용자를 정의할 수 있지만 Serviceguard는 원격 노드에서 루트가 아닌 사용자에 대해 루트 액세스 권한을 부여하지 않습니다. 이러한 구성은 해당 사용자에게 "루트 권한 없음" 액세스를 부여합니다.

이름 확인 서비스 정의

Serviceguard가 이름 확인 서비스를 사용하는 방식을 이해하는 것은 중요합니다. 모든 사용자 수준 Serviceguard 명령(cmviewcl 포함)을 사용하는 경우 해당 명령은 이름 조회를 사용하여 클러스터 노드 전체의 주소를 구합니다. 이름 확인 서비스를 사용할 수 없으면, 명령이 실행 중에 멈추거나 예기치 못한 네트워크 오류 메시지를 반환합니다. Serviceguard Manager에서 클러스터 또는 패키지 작업도 이름 확인 서비스를 사용할 수 없을 때는 오류를 반환합니다.

주

이러한 중단이나 오류가 발생하는 경우 사용자 명령은 실행되지 못하지만 Serviceguard 및 보호되는 모든 응용 프로그램은 계속해서 작동합니다. 즉, Serviceguard 구성 명령과 Serviceguard Manager 기능에만 영향을 미치며 클러스터 데몬이나 패키지 서비스에는 영향을 미치지 않습니다.

이 문제를 피하려면 DNS 또는 NIS를 비롯한 모든 클러스터 노드의 /etc/hosts 파일을 사용할 수 있습니다. 또는 복수의 DNS 서버를 사용하거나 DNS를 Serviceguard 패키지로 구성하는 방법 중 하나를 사용하여 DNS의 가용성을 높이는 것이 좋습니다.

그러기 위해선 /etc/nsswitch.conf 파일에 다음 줄 중 하나를 추가합니다.

- DNS의 경우 다음(한 줄)을 입력합니다.

```
hosts: dns [NOTFOUND=continue UNAVAIL=contine] dns  
{NOTFOUND=return UNAVAIL=return}
```

- NIS의 경우 다음(한 줄)을 입력합니다.

```
hosts: nis [NOTFOUND=continue UNAVAIL=contine] nis  
{NOTFOUND=return UNAVAIL=return}
```

다른 조회 방법을 사용할 수 없을 경우 `/etc/hosts` 파일을 검색하도록 `/etc/nsswitch.conf` 파일을 구성하면 일반적인 이름 조회 방법을 계속 사용할 수 있습니다. 그러면 이름 확인 서비스를 사용할 수 없을 때 `Serviceguard` 명령이 로컬 시스템의 `/etc/hosts` 파일을 사용하여 이름 확인 작업을 수행합니다. 물론 클러스터에 있는 모든 노드의 이름과 IP 주소는 `/etc/hosts` 파일에 있어야 합니다.

기본 LAN 장애 또는 DNS 손실 후 이름 확인

일부 특수한 구성 단계에서는 `cmrunnode` 및 `cmrunc1`과 같은 클러스터 구성 명령을 허용하여 LAN 장애 후 장애가 있는 기본 LAN에 대해 대기 LAN이 구성된 경우에도 제대로 작동하도록 해야 합니다. 또한 이러한 단계에서는 DNS 서비스가 손실된 후에 또 클러스터 노드가 다른 클러스터 노드와 통신하도록 보호해야 합니다.

1. 클러스터의 모든 노드에서 `/etc/hosts` 파일을 편집합니다. 모든 하트비트 IP 주소 및 모든 클러스터 노드의 다른 IP 주소에 대한 이름 확인을 추가합니다. 예제:

```
15.13.172.231 hasupt01
192.2.1.1 hasupt01
192.2.8.2 hasupt01
15.13.172.232 hasupt02
192.2.1.2 hasupt02
192.2.8.2 hasupt02
15.13.172.233 hasupt03
192.2.1.3 hasupt03
192.2.8.3 sgsupt03
```

이렇게 하면 공용 네트워크 뿐만 아니라 공용이 아닌 네트워크에서 수신되는 메시지도 올바른 호스트 이름에 매핑됩니다.

주

각 클러스터 노드에 대해 공용 네트워크 IP 주소가 처음에 나열되는 주소가 되어야 다른 응용 프로그램에서 공용 네트워크의 다른 노드와 통신할 수 있습니다.

2. `/etc/nsswitch.conf` 파일이 아직 없으면 모든 노드에서 해당 파일을 편집하거나 만들고 다음 행을 추가합니다.

```
hosts:          files [NOTFOUND=continue] dns
```

"hosts:" 문자열로 시작하는 행이 이미 있는 경우 다음 텍스트가 해당 문자열의 오른쪽에 바로와야 합니다.

```
files [NOTFOUND=continue] dns
```

이 단계는 DNS가 종료되거나 기본 LAN이 종료된 동안 클러스터의 노드가 IP 주소에 대한 호스트 이름을 확인하는 데 결정적인 역할을 합니다.

3. 클러스터가 노드에 없는 경우 모든 노드에서 /etc/cmclnodelist 파일을 만들거나 편집하여 모든 클러스터 노드 기본 IP 주소 및 노드 이름에 대한 액세스를 추가합니다.

```
15.13.172.231      hasupt01
15.13.172.232      hasupt02
15.13.172.233      hasupt03
```

루트 논리 볼륨의 미리 만들기

모든 클러스터 노드에서 미러링되는 루트 볼륨을 반드시 사용하는 것이 좋습니다. 다음 과정에서는 별도의 부트 및 루트 볼륨을 사용하고 있다고 가정하고 부트 볼륨(/dev/vg00/lvol1), 주 스왑(/dev/vg00/lvol2) 및 루트 볼륨(/dev/vg00/lvol3)의 미리를 만듭니다. 이 예제와 다음 명령에서 /dev/dsk/c4t5d0은 주 디스크이고 /dev/dsk/c4t6d0은 미러입니다. 시스템의 루트 디스크에 올바른 장치 파일 이름을 사용해야 합니다.

1. 미리에 사용할 부트 가능 LVM 디스크를 만듭니다.

```
# pvcreate -B /dev/rdisk/c4t6d0
```

2. 이 디스크를 현재 루트 볼륨 그룹에 추가합니다.

```
# vgextend /dev/vg00 /dev/dsk/c4t6d0
```

3. 새 디스크를 부트 디스크로 만듭니다.

```
# mkboot -l /dev/rdisk/c4t6d0
```

4. 부트, 주 스왑 및 루트 논리 볼륨을 새로운 부팅 가능 디스크로 미러링합니다.
/usr, /swap 같은 vg00의 모든 장치가 미러링되었는지 확인합니다

주

부트 볼륨이 새 디스크에서 연속적 확장 세트의 맨 앞에 위치하고 스왑 볼륨과 루트 볼륨이 그 다음에 위치하도록, 부트, 루트 및 스왑 논리 볼륨을 반드시 다음 순서대로 미리링해야 합니다.

다음은 부트 논리 볼륨의 미리링 예제입니다.

```
# lvextend -m 1 /dev/vg00/lvol1 /dev/dsk/c4t6d0
```

다음은 주 스왑 논리 볼륨의 미리링 예제입니다.

```
# lvextend -m 1 /dev/vg00/lvol2 /dev/dsk/c4t6d0
```

다음은 루트 논리 볼륨의 미리링 예제입니다.

```
# lvextend -m 1 /dev/vg00/lvol3 /dev/dsk/c4t6d0
```

- 부트, 루트 및 주 스왑의 미러 사본에 대해 BDRA에 포함된 부트 정보를 업데이트 합니다.

```
# /usr/sbin/lvlnboot -b /dev/vg00/lvol1
```

```
# /usr/sbin/lvlnboot -s /dev/vg00/lvol2
```

```
# /usr/sbin/lvlnboot -r /dev/vg00/lvol3
```

- 미러가 제대로 만들어졌는지 확인합니다.

```
# lvlnboot -v
```

이 명령의 출력은 다음과 같이 표시됩니다.

```
Boot Definitions for Volume Group /dev/vg00:
Physical Volumes belonging in Root Volume Group:
    /dev/dsk/c4t5d0 (10/0.5.0) -- Boot Disk
    /dev/dsk/c4t6d0 (10/0.6.0) -- Boot Disk
Boot:  lvol1    on:      /dev/dsk/c4t5d0
      lvol1    on:      /dev/dsk/c4t6d0
Root:  lvol3    on:      /dev/dsk/c4t5d0
      lvol3    on:      /dev/dsk/c4t6d0
Swap:  lvol2    on:      /dev/dsk/c4t5d0
      lvol2    on:      /dev/dsk/c4t6d0
Dump:  lvol2    on:      /dev/dsk/c4t6d0, 0
```

클러스터 잠금 디스크 선택

다음 지침은 잠금 디스크를 사용하는 경우에 적용됩니다. 모든 클러스터 노드에 물리적으로 연결되어 있는 볼륨 그룹에 클러스터 잠금 디스크를 구성할 수 있습니다. 이 볼륨 그룹에는 패키지에서 사용하는 데이터가 포함될 수도 있습니다.

이중 클러스터 잠금 디스크를 사용하는 경우에는 클러스터 잠금 물리 볼륨에 기본 입출력 시간 종료 값을 사용해야 합니다. 클러스터 잠금 물리 볼륨에 대한 입출력 시간 종료 값을 변경할 경우, 장애가 발생한 잠금 디스크를 클러스터 노드에서 할당된 시간 안에 감지하지 못해서 클러스터 재편성이 일어나지 못하게 될 수 있습니다. 기존 입출력 시간 종료 값을 보려면 다음 명령을 실행합니다.

```
# pvdisplay <lock device file name>
```

이 입출력 시간 종료 값은 “기본값”이어야 합니다. 입출력 시간 종료 값을 다시 기본값으로 설정하려면 다음 명령을 실행합니다.

```
# pvchange -t 0 <lock device file name>
```

이중 클러스터 잠금은 특정 하드웨어 구성에서만 사용할 수 있습니다. 자세한 내용은 3장의 “이중 클러스터 잠금” 절을 참조하십시오.

클러스터 잠금 디스크 정보 백업

클러스터를 구성하고 클러스터 잠금 볼륨 그룹과 물리 볼륨을 만든 후에는 각 잠금 볼륨 그룹의 볼륨 그룹 구성 데이터를 백업해야 합니다. 구성된 각 잠금 볼륨 그룹에 대해 `vgcfbackup` 명령을 사용하여 백업하고, 디스크 장애 발생 시 `vgcfrestore` 명령을 사용하여 잠금 구성을 새 디스크로 복원할 수 있도록 이 백업 파일을 보관해야 합니다.

주

잠금 볼륨 그룹을 만든 방식에 상관없이 잠금 볼륨 그룹 구성 데이터를 백업하고 복원하려면 `vgcfbackup` 및 `vgcfrestore` 명령을 사용해야 합니다.

커널 구성의 일관성 유지

모든 클러스터 노드의 커널 구성이 장애 조치 중의 예상 클러스터 동작과 일치하는지 확인해야 합니다. 특히, 한 클러스터 노드에서 커널 매개 변수를 변경하면 같은 패키지를 실행할 수 있는 다른 클러스터 노드에서도 해당 매개 변수를 변경해야 합니다.

네트워크 시간 프로토콜 사용

클러스터의 각 노드에서 NTP(네트워크 시간 프로토콜) 서비스를 사용할 수 있도록 하는 것이 좋습니다. 각 시스템에서 데몬 프로세스로 실행되는 NTP를 사용하면 모든 노드에서 시스템 시간이 일치하므로 로그 파일의 타임스탬프와 메시지 서비스의 동작도 일관되게 됩니다. 따라서 클러스터에서 실행되는 응용 프로그램이 올바르게 동기화됩니다. 클러스터를 구성하려면 먼저 NTP 서비스 데몬인 `xntpd`가 모든 노드에서 실행되어야 합니다. NTP 구성 파일은 `/etc/ntp.conf`입니다.

NTP 서비스 구성에 대한 자세한 내용은 HP-UX 설명서인 *Installation and Administration of Internet Services*에서 “Configuring NTP” 장을 참조하십시오.

네트워크 및 커널 매개 변수 조정

Serviceguard와 SGeSAP, SGeRAC 및 SGeFF 같은 Serviceguard 확장 제품은 `ndd` 및 `kmtune` 유틸리티에서 지원되는 네트워크 및 매개 변수의 기본값으로 테스트했습니다.

이 매개 변수를 조정할 때는 주의하십시오.

문제가 발생하면 매개 변수를 다시 기본값으로 되돌리십시오. Serviceguard 및 네트워크 관련 문제에 대해 HP 지원 센터에 문의할 때는 기본값에서 변경한 매개 변수에 대한 모든 정보를 알려 주십시오.

Serviceguard 환경에서 실행되는 타사 응용 프로그램은 네트워크 및 커널 매개 변수를 조정해야 할 수도 있습니다.

- `ndd`는 네트워크 조정 유틸리티입니다. 자세한 내용은 `ndd(1M)`에 대한 맨페이지를 참조하십시오.
- `kmtune`은 시스템 조정 유틸리티입니다. 자세한 내용은 `kmtune(1M)`에 대한 맨페이지를 참조하십시오.

또한 Serviceguard는 다음 두 개의 네트워크 매개 변수에 대해 기본값 이외의 값으로도 테스트했습니다.

- `ip6_nd_dad_solicit_count` — 이 네트워크 매개 변수를 사용하면 IPv6 주소에 중복 주소 감지 기능을 사용할 수 있습니다. 자세한 내용은 이 설명서의 431페이지의 “IPv6 변동 가능 주소 및 중복 주소 감지 기능”을 참조하십시오.
- `tcp_keepalive_interval` — 이 네트워크 매개 변수는 노드에서 사용되지 않는 네트워크 소켓의 리소스를 다시 사용할 수 있도록 회수하기 전까지 네트워크 소켓이 사용되지 않는 상태로 있을 수 있는 시간 길이를 조정합니다.

Serviceguard에서는 다음과 같은 구성에서 `tcp_keepalive_interval`을 변경할 수 있습니다.

— Serviceguard A.11.14 이상을 사용하는 경우

— HP-UX 11.11이 실행되는 노드에서만

또한 다음 요구 사항을 충족해야 합니다.

— `tcp_keepalive_interval`의 최대값은 7200000(2시간, HP-UX 기본값)입니다.

— `tcp_keepalive_interval`의 최소값은 60000(60초)입니다.

— `tcp_keepalive_interval` 값은 노드에서 Serviceguard가 시작되기 전에 설정해야 합니다. 시스템 부팅 시 자동으로 `ndd` 매개 변수를 설정하는 `/etc/rc.config.d/nddconf` 파일에서 새 `tcp_keepalive_interval`을 구성하여 값을 설정할 수 있습니다.

— `tcp_keepalive_interval` 값은 클러스터의 모든 노드에서 동일해야 합니다.

자세한 내용은 <http://docs.hp.com>에 게시된 “Tunable Kernel Parameters”와 “Transport Administrator's Guide” 설명서를 참조하십시오. “Browse by Release”를 클릭한 다음 사용 중인 운영 체제를 선택하여 해당 설명서를 찾으십시오.

클러스터 크기 변경 준비

클러스터가 실행 중일 때 온라인 상태에서 클러스터에 노드를 추가하려면 해당 노드를 다른 클러스터 노드와 동일한 하트비트 서브넷 및 잠금 디스크에 연결해야 합니다. 클러스터 잠금 구성을 선택할 때는 이후에 추가 클러스터 노드가 필요한지를 신중하게 고려해야 합니다. 다섯 개 이상의 노드로 구성된 클러스터에서는 잠금 디스크를 **사용하지 않아도 되지만** 두 개의 노드로 구성된 클러스터에서는 클러스터 잠금을 반드시 사용해야 합니다. 따라서 최종적으로 다섯 개의 노드가 필요한 경우에는 쿼럼 서버를 사용하는 초기 구성을 만들어야 합니다.

클러스터가 실행되고 있을 때 클러스터 구성에서 노드를 제거하려면 제거한 다음의 클러스터 구성이 위에서 설명한 클러스터 잠금 규칙에 위배되지 않는지 확인합니다.

클러스터 구성에서 쉽게 노드를 추가하고 제거하려면 **SCSI** 접속 단자를 그대로 둔 채 버스에서 노드를 제거할 수 있도록 **SCSI** 케이블과 인라인 단자를 함께 사용하면 됩니다. 인라인 **SCSI** 단자에 대한 자세한 내용은 “문제 해결” 장의 “인라인 **SCSI** 단자를 사용하여 온라인 하드웨어 유지 관리” 절을 참조하십시오.

온라인 상태에서 노드를 추가하고 새 노드에서 패키지를 실행하려면 해당 패키지에 대한 기존의 클러스터 연결 볼륨 그룹을 새 노드로 가져왔는지 확인하십시오. 또한 **MAX_CONFIGURED_PACKAGES** 매개 변수가 사용할 총 패키지 수를 수용할 수 있도록 충분히 설정되어 있는지 확인하십시오.

쿼럼 서버 설치

클러스터를 구성하는 동안 실행해야 하는 쿼럼 서버 소프트웨어는 클러스터가 실행될 노드가 아닌 다른 시스템에 설치해야 합니다.

주

쿼럼 서버가 실행되는 노드와 쿼럼 서버의 서비스를 받는 클러스터는 동일한 서브넷에 있는 것이 좋습니다. 이렇게 하면 쿼럼 서버 작동에 영향을 줄 수 있는 네트워크 지연을 방지할 수 있습니다. 서로 다른 서브넷을 사용할 경우에는 쿼럼 서버 시간 종료를 일으킬 수 있는 네트워크 지연이 발생할 수도 있습니다. 이러한 시간 종료 발생하지 않도록 하려면 클러스터 ASCII 파일에서 `QS_TIMEOUT_EXTENSION` 매개 변수를 사용하여 쿼럼 서버의 시간 종료 간격을 늘리면 됩니다.

쿼럼 서버에 연결하는 데 사용되는 네트워크가 클러스터 하트비트 네트워크인 경우에는 하트비트 네트워크 역할을 하는 다른 네트워크가 적어도 하나는 있어야 쿼럼 서버와 하트비트 통신에서 동시에 장애가 발생하는 것을 방지할 수 있습니다.

쿼럼 서버 설치

HP-UX `swinstall` 명령을 사용하여 쿼럼 서버(제품 번호 **B8467BA**)를 실행할 시스템에 설치합니다. 쿼럼 서버는 쿼럼 서버를 사용하는 클러스터의 외부에 설치해야 합니다. 설치에 대한 자세한 내용은 사용 중인 쿼럼 서버 버전의 **쿼럼 서버 릴리즈 노트**를 참조하십시오.

쿼럼 서버 실행 파일인 `qs`는 `/usr/sbin` 디렉토리에 설치됩니다. 설치가 완료되면 특정 호스트 시스템에서 **Quorum** 서비스를 받을 수 있도록 **QS**를 실행할 서버에서 권한 파일을 만들어야 합니다. 이 파일에 **필요한** 경로 이름은 `/etc/cmcluster/qs_authfile`입니다. 이 쿼럼 서버에서 클러스터 서비스에 액세스할 모든 클러스터 노드의 이름을 권한 파일에 추가합니다. 다음 예제와 같이 한 줄에 하나의 노드 이름을 입력합니다.

```
ftsys9.localdomain.com  
ftsys10.localdomain.com
```

모든 노드에서 액세스할 수 있도록 하려면 더하기(+) 기호를 입력한 다음 **Enter** 키를 누릅니다.

쿼럼 서버 실행

클러스터에서 다음 작업이 수행될 때는 쿼럼 서버가 실행되고 있어야 합니다.

- `cmquerycl` 명령이 실행될 때
- `cmapplyconf` 명령이 실행될 때
- 클러스터가 재편성되었을 때

기본적으로 쿼럼 서버 런타임 메시지는 `stdout` 및 `stderr`로 전달됩니다.

`/var/adm/qs` 디렉토리를 만든 다음 `stdout` 및 `stderr`의 경로를 이 디렉토리의 파일(예: `/var/adm/qs/qs.log`)로 다시 지정하는 것이 좋습니다.

쿼럼 서버를 실행하려면 루트 권한이 있어야 합니다. 단일 시스템일 경우 쿼럼 서버가 설치된 시스템이 다시 시작되거나 다시 부팅될 때마다 쿼럼 서버가 시작되도록 구성합니다. 이렇게 하려면 `/etc/inittab` 파일에 다음과 같은 항목을 만듭니다.

```
qs:345:respawn:/usr/sbin/qs >> /var/adm/qs/qs.log 2>&1
```

다음과 같이 쿼럼 서버를 시작합니다.

```
# init q
```

명령이 완료되면 프롬프트가 나타납니다.

`qs.log` 파일에서 쿼럼 서버가 실행 중인지 확인합니다.

```
# cat /var/adm/qs/qs.log
```

다음과 같이 쿼럼 서버가 시작했음을 나타내는 항목이 로그에 있어야 합니다.

```
Oct 04 12:25:06:0:main:Starting Quorum Server
Oct 04 12:25:09:0:main:Server is up and waiting for connections
at port 1238
```

쿼럼 서버 작동 방식에 대한 자세한 내용은 3장을 참조하고, 클러스터 ASCII 파일에서 `cmquerycl` 명령을 사용하여 쿼럼 서버를 지정하는 방법에 대한 자세한 내용은 이 장 뒷부분에 있는 “클러스터 구성”의 “쿼럼 서버 지정” 절을 참조하십시오.

Serviceguard 설치

Serviceguard 설치에는 Software Distributor를 사용한 소프트웨어 업데이트가 포함됩니다. 또한 HP-UX를 이미 설치했다고 가정합니다.

각 노드에 대해 다음 단계를 수행하십시오.

1. 테이프 드라이브 또는 CD ROM 판독기에 배포 미디어를 넣습니다.
2. swinstall 명령을 사용하여 Software Distributor를 실행합니다.
3. 올바른 입력 장치를 지정합니다.
4. 표시된 목록에서 다음 번들을 선택합니다.

B3935DA

Serviceguard

5. 번들을 선택한 다음 확인을 선택합니다. 소프트웨어가 로드됩니다.

swinstall 명령 실행과 새 사용자 계정 작성에 대한 자세한 내용은 HP-UX 설명서인 **시스템 및 작업 그룹 관리**를 참조하십시오.

LVM을 사용하여 저장 장치 하부 구조 만들기

클러스터를 구성하는 것 외에도 다른 노드에서 데이터에 액세스할 수 있도록 적절한 논리 볼륨 하부 구조를 만들어야 합니다. 이 작업은 Logical Volume Manager(LVM), VERITAS Cluster Volume Manager(CVM) 또는 VERITAS Volume Manager(VxVM)를 사용하여 수행합니다. 필요에 따라 볼륨 유형을 혼합하여 사용할 수도 있습니다. LVM 및 VxVM 구성은 클러스터 구성 이전에 완료되며 CVM 구성은 클러스터 구성 후에 완료됩니다.

이 절에서는 LVM을 사용한 저장 장치 구성에 대해 설명합니다. 작업 절차는 다음과 같이 구분할 수 있습니다.

- 미러링되는 개별 디스크의 볼륨 그룹 만들기
- PV 링크를 사용하는 디스크 어레이의 볼륨 그룹 만들기
- 볼륨 그룹을 다른 노드에 배포

Event Monitoring Service HA 디스크 모니터에서는 LVM 디스크의 상태를 모니터링하는 기능을 제공합니다. 미러링되는 디스크에 이 모니터를 사용하려면 디스크를 물리 볼륨 그룹에 구성해야 합니다. 자세한 내용은 **Using High Availability Monitors** 설명서를 참조하십시오.

미러링되는 개별 데이터 디스크의 볼륨 그룹 만들기

이 절에서 설명하는 과정에서는 개별 디스크 미러링에 **물리 볼륨 그룹**을 사용하여 각 논리 볼륨이 다른 입출력 버스의 디스크로 미러링되도록 합니다. 이런 종류의 미러링 방법을 **PVG 완전 미러링**이라고 합니다. 이때 미러 사본으로 사용할 디스크는 다른 사본(기본 사본)에 사용되는 버스와는 다른 버스의 각 노드에 연결되는 방식으로 디스크 하드웨어가 구성되어 있다고 가정합니다.

LVM 사용에 대한 자세한 내용은 HP-UX 설명서인 **시스템 및 작업 그룹 관리**를 참조하십시오.

SAM을 사용하여 볼륨 그룹 및 논리 볼륨 만들기

SAM을 사용하여 HA 패키지에 필요한 볼륨 그룹과 논리 볼륨 구조를 준비할 수 있습니다. SAM에서 “Disks and File Systems Area”를 선택한 다음 패키지에 사용하는 각 볼륨 그룹 및 파일 시스템에 대해 다음 단계를 수행합니다.

1. Volume Groups 하위 영역을 선택합니다.
2. Actions 메뉴에서 Create 또는 Extend를 선택합니다.
3. 볼륨 그룹에서 사용할 첫째 물리 디스크를 사용 가능한 디스크 목록에서 선택합니다.
4. 볼륨 그룹 이름을 vgdatabase와 같이 입력합니다.
5. Create Volume Group 또는 Extend Volume Group을 선택합니다.
6. Add New Logical Volumes를 선택합니다.
7. 볼륨 그룹에 논리 볼륨을 추가할 때 PVG 완전 할당 방식으로 미러링되는 논리 볼륨을 만들고 있는지 확인합니다.
8. 볼륨 그룹에 마운트할 파일 시스템을 지정합니다(예: /mnt1).
9. 추가 볼륨 그룹, 논리 볼륨 및 파일 시스템에 대해 이 절차를 반복합니다.

“볼륨 그룹 비활성화” 절로 건너뛰십시오.

LVM 명령을 사용하여 볼륨 그룹 및 논리 볼륨 만들기

볼륨 그룹을 아직 설정하지 않았으면 다음 절의 과정을 수행하십시오. 이미 LVM을 구성한 경우에는 “클러스터 구성” 절로 건너뛰십시오.

볼륨 그룹의 디스크 선택 두 노드의 디스크 목록을 표시하고 두 노드 모두에서 같은 디스크에 사용되는 장치 파일을 확인합니다. 각 노드에서 다음 명령을 사용하여 각 시스템에 알려진 사용 가능한 디스크를 나열합니다.

```
# lsssf /dev/dsk/*
```

다음 예제에 사용된 /dev/rdisk/c1t2d0과 /dev/rdisk/c0t2d0은 *ftsys9*와 *ftsys10* 모두에서 동일한 디스크의 장치 이름입니다. 노드마다 장치 파일 이름이 다를 경우에는 장치 파일 이름에 주의해야 합니다.

물리 볼륨 만들기 구성 노드(fts9)에서 pvcreate 명령을 사용하여 디스크를 물리 볼륨으로 정의합니다. 이 작업은 구성 노드에서만 수행해야 합니다. 다음 명령을 사용하여 샘플 구성을 위한 두 개의 물리 볼륨을 만드십시오.

```
# pvcreate -f /dev/rdisk/c1t2d0
# pvcreate -f /dev/rdisk/c0t2d0
```

PVG 완전 미러링을 사용하여 볼륨 그룹 만들기 다음 단계를 따라 구성 노드(fts9)에 볼륨 그룹을 만듭니다. 그러면 나중에 다른 노드에 동일한 볼륨 그룹이 만들어집니다.

1. 먼저 vgdatabase의 그룹 디렉토리를 설정합니다.

```
# mkdir /dev/vgdatabase
```

2. 다음과 같이 /dev/vgdatabase 디렉토리에 group이라는 제어 파일을 만듭니다.

```
# mknod /dev/vgdatabase/group c 64 0xhh0000
```

주 번호는 항상 64이고 16진 보조 번호는 다음 형식을 갖습니다.

```
0xhh0000
```

여기서 hh는 만들고 있는 볼륨 그룹에 대해 고유해야 합니다. 아래의 mknod 명령에 대해 모든 노드에서 사용할 수 있는 고유한 보조 번호를 사용하십시오. 이렇게 하면 나중에 NFS로 마운트된 논리 볼륨이 VG에 만들어질 때 더 이상 재구성이 일어나지 않도록 할 수 있습니다.

다음 명령을 사용하여 기존 볼륨 그룹 목록을 표시합니다.

```
# ls -l /dev/*/group
```

3. 다음 명령으로 볼륨 그룹을 만들고 물리 볼륨을 추가합니다.

```
# vgcreate -g bus0 /dev/vgdatabase /dev/dsk/c1t2d0
# vgextend -g bus1 /dev/vgdatabase /dev/dsk/c0t2d0
```

첫 번째 명령은 볼륨 그룹을 만들고 bus0이라는 물리 볼륨 그룹에 물리 볼륨을 추가합니다. 두 번째 명령은 볼륨 그룹에 두 번째 드라이브를 추가하고 bus1이라는 이름의 다른 물리 볼륨 그룹에 추가합니다. 물리 볼륨 그룹을 사용하면 디스크 및 PV 링크의 PVG 완전 미러링을 사용할 수 있습니다.

4. 추가 볼륨 그룹에 대해 이 과정을 반복합니다.

논리 볼륨 만들기

다음 명령을 사용하여 논리 볼륨(이 예제에서는 /dev/vgdatabase)을 만듭니다.

```
# lvcreate -L 120 -m 1 -s g /dev/vgdatabase
```

이 명령은 크기가 120MB이고 이름이 lv011인 미러링되는 볼륨을 만듭니다. 명령에 이름이 지정되지 않았으므로 기본적으로 이 이름이 제공됩니다. -s g 옵션은 PVG 완전 미러링을 의미하므로 데이터의 미리 사본이 다른 물리 볼륨 그룹에 위치하게 됩니다.

주

RAID 1 또는 RAID 5 모드의 디스크 어레이를 사용하는 경우에는 -m 1 및 -s g 옵션을 생략하십시오.

파일 시스템 만들기

설치 프로그램에서 파일 시스템을 사용하는 경우에는 논리 볼륨을 만든 후 파일 시스템을 만듭니다. 다음 명령을 사용하여 방금 만든 논리 볼륨에 마운트할 파일 시스템을 만듭니다.

1. 새로 만든 논리 볼륨에 파일 시스템을 만듭니다.

```
# newfs -F vxfs /dev/vgdatabase/rlvol1
```

논리 볼륨의 원시 장치 파일을 사용해야 합니다.

2. 디스크를 마운트할 디렉토리를 만듭니다.

```
# mkdir /mnt1
```

3. 디스크를 마운트하여 작업을 확인합니다.

```
# mount /dev/vgdatabase/lvol1 /mnt1
```

mount 명령에서는 논리 볼륨의 블록 장치 파일을 사용합니다.

4. 구성을 확인합니다.

```
# vgdisplay -v /dev/vgdatabase
```

PV 링크를 사용하는 디스크 어레이의 볼륨 그룹 만들기

HP의 HA 디스크 어레이에서 대용량 저장 장치를 사용하는 볼륨 그룹을 구성하는 경우 각 노드의 중복 입출력 채널을 어레이의 별도 포트에 연결하여 사용해야 합니다. 그런 다음 어레이에서 정의한 LUN 또는 논리 디스크의 대체 링크(PV 링크라고도 함)를 정의할 수 있습니다. SAM에서는 구성하려는 디스크 어레이 종류를 선택한 다음 메뉴에 따라 대체 링크를 정의하십시오.

다음 예제에서는 LVM 명령을 사용하여 대체 링크를 구성하는 방법을 보여 줍니다. 이 예제에서 디스크 구성은 다음과 같다고 가정합니다.

```
8/0.15.0 /dev/dsk/c0t15d0 /* I/O Channel 0 (8/0) SCSI address
15 LUN 0 */
8/0.15.1 /dev/dsk/c0t15d1 /* I/O Channel 0 (8/0) SCSI address
15 LUN 1 */
8/0.15.2 /dev/dsk/c0t15d2 /* I/O Channel 0 (8/0) SCSI address
15 LUN 2 */
8/0.15.3 /dev/dsk/c0t15d3 /* I/O Channel 0 (8/0) SCSI address
15 LUN 3 */
8/0.15.4 /dev/dsk/c0t15d4 /* I/O Channel 0 (8/0) SCSI address
15 LUN 4 */
8/0.15.5 /dev/dsk/c0t15d5 /* I/O Channel 0 (8/0) SCSI address
15 LUN 5 */

10/0.3.0 /dev/dsk/c1t3d0 /* I/O Channel 1 (10/0) SCSI address
3 LUN 0 */
10/0.3.1 /dev/dsk/c1t3d1 /* I/O Channel 1 (10/0) SCSI address
3 LUN 1 */
10/0.3.2 /dev/dsk/c1t3d2 /* I/O Channel 1 (10/0) SCSI address
3 LUN 2 */
10/0.3.3 /dev/dsk/c1t3d3 /* I/O Channel 1 (10/0) SCSI address
3 LUN 3 */
10/0.3.4 /dev/dsk/c1t3d4 /* I/O Channel 1 (10/0) SCSI address
3 LUN 4 */
10/0.3.5 /dev/dsk/c1t3d5 /* I/O Channel 1 (10/0) SCSI address
3 LUN 5 */
```

또한, 디스크 어레이가 구성되어 있고 ioscan 명령을 실행할 때 다음과 같은 장치 파일이 모두 같은 LUN(논리 디스크)에 대해 나타난다고 가정합니다.

```
/dev/dsk/c0t15d0  
/dev/dsk/c1t3d0
```

다음 단계에 따라 이 논리 디스크의 볼륨 그룹을 구성합니다.

1. 먼저 `vgdatabase`의 그룹 디렉토리를 설정합니다.

```
# mkdir /dev/vgdatabase
```

2. 다음과 같이 `/dev/vgdatabase` 디렉토리에 `group`이라는 제어 파일을 만듭니다.

```
# mknod /dev/vgdatabase/group c 64 0xhh0000
```

주 번호는 항상 64이고 16진 보조 번호는 다음 형식을 갖습니다.

```
0xhh0000
```

여기서 `hh`는 만들고 있는 볼륨 그룹에 대해 고유해야 합니다. 모든 노드에서 사용할 수 있는 고유한 번호를 사용하십시오. 이렇게 하면 나중에 NFS로 마운트된 논리 볼륨이 VG에 만들어질 때 더 이상 재구성이 일어나지 않도록 할 수 있습니다.

다음 명령을 사용하여 기존의 그룹 파일 목록을 표시합니다.

```
# ls -l /dev/*/group
```

3. LUN과 연관된 장치 파일 중 하나에 `pvcreate` 명령을 사용하여 LVM에 대한 LUN을 물리 볼륨으로 정의합니다.

```
# pvcreate /dev/rdisk/c0t15d0
```

LUN의 장치 파일 이름 중 하나에 대해서만 이 작업을 수행해야 합니다.

4. 다음 명령을 사용하여 볼륨 그룹을 만듭니다.

```
# vgcreate /dev/vgdatabase /dev/dsk/c0t15d0
```

```
# vextend /dev/vgdatabase /dev/dsk/c1t3d0
```

이제 `vgdisplay -v` 명령을 사용하여 주 링크와 대체 링크를 볼 수 있습니다. LVM에서는 이제 `/dev/dsk/c0t15d0`이 나타내는 입출력 채널을 디스크에 대한 주 링크로 인식하게 됩니다. 주 링크에 장애가 발생할 경우 LVM에서는 자동으로 `/dev/dsk/c1t3d0`이 나타내는 대체 입출력 채널로 전환합니다.

논리 볼륨을 만들려면 앞의 “논리 볼륨 만들기” 절에서 설명한 과정을 따르십시오.

블록 그룹을 다른 노드에 배포

클러스터 데이터의 블록 그룹을 만든 후에는 해당 블록 그룹을 활성화할 클러스터 노드로 배포해야 합니다. 클러스터 잠금 블록 그룹은 모든 노드에서 사용 가능해야 합니다.

블록 그룹 비활성화

블록 그룹을 만들 때 해당 블록 그룹은 구성 노드(예: *ftsys9*)에서 활성화되어 있습니다. 블록 그룹을 다른 노드에서 사용할 수 있도록 설정하려면 먼저 블록 그룹에 상주하는 파일 시스템을 마운트 해제한 다음 블록 그룹을 비활성화해야 합니다. 실행 시 블록 그룹 활성화와 파일 시스템 마운트 작업은 패키지 제어 스크립트를 통해 이루어집니다. 앞의 절에 표시된 예제를 계속 진행하여 *ftsys9*에서 다음을 수행합니다.

```
# umount /mnt1  
# vgchange -a n /dev/vgdatabase
```

LVM 명령을 사용하여 블록 그룹 배포

다음 명령을 사용하여 동일한 블록 그룹을 다른 클러스터 노드에 설정합니다. 이 예제의 명령은 *ftsys9*에서 사용할 수 있는 것과 동일한 물리 볼륨을 포함할 *ftsys10*에 새 블록 그룹을 설정합니다. 블록 그룹의 패키지가 실행될 수 있는 각 노드에 대해 개별적으로 같은 절차를 수행해야 합니다.

*ftsys10*에 블록 그룹을 설정하려면 다음 단계를 수행합니다.

1. *ftsys9*에서 블록 그룹의 매핑을 지정된 파일에 복사합니다.

```
# vexport -p -s -m /tmp/vgdatabase.map /dev/vgdatabase
```

2. *ftsys9*에서 맵 파일을 *ftsys10*에 복사합니다.

```
# rcp /tmp/vgdatabase.map ftsys10:/tmp/vgdatabase.map
```

3. *ftsys10*에서 블록 그룹 디렉토리를 만듭니다.

```
# mkdir /dev/vgdatabase
```

4. *ftsys10*에서 다음과 같이 */dev/vgdatabase* 디렉토리에 *group*이라는 제어 파일을 만듭니다.

```
# mknod /dev/vgdatabase/group c 64 0xhh0000
```

*ftsys9*에서 사용하는 것과 동일한 보조 번호를 사용합니다. 다음 명령을 사용하여 기존 블록 그룹 목록을 표시합니다.

```
# ls -l /dev/*/group
```

5. 노드 `ftsys9`에서 맵 파일을 사용하여 볼륨 그룹 데이터를 가져옵니다. 노드 `ftsys10`에서 다음을 입력합니다.

```
# vgimport -s -m /tmp/vgdatabase.map /dev/vgdatabase
```

`ftsys10`의 디스크 장치 이름은 `ftsys9`의 디스크 장치 이름과 다를 수 있습니다. 물리 볼륨 이름이 클러스터 전체에서 올바른지 확인해야 합니다.

이 노드에서 **VG**를 활성화할 수 있으면 기본 노드의 장애와 볼륨 그룹의 **LVM** 문제로 인해 이 노드에서 `vgcfgrestore`를 수행해야 할 경우 `vgcfgbackup`을 수행합니다.

다음 예제에서는 이러한 작업을 보여 줍니다.

```
# vgchange -a y /dev/vgdatabase
# vgcfgbackup /dev/vgdatabase
# vgchange -a n /dev/vgdatabase
```

6. 물리 볼륨 그룹에서 미러링되는 개별 디스크를 사용하는 경우 `/etc/lvmpvg` 파일에서 각 물리 볼륨 그룹에 `ftsys10`의 올바른 물리 볼륨 이름이 포함되어 있는지 확인합니다.

주

PVG 완전 미러링을 사용하는 경우 물리 볼륨 그룹 구성은 구성 노드의 `/etc/lvmpvg` 파일에 기록됩니다. 이 파일에서는 미러링의 기본이 되는 물리 볼륨 그룹을 정의하고 각 **PVG**에 속하는 물리 볼륨을 표시합니다. 각 클러스터 노드에서 `/etc/lvmpvg` 파일에는 **PVG** 디스크에 대해 **해당 노드에 알려진 것과 같은** 올바른 물리 볼륨 이름이 포함되어야 합니다. 동일한 디스크의 물리 볼륨 이름이 노드마다 다를 수 있습니다. 다른 노드에 볼륨 그룹을 배포한 후에는 각 노드의 `/etc/lvmpvg` 파일이 해당 노드에 있는 모든 물리 볼륨 그룹의 내용을 올바르게 반영하는지 확인해야 합니다. 자세한 내용은 다음의 “물리 볼륨 그룹 파일의 일관성 유지” 절을 참조하십시오.

7. `ftsys9`에서 볼륨 그룹이 비활성화되었는지 확인합니다. 그런 다음 `ftsys10`에서 볼륨 그룹을 활성화합니다.

```
# vgchange -a y /dev/vgdatabase
```

8. 디스크를 마운트할 디렉토리를 만듭니다.

```
# mkdir /mnt1
```

9. 마운트한 다음에 `ftsys10`의 볼륨 그룹을 확인합니다.

```
# mount /dev/vgdatabase/lvol1 /mnt1
```

10. `ftsys10`에서 볼륨 그룹을 마운트 해제합니다.

```
# umount /mnt1
```

11. `ftsys10`에서 볼륨 그룹을 비활성화합니다.

```
# vgchange -a n /dev/vgdatabase
```

물리 볼륨 그룹 파일의 일관성 유지

디스크 구성에서 미러링되는 개별 디스크의 물리 볼륨 그룹을 사용하지 않을 경우 다음 절로 건너뛰십시오.

서로 다른 볼륨 그룹은 **Serviceguard** 클러스터 내의 서로 다른 노드 하위 집합에 의해 활성화될 수 있습니다. 또한 특정 디스크의 물리 볼륨 이름은 노드마다 다를 수 있습니다. 따라서 각 노드가 모든 클러스터 인식 디스크와 자체의 독립 디스크(클러스터를 인식하지 않는 디스크)를 완전하고 일관되게 파악할 수 있도록 모든 노드에 있는 `/etc/lvmmpvg` 파일을 신중하게 병합해야 합니다. 파일을 보다 쉽게 병합하려면 볼륨 그룹 계획 워크시트에 물리 볼륨 그룹 이름을 잘 기록해 두어야 합니다(“계획” 장 참조).

다음 단계에 따라 구성 노드(`ftsys9`)와 볼륨 그룹을 가져올 새 노드(`ftsys10`) 간에 파일을 병합하십시오.

1. `ftsys9`의 `/etc/lvmmpvg`를 `ftsys10`의 `/etc/lvmmpvg.new`로 복사합니다.
2. `ftsys10`에 없는 볼륨 그룹이 `/etc/lvmmpvg.new`에 있으면 `/etc/lvmmpvg.new`에서 해당 볼륨 그룹에 대한 모든 항목을 제거합니다.
3. `ftsys10`의 `/etc/lvmmpvg`에 `/etc/lvmmpvg.new`에는 없는 볼륨 그룹에 대한 항목이 포함되어 있으면 해당 볼륨 그룹의 모든 **PVG** 항목을 `/etc/lvmmpvg.new`에 복사합니다.
4. `ftsys10`의 올바른 이름을 반영하도록 `/etc/lvmmpvg.new`의 물리 볼륨 이름을 수정합니다.
5. `ftsys10`에서 `/etc/lvmmpvg`를 `/etc/lvmmpvg.old`로 복사하여 백업을 만듭니다. `/etc/lvmmpvg.new`를 `ftsys10`의 `/etc/lvmmpvg`로 복사합니다.

추가 볼륨 그룹 만들기

지금까지 Serviceguard에서 사용할 볼륨 그룹과 논리 볼륨을 만드는 일반적인 방법을 설명했습니다. 만들어야 하는 볼륨 그룹의 수만큼 해당 과정을 반복합니다. 이때, 볼륨 그룹 이름, 논리 볼륨 이름 및 물리 볼륨 이름은 해당 이름으로 대체하십시오. 특히 디스크 장치 이름에 주의하십시오. 예를 들어, 한 노드의 `/dev/dsk/c0t2d0`이 다른 노드에서 `/dev/dsk/c0t2d0`이 될 수는 없습니다.

VxVM을 사용하여 저장 장치 하부 구조 만들기

클러스터를 구성하는 것 외에도 다른 노드에서 데이터에 액세스할 수 있도록 적절한 논리 볼륨 하부 구조를 만들어야 합니다. 이 작업은 Logical Volume Manager(LVM), VERITAS Volume Manager(VxVM) 또는 VERITAS Cluster Volume Manager(CVM)를 통해 수행합니다. 필요에 따라 볼륨 유형을 혼합하여 사용할 수도 있습니다. LVM 및 VxVM 구성은 클러스터 구성 이전에 수행하며 CVM 구성은 클러스터 구성 후에 수행합니다.

LVM에서 VxVM 저장 장치로 마이그레이션하는 방법에 대한 자세한 내용은 부록 G를 참조하십시오.

이 절에서는 VERITAS Volume Manager(VxVM)의 명령을 사용하여 새 저장 장치를 구성하는 방법에 대해 설명합니다. 루트 디스크 그룹을 만들었으면(다음에 설명) VxVM 명령이나 Storage Administrator GUI인 vmsa를 사용하여 구성 작업을 수행할 수 있습니다. vmsa를 사용하는 경우에는 GUI를 실행하기 전에 Storage Administrator 서버가 실행 중인지 확인합니다. 자세한 내용은 *VERITAS Volume Manager for HP-UX Release Notes*와 *VERITAS VMSA Administrator's Guide*를 참조하십시오. 명령을 사용하는 경우에는 VxVM 맨페이지를 참조하십시오.

VERITAS Volume Manager 초기화

디스크 그룹을 처음 만드는 경우 Volume Manager를 초기화해야 합니다. 하나 이상의 디스크를 포함하는 **rootdg**라는 디스크 그룹을 만들어 이 작업을 수행합니다. 각 노드에서 VxVM을 설치한 후 즉시 다음 명령을 **한 번만** 사용하십시오.

```
# vxinstall
```

이 명령을 사용하면 VxVM 초기화 과정을 단계별로 안내하는 메뉴 방식의 프로그램이 표시됩니다. 주 메뉴에서 “Custom” 옵션을 선택하고 **rootdg**에 포함할 디스크를 지정합니다.

중요

VERITAS Volume Manager의 **rootdg**는 HP-UX 루트 디스크 파일 시스템에서 LVM 볼륨 그룹을 사용하는 경우 HP-UX 루트 디스크와 동일하지 않습니다. 또한 **rootdg**는 공유 저장 장치에 사용할 수 없습니다. 그러나 **rootdg**는 다른 로컬 파일 시스템(예: /export/home)에 사용할 수 있으므로 유용하게 쓰입니다.

루트 디스크 그룹은 **각 노드에서 한 번만** 만들어야 합니다.

LVM에서 VxVM으로 디스크 변환

`vxvmconvert (1m)` 유틸리티를 사용하여 LVM 볼륨 그룹을 VxVM 디스크 그룹으로 변환할 수 있습니다. 이 작업을 수행하려면 먼저 볼륨 그룹을 비활성화해야 합니다. 즉, 해당 볼륨 그룹을 사용하는 패키지를 중지해야 합니다. *VERITAS Volume Manager Migration Guide*에 설명된 변환 과정을 따르십시오. 시작하기 전에 `vgcfgbackup` 명령을 사용하여 각 볼륨 그룹의 구성을 백업하고 볼륨 그룹의 데이터를 백업해야 합니다.

변환에 대한 자세한 내용은 부록 H “LVM에서 VxVM 데이터 저장 장치로 마이그레이션”을 참조하십시오.

VxVM용 디스크 초기화

VxVM 디스크 그룹에 사용할 물리 디스크를 초기화해야 합니다. 디스크를 초기화하려면 클러스터의 한 노드에 로그인한 다음 `vxdiskadm` 프로그램을 사용하여 여러 디스크를 한꺼번에 초기화하거나 `vxdisksetup` 명령을 사용하여 디스크를 한 번에 하나씩 초기화합니다. 예를 들면 다음과 같습니다.

```
# /usr/lib/vxvm/bin/vxdisksetup -i c0t3d2
```

이전에 LVM에서 사용된 디스크 초기화

이전에 물리 디스크를 LVM과 함께 사용한 경우 `pvremove` 명령을 사용하여 볼륨 그룹의 모든 디스크에서 LVM 헤더 데이터를 삭제해야 합니다. 또한 LVM 디스크를 이전에 클러스터에서 사용한 경우에는 `pvcreate -f` 명령으로 디스크를 다시 초기화하여 디스크에서 클러스터 ID를 제거해야 합니다.

주

이 명령은 디스크와 디스크의 데이터를 LVM에서 사용할 수 없게 만들고 VxVM에서 초기화되도록 합니다. 이전에 LVM을 사용하는 디스크를 사용했고 해당 디스크에 데이터를 저장하지 않을 경우에만 이 명령을 사용해야 합니다.

다음 예제와 같이 디스크에서 LVM 헤더 데이터를 제거할 수 있습니다. 이때 디스크의 모든 데이터가 지워지므로 주의하십시오.

```
# pvremove /dev/rdsk/c0t3d2
```

```
# pvcreate -f /dev/rdsk/c0t3d2
```

그런 다음 vxdiskadm 프로그램을 사용하여 VxVM의 여러 디스크를 한 번에 초기화하거나 vxdisksetup 명령을 사용하여 디스크를 한 번에 하나씩 초기화합니다. 예를 들면 다음과 같습니다.

```
# /usr/lib/vxvm/bin/vxdisksetup -i c0t3d2
```

디스크 그룹 만들기

vxdiskadm을 사용하거나 다음 예제와 같이 vxdg 명령을 사용하여 디스크 그룹을 만듭니다.

```
# vxdg init logdata c0t3d2
```

다음 명령으로 구성을 확인합니다.

```
# vxdg list
```

| NAME | STATE | ID |
|---------|---------|----------------------|
| rootdg | enabled | 971995699.1025.node1 |
| logdata | enabled | 972078742.1084.node1 |

볼륨 만들기

vxassist 명령을 사용하여 논리 볼륨을 만듭니다. 예를 들면 다음과 같습니다.

```
# vxassist -g logdata make log_files 1024m
```

이 명령은 logdata라는 디스크 그룹에 log_files라는 1024MB 크기의 볼륨을 만듭니다. 이 볼륨은 블록 장치 파일 /dev/vx/dsk/logdata/log_files 또는 원시(문자) 장치 파일 /dev/vx/rdsk/logdata/log_files를 사용하여 참조할 수 있습니다. 다음 명령으로 구성을 확인합니다.

```
# vxprint -g logdata
```

이 명령의 출력은 다음 예제와 같습니다.

| TY | NAME | ASSOC | KSTATE | LENGTH | PLOFFS | STATE | TUTILO |
|----|------|-------|--------|--------|--------|-------|--------|
| | | | | | | | PUTILO |

HA 클러스터 구성 만들기

VxVM을 사용하여 저장 장치 하부 구조 만들기

| | | | | | |
|----|------------|--------|---------|---------|--------|
| v | logdata | fsgen | ENABLED | 1024000 | ACTIVE |
| p1 | logdata-01 | system | ENABLED | 1024000 | ACTIVE |

주

VxVM을 사용하는 미러링되는 다중 경로 저장 장치를 만들기 위한 특정 명령에 대한 자세한 내용은 *VERITAS Volume Manager Reference Guide*를 참조하십시오.

파일 시스템 만들기

설치 프로그램에서 파일 시스템을 사용하는 경우에는 논리 볼륨을 만든 후 파일 시스템을 만듭니다. 다음 명령을 사용하여 방금 만든 논리 볼륨에 마운트할 파일 시스템을 만듭니다.

1. 새로 만든 볼륨에 파일 시스템을 만듭니다.

```
# newfs -F vxfs /dev/vx/rdisk/logdata/log_files
```

2. 볼륨을 마운트할 디렉토리를 만듭니다.

```
# mkdir /logs
```

3. 볼륨을 마운트합니다.

```
# mount /dev/vx/dsk/logdata/log_files /logs
```

4. 파일 시스템이 있는지 확인한 다음 파일 시스템을 마운트 해제합니다.

```
# umount /logs
```

디스크 그룹 내보내기

Serviceguard 패키지에서 사용할 디스크 그룹을 만든 후 각 디스크 그룹에 다음 명령을 사용하여 여러 클러스터 노드에서 패키지 제어 스크립트를 통해 디스크 그룹을 가져오도록 합니다.

```
# vxdbg deport <DiskGroupName>
```

여기서 <DiskGroupName>은 제어 스크립트를 통해 활성화될 디스크 그룹의 이름입니다.

모든 디스크 그룹을 내보낸 후에는 모든 클러스터 노드에서 다음 명령을 실행하여 디스크 그룹에 액세스할 수 있도록 해야 합니다.

```
# vxdctl enable
```

디스크 그룹 다시 가져오기

디스크 그룹을 내보낸 후에는 패키지 제어 스크립트 또는 `vxdg import` 명령을 사용하여 다시 가져와야 해당 노드에서 사용할 수 있습니다. 유지 관리 또는 다른 목적으로 직접 디스크 그룹을 가져와야 하는 경우 다음 예제와 같이 디스크 그룹을 가져오고 이 디스크 그룹의 논리 볼륨을 모두 시작한 다음 파일 시스템을 마운트합니다.

```
# vxdg import dg_01  
  
# vxvol -g dg_01 startall  
  
# mount /dev/vx/dsk/dg_01/myvol /mountpoint
```

주

VxVM 디스크 그룹은 LVM 볼륨 그룹과 달리 클러스터 ASCII 구성 파일에 입력되지 않으며 패키지 ASCII 구성 파일에도 입력되지 않습니다.

시스템 재부팅 시 Clearimport

시스템을 다시 부팅하면 `cmcluster RC` 스크립트는 시스템에서 이전에 가져온 모든 디스크에 대해 `vxdisk clearimport`를 수행합니다. 단, 이 디스크는 `noautoimport` 플래그가 설정되어 있고 현재 이 디스크에 대해 가져오기를 수행하는 다른 실행 노드가 없어야 합니다. `clearimport`는 디스크 그룹에서 호스트 ID를 삭제하여 패키지가 한 노드에서 다른 노드로 이동할 때 해당 디스크 그룹에 연결된 모든 노드에서 이를 가져올 수 있도록 합니다.

시스템을 다시 부팅할 때 `clearimport`를 사용하면 `Serviceguard`에서는 전원 장애 시 시스템이 중단되는 등의 노드 장애가 발생한 후에도 완전히 복구할 수 있습니다. 장애 시 가져온 디스크에는 여전히 해당 노드의 ID가 기록되어 있으므로 다시 부팅하는 노드나 다른 노드에서 패키지 제어 스크립트를 사용하여 이를 가져올 수 있도록 하려면 먼저 이 ID를 삭제해야 합니다.

`clearimport`는 이전에 `Serviceguard`가 설치된 모든 시스템에서 `noautoimport`가 설정된 상태로 가져온 디스크에 대해 수행됩니다. 디스크가 클러스터에 구성되었는지 여부는 상관이 없습니다.

클러스터 구성

이 절에서는 기본 클러스터 구성을 정의하는 방법에 대해 설명합니다. 그래픽 사용자 인터페이스인 **Serviceguard Manager**에서 기본 클러스터 구성을 정의하려면 다음 절을 읽어 보십시오. **Serviceguard** 명령을 사용하려면 “**Serviceguard** 명령을 사용하여 클러스터 구성” 절로 건너뛰십시오.

Serviceguard Manager를 사용하여 클러스터 구성

Serviceguard Manager에서 세션을 만듭니다. 그런 다음 미사용 노드를 찾기 위한 옵션을 선택하고, 맵 또는 트리의 미사용 노드 목록에서 클러스터를 시작할 위치를 선택합니다. 동작 메뉴에서 구성을 선택합니다.

노드의 루트 암호를 지정하면 구성 화면이 열리고 프로세스 안내가 표시됩니다. 각 탭에는 관련 정보가 들어 있습니다. **Serviceguard Manager**에서는 많은 정보를 검색하므로 사용 가능한 옵션에서 볼륨 그룹, 네트워크 및 노드 목록 등을 선택할 수 있습니다.

각 단계에서는 필요한 사항을 결정하는 데 유용한 온라인 도움말도 볼 수 있습니다.

클러스터를 구성하기 전에 먼저 볼륨 그룹을 구성합니다. 퀴럼 서버를 클러스터 잠금으로 사용하는 경우에는 퀴럼 서버를 실행한 다음 클러스터를 구성합니다.

정보를 모두 입력한 후 적용을 클릭합니다. 오류가 발생하면 로그 창에 오류가 표시됩니다. 오류가 발생하지 않으면 “**successful**”이라는 메시지가 표시되고 이진 구성이 노드에 자동으로 배포됩니다.

새로 고침을 수행하면 새 클러스터 구성 및 상태 정보가 트리, 맵 및 속성에 나타납니다.

구성을 수정하거나 삭제하려면 트리 또는 맵에서 클러스터를 선택하고 동작 메뉴에서 구성을 선택합니다.

Serviceguard 명령을 사용하여 클러스터 구성

`cmquerycl` 명령을 사용하여 클러스터에 포함할 노드 집합을 지정하고 클러스터 구성 파일을 위한 템플릿을 생성합니다. 노드 이름은 31바이트 이하여야 합니다. 다음은 명령의 예입니다.

```
# cmquerycl -v -C /etc/cmcluster/clust1.config -n ftsys9 -n ftsys10
```

이 예제에서는 기본 클러스터 구성 디렉토리인 `/etc/cmcluster`에 ASCII 템플릿 파일을 만듭니다. ASCII 파일에는 두 개의 노드 `ftsys9`와 `ftsys10`에 있는 클러스터 구성 요소의 이름 및 특성이 포함됩니다. 노드 이름을 지정할 때 도메인 이름은 포함하지 마십시오. 예를 들면, `ftsys9.cup.hp.com`이 아니라 `ftsys9`로 지정합니다. 파일에 미리 지정된 클러스터 특성을 원하는 클러스터에 맞게 편집하십시오. 다음 예제와 같이 가능한 모든 네트워크에서 하트비트를 전송하도록 파일을 편집하는 것이 가장 좋습니다.

주

많은 수의 노드, 네트워크 또는 디스크가 연결된 더 크고 복잡한 클러스터 구성에서는 `cmquerycl` 명령이 완료되는 데 몇 분씩 걸릴 수 있습니다. 구성 작업의 속도를 빠르게 하기 위해 명령에 `-k` 옵션과 `-w` 옵션을 사용하여 선택한 정보만 반환되도록 지정할 수 있습니다.

`-k` 옵션을 사용하면 일부 디스크 검사가 제외되고 잠재적인 클러스터 잠금 볼륨 그룹과 잠금 물리 볼륨에 대한 정보가 반환되지 않습니다.

`-w local` 옵션을 사용하면 로컬 네트워크 검사를 지정하여 각 노드 내의 인터페이스 간 LAN 연결만 검사하도록 할 수 있습니다.

`-w full` 옵션을 사용하면 전체 네트워크 검사를 지정하여 클러스터에 있는 모든 노드에서 모든 LAN 인터페이스 간의 실제 연결을 검사하도록 할 수 있습니다. 이 옵션이 기본값입니다. `-w none` 옵션을 사용하면 네트워크 쿼리를 건너뛵니다. 최근에 네트워크를 확인한 적이 있는 경우 이 옵션을 사용하면 시간이 절약됩니다.

자세한 내용은 `cmquerycl(1m)`에 대한 맨페이지를 참조하십시오.

클러스터 구성 템플릿 파일

다음은 cmquerycl 명령과 -w full 옵션을 사용하여 생성한 ASCII 구성 파일의 예제입니다.

```
# *****  
# ***** HIGH AVAILABILITY CLUSTER CONFIGURATION FILE *****  
# ***** For complete details about cluster parameters and how to *****  
# ***** set them, consult the Serviceguard manual. *****  
# *****  
# Enter a name for this cluster. This name will be used to identify the  
# cluster when viewing or manipulating it.  
  
CLUSTER_NAME cluster1  
  
# Cluster Lock Parameters  
# The cluster lock is used as a tie-breaker for situations  
# in which a running cluster fails, and then two equal-sized  
# sub-clusters are both trying to form a new cluster. The  
# cluster lock may be configured using either a lock disk  
# or a quorum server.  
#  
# You can use either the quorum server or the lock disk as  
# a cluster lock but not both in the same cluster.  
#  
# Consider the following when configuring a cluster.  
# For a two-node cluster, you must use a cluster lock. For  
# a cluster of three or four nodes, a cluster lock is strongly  
# recommended. For a cluster of more than four nodes, a  
# cluster lock is recommended. If you decide to configure  
# a lock for a cluster of more than four nodes, it must be  
# a quorum server.  
  
# Lock Disk Parameters. Use the FIRST_CLUSTER_LOCK_VG and  
# FIRST_CLUSTER_LOCK_PV parameters to define a lock disk.  
# The FIRST_CLUSTER_LOCK_VG is the LVM volume group that  
# holds the cluster lock. This volume group should not be  
# used by any other cluster as a cluster lock device.  
  
# Quorum Server Parameters. Use the QS_HOST, QS_POLLING_INTERVAL,  
# and QS_TIMEOUT_EXTENSION parameters to define a quorum server.  
# The QS_HOST is the host name or IP address of the system  
# that is running the quorum server process. The  
# QS_POLLING_INTERVAL (microseconds) is the interval at which  
# Serviceguard checks to make sure the quorum server is running.  
# The optional QS_TIMEOUT_EXTENSION (microseconds) is used to increase  
# the time interval after which the quorum server is marked DOWN.
```



```
#
# The default quorum server timeout is calculated from the
# Serviceguard cluster parameters, including NODE_TIMEOUT and
# HEARTBEAT_INTERVAL.  If you are experiencing quorum server

# timeouts, you can adjust these parameters, or you can include
# the QS_TIMEOUT_EXTENSION parameter.
#
# The value of QS_TIMEOUT_EXTENSION will directly effect the amount
# of time it takes for cluster reformation in the event of failure.
# For example, if QS_TIMEOUT_EXTENSION is set to 10 seconds, the cluster
# reformation will take 10 seconds longer than if the QS_TIMEOUT_EXTENSION
# was set to 0. This delay applies even if there is no delay in
# contacting the Quorum Server.  The recommended value for
# QS_TIMEOUT_EXTENSION is 0, which is used as the default
# and the maximum supported value is 30000000 (5 minutes).
#
# For example, to configure a quorum server running on node
# "qshost" with 120 seconds for the QS_POLLING_INTERVAL and to
# add 2 seconds to the system assigned value for the quorum server
# timeout, enter:
#
# QS_HOST qshost
# QS_POLLING_INTERVAL 12000000
# QS_TIMEOUT_EXTENSION 2000000

QS_HOST                sysman5
QS_POLLING_INTERVAL    30000000
# Definition of nodes in the cluster.
# Repeat node definitions as necessary for additional nodes.
# NODE_NAME is the specified nodename in the cluster.
# It must match the hostname and both cannot contain full domain name.
# Each NETWORK_INTERFACE, if configured with IPv4 address,
# must have ONLY one IPv4 address entry with it which could
# be either HEARTBEAT_IP or STATIONARY_IP.
# Each NETWORK_INTERFACE, if configured with IPv6 address(es)
# can have multiple IPv6 address entries(up to a maximum of 2,
# only one IPv6 address entry belonging to site-local scope
# and only one belonging to global scope) which must be all
# STATIONARY_IP. They cannot be HEARTBEAT_IP.

NODE_NAME              fresno
  NETWORK_INTERFACE    lan0
    HEARTBEAT_IP       15.13.168.91
# List of serial device file names
# For example:
# SERIAL_DEVICE_FILE   /dev/tty0p0
```

HA 클러스터 구성 만들기

클러스터 구성

```
# Warning: There are no standby network interfaces for lan0.

NODE_NAME                lodi

    NETWORK_INTERFACE     lan0
    HEARTBEAT_IP          15.13.168.94
# List of serial device file names
# For example:
# SERIAL_DEVICE_FILE     /dev/tty0p0

# Warning: There are no standby network interfaces for lan0.

# Cluster Timing Parameters (microseconds).

# The NODE_TIMEOUT parameter defaults to 2000000 (2 seconds).
# This default setting yields the fastest cluster reformations.
# However, the use of the default value increases the potential
# for spurious reformations due to momentary system hangs or
# network load spikes.
# For a significant portion of installations, a setting of
# 5000000 to 8000000 (5 to 8 seconds) is more appropriate.
# The maximum value recommended for NODE_TIMEOUT is 30000000
# (30 seconds).

HEARTBEAT_INTERVAL       1000000
NODE_TIMEOUT              2000000

# The FAILOVER_OPTIMIZATION parameter enables Failover Optimization,
# which reduces the time Serviceguard takes for failover. (Failover
# Optimization cannot, however, change the time an application
# needs to shut down or restart.)
#
# There are four requirements:
# * The Serviceguard Extension for Faster Failover product
#   (SGeFF) must be installed on all cluster nodes.
# * Only one or two node clusters are supported.
# * A quorum server must be configured as the tie-breaker.
# * The cluster must have more than one heartbeat subnet,
#   and neither can be a serial line (RS232).
#
# Other considerations are listed in the SGeFF Release Notes
# and the Serviceguard manual.
#
# You must halt the cluster to change FAILOVER_OPTIMIZATION
# parameter.
#
```

```
# To enable Failover Optimization, set FAILOVER_OPTIMIZATION
# to TWO_NODE.
# The default is NONE.
#

# FAILOVER_OPTIMIZATION <NONE/TWO_NODE>

FAILOVER_OPTIMIZATION NONE

# Configuration/Reconfiguration Timing Parameters (microseconds).

AUTO_START_TIMEOUT      600000000
NETWORK_POLLING_INTERVAL 2000000

# Network Monitor Configuration Parameters.
# The NETWORK_FAILURE_DETECTION parameter determines how LAN card failures are detected.
# If set to INONLY_OR_INOUT, a LAN card will be considered down when its inbound
# message count stops increasing or when both inbound and outbound
# message counts stop increasing.
# If set to INOUT, both the inbound and outbound message counts must
# stop increasing before the card is considered down.
NETWORK_FAILURE_DETECTION INOUT

# Package Configuration Parameters.
# Enter the maximum number of packages which will be configured in the cluster.
# You can not add packages beyond this limit.
# This parameter is required.
MAX_CONFIGURED_PACKAGES 150

# Access Control Policy Parameters.
#
# Three entries set the access control policy for the cluster:
# First line must be USER_NAME, second USER_HOST, and third USER_ROLE.
# Enter a value after each.
#
# 1. USER_NAME can either be ANY_USER, or a maximum of
# 8 login names from the /etc/passwd file on user host.
# 2. USER_HOST is where the user can issue Serviceguard commands.
# If using Serviceguard Manager, it is the COM server.
# Choose one of these three values: ANY_Serviceguard_NODE, or
# (any) CLUSTER_MEMBER_NODE, or a specific node. For node,
# use the official hostname from domain name server, and not
# an IP addresses or fully qualified name.
# 3. USER_ROLE must be one of these three values:
```

HA 클러스터 구성 만들기

클러스터 구성

```
# * MONITOR: read-only capabilities for the cluster and packages
# * PACKAGE_ADMIN: MONITOR, plus administrative commands for packages
#   in the cluster
# * FULL_ADMIN: MONITOR and PACKAGE_ADMIN plus the administrative

#   commands for the cluster.
#
# Access control policy does not set a role for configuration
# capability. To configure, a user must log on to one of the
# cluster's nodes as root (UID=0). Access control
# policy cannot limit root users' access.
#
# MONITOR and FULL_ADMIN can only be set in the cluster configuration file,
# and they apply to the entire cluster. PACKAGE_ADMIN can be set in the
# cluster or a package configuration file. If set in the cluster
# configuration file, PACKAGE_ADMIN applies to all configured packages.
# If set in a package configuration file, PACKAGE_ADMIN applies to that
# package only.
#
# Conflicting or redundant policies will cause an error while applying
# the configuration, and stop the process. The maximum number of access
# policies that can be configured in the cluster is 200.
#
#
# Example: to configure a role for user john from node noir to
# administer a cluster and all its packages, enter:
# USER_NAME john
# USER_HOST noir
# USER_ROLE FULL_ADMIN

USER_NAME          root
USER_HOST          ANY_Serviceguard_NODE
USER_ROLE          full_admin

# List of cluster aware LVM Volume Groups. These volume groups will
# be used by package applications via the vgchange -a e command.
# Neither CVM or VxVM Disk Groups should be used here.
# For example:
# VOLUME_GROUP      /dev/vgdatabase
# VOLUME_GROUP      /dev/vg02

# List of OPS Volume Groups.
# Formerly known as DLM Volume Groups, these volume groups
# will be used by OPS or RAC cluster applications via
```

```
# the vgchange -a s command. (Note: the name DLM_VOLUME_GROUP
# is also still supported for compatibility with earlier versions.)
# For example:
# OPS_VOLUME_GROUP          /dev/vgdatabase
# OPS_VOLUME_GROUP          /dev/vg02
```

cmqueryc1 명령에 대한 맨페이지에는 이 파일에 나타나는 모든 매개 변수의 정의 목록이 있습니다. “계획” 장에서도 다양한 매개 변수에 대해 설명합니다. 클러스터 워크시트의 데이터를 사용하여 /etc/cmcluster/clust1.config 파일을 필요에 맞게 수정하십시오.

이 파일에서 키워드와 정의는 공백으로 구분됩니다. 주석을 사용할 수 있으며 주석의 맨 왼쪽 옆에는 파운드(#) 기호를 붙여야 합니다. 자세한 내용은 cmqueryc1 명령에 대한 맨페이지를 참조하십시오.

잠금 디스크 지정

클러스터 잠금은 이 예제에서처럼 두 개의 노드로 구성된 클러스터에 필요합니다. 잠금은 모든 노드에서 액세스할 수 있고 노드와는 별도로 전원을 공급 받아야 합니다. 자세한 내용은 3장의 “클러스터 잠금” 절을 참조하십시오. 클러스터 이름 다음에 잠금 디스크 정보를 입력합니다. 잠금 디스크는 클러스터의 모든 노드에서 액세스할 수 있는 LVM 볼륨 그룹에 포함되어야 합니다.

cmqueryc1로 만든 ASCII 템플릿에 제공된 기본 FIRST_CLUSTER_LOCK_VG 및 FIRST_CLUSTER_LOCK_PV는 최소 장애 조치 시간 계산을 기초로 선택된 디스크의 볼륨 그룹과 물리 볼륨 이름입니다. 이 디스크는 전원 배선 요구 사항을 충족해야 합니다. 필요한 경우 클러스터의 노드 중 절반 **미만**의 노드에 전원을 공급하는 회로에서 전원을 공급 받는 디스크를 선택합니다.

디스크 장애 조치 시간을 표시하려면 클러스터의 모든 노드를 지정하여 cmqueryc1 명령을 실행합니다. 이때 노드의 전체 도메인 이름을 포함시키면 안 됩니다. 예를 들어, *ftsys9.cup.hp.com*이 아니라 *ftsys9*로 지정합니다.

```
# cmqueryc1 -v -n ftsys9 -n ftsys10
```

명령의 출력에는 각 노드에 연결된 디스크와 각각에 연관된 재편성 시간이 함께 표시됩니다.

주

클러스터 구성에 특별히 필요한 경우가 아니면 두 번째 잠금 볼륨 그룹 또는 물리 볼륨을 구성하지 **않아야** 합니다. 자세한 내용은 3장의 “클러스터 잠금” 절에서 “이중 클러스터 잠금” 부분을 참조하십시오.

클러스터 구성에 두 번째 클러스터 잠금이 필요한 경우 클러스터 구성 파일에 다음과 같은 매개 변수를 입력합니다.

```
SECOND_CLUSTER_LOCK_VG /dev/volume-group  
SECOND_CLUSTER_LOCK_PV /dev/dsk/block-special-file
```

여기서 **/dev/volume-group**은 두 번째 볼륨 그룹의 이름이고 **block-special-file**은 선택한 볼륨 그룹에 있는 잠금 디스크의 물리 볼륨 이름입니다. 이 줄을 각 노드에 대해 추가해야 합니다.

쿼럼 서버 지정

잠금 디스크 대신 쿼럼 서버를 지정하려면 **cmquerycl** 명령의 **-q** 옵션을 사용하고 쿼럼 서버 호스트 서버를 지정합니다. 예를 들면 다음과 같습니다.

```
# cmquerycl -n ftsys9 -n ftsys10 -q qshost
```

이 경우 생성된 클러스터 **ASCII** 파일에는 쿼럼 서버를 정의하는 매개 변수가 들어 있습니다. 예를 들면 다음과 같습니다.

```
# Quorum Server Parameters. Use the QS_HOST, QS_POLLING_INTERVAL,  
# and QS_TIMEOUT_EXTENSION parameters to define a quorum  
server.  
# The QS_HOST is the host name or IP address of the system  
# that is running the quorum server process. The  
# QS_POLLING_INTERVAL (microseconds) is the interval at which  
# The optional QS_TIMEOUT_EXTENSION (microseconds) is used to  
increase  
# the time interval after which the quorum server is marked DOWN.  
#  
# The default quorum server interval is calculated from the  
# Serviceguard cluster parameters, including NODE_TIMEOUT and  
# HEARTBEAT_INTERVAL. If you are experiencing quorum server  
# timeouts, you can adjust these parameters, or you can include  
# the QS_TIMEOUT_EXTENSION parameter.  
#  
# For example, to configure a quorum server running on node  
# "qshost" with 120 seconds for the QS_POLLING_INTERVAL and to  
# add 2 seconds to the system assigned value for the quorum server
```

```
# timeout, enter:  
#  
# QS_HOST qshost  
# QS_POLLING_INTERVAL 120000000  
# QS_TIMEOUT_EXTENSION 2000000
```

QS_HOST 및 QS_POLLING_INTERVAL과 필요한 경우 QS_TIMEOUT_EXTENSION을 입력합니다.

하트비트 서버넷 지정

클러스터 ASCII 파일에는 하트비트 서버넷의 IP 주소에 대한 항목이 들어 있습니다. 전용 하트비트 서버넷을 사용하는 것이 좋지만 데이터 서버넷을 포함한 다른 서버넷에서도 하트비트를 구성할 수 있습니다.

하트비트는 IPv4 서버넷에 있어야 하며 IPv4 주소를 사용해야 합니다. IPv6 하트비트는 지원되지 않습니다.

주

VERITAS CVM 디스크 그룹을 사용하고 있는 경우 단일 하트비트 서버넷만 구성할 수 있으며 이는 전용 서버넷이어야 합니다. 가용성이 높은 하트비트 경로를 갖도록 이 서버넷의 각 시스템에는 대기 LAN을 구성해야 합니다.

구성된 최대 패키지 수 지정

Serviceguard에서는 클러스터가 시작될 때 메모리와 스레드를 미리 할당합니다. 이 값은 클러스터 구성 파일의 MAX_CONFIGURED_PACKAGES 매개 변수에 지정된 패키지 수에 따라 계산됩니다. 이 값은 현재 클러스터에 구성된 패키지 수보다 크거나 같아야 합니다. 기본값은 0이므로 패키지를 사용하려면 값을 입력해야 합니다. 절대적인 클러스터당 최대 패키지 수는 150입니다. Serviceguard에서는 잠금 가능 메모리에 6MB와 패키지당 100KB 정도의 메모리를 할당합니다. VERITAS CVM 디스크 저장 장치를 사용할 경우에는 MAX_CONFIGURED_PACKAGES 값을 선택할 때 CVM-VxVM-PKG를 MAX_CONFIGURED_PACKAGES의 총계에 포함시켜야 합니다.

주

각 노드에서 HP-UX 커널 매개 변수를 조정하여 해당 노드에서 동시에 실행될 최대 패키지 수에 충분하게 설정해야 합니다.

클러스터 타이밍 매개 변수 수정

cmquerycl 명령은 HEARTBEAT_INTERVAL과 NODE_TIMEOUT에 대한 기본 클러스터 타이밍 매개 변수를 제공합니다. 이 매개 변수를 변경하면 클러스터 재편성 및 장애 조치 시간에 직접적인 영향을 줍니다. 시스템 과부하 또는 네트워크 트래픽 과다로 인해 클러스터가 재편성되는 일이 종종 있다면 이 매개 변수를 수정하는 것이 유용합니다.

NODE_TIMEOUT의 기본값은 2초이며, 이 경우 최적 상황에서의 장애 조치 시간은 30초가 됩니다. NODE_TIMEOUT을 10초로 변경하면 노드 종료 시간까지 클러스터 관리자가 대기하는 시간이 5배 더 길어지므로 장애 조치 시간도 5배 증가하여 대략 150초가 됩니다. NODE_TIMEOUT은 적어도 2*HEARTBEAT_INTERVAL이 되어야 합니다. 일반적으로 하나의 NODE_TIMEOUT에 적어도 2~3개의 하트비트가 들어가게 하는 것이 좋습니다.

직렬 하트비트 연결 수 지정

직렬(RS232) 회선을 하트비트 연결로 사용할 경우 SERIAL_DEVICE_FILE 매개 변수를 사용하고 각 노드에서 사용할 직렬 포트에 해당하는 장치 파일 이름을 입력합니다. 구성하는 동안과 구성 후에 직렬 케이블이 안전하게 연결되어 있어야 합니다.

최적화

Serviceguard Extension for Faster Failover(SGeFF)는 별도로 구입하는 제품으로서, 이 제품이 설치되어 있으면 구성 파일에 제품을 활성화하기 위한 매개 변수가 표시됩니다.

SGeFF를 사용하면 Serviceguard에서 장애 조치를 처리하는 데 소요되는 시간을 줄일 수 있습니다. 그러나 패키지 및 응용 프로그램의 정상적인 종료 및 재시작에 소요되는 시간은 바뀌지 않습니다.

SGeFF에는 클러스터 구성 요구 사항이 있으며 이 요구 사항은 클러스터 구성 템플릿 파일에서 요약되어 있습니다.

자세한 내용은 <http://www.docs.hp.com/hpux/ha>에 게시된 “Serviceguard Extension for Faster Failover Release Notes”를 참조하십시오.

액세스 제어 정책

Serviceguard 버전 11.16의 새로운 기능인 액세스 제어 정책에서는 루트 권한이 없는 사용자가 일반적인 관리 명령을 사용할 수 있도록 합니다.

그래픽 사용자 인터페이스인 **Serviceguard Manager**의 루트 권한이 없는 사용자는 적절하게 구성된 액세스 정책이 있어야 **Serviceguard** 클러스터 및 패키지를 보고 관리할 수 있습니다. 새 구성에서 하나 이상의 모니터 액세스 정책을 즉시 구성하는 것이 좋습니다.

텍스트를 입력할 때, 특히 **ANY_USER**, **CLUSTER_MEMBER_NODE**와 같은 와일드카드를 입력할 때는 맞춤법을 확인합니다. 와일드카드의 철자가 잘못되면 해당 와일드카드가 특정 사용자나 노드를 나타내는 것으로 간주되어 구성하려는 액세스 정책을 구성하지 못할 수도 있습니다.

클러스터의 루트 사용자는 클러스터가 실행되는 동안 액세스 정책을 만들거나 수정할 수 있습니다.

볼륨 그룹 추가

클러스터에서 사용할 각 클러스터 인식 볼륨 그룹에 대해 개별적으로 **VOLUME_GROUP** 매개 변수를 사용하여, 구성된 **LVM** 볼륨 그룹을 **ASCII** 클러스터 구성 파일에 추가합니다. 이러한 볼륨 그룹은 **cmapplyconf** 명령을 사용할 때 클러스터 **ID**로 초기화됩니다. 또한 볼륨 그룹을 활성화하는 각 패키지 제어 스크립트에 적절한 볼륨 그룹, 논리 볼륨 및 파일 시스템 정보를 추가해야 합니다. 이 절차에 대해서는 6장에서 설명합니다.

주

CVM 디스크 그룹을 사용할 경우에는 클러스터 구성이 완료된 후 230페이지의 “**CVM**을 사용하여 저장 장치 하부 구조 만들기”에 설명된 절차를 따라 이 그룹을 구성해야 합니다. **VERITAS** 디스크 그룹은 6장에 설명된 것처럼 패키지 구성 파일에 추가됩니다.

클러스터 구성 확인

Serviceguard Manager에서 선택 단추를 클릭하여 구성을 확인합니다.

명령줄을 사용하여 **ASCII** 클러스터 구성 파일을 편집한 경우에는 다음 명령을 사용하여 파일의 내용을 확인합니다.

```
# cmcheckconf -k -v -C /etc/cmcluster/clust1.config
```

두 방법 모두 다음 사항을 확인할 수 있습니다.

- 네트워크 주소 및 연결
- 클러스터 잠금 연결(잠금 디스크를 구성하는 경우)
- 클러스터 및 패키지에 대한 구성 매개 변수의 유효성
- 이름의 고유성
- 명령줄에 지정된 스크립트의 존재 여부 및 권한
- 지정된 모든 노드가 동일한 하트비트 서브넷에 있는지 여부
- 잘못된 구성 파일 이름을 지정했는지 여부
- 모든 노드에 액세스할 수 있는지 여부
- CLUSTER_NAME, HEARTBEAT_INTERVAL 및 AUTO_START_TIMEOUT이 하나만 지정되었는지 여부
- 패키지 실행 및 중지 스크립트의 시간 종료 값이 4294초 미만인지 여부
- HEARTBEAT_INTERVAL의 값이 1초 이상인지 여부
- NODE_TIMEOUT의 값이 HEARTBEAT_INTERVAL 값의 두 배 이상인지 여부
- AUTO_START_TIMEOUT 변수의 값이 0보다 크거나 같은지 여부
- 하트비트 네트워크 최소 요구 사항. 클러스터에는 하나의 하트비트 LAN과 대기 LAN이 함께 구성되거나, 두 개의 하트비트 LAN이 구성되거나, 하나의 하트비트 LAN과 RS232 연결이 구성되거나, 로컬 전환 기능은 없지만 최소 두 개의 인터페이스의 링크 집합으로 구성된 주 LAN이 있는 하트비트 네트워크가 구성되어야 합니다.
- NODE_NAME을 하나 이상 지정했는지 여부
- 각 노드가 각 하트비트 네트워크에 연결되었는지 여부
- 모든 하트비트 네트워크의 LAN 유형이 동일한지 여부
- 지정된 네트워크 인터페이스 장치 파일이 유효한 LAN 장치 파일인지 여부
- 직렬(RS-232) 하트비트가 구성된 경우 클러스터의 노드 수가 두 개를 초과하지 않고 직렬(RS232) 포트 연결이 노드당 하나만 있는지 여부
- VOLUME_GROUP 항목이 현재 클러스터 인식 볼륨 그룹으로 표시되어 있지 않은지 여부
- CVM 디스크 저장 장치를 사용하고 있는 경우 하트비트 서브넷이 하나만 구성되어 있는지 여부

클러스터가 온라인 상태인 경우에는 특정 구성을 변경하는 데 필요한 모든 조건이 충족되었는지도 확인합니다.

주

-k 옵션을 사용하여 cmcheckconf를 실행하면 ASCII 파일에 지정된 LVM 디스크에 대한 디스크 연결만 확인됩니다. -k 옵션을 생략(기본 동작)하면 cmcheckconf는 모든 노드에 있는 LVM 디스크의 연결을 테스트합니다. -k를 사용하면 명령 실행 속도가 훨씬 빨라집니다.

이진 구성 파일 배포

모든 클러스터 매개 변수를 지정한 후에는 구성을 적용합니다. 그러면 클러스터의 모든 노드에 이진 구성 파일이 배포됩니다. 패키지 구성(다음 장 참조) 단계 **전에** 별도로 이 작업을 수행하는 것이 좋습니다. 이렇게 하면 실행 중인 클러스터에서 cmviewcl 명령을 사용하여 클러스터 잠금, 하트비트 네트워크, 기타 클러스터 수준의 작업을 확인할 수 있습니다. 구성을 배포하기 전에 보안 파일을 클러스터 노드 간에 복사할 수 있는지 확인하십시오. 자세한 내용은 이 장 맨 앞의 “시스템 준비”를 참조하십시오.

Serviceguard Manager를 사용하여 이진 파일 배포

정보를 모두 입력한 다음 적용을 클릭합니다.

명령줄에서 이진 파일 배포

다음 단계에 따라 이진 구성 파일을 생성하고 클러스터의 모든 노드에 구성을 배포할 수 있습니다.

- 잠금 디스크를 초기화할 수 있도록 클러스터 잠금 볼륨 그룹을 활성화합니다.

```
# vgchange -a y /dev/vglock
```

- 이진 구성 파일을 만들어 배포합니다.

```
# cmapplyconf -k -v -C /etc/cmcluster/clust1.config
```

주

-k 옵션을 사용하여 cmapplyconf를 실행하면 ASCII 파일에 지정된 LVM 디스크에 대한 디스크 연결만 확인됩니다. -k 옵션을 생략(기본 동작)하면 cmapplyconf는 모든 노드에 있는 LVM 디스크의 연결을 테스트합니다. -k를 사용하면 명령 실행 속도가 훨씬 빨라집니다.

-
- 클러스터 잠금 볼륨 그룹을 비활성화합니다.

```
# vgchange -a n /dev/vglock
```

cmapplyconf 명령은 이전 버전의 클러스터 구성 파일을 만들어 클러스터의 모든 노드에 배포합니다. 이 작업을 통해 모든 노드에서 파일 내용이 일관되게 유지됩니다. cmapplyconf 명령을 사용할 경우 ASCII 구성 파일은 배포되지 않습니다.

주의

클러스터 잠금 볼륨 그룹은 구성을 적용하기 전에 정확히 한 노드에서만 활성화되고 구성을 적용한 후에는 비활성화되어야 합니다.

잠금 볼륨 그룹이 하나의 노드가 아니라 둘 이상의 노드에서 활성화된 경우에는 적용이 완료되지 않습니다.

cmapplyconf가 실행된 후에는 구성 노드에서 클러스터 잠금 볼륨 그룹을 비활성화해야 합니다.

볼륨 그룹 및 클러스터 잠금 구성 데이터 저장

클러스터를 구성한 다음에는 만든 각 볼륨 그룹에 대해 vgcfgbackup 명령을 사용하여 LVM 볼륨 그룹 구성을 백업합니다. 이렇게 하면 볼륨 그룹의 디스크를 교체해야 하는 경우 vgcfgrestore 명령을 사용하여 디스크의 메타데이터를 복원할 수 있습니다.

이 방법에 대해서는 “문제 해결” 장의 “디스크 교체” 절을 참조하십시오.

클러스터 잠금 볼륨 그룹을 비롯한 모든 볼륨 그룹에 대해 vgcfgbackup을 사용해야 합니다.

주

볼륨 그룹을 만들 때 **SAM**을 사용했는지 또는 **HP-UX** 명령을 사용했는지에 상관없이 **반드시** `vgcfgbackup` 명령을 사용하여 클러스터 잠금 디스크의 구성 데이터 사본을 저장해야 합니다.

클러스터 실행 중에 클러스터 잠금 디스크를 교체해야 하는 경우에는 **반드시** `vgcfgrestore` 명령을 사용하여 잠금 정보를 교체 디스크에 복원해야 합니다. 그렇지 않으면 잠금 디스크의 모든 중복 사본에 장애가 발생하고 교체 메커니즘이나 **LUN**에서 잠금 구성을 복원하지 못할 경우 클러스터 전체에 장애가 발생할 수 있습니다. 클러스터 잠금 디스크가 디스크 어레이에 구성되어 있는 경우에는 **RAID** 보호 기능이 클러스터 잠금 데이터의 중복 사본을 제공합니다. **MirrorDisk/UX**는 클러스터 잠금 정보를 미러링하지 않습니다.

CVM을 사용하여 저장 장치 하부 구조 만들기

클러스터를 구성하는 것 외에도 다른 노드에서 데이터에 액세스할 수 있도록 적절한 논리 볼륨 하부 구조를 만들어야 합니다. 이 작업은 Logical Volume Manager(LVM), VERITAS Volume Manager(VxVM) 또는 VERITAS Cluster Volume Manager(CVM)를 통해 수행합니다. 필요에 따라 볼륨 유형을 혼합하여 사용할 수도 있습니다. LVM 및 VxVM 구성은 클러스터 구성 이전에 수행하며 CVM 구성은 클러스터 구성 후에 수행합니다.

LVM에서 VxVM 또는 CVM 저장 장치로 마이그레이션하는 방법에 대한 자세한 내용은 부록 H를 참조하십시오.

이 절에서는 VERITAS Cluster Volume Manager(CVM)에 설정된 명령을 사용하여 저장 장치를 구성하는 방법에 대해 설명합니다. 시작하기 전에 VxVM 명령이 저장된 디렉토리(/usr/lib/vxvm/bin)가 경로에 있는지 확인합니다. vxinstall을 사용하여 루트 디스크 그룹을 만들었으면 VxVM 명령이나 VERITAS Storage Administrator GUI인 vmsa를 사용하여 구성 작업을 수행할 수 있습니다. vmsa를 사용하는 경우에는 GUI를 실행하기 전에 Storage Administrator 서버가 실행 중인지 확인합니다. vxinstall 실행 방법에 대한 자세한 내용은 VERITAS Volume Manager 3.5 Release Notes를 참조하십시오. 자세한 내용은 VERITAS Volume Manager 3.5 Administrator's Guide를 참조하십시오.

작업 절차는 다음과 같이 구분할 수 있습니다.

- 루트 디스크 그룹 만들기
- 클러스터를 CVM과 함께 사용할 수 있도록 준비
- 공유 저장 장치용 디스크 그룹 만들기

Plex(미러), 다중 경로 지정 및 RAID 구성 방법 등에 대한 자세한 내용은 VERITAS Volume Manager에 대한 HP-UX 설명서를 참조하십시오.

VERITAS Volume Manager 초기화

디스크 그룹을 처음 만드는 경우 Volume Manager를 초기화해야 합니다. 하나 이상의 디스크를 포함하는 rootdg라는 디스크 그룹을 만들어 이 작업을 수행합니다. 각 노드에서 VxVM/CVM을 설치한 후 다음 명령을 실행합니다.

vxinstall

이 명령을 사용하면 VxVM/CVM 초기화 과정을 단계별로 안내하는 메뉴 방식의 프로그램이 표시됩니다. 주 메뉴에서 “Custom” 옵션을 선택하고 **rootdg**에 포함할 디스크를 지정합니다.

중요

VERITAS Volume Manager의 **rootdg**는 HP-UX 루트 파일 시스템(/)에서 LVM 볼륨 그룹을 사용하는 경우 HP-UX 루트 디스크와 동일하지 않습니다. 또한 **rootdg**는 공유 저장 장치에 사용할 수 없습니다. 그러나 **rootdg**는 다른 로컬 파일 시스템(예: /export/home)에 사용할 수 있으므로 유용하게 쓰입니다.

루트 디스크 그룹은 각 노드에서 한 번만 만들어야 합니다.

클러스터를 CVM과 함께 사용할 수 있도록 준비

VERITAS Cluster Volume Manager(CVM)를 사용하려면 SMP(시스템 다중 노드 패키지)라는 특수 CVM 패키지와 함께 실행되는 클러스터가 필요합니다. 즉, 디스크 그룹을 만들기 전에 클러스터를 미리 구성하여 실행하고 있어야 합니다.

Serviceguard Manager를 통해 SMP 패키지를 구성할 수는 없습니다. 그러나 SMP를 구성한 후에는 클러스터 노드 중 하나에 루트로 로그인하면 클러스터의 구성을 수정할 수 있습니다. 모니터링 액세스 역할이 있으면 SMP 패키지를 볼 수 있습니다. 전체 관리 액세스 역할이 있는 경우에는 Serviceguard Manager에서 SMP 패키지를 중지하거나 시작할 수 있습니다.

주

클러스터 구성은 앞의 절에 설명되어 있습니다.

CVM 디스크 그룹용 클러스터 구성을 준비하려면 클러스터 ASCII 구성 파일에서 MAX_CONFIGURED_PACKAGES를 1 이상으로 설정하고 하트비트 서브넷은 하나만 구성합니다. 그런 다음 클러스터 정보를 CVM에 전달하는 특수 패키지를 만드는 다음 명령을 실행합니다.

```
# cmapplyconf -P /etc/cmcluster/cvm/VxVM-CVM-pkg.conf
```

경고

VxVM-CVM-pkg.conf 파일은 편집하면 안 됩니다.

이 명령이 성공적으로 완료되면 공유용으로 디스크 그룹을 만들 수 있습니다(다음 절 참조). 또한 클러스터는 모든 노드에 있는 VxVM-CVM-pkg라는 특수한 **시스템 다중 노드 패키지**와 함께 실행됩니다. 이 패키지는 다음과 같은 `cmviewcl` 명령의 출력에서 볼 수 있습니다.

```

CLUSTER      STATUS
example      up

NODE          STATUS      STATE
ftsys7       up          running
ftsys8       up          running
ftsys9       up          running
ftsys10      up          running

SYSTEM_MULTI_NODE_PACKAGES:

PACKAGE      STATUS      STATE
VxVM-CVM-pkg up          running

```

클러스터 시작 및 마스터 노드 확인

클러스터가 실행되고 있지 않으면 클러스터를 시작합니다. 그러면 특수 CVM 패키지가 활성화됩니다.

```
# cmruncl
```

CVM이 시작되면 마스터 노드가 선택됩니다. 이 노드에서 디스크 그룹 구성 명령을 실행해야 합니다. 마스터 노드를 확인하려면 클러스터의 각 노드에서 다음 명령을 실행합니다.

```
# vxdctl -c mode
```

하나의 노드가 마스터 노드로 표시됩니다. 이 노드에서 디스크 그룹을 만듭니다.

CVM용 디스크 초기화

CVM 디스크 그룹에 사용할 물리 디스크를 초기화해야 합니다. 이전에 물리 디스크를 LVM과 함께 사용한 경우 `pvremove` 명령을 사용하여 볼륨 그룹의 모든 디스크에서 LVM 헤더 데이터를 삭제해야 합니다. 이전에 디스크를 LVM과 함께 사용한 적이 없는 경우에는 이 작업이 필요하지 않습니다.

CVM에 사용할 디스크를 초기화하려면 마스터 노드에 로그인한 다음 `vxdiskadm` 프로그램을 사용하여 여러 디스크를 한 번에 초기화하거나 `vxdisksetup` 명령을 사용하여 디스크를 한 번에 하나씩 초기화합니다. 예를 들면 다음과 같습니다.

```
# /usr/lib/vxvm/bin/vxdisksetup -i c0t3d2
```

디스크 그룹 만들기

디스크 그룹을 만드는 단계는 다음과 같습니다.

1. `vxldg` 명령을 사용하여 디스크 그룹을 만듭니다. 다음 예제처럼 `-s` 옵션을 사용하여 공유 모드를 지정합니다.

```
# vxldg -s init logdata c0t3d2
```

2. 다음 명령으로 구성을 확인합니다.

```
# vxldg list
```

| NAME | STATE | ID |
|---------|-----------------|----------------------|
| rootdg | enabled | 971995699.1025.node1 |
| logdata | enabled, shared | 972078742.1084.node2 |

3. 볼륨을 만들기 전에 다음과 같이 디스크 그룹을 활성화합니다.

```
# vxldg -g logdata set activation=ew
```

볼륨 만들기

다음 예제와 같이 `vxassist` 명령을 사용하여 논리 볼륨을 만듭니다.

```
# vxassist -g logdata make log_files 1024m
```

이 명령은 `logdata`라는 디스크 그룹에 `log_files`라는 1024MB 크기의 볼륨을 만듭니다. 이 볼륨은 블록 장치 파일 `/dev/vx/dsk/logdata/log_files` 또는 원시(문자) 장치 파일 `/dev/vx/rdsk/logdata/log_files`를 사용하여 참조할 수 있습니다.

다음 명령으로 구성을 확인합니다.

```
# vxvg list
```

CVM의 미러 분리 정책

기본 CVM 디스크 미러 분리 정책은 ‘전체적’으로 적용됩니다. 즉, 한 노드가 특정 미러 사본(plex)을 볼 수 없게 되면 즉시 모든 노드에서도 해당 사본을 볼 수 없게 됩니다. 반면, 대체 정책은 ‘국지적’으로 적용되므로 한 노드가 특정 미러 사본을 볼 수 없게 되면 CVM에서는 해당 노드에 대해서만 볼륨 액세스를 비활성화합니다. 이 정책은 다음과 같이 `vxedit` 명령을 사용하여 디스크 그룹별로 다시 설정할 수 있습니다.

```
# vxedit set diskdetpolicy=[global|local] <DiskGroupName>
```

주

CVM을 사용하는 미러링되는 다중 경로 저장 장치를 만들기 위한 특정 명령에 대해서는 VERITAS Volume Manager에 대한 HP-UX 설명서를 참조하십시오.

파일 시스템 만들기

설치 프로그램에서 파일 시스템을 사용하는 경우에는 논리 볼륨을 만든 후 파일 시스템을 만듭니다. 다음 명령을 사용하여 방금 만든 논리 볼륨에 마운트할 파일 시스템을 만듭니다.

1. 새로 만든 볼륨에 파일 시스템을 만듭니다.

```
# newfs -F vxfs /dev/vx/rdisk/logdata/log_files
```

2. 볼륨을 마운트할 디렉토리를 만듭니다.

```
# mkdir /logs
```

3. 볼륨을 마운트합니다.

```
# mount /dev/vx/dsk/logdata/log_files /logs
```

4. 파일 시스템이 있는지 확인한 다음 파일 시스템을 마운트 해제합니다.

```
# umount /logs
```

5. 다음 명령을 사용하여 디스크 그룹을 비활성화합니다.

```
# vxvg -g logdata set activation=off
```

패키지 구성에 디스크 그룹 추가

VxVM 명령으로 저장 장치를 만든 후에는 각 패키지 구성 ASCII 파일에서 CVM 디스크 그룹을 지정해야 합니다. 패키지에서 사용할 각 디스크 그룹마다 하나의 DISK_GROUP 매개 변수를 사용합니다. 또한 패키지 제어 스크립트에서 CVM 디스크 그룹, 파일 시스템, 논리 볼륨 및 마운트 옵션을 지정해야 합니다. 패키지 구성 과정에 대한 자세한 내용은 6장에서 설명합니다.

주

CVM 디스크 그룹은 LVM 볼륨 그룹과 달리 클러스터 ASCII 구성 파일에 입력되지 않습니다.

실행 중인 클러스터 관리

이 절에서는 일상적인 클러스터 관리 방법 몇 가지를 설명합니다. 추가 도구 및 방법은 7장의 “클러스터 및 패키지 유지 관리”를 참조하십시오.

Serviceguard Manager를 사용하여 클러스터 작동 검사

Serviceguard Manager에서는 클러스터 내의 모든 노드와 패키지를 표시하고 현재 상태를 보여 줍니다. 자세한 내용은 7장의 “Serviceguard Manager 사용” 절을 참조하십시오. 다음과 같이 Serviceguard Manager를 사용하여 구성 및 상태 정보를 확인할 수 있습니다.

- 구성된 모든 노드가 실행 중인지 확인할 수 있습니다.
- 구성된 모든 패키지가 실행 중인지 확인하고 패키지가 실행되는 노드를 확인할 수 있습니다.
- 클러스터, 노드 및 패키지의 속성 시트에서 자세한 정보를 볼 수 있습니다.

패키지 또는 클러스터 구성을 만들거나 수정할 때는 클러스터를 시작하여 실행하고 구성을 Serviceguard Manager 파일(.sgm)로 보관할 수 있습니다. 이 파일의 데이터를 나중에 변경된 클러스터와 비교하면 그 동안의 변경 내용을 알 수 있으므로 클러스터에 문제가 발생할 때 이 파일을 참조하여 문제를 해결할 수 있습니다.

세션 서버 노드와 대상 노드 모두에 Serviceguard 버전 A.11.12 이상이 설치되어 있으면 Serviceguard Manager를 통해 몇 가지 관리 명령을 사용할 수 있습니다.

Serviceguard 명령을 사용하여 클러스터 작동 검사

Serviceguard에는 클러스터를 제어하기 위한 몇 가지 명령이 포함되어 있습니다.

- `cmviewcl`는 클러스터의 상태와 클러스터의 여러 구성 요소를 확인하는 데 사용됩니다. 모니터링 역할이 있지만 루트 권한은 없는 사용자는 클러스터 노드에서 이 명령을 실행하거나 Serviceguard Manager에서 상태 정보를 볼 수 있습니다.

- `cmrunnode`는 노드를 시작하는 데 사용됩니다. 전체 관리 역할이 있지만 루트 권한은 없는 사용자는 클러스터 노드에서 또는 **Serviceguard Manager**를 통해 이 명령을 실행할 수 있습니다.
- `cmhaltnode`는 실행 중인 노드를 수동으로 중지하는 데 사용됩니다. 이 명령은 `shutdown(1m)`에서도 사용됩니다. 전체 관리 역할이 있지만 루트 권한은 없는 사용자는 클러스터 노드에서 또는 **Serviceguard Manager**를 통해 이 명령을 실행할 수 있습니다.
- `cmruncl`은 중지된 클러스터를 수동으로 시작하는 데 사용됩니다. 전체 관리 액세스 권한이 있지만 루트 권한은 없는 사용자는 클러스터 노드에서 또는 **Serviceguard Manager**를 통해 이 명령을 실행할 수 있습니다.
- `cmhaltcl`은 클러스터를 수동으로 중지하는 데 사용됩니다. 전체 관리 액세스 권한이 있지만 루트 권한은 없는 사용자는 클러스터 노드에서 또는 **Serviceguard Manager**를 통해 이 명령을 실행할 수 있습니다.

다음과 같이 이 명령을 사용하여 클러스터 작동을 테스트할 수 있습니다.

1. 클러스터가 아직 온라인 상태가 아니면 클러스터를 시작합니다. **Serviceguard Manager** 메뉴에서 클러스터 실행을 선택합니다. 명령줄에서 `cmruncl -v`를 실행합니다.

기본적으로 `cmruncl`은 네트워크를 검사합니다. **Serviceguard**에서는 클러스터 구성에 포함된 네트워크 정보를 사용하여 실제 네트워크 구성을 조사합니다. 이 구성 확인 과정이 필요하지 않으면 대신 `cmruncl -v -w none` 옵션을 사용하여 확인 기능을 해제하고 시간을 절약할 수 있습니다.

2. 클러스터가 시작되었으면 클러스터 구성 요소가 올바르게 작동하는지 확인합니다. **Serviceguard Manager**의 맵 또는 트리에서 클러스터를 열고 해당 속성을 확인합니다. 명령줄에서 `cmviewcl -v` 명령을 실행합니다.

모든 노드와 네트워크가 예상대로 작동하는지 확인합니다. 자세한 내용은 “클러스터 및 패키지 유지 관리” 장을 참조하십시오.

3. 다음 단계에 따라 노드가 예상대로 클러스터에 참가하고 클러스터에서 나가는지 확인합니다.
 - 클러스터를 중지합니다. **Serviceguard Manager** 메뉴에서 클러스터 중지를 실행합니다. 명령줄에서 `cmhaltnode` 명령을 실행합니다.

- 맵 또는 트리에서 클러스터 구성원을 검사하여 노드가 클러스터에서 나갔는지 확인합니다. **Serviceguard Manager**에서 맵 또는 트리나 클러스터 속성을 엽니다. 명령줄에서 `cmviewcl` 명령을 실행합니다.
 - 노드를 시작합니다. **Serviceguard Manager**에서는 노드 실행 명령을 실행하고 명령줄에서는 `cmrunnode` 명령을 실행합니다.
 - 노드가 다시 작동하는지 확인하려면 **Serviceguard Manager**의 맵 또는 트리를 확인하거나 `cmviewcl` 명령을 다시 실행합니다.
4. 클러스터를 종료합니다. **Serviceguard Manager**에서 클러스터 중지 명령을 실행합니다. 명령줄에서 `cmhaltcl -v -f` 명령을 실행합니다.

클러스터 테스트에 대한 자세한 내용은 “문제 해결” 장을 참조하십시오.

Serviceguard 명령의 전체 목록은 부록 A를 참조하십시오. **Serviceguard** 관리 명령의 목록은 **Serviceguard Manager** 도움말을 참조하십시오.

볼륨 그룹의 자동 활성화 방지

`/etc/lvmrc` 파일을 사용하여 시스템 부팅 시 패키지에 사용될 **LVM** 볼륨 그룹이 자동으로 활성화되도록 하지 않도록 해야 합니다. 자동 활성화를 방지하려면 모든 노드에서 `/etc/lvmrc` 파일을 편집합니다. `AUTO_VG_ACTIVATE`를 0으로 설정한 다음 클러스터가 바인딩되지 않은 모든 볼륨 그룹을 `custom_vg_activation` 함수에 포함시킵니다. 패키지에서 사용할 볼륨 그룹은 제어 스크립트에 의해 활성화되거나 비활성화되므로 이 파일의 어느 부분에도 포함되면 **안 됩니다**.

주

루트 볼륨 그룹은 부팅 시 `/etc/lvmrc` 파일이 사용되기 전에 자동으로 활성화되므로 `custom_vg_activation` 함수에 포함할 필요가 없습니다.

자동 시작 기능 설정

자동 시작은 각 노드가 개별적으로 클러스터에 참가하는 과정이며, **Serviceguard**에서는 이 시작 과정을 제어하는 시작 스크립트를 제공합니다. 자동 클러스터 시작은 기본적인 클러스터 시작 방법입니다. 이 경우 시스템 관리자는 아무 작업도 할 필요가 없습니다.

클러스터가 자동으로 시작되는 경우에는 다음 세 가지가 있습니다.

- 클러스터가 어떤 노드에서도 실행되지 않고 있습니다. 단, 모든 클러스터 노드가 연결 가능해야 하고, 모든 노드에서 클러스터를 시작하려고 시도해야 합니다. 이 경우 노드는 구성된 모든 노드로 클러스터를 구성하려고 시도합니다.
- 클러스터가 하나 이상의 노드에서 이미 실행 중입니다. 이 경우 노드는 해당 클러스터에 참가하려고 시도합니다.
- 위의 두 경우 외에, 클러스터가 어떤 노드에서도 실행되지 않고 있으며 일부 노드는 연결 가능하지 않고 클러스터를 시작하려고 시도하지 않습니다. 이 경우 노드에서는 **AUTO_START_TIMEOUT** 기간 동안 클러스터를 시작하려고 시도합니다. 이 기간 동안에도 위의 조건이 충족되지 않으면 자동 시작이 실패하게 됩니다.

자동 클러스터 시작을 사용하려면 클러스터에 있는 각 노드의 `/etc/rc.config.d/cmcluster` 파일에서 **AUTOSTART_CMCLD** 플래그를 1로 설정합니다. 그러면 노드가 부팅 시 클러스터에 참가하게 됩니다.

다음은 `/etc/rc.config.d/cmcluster` 파일의 예입니다.

```
***** CMCLUSTER *****
# Highly Available Cluster configuration
#
# @(#) $Revision: 72.2 $
#
# AUTOSTART_CMCLD:    If set to 1, the node will attempt to
#                    join it's CM cluster automatically when
#                    the system boots.
#                    If set to 0, the node will not attempt
#                    to join it's CM cluster.
#
AUTOSTART_CMCLD=1
```

시스템 메시지 변경

다음과 같은 내용이 포함되도록 시스템의 로그인 메시지를 수정하면 유용합니다.

This system is a node in a high availability cluster.
Halting this system may cause applications and services to
start up on another node in the cluster.

추가 클러스터별 정보와 함께 이 메시지에 모든 클러스터 노드 목록을 포함시킬 수도
있습니다.

`/etc/issue` 및 `/etc/motd` 파일을 사용자 지정하여 클러스터 관련 정보를 포함할 수 있습
니다.

단일 노드 클러스터 관리

Serviceguard 클러스터에 필요한 노드 수는 보호할 응용 프로그램의 처리 요구 사항
에 따라 달라집니다. 단일 노드 클러스터의 경우 **Serviceguard**의 네트워크 장애 보호
기능을 사용하도록 구성할 수 있습니다.

단일 노드 클러스터에서는 클러스터에 다른 노드가 없으므로 클러스터 잠금이 필요하
지 않습니다. 노드가 하나뿐이면 `cmqueryc1` 명령의 출력에서 클러스터 잠금 정보 영
역은 생략됩니다.

중복 네트워크가 필요할 수도 있지만 하트비트를 보낼 다른 노드가 없으므로 하트비트
LAN을 지정할 필요가 없습니다. **Serviceguard**에서 모니터링할 모든 **LAN**을 클러스
터 구성 **ASCII** 파일에 지정합니다. **IP** 주소가 이미 있는 **LAN**의 경우 **HEARTBEAT_IP**
키워드 대신 **STATIONARY_IP** 키워드를 사용하여 지정합니다. 대기 **LAN**의 경우에는
LAN 장치 이름으로 **NETWORK_INTERFACE** 키워드만 지정하면 됩니다.

단일 노드 작동

단일 노드 작동은 단일 노드 클러스터에서 발생하거나, 다중 노드 클러스터에서 한 노
드를 제외한 모든 노드에 장애가 발생할 경우 또는 한 노드를 제외한 모든 노드를 종료
한 경우에 발생합니다. 이때 실행 중인 응용 프로그램이 있을 수 있습니다.

Serviceguard 데몬 `cmclsd`가 활성 상태에 있으면 다른 노드는 나중에 클러스터에 다
시 참가할 수 있습니다.

단일 노드 작동 상태에서 **Serviceguard** 데몬에 장애가 발생하는 경우 단일 노드와 응용
프로그램은 계속 실행됩니다. 이것은 다중 노드 클러스터에서 **Serviceguard** 데몬에
장애가 발생하는 경우와 다른 점입니다. 다중 노드 클러스터에서는 **TOC**와 함께 노드
가 중지되고 패키지는 대체 노드로 전환됩니다.

이 시나리오에서는 응용 프로그램이 계속 실행 중이고 현재 패키지를 전환할 수 있는
다른 노드가 없으므로 단일 노드를 중지할 필요가 없습니다.

그러나 **Serviceguard**를 다시 시작해서는 **안 됩니다**. 노드에서 계속 실행 중인 응용 프로그램의 새 인스턴스를 해당 노드에서 시작하려고 시도하면 데이터가 손상될 수 있기 때문입니다. 따라서 클러스터를 다시 시작하지 말고 적당한 때 노드를 종료한 다음 다시 시작하십시오. 이렇게 하면 응용 프로그램이 종료되고 노드가 다시 부팅되었을 때 **Serviceguard**에서 클러스터를 다시 시작할 수 있습니다.

클러스터 구성 삭제

루트로 로그인하면 **Serviceguard Manager**나 명령줄을 사용하여 모든 클러스터 노드에서 클러스터 구성을 삭제할 수 있습니다. `cmdeleteconf` 명령을 실행할 때 `-f` 옵션을 사용하지 않으면 파일을 삭제하기 전에 이를 확인하는 메시지가 표시됩니다. 클러스터가 작동 중지되었을 때만 구성을 삭제할 수 있습니다. 이 작업을 수행하면 클러스터의 모든 노드에서 이전 구성 파일이 제거되고 모든 클러스터 인식 볼륨 그룹이 더 이상 클러스터를 인식하지 않도록 다시 설정됩니다.

주

`cmdeleteconf` 명령은 클러스터 이진 파일 `/etc/cmcluster/cmclconfig`만 제거하고 `/etc/cmcluster` 디렉토리의 다른 파일은 제거하지 **않습니다**.

`cmdeleteconf` 명령을 사용하려면 클러스터는 중지되어야 하지만 클러스터의 모든 노드는 전원이 계속 공급되고 액세스 가능한 상태여야 합니다. 노드의 전원이 꺼져 있으면 전원을 켜고 부팅하십시오. 노드에 액세스할 수 없으면 다음과 같은 메시지와 함께 액세스할 수 없는 노드 목록이 표시됩니다.

```
It is recommended that you do not proceed with the configuration
operation unless you are sure these nodes are permanently unavailable.
Do you want to continue?
```

구성을 제거하려면 **Yes**로 응답합니다. 액세스할 수 없는 노드가 나중에 액세스할 수 있게 되면 해당 노드에서 `cmdeleteconf` 명령을 실행하여 구성 파일을 제거해야 합니다.

HA 클러스터 구성 만들기
실행 중인 클러스터 관리

6

패키지 및 서비스 구성

클러스터를 구성한 다음에는 패키지로 그룹화할 응용 프로그램과 서비스를 지정해야 합니다. 이 장에서는 명령줄에서 패키지를 만들기 위한 다음의 **패키지 구성** 작업에 대해 설명합니다.

- 패키지 구성 만들기
- 패키지 제어 스크립트 작성
- 패키지 구성 확인
- 구성 배포

이 작업은 각각 별도의 절에서 설명합니다.

Serviceguard Manager에서 패키지와 패키지 제어 스크립트를 만들 수도 있습니다. 유도 모드에서는 프로세스를 단계별로 안내해 줍니다. **Serviceguard Manager**에서 만든 패키지는 **Serviceguard Manager**에서 수정할 수 있습니다.

패키지를 구성할 때는 “계획” 장에서 설명하는 패키지 구성 워크시트의 데이터를 사용하십시오. 워크시트의 패키지 구성 데이터는 클러스터의 모든 노드에 있는 이진 클러스터 구성 파일의 일부가 됩니다. 워크시트의 제어 스크립트 데이터는 특정 응용 프로그램을 실행하고 해당 응용 프로그램의 작업을 모니터링하는 실행 가능한 패키지 제어 스크립트에 사용됩니다.

Serviceguard Manager에서는 워크시트에 있는 대부분의 정보를 자동으로 찾습니다.

패키지 구성 만들기

패키지 구성은 패키지가 클러스터의 한 노드에서 시작될 때 패키지 관리자에 의해 실행되는 일련의 응용 프로그램 서비스를 정의하는 과정입니다. 또한 이 구성에는 패키지가 실행될 수 있는 클러스터 노드와 해당 패키지에 대해 적용할 수 있는 장애 조치 유형의 정의를 우선 순위별로 정리한 목록도 포함됩니다.

Serviceguard Manager를 사용하거나 **HP-UX** 명령 및 편집기를 사용하여 패키지를 만들 수 있습니다. 다음 절에서는 **Serviceguard Manager** 구성에 대해 설명합니다. **Serviceguard** 명령줄을 사용할 경우에는 “**Serviceguard** 명령을 사용하여 패키지 만들기” 절로 건너뛰십시오.

Serviceguard Manager를 사용하여 패키지 구성

고가용성 패키지를 구성하려면 구성 노드(fts9)에서 다음 단계를 따르십시오.

Serviceguard 버전 11.16 이상에서는 **Serviceguard Manager**를 사용하여 패키지를 구성할 수 있습니다.

1. **Serviceguard** 버전 11.16이 설치된 세션 서버 노드에 연결합니다. 버전 11.16 이상이 설치된 클러스터를 찾습니다. 구성을 시작하기 전에 대상 클러스터의 노드에 대한 루트 암호를 입력하라는 메시지가 표시됩니다.
2. 패키지를 만들려면 맵 또는 트리에서 클러스터를 선택합니다. 동작 메뉴에서 구성 -> 패키지 작성을 선택합니다. 패키지를 수정하려면 패키지를 선택하고 구성 -> 패키지 <pkgname> 수정을 선택합니다.
3. 패키지 구성 및 패키지 제어 스크립트 만들기: 이 인터페이스는 과정을 단계별로 안내해 줍니다. 온라인 도움말은 구성 중에도 사용할 수 있습니다. 대부분의 단계는 순서대로 수행하지 않아도 되므로 아래의 단계별 구성에 대한 지침은 단지 참조로 사용할 수 있습니다. 원하는 경우에는 제어 스크립트를 자동으로 만들 수 있습니다.
4. 패키지 구성 확인: 선택 단추를 클릭합니다. 로그 창이 표시되지 않으면 보기 메뉴에서 로그 창을 엽니다.
5. 구성 배포: 적용을 클릭합니다. 이진 구성 파일이 만들어진 다음 이 이진 파일과 생성된 제어 스크립트가 패키지의 노드에 배포됩니다.

원하는 경우 제어 스크립트를 직접 구성할 수 있습니다. 이 작업은 표준 제어 스크립트를 사용하지 않는 경우에 필요할 수 있습니다. 그러나 제어 스크립트를 직접 편집한 후에는 **Serviceguard Manager**에서 다시 보거나 수정할 수 없습니다. 제어 스크립트를 편집할 경우에는 제어 스크립트 배포 작업도 사용자가 직접 수행해야 합니다.

자세한 내용은 245페이지의 “단계별 구성”을 참조하십시오.

Serviceguard 명령을 사용하여 패키지 구성

다음 절차에 따라 패키지 구성 파일을 편집하고 처리하여 패키지를 만드십시오.

1. 먼저, `/etc/cmcluster` 디렉토리에 구성할 각 패키지의 하위 디렉토리를 만듭니다.

```
# mkdir /etc/cmcluster/pkg1
```

원하는 어떠한 디렉토리 이름도 사용할 수 있습니다.

2. 그런 다음 해당 패키지의 패키지 구성 템플릿을 생성합니다.

```
# cmmakepkg -p /etc/cmcluster/pkg1/pkg1.config
```

ASCII 템플릿에 원하는 어떠한 파일 이름도 사용할 수 있습니다.

3. 이 템플릿 파일을 편집하여 각 패키지에 대한 패키지 이름, 우선 순위별 노드 목록 (이름은 31바이트 이하로 지정), 제어 스크립트의 위치 및 장애 조치 매개 변수를 지정합니다. 패키지 구성 워크시트에 기록된 데이터를 포함시키십시오.

단계별 구성

다음과 같이 단계별로 클러스터의 패키지를 구성하는 것이 좋습니다.

1. 볼륨 그룹과 마운트 지점만 구성합니다.
2. 구성을 적용합니다.
3. 모든 노드에 제어 스크립트를 배포합니다.
4. 패키지를 실행하고 노드 간에 패키지가 이동되는지 확인합니다.
5. 패키지를 중지합니다.
6. 제어 스크립트에서 패키지 IP 주소 및 응용 프로그램 서비스를 구성합니다.

7. 모든 노드에 제어 스크립트를 배포합니다.
8. 패키지를 실행한 다음 응용 프로그램이 예상대로 실행되고 서비스가 중단되었을 때 패키지의 장애 조치가 올바르게 실행되는지 확인합니다.

패키지 구성 템플릿 파일

다음은 일반적인 패키지에 대해 사용자 정의된 샘플 패키지 구성 파일 템플릿입니다.

패키지 구성 워크시트의 정보를 사용하여 이 파일을 완성하십시오. 또한 각 매개 변수에 대한 추가 설명을 보려면 구성 템플릿의 주석을 참조하십시오. 다음 정보를 포함할 수 있습니다.

```
# *****
# ***** HIGH AVAILABILITY PACKAGE CONFIGURATION FILE (template) *****
# *****
# ***** Note: This file MUST be edited before it can be used. *****
# * For complete details about package parameters and how to set them, *
# * consult the Serviceguard Extension for RAC manuals.
# *****

# Enter a name for this package. This name will be used to identify the
# package when viewing or manipulating it. It must be different from
# the other configured package names.

PACKAGE_NAME

# Enter the package type for this package. PACKAGE_TYPE indicates
# whether this package is to run as a FAILOVER or SYSTEM_MULTI_NODE
# package.
#
#     FAILOVER      package runs on one node at a time and if a failure
#                   occurs it can switch to an alternate node.
#
#     SYSTEM_MULTI_NODE
#                   package runs on multiple nodes at the same time.
#                   It can not be started and halted on individual nodes.
#                   Both NODE_FAIL_FAST_ENABLED and AUTO_RUN must be set
#                   to YES for this type of package. All SERVICES must
#                   have SERVICE_FAIL_FAST_ENABLED set to YES.
#
# NOTE: Packages which have a PACKAGE_TYPE of SYSTEM_MULTI_NODE are
#       not failover packages and should only be used for applications
#       provided by Hewlett-Packard.
#
```

```
# Since SYSTEM_MULTI_NODE packages run on multiple nodes at
# one time, following parameters are ignored:
#
#     FAILOVER_POLICY
#     FAILBACK_POLICY
#
# Since an IP address can not be assigned to more than node at a
# time, relocatable IP addresses can not be assigned in the
# package control script for multiple node packages. If
# volume groups are assigned to multiple node packages they must
# activated in a shared mode and data integrity is left to the
# application. Shared access requires a shared volume manager.
#
#
# Examples : PACKAGE_TYPE    FAILOVER (default)
#           PACKAGE_TYPE    SYSTEM_MULTI_NODE
#
```

```
PACKAGE_TYPE                FAILOVER
```

```
# Enter the failover policy for this package. This policy will be used
# to select an adoptive node whenever the package needs to be started.
# The default policy unless otherwise specified is CONFIGURED_NODE.
# This policy will select nodes in priority order from the list of
# NODE_NAME entries specified below.
#
# The alternative policy is MIN_PACKAGE_NODE. This policy will select
# the node, from the list of NODE_NAME entries below, which is
# running the least number of packages at the time this package needs
# to start.
```

```
FAILOVER_POLICY             CONFIGURED_NODE
```

```
# Enter the failback policy for this package. This policy will be used
# to determine what action to take when a package is not running on
# its primary node and its primary node is capable of running the
# package. The default policy unless otherwise specified is MANUAL.
# The MANUAL policy means no attempt will be made to move the package
# back to its primary node when it is running on an adoptive node.
#
# The alternative policy is AUTOMATIC. This policy will attempt to
# move the package back to its primary node whenever the primary node
# is capable of running the package.
```

```
FAILBACK_POLICY            MANUAL
```

패키지 및 서비스 구성

패키지 구성 만들기

```
# Enter the names of the nodes configured for this package. Repeat
# this line as necessary for additional adoptive nodes.
#
# NOTE: The order is relevant.
#       Put the second Adoptive Node after the first one.
#
# Example : NODE_NAME original_node
#          NODE_NAME adoptive_node
#
# If all nodes in the cluster are to be specified and order is not
# important, "NODE_NAME *" may be specified.
#
# Example : NODE_NAME *

NODE_NAME

# Enter the value for AUTO_RUN. Possible values are YES and NO.
# The default for AUTO_RUN is YES. When the cluster is started the
# package will be automatically started. In the event of a failure the
# package will be started on an adoptive node. Adjust as necessary.
#
# AUTO_RUN replaces obsolete PKG_SWITCHING_ENABLED.

AUTO_RUN YES

# Enter the value for LOCAL_LAN_FAILOVER_ALLOWED.
# Possible values are YES and NO.
# The default for LOCAL_LAN_FAILOVER_ALLOWED is YES. In the event of a
# failure, this permits the cluster software to switch LANs locally
# (transfer to a standby LAN card). Adjust as necessary.
#
# LOCAL_LAN_FAILOVER_ALLOWED replaces obsolete NET_SWITCHING_ENABLED.

LOCAL_LAN_FAILOVER_ALLOWED YES

# Enter the value for NODE_FAIL_FAST_ENABLED.
# Possible values are YES and NO.
# The default for NODE_FAIL_FAST_ENABLED is NO. If set to YES,
# in the event of a failure, the cluster software will halt the node
# on which the package is running. All SYSTEM_MULTI_NODE packages must have
# NODE_FAIL_FAST_ENABLED set to YES. Adjust as necessary.

NODE_FAIL_FAST_ENABLED NO
```



```
# Enter the complete path for the run and halt scripts. In most cases
# the run script and halt script specified here will be the same script,
# the package control script generated by the cmmakepkg command. This
# control script handles the run(ning) and halt(ing) of the package.
# Enter the timeout, specified in seconds, for the run and halt scripts.
# If the script has not completed by the specified timeout value,
# it will be terminated. The default for each script timeout is
# NO_TIMEOUT. Adjust the timeouts as necessary to permit full
# execution of each script.
# Note: The HALT_SCRIPT_TIMEOUT should be greater than the sum of
# all SERVICE_HALT_TIMEOUT values specified for all services.
```

```
RUN_SCRIPT
RUN_SCRIPT_TIMEOUT          NO_TIMEOUT
HALT_SCRIPT
HALT_SCRIPT_TIMEOUT        NO_TIMEOUT
```

```
# Enter the names of the storage groups configured for this package.
# Repeat this line as necessary for additional storage groups.
#
```

```
# Storage groups are only used with CVM disk groups. Neither
# VxVM disk groups or LVM volume groups should be listed here.
# By specifying a CVM disk group with the STORAGE_GROUP keyword
# this package will not run until the VxVM-CVM-pkg package is
# running and thus the CVM shared disk groups are ready for
# activation.
```

```
# NOTE: Should only be used by applications provided by
# Hewlett-Packard.
```

```
# Example : STORAGE_GROUP dg01
#           STORAGE_GROUP dg02
#           STORAGE_GROUP dg03
#           STORAGE_GROUP dg04
```

```
# Enter the SERVICE_NAME, the SERVICE_FAIL_FAST_ENABLED and the
# SERVICE_HALT_TIMEOUT values for this package. Repeat these
# three lines as necessary for additional service names. All
# service names MUST correspond to the SERVICE_NAME[] entries in
# the package control script.
```

```
# The value for SERVICE_FAIL_FAST_ENABLED can be either YES or
# NO. If set to YES, in the event of a service failure, the
# cluster software will halt the node on which the service is
# running. If SERVICE_FAIL_FAST_ENABLED is not specified, the
```

패키지 및 서비스 구성

패키지 구성 만들기

```
# default will be NO.
#
# SERVICE_HALT_TIMEOUT is represented as a number of seconds.
# This timeout is used to determine the length of time (in
# seconds) the cluster software will wait for the service to
# halt before a SIGKILL signal is sent to force the termination
# of the service. In the event of a service halt, the cluster
# software will first send a SIGTERM signal to terminate the
# service. If the service does not halt, after waiting for the
# specified SERVICE_HALT_TIMEOUT, the cluster software will send
# out the SIGKILL signal to the service to force its termination.
# This timeout value should be large enough to allow all cleanup
# processes associated with the service to complete. If the
# SERVICE_HALT_TIMEOUT is not specified, a zero timeout will be
# assumed, meaning the cluster software will not wait at all
# before sending the SIGKILL signal to halt the service.
#
# Example: SERVICE_NAME                DB_SERVICE
#          SERVICE_FAIL_FAST_ENABLED   NO
#          SERVICE_HALT_TIMEOUT        300
#
# To configure a service, uncomment the following lines and
# fill in the values for all of the keywords.
#
#SERVICE_NAME                <service name>
#SERVICE_FAIL_FAST_ENABLED   <YES/NO>
#SERVICE_HALT_TIMEOUT        <number of seconds>

# Enter the network subnet name that is to be monitored for this package.
# Repeat this line as necessary for additional subnet names. If any of
# the subnets defined goes down, the package will be switched to another
# node that is configured for this package and has all the defined subnets
# available.
# The subnet names could be IPv4 or IPv6. The network subnet
# names that are to be monitored for this package could be a mix
# of IPv4 or IPv6 subnet names

#SUBNET

# The keywords RESOURCE_NAME, RESOURCE_POLLING_INTERVAL,
# RESOURCE_START, and RESOURCE_UP_VALUE are used to specify Package
# Resource Dependencies. To define a package Resource Dependency, a
# RESOURCE_NAME line with a fully qualified resource path name, and
# one or more RESOURCE_UP_VALUE lines are required. The
# RESOURCE_POLLING_INTERVAL and the RESOURCE_START are optional.
```

```

#
# The RESOURCE_POLLING_INTERVAL indicates how often, in seconds, the
# resource is to be monitored. It will be defaulted to 60 seconds if
# RESOURCE_POLLING_INTERVAL is not specified.
#
# The RESOURCE_START option can be set to either AUTOMATIC or DEFERRED.
# The default setting for RESOURCE_START is AUTOMATIC. If AUTOMATIC
# is specified, Serviceguard will start up resource monitoring for
# these AUTOMATIC resources automatically when the node starts up.
# If DEFERRED is selected, Serviceguard will not attempt to start
# resource monitoring for these resources during node start up. User
# should specify all the DEFERRED resources in the package run script
# so that these DEFERRED resources will be started up from the package
# run script during package run time.
#
# RESOURCE_UP_VALUE requires an operator and a value. This defines
# the resource 'UP' condition. The operators are =, !=, >, <, >=,
# and <=, depending on the type of value. Values can be string or
# numeric. If the type is string, then only = and != are valid
# operators. If the string contains whitespace, it must be enclosed
# in quotes. String values are case sensitive. For example,
#
#
#                               Resource is up when its value is
#                               -----
# RESOURCE_UP_VALUE      = UP                "UP"
# RESOURCE_UP_VALUE      != DOWN            Any value except "DOWN"
# RESOURCE_UP_VALUE      = "On Course"      "On Course"
#
# If the type is numeric, then it can specify a threshold, or a range to
# define a resource up condition. If it is a threshold, then any operator
# may be used. If a range is to be specified, then only > or >= may be used
# for the first operator, and only < or <= may be used for the second operator.
# For example,
#
#                               Resource is up when its value is
#                               -----
# RESOURCE_UP_VALUE      = 5                  5                (threshold)
# RESOURCE_UP_VALUE      > 5.1              greater than 5.1    (threshold)
# RESOURCE_UP_VALUE      > -5 and < 10      between -5 and 10    (range)
#
# Note that "and" is required between the lower limit and upper limit
# when specifying a range. The upper limit must be greater than the lower
# limit. If RESOURCE_UP_VALUE is repeated within a RESOURCE_NAME block, then
# they are inclusively OR'd together. Package Resource Dependencies may be
# defined by repeating the entire RESOURCE_NAME block.
#
# Example : RESOURCE_NAME                /net/interfaces/lan/status/lan0

```

패키지 및 서비스 구성

패키지 구성 만들기

```
#          RESOURCE_POLLING_INTERVAL  120
#          RESOURCE_START              AUTOMATIC
#          RESOURCE_UP_VALUE          = RUNNING
#          RESOURCE_UP_VALUE          = ONLINE
#
#          Means that the value of resource /net/interfaces/lan/status/lan0
#          will be checked every 120 seconds, and is considered to
#          be 'up' when its value is "RUNNING" or "ONLINE".
## Uncomment the following lines to specify Package Resource Dependencies.
#
#RESOURCE_NAME          <Full_path_name>
#RESOURCE_POLLING_INTERVAL <numeric_seconds>
#RESOURCE_START        <AUTOMATIC/DEFERRED>
#RESOURCE_UP_VALUE     <op> <string_or_numeric> [and <op> <numeric>]

# Access Control Policy Parameters.
#
# Three entries set the access control policy for the package:
# First line must be USER_NAME, second USER_HOST, and third USER_ROLE.
# Enter a value after each.
#
# 1. USER_NAME can either be ANY_USER, or a maximum of
#    8 login names from the /etc/passwd file on user host.
# 2. USER_HOST is where the user can issue Serviceguard commands.
#    If using Serviceguard Manager, it is the COM server.
#    Choose one of these three values: ANY_SERVICEGUARD_NODE, or
#    (any) CLUSTER_MEMBER_NODE, or a specific node. For node,
#    use the official hostname from domain name server, and not
#    an IP addresses or fully qualified name.
# 3. USER_ROLE must be PACKAGE_ADMIN. This role grants permission
#    to MONITOR, plus for administrative commands for the package.
#
# These policies do not effect root users. Access Policies here
# should not conflict with policies defined in the cluster configuration file.
#
# Example: to configure a role for user john from node noir to
# administer the package, enter:
# USER_NAME  john
# USER_HOST  noir
# USER_ROLE  PACKAGE_ADMIN
```

- FAILOVER_POLICY — CONFIGURED_NODE 또는 MIN_PACKAGE_NODE를 입력합니다.

- FAILBACK_POLICY — MANUAL 또는 AUTOMATIC을 입력합니다.
- NODE_NAME — 클러스터의 각 노드 이름을 별도의 줄에 입력합니다.
- AUTO_RUN — 패키지가 사용 가능한 첫 번째 노드에서 시작되도록 하려면 YES를 입력하고, 패키지가 자동으로 시작되지 않도록 하려면 NO를 입력합니다.
- LOCAL_LAN_FAILOVER_ALLOWED — 패키지 IP 주소가 대기 LAN으로 전환되는 것을 허용하려면 YES를 입력하고, 패키지의 로컬 전환을 허용하지 않으려면 NO를 입력합니다.
- RUN_SCRIPT 및 HALT_SCRIPT — 패키지 제어 스크립트의 경로 이름을 지정합니다(다음 절 참조). 기본 경로 이름은 없습니다.
- STORAGE_GROUP — 이 패키지에서 사용할 CVM 저장 장치 그룹의 이름을 지정합니다. 각 저장 장치 그룹(CVM 디스크 그룹)을 별도의 줄에 입력합니다. CVM 저장 장치 그룹은 클러스터 ASCII 구성 파일에 입력되지 **않습니다**.

주

이 파일에는 LVM 볼륨 그룹 또는 VxVM 디스크 그룹을 입력할 수 없습니다.

- 패키지에 서비스가 포함되어 있으면 SERVICE_NAME, SERVICE_FAIL_FAST_ENABLED 및 SERVICE_HALT_TIMEOUT 값을 입력합니다. 각 서비스에 대해 이 세 개의 값을 모두 입력하십시오. 패키지당 최대 30개의 서비스를 구성할 수 있습니다.
 - 패키지에 연관된 IP 주소가 있으면 SUBNET을 입력합니다. 이 값은 클러스터 구성에 이미 지정된 서브넷이어야 하며 IPv4 서브넷이나 IPv6 서브넷일 수 있습니다. 링크 로컬 서브넷은 패키지 ASCII 파일에 입력될 수 없으므로 링크 로컬 패키지 IP를 사용할 수 없습니다.
 - NODE_FAIL_FAST_ENABLED 매개 변수 — YES 또는 NO를 입력합니다.
- 등록된 리소스의 패키지 내에서 모니터링을 구성하려면 다음 매개 변수 값을 입력합니다.
- RESOURCE_NAME — Serviceguard에서 모니터링할 등록된 리소스의 이름을 입력합니다.
 - RESOURCE_POLLING_INTERVAL — 리소스가 사용 가능한 상태인지 확인하는 시간 주기를 입력합니다.

- RESOURCE_UP_VALUE — 리소스가 작동 중인 것으로 간주될 때를 지정하는 값을 입력합니다. 모니터링 중에 리소스에 대해 다른 값이 발견되면 패키지에 오류가 발생합니다.
- RESOURCE_START — RESOURCE_START 옵션은 **Serviceguard**에서 EMS 리소스에 대한 리소스 모니터링을 시작할 시점을 결정하는 데 사용됩니다.

RESOURCE_START 옵션은 AUTOMATIC 또는 DEFERRED로 설정할 수 있습니다.

AUTOMATIC을 지정할 경우 **Serviceguard** 클러스터 데몬이 노드에서 시작되면

Serviceguard에서는 자동으로 이 리소스에 대한 리소스 모니터링을 시작합니다.

리소스가 AUTOMATIC으로 구성되어 있으면 패키지 제어 스크립트에서

DEFERRED_RESOURCE_NAME을 정의할 필요가 없습니다.

DEFERRED를 선택하면 **Serviceguard**에서는 노드가 시작될 때 이 DEFERRED 리소

스에 대해 리소스 모니터링을 시작하려고 시도하지 않습니다. 그러나 이

DEFERRED 리소스는 패키지 제어 스크립트에 지정되어야 합니다.

다음 예제에서는 DEFERRED 및 AUTOMATIC 리소스를 구성하는 방법을 보여 줍니다. 패키지 구성 파일에서 다음과 같이 리소스를 지정하십시오.

```
RESOURCE_NAME           /net/interfaces/lan/status/lan0
RESOURCE_POLLING_INTERVAL 60
RESOURCE_START          DEFERRED
RESOURCE_UP_VALUE       = UP
```

```
RESOURCE_NAME           /net/interfaces/lan/status/lan1
RESOURCE_POLLING_INTERVAL 60
RESOURCE_START          DEFERRED
RESOURCE_UP_VALUE       = UP
```

```
RESOURCE_NAME           /net/interfaces/lan/status/lan2
RESOURCE_POLLING_INTERVAL 60
RESOURCE_START          AUTOMATIC
RESOURCE_UP_VALUE       = UP
```

액세스 제어 정책은 **Serviceguard** 버전 11.16의 새로운 기능입니다. 이 기능을 사용하면 루트 권한이 없는 사용자가 패키지 및 클러스터를 관리하도록 할 수 있습니다. 클러스터 범위 역할은 클러스터 구성 파일에서 정의됩니다. 패키지 관련 역할은 패키지 구성 파일에서 정의됩니다.

역할이 중복되거나 충돌하지 않아야 합니다. 그렇지 않으면 구성에 실패하게 되고 메시지가 표시됩니다. 따라서 패키지 파일에 역할을 만들기 전에 클러스터 구성 파일 (cmgetconf)을 확인하는 것이 좋습니다.

클러스터에서 역할을 구성하는 경우 패키지에 있는 동일한 사용자 이름/호스트 노드에 대한 역할을 구성하지 마십시오. 와일드카드도 예외입니다. 예를 들어, 클러스터 구성에서 *ftsys9*의 **ANY_USER**에 대한 **MONITOR** 역할이 있었는데 패키지 구성 파일에서 *ftsys9*의 사용자 *Lee*에 대한 **ADMIN** 역할을 만들었다면 사용자 *ftsys9*는 이 정책에 대해 패키지 관리 역할을 갖게 됩니다. (이 역할은 이미 전체 클러스터에 대한 모니터를 포함합니다.)

패키지 구성 파일 및 클러스터 구성 파일의 액세스 제어 정책에 대한 데이터는 나중에 이진 클러스터 구성 파일에 결합됩니다.

실행 중인 클러스터에 패키지 추가 또는 제거

클러스터가 실행되고 있을 때 패키지를 추가하거나 제거할 수 있습니다. 이때 **MAX_CONFIGURED_PACKAGES**의 제한을 받습니다. 온라인 상태에서 패키지를 추가하거나 제거하려면 “클러스터 및 패키지 유지 관리” 장을 참조하십시오.

패키지 제어 스크립트 작성

패키지 제어 스크립트에는 패키지의 모든 서비스를 실행하고, 실행 중에 모니터링하고, 장애에 대처하고, 필요한 경우 패키지를 중지하는 데 필요한 모든 정보가 포함되어 있습니다. **Serviceguard Manager**의 유도 모드에서 패키지 구성을 만들 때와 같이 제어 스크립트를 만들 수 있습니다. **HP-UX** 명령을 사용하여 패키지 제어 스크립트를 만들거나 수정할 수도 있습니다.

Serviceguard Manager를 사용하여 패키지 제어 스크립트 작성

Serviceguard Manager의 유도 모드를 사용하여 구성 화면에서 표준 제어 스크립트를 자동으로 만들 수 있습니다. 적용을 클릭하면 모든 노드에 자동으로 스크립트가 배포됩니다. 스크립트를 만들고 수정하는 도중 스크립트를 보려면 미리 보기 단추를 클릭합니다.

원하는 경우 제어 스크립트를 직접 편집할 수 있습니다. 그러나 스크립트를 직접 편집한 후에는 **Serviceguard Manager**에서 더 이상 스크립트의 편집 내용을 읽거나 스크립트를 수정할 수 없게 됩니다.

Serviceguard Manager에서는 **HA-NFS** 도구 키트, **ECM** 도구 키트 또는 **Metro** 클러스터 스크립트 등의 다른 파일은 배포하지 않습니다. 따라서 이러한 파일은 사용자가 직접 배포해야 합니다.

명령을 사용하여 패키지 제어 스크립트 작성

패키지마다 별도의 제어 스크립트가 있어야 하며 이 스크립트는 실행 가능해야 합니다.

제어 스크립트는 보안상의 이유로 경로에 *cmcluster*라는 문자열이 포함된 디렉토리에 있어야 합니다. 제어 스크립트는 패키지 디렉토리에 위치하며 패키지 **ASCII** 구성 파일의 **RUN_SCRIPT** 및 **HALT_SCRIPT** 매개 변수에서 지정한 이름과 동일한 이름이 제어 스크립트에 지정됩니다. 패키지 제어 스크립트 템플릿에는 패키지 실행 명령과 중지 명령이 모두 들어 있습니다. 실행 및 중지 작업 모두에 하나의 스크립트를 사용할 수 있으며, 원하는 경우 별도의 스크립트를 만들 수도 있습니다.

다음 단계에 따라 예제 패키지 *pkg1*의 제어 스크립트를 만드십시오.

먼저 제어 스크립트 템플릿을 생성합니다.


```
# cmmakepkg -s /etc/cmcluster/pkg1/pkg1.sh
```

“패키지 제어 스크립트 사용자 정의” 절에서 설명한 것처럼 스크립트를 사용자 정의할 수도 있습니다.

데이터베이스 제품의 패키지 만들기

데이터베이스 소프트웨어의 시작 및 종료를 클러스터 노드의 시작 및 종료에 맞게 조정하려면 별도로 구입 가능한 **Enterprise Cluster Master Toolkit** 제품(B5139DA)과 함께 제공되는 데이터베이스 템플릿 파일을 사용하면 됩니다. 이 파일은 `/opt/cmcluster/toolkit/DB/`에 있습니다. 별도의 도구 키트는 **Oracle**, **Informix** 및 **Sybase**에서 구입할 수 있습니다. 표준 패키지 제어 스크립트뿐만 아니라 데이터베이스에 맞는 특수 스크립트를 사용할 수 있습니다. 이 스크립트를 설정하려면 각 도구 키트와 함께 제공되는 **README** 파일의 설명을 따르십시오.

패키지 제어 스크립트 사용자 정의

패키지 구성 워크시트의 정보를 사용하여 제어 스크립트의 처음 부분에 있는 정의 및 선언을 확인합니다. 다음과 같이 사용자 정의해야 합니다.

- 서비스를 시작하는 데 필요한 경로와 일치하도록 **PATH** 문을 업데이트합니다.
- **LVM**을 사용하고 있는 경우 **VG[]** 배열 매개 변수를 사용하여 활성화될 볼륨 그룹의 이름을 입력하고 저장 장치 활성화 명령에 적절한 옵션을 선택합니다. 원하는 경우 파일 시스템 마운트 및 마운트 해제 옵션도 포함할 수 있습니다. **LVM** 볼륨 그룹에 대해서는 **VXVM_DG[]** 또는 **CVM_DG[]** 매개 변수를 사용하지 **마십시오**.
- **CVM**을 사용하고 있는 경우 **CVM_DG[]** 배열 매개 변수를 사용하여 활성화될 디스크 그룹의 이름을 입력하고 해당 저장 장치 활성화 명령인 **CVM_ACTIVATION_CMD**를 선택합니다. **CVM** 디스크 그룹에 대해서는 **VG[]** 또는 **VXVM_DG[]** 매개 변수를 사용하지 **마십시오**.
- **CVM** 없이 **VxVM** 디스크 그룹을 사용하고 있는 경우에는 **VXVM_DG[]** 배열 매개 변수를 사용하여 가져올 **VxVM** 디스크 그룹의 이름을 입력합니다. 배열 요소당 하나의 디스크 그룹을 입력하십시오. **CVM**을 사용하지 않는 **VxVM** 디스크 그룹에 대해서는 **CVM_DG[]** 또는 **VG[]** 매개 변수를 사용하지 **마십시오**. 또한 활성화 명령도 지정하지 **마십시오**.
- 미러링되는 **VxVM** 디스크를 사용하고 있는 경우 미러 복구 옵션인 **VXVOL**을 지정합니다.

- 논리 볼륨과 논리 볼륨에 마운트할 파일 시스템의 이름을 추가합니다.
- 저장 장치 활성화 명령에 적절한 옵션을 선택하고(기본 **VxVM** 디스크 그룹에는 적용되지 않음) 원하는 경우 파일 시스템 마운트 옵션도 포함합니다.
- 파일 시스템 마운트 재시도 및 마운트 해제 횟수 옵션을 지정합니다.
- 패키지에서 많은 수의 볼륨 그룹 또는 디스크 그룹을 사용하거나 많은 수의 파일 시스템을 마운트하는 경우 **동시에 작업할 수 있는** `vgchange`, `mount/umount` 및 `fsck`의 수를 늘리는 것이 좋습니다. 대부분의 패키지에는 기본값 **1**이 적합합니다.
- 패키지의 **IP** 서브넷 및 **IP** 주소 쌍을 정의합니다. **IPv4** 또는 **IPv6** 주소를 사용할 수 있습니다.
- 서비스 이름을 추가합니다.
- 서비스 명령을 추가합니다.
- 원하는 경우 서비스 재시작 매개 변수를 추가합니다.

주

서비스 실행 명령을 정의할 때는 주의하십시오. 각 실행 명령은 다음과 같은 방법으로 제어 스크립트에 의해 실행됩니다.

- `cmrunserv` 명령이 각 실행 명령을 실행한 후 **Serviceguard**에서 이 실행 명령으로 만들어진 프로세스의 프로세스 **ID**를 모니터링합니다.
- `cmrunserv`에 의해 시작된 명령이 종료될 때 **Serviceguard**에서는 장애가 발생했음을 인식하고 적절한 조치를 취하는데, 이 장애 조치에 의해 패키지가 대체 노드로 전달될 수도 있습니다.
- 실행 명령이 다른 명령을 실행하는 셸 스크립트인데 종료된 경우 **Serviceguard**에서는 이러한 정상 종료를 **오류**로 간주합니다.

제어 스크립트를 실행할 때 문제가 발생하지 않도록 하려면 각 실행 명령이 실제 서비스의 이름을 나타내고 해당 프로세스는 실제 서비스가 중지되기 전까지 활성 상태를 유지하도록 해야 합니다.

기본값 이외에도 실행 및 중지 작업을 정의해야 하는 경우에는 **CUSTOMER DEFINED FUNCTIONS**라는 제목의 섹션에서 이에 대한 함수를 만드십시오.

제어 스크립트에서 VxVM 디스크 그룹을 관리하는 방법

CVM 외부의 VxVM 디스크 그룹은 Serviceguard 클러스터에 의해 제어되지 않습니다. 패키지 제어 스크립트에서는 표준 VxVM 명령을 사용하여 이러한 디스크 그룹을 가져오고 내보냅니다. 디스크 그룹 가져오기 및 내보내기에 대한 자세한 내용은 vxvg 맨페이지의 *import* 및 *deport* 옵션 부분을 참조하십시오.

제어 스크립트에서는 vxvg 명령과 -tfc 옵션을 사용하여 디스크 그룹을 가져옵니다. -t 옵션은 *noautoimport* 플래그를 사용하여 디스크를 가져오도록 지정합니다. 즉, 부팅 시 자동으로 디스크를 다시 가져오지 않습니다. 패키지 제어 스크립트에 포함된 디스크 그룹은 Serviceguard 패키지에서만 가져오므로 자동으로 가져올 수 없습니다.

-f 옵션을 사용하면 현재 하나 이상의 디스크(예: 미러)를 사용할 수 없는 경우에도 디스크 그룹을 가져올 수 있습니다. -c 옵션은 이전에 클러스터의 다른 노드에서 활성화되어 디스크에 기록된 기존 호스트 ID를 모두 삭제합니다. TOC와 함께 종료된 다른 노드에서 디스크가 사용된 경우 해당 호스트 ID는 여전히 디스크에 기록된 상태이므로 새 노드의 ID를 디스크에 쓰려면 이전 호스트 ID를 삭제해야 합니다. 호스트 ID가 설정되어 있으며 이 ID가 장애 상태가 아닌 노드와 일치하는 경우에는 디스크 그룹을 가져와 호스트 ID를 삭제하지 않습니다. 이렇게 함으로써 여러 노드에 디스크 그룹을 가져와 데이터가 손상되는 결과를 방지합니다.

경고

패키지 제어 스크립트 프레임워크 내에서 -C 옵션을 사용해도 이 옵션은 명령줄에서 정상적으로 사용되지 않습니다. “문제 해결” 절에서는 명령줄에서 -C를 사용해야 하는 몇 가지 상황을 보여 줍니다.

다음 예제에서는 명령과 제어 스크립트에서 사용되는 동일한 옵션을 함께 보여 줍니다.

```
# vxvg -tfc import dg_01
```

이 명령은 현재 디스크에 기록된 호스트 ID가 서로 다른 경우에도 디스크 그룹 dg_01에 있는 모든 디스크의 소유권을 받습니다. 이 명령은 현재 노드의 호스트 ID를 디스크 그룹 dg_01에 있는 모든 디스크에 기록하고 디스크에 대해 *noautoimport* 플래그를 설정합니다. 이 플래그를 설정하면 다시 부팅한 후 노드에서 디스크 그룹을 자동으로 다시 가져오지 않습니다. 클러스터의 노드에 장애가 발생해도 호스트 ID는 디스크 그룹의 각 디스크에 여전히 기록되어 있습니다. 그러나 노드가 Serviceguard 클러스터의 일부인 경우 다시 부팅하면 Serviceguard에서 디스크 그룹을 제어하지 않는 경우에도

호스트 ID를 소유하고 있는 노드에 의해 `noautoimport` 플래그가 설정된 모든 디스크에서 해당 호스트 ID가 삭제됩니다. 따라서 디스크 그룹에 액세스할 수 있는 모든 클러스터 노드에서 클러스터 작업의 일부로 디스크를 가져올 수 있게 됩니다.

또한 제어 스크립트에서는 `vxvol startall` 명령을 사용하여 가져온 각 디스크 그룹에서 논리 볼륨을 시작합니다.

다수의 저장 장치를 위한 최적화

다수의 파일 시스템 또는 저장 장치 그룹을 사용하는 경우 네 개의 변수 집합을 사용하여 성능을 향상시킬 수 있습니다. 자세한 내용은 제어 스크립트 템플릿의 주석을 참조하십시오. 네 개의 변수는 다음과 같이 요약됩니다.

- `CONCURRENT_VGCHANGE_OPERATIONS` — 패키지가 시작될 때 동시에 수행되는 LVM 볼륨 그룹 활성화 작업과 패키지가 종료될 때 동시에 수행되는 LVM 볼륨 그룹 비활성화 작업의 수를 정의합니다.
- `CONCURRENT_DISKGROUP_OPERATIONS` — 패키지가 시작될 때 동시에 수행되는 VxVM 볼륨 그룹 활성화 작업과 패키지가 종료될 때 동시에 수행되는 VxVM 볼륨 그룹 비활성화 작업의 수를 정의합니다.
- `CONCURRENT_FSCK_OPERATIONS` — 패키지가 시작될 때 동시에 수행되는 `fsck` 작업의 수를 정의합니다.
- `CONCURRENT_MOUNT_AND_UMOUNT_OPERATIONS` — 패키지가 시작될 때 동시에 수행되는 마운트 작업과 패키지가 종료될 때 동시에 수행되는 마운트 해제 작업의 수를 정의합니다.

다수의 파일 시스템을 사용하는 환경에서는 `FSCK_OPT` 및 `FS_UMOUNT_OPT` 매개 변수에 `-s` 옵션을 사용할 수 있습니다. `-s` 옵션은 마운트/마운트 해제 및 `fsck` 작업이 동시에 수행되도록 합니다. 표준 `11i v1(11.11) HP-UX`에서 이 옵션을 사용하려면 패치를 설치해야 합니다.

패키지 제어 스크립트 템플릿 파일

다음은 패키지 제어 스크립트 템플릿 파일의 일부입니다. 이 파일에는 VxVM 및 LVM 저장 장치 그룹의 활성화 섹션이 포함되어 있습니다.

```
# *****
# *
# *          HIGH AVAILABILITY PACKAGE CONTROL SCRIPT (template)          *
# *
# *          Note: This file MUST be edited before it can be used.        *
# *
# *****

# The PACKAGE and NODE environment variables are set by
# Serviceguard at the time the control script is executed.
# Do not set these environment variables yourself!
# The package may fail to start or halt if the values for
# these environment variables are altered.

. ${SGCONFIGFILE:=/etc/cmcluster.conf}

# UNCOMMENT the variables as you set them.

# Set PATH to reference the appropriate directories.
PATH=$SGSBIN:/usr/bin:/usr/sbin:/etc:/bin

# VOLUME GROUP ACTIVATION:
# Specify the method of activation for volume groups.
# Leave the default ("VGCHANGE="vgchange -a e") if you want volume
# groups activated in exclusive mode. This assumes the volume groups have
# been initialized with 'vgchange -c y' at the time of creation.
#
# Uncomment the first line (VGCHANGE="vgchange -a e -q n"), and comment
# out the default, if your disks are mirrored on separate physical paths,
#
# Uncomment the second line (VGCHANGE="vgchange -a e -q n -s"), and comment
# and you want the mirror resynchronization to occur in parallel with
# the package startup.
#
# Uncomment the third line (VGCHANGE="vgchange -a y") if you wish to
# use non-exclusive activation mode. Single node cluster configurations
# must use non-exclusive activation.
#
# VGCHANGE="vgchange -a e -q n"
# VGCHANGE="vgchange -a e -q n -s"
# VGCHANGE="vgchange -a y"
VGCHANGE="vgchange -a e"                # Default

# CVM DISK GROUP ACTIVATION:
# Specify the method of activation for CVM disk groups.
```

패키지 및 서비스 구성

패키지 제어 스크립트 작성

```
# Leave the default
# (CVM_ACTIVATION_CMD="vxdg -g \${DiskGroup} set activation=exclusivewrite")
# if you want disk groups activated in the exclusive write mode.
#
# Uncomment the first line
# (CVM_ACTIVATION_CMD="vxdg -g \${DiskGroup} set activation=readonly"),
# and comment out the default, if you want disk groups activated in
# the readonly mode.
#
# Uncomment the second line
# (CVM_ACTIVATION_CMD="vxdg -g \${DiskGroup} set activation=sharedread"),
# and comment out the default, if you want disk groups activated in the
# shared read mode.
#
# Uncomment the third line
# (CVM_ACTIVATION_CMD="vxdg -g \${DiskGroup} set activation=sharedwrite"),
# and comment out the default, if you want disk groups activated in the
# shared write mode.
#
# CVM_ACTIVATION_CMD="vxdg -g \${DiskGroup} set activation=readonly"
# CVM_ACTIVATION_CMD="vxdg -g \${DiskGroup} set activation=sharedread"
# CVM_ACTIVATION_CMD="vxdg -g \${DiskGroup} set activation=sharedwrite"
CVM_ACTIVATION_CMD="vxdg -g \${DiskGroup} set activation=exclusivewrite"

# VOLUME GROUPS
# Specify which volume groups are used by this package. Uncomment VG[0]="
# and fill in the name of your first volume group. You must begin with
# VG[0], and increment the list in sequence.
#
# For example, if this package uses your volume groups vg01 and vg02, enter:
#     VG[0]=vg01
#     VG[1]=vg02
#
# The volume group activation method is defined above. The filesystems
# associated with these volume groups are specified below.
#
#VG[0]=" "

# CVM DISK GROUPS
# Specify which cvm disk groups are used by this package. Uncomment
# CVM_DG[0]=" " and fill in the name of your first disk group. You must
# begin with CVM_DG[0], and increment the list in sequence.
#
# For example, if this package uses your disk groups dg01 and dg02, enter:
#     CVM_DG[0]=dg01
#     CVM_DG[1]=dg02
```

```
#
# The cvm disk group activation method is defined above. The filesystems
# associated with these volume groups are specified below in the CVM_*
# variables.
#
#CVM_DG[0]=" "

# VxVM DISK GROUPS
# Specify which VxVM disk groups are used by this package. Uncomment
# VXVM_DG[0]=" " and fill in the name of your first disk group. You must
# begin with VXVM_DG[0], and increment the list in sequence.
#
# For example, if this package uses your disk groups dg01 and dg02, enter:
#     VXVM_DG[0]=dg01
#     VXVM_DG[1]=dg02
#
# The cvm disk group activation method is defined above.
#
#VXVM_DG[0]=" "

#
# NOTE: A package could have LVM volume groups, CVM disk groups and VxVM
#       disk groups.
#
# NOTE: When VxVM is initialized it will store the hostname of the
#       local node in its volboot file in a variable called 'hostid'.
#       The MC Serviceguard package control scripts use both the values of
#       the hostname(1m) command and the VxVM hostid. As a result
#       the VxVM hostid should always match the value of the
#       hostname(1m) command.
#
#       If you modify the local host name after VxVM has been
#       initialized and such that hostname(1m) does not equal uname -n,
#       you need to use the vxdctl(1m) command to set the VxVM hostid
#       field to the value of hostname(1m). Failure to do so will
#       result in the package failing to start.

# FILESYSTEMS
# Filesystems are defined as entries specifying the logical volume, the
# mount point, the mount, umount and fsck options and type of the file system.
# Each filesystem will be fsck'd prior to being mounted. The filesystems
# will be mounted in the order specified during package startup and will
# be unmounted in reverse order during package shutdown. Ensure that
# volume groups referenced by the logical volume definitions below are
# included in volume group definitions above.
#
```

패키지 및 서비스 구성

패키지 제어 스크립트 작성

```
# Specify the filesystems which are used by this package. Uncomment
# LV[0]=""; FS[0]=""; FS_MOUNT_OPT[0]=""; FS_UMOUNT_OPT[0]="";
FS_FSCK_OPT[0]=" "
# FS_TYPE[0]=" " and fill in the name of your first logical volume,
# filesystem, mount, umount and fsck options and filesystem type
# for the file system. You must begin with LV[0], FS[0],
# FS_MOUNT_OPT[0], FS_UMOUNT_OPT[0], FS_FSCK_OPT[0], FS_TYPE[0]
# and increment the list in sequence.
#
# Note: The FS_TYPE parameter lets you specify the type of filesystem to be
# mounted. Specifying a particular FS_TYPE will improve package failover time.

# The FSCK_OPT and FS_UMOUNT_OPT parameters can be used to include the
# -s option with the fsck and umount commands to improve performance for
# environments that use a large number of filesystems. (An example of a
# large environment is given below following the description of the
# CONCURRENT_MOUNT_AND_UMOUNT_OPERATIONS parameter.)
#
# Example: If a package uses two JFS filesystems, pkg01a and pkg01b,
# which are mounted on LVM logical volumes lv01 and lv02 for read and
# write operation, you would enter the following:
#     LV[0]=/dev/vg01/lv01; FS[0]=/pkg01a; FS_MOUNT_OPT[0]="-o rw";
#     FS_UMOUNT_OPT[0]=""; FS_FSCK_OPT[0]=""; FS_TYPE[0]="vxfs"
#
#     LV[1]=/dev/vg01/lv02; FS[1]=/pkg01b; FS_MOUNT_OPT[1]="-o rw"
#     FS_UMOUNT_OPT[1]=""; FS_FSCK_OPT[1]=""; FS_TYPE[1]="vxfs"
#
# LV[0]=""; FS[0]=""; FS_MOUNT_OPT[0]=""; FS_UMOUNT_OPT[0]="";
FS_FSCK_OPT[0]=" "
#FS_TYPE[0]=" "
#
# VOLUME RECOVERY
#
# When mirrored VxVM volumes are started during the package control
# bring up, if recovery is required the default behavior is for
# the package control script to wait until recovery has been
# completed.
#
# To allow mirror resynchronization to occur in parallel with
# the package startup, uncomment the line
# VXVOL="vxvol -g \${DiskGroup} -o bg startall" and comment out the default.
#
# VXVOL="vxvol -g \${DiskGroup} -o bg startall"
VXVOL="vxvol -g \${DiskGroup} startall"      # Default

# FILESYSTEM UNMOUNT COUNT
```



```
# Specify the number of unmount attempts for each filesystem during package
# shutdown. The default is set to 1.
FS_UMOUNT_COUNT=1

# FILESYSTEM MOUNT RETRY COUNT.
# Specify the number of mount retrys for each filesystem.
# The default is 0. During startup, if a mount point is busy
# and FS_MOUNT_RETRY_COUNT is 0, package startup will fail and
# the script will exit with 1. If a mount point is busy and
# FS_MOUNT_RETRY_COUNT is greater than 0, the script will attempt
# to kill the user responsible for the busy mount point
# and then mount the file system. It will attempt to kill user and
# retry mount, for the number of times specified in FS_MOUNT_RETRY_COUNT.
# If the mount still fails after this number of attempts, the script
# will exit with 1.

# NOTE: If the FS_MOUNT_RETRY_COUNT > 0, the script will execute
# "fuser -ku" to freeup busy mount point.
FS_MOUNT_RETRY_COUNT=0
#

# Configuring the concurrent operations below can be used to improve the
# performance for starting up or halting a package. The maximum value for
# each concurrent operation parameter is 1024. Set these values carefully.
# The performance could actually decrease if the values are set too high
# for the system resources available on your cluster nodes. Some examples
# of system resources that can affect the optimum number of concurrent
# operations are: number of CPUs, amount of available memory, the kernel
# configuration for nfile and nproc. In some cases, if you set the number
# of concurrent operations too high, the package may not be able to start
# or to halt. For example, if you set CONCURRENT_VGCHANGE_OPERATIONS=5
# and the node where the package is started has only one processor, then
# running concurrent volume group activations will not be beneficial.
# It is suggested that the number of concurrent operations be tuned
# carefully, increasing the values a little at a time and observing the
# effect on the performance, and the values should never be set to a value
# where the performance levels off or declines. Additionally, the values
# used should take into account the node with the least resources in the
# cluster, and how many other packages may be running on the node.
# For instance, if you tune the concurrent operations for a package so
# that it provides optimum performance for the package on a node while
# no other packages are running on that node, the package performance
# may be significantly reduced, or may even fail when other packages are
# already running on that node.
#
# CONCURRENT VGCHANGE OPERATIONS
# Specify the number of concurrent volume group activations or
# deactivations to allow during package startup or shutdown.
# Setting this value to an appropriate number may improve the performance
```

패키지 및 서비스 구성

패키지 제어 스크립트 작성

```
# while activating or deactivating a large number of volume groups in the
# package. If the specified value is less than 1, the script defaults it
# to 1 and proceeds with a warning message in the package control script
# logfile.
CONCURRENT_VGCHANGE_OPERATIONS=1

# CONCURRENT DISK GROUP OPERATIONS
# Specify the number of concurrent VxVM DG imports or exports to allow
# during package startup or shutdown.
# Setting this value to an appropriate number may improve the performance
# while importing or exporting a large number of disk groups in the
# package. If the specified value is less than 1, the script defaults it
# to 1 and proceeds with a warning message in the package control script
# logfile.
CONCURRENT_DISKGROUP_OPERATIONS=1

# CONCURRENT FSCK OPERATIONS
# Specify the number of concurrent fsck to allow during package startup.
# Setting this value to an appropriate number may improve the performance
# while checking a large number of file systems in the package. If the
# specified value is less than 1, the script defaults it to 1 and proceeds
# with a warning message in the package control script logfile.
CONCURRENT_FSCK_OPERATIONS=1

# CONCURRENT MOUNT AND UMOUNT OPERATIONS
# Specify the number of concurrent mounts and umounts to allow during
# package startup or shutdown.
# Setting this value to an appropriate number may improve the performance
# while mounting or un-mounting a large number of file systems in the package.

# If the specified value is less than 1, the script defaults it to 1 and
# proceeds with a warning message in the package control script logfile.
CONCURRENT_MOUNT_AND_UMOUNT_OPERATIONS=1

# Example: If a package uses 50 JFS filesystems, pkg01aa through pkg01bx,
# which are mounted on the 50 logical volumes lv011..lv0150 for read and write
# operation, you may enter the following:
#
#     CONCURRENT_DISKGROUP_OPERATIONS=50
#     CONCURRENT_FSCK_OPERATIONS=50
#     CONCURRENT_MOUNT_AND_UMOUNT_OPERATIONS=50
#
#     LV[0]=/dev/vg01/lv011; FS[0]=/pkg01aa; FS_MOUNT_OPT[0]="-o rw";
#     FS_UMOUNT_OPT[0]="-s"; FS_FSCK_OPT[0]="-s"; FS_TYPE[0]="vxfs"
```

```
#
# LV[1]=/dev/vg01/lvol2; FS[1]=/pkg01ab; FS_MOUNT_OPT[1]="-o rw"
# FS_UMOUNT_OPT[1]="-s"; FS_FSCK_OPT[1]="-s"; FS_TYPE[0]="vxfs"
#   :           :           :
#   :           :           :
#   :           :           :
# LV[49]=/dev/vg01/lvol50; FS[49]=/pkg01bx; FS_MOUNT_OPT[49]="-o rw"
# FS_UMOUNT_OPT[49]="-s"; FS_FSCK_OPT[49]="-s"; FS_TYPE[0]="vxfs"
#
# IP ADDRESSES
# Specify the IP and Subnet address pairs which are used by this package.
# You could specify IPv4 or IPv6 IP and subnet address pairs.
# Uncomment IP[0]=" " and SUBNET[0]=" " and fill in the name of your first
# IP and subnet address. You must begin with IP[0] and SUBNET[0] and
# increment the list in sequence.
#
# For example, if this package uses an IP of 192.10.25.12 and a subnet of
# 192.10.25.0 enter:
#     IP[0]=192.10.25.12
#     SUBNET[0]=192.10.25.0
#     (netmask=255.255.255.0)
#
# Hint: Run "netstat -i" to see the available subnets in the Network field.
#
# For example, if this package uses an IPv6 IP of 2001::1/64
# The address prefix identifies the subnet as 2001::/64 which is an
available
# subnet.
# enter:
#     IP[0]=2001::1
#     SUBNET[0]=2001::/64
#     (netmask=ffff:ffff:ffff:ffff::)
# Alternatively the IPv6 IP/Subnet pair can be specified without the prefix
# for the IPv6 subnet.
#     IP[0]=2001::1
#     SUBNET[0]=2001::
#     (netmask=ffff:ffff:ffff:ffff::)
#
# Hint: Run "netstat -i" to see the available IPv6 subnets by looking
# at the address prefixes
# IP/Subnet address pairs for each IP address you want to add to a subnet
# interface card. Must be set in pairs, even for IP addresses on the same
# subnet.
#
#IP[0]=" "
#SUBNET[0]=" "
```

패키지 및 서비스 구성

패키지 제어 스크립트 작성

```
# SERVICE NAMES AND COMMANDS.
# Specify the service name, command, and restart parameters which are
# used by this package. Uncomment SERVICE_NAME[0]="", SERVICE_CMD[0]="",
# SERVICE_RESTART[0]=" " and fill in the name of the first service, command,
# and restart parameters. You must begin with SERVICE_NAME[0],SERVICE_CMD[0],
# and SERVICE_RESTART[0] and increment the list in sequence.
#
# For example:
#     SERVICE_NAME[0]=pkg1a
#     SERVICE_CMD[0]="/usr/bin/X11/xclock -display 192.10.25.54:0"
#     SERVICE_RESTART[0]=" " # Will not restart the service.
#
#     SERVICE_NAME[1]=pkg1b
#     SERVICE_CMD[1]="/usr/bin/X11/xload -display 192.10.25.54:0"
#     SERVICE_RESTART[1]="-r 2" # Will restart the service twice.
#
#     SERVICE_NAME[2]=pkg1c
#     SERVICE_CMD[2]="/usr/sbin/ping"
#     SERVICE_RESTART[2]="-R" # Will restart the service an infinite
#                             number of times.
#
# Note: No environmental variables will be passed to the command, this
# includes the PATH variable. Absolute path names are required for the
# service command definition. Default shell is /usr/bin/sh.
#
#SERVICE_NAME[0]=" "
#SERVICE_CMD[0]=" "
#SERVICE_RESTART[0]=" "

# DEFERRED_RESOURCE NAME
# Specify the full path name of the 'DEFERRED' resources configured for
# this package. Uncomment DEFERRED_RESOURCE_NAME[0]=" " and fill in the
# full path name of the resource.
#
#DEFERRED_RESOURCE_NAME[0]=" "

# DTC manager information for each DTC.
# Example: DTC[0]=dtc_20
#DTC_NAME[0]=
#
#HA_NFS_SCRIPT_EXTENSION
# If the package uses HA NFS, this variable can be used to alter the
# name of the HA NFS script. If not set, the name of this script is
# assumed to be "ha_nfs.sh". If set, the "sh" portion of the default
# script name is replaced by the value of this variable. So if
```

```
# HA_NFS_SCRIPT_EXTENSION is set to "package1.sh", for example, the name
# of the HA NFS script becomes "ha_nfs.package1.sh". In any case,
# the HA NFS script must be placed in the same directory as the package
# control script. This allows multiple packages to be run out of the
# same directory, as needed by SGeSAP.
#HA_NFS_SCRIPT_EXTENSION=" "
```

제어 스크립트의 일부인 위 스크립트에서는 변수 집합에 값을 할당하는 것을 보여 줍니다. 나머지 스크립트에서는 이러한 변수로 **Logical Volume Manager** 명령, **HP-UX** 명령 및 **Serviceguard** 명령(`cmrunserv`, `cmmodnet`, `cmhaltserv` 등)을 실행하여 패키지를 제어합니다. 제어 스크립트 템플릿의 복사본을 조사하여 논리의 흐름을 살펴 보십시오. 다음 명령을 사용하십시오.

```
# cmmakepkg -s | more
```

기본 함수는 스크립트 맨 마지막에 나타납니다.

각 변수는 선택적이며 패키지 작업에 필요한 만큼만 적절하게 사용해야 합니다. 예를 들어, 패키지에서 볼륨 그룹을 활성화할 필요가 없는 경우에는 VG 변수를 생략하고, 패키지에서 서비스를 사용하지 않을 경우에는 해당 `SERVICE_NAME`, `SERVICE_CMD`, `SERVICE_RESTART` 변수를 생략하십시오.

패키지 구성 파일에서 **EMS** 리소스를 **DEFERRED** 리소스로 정의한 경우에는 패키지 제어 스크립트에서 `DEFERRED_RESOURCE_NAME`을 정의해야 합니다.

`DEFERRED_RESOURCE_NAME` 매개 변수를 사용하여 지연 시작 리소스만 지정합니다.

```
DEFERRED_RESOURCE_NAME[0]="/net/interfaces/lan/status/lan0"
DEFERRED_RESOURCE_NAME[1]="/net/interfaces/lan/status/lan1"
```

스크립트를 사용자 정의한 후에는 `rcp` 또는 `ftp`나 원하는 복사 방법을 사용하여 클러스터의 각 노드에 스크립트를 배포합니다.

패키지 제어 스크립트에 사용자 정의 함수 추가

패키지 제어 스크립트에 다른 셸 명령을 추가하여 패키지가 시작되거나 중단될 때마다 실행되도록 할 수 있습니다. 스크립트의 `CUSTOMER DEFINED FUNCTIONS` 영역에 이 명령을 입력하면 됩니다. 이렇게 하면 제어 스크립트를 보다 폭 넓게 사용자 정의할 수 있습니다.

스크립트의 이 부분에 대한 예는 아래에서 볼 수 있습니다. 여기에는 패키지의 시작 및 중지를 특수 파일에 기록하기 위한 `date` 및 `echo` 명령이 포함되어 있습니다.

```
# START OF CUSTOMER DEFINED FUNCTIONS

# This function is a place holder for customer defined functions.
# You should define all actions you want to happen here, before the service is
# started. You can create as many functions as you need.

function customer_defined_run_cmds
{
# ADD customer defined run commands.
: # do nothing instruction, because a function must contain some command.
date >> /tmp/pkg1.datelog
echo 'Starting pkg1' >> /tmp/pkg1.datelog
test_return 51
}

# This function is a place holder for customer defined functions.
# You should define all actions you want to happen here, before the service is
# halted.

function customer_defined_halt_cmds
{
# ADD customer defined halt commands.
: # do nothing instruction, because a function must contain some command.
date >> /tmp/pkg1.datelog
echo 'Halting pkg1' >> /tmp/pkg1.datelog
test_return 52
}

# END OF CUSTOMER DEFINED FUNCTIONS
```

사용자 정의 함수에 Serviceguard 명령 추가

패키지 제어 스크립트의 **Customer Defined Functions** 영역에 `cmmodpkg`와 같은 **Serviceguard** 명령을 추가할 수 있습니다. 그러나 이 명령은 패키지과 직접 상호 작용하면 안 됩니다. 또한 **Serviceguard** 명령이 다른 패키지과 상호 작용할 경우에는 **Serviceguard** 명령을 사용하는 모든 패키지에서 명령 루프가 있는지 확인해야 합니다.

예를 들어, 명령 루프는 다음과 같은 경우에 발생할 수 있습니다. **Pkg1**이 **Pkg2**의 `cmmodpkg -d`를 실행하고, **Pkg2**가 **Pkg1**의 `cmmodpkg -d`를 실행한다고 가정합니다. **Pkg1**과 **Pkg2**가 모두 동시에 시작될 경우 **Pkg1**은 **Pkg2**의 `cmmodpkg`를 실행하려고 시도합니다. 그러나 `cmmodpkg` 명령은 **Pkg2** 시작 과정이 완료될 때까지 기다려야 합니다. 마찬가지로 **Pkg2**는 **Pkg1**의 `cmmodpkg`를 실행하려고 시도하지만 **Pkg2**는 **Pkg1**의 시작 과정이 완료될 때까지 기다려야 하므로 명령 루프가 발생합니다.

이러한 명령 루프를 방지하려면 특히 제어 스크립트에서 **Serviceguard** 명령을 사용하는 패키지를 포함한 모든 패키지에 대해 항상 `RUN_SCRIPT_TIMEOUT`과 `HALT_SCRIPT_TIMEOUT`을 지정하는 것이 좋습니다. 위에서 설명한 것처럼 시간 종료 값이 지정되지 않고 구성에 명령 루프가 있을 경우 중단된 클러스터를 포함하여 일관되지 못한 결과가 발생할 수 있습니다.

추가 제품 지원

패키지 제어 스크립트 템플릿에서는 **Continuous Access/CA**를 사용하는 **Metro** 클러스터, **EMC SRDF**를 사용하는 **Metro** 클러스터, **HA NFS** 도구 키트 등의 추가 제품에 사용할 수 있는 종료 명령을 제공합니다. 제어 스크립트에 제공되는 후크를 사용하여 패키지를 만드는 방법에 대한 자세한 내용은 추가 제품의 설명서를 참조하십시오.

패키지 구성 확인

Serviceguard에서는 사용자가 입력하는 구성을 자동으로 검사하여 오류가 있으면 보고합니다.

Serviceguard Manager에서 구성 파일을 만든 경우에는 선택 단추나 적용 단추를 클릭합니다.

ASCII 패키지 구성 파일을 편집한 경우에는 다음 명령을 사용하여 파일의 내용을 확인합니다.

```
# cmcheckconf -v -P /etc/cmcluster/pkg1/pkg1.config
```

표준 출력 결과에 오류가 표시됩니다. 필요한 경우 파일을 편집하여 오류를 수정한 다음 오류가 발생하지 않고 완료될 때까지 명령을 다시 실행합니다.

Serviceguard Manager의 선택 단추나 cmcheckconf 명령을 사용하여 다음 사항을 확인합니다.

- 패키지 이름이 올바른지, 하나 이상의 NODE_NAME 항목이 포함되어 있는지 여부
- 매개 변수 항목이 중복되지 않는지 여부
- 매개 변수의 값이 허용 범위를 넘지 않는지 여부
- 클러스터의 모든 노드에 실행 및 중지 스크립트가 있고 해당 스크립트가 실행 가능한지 여부
- 실행 및 중지 시간 종료 값이 4294초 미만인지 여부
- 구성된 리소스를 클러스터 노드에서 사용할 수 있는지 여부

구성 배포

Serviceguard Manger 또는 HP-UX 명령을 사용하여 클러스터의 노드 간에 이진 클러스터 구성 파일을 배포할 수 있습니다.

Serviceguard Manager를 사용하여 구성 파일 및 제어 스크립트 배포

Serviceguard Manager에서 패키지 만들기를 완료한 다음 적용 단추를 클릭합니다. 그러면 이진 구성 파일이 만들어지고 클러스터 노드에 자동으로 배포됩니다. 제어 스크립트도 배포됩니다. 유도 모드를 선택하면 Serviceguard Manger에서 자동으로 제어 스크립트를 만듭니다. 제어 스크립트를 사용자가 직접 편집한 경우에는 마지막으로 저장된 버전을 사용합니다.

HP-UX 명령을 사용하여 패키지 제어 스크립트 복사

HP-UX 명령을 사용하여 구성 노드의 패키지 제어 스크립트를 패키지 실행이 가능한 모든 노드의 동일한 경로 이름에 복사할 수 있습니다. 원하는 파일 전송 방법(예: rcp 또는 ftp)을 사용합니다. 예를 들어, *ftsys9*에서 rcp 명령을 실행하여 *ftsys10*에 패키지 제어 스크립트를 복사할 수 있습니다.

```
# rcp /etc/cmcluster/pkg1/control.sh  
ftsys10:/etc/cmcluster/pkg1/control.sh
```

HP-UX 명령을 사용하여 이진 클러스터 구성 파일 배포

ASCII 클러스터 및 패키지 구성 파일이 있는 노드에서 다음 단계를 수행합니다.

- 다음 명령을 사용하여 구성 파일이 올바른지 확인합니다.

```
# cmcheckconf -C /etc/cmcluster/cmcl.config -P \  
/etc/cmcluster/pkg1/pkg1.config
```
- 잠금 디스크를 초기화할 수 있도록 클러스터 잠금 볼륨 그룹을 활성화합니다.

```
# vgchange -a y /dev/vg01
```

- 이진 구성 파일을 생성하여 여러 노드에 배포합니다.

```
# cmapplyconf -v -C /etc/cmcluster/cmcl.config -P \  
/etc/cmcluster/pkg1/pkg1.config
```

- 잠금 디스크를 사용하고 있는 경우 클러스터 잠금 볼륨 그룹을 비활성화합니다.

```
# vgchange -a n /dev/vg01
```

cmapplyconf 명령은 이전 버전의 클러스터 구성 파일을 만들어 클러스터의 모든 노드에 배포합니다. 이 작업을 통해 모든 노드에서 파일 내용이 일관되게 유지됩니다.

주

클러스터 및 패키지 구성 파일이 변경될 때마다 cmcheckconf와 cmapplyconf를 사용해야 합니다.

클러스터 및 패키지 작동 테스트

Serviceguard 클러스터를 구성하는 동안 장애가 발생하면 클러스터의 여러 구성 요소가 제대로 작동하는지 테스트하는 것이 좋습니다. 패키지, 노드 또는 LAN에 장애가 발생했을 때 클러스터에서 올바르게 응답하는지 테스트하는 방법에 대한 자세한 내용은 “클러스터 문제 해결” 장을 참조하십시오.

7 클러스터 및 패키지 유지 관리

이 장에서는 클러스터 구성 및 상태 정보를 확인하는 방법, 클러스터 또는 개별 노드를 시작하고 중지하는 방법, 영구적인 재구성을 수행하는 방법, 클러스터의 일상적인 유지 관리 중에 패키지를 시작, 중지, 이동 및 수정하는 방법에 대해 설명합니다. 이 장에서 다루는 내용은 다음과 같습니다.

- 클러스터 및 패키지 상태 확인
- 클러스터 및 노드 관리
- 패키지 및 서비스 관리
- 클러스터 재구성
- 패키지 재구성
- 클러스터 이벤트에 대응
- 시스템에서 Serviceguard 제거

대부분의 관리 작업은 Serviceguard Manager 또는 HP-UX 명령을 사용하여 수행할 수 있습니다.

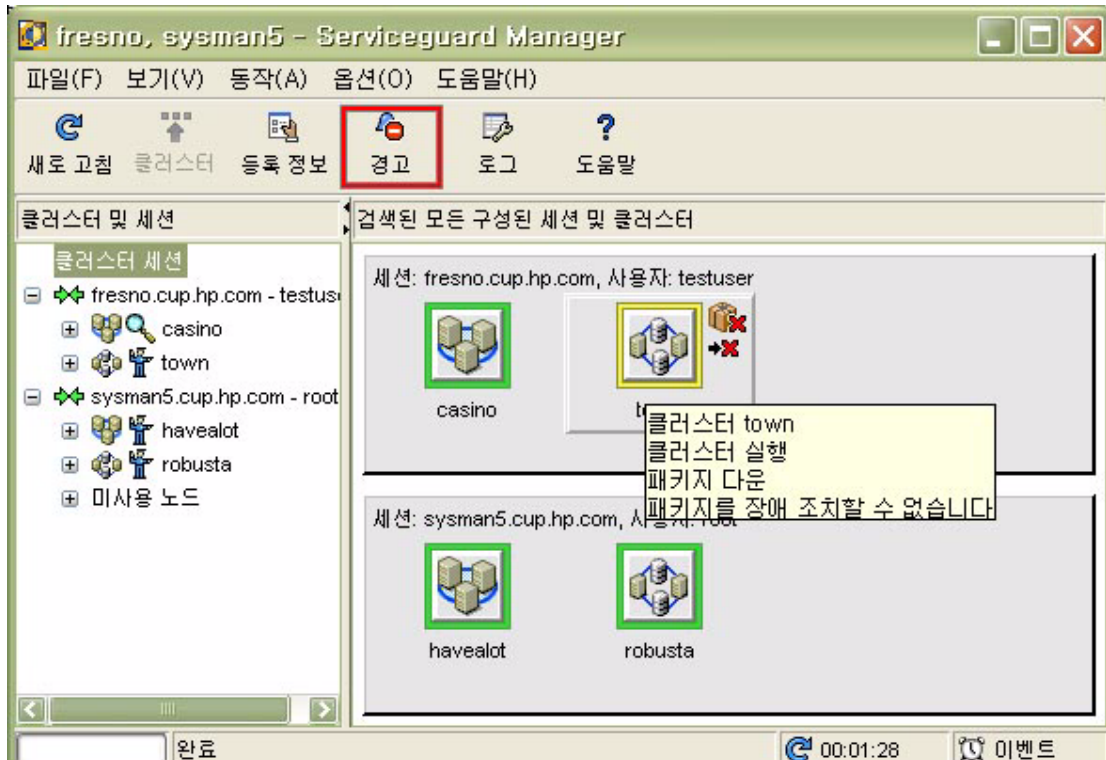
클러스터 및 패키지 상태 확인

Serviceguard Manager를 사용하거나 클러스터 노드의 명령줄에서 상태를 확인할 수 있습니다.

Serviceguard Manger를 사용하여 클러스터 및 패키지 상태 확인

Serviceguard Manger에서는 여러 가지 방식으로 상태를 보여 줍니다.

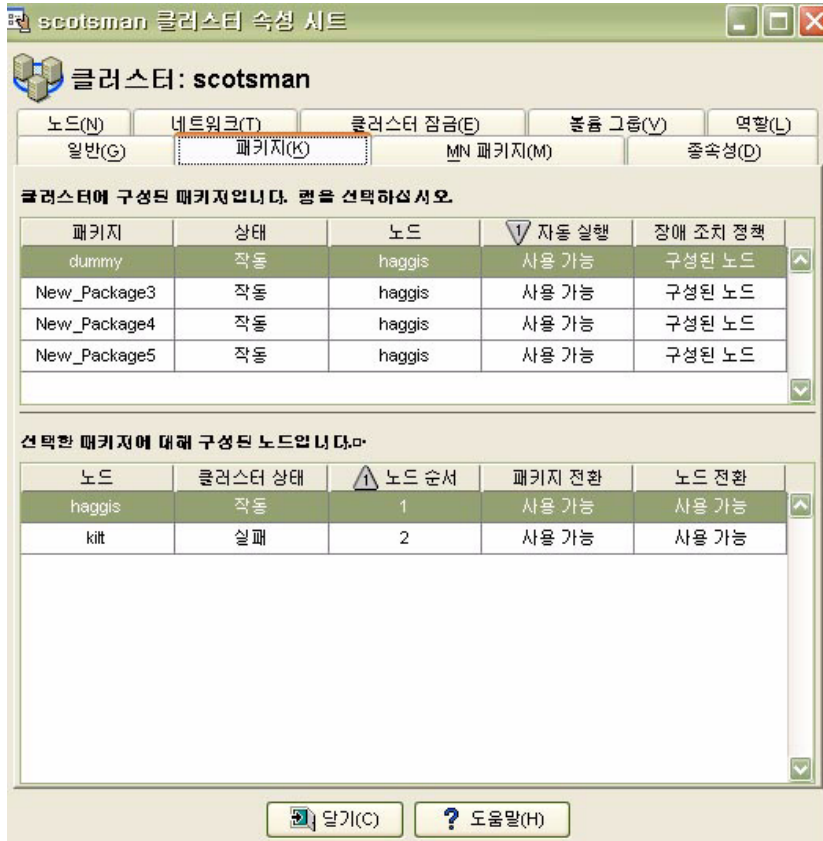
그림 7-1 상태 확인: Serviceguard Manager 맵



- 맵에서 클러스터 객체 아이콘에 테두리가 있으면 문제가 있다는 것을 나타냅니다. 아이콘의 오른쪽에 있는 배지를 보면 문제의 종류에 대한 정보를 알 수 있습니다. 그 위에 마우스를 놓으면 정보가 팝업으로 표시됩니다.

- 클러스터, 노드 및 패키지 속성 시트에는 자세한 정보가 표시됩니다.

그림 7-2 상태 확인: Serviceguard Manager 속성 시트



cmviewcl 명령을 사용하여 클러스터 및 패키지 상태 확인

클러스터 상태에 대한 정보는 클러스터의 개별 노드에 대해 유지 관리되는 상태 데이터베이스에 저장됩니다. cmviewcl 명령을 실행하여 이 데이터베이스에 포함된 정보를 표시할 수 있습니다.

```
# cmviewcl -v
```

비 루트 액세스를 사용하여 `cmviewcl` 명령을 실행할 수 있습니다. **Serviceguard** 버전 11.16 이상이 설치된 클러스터에서는 클러스터 구성 파일에 모니터링 역할을 만듭니다. 이전 버전에서는 `cmclodelist` 파일에 비 루트 쌍(<nodename> <nonrootuser>)에 추가합니다.

`-v` 옵션을 사용하여 `cmviewcl` 명령을 실행하면 실행 중인 클러스터의 모든 노드와 패키지에 대한 정보와 장애 조치 동작을 결정하는 매개 변수의 설정 값이 표시됩니다.

팁

대규모 구성에서 일부 명령은 완료되는 데 더 오랜 시간이 소요됩니다. 특히 `cmviewcl -v`를 실행하는 동안에는 패키지 및 서비스의 수가 많아질수록 **Serviceguard**의 CPU 사용량도 많아질 수 있습니다.

이전의 특정 릴리즈에서 사용하던 형식으로 출력하려면 `-r` 옵션으로 원하는 릴리즈를 지정하면 됩니다. 예를 들면 다음과 같습니다.

```
# cmviewcl -r A.11.09
```

다른 `cmviewcl` 옵션에 대한 자세한 내용은 맨페이지를 참조하십시오.

클러스터 및 패키지 상태의 종류

클러스터나 클러스터의 구성 요소 노드의 상태는 시기에 따라 여러 가지로 달라질 수 있습니다. 다음 절에서는 클러스터나 패키지의 일반적인 여러 상태에 대해 설명합니다.

클러스터 상태

클러스터의 상태는 다음 중 하나일 수 있습니다.

- 작동 — 하나 이상의 노드에 작동 중인 클러스터 데몬이 있고 재구성이 일어나지 않습니다.
- 작동 중지 — 어떤 클러스터 노드에도 작동 중인 클러스터 데몬이 없습니다.
- 시작 중 — 클러스터가 활성 구성원을 확인하는 과정에 있습니다. 하나 이상의 클러스터 데몬이 실행 중입니다.
- 알 수 없음 — `cmviewcl` 명령이 실행된 노드가 클러스터의 다른 노드와 통신할 수 없습니다.

노드 상태 및 세부 상태 노드의 상태는 클러스터 데몬이 실행 중인지 여부에 따라 작동 (클러스터의 구성원으로서 **활성** 상태) 또는 작동 중지(클러스터에서 **비활성** 상태) 중 하나가 됩니다. 클러스터에서 볼 때 중지 상태인 노드라도 실제로는 여전히 실행 중이며 HP-UX를 실행하고 있을 수 있습니다.

또한 노드는 다음 세부 상태 중 하나에 있을 수 있습니다.

- **실패** — 노드는 이 상태에서는 자신을 볼 수 없습니다. 노드가 활성 상태에 있다가 비활성 상태로 되었으며 중지되지는 않은 경우 클러스터의 다른 활성 구성원에서는 이 노드가 실패 상태로 표시됩니다.
- **수정** — 클러스터가 재편성 중일 때 노드는 이 상태가 됩니다. 이 상태의 노드는 모든 노드가 활성 클러스터의 새로운 구성원 정보에 동의하도록 하는 프로토콜을 실행하는 중입니다. 동의에 도달하면 상태 데이터베이스는 새로운 클러스터 구성원 정보를 반영하도록 업데이트됩니다.
- **실행 중** — 이 상태의 노드는 마지막 재편성에 필요한 모든 작업을 완료하고 정상적으로 작동 중입니다.
- **중지됨** — 노드는 이 상태에서는 자신을 볼 수 없습니다. `cmhaltnode` 명령을 사용하는 경우처럼 노드가 활성 클러스터에서 정상적으로 나간 후에 다른 노드에는 해당 노드가 이 상태로 표시됩니다.
- **알 수 없음** — 노드는 이 상태에서는 자신을 볼 수 없습니다. 활성 클러스터 구성원이었던 적이 없는 노드에 대해 다른 노드에서는 이 상태로 할당합니다.

패키지 상태 및 세부 상태 패키지의 상태는 다음 중 하나일 수 있습니다.

- **작동** — 패키지 제어 스크립트가 활성 상태입니다.
- **작동 중지** — 패키지 제어 스크립트가 활성 상태가 아닙니다.
- **알 수 없음**

패키지의 **세부 상태**는 다음 중 하나일 수 있습니다.

- **시작 중** — 제어 스크립트의 시작 명령이 실행 중입니다.
- **실행 중** — 서비스가 활성 상태이며 모니터링되고 있습니다.
- **중지 중** — 제어 스크립트의 중지 명령이 실행 중입니다.

패키지 전환 속성 패키지에는 다음과 같은 전환 속성도 있습니다.

- **자동 실행** — 사용 가능한 장애가 발생할 경우 패키지가 다른 노드로 전환할 수 있음을 의미합니다.

- **노드 전환 활성화** — 사용 가능한 패키지가 참조되는 노드로 전환할 수 있음을 의미합니다. 사용 불가능은 `cmmodpkg` 명령으로 노드에서 패키지를 사용할 수 있도록 설정하기 전까지는 지정한 노드로 전환할 수 없음을 의미합니다.

모든 패키지는 패키지의 기본 노드 또는 대체 노드가 되는 각 노드에 대해 사용 가능 또는 사용 불가능으로 표시됩니다.

서비스 상태 서비스의 상태는 다음 중 하나입니다.

- **작동** — 서비스가 모니터링되고 있습니다.
- **작동 중지** — 서비스가 실행되고 있지 않습니다. 서비스가 중지되었거나 장애가 발생했을 수 있습니다.
- **초기화되지 않음** — 서비스가 클러스터 구성에 포함되어 있지만 제어 스크립트의 실행 명령으로 시작되지는 않았습니다.
- **알 수 없음**

네트워크 상태 네트워크 인터페이스의 상태는 다음 중 하나입니다.

- **작동**
- **작동 중지**
- **알 수 없음** — 인터페이스가 작동 중인지 작동 중단 상태인지 확인할 수 없습니다. 이것은 클러스터가 작동 중단 상태일 때 발생할 수 있습니다. 대기 인터페이스는 이 상태만 갖습니다.

직렬 회선 상태 직렬 회선의 상태는 다음 중 하나입니다.

- **작동** — 직렬 회선을 통해 하트비트가 수신됩니다.
- **작동 중지** — `NODE_TIMEOUT` 값의 2배 크기의 시간 내에 직렬 회선을 통해 하트비트가 수신되지 않았습니다.
- **복구 중** — 직렬 회선으로 잘못된 메시지가 수신되어 다시 동기화 과정이 진행되고 있습니다.
- **알 수 없음** — 직렬 회선이 작동 중인지 작동 중단 상태인지 확인할 수 없습니다. 이 상태는 원격 노드가 작동 중단될 때 발생할 수 있습니다.

장애 조치 및 장애 복구 정책 `FAILOVER_POLICY` 매개 변수에 두 값 중 하나를 사용하여 패키지를 구성할 수 있습니다.

- `CONFIGURED_NODE` — 장애 조치 시 패키지가 패키지 구성 파일의 노드 목록에 있는 다음 노드로 전달됩니다.

- `MIN_PACKAGE_NODE` — 장애 조치 시 패키지가 가장 적은 수의 패키지를 실행하고 있는 노드로 전달됩니다.

또한 `FAILBACK_POLICY` 매개 변수에 두 값 중 하나를 사용하여 패키지를 구성할 수도 있습니다.

- `AUTOMATIC` — 이 설정을 사용하는 패키지는 장애 조치 후 기본 노드가 다시 사용 가능하게 되면 기본 노드로 돌아갑니다.
- `MANUAL` — 이 설정을 사용하는 패키지는 장애 조치 후 시스템 관리자가 직접 원래 노드로 되돌려야 합니다.

장애 조치 및 장애 복구 정책은 `cmviewcl -v` 명령의 출력에 표시됩니다.

클러스터 및 패키지 상태의 예

다음은 `cmviewcl -v` 명령의 출력 예로서, 샘플 구성의 클러스터 상태를 보여 줍니다.

정상 실행 상태 클러스터의 두 노드가 모두 실행 중이고 패키지는 주 위치에 있는 정상적인 실행 모습을 보여 줍니다.

```

CLUSTER      STATUS
example      up
  NODE      STATUS      STATE
  ftsys9    up          running

Network_Parameters:
INTERFACE    STATUS      PATH      NAME
PRIMARY      up          56/36.1   lan0
STANDBY      up          60/6      lan1

PACKAGE      STATUS      STATE      AUTO_RUN  NODE
pkg1         up          running    enabled   ftsys9

Policy_Parameters:
POLICY_NAME    CONFIGURED_VALUE
Failover       configured_node
Failback       manual

Script_Parameters:
ITEM          STATUS      MAX_RESTARTS  RESTARTS
NAME
Service      up          0              0
servicel

```

```

Subnet          up          0          0
15.13.168.0

Node_Switching_Parameters:
NODE_TYPE      STATUS      SWITCHING    NAME
Primary        up          enabled      ftsys9      (cur
rent)
Alternate      up          enabled      ftsys10

NODE           STATUS      STATE
ftsys10       up          running

Network_Parameters:
INTERFACE      STATUS      PATH          NAME
PRIMARY        up          28.1          lan0
STANDBY        up          32.1          lan1

PACKAGE        STATUS      STATE          AUTO_RUN      NODE
pkg2           up          running       enabled       ftsys1
0

Policy_Parameters:
POLICY_NAME     CONFIGURED_VALUE
Failover        configured_node
Failback        manual

Script_Parameters:
ITEM           STATUS      MAX_RESTARTS  RESTARTS
NAME
Service        up          0             0
service2
Subnet         up          0             0
15.13.168.0

Node_Switching_Parameters:
NODE_TYPE      STATUS      SWITCHING    NAME
Primary        up          enabled      ftsys10      (cur
rent)
Alternate      up          enabled      ftsys9

```

쿼럼 서버 상태 클러스터에서 타이 브레이크 서비스에 쿼럼 서버를 사용할 경우에는 다음에 표시된 `cmviewcl -v`의 일부 출력에서처럼 각 노드 항목 뒤에 서버 이름, 상태, 세부 상태 등이 표시됩니다.

```

CLUSTER      STATUS
example      up

      NODE      STATUS      STATE
      ftsys9    up          running

Quorum Server Status:
      NAME      STATUS      STATE
      lp-qs     up          running
...

      NODE      STATUS      STATE
      ftsys10   up          running

Quorum Server Status:
      NAME      STATUS      STATE
      lp-qs     up          running

```

CVM 패키지 상태 클러스터에서 디스크 저장 장치에 VERITAS Cluster Volume Manager를 사용하는 경우 응용 프로그램에서 CVM 디스크 그룹에 액세스하려면 모든 활성 노드에서 시스템 다중 노드 패키지인 CVM-VxVM-pkg가 실행되어야 합니다. 이 패키지는 다음과 같은 cmviewcl 명령의 출력에서 볼 수 있습니다.

```

CLUSTER      STATUS
example      up

      NODE      STATUS      STATE
      ftsys7    down       halted
      ftsys8    down       halted
      ftsys9    up         running
      ftsys10   up         running

SYSTEM_MULTI_NODE_PACKAGES:

      PACKAGE   STATUS      STATE
      VxVM-CVM-pkg up          running

```

-v 옵션을 사용하면 다음과 같이 클러스터의 각 활성 노드와 관련된 시스템 다중 노드 패키지가 표시됩니다.

클러스터 및 패키지 유지 관리
클러스터 및 패키지 상태 확인

SYSTEM_MULTI_NODE_PACKAGES:

```

PACKAGE      STATUS      STATE
VxVM-CVM-pkg up          running

NODE         STATUS      STATE
ftsys7       down        halted

NODE         STATUS      STATE
ftsys8       down        halted

NODE         STATUS      STATE
ftsys9       up          running
Script_Parameters:
ITEM        STATUS      MAX_RESTARTS  RESTARTS  NAME
Service     up          0              0
VxVM-CVM-pkg.srv

NODE         STATUS      STATE
ftsys10      up          running
Script_Parameters:
ITEM        STATUS      MAX_RESTARTS  RESTARTS  NAME
Service     up          0              0
VxVM-CVM-pkg.srv

```

패키지 중지 후의 상태 cmhaltpkg 명령으로 pkg2를 중지한 이후 cmviewcl-v의 출력은 다음과 같습니다.

```

CLUSTER      STATUS
example      up

NODE         STATUS      STATE
ftsys9       up          running

Network_Parameters:
INTERFACE    STATUS      PATH          NAME
PRIMARY     up          56/36.1       lan0
STANDBY     up          60/6          lan1

PACKAGE      STATUS      STATE          AUTO_RUN      NODE
pkg1         up          running        enabled        ftsys9

Policy_Parameters:
POLICY_NAME    CONFIGURED_VALUE
Failover       configured_node

```

```

Failback          manual

Script_Parameters:
ITEM              STATUS  MAX_RESTARTS  RESTARTS      NAME
Service          up      0              0              servi
cel
Subnet           up      0              0              15.13
.168.0
Resource        up      /exam
ple/float

Node_Switching_Parameters:
NODE_TYPE        STATUS  SWITCHING     NAME
Primary         up      enabled       ftsys9        (cur
rent)
Alternate       up      enabled       ftsys10

NODE              STATUS    STATE
ftsys10          up        running

Network_Parameters:
INTERFACE        STATUS    PATH          NAME
PRIMARY         up        28.1          lan0
STANDBY         up        32.1          lan1

UNOWNED_PACKAGES

PACKAGE          STATUS    STATE          AUTO_RUN      NODE
pkg2             down     unowned       disabled      unowne
d

Policy_Parameters:
POLICY_NAME      CONFIGURED_VALUE
Failover         configured_node
Failback         manual

Script_Parameters:
ITEM              STATUS    NODE_NAME      NAME
Resource        down     ftsys9         /example/float
Subnet          up       ftsys9         15.13.168.0
Resource        up       ftsys10        /example/float
Subnet          up       ftsys10        15.13.168.0

Node_Switching_Parameters:

```

클러스터 및 패키지 유지 관리
 클러스터 및 패키지 상태 확인

| NODE_TYPE | STATUS | SWITCHING | NAME |
|-----------|--------|-----------|---------|
| Primary | up | enabled | ftsys10 |
| Alternate | up | enabled | ftsys9 |

이제 **pkg2**의 상태는 “중지됨”이 되고 **Unowned** 상태에서처럼 표시되며 패키지 전환은 사용할 수 없게 됩니다. **pkg2**에 의존하도록 구성된 “/example/float” 리소스는 한 노드에서 작동 중지됩니다. 그러나 두 노드 모두에서 전환은 가능합니다. 즉, 패키지에서 전체 전환을 다시 사용할 수 있게 되면 기본 노드에서 패키지를 시작하려고 시도합니다.

패키지를 다른 노드로 이동한 후의 상태 다음 명령을 실행하십시오.

cmrunpkg -n ftsys9 pkg2

cmviewcl -v 명령의 출력은 다음과 같습니다.

| CLUSTER | STATUS | | | |
|---------------------|------------------|--------------|----------|--------|
| example | up | | | |
| NODE | STATUS | STATE | | |
| ftsys9 | up | running | | |
| Network_Parameters: | | | | |
| INTERFACE | STATUS | PATH | NAME | |
| PRIMARY | up | 56/36.1 | lan0 | |
| STANDBY | up | 60/6 | lan1 | |
| PACKAGE | STATUS | STATE | AUTO_RUN | NODE |
| pkg1 | up | running | enabled | ftsys9 |
| Policy_Parameters: | | | | |
| POLICY_NAME | CONFIGURED_VALUE | | | |
| Failover | configured_node | | | |
| Failback | manual | | | |
| Script_Parameters: | | | | |
| ITEM | STATUS | MAX_RESTARTS | RESTARTS | NAME |
| Service | up | 0 | 0 | servi |
| cel | Subnet | up | 0 | 15.13 |
| .168.0 | Resource | up | | /exam |

```

ple/float

Node_Switching_Parameters:
NODE_TYPE      STATUS      SWITCHING     NAME
Primary        up          enabled       ftsys9        (cur
rent)
Alternate      up          enabled       ftsys10

PACKAGE        STATUS      STATE          AUTO_RUN      NODE
pkg2           up          running        disabled      ftsys9

Policy_Parameters:
POLICY_NAME     CONFIGURED_VALUE
Failover        configured_node
Failback        manual

Script_Parameters:
ITEM            STATUS      NAME           MAX_RESTARTS
RESTARTS
Service         up          service2.1     0
0
Subnet          up          15.13.168.0   0
0
Node_Switching_Parameters:
NODE_TYPE      STATUS      SWITCHING     NAME
Primary        up          enabled       ftsys10
Alternate      up          enabled       ftsys9        (curre
nt)

NODE            STATUS      STATE
ftsys10         up          running

Network_Parameters:
INTERFACE      STATUS      PATH           NAME
PRIMARY        up          28.1           lan0
STANDBY        up          32.1           lan1

```

현재 **pkg2**는 노드 **ftsys9**에서 실행 중입니다. 이 패키지는 여전히 전환이 불가능합니다.

자동 실행이 사용 가능으로 설정된 후의 상태 다음 명령을 사용하면 패키지 전환 플래그가 다시 자동 실행 사용 가능으로 변경됩니다.

```
# cmmodpkg -e pkg2
```

이제 `cmviewcl` 명령의 출력은 다음과 같습니다.

```
CLUSTER      STATUS
example      up

      NODE      STATUS      STATE
      ftsys9    up          running

      PACKAGE  STATUS      STATE      AUTO_RUN  NODE
      pkg1     up          running   enabled   ftsys9
      pkg2     up          running   enabled   ftsys9

      NODE      STATUS      STATE
      ftsys10   up          running
```

두 패키지 모두가 현재 `ftsys9`에서 실행 중이고 `pkg2`는 전환이 가능합니다. `Ftsys10`은 데몬을 실행 중이며 `ftsys10`에서 실행 중인 패키지는 없습니다.

노드 중지 후의 상태 다음 명령을 사용하여 `ftsys10`을 중지하십시오.

```
# cmhaltnode ftsys10
```

`ftsys9`에서 `cmviewcl`을 실행할 경우 출력은 다음과 같습니다.

```
CLUSTER      STATUS
example      up

      NODE      STATUS      STATE
      ftsys9    up          running

      PACKAGE  STATUS      STATE      AUTO_RUN  NODE
      pkg1     up          running   enabled   ftsys9
      pkg2     up          running   enabled   ftsys9

      NODE      STATUS      STATE
      ftsys10   down        halted
```

이 출력 결과는 `ftsys9`와 `ftsys10` 모두에서 볼 수 있습니다.

RS232 상태 보기 하트비트 연결로 직렬(RS232) 회선을 사용하는 경우 `cmviewcl -v` 명령의 출력에는 구성된 RS232 장치 파일의 목록이 표시됩니다. 다음은 정상적으로 실행 중인 상태를 나타냅니다.


```

CLUSTER      STATUS
example      up
  NODE        STATUS      STATE
  ftsys9      up          running

  Network_Parameters:
  INTERFACE    STATUS      PATH        NAME
  PRIMARY      up          56/36.1     lan0

  Serial_Heartbeat:
  DEVICE_FILE_NAME  STATUS      CONNECTED_TO:
  /dev/tty0p0       up          ftsys10     /dev/tty0

p0
  NODE        STATUS      STATE
  ftsys10     up          running

  Network_Parameters:
  INTERFACE    STATUS      PATH        NAME
  PRIMARY      up          28.1        lan0

  Serial_Heartbeat:
  DEVICE_FILE_NAME  STATUS      CONNECTED_TO:
  /dev/tty0p0       up          ftsys9
  /dev/tty0p0

```

다음은 노드 *ftsys10*이 중지된 후의 상태를 보여 줍니다.

```

CLUSTER      STATUS
example      up
  NODE        STATUS      STATE
  ftsys9      up          running

  Network_Parameters:
  INTERFACE    STATUS      PATH        NAME
  PRIMARY      up          56/36.1     lan0

  Serial_Heartbeat:
  DEVICE_FILE_NAME  STATUS      CONNECTED_TO:
  /dev/tty0p0       unknown     ftsys10     /dev/tty0

p0
  NODE        STATUS      STATE
  ftsys10     down       running

  Network_Parameters:
  INTERFACE    STATUS      PATH        NAME
  PRIMARY      up          28.1        lan0

```

```

Serial_Heartbeat:
DEVICE_FILE_NAME      STATUS      CONNECTED_TO:
/dev/tty0p0           unknown    ftsys9   /dev/tty0p0

```

다음은 직렬 회선이 작동하지 않고 있을 때의 상태를 보여 줍니다.

```

CLUSTER      STATUS
example      up
  NODE      STATUS      STATE
  ftsys9    up          running

```

```

Network_Parameters:
INTERFACE    STATUS      PATH      NAME
PRIMARY      up          56/36.1   lan0

```

```

Serial_Heartbeat:
DEVICE_FILE_NAME      STATUS      CONNECTED_TO:
/dev/tty0p0           down        ftsys10   /dev/tty0p0
  NODE      STATUS      STATE
  ftsys10    up          running

```

```

Network_Parameters:
INTERFACE    STATUS      PATH      NAME
PRIMARY      up          28.1      lan0

```

```

Serial_Heartbeat:
DEVICE_FILE_NAME      STATUS      CONNECTED_TO:
/dev/tty0p0           down        ftsys9
/dev/tty0p0

```

무소속 패키지 데이터 보기 다음 예제에서는 무소속 패키지(Unowned Package), 즉 구성된 노드 어느 곳에서도 실행되고 있지 않은 패키지를 보여 줍니다. 패키지가 실행 될 수 있는 각 노드에 대해 모니터링되는 리소스에 대한 정보가 표시됩니다. 이를 이용하여 장애의 원인을 찾아내고 패키지의 재시작 위치를 결정할 수 있습니다.

```
UNOWNED_PACKAGES
```

```

PACKAGE      STATUS      STATE      AUTO_RUN      NODE
PKG3         down       halted     disabled      unowned

```

```

Policy_Parameters:
POLICY_NAME    CONFIGURED_VALUE
Failover       min_package_node
Failback       automatic

```

```
Script_Parameters:
```

```

ITEM          STATUS  NODE_NAME  NAME
Resource     up      manx       /resource/random
Subnet       up      manx       192.8.15.0
Resource     up      burmese   /resource/random
Subnet       up      burmese   192.8.15.0
Resource     up      tabby     /resource/random
Subnet       up      tabby     192.8.15.0
Resource     up      persian   /resource/random
Subnet       up      persian   192.8.15.0

```

Node_Switching_Parameters:

```

NODE_TYPE    STATUS    SWITCHING  NAME
Primary      up       enabled    manx
Alternate    up       enabled    burmese
Alternate    up       enabled    tabby
Alternate    up       enabled    persian

```

시스템 다중 노드 패키지 데이터 보기 다음 예제에서는 표준 Serviceguard 패키지뿐만 아니라 시스템 다중 노드 패키지도 포함된 클러스터를 보여 줍니다. 시스템 다중 노드 패키지는 클러스터의 모든 노드에서 실행되는 반면 표준 패키지는 한 번에 한 노드에서만 실행됩니다.

```

CLUSTER      STATUS
cats         up

```

```

NODE          STATUS    STATE
manx         up       running

```

```

PACKAGE      STATUS    STATE    AUTO_RUN  NODE
pkg1         up       running  enabled   manx

```

```

NODE          STATUS    STATE
tabby        up       running

```

```

PACKAGE      STATUS    STATE    AUTO_RUN  NODE
pkg2         up       running  enabled   tabby

```

SYSTEM_MULTI_NODE_PACKAGES:

```

PACKAGE      STATUS    STATE
VxVM-CVM-pkg up       running

```

클러스터 및 노드 관리

클러스터 관리에는 다음 작업이 포함됩니다.

- 모든 노드가 중지되었을 때 클러스터 시작
- 실행 중인 클러스터에 이미 구성된 노드 추가
- 실행 중인 클러스터에서 노드 제거
- 전체 클러스터 중지

Serviceguard 11.16 이상에서는 클러스터 구성 파일의 액세스 정책에 따라 루트 권한이 없는 사용자가 이러한 패키지 관리 및 클러스터 관리 명령을 사용할 수 있습니다. 액세스 구성에 대한 자세한 내용은 5장의 “보안 파일 편집”을 참조하십시오.

Serviceguard Manager나 **Serviceguard** 명령줄을 사용하여 클러스터를 시작 또는 중지하거나 노드를 추가 또는 중지할 수 있습니다. 클러스터를 시작한다는 것은 클러스터에 포함된 하나 이상의 노드에서 클러스터 데몬을 실행하는 것을 의미합니다. 모든 노드가 현재 중지되었는지(즉, 클러스터 데몬이 실행되고 있지 않은지) 또는 개별 노드에서 클러스터 데몬을 시작할 것인지에 따라 다양한 **Serviceguard** 명령을 사용하여 클러스터를 시작할 수 있습니다.

이 장에서 이미 구성된 노드를 클러스터에 추가하는 작업과 새 노드를 클러스터 구성에 추가하는 작업은 서로 다릅니다. 이미 구성된 노드는 클러스터 구성 파일에 이미 입력된 노드이며 새 노드는 클러스터 구성 파일을 수정함으로써 클러스터에 추가되는 노드입니다.

주

쿼럼 서버가 구성된 경우 클러스터 또는 개별 노드를 수동으로 시작하거나 중지하기 위해 쿼럼 서버에 액세스할 필요는 없습니다. 쿼럼 서버는 클러스터 파티션 후 타이 브레이크가 필요한 경우에만 사용됩니다.

모든 노드가 중지되었을 때 클러스터 시작

Serviceguard Manager나 **Serviceguard** 명령을 사용하여 클러스터를 시작할 수 있습니다.

Serviceguard Manager를 사용하여 클러스터 시작

클러스터 아이콘을 선택한 다음 마우스 오른쪽 단추를 클릭하여 동작 메뉴를 표시합니다. “클러스터 <clustername> 실행”을 선택합니다. 작업이 수행됨에 따라 진행 창에 메시지가 표시됩니다. 여기에는 각 노드 및 패키지의 시작에 대한 메시지가 포함됩니다. 작업이 완료되면 진행 창에서 확인을 클릭합니다.

Serviceguard 명령을 사용하여 클러스터 시작

모든 클러스터 노드가 중지된 경우 `cmruncl` 명령을 사용하여 클러스터를 시작합니다. 특정 명령 옵션은 특정 상황에서 클러스터를 시작하는 데 사용할 수 있습니다.

`-v` 옵션은 최대 메시지 수를 표시합니다. 다음 명령은 클러스터에 구성된 모든 노드를 시작하고 네트워크 정보를 확인합니다.

```
# cmruncl -v
```

기본적으로 `cmruncl`은 네트워크 유효성 검사를 수행하여 실제 네트워크 설정이 구성된 네트워크 설정과 일치하는지 확인합니다. 이 방법을 사용하는 것이 좋습니다. 최근에 네트워크를 확인한 적이 있는데 이 작업에 시간이 오래 걸렸던 경우에는 `-w none` 옵션을 사용하여 유효성 검사를 생략할 수 있습니다.

`-n` 옵션은 특정 노드 그룹을 지정합니다. 이 옵션을 사용하지 않으면 모든 노드가 시작됩니다. 다음 예제에서는 `ftsys9`와 `ftsys10`에서만 로컬로 구성된 클러스터를 시작합니다. 이 명령은 클러스터가 어떤 노드에서도 아직 실행되고 있지 않은 경우에만 사용해야 합니다.

```
# cmruncl -v -n ftsys9 -n ftsys10
```

경고

특정 클러스터 노드가 이미 클러스터를 실행 중인 동안 `cmruncl -n` 명령을 사용하여 클러스터를 시작하면 데이터의 무결성이 보장되지 않습니다. 노드 간에 네트워크 연결이 끊어진 경우 `cmruncl -n`을 사용하면 두 번째 클러스터가 편성되고 이 클러스터는 다른 클러스터에서 이미 실행 중인 동일한 응용 프로그램을 시작하게 됩니다. 따라서 두 응용 프로그램이 서로 디스크에 있는 다른 응용 프로그램의 데이터를 서로 덮어쓸 수 있습니다.

실행 중인 클러스터에 이미 구성된 노드 추가

Serviceguard Manager나 Serviceguard 명령을 사용하여 이미 구성된 노드를 실행 중인 클러스터로 가져올 수 있습니다.

Serviceguard Manager를 사용하여 클러스터에 구성된 노드 추가

노드 아이콘을 선택한 다음 마우스 오른쪽 단추를 클릭하여 동작 메뉴를 표시합니다. “노드 <hostname> 실행”을 선택합니다. 작업이 수행됨에 따라 진행 창에 메시지가 표시됩니다. 또한 노드에서 실행할 수 있는 패키지가 모두 시작됩니다. 작업이 완료되면 진행 창에서 확인을 클릭합니다.

Serviceguard 명령을 사용하여 실행 중인 클러스터에 이미 구성된 노드 추가

cmrunnode 명령을 사용하여 이미 실행 중인 클러스터에 하나 이상의 노드를 추가할 수 있습니다. 추가할 노드는 이미 클러스터 구성의 일부여야 합니다. 다음 예제에서는 노드 *ftsys9*와 *ftsys10*으로만 시작된 클러스터에 노드 *ftsys8*을 추가합니다.

```
# cmrunnode -v ftsys8
```

기본적으로 *cmruncl*은 네트워크 유효성 검사를 수행하여 실제 네트워크 설정이 구성된 네트워크 설정과 일치하는지 확인합니다. 이 방법을 사용하는 것이 좋습니다. 최근에 네트워크를 확인한 적이 있는데 이 작업에 시간이 오래 걸렸던 경우에는 *-w none* 옵션을 사용하여 유효성 검사를 생략할 수 있습니다.

노드의 클러스터가 이미 실행 중이므로 해당 노드는 클러스터에 참가하고 패키지가 시작될 수 있습니다. 노드가 실행 중인 해당 클러스터를 찾지 못하거나 노드가 클러스터 구성의 일부가 아니면 이 명령은 실패합니다.

실행 중인 클러스터에서 노드 제거

Serviceguard Manager나 HP-UX 명령을 사용하여 클러스터에서 노드를 제거할 수 있습니다. 이 작업에서는 클러스터 데몬을 중지하여 클러스터에서 노드를 제거하지만 클러스터 구성을 수정하지는 않습니다. 클러스터 구성에서 노드를 영구적으로 제거하려면 클러스터 구성 파일을 다시 작성해야 합니다. 자세한 내용은 다음 절을 참조하십시오.

Serviceguard Manager를 사용하여 클러스터에서 노드 제거

노드 아이콘을 선택한 다음 마우스 오른쪽 단추를 클릭하여 동작 메뉴를 표시합니다. “노드 <nodename> 중지”를 선택합니다. 작업이 수행됨에 따라 진행 창에 메시지가 표시됩니다. 이 작업에는 해당되는 경우 한 노드에서 대체 노드로 패키지를 이동하는 작업이 포함됩니다. 작업이 완료되면 진행 창에서 확인을 클릭합니다.

Serviceguard 명령을 사용하여 작업에서 노드 제거

cmhaltnode 명령을 사용하여 클러스터에서 하나 이상의 노드를 중지할 수 있습니다. 이 경우 지정된 노드의 클러스터 데몬이 중지된 다음 클러스터에 참가한 활성 노드가 제거됩니다.

실행 중인 패키지를 포함하여 노드를 중지하려면 `-f` 옵션을 사용합니다. 대체 노드로 전환될 수 있는 패키지가 실행 중이면 해당 패키지는 대체 노드로 전환되어 대체 노드에서 시작됩니다. 예를 들어, 다음 명령은 샘플 구성의 노드 `ftsys9`에서 실행 중인 Serviceguard 데몬을 중지하고 `ftsys9`에서 실행 중인 패키지를 `ftsys10`으로 이동합니다.

```
# cmhaltnode -f -v ftsys9
```

이 명령은 각 패키지의 제어 스크립트에서 중지 명령을 실행하여 노드 `ftsys9`에서 실행 중인 모든 패키지를 중지합니다. 그러면 `ftsys9`가 중지되고 패키지는 대체 노드 `ftsys10`에서 시작됩니다.

cmhaltnode를 사용하면 편리하게 시스템 유지 관리를 위해 노드를 중지하면서 해당 패키지는 다른 노드에서 계속 사용할 수 있도록 할 수 있습니다. 유지 관리가 끝나면 패키지는 기본 노드로 돌아갈 수 있습니다. 다음의 “패키지 이동”을 참조하십시오.

노드를 클러스터로 다시 보내려면 `cmrunnode`를 사용합니다.

주

특히 종료할 때 패키지 응용 프로그램에 문제가 발생하여 제대로 중지되지 않을 경우를 대비하여 HP-UX의 `shutdown` 명령을 실행하기 전에 `cmhaltnode`를 실행하는 것이 좋습니다.

전체 클러스터 중지

Serviceguard Manager 또는 Serviceguard 명령을 사용하여 실행 중인 클러스터를 중지할 수 있습니다.

Serviceguard Manager를 사용하여 클러스터 중지

클러스터를 선택한 다음 마우스 오른쪽 단추를 클릭하여 동작 메뉴를 표시합니다. “클러스터 <clustername> 중지”를 선택합니다. 작업이 수행됨에 따라 진행 창에 메시지가 표시됩니다. 여기에는 각 패키지 및 노드의 중지 상태에 대한 메시지가 포함됩니다. 작업이 완료되면 진행 창에서 확인을 클릭합니다.

Serviceguard 명령을 사용하여 클러스터 중지

cmhaltcl 명령을 사용하여 전체 클러스터를 중지할 수 있습니다. 이 명령은 구성된 클러스터의 모든 노드에서 Serviceguard 데몬을 중지합니다. -f 옵션을 사용하면 패키지가 실행 중인 경우에도 강제로 클러스터를 중지할 수 있습니다. 이 명령은 실행 중인 모든 노드에서 실행할 수 있습니다. 예를 들면 다음과 같습니다.

```
# cmhaltcl -f -v
```

이 명령은 클러스터에 구성된 모든 노드를 중지합니다.

클러스터 자동 재시작

장시간의 전원 장애가 발생하여 클러스터의 모든 노드가 중지되는 것과 같은 경우에 클러스터가 자동으로 다시 시작되도록 구성할 수 있습니다. 이렇게 하려면 /etc/rc.config.d/cmcluster 파일에서 AUTOSTART_CMCLD를 1로 설정합니다.

패키지 및 서비스 관리

패키지 및 서비스 관리 작업에는 다음 작업이 포함됩니다.

- 패키지 시작
- 패키지 중지
- 패키지 이동(중지한 다음 시작)
- 패키지 전환 동작 변경

Serviceguard 11.16 이상에서는 클러스터 구성 파일의 액세스 정책에 따라 루트 권한이 없는 사용자가 이러한 명령을 사용할 수 있습니다. 액세스 구성에 대한 자세한 내용은 5장의 “보안 파일 편집”을 참조하십시오.

Serviceguard Manager나 **Serviceguard** 명령을 사용하여 이러한 작업을 수행할 수 있습니다.

패키지 시작

일반적으로 클러스터의 일부로 구성된 패키지는 클러스터가 시작될 때 기본 노드에서 시작됩니다. 패키지를 수동으로 중지한 후에는 수동으로 시작해야 할 수도 있습니다. **Serviceguard Manager**나 **Serviceguard** 명령줄을 사용하여 이 작업을 수행할 수 있습니다.

Serviceguard Manager를 사용하여 패키지 시작

시작할 패키지를 선택하고 마우스 오른쪽 단추를 클릭하여 동작 목록을 표시합니다. 구성된 기본 노드나 패키지 노드 목록의 모든 노드에서 패키지를 시작할 수 있습니다. 패키지를 시작할 노드를 선택할 수 있는 사용 가능한 실행 노드 목록에서 “패키지 <pkgname> 실행” 또는 “노드에서 패키지 <pkgname> 실행”을 선택합니다.

작업이 수행됨에 따라 진행 창에 메시지가 표시됩니다. 여기에는 패키지의 시작에 대한 메시지가 포함됩니다.

패키지를 시작하려면 클러스터가 실행 중이어야 합니다.

Serviceguard 명령을 사용하여 패키지 시작

다음과 같이 `cmrunpkg` 명령을 사용하여 특정 노드에서 패키지를 실행한 다음 `cmmodpkg` 명령을 사용하여 패키지의 전환을 사용 가능하게 할 수 있습니다.

```
# cmrunpkg -n ftsys9 pkg1
# cmmodpkg -e pkg1
```

이 명령은 `ftsys9`에서 패키지를 시작한 다음 패키지 전환을 활성화합니다. 패키지를 중지하면 전환이 비활성화되므로 패키지가 특정 노드에서 이미 중지되었던 경우에는 이 순서를 따라야 합니다.

패키지 중지

패키지 사용을 중지하려고 하지만 노드는 계속 작동되도록 하려면 **Serviceguard**의 패키지를 중지합니다. **Serviceguard Manager**를 사용하거나 **Serviceguard** 명령줄에서 패키지를 중지할 수 있습니다. 패키지를 중지하면 노드를 중지하는 것과는 다른 결과가 발생합니다. 노드를 중지하면 해당 패키지가 대체 노드로 전환할 수 있지만(패키지에 대한 전환 기능이 사용 가능한 경우) 패키지를 중지하면 다른 노드로 전환할 수 없게 되므로 다른 노드나 같은 노드에서 수동으로 다시 시작해야 합니다.

Serviceguard Manager를 사용하여 패키지 중지

중지할 패키지를 선택하고 마우스 오른쪽 단추를 클릭하여 동작 목록을 표시합니다. “패키지 <pkgname> 중지”를 선택합니다. 이때 패키지는 실행 중이어야 합니다.

작업이 수행됨에 따라 진행 창에 메시지가 표시됩니다. 여기에는 패키지의 중지 관련 메시지가 포함됩니다.

Serviceguard 명령을 사용하여 패키지 중지

다음과 같이 `cmhaltpkg` 명령을 사용하여 패키지를 중지할 수 있습니다.

```
# cmhaltpkg pkg1
```

이 작업은 `pkg1`을 중지하고 이 패키지를 다른 노드로 전환할 수 없도록 합니다.

패키지 이동

Serviceguard Manager나 **Serviceguard** 명령을 사용하여 한 노드에서 다른 노드로 패키지를 이동할 수 있습니다.

Serviceguard Manager를 사용하여 패키지 이동

작업을 시작하려면 패키지가 실행 중이어야 합니다. 맵이나 트리에서 패키지를 선택한 다음 마우스로 다른 클러스터 노드로 끕니다.

또는 중지할 패키지의 아이콘을 선택하고 마우스 오른쪽 단추를 클릭하여 동작 목록을 표시합니다. “노드로 패키지 이동”을 선택하거나, 원하는 패키지를 선택하고 도구 모음 메뉴로 이동하여 동작 -> 관리를 선택합니다.

작업이 수행됨에 따라 작업 로그 창에 메시지가 표시됩니다. 여기에는 패키지 중지 및 대상 노드에서의 패키지 시작에 대한 메시지가 포함됩니다.

패키지를 이동한 후에는 전환 플래그를 변경할 수 있습니다. 동작 메뉴를 사용하거나 팝업 메뉴를 마우스 오른쪽 단추로 클릭하여 이 작업을 수행할 수 있습니다.

Serviceguard 명령을 사용하여 실행 중인 패키지 이동

패키지를 이동하기 전에 `cmhaltpkg` 명령을 사용하여 원래 노드에서 패키지를 중지합니다. 이렇게 하면 패키지가 중지될 뿐 아니라 패키지를 중지한 노드로는 패키지가 다시 전환될 수 없게 됩니다.

패키지를 이동한 후에는 다시 시작하여 전환이 가능하도록 해야 합니다. 일부 오류로 인해 패키지를 특정 노드에서 실행할 수 없는 경우 `cmmodpkg`와 `-n` 옵션을 사용하여 패키지를 해당 노드에서 실행 가능하게 할 수 있습니다. 노드를 지정하지 않으면 명령이 실행된 노드가 지정된 것으로 간주됩니다.

예를 들면 다음과 같습니다.

```
# cmhaltpkg pkg1
# cmrunpkg -n ftsys10 pkg1
# cmmodpkg -e pkg1
```

패키지 전환 동작 변경

패키지에 적용되는 전환 플래그의 종류에는 다음과 같이 두 가지가 있습니다.

- 패키지 전환 또는 한 노드에서 다른 노드로 패키지를 이동할 수 있는 기능
- 노드 전환 또는 특정 노드로 이동할 수 있는 기능

패키지 전환의 초기 설정은 `AUTO_RUN` 매개 변수에 의해 결정됩니다. 이 매개 변수는 패키지 `ASCII` 구성 파일에 설정됩니다. `AUTO_RUN`을 `YES`로 설정하면 패키지가 클러스터에서 처음 시작될 때 패키지 전환이 활성화됩니다. 노드 전환의 초기 설정에서는 구성된 모든 노드로 전환하여 패키지를 실행하도록 허용합니다. 클러스터가 실행 중이면 노드 전환과 패키지 전환이 모두 동적으로 변경될 수 있습니다.

Serviceguard Manager를 사용하여 패키지 전환 변경

Serviceguard Manager에서 패키지 전환이나 노드 전환을 변경하려면 트리 또는 맵에서 변경할 패키지를 선택합니다. 패키지를 마우스 오른쪽 단추로 클릭하거나 도구 모음의 동작 메뉴로 이동합니다. 라디오 단추를 클릭하여 활성화 또는 비활성화 사이를 변경합니다. 노드별로 노드 전환을 설정할 수 있습니다.

현재 패키지가 노드에서 실행 중이고 해당 노드에 대해 패키지 전환을 비활성화한 경우 패키지는 계속 실행되지만 나중에 다시 해당 노드로 전환할 수 없게 됩니다.

AUTO-RUN 매개 변수를 변경하려면 패키지를 다시 구성해야 합니다. 이 작업은 Serviceguard 11.16 이상에서만 수행할 수 있습니다. 패키지를 선택한 다음 동작 메뉴(도구 모음이나 마우스 오른쪽 단추를 클릭할 때 나타나는 메뉴 사용)에서 구성을 선택합니다. 명령줄을 사용할 때는 구성을 만들거나 수정할 수 있도록 클러스터에 대한 루트 권한이 있어야 합니다.

Serviceguard 명령을 사용하여 패키지 전환 변경

Serviceguard 명령을 사용하여 패키지의 전환 동작을 일시 또는 영구적으로 변경할 수 있습니다. 실행 중인 패키지가 일시적으로 다른 노드로 전환되지 못하도록 하려면 `cmmodpkg` 명령을 사용합니다. 예를 들어, 현재 실행 중인 `pkg1`이 다른 노드에서 시작되지 않도록 하려면 다음 명령을 입력합니다.

```
# cmmodpkg -d pkg1
```

이 명령은 패키지를 중지시키지는 않지만 다른 노드에서 시작되지 않도록 합니다.

`cmmodpkg` 명령에 `-n` 옵션을 사용하여 특정 노드로의 패키지 전환을 비활성화할 수도 있습니다. 다음 명령은 `pkg1`이 `lptest3` 노드로 전환하지 못하도록 합니다.

```
# cmmodpkg -d -n lptest3 pkg1
```

다음에 클러스터가 재시작된 후에도 패키지 전환에 대한 변경 동작이 계속해서 유지될 수 있도록 패키지 전환을 영구적으로 비활성화하려면 패키지 구성 파일의 `AUTO_RUN` 플래그를 변경한 다음 구성을 다시 적용해야 합니다. 이렇게 하면 다음에 클러스터가 재시작된 후에도 변경 내용이 유지됩니다.

재구성에 대한 자세한 내용은 다음의 “실행 중인 클러스터에서 패키지 재구성” 절을 참조하십시오.

클러스터 재구성

클러스터가 중지되었거나 실행 중일 때 클러스터를 다시 구성할 수 있습니다. 일부 작업은 클러스터가 중지된 경우에만 수행할 수 있습니다. 표 7-1에서는 여러 종류의 변경 시 필요한 클러스터의 상태를 보여 줍니다.

표 7-1

영구적인 클러스터 구성 변경의 종류

| 클러스터 구성 변경 | 필요한 클러스터 상태 |
|-------------------------------------|--|
| 새 노드 추가 | 모든 클러스터 노드가 실행되고 있어야 합니다. 직렬 하트비트가 구성되어 있으면 안 됩니다. |
| 노드 삭제 | 사용할 수 없거나 연결할 수 없는 상태의 노드도 삭제할 수 있습니다. |
| 볼륨 그룹 추가 | 클러스터가 실행 중일 때나 중지된 경우 모두 가능합니다. |
| 볼륨 그룹 삭제 | 클러스터가 실행 중일 때나 중지된 경우 모두 가능합니다. 볼륨 그룹을 사용하는 패키지는 제어 스크립트를 수정할 때까지는 다시 시작되지 않습니다. |
| 구성된 최대 패키지 수 변경 | 클러스터가 실행되고 있으면 안 됩니다. |
| 타이밍 매개 변수 변경 | 클러스터가 실행되고 있으면 안 됩니다. |
| 쿼럼 서버 구성 변경 | 클러스터가 실행되고 있으면 안 됩니다. |
| 클러스터 잠금 구성 변경 | 클러스터가 실행되고 있으면 안 됩니다. |
| 직렬 장치 파일 변경 | 클러스터가 실행되고 있으면 안 됩니다. |
| 하트비트 또는 모니터링되는 서브넷의 IP 주소 변경 | 클러스터가 실행되고 있으면 안 됩니다. |
| 액세스 제어 정책 변경(Serviceguard 11.16 이상) | 클러스터 및 패키지가 실행 중일 때나 중지된 경우 모두 가능합니다. |

표 7-1

영구적인 클러스터 구성 변경의 종류 (계속)

| 클러스터 구성 변경 | 필요한 클러스터 상태 |
|--|-----------------------|
| Faster Failover 제품을 활성화하거나 비활성화하기 위한 장애 조치 최적화 | 클러스터가 실행되고 있으면 안 됩니다. |

중지된 클러스터 재구성

클러스터가 중지되었을 때 클러스터 구성을 영구적으로 변경할 수 있습니다. 이 절차는 퀵럼 서버 또는 잠금 디스크 구성을 변경하고, 타이밍 매개 변수를 변경하고, 최대 패키지 수 매개 변수를 변경할 때 **반드시** 수행해야 하지만 다른 클러스터 구성을 변경할 때도 사용할 수 있습니다.

다음 단계를 따르십시오.

1. Serviceguard의 클러스터 중지 명령을 사용하거나 명령줄에서 `cmhaltcl` 명령을 실행하여 모든 노드에서 클러스터를 중지합니다.
2. “HA 클러스터 구성 만들기” 장에서 설명한 것처럼 하나의 노드에서 클러스터를 재구성합니다. Serviceguard Manager를 사용하거나(Serviceguard 버전 11.16에서만), 명령줄에서 `cmquerycl` 명령을 실행하여 ASCII 파일을 생성한 다음 편집합니다.
3. 클러스터 구성 파일에 나열된 모든 노드에 전원이 공급되고 액세스할 수 있는지 확인합니다. Serviceguard의 적용 단추를 사용하거나 명령줄에서 `cmapplyconf` 명령을 실행하여 모든 노드에 이진 클러스터 구성 파일을 복사합니다. 이 파일은 이전 버전의 이진 클러스터 구성 파일을 덮어씁니다.
4. 모든 노드 또는 일부 노드에서 클러스터를 시작합니다. Serviceguard의 클러스터 실행 관리 명령을 사용하거나 명령줄에서 `cmrunc1` 명령을 실행하면 됩니다.

Serviceguard Manager를 사용하여 MAX_CONFIGURED_PACKAGES 변경

Serviceguard 11.16에서만 클러스터에 루트로 로그인해야 합니다. 먼저 클러스터를 중지한 다음 트리나 맵에서 클러스터를 선택합니다. 동작 메뉴에서 관리를 선택한 다음 클러스터 중지를 선택합니다.

클러스터가 중지되면 맵이나 트리에서 클러스터를 선택합니다. 동작 메뉴에서 구성을 선택합니다. 클러스터 구성 창이 열리면 매개 변수 탭을 클릭합니다. 새 번호를 입력하고 적용을 클릭한 다음 구성 창을 닫습니다. 새로 고침을 수행한 후에 클러스터 속성에서 변경 내용을 확인합니다.

Serviceguard 명령을 사용하여 MAX_CONFIGURED_PACKAGES 변경

클러스터를 중지해야 합니다.

cmgetconf 명령을 사용하여 클러스터의 기존 구성에 대한 현재 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
# cmgetconf -c clconfig.ascii
```

원하는 MAX_CONFIGURED_PACKAGES의 값이 포함되도록 clconfig.ascii 파일을 편집합니다. 그런 다음 cmcheckconf 명령을 사용하여 새 구성을 확인합니다. cmcheckconf와 -k 또는 -K 옵션을 사용하면 시간을 상당히 줄일 수 있습니다. cmcheckconf 명령과 -k 또는 -K 옵션을 사용하면 응답 시간을 상당히 줄일 수 있습니다.

cmapplyconf 명령을 사용하여 구성에 변경 내용을 적용하고 모든 클러스터 노드로 새 구성 파일을 전송합니다. cmapplyconf와 -k 또는 -K 옵션을 사용하면 시간을 상당히 줄일 수 있습니다.

실행 중인 클러스터 재구성

클러스터가 시작되어 실행 중일 때 클러스터 구성에 새 노드를 추가하거나 삭제할 수 있습니다. 그러나 다음 사항에 주의하십시오.

- 클러스터가 실행 중일 때는 쿼럼 서버 또는 잠금 디스크 구성을 변경할 수 없습니다.
- 클러스터에서 활성 노드를 제거할 수 없습니다. 먼저 노드를 중지해야 합니다.
- 클러스터 구성에서 활성 볼륨 그룹을 삭제할 수 없습니다. 삭제하기 전에 볼륨이 비활성화된 상태인지 확인한 다음 해당 볼륨 그룹을 사용하는 패키지를 중지해야 합니다.
- 클러스터 타이밍 매개 변수를 변경할 수 없습니다.
- 노드에 연결할 수 없을 때(예를 들어, 네트워크 연결이 완전히 끊어진 경우) 허용되는 유일한 구성 변경은 클러스터 구성에서 연결되지 않은 노드를 삭제하는 것입니다. 또한 해당 노드에 의존하는 패키지가 있으면 패키지 구성도 수정하여 해당 노드를 삭제해야 합니다. 이 작업은 모두 하나의 구성 요청(cmapplyconf 명령)으로 수행되어야 합니다.

패키지 구성 변경에 대한 자세한 내용은 뒤에서 설명합니다.

다음 절에서는 **Serviceguard Manager**나 **Serviceguard** 명령을 사용하여 동적 재구성 작업을 수행하는 방법에 대해 설명합니다.

클러스터 실행 중에 Serviceguard Manager를 사용하여 구성에 노드 추가

트리나 맵에서 클러스터를 선택합니다. 동작 메뉴에서 구성을 선택합니다. 클러스터에 대한 루트 권한이 있어야 합니다. 노드 탭의 사용 가능한 노드에서 추가할 노드를 강조 표시한 다음 추가를 클릭하고 적용을 클릭합니다. 새로 고침을 수행한 후에 클러스터 속성에서 변경 내용을 확인합니다.

클러스터 실행 중에 Serviceguard 명령을 사용하여 구성에 노드 추가

다음 단계에 따라 **HP-UX** 명령을 사용하여 노드를 추가할 수 있습니다. 이 예제에서는 노드 *ftsys8* 및 *ftsys9*가 *cluster1*이라는 실행 중인 클러스터에 이미 구성되어 있으며 이 클러스터에 노드 *ftsys10*을 추가하려고 합니다.

1. 다음 명령을 사용하여 임시 파일에 기존 클러스터 구성의 현재 사본을 저장할 수 있습니다.

```
# cmgetconf -c temp.ascii
```

2. 구성할 새 노드 집합을 지정하고 새 구성의 템플릿을 만듭니다. 전체 도메인 이름을 제외하고 노드 이름만 지정(31 바이트 이하)해야 합니다. 예를 들어, *ftsys8.cup.hp.com*이 아니라 *ftsys8*로 지정합니다.

```
# cmquerycl -C clconfig.ascii -c cluster1 \  
-n ftsys8 -n ftsys9 -n ftsys10
```

3. *clconfig.ascii* 파일을 편집하여 새 노드에 대한 정보를 확인합니다.
4. 다음과 같이 새 구성을 확인합니다.

```
# cmcheckconf -C clconfig.ascii
```

5. 구성에 변경 내용을 적용하고 모든 클러스터 노드에 새 이진 구성 파일을 전송합니다.

```
# cmapplyconf -C clconfig.ascii
```

cmrunnode를 사용하여 새 노드를 시작하고, 원하는 경우 /etc/rc.config.d/cmcluster 파일에서 AUTOSTART_CMCLD 매개 변수를 1로 설정하여 다시 부팅할 때마다 자동으로 새 노드가 클러스터에 참가하도록 합니다.

주

CVM 디스크 그룹을 사용하는 실행 중인 클러스터에 노드를 추가하면 해당 노드가 클러스터에 참가할 때 디스크 그룹을 가져올 수 있게 됩니다.

클러스터 실행 중에 Serviceguard Manager를 사용하여 구성에서 노드 삭제

노드는 중지 상태여야 합니다. 그렇지 않으면 노드를 선택하고 동작 메뉴에서 관리를 선택한 다음 노드 삭제를 선택합니다.

트리나 맵에서 클러스터를 선택합니다. 동작 메뉴에서 구성을 선택합니다. 클러스터에 대한 루트 권한이 있어야 합니다. 노드 탭의 사용 가능한 노드에서 제거할 노드를 강조 표시한 다음 삭제를 클릭합니다. 그런 다음 적용을 클릭합니다. 새로 고침을 수행한 후에 클러스터 속성에서 변경 내용을 확인합니다.

삭제할 노드에 연결할 수 없는 경우(예를 들어, LAN에서 연결이 끊어진 경우) 이 연결할 수 없는 노드를 지정하는 패키지가 없을 때만 노드를 삭제할 수 있습니다. 연결할 수 없는 노드에 의존하는 패키지가 있는 경우 다음 절에서 설명하는 것처럼 클러스터를 중지하거나 Serviceguard 명령을 사용하십시오.

클러스터 실행 중에 Serviceguard 명령을 사용하여 구성에서 노드 삭제

다음 단계에 따라 HP-UX 명령을 사용하여 노드를 삭제할 수 있습니다. 이 예제에서는 노드 *ftsys8*, *ftsys9* 및 *ftsys10*이 *cluster1*이라는 실행 중인 클러스터에 이미 구성되어 있으며 노드 *ftsys10*을 삭제하려고 합니다.

1. 다음 명령을 사용하여 임시 파일에 기존 클러스터 구성의 현재 사본을 저장할 수 있습니다.

```
# cmgetconf -c temp.ascii
```

2. 구성할 새 노드 집합을 지정하고(*ftsys10*은 생략) 새 구성의 템플릿을 만듭니다.

```
# cmquerycl -C clconfig.ascii -c cluster1 -n ftsys8 -n ftsys9
```

3. `clconfig.ascii` 파일을 편집하여 클러스터에 남아 있는 노드에 대한 정보를 확인합니다.

4. 다음과 같이 새 구성을 확인합니다.

```
# cmcheckconf -C clconfig.ascii
```

5. 구성에 변경 내용을 적용하고 모든 클러스터 노드에 새 이진 구성 파일을 전송합니다.

```
# cmapplyconf -C clconfig.ascii
```

주

연결할 수 없으면서 많은 패키지가 의존하고 있는 노드를 제거할 경우, 특히 종속 패키지에서 많은 수의 **EMS** 리소스를 사용하고 있으면 다음과 같은 메시지가 표시될 수 있습니다.

```
The configuration change is too large to process while the cluster is running.
```

```
Split the configuration change into multiple requests or halt the cluster.
```

이 경우, 클러스터를 중지하여 노드를 제거해야 합니다.

클러스터 실행 중에 Serviceguard Manager를 사용하여 LVM 구성 변경

트리나 탭에서 클러스터를 선택합니다. 동작 메뉴에서 구성을 선택합니다. 클러스터에 대한 루트 권한이 있어야 합니다. 논리 볼륨 탭에서 추가하거나 제거할 노드를 강조 표시한 다음 추가 또는 삭제를 클릭합니다. 그런 다음 적용을 클릭합니다. 새로 고침을 수행한 후에 클러스터 속성에서 변경 내용을 확인합니다.

클러스터가 실행되고 있을 때는 클러스터 잠금 볼륨 그룹 또는 물리 볼륨 구성을 변경할 수 없습니다.

주

클러스터 구성에서 볼륨 그룹을 제거하려면 이 볼륨 그룹을 활성화 또는 비활성화하는 패키지 제어 스크립트도 수정하거나 삭제해야 합니다. 또한 볼륨 그룹을 더 이상 사용하지 않을 각 노드의 제거된 볼륨 그룹에 대해 **LVM** `vgexport` 명령을 사용해야 합니다.

클러스터 실행 중에 Serviceguard 명령을 사용하여 LVM 구성 변경

cmgetconf 명령을 사용하여 클러스터의 기존 구성에 대한 현재 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
# cmgetconf -c clconfig.ascii
```

clconfig.ascii 파일을 편집하여 볼륨 그룹을 추가하거나 삭제합니다. 그런 다음 cmcheckconf 명령을 사용하여 새 구성을 확인합니다. cmapplyconf 명령을 사용하여 변경 내용을 구성에 적용하고 모든 클러스터 노드로 새 구성 파일을 전송합니다.

다음 예제에서는 두 개의 볼륨 그룹을 추가합니다.

```
# cmcheckconf -C clconfig.ascii vgname1 vgname2
```

여러 개의 볼륨 그룹이 있는 경우 모든 볼륨 그룹의 목록이 있는 참조 파일을 사용할 수 있습니다. 다음은 참조 파일을 사용하는 예입니다.

```
# cmcheckconf -C clconfig.ascii vgreferencefile
```

다음 예제에서는 앞의 예제에서 추가한 볼륨 그룹을 제거합니다.

```
# cmcheckconf -C clconfig.ascii vgname1 -R vgname2
```

```
# cmcheckconf -C clconfig.ascii vgreferencefile
```

cmapplyconf 명령을 사용하여 변경 사항을 구성에 적용하고 모든 클러스터 노드에 새로운 구성 파일을 전송하십시오. cmapplyconf 및 해당 옵션 구문은 위의 cmcheckconf 명령과 같습니다.

주

패키지에 의해 현재 활성화된 볼륨 그룹을 클러스터에서 삭제하는 경우 구성은 변경되지만 패키지를 중지할 때까지는 삭제가 적용되지 않습니다. 따라서 패키지 제어 스크립트에서 볼륨 그룹을 제거하는 것과 같은 추가 수정 작업이 없으면 더 이상 패키지를 실행할 수 없게 됩니다.

VxVM 또는 CVM 저장 장치 구성 변경

클러스터가 실행되고 있을 때 클러스터 구성에 VxVM 디스크 그룹을 추가할 수 있습니다. 새 CVM 디스크 그룹을 추가하려면 클러스터가 반드시 실행 중이어야 합니다.

이 노드에서 **CVM** 디스크 그룹을 만듭니다. **CVM** 저장 장치를 사용하는 패키지의 구성 **ASCII** 파일을 열고 **CVM** 저장 장치 그룹을 **STORAGE_GROUP** 문에 추가합니다. 그런 다음 `cmapplyconf` 명령을 실행합니다.

이와 비슷하게 해당 시점에서 클러스터 노드에서 사용하지 않고 있는 **VxVM** 또는 **CVM** 디스크 그룹을 삭제할 수 있습니다.

주

클러스터 구성에서 디스크 그룹을 제거하려면 이 디스크 그룹을 가져오거나 내보내는 패키지 제어 스크립트도 수정하거나 삭제해야 합니다. **CVM** 디스크 그룹을 제거하려면 패키지 **ASCII** 파일에서 디스크 그룹에 대한 **STORAGE_GROUP** 항목을 제거해야 합니다.

패키지 재구성

패키지 재구성 과정은 6장에서 설명한 기본 구성 과정과 비슷합니다. 구성 과정에 대한 자세한 내용은 6장을 참조하십시오.

패키지를 재구성하는 동안 클러스터를 중지하거나 실행할 수 있습니다. 가능한 변경 종류와 변경 내용이 적용되는 시점은 패키지가 실행 중인지 여부에 따라 달라집니다.

중지된 클러스터에서 패키지 재구성

클러스터가 실행되고 있지 않을 때도 패키지 구성을 영구적으로 변경할 수 있습니다. 다음 단계를 따르십시오.

- 이 장의 앞부분에서 설명한 방법대로 한 노드에서 패키지를 재구성합니다. **Serviceguard Manager**를 사용하거나 패키지 ASCII 파일을 편집하여 이 작업을 수행할 수 있습니다.
- **Serviceguard**의 유도 모드에 있지 않은 경우에는 패키지 제어 스크립트를 직접 편집합니다. 유도 모드에서는 재구성 시 **Serviceguard Manager**가 자동으로 패키지 제어 스크립트를 편집합니다. 서비스 이름을 변경한 경우에는 패키지 구성 파일에서 서비스 이름을 변경해야 합니다.
- **Serviceguard Manager**의 적용 단추를 클릭하거나 명령줄에 `cmapplyconf`를 입력합니다. 이렇게 하면 이전 클러스터 구성 파일이 모든 노드에 복사됩니다. 명령줄에서 패키지가 변경되도록 지정하는 **-P** 옵션을 사용합니다. **-C** 옵션은 사용하지 않습니다. 이 파일은 이전 버전의 이전 클러스터 구성 파일을 덮어씁니다.
- 해당 패키지를 실행될 수 있는 모든 노드에 수정된 제어 스크립트를 복사합니다. 적용 단추를 클릭하면 **Serviceguard Manager**에서 자동으로 수행됩니다.
- 원하는 경우 **Serviceguard Manager**의 클러스터 실행 명령을 사용하거나 명령줄에 `cmruncl`을 입력하여 모든 노드 또는 일부 노드에서 클러스터를 시작합니다. 패키지는 노드가 온라인 상태가 되면 시작됩니다.

실행 중인 클러스터에서 패키지 재구성

클러스터가 실행되고 있을 때 패키지를 재구성할 수 있으며, 일부 경우에는 패키지가 실행되고 있을 때 패키지를 재구성할 수도 있습니다. 패키지가 실행되고 있을 때는 특정 사항만 변경할 수 있습니다.

Serviceguard Manager에서 패키지를 수정하려면 수정할 패키지를 선택하고 동작 메뉴에서 구성을 선택합니다. 구성 창이 열리면 6장에서 설명한 것과 같은 옵션을 선택합니다. 또는 HP-UX 명령으로 다음 절차를 수행합니다(*pkg1*을 예로 사용).

1. 필요한 경우 패키지를 중지합니다.

```
# cmhaltpkg pkg1
```

표 7-2를 참조하여 이 단계가 필요한지 판별하십시오.

2. 아직 준비하지 않은 경우, 패키지 이름을 지정하는 `cmgetconf` 명령을 사용하여 패키지 ASCII 구성 파일의 복사본을 얻을 수 있습니다.

```
# cmgetconf -p pkg1 pkg1.ascii
```

3. ASCII 패키지 구성 파일을 편집합니다.

4. 다음과 같은 방법으로 변경 내용을 확인합니다.

```
# cmcheckconf -v -P pkg1.ascii
```

5. 변경된 구성을 모든 노드에 배포합니다.

```
# cmapplyconf -v -P pkg1.ascii
```

6. 패키지를 실행할 수 있는 모든 노드에 패키지 제어 스크립트를 복사합니다.

실행 중인 클러스터에 패키지 추가

클러스터가 실행 중이고 다른 패키지가 실행되고 있을 때 새 패키지를 만들어 클러스터 구성에 추가할 수 있습니다. 추가할 수 있는 패키지의 수는 클러스터 구성 파일의 *Maximum Configured Packages* 값에 제한을 받습니다.

패키지를 만들려면 “패키지 및 서비스 구성” 장에 설명된 단계를 따르십시오. 그러나 Serviceguard 명령줄을 사용하는 경우에는 HP-UX 명령을 사용하여 구성을 확인하고 배포할 때 클러스터 ASCII 파일을 지정하지 **마십시오**. 예를 들어, 실행 중인 클러스터에서 HP-UX 명령을 사용하여 새로 만든 *pkg1*을 확인하는 경우가 해당됩니다.

```
# cmcheckconf -P /etc/cmcluster/pkg1/pkg1conf.ascii
```

다음과 같은 HP-UX 명령을 사용하여 새 패키지 구성을 클러스터의 모든 노드에 배포합니다.

```
# cmapplyconf -P /etc/cmcluster/pkg1/pkg1conf.ascii
```

패키지를 실행할 수 있는 모든 노드의 /etc/cmcluster/pkg1 디렉토리에 제어 스크립트를 복사해야 합니다.

실행 중인 클러스터에서 패키지 삭제

Serviceguard Manager에서 먼저 패키지를 중지하고 클러스터를 선택합니다. 동작 메뉴에서 구성 -> 패키지 삭제를 선택합니다.

Serviceguard 명령줄에서 `cmdeleteconf` 명령을 사용하여 모든 클러스터 노드에서 패키지를 삭제할 수 있습니다. 이 명령은 패키지가 실행되고 있지 않을 때만(클러스터는 실행 중이어도 됨) 실행할 수 있습니다. 이 명령은 클러스터에 있는 모든 노드의 이진 구성 파일에서 해당 패키지 정보를 제거합니다.

다음 예제에서는 `mypkg` 패키지를 중지하고 클러스터에서 해당 패키지 구성을 제거합니다.

```
# cmhaltpkg mypkg  
# cmdeleteconf -p mypkg
```

-f 옵션을 사용하지 않으면 파일을 삭제하기 전에 이를 확인하는 메시지가 표시됩니다. /etc/cmcluster/mypkg 디렉토리는 이 명령으로 삭제되지 않습니다.

서비스 재시작 카운터 재설정

서비스 재시작 카운터는 패키지 서비스가 자동으로 재시작된 횟수를 나타냅니다.

이 값은 패키지 서비스가 허용되는 최대 자동 재시작 횟수를 초과했는지를 판단할 때 사용됩니다.

주

특정 서비스의 허용되는 최대 재시작 횟수는 패키지 제어 스크립트의 `SERVICE_RESTART[]` 매개 변수로 설정됩니다. 이 매개 변수는 별도로 패키지 관리자가 별도로 관리하는 재시작 카운터와는 다릅니다.

몇 번의 시도 후에 패키지 서비스가 성공적으로 재시작되면 패키지 관리자는 이 재시작 횟수를 자동으로 재설정하지 않습니다. 그러나 `cmmodpkg -R -s` 명령을 사용하여 온라인 상태에서 이 카운터를 재설정할 수 있으며, 이 경우 나중에 서비스가 재시작되는 상황이 발생하면 구성된 `SERVICE_RESTART` 횟수만큼 재시작 시도가 이루어지게 할 수 있습니다. 예를 들면 다음과 같습니다.

```
# cmmodpkg -R -s myservice pkg1
```

재시작 카운터의 현재 값은 `cmviewcl -v` 명령의 출력에서 볼 수 있습니다.

재구성 중 허용되는 패키지 상태

구성을 변경할 때 클러스터의 모든 노드는 전원이 공급되고 액세스 가능한 상태여야 합니다.

특정 종류의 변경 작업 중 패키지가 실행될 수 있는지 여부는 표 7-2를 참조하십시오. 다음과 같은 모든 경우에는 클러스터가 실행 중이고 재구성되는 패키지를 제외한 나머지 패키지가 실행 중이라고 가정합니다.

표 7-2

패키지의 변경 종류

| 패키지의 변경 | 필요한 패키지 상태 |
|----------|--|
| 새 패키지 추가 | 다른 패키지는 어떤 상태여도 상관없습니다. |
| 패키지 삭제 | 패키지가 실행되고 있으면 안 됩니다. |
| 서비스 추가 | 패키지가 실행되고 있으면 안 됩니다. |
| 서비스 제거 | 패키지가 실행되고 있으면 안 됩니다. |
| 서브넷 추가 | 패키지가 실행되고 있으면 안 됩니다. 서브넷은 클러스터에 이미 구성되어 있어야 합니다. |

표 7-2

패키지의 변경 종류(계속)

| 패키지의 변경 | 필요한 패키지 상태 |
|---------------|--|
| 서브넷 제거 | 패키지가 실행되고 있으면 안 됩니다. |
| 리소스 추가 | 패키지가 실행되고 있으면 안 됩니다. |
| 리소스 제거 | 패키지가 실행되고 있으면 안 됩니다. |
| 볼륨 그룹 추가 | 클러스터가 실행되고 있을 때 볼륨 그룹을 클러스터에 구성할 수 있습니다. 변경은 제어 스크립트에서 이루어지므로 패키지는 어떤 상태여도 상관없습니다. 그러나 변경 내용을 적용하려면 패키지를 중지한 다음 다시 시작해야 합니다. |
| 볼륨 그룹 제거 | 패키지가 실행되고 있으면 안 됩니다. |
| 실행 스크립트 내용 변경 | 패키지가 중지되었을 때 수행하는 것이 좋습니다. 패키지가 실행 중일 때 패키지의 실행 스크립트를 변경하면 타이밍상의 문제가 발생할 수 있습니다. |
| 중지 스크립트 내용 변경 | 패키지가 중지되었을 때 수행하는 것이 좋습니다. 패키지가 실행되고 있을 때 패키지의 중지 스크립트를 변경하면 타이밍상의 문제가 발생할 수 있습니다. |
| 스크립트 시간 종료 | 패키지가 실행 중일 때나 중지된 경우 모두 가능합니다. |
| 서비스 시간 종료 | 패키지가 실행되고 있으면 안 됩니다. |
| 서비스 빠른 장애 조치 | 패키지가 실행되고 있으면 안 됩니다. |
| 패키지 자동 실행 | 패키지가 실행 중일 때나 중지된 경우 모두 가능합니다. |
| 로컬 LAN 장애 조치 | 패키지가 실행 중일 때나 중지된 경우 모두 가능합니다. |

표 7-2

패키지의 변경 종류(계속)

| 패키지의 변경 | 필요한 패키지 상태 |
|-----------------------|--------------------------------|
| 패키지 실행 가능 노드 순서 변경 | 패키지가 실행 중일 때나 중지된 경우 모두 가능합니다. |
| 패키지 장애 조치 정책 변경 | 패키지가 실행 중일 때나 중지된 경우 모두 가능합니다. |
| 패키지 장애 복구 정책 변경 | 패키지가 실행 중일 때나 중지된 경우 모두 가능합니다. |
| 액세스 정책 변경 | 패키지가 실행 중일 때나 중지된 경우 모두 가능합니다. |

클러스터 이벤트에 대응

Serviceguard에서는 시스템 관리 작업에 사용자가 많이 개입할 필요가 없습니다. 장애가 발생하지 않는 한 클러스터는 계속 모니터링되고 보호됩니다. 장애가 발생하는 경우에는 다른 노드로 전달되도록 지정된 패키지가 자동으로 전달됩니다. 시스템 관리자가 수행해야 하는 작업은 클러스터를 모니터링하고 패키지가 전달되었는지를 확인하는 것입니다. 패키지 전달이 발생한 경우에는 그 원인을 확인하고 올바른 조치를 취해야 합니다.

Event Monitoring Service와 HA 모니터는 디스크, LAN 카드 및 일부 시스템 이벤트에 대한 모니터링 기능을 제공할 수 있습니다. 자세한 내용은 *Using HA Monitors* 설명서를 참조하십시오.

일반적으로 패키지 전달이 발생할 때 취해야 할 일반적인 조치는 다음과 같습니다.

- 전달이 발생한 시기 확인
- 전달 원인 확인
- 하드웨어 장애 수리
- 소프트웨어 문제점 해결
- 노드 재시작
- 패키지를 원래의 노드로 다시 전달

시스템에서 Serviceguard 제거

사용하는 Serviceguard에서 노드를 제거하려면 `swremove` 명령을 사용하여 소프트웨어를 삭제합니다. 다음 사항을 주의하십시오.

- Serviceguard를 삭제할 노드에서 클러스터가 실행되고 있지 않아야 합니다.
- Serviceguard를 삭제할 노드가 클러스터 구성에 포함되어 있으면 안 됩니다.
- 두 개 이상의 노드에서 Serviceguard를 제거하는 경우 `swremove`는 한 번에 하나의 노드에서만 실행해야 합니다.

클러스터 및 패키지 유지 관리
시스템에서 Serviceguard 제거

8 클러스터 문제 해결

이 장에서는 클러스터 작동을 확인하고, 클러스터 상태를 확인하고, 하드웨어를 추가 또는 교체하고, 몇 가지 일반적인 클러스터 문제를 해결하는 방법에 대해 설명합니다. 이 장에서 다루는 내용은 다음과 같습니다.

- 클러스터 작동 테스트
- 하드웨어 모니터링
- 디스크 교체
- 입출력 카드 교체
- LAN 카드 교체
- 잘못된 쿼럼 서버 시스템 교체
- 문제 해결 방법
- 문제 해결

클러스터 작동 테스트

Serviceguard 클러스터를 구성한 후에 장애가 발생하면 클러스터의 다양한 구성 요소가 제대로 동작하는지 확인해야 합니다. 이 절에서는 다음 과정에 따라 패키지, 노드 또는 LAN에 장애가 발생할 경우 클러스터가 올바르게 응답하는지 테스트합니다.

주의

다음 과정의 클러스터 테스트에서는 클러스터의 여러 구성 요소에 장애를 발생시켜 클러스터가 장애 상황에 올바르게 대처하는지 확인하게 됩니다. 따라서 노드 및 응용 프로그램을 사용하지 못할 수도 있습니다.

Serviceguard Manager를 사용하여 클러스터 시작

클러스터 구성을 마쳤으면 클러스터가 자동으로 시작됩니다. 클러스터가 나중에 중지된 경우에는 다시 시작하십시오. 트리 또는 맵에서 클러스터를 선택한 다음 동작 메뉴에서 관리 -> 클러스터 실행을 선택하면 됩니다.

패키지 관리자 테스트

패키지 관리자가 올바르게 작동 중인지 테스트하려면 클러스터의 각 패키지에 대해 다음 과정을 수행하십시오.

1. 다음과 같이 입력하여 패키지에서 실행 중인 서비스의 PID 번호를 확인합니다.

```
# ps -ef | grep <service_cmd>
```

여기서 *service_cmd*는 패키지 제어 스크립트에서 SERVICE_CMD 매개 변수를 사용하여 지정한 실행 파일입니다. 선택한 서비스에는 SERVICE_RESTART가 지정되어 있으면 안 됩니다.

2. *service_cmd* PID를 강제 종료하려면 다음을 입력합니다.

```
# kill PID
```

3. 패키지 상태를 보려면 다음을 입력합니다.

```
# cmviewcl -v
```

패키지는 지정된 대체 노드에서 실행되고 있어야 합니다.

4. **Serviceguard**를 사용하여 패키지를 기본 노드로 다시 이동합니다. 패키지를 선택한 다음 동작 메뉴에서 관리 -> 패키지 이동을 선택하면 됩니다.

클러스터 관리자 테스트

클러스터 관리자가 올바르게 작동 중인지 테스트하려면 클러스터의 각 노드에 대해 다음 과정을 수행하십시오.

1. 노드 SPU의 전원을 끕니다.
2. 클러스터 재편성 상태를 확인하려면 구성된 다른 노드에서 다음 명령을 입력합니다.

```
# cmviewcl -v
```

전원을 끈 노드가 중지되고 해당 패키지가 다른 노드로 올바르게 전환되었는지 확인할 수 있어야 합니다.

3. 노드 SPU의 전원을 켭니다.
4. 노드가 클러스터에 다시 참가하고 있는지 확인하려면 구성된 노드에 대해 다음 명령을 입력합니다.

```
# cmviewcl -v
```

클러스터에서 해당 노드를 인식해야 하지만 패키지는 실행되고 있으면 **안 됩니다**.

5. **Serviceguard Manager**를 사용하여 패키지를 원래 노드로 다시 이동합니다.
맵 또는 트리에서 패키지를 선택한 다음 동작 메뉴에서 관리 -> 패키지 이동을 선택합니다.
6. 클러스터의 모든 노드에 대해 한 번에 한 노드씩 이 절차를 반복합니다.

네트워크 관리자 테스트

네트워크 관리자가 올바르게 작동 중인지 테스트하려면 클러스터의 각 노드에 대해 다음을 수행하십시오.

1. 노드의 주 LAN 카드와 대기 LAN 카드를 확인하려면 다음을 입력합니다.

```
# lanscan
```

그런 다음 다음을 입력합니다.

```
# cmviewcl -v
```

2. 주 카드에서 LAN 연결을 해제합니다. ThinLAN 케이블을 사용하는 경우 서브넷이 중단되지 않도록 주의하십시오.
3. 대기 카드가 주 카드가 되도록 로컬 전환이 발생했는지 확인합니다. Serviceguard Manager에서 클러스터 속성을 확인합니다. 또는 명령줄에서 `cmviewcl -v` 명령을 사용합니다.
4. LAN을 원래의 주 카드에 다시 연결한 다음 상태를 확인합니다. Serviceguard Manager에서 클러스터 속성을 확인합니다. 또는 명령줄에서 `cmviewcl -v` 명령을 사용합니다.

하드웨어 모니터링

일반적으로 고가용성 시스템을 처리하는 데 적절한 방법에는 장애를 가능한 한 방지하고 그렇지 못할 경우에는 적어도 장애가 발생할 때 민첩하게 대처할 수 있도록 하는 주의 깊은 장애 모니터링 과정이 포함됩니다. 모든 종류의 오류나 경고에 대해 다음 사항을 모니터링해야 합니다.

- 디스크
- CPU
- 메모리
- LAN 카드
- 전원 공급 장치
- 모든 케이블
- 디스크 인터페이스 카드

일부 모니터링 작업은 단순한 실제 검사를 통해 수행할 수 있지만 최대한 포괄적인 모니터링을 위해서는 주기적으로 시스템 로그 파일(/var/adm/syslog/syslog.log)을 검사하여 구성된 모든 HA 장치에 대한 보고서를 생성해야 합니다. 장치와 관련된 오류가 발생하면 유지 관리가 필요하다는 것을 나타냅니다.

Event Monitoring Service 사용

Event Monitoring Service(EMS)를 사용하면 특정 장치 및 시스템 리소스의 모니터를 구성할 수 있습니다. 문제가 발생할 경우 운영자에게 추가 조치를 알릴 수 있는 관리 워크스테이션으로 경고를 보낼 수 있습니다. 예를 들어, 클러스터에서 사용하는 미러링 되는 볼륨 그룹에서 미러가 손상되면 디스크 모니터에서 이를 보고하도록 구성할 수 있습니다.

자세한 내용은 *Using HA Monitors* 설명서를 참조하십시오.

EMS 하드웨어 모니터 사용

메모리, CPU를 비롯한 많은 시스템 값을 모니터링하고 보고하기 위해 일련의 하드웨어 모니터를 사용할 수 있습니다. 일부 모니터는 특정 하드웨어 제품과 함께 제공됩니다.

자세한 내용은 *EMS Hardware Monitors User's Guide(B6191-90020)*를 참조하십시오.

하드웨어 모니터 및 영속성 요청

monconfig 도구를 사용하여 하드웨어 모니터를 비활성화하면 영속성 파일에서 관련된 하드웨어 모니터 영속성 요청이 제거됩니다. 하드웨어 모니터링을 다시 활성화하면 monconfig 도구를 사용하여 초기화된 모니터 요청이 다시 만들어집니다.

그러나 Serviceguard Manager를 사용하여 만들거나 Serviceguard가 시작될 때 설정된 하드웨어 모니터 요청은 다시 만들어지지 않습니다. 이 요청은 psmmmon 하드웨어 모니터와 관련된 요청입니다.

영속성 모니터 요청을 다시 만들려면 노드에서 Serviceguard를 중지한 다음 다시 시작합니다. 그러면 영속성 모니터 요청이 다시 만들어집니다.

HP Predictive 모니터링 사용

실제 장치 장애를 보고하는 메시지 외에도 시간 경과에 따라 곧 장애가 발생할 수도 있음을 나타내는 보다 덜 심각한 메시지가 로그에 쌓여 있을 수도 있습니다. HP Predictive는 모니터링 작업에 어느 정도의 자동화 기능을 제공하는 제품으로서, 모니터링되는 시스템의 상태 대기열에서 정보를 수집하여 어떤 오류가 쌓여 있는지 확인합니다. 이 도구는 장애를 보고하고 시간 경과에 따라 덜 치명적인 특정 오류가 발행하는 장치에 대한 통계를 기반으로 장애를 예측하기도 합니다. Serviceguard 클러스터의 경우 모든 노드에서 HP Predictive를 실행해야 합니다.

또한 HP Predictive는 오류 조건을 직접 HP 레스판스 센터에 보고하고 고객 지원 담당자에게 잠재적인 문제점에 대해 경고합니다. HP Predictive는 다양한 지원 계약을 통해 사용할 수 있습니다. 자세한 내용은 HP 담당자에게 문의하십시오.

디스크 교체

고장난 디스크 메커니즘을 교체하는 절차는 사용 중인 디스크 구성 유형에 따라 다릅니다. 고가용성 엔클로저에서 어레이 메커니즘 및 디스크를 교체하는 방법에 대해서는 별도로 설명합니다.

고장난 어레이 메커니즘 교체

RAID 1 또는 RAID 5에 HA 디스크 어레이를 구성한 경우 고장난 장치를 교체하는 방법에 대해서는 해당 어레이의 설명서를 참조하십시오. 교체한 후에는 해당 장치에서 자동으로 손실된 데이터를 새 디스크에 다시 만듭니다. LVM 작업은 필요하지 않습니다. 이 프로세스를 디스크 핫 스와핑이라고 합니다.

HA 엔클로저의 고장난 메커니즘 교체

MirrorDisk/UX와 함께 소프트웨어 미러링을 사용하고 있으며 미러링되는 디스크가 고가용성 디스크 엔클로저에 마운트된 경우에는 다음 단계에 따라 디스크 장치를 핫 플러그할 수 있습니다.

1. 장애가 발생한 디스크의 물리 볼륨 이름과 해당 볼륨이 구성된 볼륨 그룹의 이름을 지정합니다. 다음 예제에서 볼륨 그룹 이름은 `/dev/vg_sg01`이고 물리 볼륨 이름은 `/dev/dsk/c2t3d0`입니다. 이 이름을 사용 중인 시스템에 해당되는 볼륨 그룹 및 물리 볼륨 이름으로 대체하십시오.
2. 장애가 발생한 물리 볼륨에 정의된 확장 영역이 있는 논리 볼륨의 이름을 지정합니다.
3. 해당 볼륨 그룹이 현재 활성 상태인 노드에서 **장애가 발생한 물리 볼륨에 확장 영역이 있는 각 논리 볼륨에 대해** 다음 명령을 실행합니다.

```
# lvreduce -m 0 /dev/vg_sg01/lvolname /dev/dsk/c2t3d0
```
4. 이때 장애가 발생한 디스크를 제거하고 새 디스크를 넣습니다. 새 디스크는 이전 디스크와 같은 HP-UX 장치 이름을 갖게 됩니다.
5. `lvreduce` 명령을 실행한 노드에서 다음 명령을 실행하여 새로 삽입한 디스크에 볼륨 그룹 구성 데이터를 복원합니다.

```
# vgcfgrestore -n /dev/vg_sg01 /dev/dsk/c2t3d0
```

6. 다음 명령을 실행하여 새로 삽입한 디스크로 논리 볼륨을 확장합니다.

```
# lvextend -m 1 /dev/vg_sg01 /dev/dsk/c2t3d0
```

7. 마지막으로 장애가 발생한 물리 볼륨에 확장 영역이 있는 각 논리 볼륨에 대해

lvsync 명령을 실행합니다. 이 명령은 다른 미러의 확장 영역과 새 디스크의 확장 영역을 동기화합니다.

```
# lvsync /dev/vg_sg01/lvolname
```

잠금 디스크 교체

장애가 있는 잠금 디스크 메커니즘을 교체하는 방법은 데이터 디스크를 교체하는 방법과 같습니다. 사용자 데이터가 없는 전용 잠금 디스크를 사용하는 경우 다음 예제와 같이 하나의 LVM 명령만 실행해야 합니다.

```
# vgcfgrestore -n /dev/vg_lock /dev/dsk/c2t3d0
```

Serviceguard에서는 매시간 잠금 디스크를 검사합니다. vgcfgrestore 명령을 실행한 다음 최대 한 시간 동안의 활성 클러스터에 대한 syslog 파일을 검사합니다. 그런 다음 잠금 디스크가 다시 정상으로 돌아왔다는 메시지를 찾습니다.

인라인 SCSI 접속 단자를 사용하여 온라인 하드웨어 유지 관리

일부 공유 SCSI 버스 구성에서는 HP ILT(인라인 단자) 케이블을 사용하는 경우 온라인 SCSI 디스크 컨트롤러 하드웨어의 문제를 해결할 수 있습니다. 인라인 단자 케이블은 대부분의 SCSI-2 Fast-Wide 구성에서 지원됩니다.

Ultra2 SCSI 호스트 버스 어댑터에서는 SC10 디스크 엔클로저와 함께 사용하는 경우에만 인라인 단자 케이블이 지원됩니다. 이것은 SC10이 보다 느린 SCSI 버스 속도로 작동하므로 ILT 케이블을 사용하는 데 안전하기 때문입니다. ILT 케이블이 사용되는 경우 SCSI 버스 속도가 빠르면 신호 없이 데이터가 손상될 수 있으므로 Ultra160 또는 Ultra3 SCSI 구성에서는 인라인 단자 케이블을 사용할 수 없습니다.

인라인 단자 케이블은 많은 Form Factor에서 사용할 수 있습니다. 각각의 경우 한쪽 끝에 있는 커넥터에는 SCSI 접속 단자가 포함되며 이는 노드의 SCSI 호스트 버스 어댑터(HBA) 대신 SCSI 버스의 끝에서 연결을 종결하는 데 사용됩니다. ILT 케이블의 단자가 있는 쪽 끝이 HBA에 연결되어 있으면 해당 HBA에서는 접속 단자가 비활성화되어야 합니다. HBA에서 접속 단자를 비활성화하려면 사용되는 HBA에 따라 접속 단자 지

항기 팩을 제거하고 HBA에 적절한 DIP 스위치를 설정하거나, 프로그래밍 방식으로 접속 단자를 비활성화합니다. 특정 HBA에서 사용되는 방법을 보려면 해당 HBA의 설명서를 참조하십시오.

ILT 케이블을 Y 케이블과 함께 사용하면 공유 SCSI 버스에 추가 노드를 연결할 수 있습니다. HP에서 구할 수 있는 일부 SCSI 케이블은 ILT 및 Y 케이블 기능과 결합되어 있습니다. Y 케이블의 중간 커넥터에 연결된 모든 노드에는 HBA에서 비활성화된 SCSI 접속 단자도 있어야 합니다.

ILT 케이블이 사용되는 경우 버스의 접속 단자를 분리하지 않고 공유 SCSI 버스의 끝에서 호스트의 연결을 물리적으로 끊을 수 있으며, 복구되는 동안 클러스터의 나머지 노드에서는 공유 SCSI 버스에 계속 액세스할 수 있습니다.

마찬가지로 버스의 접속 단자를 분리하지 않고 공유 SCSI 버스에 있는 Y 케이블의 중간 커넥터에서 호스트의 연결을 물리적으로 끊을 수 있습니다.

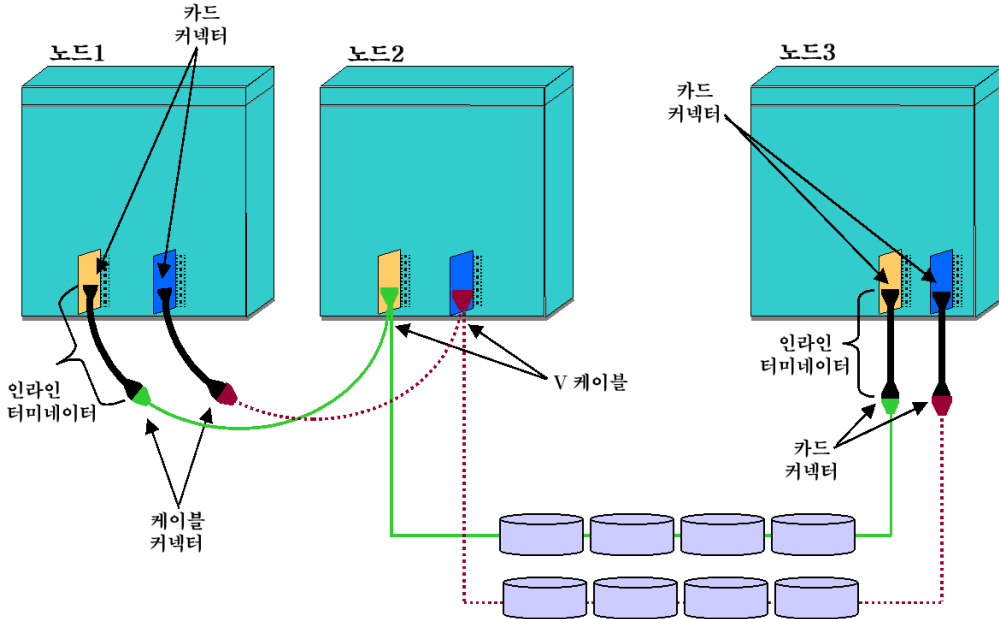
그러나 ILT 케이블을 사용하던 Y 케이블을 사용하던 해당 공유 SCSI 버스에 연결된 모든 노드를 먼저 중지하지 않고 HBA를 공유 SCSI 버스에 물리적으로 다시 연결하려고 하지 않는 것이 좋습니다. 새시 접지에 대해 커넥터의 핀을 실수로 단락시키기 쉽고 이 경우 공유 SCSI 버스에 연결된 다른 HBA가 손상되고 전체 SCSI 버스가 작동 중지될 수 있기 때문입니다.

주

D 및 K 시리즈 시스템에서는 내부 FW/SCSI 버스에서 인라인 단자를 사용할 수 없으며 단일 중단 SCSI 버스에서도 인라인 단자를 사용할 수 없습니다. 노드를 Y 케이블에 연결할 때는 인라인 단자를 사용하지 **말아야 합니다**.

그림 8-1에서는 두 개의 F/W SCSI-2 버스가 있는 3-노드 클러스터를 보여 줍니다. 실선과 점선은 각각 노드 1과 3에 인라인 단자가 연결된 서로 다른 두 개의 버스를 나타냅니다. 또한 Y 케이블은 노드 2에 연결되어 있습니다.

그림 8-1 인라인 단자가 있는 F/W SCSI-2 버스



인라인 SCSI 단자를 사용하면 일시적으로 패키지를 다른 노드로 이동한 후 하드웨어를 정비하는 동안 원래 노드를 중지하여 주어진 노드에서 하드웨어 유지 관리를 수행할 수 있습니다. 교체한 후에는 패키지를 원래 노드로 다시 이동할 수 있습니다.

다음 절차에 따라 인라인 SCSI 단자 또는 Y 케이블을 사용하여 버스에 연결된 노드를 연결 해제합니다.

1. 유지 관리가 필요한 노드의 패키지를 다른 노드로 이동합니다.
2. 유지 관리가 필요한 노드를 중지합니다. 클러스터가 재편성되고 다른 노드에서 작업이 계속됩니다. 중지된 노드의 패키지가 다른 노드로 전환되도록 구성된 경우에는 사용 가능한 다른 노드로 전환됩니다.
3. 노드에서 전원을 끕니다.
4. 필요하면 인라인 단자 케이블이나 Y 케이블에서 노드의 연결을 끊습니다. 인라인 단자 또는 Y 케이블이 버스에 연결되어 있는 동안에는 버스에 액세스하는 다른 노드에 문제가 발생하지 않습니다.

5. 필요하면 노드에서 하드웨어를 교체 또는 업그레이드합니다.
6. 공유 SCSI 버스에 연결된 노드를 모두 중지하고 전원을 끕니다.
7. 필요하면 인라인 단자 케이블 또는 Y 케이블에 노드를 다시 연결합니다.
8. 클러스터의 노드에 전원을 공급합니다. `/etc/rc.config.d/cmcluster` 파일에 `AUTOSTART_CMCLD`가 1로 설정되어 있으면 노드에서 클러스터 및 패키지를 자동으로 시작합니다.
9. 필요하면 대체 위치에서 노드로 패키지를 다시 이동하고 다시 시작합니다.

입출력 카드 교체

입출력 카드가 고장났을 때는 다음과 같은 절차로 카드를 교체할 수 있습니다. 각 노드에서 SCSI 인라인 단자나 Y 케이블을 사용하는 경우에는 이 작업을 위해 클러스터를 종료하지 않아도 됩니다.

1. 노드를 중지합니다. **Serviceguard Manager**에서 중지할 노드를 선택하고 동작 메뉴에서 관리 -> 노드 중지를 선택합니다. 또는 **Serviceguard** 명령줄에서 `cmhaltnode` 명령을 실행합니다. 패키지가 장애 조치를 위해 정상적으로 다른 노드로 전달되어야 합니다.
2. 카드에서 입출력 케이블을 제거합니다. SCSI 인라인 단자가 있으면 이렇게 해도 버스상의 디스크 또는 다른 노드에는 영향을 주지 않습니다.
3. **SAM**을 사용하여 온라인 입출력 카드 교체 옵션을 선택합니다.
4. 고장난 입출력 카드를 제거합니다.
5. 새 카드를 설치합니다. 새 카드는 정확하게 같은 종류의 카드이어야 하며 제거한 카드와 같은 슬롯에 설치해야 합니다.
6. **SAM**에서 새 입출력 카드 연결 옵션을 선택합니다.
7. 노드를 다시 클러스터에 추가합니다. **Serviceguard Manager**에서 추가할 노드를 선택하고 동작 메뉴에서 관리 -> 노드 이동을 선택합니다. 또는 **Serviceguard** 명령줄에서 `cmrunnode` 명령을 실행합니다.

LAN 카드 교체

LAN 카드 장애가 발생하여 LAN 카드를 교체해야 하는 경우 실행 중인 하드웨어 및 운영 체제의 종류에 따라 온라인이나 오프라인으로 교체할 수 있습니다. 이 작업을 위해 클러스터를 중지할 필요는 없습니다.

오프라인 교체

다음 단계에서는 LAN 카드를 오프라인으로 교체하는 방법을 보여 줍니다. 이 단계는 HP-UX 11.0과 11i에 모두 적용됩니다.

1. `cmhaltnode` 명령을 사용하여 노드를 중지합니다.
2. `/usr/sbin/shutdown`을 사용하여 시스템을 종료한 다음 시스템 전원을 끕니다.
3. 고장난 LAN 카드를 제거합니다.
4. 새 LAN 카드를 설치합니다. 새 카드는 정확하게 같은 종류의 카드이어야 하며 제거한 카드와 같은 슬롯에 설치해야 합니다.
5. 시스템의 전원을 켭니다.
6. 필요한 경우 `cmrunnode` 명령을 사용하여 클러스터에 노드를 다시 추가합니다. 노드가 자동으로 클러스터에 참가하도록 구성한 경우 이 단계를 생략할 수 있습니다.

온라인 교체

시스템 하드웨어에서 핫 스왑 입출력 카드를 지원하고 시스템에서 HP-UX 11i(B.11.11 이상)가 실행 중이면 문제가 있는 LAN 카드를 온라인으로 교체하도록 선택할 수 있습니다. 이렇게 하면 시스템의 전체 가용성이 크게 향상됩니다. 이 작업을 수행하려면 *Configuring HP-UX for Peripherals* 설명서의 “How to On-line Replace (OLR) a PCI Card Using SAM” 절에서 설명하는 단계를 따릅니다. OLR 절차에서도 새 카드는 네트워크 드라이버의 잘못된 동작을 방지하기 위해 제거한 카드와 똑같은 종류의 카드여야 합니다. Serviceguard에서는 LAN 카드가 교체되고 네트워크에 다시 연결되면 카드를 자동으로 복구합니다.

카드 교체 이후

LAN 카드의 온라인 또는 오프라인 교체가 완료되면 Serviceguard에서는 카드의 MAC 주소(LLA)가 클러스터의 이진 구성 파일에 저장된 값에서 변경되었음을 감지하고 새로운 MAC 주소를 클러스터의 다른 노드에 알립니다. 그러면 클러스터가 정상적으로 작동됩니다.

또한 클러스터 구성을 다시 적용하여 클러스터의 이진 구성 파일에서 새로운 MAC 주소를 업데이트하는 것도 좋습니다. 다음 단계를 따라 온라인 재구성을 수행하십시오.

1. 다음과 같이 cmgetconf 명령을 사용하여 최신 ASCII 구성 파일을 얻습니다.

```
# cmgetconf config.ascii
```

2. cmapplyconf 명령을 사용하여 구성을 적용하고 새 이진 파일을 모든 클러스터 노드에 복사합니다.

```
# cmapplyconf -C config.ascii
```

이 과정은 새로운 MAC 주소로 이진 파일을 업데이트하여 cmviewconcl 및 lanscan 명령의 출력 데이터가 서로 일치하도록 만듭니다.

잘못된 퀴럼 서버 시스템 교체

퀴럼 서버에 장애가 발생하거나 Quorum 서비스를 제공 받는 클러스터에서 퀴럼 서버를 사용할 수 없게 되는 경우 어떤 클러스터에서도 장애가 발생하지 않습니다. 그러나 퀴럼 서버가 손실되면 추가 장애가 있을 경우 클러스터의 취약성이 증가합니다. 다음 절차를 사용하여 결함 있는 퀴럼 서버 시스템을 교체합니다.

이 절차를 사용할 경우 클러스터 노드의 구성은 변경할 필요가 없습니다.

1. 네트워크에서 이전 퀴럼 서버 시스템을 제거합니다.
2. 새 시스템을 설정하고 이전 퀴럼 서버의 IP 주소 및 호스트 이름으로 새 시스템을 구성합니다.
3. 새 시스템에 퀴럼 서버 소프트웨어를 설치하고 구성합니다. 이전 퀴럼 서버에 대해 구성된 모든 노드를 새 QS 권한 파일(/etc/cmcluster/qs_authfile)에 포함해야 합니다. QS 권한 파일의 구성 방법에 대한 자세한 내용은 `qs(1)` 맨페이지를 참조하십시오.
4. 다음과 같이 퀴럼 서버를 시작합니다.
 - /etc/inittab 파일을 편집하여 퀴럼 서버 항목을 추가합니다.
 - `init q` 명령을 사용하여 퀴럼 서버를 실행합니다.

자세한 내용은 `qs(1)` 맨페이지를 참조하십시오.

5. 이전 퀴럼 서버를 사용하던 모든 클러스터의 모든 노드가 새 퀴럼 서버에 연결됩니다. 퀴럼 서버를 사용하고 있는 임의의 클러스터에서 `cmviewcl -v` 명령을 사용하여 해당 클러스터의 노드가 퀴럼 서버에 연결되었는지 확인합니다.
6. 새 퀴럼 서버의 퀴럼 서버 로그 파일에는 퀴럼 서버를 사용하는 각 클러스터에 대해 다음과 같은 로그 메시지가 표시됩니다.

```
Request for lock /sg/<ClusterName> succeeded. New lock owners: N1, N2
```

7. 퀴럼 서버가 올바르게 구성되었는지, 노드가 퀴럼 서버에 연결되었는지 확인하려면 클러스터 노드에서 다음 명령을 실행하면 됩니다.

```
# cmquerycl -q <QSHostName> -n <Node1> -n <Node2> ...
```

지정한 노드가 퀴럼 서버에 연결할 수 없는 경우 오류 메시지가 출력됩니다.

주

이전 퀴럼 서버가 종료되고 새 퀴럼 서버가 설정되는 동안에는 다음과 같이 됩니다.

- `cmquerycl`, `cmcheckconf` 및 `cmapplyconf` 명령이 동작하지 않습니다.
- `cmruncl`, `cmhaltcl`, `cmrunnode` 및 `cmhaltnode` 명령이 동작하지 않습니다.
- 구성원을 정확히 반씩 나누는 노드 장애 또는 네트워크 장애가 발생할 경우에는 퀴럼 서버를 타이 브레이커로 사용할 수 없고 클러스터에 장애가 발생하게 됩니다.

경고

이전 시스템이 이전 IP 주소로 네트워크에 다시 참가하지 않도록 하십시오.

문제 해결 방법

다음 절에서는 실행 중인 시스템의 상태를 검토하고 클러스터 상태 데이터, 로그 파일 및 구성 파일을 조사하여 문제 해결을 수행할 수 있는 몇 가지 방법을 제안합니다. 다음과 같은 항목이 제공됩니다.

- 패키지 IP 주소 확인
- 시스템 로그 파일 확인
- 구성 파일 확인
- 패키지 제어 스크립트 확인
- `cmquerycl` 및 `cmcheckconf` 사용
- `cmscancl` 및 `cmviewcl` 사용
- LAN 구성 확인

주

클러스터의 현재 상태를 확인하고 클러스터 객체의 속성을 보려면 **Serviceguard Manager**를 사용하는 것이 좋습니다. **Serviceguard Manager** 실행 방법에 대한 자세한 내용은 7장의 “**Serviceguard Manager 사용**”을 참조하십시오.

패키지 IP 주소 확인

`netstat -in` 명령을 사용하여 LAN 구성을 조사할 수 있습니다. 노드 `ftsys10`을 중지한 다음 `ftsys9`에 대해 이 명령을 실행하면 다음과 같이 패키지 IP 주소가 하트비트 IP 주소와 함께 `ftsys9`의 `lan0`에 지정된 것으로 표시됩니다.

| Name | Mtu | Network | Address | Ipkts |
|--------|------|-----------|--------------|--------|
| Opkts | | | | |
| ni0* | 0 | none | none | 0 |
| 0 | | | | |
| ni1* | 0 | none | none | 0 |
| 0 | | | | |
| lo0 | 4608 | 127 | 127.0.0.1 | 10114 |
| 10114 | | | | |
| lan0 | 1500 | 15.13.168 | 15.13.171.14 | 959269 |
| 305189 | | | | |

| | | | | |
|----------------|------|-----------|--------------|--------|
| lan0 305189 | 1500 | 15.13.168 | 15.13.171.23 | 959269 |
| lan0 305189 | 1500 | 15.13.168 | 15.13.171.20 | 959269 |
| lan1* 41716 | 1500 | none | none | 418623 |

시스템 로그 파일 확인

클러스터 관리자 및 패키지 관리자에서 생성하는 메시지는 시스템 로그 파일에 기록됩니다. 로그 파일의 기본 위치는 `/var/adm/syslog/syslog.log`입니다. `vi`와 같은 텍스트 편집기나 `more` 명령을 사용하여 클러스터에 대해 기록된 정보의 로그 파일을 볼 수 있습니다.

이 로그는 다음과 같은 정보를 제공합니다.

- 실행된 명령과 그 결과
- 오류 또는 오류가 아닌 주요 클러스터 이벤트
- 클러스터 상태 정보

주

`Serviceguard` 및 `HP-UX`에서 실행 중인 다른 많은 제품이 `syslog.log` 파일을 사용하여 메시지를 저장합니다. `HP-UX`의 **시스템 및 작업 그룹 관리** 설명서에서는 시스템 로그 사용에 대해 자세히 설명합니다.

샘플 시스템 로그 항목

`/var/adm/syslog/syslog.log` 파일의 다음 항목은 `pkg5_run` 스크립트의 문제로 실행되지 못한 패키지를 보여 줍니다. 자세한 내용은 `pkg5_run.log`를 참조하십시오.

```
Dec 14 14:33:48 star04 cmcld[2048]: Starting cluster management protocols.
Dec 14 14:33:48 star04 cmcld[2048]: Attempting to form a new cluster
Dec 14 14:33:53 star04 cmcld[2048]: 3 nodes have formed a new cluster
Dec 14 14:33:53 star04 cmcld[2048]: The new active cluster membership is:
    star04(id=1) , star05(id=2), star06(id=3)
```



```
Dec 14 17:33:53 star04 cmlvmd[2049]: Clvmd initialized success
fully.
Dec 14 14:34:44 star04 CM-CMD[2054]: cmrunpkg -v pkg5
Dec 14 14:34:44 star04 cmclld[2048]: Request from node star04 t
o start
    package pkg5 on node star04.
Dec 14 14:34:44 star04 cmclld[2048]: Executing '/etc/cmcluster/
pkg5/pkg5_run
    start' for package pkg5.
Dec 14 14:34:45 star04 LVM[2066]: vgchange -a n /dev/vg02
Dec 14 14:34:45 star04 cmclld[2048]: Package pkg5 run script ex
ited with
    NO_RESTART.
Dec 14 14:34:45 star04 cmclld[2048]: Examine the file
    /etc/cmcluster/pkg5/pkg5_run.log for more details.
다음은 성공적인 패키지 시작의 예입니다.
Dec 14 14:39:27 star04 CM-CMD[2096]: cmruncl
Dec 14 14:39:27 star04 cmclld[2098]: Starting cluster managemen
t protocols.
Dec 14 14:39:27 star04 cmclld[2098]: Attempting to form a new c
luster
Dec 14 14:39:27 star04 cmclconfd[2097]: Command execution mess
age
Dec 14 14:39:33 star04 cmclld[2098]: 3 nodes have formed a new
cluster
Dec 14 14:39:33 star04 cmclld[2098]: The new active cluster mem
bership is:
    star04(id=1), star05(id=2), star06(id=3)
Dec 14 17:39:33 star04 cmlvmd[2099]: Clvmd initialized success
fully.
Dec 14 14:39:34 star04 cmclld[2098]: Executing '/etc/cmcluster/
pkg4/pkg4_run
    start' for package pkg4.
Dec 14 14:39:34 star04 LVM[2107]: vgchange /dev/vg01
Dec 14 14:39:35 star04 CM-pkg4[2124]: cmmodnet -a -i 15.13.168
.0 15.13.168.4
Dec 14 14:39:36 star04 CM-pkg4[2127]: cmrunserv Service4 /vg01
/MyPing 127.0.0.1
    >>/dev/null
Dec 14 14:39:36 star04 cmclld[2098]: Started package pkg4 on no
de star04.
```

객체 관리자 로그 파일 확인

Serviceguard 객체 관리자 데몬 cmomd는 /var/opt/cmom/cmomd.log 파일에 메시지를 기록합니다. 다음과 같이 cmreadlog 명령을 사용하여 이 메시지를 확인할 수 있습니다.

```
# cmreadlog /var/opt/cmom/cmomd.log
```

cmomd가 기록하는 메시지에는 객체 관리자에 데이터를 요청한 프로세스에 대한 정보가 들어 있으며, 데이터 유형, 타임스탬프 등이 포함됩니다. 객체 관리자에 데이터를 요청하는 클라이언트의 예로는 Serviceguard Manager가 있습니다.

Serviceguard Manager 로그 파일 확인

Serviceguard Manager에서는 사용자 작업에 대한 로그 파일을 유지 관리합니다.

이 파일은 HP-UX 디렉토리 /var/opt/sgmgr 또는 Windows 디렉토리

X:\Program Files\Hewlett-Packard\Serviceguard Manager\log에 저장됩니다. 여기서 X는 Serviceguard Manager가 설치된 드라이브를 나타냅니다. 다음의 HP-UX 예제와 같이 cmreadlog 명령을 사용하여 이 메시지를 확인할 수 있습니다.

```
# cmreadlog /var/opt/sgmgr/929917sgmgr.log
```

Serviceguard Manager에서 기록하는 메시지에는 로그인 날짜 및 시간, 객체 관리자 서버 시스템, 타임스탬프 등이 들어 있습니다.

구성 파일 확인

다음의 ASCII 구성 파일을 확인하십시오.

- 클러스터 구성 파일
- 패키지 구성 파일

구성 계획 워크시트에 따라 파일이 완전하고 올바른지 확인하십시오.

패키지 제어 스크립트 확인

패키지가 실행될 수 있고 해당 파일이 동일한 모든 노드에서 패키지 제어 스크립트를 찾을 수 있는지 확인합니다. 스크립트가 모든 노드에서 실행 가능한지 확인합니다.

제어 스크립트의 이름이 패키지 구성 파일에 나타나는지 확인하고 패키지 구성 파일에 명명된 모든 서비스도 패키지 제어 스크립트에 나타나는지 확인합니다.

각 패키지의 시작 및 중지 관련 정보는 패키지의 제어 스크립트 로그에서 찾을 수 있습니다. 이 로그는 패키지 제어 스크립트의 작동에 대한 기록을 제공합니다. 이 기록은 `/etc/cmcluster/package_name/control_script.log`에 있습니다. 이 로그에는 모든 패키지 실행 및 중지 활동이 기록됩니다. 패키지에 대해 별도의 실행 및 중지 스크립트를 작성한 경우 각 스크립트에는 자체의 로그가 있습니다.

cmcheckconf 명령 사용

구성을 확인할 때와 같은 방법으로 `cmcheckconf`를 사용하여 클러스터의 문제를 해결할 수도 있습니다.

다음 예제에서는 `ftsys9` 및 `ftsys10`의 기존 클러스터 구성을 확인하는 데 사용된 명령을 보여 줍니다.

```
# cmquerycl -v -C /etc/cmcluster/verify.ascii -n ftsys9 -n ftsys10
# cmcheckconf -v -C /etc/cmcluster/verify.ascii
```

`cmcheckconf` 명령을 사용하면 다음 사항을 확인할 수 있습니다.

- 네트워크 주소 및 연결
- 클러스터 잠금 디스크 연결
- 다음에 대한 클러스터 및 패키지의 구성 매개 변수 유효성
 - 이름의 고유성
 - 스크립트의 존재 여부 및 사용 권한

다음 사항은 확인하지 않습니다.

- 올바른 전원 회로 설정
- 패키지 구성 스크립트의 정확성

cmscanc1 명령 사용

`cmscanc1` 명령은 IP 주소 또는 서브넷, 디스크의 물리 볼륨 이름, 클러스터의 모든 노드에 있는 노드에 특정한 기타 항목을 비교할 수 있도록 체계적인 보고서에 클러스터의 모든 노드에 대한 정보를 표시합니다. `cmscanc1`은 실제로 모든 노드에 대해 몇 가지 다른 HP-UX 명령을 실행하고 명령을 실행한 노드의 보고서에 실행 결과를 기록합니다.

다음은 cmsscanc1이 각 노드에 대해 표시하는 구성 데이터의 유형입니다.

표 8-1 cmsscanc1 명령으로 표시되는 데이터

| 설명 | 데이터 원본 |
|--------------------|----------------|
| LAN 장치 구성 및 상태 | lanscan 명령 |
| 네트워크 상태 및 인터페이스 | netstat 명령 |
| 파일 시스템 | mount 명령 |
| LVM 구성 | /etc/lvmtab 파일 |
| LVM 물리 볼륨 그룹 데이터 | /etc/lvmimg 파일 |
| 모든 링크에 대한 링크 수준 연결 | linkloop 명령 |
| 이진 구성 파일 | cmviewconf 명령 |

cmviewconf 명령 사용

cmviewconf를 사용하면 클러스터가 실행 중이지 않을 때에도 이진 클러스터 구성 파일을 검사할 수 있습니다. 이 명령을 사용하면 명령을 실행 중인 노드에 있는 구성 파일의 내용이 표시됩니다.

LAN 구성 확인

다음 네트워킹 명령은 문제를 진단하는 데 사용할 수 있습니다.

- netstat -in은 LAN 구성을 검사하는 데 사용할 수 있습니다. 이 명령은 각 LAN 인터페이스 카드에 지정된 모든 IP 주소를 나열합니다.
- lanscan은 LAN 구성을 검사하는 데에도 사용할 수 있습니다. 이 명령은 노드에 있는 각 LAN 인터페이스 카드의 MAC 주소와 상태를 나열합니다.
- arp -a는 arp 테이블을 확인하는 데 사용할 수 있습니다.

- `landiag`는 LAN 카드 정보를 표시, 진단 및 재설정하는 데 유용합니다.
- `linkloop`는 MAC 주소 수준에서 LAN 카드 간의 통신을 확인합니다. 예를 들어 다음과 같이 입력합니다.

```
# linkloop -i4 0x08000993AB72
```

그 결과로 다음과 같은 메시지를 볼 수 있습니다.

```
Link Connectivity to LAN station: 0x08000993AB72 OK
```

- `cmscancl`은 주 LAN과 대기 LAN이 동일한 브리지 연결 네트워크에 있는지 확인하는 데 사용할 수 있습니다.
- `cmviewcl -v`는 주 LAN과 대기 LAN의 상태를 보여 줍니다.

모든 노드에 대해 이러한 명령을 사용하십시오.

문제 해결

Serviceguard 관련 문제는 몇 가지 유형으로 구분할 수 있습니다. 다음 목록은 문제를 일반적인 범주로 구분한 것입니다.

- Serviceguard 명령 중지
- 클러스터 재편성
- 시스템 관리 오류
- 패키지 제어 스크립트 중지
- VxVM 디스크 그룹 문제
- 패키지 이동 오류
- 노드 및 네트워크 장애
- 퀘럼 서버 문제

처음 두 가지 범주의 문제는 잘못된 Serviceguard 구성으로 인해 발생합니다. 마지막 범주에는 응용 프로그램의 가용성을 유지할 수 있도록 Serviceguard에서 반응하도록 설계된 “보통” 장애가 포함됩니다.

Serviceguard 명령 중지

cmviewcl을 포함하여 대부분의 Serviceguard 명령은 이름 확인 서비스에 의존하여 클러스터 노드의 주소를 알아냅니다. 이름 확인 서비스를 사용할 수 없을 경우(예: 네임 서버가 다운되었을 때), Serviceguard 명령이 실행 중지되거나 네트워크 관련 오류 메시지를 반환할 수 있습니다. 이런 문제가 발생하면 각 클러스터 노드에서 nslookup 명령을 사용하여 이름 확인 기능이 올바르게 작동하는지 확인하십시오. 예를 들면 다음과 같습니다.

```
# nslookup ftsys9
```

```
Name Server:  server1.cup.hp.com  
Address:  15.13.168.63
```

```
Name:  ftsys9.cup.hp.com  
Address:  15.13.172.229
```

이 명령의 출력에 노드의 올바른 IP 주소가 나타나지 않으면 사용하고 있는 이름 확인 서비스의 이상 유무를 검사하십시오.

클러스터 재편성

현재 클러스터 상황에 따라 때때로 클러스터 재편성이 일어날 수 있습니다. 다음과 같은 원인이 있습니다.

- 이더넷 LAN상의 로컬 전환에서 전환 과정이 클러스터 NODE_TIMEOUT 값보다 오래 걸리는 경우. 클러스터 NODE_TIMEOUT 값을 늘리거나 다른 종류의 LAN을 사용하여 이 문제를 방지할 수 있습니다.
- 하트비트 LAN상에 과도한 네트워크 트래픽이 발생하는 경우. 전용 하트비트 LAN을 사용하거나 보다 트래픽이 적은 LAN을 사용하여 이 문제를 방지할 수 있습니다.
- 시스템 과부하, 입출력 및 네트워크 트래픽의 총량이 너무 많은 경우
- 올바르게 않게 구성된 네트워크(예: 라우팅 테이블이 너무 큰 경우)

이러한 경우 응용 프로그램은 계속 실행되지만 클러스터를 재편성하는 동안 성능이 약간 저하될 수 있습니다.

시스템 관리 오류

Serviceguard를 구성할 때는 클러스터 시작 시 표시되지 않는 많은 오류를 저지를 수 있습니다. 하드웨어 또는 소프트웨어 장애가 발생하고 패키지 제어권이 예상대로 다른 노드로 전달되지 않을 때까지 클러스터가 실행되고 아무런 문제가 없는 것처럼 보일 수 있습니다.

이러한 오류는 특히 클러스터 구성 파일과 패키지 구성 스크립트의 오류로 인해 발생합니다. 다음은 이러한 오류의 예입니다.

- 볼륨 그룹이 대체 노드에 정의되어 있지 않습니다.
- 마운트 지점이 대체 노드에 없습니다.
- 대체 노드에 네트워크 오류가 있습니다(구성 오류).
- 사용자 정보가 대체 노드에서 올바르지 않습니다.

다음 명령을 사용하여 디스크의 상태를 확인할 수 있습니다.

- `bdf` - 패키지의 볼륨 그룹이 마운트되었는지 여부를 확인합니다.
- `vgdisplay -v` - 모든 볼륨이 있는지 여부를 확인합니다.
- `lvdisplay -v` - 미러가 동기화되었는지 여부를 확인합니다.
- `strings /etc/lvmtab` - 구성이 올바른지 확인합니다.

- `ioscan -fnC disk` - 물리 디스크를 확인합니다.
- `diskinfo -v /dev/rdisk/cxtydz` - 디스크에 대한 정보를 표시합니다.
- `lssfs /dev/dsk/*d0 -LV` 및 경로를 검사합니다.
- `vxvg list -VERITAS` 디스크 그룹을 나열합니다.
- `vxprint -VERITAS` 디스크 그룹의 세부 사항을 표시합니다.

패키지 제어 스크립트 중지 또는 실패

`RUN_SCRIPT_TIMEOUT` 또는 `HALT_SCRIPT_TIMEOUT` 값이 설정되어 있을 때 제어 스크립트가 중지되어 시간 종료 값이 초과되면 **Serviceguard**에서는 스크립트를 강제 종료하고 해당 패키지를 “꺼짐”으로 표시합니다. 마찬가지로 패키지 제어 스크립트에 오류가 발생하면 **Serviceguard**에서는 스크립트를 강제 종료하고 해당 패키지를 “꺼짐”으로 표시합니다. 두 경우 모두 다음과 같은 상황이 발생합니다.

- 패키지 제어권이 전달되지 않습니다.
- 실행 또는 중지 명령이 완료되지 않습니다.
- 글로벌 전환을 할 수 없게 됩니다.
- 현재 노드에서 패키지를 실행할 수 없게 됩니다.

제어 스크립트가 중단되었으므로 이러한 장애가 발생했을 때는 패키지의 리소스 중 일부가 활성 상태로 있을 수 있습니다. 특히 다음과 같은 경우가 있을 수 있습니다.

- 볼륨 그룹이 활성 상태임
- 파일 시스템이 마운트된 상태임
- IP 주소가 설치된 상태임
- 서비스가 계속 실행 중임

이러한 상황에서 관리자가 직접 조치하지 않으면 **Serviceguard**에서는 패키지를 다시 시작하지 않습니다. 패키지를 다시 시작하기 전에 관리자가 직접 정리해야 합니다.

다음 단계를 기본 지침으로 삼으십시오.

1. 응용 프로그램 관련 부분을 정리합니다. 다른 노드에서 패키지가 정상적으로 시작 되도록 하려면 제어 스크립트가 응용 프로그램에 대해 수행한 모든 작업을 복원해야 합니다. 여기에는 응용 프로그램 프로세스를 종료시키고 잠금 파일을 제거하고 임시 파일을 제거하는 등의 조치가 포함됩니다.

2. 패키지 IP 주소가 시스템에서 제거되었는지 확인합니다. 이 단계는 `cmmodnet (1m)` 명령으로 수행할 수 있습니다. 먼저 `netstat -in` 명령을 실행하여 출력된 결과를 검사하여 설치된 패키지 IP 주소를 확인합니다. 패키지 제어 스크립트에 지정된 IP 주소가 `netstat` 출력에서 “Address” 열(IPv4의 경우)이나 “Address/Prefix” 열(IPv6의 경우)에 나타날 경우 `cmmodnet`를 사용하여 이 IP 주소를 제거하십시오.

```
# cmmodnet -r -i <ip-address> <subnet>
```

여기서 `<ip-address>`는 “Address” 또는 “Address/Prefix” 열의 주소이고 `<subnet>`은 “Network” 열(IPv4의 경우) 또는 접두사(IPv6의 경우, IPV6 주소에서 파생될 수 있음) 열에 있는 해당 항목입니다.

3. 패키지 볼륨 그룹이 비활성화되었는지 확인합니다. 먼저 파일 시스템으로 사용되고 있는 패키지 논리 볼륨이 있는지 확인하여 마운트 해제합니다. `bdf -l` 명령을 실행하여 출력된 결과를 조사하여 이를 확인할 수 있습니다. 패키지 논리 볼륨이 패키지 제어 스크립트의 `LV[]` 배열 변수에 의해 지정된 것과 같이 “Filesystem” 열 아래에 나타날 경우 `umount`를 사용하여 마운트 해제하십시오.

```
# fuser -ku <logical-volume>
```

```
# umount <logical-volume>
```

그런 다음 패키지 볼륨 그룹을 비활성화합니다. 이 작업은 패키지 제어 스크립트의 `VG[]` 배열 항목에서 지정합니다.

```
# vgchange -a n <volume-group>
```

4. 마지막으로 패키지를 다시 전환 가능하게 합니다.

```
# cmmodpkg -e <package-name>
```

시간 종료가 발생한 노드를 정리한 후 해당 노드가 패키지를 실행할 수 있는 대체 노드가 되도록 하려면 패키지를 이 노드에서 실행할 수 있도록 다시 설정해야 합니다.

```
# cmmodpkg -e -n <node-name> <package-name>
```

기본 Serviceguard 제어 스크립트는 간단한 단계만으로 응용 프로그램을 실행하고 중지할 수 있도록 설계되었습니다. 패키지 관리자가 이 절차의 각 단계가 완료되어야 하는 시간 종료 값을 설정한 경우 어떤 이유로 이 시간 종료 값이 초과되면, Serviceguard에서는 보수적인 접근 방법을 택하여 제어 스크립트의 논리가 어떤 식으로든 중지되었거나 손상되었을 것이라고 판단합니다. 이 시점부터는 제어 스크립트가 정리 작업을 올바르게 수행할 수 있을 것이라고 신뢰할 수 없으므로 스크립트를 강제 종료시키고 필요한 정리 단계를 패키지 관리자가 직접 수행할 수 있게 합니다.

제어 스크립트 시간 종료 발생 시 패키지 자동 전환을 하려면 `NODE_FAIL_FAST_ENABLED` 매개 변수를 `YES`로 설정합니다. **Serviceguard Manager**를 사용하는 경우에는 **패키지 빠른 장애 조치를 사용 가능**으로 설정합니다. 이 경우 **Serviceguard**에서는 제어 스크립트 시간 종료 발생 시 **TOC**를 발생시킵니다. 이렇게 하면 패키지의 실행 또는 중지 시도 중에 발생한 모든 부수 효과가 효과적으로 정리됩니다. 이 경우, 패키지는 해당 패키지에 대해 구성된 대체 노드 중 사용 가능한 노드에서 자동으로 다시 시작됩니다.

VxVM 디스크 그룹 문제

이 절에서는 클러스터 환경의 **VxVM** 디스크 그룹에서 발생할 수 있는 문제를 해결하기 위한 몇 가지 방법을 설명합니다. 대부분의 문제에서 `vxdg list` 명령을 사용하여 현재 특정 노드에 가져온 디스크 그룹을 표시하는 것이 유용합니다. 또한 패키지 제어 스크립트 로그 파일에서 특정 노드에서 디스크 그룹을 가져오고 내보내는 데 관련된 메시지를 확인해야 합니다.

노드 장애 후 강제로 가져오기 및 내보내기

특정 장애가 발생한 후에는 **VxVM** 디스크 그룹으로 구성된 패키지가 시작되지 못하고 패키지 로그 파일에 다음 오류가 나타납니다.

```
vxdg: Error dg_01 may still be imported on ftsys9
      ERROR: Function check_dg failed
```

패키지가 노드에서 실행 중이고 패키지 제어 스크립트에서 디스크 그룹을 내보내기 전에 장애가 발생하면 이 오류 메시지가 나타날 수 있습니다. 이 경우 장애가 발생한 노드의 호스트 이름이 디스크 그룹 헤더에 기록되어 있습니다.

패키지가 클러스터의 다른 노드에서 시작되면 다음 예제와 같은 일련의 메시지가 출력됩니다. 여기서 장애가 발생한 시스템의 호스트 이름은 **ftsys9**이고 디스크 그룹은 **dg_01**입니다.

```
check_dg: Error dg_01 may still be imported on ftsys9
```

To correct this situation, logon to ftsys9 and execute the following command:

```
vxdg deport dg_01
```

Once dg_01 has been deported from ftsys9,
this package may be restarted via either `cmmodpkg(1M)`
or `cmrunpkg(1M)`.

In the event that ftsys9 is either powered off
or unable to boot, then dg_01 must be force
imported.

***** WARNING*****

The use of force import can lead to data corruption if
ftsys9 is still running and has dg_01
imported. It is imperative to positively determine that
ftsys9 is not running prior to performing the force
import. See `-C` option on `vx dg(1M)`.

To force import dg_01, execute the following
commands on the local system:

```
vx dg -tfC import $vg  
vx dg deport $vg
```

메시지의 지침에 따라 강제 가져오기 옵션(-C)을 사용하여 현재 노드에서 디스크 그룹
을 가져올 수 있도록 합니다. 그런 다음 디스크 그룹을 내보내 패키지에서 다시 사용할
수 있도록 합니다. 예를 들면 다음과 같습니다.

```
# vx dg -tfC import dg_01
```

```
# vx dg deport dg_01
```

강제 가져오기는 현재 디스크 그룹의 디스크에 쓰여진 호스트 이름을 삭제하고 오류 없
이 디스크 그룹을 내보내어 다른 노드에서 실행 중인 패키지에서 가져올 수 있도록 합
니다.

주의

이러한 강제 가져오기 절차는 현재 다른 노드에서 디스크에 액세스하지 않은 것이 확실
한 경우에만 사용해야 합니다. 다른 노드에서 액세스하고 있는 디스크를 강제로 가져오
면 데이터가 손상될 수 있습니다.

패키지 이동 오류

이 오류는 패키지 제어 스크립트의 오류로 인해 발생한 경우를 제외하고는 시스템 관리 오류와 유사합니다. 이러한 오류를 방지하는 가장 좋은 방법은 고가용성 응용 프로그램을 온라인 상태로 바꾸기 전에 패키지 제어 스크립트를 테스트하는 것입니다.

제어 스크립트의 둘째 줄에 “set -x” 문을 추가하면 스크립트가 어느 부분에서 실패했는지 자세한 정보가 표시됩니다.

노드 및 네트워크 장애

이 장애가 발생하면 Serviceguard에서는 패키지의 제어권을 다른 노드로 전달합니다. 이 작업은 Serviceguard의 일반적인 동작이지만 사용자는 전송이 발생한 때를 알 수 있어야 하고 클러스터를 계속 현재 상태로 둘 것인지 또는 원래 상태로 복원할 것인지를 결정해야 합니다.

다음과 같은 경우 노드 장애가 발생할 수 있습니다.

- HPMC. High Priority Machine Check의 줄임말로, 하드웨어 오류로 인해 일어난 시스템 패닉 상태를 의미합니다.
- TOC
- 패닉
- 중지
- 정전

TOC의 경우 장애가 발생한 노드에서 시스템 덤프가 수행되고 많은 메시지가 콘솔에 나타납니다.

다음 명령을 사용하여 네트워크 및 서버넷의 상태를 확인할 수 있습니다.

- netstat -in - LAN 상태를 표시하고 패키지 IP 주소가 LAN 카드에 있는지 여부를 확인합니다.
- lanscan - 해당 LAN이 기본 인터페이스에 있는지 아니면 대기 인터페이스로 전환되었는지를 확인합니다.
- arp -a - arp 테이블을 검사합니다.
- lanadmin - LAN 카드를 표시하고 테스트하고 재설정합니다.

사용하는 클러스터가 모두 동일하지 않으므로 모든 문제에 대한 완전한 해결책은 없습니다. 그러나 이 검사 내용과 명령을 적용하여 로그 파일에 따라 작업하면 성공적으로 문제를 식별하고 해결할 수 있을 것입니다.

쿼럼 서버 문제 해결

권한 파일 문제

Serviceguard 노드의 `syslog` 파일이나 `cmviewc1 -v`의 출력에서 나타나는 다음 종류의 메시지는 권한 문제를 나타낼 수 있습니다.

```
Access denied to quorum server 192.6.7.4
```

이 문제는 권한 파일을 업데이트하지 않은 경우에 발생할 수 있습니다. 파일에 노드가 들어 있는지 확인한 다음 `/usr/sbin/qs -update`를 사용하여 권한 파일을 다시 읽어 보십시오.

시간 종료 문제

Serviceguard 노드의 `syslog` 파일에 나타나는 다음 종류의 메시지는 시간 종료 문제를 나타낼 수 있습니다.

```
Unable to set client version at quorum server  
192.6.7.2:reply timed out  
Probe of quorum server 192.6.7.2 timed out
```

이러한 메시지는 기본 쿼럼 서버 시간 종료뿐만 아니라 가끔 발생하는 네트워크 문제를 나타낼 수 있습니다. `QS_TIMEOUT_EXTENSION`을 설정하여 시간 종료 값을 늘리거나, 하트비트 또는 노드 시간 종료 값을 늘리십시오.

Serviceguard 노드의 `syslog` 파일에 나타나는 다음 종류의 메시지는 노드가 적절한 시간에 노드의 잠금 요청에 대한 응답을 받지 않았다는 것을 나타냅니다. 클러스터에서 노드와 `qs` 또는 `qs`와 다른 노드 간의 통신 지연이 원인일 수 있습니다.

```
Attempt to get lock /sg/cluser1 unsuccessful. Reason:  
request_timedout
```

메시지

Serviceguard의 코디네이터 노드에서 잠금 상태를 설정하도록 쿼럼 서버에 요청을 보내는 경우가 있습니다. 이는 타이 브레이크에서 잠금을 얻기 위한 요청과는 다릅니다. 쿼럼 서버와 클러스터 노드 중 하나의 연결이 완전하지 않으면 설정 요청에 장애가 발생하고 쿼럼 서버의 로그 파일에 다음과 같은 두 줄의 메시지가 나타날 수 있습니다.

```
Oct 08 16:10:05:0: There is no connection to the applicant  
2 for lock /sg/lockTest1  
Oct 08 16:10:05:0:Request for lock /sg/lockTest1 from  
applicant 1 failed: not connected to all applicants.
```

이 경우는 무시할 수 있습니다. 요청은 몇 초 후에 다시 시도되어 성공하게 될 것입니다. 이때는 다음과 같은 메시지가 기록됩니다.

```
Oct 008 16:10:06:0: Request for lock /sg/lockTest1  
succeeded. New lock owners: 1,2.
```

A Serviceguard 명령

다음은 Serviceguard 클러스터 구성 및 유지 관리에 사용되는 명령의 알파벳순 목록입니다. 이러한 명령의 맨페이지는 **설치 후** 시스템에서 사용할 수 있습니다.

표 A-1

MC/Serviceguard 명령

| 명령 | 설명 |
|-------------|---|
| cmapplyconf | <p>Serviceguard 클러스터 구성 및 패키지 구성 파일을 확인하고 적용합니다.</p> <p>cmapplyconf는 cluster_ascii_file 및 관련된 pkg_ascii_file에 지정된 클러스터 구성과 패키지 구성을 확인하고, cmclconfig라는 이진 구성 파일을 만들거나 업데이트하여 모든 노드에 배포합니다. 이 이진 구성 파일에는 클러스터 구성 정보와 지정된 모든 패키지에 대한 패키지 구성 정보가 포함됩니다. 클러스터 데몬에서 전체 클러스터 및 패키지 환경을 관리하는 데 사용되는 이 파일은 /etc/cmcluster 디렉토리에 보관됩니다.</p> <p>클러스터 구성이나 패키지 구성 파일을 변경해야 하는 경우에는 먼저 해당 ASCII 파일(클러스터 또는 패키지)을 업데이트하고 cmcheckconf 명령을 사용하여 변경 내용을 확인한 다음 다시 cmapplyconf를 사용하여 이진 파일을 확인하고 모든 노드에 배포합니다. 클러스터 및 패키지 구성은 클러스터의 작동이 중지된 때만 수정할 수 있습니다. 클러스터 ASCII 파일은 처음으로 클러스터를 구성하거나 클러스터에 노드를 추가 또는 삭제하는 경우에만 지정해야 합니다. 패키지 ASCII 파일은 패키지를 추가하거나 패키지 구성을 수정하는 경우에만 지정해야 합니다.</p> |

표 A-1 MC/Serviceguard 명령(계속)

| 명령 | 설명 |
|---------------------|--|
| cmapplyconf (계속) | <p>기존 구성을 변경해야 할 때마다 cmgetconf 명령을 실행하여 클러스터 ASCII 구성 파일이나 패키지 ASCII 구성 파일을 가져오는 것이 좋습니다.</p> <p>cmapplyconf는 클러스터 구성 또는 패키지 파일을 확인하고 배포합니다. 이 명령으로 클러스터 데몬이 클러스터 구성에서 시작되거나 제거되지는 않습니다. 패키지 노드, 패키지 서브넷 등을 추가하거나 삭제할지 여부를 결정할 때도 같은 종류의 처리 과정이 패키지 구성에 적용됩니다. 패키지 구성을 변경하려면 항상 패키지를 중단해야 합니다.</p> |
| cmcheckconf | <p>고가용성 클러스터 구성 및/또는 패키지 구성 파일을 확인합니다.</p> <p>cmcheckconf는 cluster_ascii_file에 의해 지정된 클러스터 구성이나 각 pkg_ascii_file에 의해 명령으로 지정된 패키지 구성 파일을 확인합니다. 클러스터가 이미 구성되어 있으면 cmcheckconf 명령은 이진 구성 파일에 저장된 이전의 구성 정보와 cluster_ascii_file의 구성을 비교하고 변경 내용을 확인합니다. pkg_ascii_file에도 같은 규칙이 적용됩니다. cmcheckconf 명령을 실행하려면 클러스터를 중지해야 합니다.</p> |

표 A-1

MC/Serviceguard 명령(계속)

| 명령 | 설명 |
|--------------|--|
| cmdeleteconf | <p>클러스터 또는 패키지 구성을 삭제합니다.</p> <p>cmdeleteconf는 모든 패키지를 포함한 전체 클러스터 구성을 삭제하거나 지정된 패키지 구성만 삭제합니다. <i>cluster_name</i>이나 <i>package_name</i>을 지정하지 않으면 cmdeleteconf는 로컬 클러스터의 구성과 해당 패키지를 모두 삭제합니다. <i>package_name</i>만 지정하면 로컬 클러스터의 <i>package_name</i> 구성이 삭제됩니다. <i>cluster_name</i>과 <i>package_name</i>을 모두 지정하면 패키지는 <i>cluster_name</i>에 구성되어야 하고 <i>package_name</i> 패키지만 삭제됩니다. 로컬 클러스터는 cmdeleteconf 명령을 실행 중인 노드가 속해 있는 클러스터입니다.</p> |
| cmgetconf | <p>클러스터 또는 패키지 구성 정보를 가져옵니다.</p> <p>cmgetconf는 클러스터 구성(패키지 구성은 제외)이나 지정된 패키지의 구성 정보를 가져와서 <i>output_filename</i> 파일이나 stdout에 기록합니다. 클러스터의 작동 여부에 관계없이 이 명령을 실행할 수 있습니다. <i>cluster_name</i>이나 <i>package_name</i>을 지정하지 않으면 cmgetconf는 로컬 클러스터의 구성 정보를 가져옵니다. <i>cluster_name</i>과 <i>package_name</i>을 모두 지정하면 패키지는 <i>cluster_name</i>에 구성되어야 하고 <i>package_name</i>에 대한 패키지 구성만 <i>output_filename</i> 또는 stdout에 기록됩니다.</p> |

표 A-1

MC/Serviceguard 명령(계속)

| 명령 | 설명 |
|------------|---|
| cmhaltctl | <p>고가용성 클러스터를 중지합니다.</p> <p>cmhaltctl을 사용하면 구성된 클러스터의 모든 노드에서 해당 클러스터 데몬을 중지하고 선택적으로 프로세스의 모든 패키지나 응용 프로그램을 중지합니다.</p> <p>이 명령은 현재 실행 중인 모든 시스템에서 모든 데몬을 중지합니다. 일부 데몬만 종료하려면 대신 cmhaltnode 명령을 사용해야 합니다.</p> |
| cmhaltnode | <p>고가용성 클러스터의 노드를 중지합니다.</p> <p>cmhaltnode를 사용하면 노드에서는 해당 노드의 클러스터 데몬을 중지하고 자신을 기존 클러스터에서 제거합니다.</p> <p>cmhaltnode가 노드에서 실행되면 클러스터 데몬이 중지되며 선택적으로 해당 노드에서 실행 중인 모든 패키지가 다른 노드로 이동합니다.</p> <p><i>node_name</i>을 지정하지 않으면 로컬 노드에서 실행 중인 클러스터 데몬이 중지되고 기존 클러스터에서 제거됩니다.</p> |
| cmhaltpkg | <p>고가용성 패키지를 중지합니다.</p> <p>cmhaltpkg는 Serviceguard 클러스터에서 실행 중인 고가용성 패키지의 수동 중지를 수행합니다. 이 명령은 클러스터의 노드에서 실행될 수 있으며 클러스터의 모든 패키지에 대해 작동할 수 있습니다.</p> |

표 A-1

MC/Serviceguard 명령(계속)

| 명령 | 설명 |
|------------|--|
| cmhaltserv | <p>고가용성 패키지 중지 스크립트에서 서비스를 중지합니다. 이 명령은 명령줄 실행 명령이 아니며 패키지 제어 스크립트에서만 실행됩니다.</p> <p>cmhaltserv는 고가용성 패키지 중지 스크립트에서 서비스를 중지하는 데 사용됩니다. 패키지의 일부가 작동 중단 상태로 표시된 경우 패키지 중지 스크립트는 복구 프로세스의 일부로 실행됩니다.</p> <p>이 명령은 PID와 모니터링되는 서비스의 해당 프로세스 그룹에 SIGTERM 신호를 보냅니다. 실행 중인 응용 프로그램에서 이 신호를 받으면 응용 프로그램이 해당 프로세스의 종료를 담당하게 됩니다.</p> |
| cmmakepkg | <p>고가용성 패키지 템플릿 파일을 만듭니다.</p> <p>cmmakepkg는 선택한 옵션에 지정된 대로 템플릿 ASCII 패키지 구성 파일 또는 패키지 제어 스크립트를 만듭니다.</p> <p>output_file_name은 특정 클러스터 환경에 맞게 사용자 정의해야 합니다. 사용자 정의 후 이 파일은 cmcheckconf 명령으로 확인해야 합니다. output_file_name을 지정하지 않으면 출력은 stdout으로 보내집니다.</p> |

표 A-1

MC/Serviceguard 명령(계속)

| 명령 | 설명 |
|----------|--|
| cmmodnet | <p>고가용성 클러스터에서 주소를 추가하거나 제거합니다.</p> <p>cmmodnet은 고가용성 패키지 제어 스크립트에서 사용되어 주어진 <i>subnet_name</i>이 실행되는 현재 네트워크 인터페이스에서 <i>IP_address</i>를 추가하거나 제거합니다.</p> <p>패키지 제어 스크립트의 컨텍스트 외부에서 이 명령을 실행할 때는 상당히 주의해야 합니다. 이 명령은 장애가 발생하여 “꺼짐” 상태에 있는 변동 가능 IP 주소를 삭제할 때만 사용해야 합니다. 패키지 실행 중에 이 명령을 사용하면 클라이언트 연결이 끊어질 수 있습니다.</p> |
| cmmodpkg | <p>고가용성 패키지의 전환 속성을 활성화하거나 비활성화합니다.</p> <p>cmmodpkg는 패키지에 장애가 발생할 때 패키지가 다른 노드로 전환하는 기능을 활성화하거나 비활성화하고, 실행 중인 특정 패키지에서 특정 노드를 활성화하거나 비활성화합니다. 패키지의 전환 기능은 전역적으로 활성화하거나 비활성화할 수 있습니다. 예를 들어, 전역적으로 비활성화된 패키지에 장애가 발생하면 다른 노드로 전환되지 않으며, 전역적으로 활성화된 패키지에 장애가 발생하면 패키지가 실행되도록 구성된 첫 번째 사용 가능한 노드로 전환이 시도됩니다.</p> |

표 A-1

MC/Serviceguard 명령(계속)

| 명령 | 설명 |
|-----------|--|
| cmquerycl | <p>클러스터 또는 노드 구성 정보를 쿼리합니다.</p> <p>cmquerycl은 지정된 모든 노드에서 클러스터 구성 및 Logical Volume Manager(LVM) 정보를 검색합니다. 클러스터 구성 정보에는 LAN 인터페이스, IP 주소, 브리지 연결 네트워크 및 가능한 하트비트 네트워크와 같은 네트워크 정보가 포함됩니다. LVM 정보에는 볼륨 그룹(VG) 상호 연결 및 파일 시스템 마운트 지점 정보가 포함됩니다. 이 명령은 클러스터 구성 준비의 첫 번째 단계로 실행되어야 합니다. 클러스터의 현재 구성을 확인하기 위한 문제 해결 도구로 이 명령을 사용할 수도 있습니다.</p> |
| cmreadlog | <p>쉽게 알아볼 수 있도록 객체 관리자 로그 파일의 서식을 지정합니다.</p> <p>이 명령은 객체 관리자가 MOF(Managed Object File) 형식으로 만든 로그 파일을 읽어서 한 줄에 한 항목씩 보고서에 표시합니다. 객체 관리자의 동작을 검토하거나 관련 문제를 해결할 때 이 명령을 사용합니다.</p> |
| cmrunc1 | <p>고가용성 클러스터를 실행합니다.</p> <p>cmrunc1을 사용하면 구성된 클러스터의 모든 노드나 지정된 모든 노드에서 클러스터 데몬을 시작하고 새 클러스터를 구성합니다. 이 명령은 구성된 모든 노드에서 클러스터가 활성 상태가 아닌 경우에만 실행해야 합니다. 클러스터가 이미 일부 노드에서 실행 중인 경우에는 cmrunnode 명령을 사용하여 나머지 노드를 시작하고 해당 노드를 기존 클러스터에 강제로 참가시켜야 합니다.</p> |

표 A-1 MC/Serviceguard 명령(계속)

| 명령 | 설명 |
|-----------|--|
| cmrunnode | <p>고가용성 클러스터의 노드를 실행합니다.</p> <p>cmrunnode를 사용하면 노드에서 클러스터 데몬을 시작하여 기존 클러스터에 참가하게 됩니다.</p> <p>노드를 시작해도 활성 패키지가 새 노드로 이동하지는 않습니다. 그러나 패키지의 작동이 중단되었고 해당 패키지의 전환 기능이 활성화되어 있으며 패키지가 새 노드에서 실행될 수 있는 경우에는 패키지가 자동으로 새 노드에서 실행됩니다.</p> |
| cmrunpkg | <p>고가용성 패키지를 실행합니다.</p> <p>cmrunpkg는 이전에 중지된 고가용성 패키지를 실행합니다. 이 명령은 클러스터 내의 모든 노드에서 실행될 수 있으며 클러스터 내의 모든 패키지에 대해 작동할 수 있습니다. 노드를 지정하지 않으면 이 명령이 실행된 노드가 사용됩니다. 따라서 현재 노드가 패키지를 실행할 수 없거나 패키지의 가능한 소유자 목록에 없을 경우에는 오류가 발생합니다. 패키지가 새 노드에서 시작될 때는 패키지의 실행 스크립트가 실행됩니다.</p> |

표 A-1

MC/Serviceguard 명령(계속)

| 명령 | 설명 |
|-----------|--|
| cmrunserv | <p>고가용성 패키지 실행 스크립트에서 서비스를 실행합니다. 이 명령은 명령줄 실행 명령이 아니며 패키지 제어 스크립트에서만 실행됩니다.</p> <p>cmrunserv는 고가용성 패키지 실행 스크립트에서 서비스를 실행하는 데 사용됩니다. 서비스 프로세스가 종료되면 cmrunserv는 서비스의 상태를 중지 상태로 업데이트합니다. 클러스터 소프트웨어에서는 변경된 상태를 인식하고 정상적인 패키지 복구 절차를 실행합니다. 여기에는 패키지 중지 스크립트를 실행하고 패키지가 다른 노드에서 실행될 수 있는지 여부를 확인하고 가능한 경우 새 노드에서 패키지 실행 스크립트를 실행하는 과정이 포함됩니다.</p> <p><i>service_command</i>가 cmhaltserv 명령에 의해 중지된 경우에는 SIGTERM 신호가 해당 프로세스에 보내집니다. 이 실행 스크립트나 셸 스크립트는 SIGTERM 신호를 처리하고 정상적인 종료 절차를 실행하여 필요한 정리 작업을 수행할 수 있어야 합니다. 프로세스에서 SIGTERM을 무시하면 SIGKILL이 프로세스로 보내집니다. SIGKILL이 보내지면 프로세스는 즉시 종료되고 정리 작업을 수행할 수 없게 됩니다.</p> |

표 A-1 MC/Serviceguard 명령(계속)

| 명령 | 설명 |
|----------|--|
| cmsscand | <p>Serviceguard가 설치된 노드에서 시스템 구성 정보를 수집합니다.</p> <p>cmsscand는 노드 목록이나 클러스터의 모든 노드에서 시스템의 소프트웨어 및 하드웨어 구성 정보를 수집하는 구성 보고 및 진단 도구입니다. 이 명령이 표시하는 정보에는 LAN 장치 구성, 네트워크 상태와 인터페이스, 파일 시스템, LVM 구성, 링크 수준 연결 및 이진 클러스터 구성 파일의 데이터가 포함됩니다. 이 명령은 문제 해결 도구나 데이터 수집 도구로 사용될 수 있습니다.</p> <p>output_file을 지정하지 않으면 정보는 stdout으로 보내집니다. 출력 파일에는 다음이 포함됩니다.</p> <ul style="list-style-type: none"> • LAN 장치 구성(lanscan의 출력) • 네트워크 상태 및 인터페이스(netstat의 출력) • 파일 시스템(mount의 출력) • LVM 구성(/etc/lvm/tab 파일의 내용) • LVM 물리 볼륨 그룹 정보(/etc/lvmpvg 파일의 내용) • 링크 수준 연결(linkloop의 출력) • 이진 구성 파일 데이터(cmviewconf의 출력) |

표 A-1

MC/Serviceguard 명령(계속)

| 명령 | 설명 |
|------------|--|
| cmstartres | <p>Serviceguard 패키지에 구성된 EMS 리소스의 로컬 노드에서 리소스 모니터링을 시작합니다.</p> <p>cmstartres는 로컬 노드에서 EMS 리소스에 대한 리소스 모니터링을 시작합니다. 이 리소스는 지정된 <i>package_name</i>에 구성되어 있어야 합니다.</p> |
| cmstopres | <p>Serviceguard 패키지에 구성된 EMS 리소스의 로컬 노드에서 리소스 모니터링을 중지합니다.</p> <p>cmstopres는 로컬 노드에서 EMS 리소스에 대한 리소스 모니터링을 중지합니다. 이 리소스는 지정된 <i>package_name</i>에 구성되어 있어야 합니다.</p> |
| cmviewcl | <p>현재 고가용성 클러스터에 대한 정보를 표시합니다.</p> <p>cmviewcl은 클러스터의 현재 상태 정보를 표시합니다. 전체 클러스터에 대한 출력이 표시되거나 특정 노드나 패키지에 대한 출력만 표시될 수 있습니다.</p> |
| cmviewconf | <p>MC/Serviceguard 또는 Serviceguard 클러스터 구성 정보를 표시합니다.</p> <p>cmviewconf는 기존 클러스터의 이전 구성 파일에서 클러스터 구성 정보를 수집하여 ASCII 형식으로 표시합니다. 선택적으로 파일에 출력을 기록할 수도 있습니다. 이 명령은 클러스터 구성을 확인하기 위한 문제 해결 도구로 사용될 수 있습니다.</p> |

B Enterprise Cluster Master Toolkit

Enterprise Cluster Master Toolkit(ECMT)에는 몇 가지 주요 데이터베이스 및 인터넷 소프트웨어 제품에 대한 Serviceguard 패키지를 만드는 데 이용할 수 있는 예제 스크립트와 패키지 구성 파일 모음이 들어 있습니다. 각각의 도구 키트에는 사용자의 필요에 맞게 패키지를 편집하는 방법을 설명하는 README 파일이 들어 있습니다.

HP-UX 11iv 1용 ECMT,

ECMT는 HP-UX 11i v1(HP 제품 번호 B5139EA) 또는 11i v2(HP 제품 번호 T1909BA)에 설치할 수 있습니다.

ECMT에는 다음 인터넷 응용 프로그램의 도구 키트가 포함됩니다.

- HP Apache
- HP Tomcat
- HP CIFS/909i

ECMT에는 다음 데이터베이스 응용 프로그램의 도구 키트가 포함됩니다.

- Oracle 9i
- Oracle10g
- Informix(11iv 1만 해당)
- Sybase(11iv 1만 해당)
- DB2(11iv 1만 해당)
- Progress(11iv 1만 해당)

별도로 NFS 도구 키트도 제공되고 있습니다. 더 자세한 사항을 알고 싶으면 *Managing Highly Available NFS*(HP 제품 번호 B5140-90017)를 참고하십시오.

HP에서는 이외에도 다른 응용 프로그램을 위한 통합 스크립트를 제공하고 있습니다.

C 고가용성 클러스터 응용 프로그램 설계

이 부록에서는 고가용성을 구현하도록 응용 프로그램을 만들거나 이식하는 방법을 다음과 같은 항목에 중점을 두어 설명합니다.

- 응용 프로그램 작동 자동화
- 응용 프로그램의 장애 조치 속도 제어
- 다중 시스템에서 실행할 응용 프로그램 설계
- 클라이언트 연결 복원
- 응용 프로그램 장애 처리
- 계획된 중단 시간 최소화

고가용성 설계의 목적은 사용자가 경험하게 되는 계획되지 않은 중단 시간과 계획된 중단 시간을 줄이는 것입니다. 계획되지 않은 중단 시간에는 정전, 시스템 장애, 네트워크 장애, 디스크 고장 또는 응용 프로그램 장애와 같이 계획되지 않은 중단이 포함됩니다. 계획된 중단 시간에는 정기적인 백업, 새로운 OS 버전으로의 시스템 업그레이드 또는 하드웨어 교체와 같은 계획된 중단이 포함됩니다.

고가용성 설계 시 염두에 두어야 할 두 가지 핵심 전략은 다음과 같습니다.

1. 시스템의 재부팅 또는 패닉 상태를 처리할 수 있도록 응용 프로그램을 설계합니다. 기존 응용 프로그램을 고가용성 환경에 맞게 수정하려면 시스템 패닉이 발생한 후에 응용 프로그램에 어떤 일이 발생하는지를 확인합니다. 고가용성 환경에서는 이런 경우에 응용 프로그램을 다시 시작하기 위한 정의된(스크립트화된) 프로시저가 있어야 합니다. 응용 프로그램을 시작하고 중지하는 프로시저는 사용자의 개입이 필요하지 않도록 자동으로 수행되어야 합니다.
2. 장애 시 응용 프로그램이 다른 시스템으로 전달되어 정상적으로 실행되도록 하려면 프로그램에서 다음과 같은 시스템 관련 정보를 사용하면 안 됩니다.
 - 응용 프로그램에서 `uname()` 또는 `gethostname()` 을 참조해서는 안 됩니다.
 - 응용 프로그램에서 `SPU ID` 를 참조해서는 안 됩니다.
 - 응용 프로그램에서 `MAC 주소` (링크 수준 주소)를 참조해서는 안 됩니다.

응용 프로그램 작동 자동화

응용 프로그램이 자동으로 시작되고 중지될 수 있습니까? 아니면 사용자의 개입이 필요합니까?

이 절에서는 사용자 개입이 필요 없도록 응용 프로그램의 작동을 자동화하는 방법에 대해 설명합니다. 고가용성의 첫 번째 원칙 중 하나는 수동 개입을 최소화하는 것입니다. 사용자가 터미널, 콘솔 또는 GUI 인터페이스로 명령을 입력해야만 하위 시스템을 시작할 수 있다면 사용자가 시스템의 핵심 요소가 됩니다. 필요한 작업을 수행하기 위해 사용자가 시스템 콘솔 앞에 앉기까지는 몇 시간이 걸릴 수도 있습니다. 문제가 발생한 하드웨어가 멀리 떨어져 있고 훈련된 직원이 없는 장소에 있을 수도 있습니다. 또는 시스템이 접근이 통제된 데이터 센터에 있을 수도 있으며, 업무 시간 이후에는 모뎀을 통해 시스템에 접속해야 할 수도 있습니다.

응용 프로그램 재배포를 자동화하려면 다음 두 가지 원칙을 염두에 두어야 합니다.

- 사용자의 개입이 최소화되어야 합니다.
- 응용 프로그램의 시작 및 종료 프로시저가 정의되어야 합니다.

응용 프로그램이 실행되고 있는 시스템이 재부팅되었을 때 어떤 일이 발생하는지 파악해야 하며, 고가용성을 위한 응용 프로그램의 대응 방식에서 변경해야 할 사항이 있는지를 결정해야 합니다.

사용자 개입 최소화

가능하면 일반 사용자의 개입을 최소화하십시오. 여기에는 다음 사항이 포함됩니다.

- 서버의 장애로 연결이 끊어졌을 때 다시 연결하기 위해 사용자의 개입이 필요하면 안 됩니다.
- 가능하면 사용자에게 장애 조치로 인해 약간의 지연이 있음을 알립니다.
- 데이터 재입력을 최소화합니다.
- 사용자가 느끼는 성능 저하를 최소화할 수 있도록 시스템 용량에 충분한 여유를 두고 설계합니다.

응용 프로그램 시작 및 종료 정의

응용 프로그램은 수동 개입 없이 다시 시작될 수 있어야 합니다. 하드웨어 장치에서 직접 스위치를 켜야 하는 응용 프로그램에서는 재시작을 자동화하는 것이 불가능합니다. HA 소프트웨어에서 응용 프로그램 시작, 종료 및 모니터링을 자동으로 수행하도록 하려면 이러한 작업을 위한 프로시저를 만들어야 합니다.

자동화된 대응을 위해서는 응용 프로그램을 시작하고 중지하기 위한 프로시저가 정의되어야 합니다. Serviceguard에서 이러한 프로시저는 패키지 제어 스크립트에 기록되어 있습니다. 이러한 프로시저에서는 오류를 검사하고 HA 제어 소프트웨어에 현재 상태를 전달해야 합니다. 시작과 종료는 명령줄을 통해 수행되어야 하며, 모든 응답을 사전에 지정했거나 스크립트로 처리할 수 있는 경우가 아니면 대화식으로 수행되어서는 안 됩니다.

HA 장애 조치 환경에서 HA 소프트웨어는 필요한 디스크 드라이브에 대한 액세스 권한과 같이 필요한 리소스가 있는 클러스터의 사용 가능한 시스템에서 응용 프로그램을 다시 시작합니다. 응용 프로그램은 다음과 같은 두 가지 측면에서 다시 시작될 수 있어야 합니다.

- 응용 프로그램은 백업 시스템이나 동일한 시스템(응용 프로그램 재시작 옵션이 선택된 경우)에서 다시 시작되어 복구될 수 있어야 합니다.
- 응용 프로그램은 시작 중 장애가 발생할 경우 장애 원인이 해결되면 다시 시작될 수 있어야 합니다.

응용 프로그램 관리자는 적절한 HA 명령을 사용하여 응용 프로그램을 시작하고 종료하는 방법을 숙지해야 합니다. 응용 프로그램 종료 방법이 올바르지 않으면 원하지 않는 장애 조치가 시작될 수 있습니다. 또한, 응용 프로그램 관리자는 개발 환경에서 실수로 응용 프로그램의 테스트 인스턴스가 아니라 프로덕션 인스턴스를 종료하지 않도록 주의해야 합니다.

응용 프로그램에 장애가 발생한 시간을 HA 소프트웨어에서 알 수 있도록 응용 프로그램의 활성 여부를 모니터링하는 기능이 필요합니다. 이 기능은 응용 프로그램의 모든 프로세스에 대해 `ps -ef | grep xxx` 명령을 실행하는 간단한 스크립트로 구현할 수 있습니다.

응용 프로그램을 중단하게 되면 백업 시스템으로 불필요한 장애 조치가 수행되기 때문에, 사용자에게 미치는 영향을 최소화하기 위해서는 오류가 발생했을 때 응용 프로그램을 단순히 중단하면 안 됩니다. 이때는 응용 프로그램을 단순히 중단하지 말고 정확한 오류를 파악하여 복구하기 위한 조치를 취해야 합니다.

응용 프로그램의 장애 조치 속도 제어

신속한 장애 조치를 위해 수행해야 하는 절차는 무엇입니까?

장애로 인해 응용 프로그램이 다른 노드로 이동(장애 조치)되는 경우 응용 프로그램이 백업되고 다시 실행되기까지 걸리는 시간을 줄일 수 있는 방법에는 여러 가지가 있습니다. 다음 절에서는 이러한 방법에 대해 설명합니다.

- 비 데이터 파일 시스템 복제
- 원시 볼륨 사용
- JFS 사용 검토
- 데이터 손실 최소화
- 재시작 가능한 트랜잭션 사용
- 검사점 사용
- 다중 서버의 설계
- 복제된 데이터 지점의 설계

비 데이터 파일 시스템 복제

비 데이터 파일 시스템은 공유하지 말고 복제해야 합니다. 응용 프로그램 데이터의 사본은 하나만 있을 수 있습니다. 데이터는 응용 프로그램을 실행하는 시스템에서 액세스하는 디스크 집합에 보관됩니다. 이러한 데이터 디스크가 파일 시스템이면 장애 조치 후 파일 시스템 복구(fsck)를 수행해야 데이터에 액세스할 수 있습니다. 데이터 파일 시스템의 크기가 작을수록 복구가 신속히 수행되므로 파일 시스템 크기를 줄여 복구 시간을 줄일 수 있습니다. 따라서 복제할 수 있는 것을 모두 데이터 파일 시스템 외부에 유지하는 것이 가장 좋습니다. 예를 들어, 응용 프로그램 실행 파일의 사본 하나를 공유 파일 시스템에 두는 것보다는 각 시스템에 실행 파일의 사본을 하나씩 두어야 합니다. 또한, 응용 프로그램 실행 파일을 복제하면 필요할 때 응용 프로그램을 롤링 업그레이드할 수 있습니다.

원시 볼륨 사용

응용 프로그램에서 데이터를 사용하는 경우에는 파일 시스템 대신 원시 볼륨을 사용하십시오. 원시 볼륨에는 파일 시스템의 fsck가 필요하지 않으므로 장애 조치 중 시간이 많이 걸릴 수 있는 단계 하나가 줄어듭니다.

JFS 사용 검토

파일 시스템을 사용해야 하는 경우에는 HFS에 비해 파일 시스템 복구 속도가 훨씬 빠른 JFS를 사용하는 것이 좋습니다. 그러나 JFS의 성능은 응용 프로그램에 따라 차이가 날 수 있습니다.

데이터 손실 최소화

계획되지 않은 중지가 발생할 경우에 손실될 수 있는 데이터의 양을 최소화하십시오. 장애 발생 시 일부 데이터가 손실되는 것은 피할 수 없습니다. 하지만 손실되는 데이터의 양을 최소화할 수 있도록 노력해야 합니다. 자세한 내용은 다음 절에서 설명합니다.

메모리 기반 데이터의 크기 및 사용 최소화

장애가 발생하면 모든 메모리 내장 데이터(메모리 내장 컨텍스트)가 손실됩니다. 데이터를 쉽게 다시 계산할 수 있는 경우가 아니면 응용 프로그램은 메모리 내장 데이터의 크기를 최소화하도록 설계되어야 합니다. 응용 프로그램은 대기 노드에서 다시 시작될 때 메모리에 있어야 하는 정보를 다시 계산하거나 디스크에서 다시 읽어야 합니다.

장애 조치 속도를 측정하는 한 가지 방법은 응용 프로그램이 일반 시스템에서 재부팅 후 시작되는 데 걸리는 시간을 측정하는 것입니다. 응용 프로그램이 즉시 시작됩니까? 아니면, 응용 프로그램에서 여러 단계를 거쳐야만 일반 사용자가 응용 프로그램에 연결할 수 있습니까? 이상적인 경우라면 응용 프로그램은 메모리 내장 데이터 구조나 테이블을 다시 초기화할 필요 없이 신속하게 시작될 수 있습니다.

성능을 높이려면 데이터를 디스크가 아닌 메모리에 유지해야 합니다. 그러나 데이터를 디스크에 유지할 때 저하되는 성능 문제만큼 메모리에 데이터를 유지할 때 발생할 수 있는 데이터 손실 문제도 고려해야 합니다.

공유 디스크에서 메모리로 가져와서 읽기 전용으로 사용되는 데이터는 손실에 대해 걱정할 필요가 없습니다.

로그 크기 최소화

일부 데이터베이스에서는 온라인 성능을 향상시키기 위해 로그를 메모리에 버퍼링합니다. 물론, 장애가 발생하면 진행 중인 트랜잭션은 손실됩니다. 그러나 메모리에 있는 로그 크기를 최소화하면 장애 발생 시 완료된 트랜잭션 데이터가 손실되는 양을 줄일 수 있습니다.

디스크에 유지하는 로그의 크기를 최소화하면 로그가 더 자주 보관되거나 복제되므로 장애가 발생했을 때 데이터가 손실될 위험이 줄어듭니다. 물론, 로그의 크기를 줄이면 대신 온라인 성능이 저하된다는 점을 고려해야 합니다.

로컬 데이터의 필요성 최소화

가능하면 로컬 데이터의 필요성을 최소화하십시오. 3계층 클라이언트/서버 환경에서 중간 계층에는 데이터가 없는 경우가 많습니다. 즉, 클라이언트와 관련이 있거나 클라이언트에서 수정해야 하는 로컬 데이터가 없습니다. 이 “응용 프로그램 서버” 계층에서는 더 높은 수준의 가용성, 로드 균형 조정 및 장애 조치를 제공할 수 있습니다. 하지만 이 경우 모든 데이터가 클라이언트(계층 1)나 데이터베이스 서버(계층 3)에 저장되어야 합니다.

재시작 가능한 트랜잭션 사용

서버에 장애가 발생하여 응용 프로그램이 다른 시스템에서 다시 시작되는 경우 클라이언트가 트랜잭션을 다시 시작하거나 처음부터 다시 시작하지 않도록 하기 위해 트랜잭션은 다시 시작될 수 있어야 합니다. 즉, 트랜잭션 도중에 장애가 발생해도 처음부터 다시 시작하지 않도록 해야 합니다. 이 기능은 응용 프로그램을 더욱 안정적으로 만들며 장애 조치 시 사용자에게 미치는 영향을 줄여 줍니다.

일반적인 예로 인쇄 작업이 있습니다. 프린터 응용 프로그램은 일반적으로 작업 스케줄러를 사용합니다. 스케줄러는 한 작업이 완료되면 다음 작업으로 진행합니다. 그러나 3시간 동안 급여 명세서를 인쇄하는 경우와 같이 오래 걸리는 작업 도중에 시스템이 중지되었다가 다시 정상으로 돌아온 경우, 작업을 처음부터 다시 시작하여 급여 명세표를 모두 다시 인쇄하거나, 작업이 중단된 부분부터 다시 시작하거나, 작업 스케줄러가 작업이 완료된 것으로 간주하고 남은 시간의 인쇄 작업을 수행하지 않을 수 있습니다. 고가용성 환경에서 올바른 동작은 중단된 부분에서 인쇄를 다시 시작하여 모두가 정확하게 한 장의 급여 명세표만 받을 수 있도록 하는 것입니다.

또 다른 예로 신입 사원에 대한 데이터를 입력하는 응용 프로그램이 있습니다. 고유한 직원 번호가 필요한 이 응용 프로그램에서 신입 사원의 이름과 번호를 입력하는 도중에 장애가 발생한 경우 직원 번호를 이미 입력한 후에 장애가 발생했다면 응용 프로그램에 이 데이터를 다시 입력할 필요가 없거나, 부분적으로 입력된 정보를 먼저 삭제해야 할 수 있습니다. 고가용성 환경의 응용 프로그램에서는 해당 항목을 쉽게 다시 입력하거나 다음 데이터 항목에서 입력을 계속할 수 있도록 합니다.

검사점 사용

복잡한 트랜잭션에는 검사점을 사용하도록 응용 프로그램을 설계하십시오. 사용자 관점에서 단일 트랜잭션은 여러 개의 실제 데이터베이스 트랜잭션으로 나눌 수 있습니다. 이 문제는 재시작 가능한 트랜잭션과 관련되지만 여기에서는 시스템 장애로 중단된 트랜잭션을 장애 조치 후 완료할 수 있도록 클라이언트에서 로컬로 진행 상황을 기록하는 방법만 설명합니다.

예를 들어, PI를 계산하는 응용 프로그램을 가정합니다. 원래 시스템에서 이 응용 프로그램이 소수점 1000자리까지 계산했지만 디스크에는 아직 아무 것도 기록하지 않은 상태에서 해당 노드에서 장애가 발생하면 응용 프로그램이 두 번째 노드에서 다시 시작되는 하지만 처음부터 다시 시작됩니다. 따라서 응용 프로그램은 소수점 1000자리까지 다시 계산해야 합니다. 하지만 응용 프로그램이 주기적으로 디스크에 소수점을 기록했다면 응용 프로그램은 중단된 부분에서 다시 시작할 수 있습니다.

검사점 수와 성능의 균형

검사점 수와 성능의 균형을 맞추는 것이 중요합니다. 디스크의 검사점 수와 성능은 반비례합니다. 검사점이 너무 많으면 응용 프로그램이 느려지며 검사점이 충분하지 않으면 장애 조치 후 응용 프로그램이 현재 상태로 돌아오는 데 시간이 오래 걸립니다. 이상적인 방법은 일반 사용자가 검사점 수를 결정할 수 있게 하는 것입니다. 응용 프로그램은 사용자가 정의할 수 있는 매개 변수를 제공하여 직접 검사점 수를 조정할 수 있게 해야 합니다.

다중 서버의 설계

활성 서버를 여러 개 사용하는 경우 여러 서비스 지점에서 클라이언트에 상대적으로 투명한 서비스를 제공할 수 있습니다. 그러나 이렇게 하려면 클라이언트가 복수 서버와 서버의 처리 우선 순위에 대해 잘 알고 있어야 합니다. 또한 장애가 있는 서버의 데이터나 복제된 데이터에 액세스할 수 있어야 합니다.

예를 들어, 두 번째 시스템으로 장애 조치되는 하나의 응용 프로그램을 사용하는 것보다 두 시스템에서 응용 프로그램을 실행하는 것을 고려할 수 있습니다. 첫 번째 시스템에서 장애가 발생하면 두 번째 시스템이 첫 번째 시스템의 작업을 넘겨받으면 됩니다. 이렇게 하면 응용 프로그램의 시작 시간이 필요하지 않게 됩니다. 이러한 구조를 설계하는 방법은 여러 가지가 있으며 이와 관련된 많은 내용이 있습니다. 여기에서는 자세히 설명하지 않고 몇 가지 예제를 들어 설명합니다.

가장 단순한 방법은 두 응용 프로그램을 마스터/슬레이브 관계로 실행하는 것입니다. 여기서 슬레이브는 마스터에 대한 활성 대기 응용 프로그램입니다. 마스터에 장애가 발생하면 두 번째 시스템의 슬레이브는 데이터의 최종 상태(즉, 데이터 복구 가능 여부 등)를 알아내야 합니다. 그러나 응용 프로그램을 분기하여 처음 시작하는 시간은 절약됩니다.

또 다른 방법은 두 응용 프로그램을 모두 활성화하는 것입니다. 예를 들어, 데이터베이스에 데이터를 제공하는 응용 프로그램 서버가 두 대 있는 경우 클라이언트의 절반은 첫 번째 응용 프로그램 서버에 연결되고 나머지 절반은 두 번째 응용 프로그램 서버에 연결됩니다. 한 서버에서 장애가 발생하면 모든 클라이언트가 남아 있는 응용 프로그램 서버에 연결됩니다.

복제된 데이터 지점의 설계

복제된 데이터 지점은 신속한 장애 조치 및 손상 복구에 도움이 됩니다. 데이터를 복제하면 디스크는 시스템 사이에서 공유되지 **않습니다**. 따라서 수행해야 하는 데이터 복구가 없습니다. 또한, 복구도 신속하게 수행됩니다. 그러나 데이터 복제는 성능과 반비례합니다. 데이터 복제를 수행하는 방법에는 여러 가지가 있으며, 응용 프로그램 설계자는 모든 방법을 충분히 검토해야 합니다.

많은 표준 데이터베이스 제품이 클라이언트 응용 프로그램에 투명한 데이터 복제를 제공합니다. 표준 데이터베이스를 사용하도록 응용 프로그램을 설계하면 일반 사용자가 데이터 복제의 사용 여부를 결정할 수 있습니다.

다중 시스템에서 실행할 응용 프로그램 설계

응용 프로그램이 백업 노드로 장애 조치될 수 있으면 다른 시스템에서 응용 프로그램은 어떻게 작동될까요?

이전 절에서는 응용 프로그램을 자동으로 다시 시작하는 방법에 대해 설명했습니다. 이 절에서는 응용 프로그램을 다중 시스템에서 실행할 수 있는 방법에 대해 설명합니다. 이 장에서 다루는 내용은 다음과 같습니다.

- 노드에 특정한 정보 최소화
- 응용 프로그램에 고유 이름 지정
- Uname(2) 사용 시 주의점
- 고정 포트에 바인딩
- 변동 가능 IP 주소에 바인딩
- 응용 프로그램마다 고유한 볼륨 그룹 부여
- SNA 응용 프로그램을 위한 복수 목적지 사용
- 파일 잠금 방지

노드에 특정한 정보 최소화

일반적으로 새 시스템을 설치하면 모든 활성 네트워크 인터페이스에 IP 주소를 할당해야 합니다. 이 IP 주소는 항상 노드와 연관되며 **고정 IP** 주소라고 합니다.

고가용성 응용 프로그램을 포함하는 패키지를 사용하면 응용 프로그램에 할당할 추가적인 IP 주소 집합이 필요합니다. 이 주소를 **변동 가능한 응용 프로그램 IP** 주소라고 합니다. Serviceguard의 네트워크 센서는 이 변동 가능한 응용 프로그램 IP 주소가 있는 서버넷에 액세스하는 노드를 모니터링합니다. Serviceguard에서 패키지가 구성되면 연관된 서버넷 주소가 패키지 의존 주소로 지정되며 패키지가 실행될 수 있는 노드의 목록도 제공됩니다. 장애 조치를 위해 패키지를 원격 노드로 이동할 때는 대상 노드에서 서버넷이 이미 활성화되어 있어야 합니다.

각 응용 프로그램이나 패키지에는 고유한 이름과 변동 가능 IP 주소를 부여해야 합니다. 이렇게 하면 응용 프로그램과 응용 프로그램이 실행되는 시스템이 분리되므로 응용 프로그램이 어떤 시스템에서 실행되고 있는지 알아야 할 필요가 없어집니다. 또한, 로드 균형 조정 등의 목적으로 응용 프로그램을 클러스터에 있는 다른 시스템으로 이동하

는 작업도 더 쉬워집니다. 두 응용 프로그램이 단일 IP 주소를 공유하는 경우에는 함께 이동해야 합니다. 대신 독립적인 이름과 주소를 사용하면 두 응용 프로그램이 개별적으로 이동할 수 있습니다.

외부에서 클러스터에 액세스할 때 클라이언트는 응용 프로그램을 참조하는 방법을 알아야 합니다. 한 가지 방법은 클라이언트에게 응용 프로그램과 연관된 변동 가능 IP 주소를 알려주는 것입니다. 다른 방법은 응용 프로그램의 이름을 호스트 이름처럼 취급하고 DNS(Domain Name System)의 이름-주소 매핑을 구성하는 것입니다. 두 방법 모두에서 클라이언트는 궁극적으로 응용 프로그램의 변동 가능 IP 주소를 사용하여 통신합니다. 응용 프로그램이 다른 노드로 이동하게 되면 IP주소도 함께 이동하므로 클라이언트는 응용 프로그램의 현재 위치를 몰라도 응용 프로그램을 사용할 수 있습니다. 각 네트워크 인터페이스에는 고정 IP 주소가 할당되어야 합니다. 네트워크 장애가 발생했을 때 이 IP 주소는 원격 시스템으로 이동하지 **않습니다**.

충분한 IP 주소 확보

시스템 자체에 할당되는 고정 IP와는 별도로 각 응용 프로그램은 **변동 가능 IP** 주소를 받습니다. 따라서 한 시스템에는 자체 IP 주소 하나와 시스템에서 실행되는 각 응용 프로그램마다 하나씩 있는 여러 개의 IP 주소가 있을 수 있습니다. 따라서 주어진 서브넷 범위의 IP 주소는 고가용성 기능이 없는 시스템보다 빨리 소비됩니다. 이 경우 추가 IP 주소를 얻어야 할 수도 있습니다.

하나의 네트워크 인터페이스에 여러 IP 주소를 사용하려면 이 주소들이 모두 동일한 서브넷에 있어야 합니다.

동일한 시스템에서 다중 인스턴스 허용

응용 프로그램은 서로 다른 응용 프로그램 이름과 IP 주소를 갖는 복수의 인스턴스가 하나의 시스템에서 실행될 수 있도록 작성되어야 합니다. 응용 프로그램을 실행할 때 현재 실행 중인 인스턴스를 보여 주는 매개 변수를 사용해야 할 수도 있습니다. 이렇게 하면 정상적인 상황에서 여러 시스템으로 사용자를 분산할 수 있을 뿐만 아니라 한 시스템에서 장애가 발생해도 모든 사용자가 서비스를 받을 수 있습니다.

SPU ID 또는 MAC 주소 사용 최소화

SPU ID나 MAC 주소(링크 수준 주소)에 의존하지 않도록 응용 프로그램을 설계하십시오. SPU ID는 비휘발성 메모리에 저장되는 고유 하드웨어 ID로서 변경이 불가능합니다. MAC 주소(LANIC ID라고도 함)는 LAN 하드웨어와 연관된 링크 특정 주소입니다. 이 주소를 라이선스 서버에서 사용할 때는 공통적인 문제가 있습니다. 보안상의 이유로 라이선스 서버는 하드웨어에 특정한 ID를 사용하여 라이선스가 여러 노드로 복사될 수 없도록 합니다. 한 가지 해결 방법은 응용 프로그램이 실행되는 노드마다 하나의 라이선스를 획득하는 것입니다. 또 다른 방법은 SPU ID나 노드 이름의 집합을 나열하는 클러스터 범위의 메커니즘을 사용하는 것입니다. 응용 프로그램이 지정된 집합의 시스템에서 실행되고 있으면 라이선스가 승인됩니다.

이전 세대의 HA 소프트웨어는 서비스가 백업 시스템으로 이동할 때 네트워크 카드의 MAC 주소를 IP 주소와 함께 이동했습니다. 이 작업은 Serviceguard에서 더 이상 허용되지 않습니다.

MAC 주소를 사용해야 하는 몇 가지 이유는 다음과 같습니다.

- 라우터처럼 송신측과 수신측 사이에 있는 기존의 네트워크 장치는 MAC 주소와 IP 주소 쌍을 사용하여 수동으로 프로그래밍되어야 합니다. 이 문제의 해결 방법은 장애 조치 시 MAC 주소도 IP 주소와 함께 이동하는 것입니다.
- 시스템 작동 중지와 연관된 시간 제한으로 인해 네트워크 장치 캐시가 업데이트되는 동안 최대 20분까지 지연될 수 있습니다. 현재의 HA 소프트웨어에서 이 문제는 이전 IP 주소의 새로운 ARP 변환을 새 MAC 주소와 함께 브로드캐스트하여 처리됩니다.

응용 프로그램에 고유 이름 지정

각 응용 프로그램에 고유한 이름을 지정해야 합니다. 이 이름은 다음에 설명하는 것처럼 `gethostbyname()`의 입력으로 사용될 수 있도록 DNS에서 구성되어야 합니다.

DNS 사용

DNS는 호스트 이름과 IP 주소를 서로 매핑하는 데 사용할 수 있는 API를 제공합니다. 이것은 먼저 대상 시스템 이름을 알 수 있는 telnet과 같은 BSD 소켓 응용 프로그램에 유용합니다. 이런 응용 프로그램은 이름을 IP 주소로 매핑한 다음 연결할 수 있습니다. 그러나 몇 가지 주의해야 할 사항이 있습니다.

응용 프로그램은 정식 호스트 이름이나 IP 주소를 참조하면 **안 됩니다**. 정식 호스트 이름과 호스트 이름의 해당 IP 주소는 기본 LAN 카드와 이 카드의 **고정 IP 주소**를 참조합니다. 따라서 호스트 이름이나 기본 IP 주소를 참조하거나 필요로 하는 응용 프로그램은 특정 응용 프로그램을 지원하는 시스템의 네트워크 ID가 한 시스템에서 다른 시스템으로 이동하지만 호스트 이름은 이동하지 않는 고가용성 환경에서 작동하지 않을 수 있습니다.

이러한 문제를 확인하는 한 가지 방법은 응용 프로그램에서 `gethostname(2)` 호출을 검사하는 것입니다. 응용 프로그램이 이동할 때 응답이 변경될 수 있으므로 HA 서비스에서 `gethostname()` 을 사용할 때는 주의해야 합니다. 같은 이유로 `gethostname()` 을 사용하여 `gethostbyname(2)` 호출에 사용할 이름을 확인하는 응용 프로그램도 피해야 합니다. 또한, `gethostbyaddr()` 를 고정 IP 주소로 호출하면 시간이 지남에 따라 응답이 달라질 수 있습니다.

응용 프로그램은 항상 호스트 이름과 고정 IP 주소가 아닌 응용 프로그램 이름과 변동 가능 IP 주소를 참조해야 합니다. 응용 프로그램에서는 호스트 이름이 아니라 응용 프로그램 이름으로 `gethostbyname(2)` 를 호출하는 것이 바람직합니다.

`gethostbyname(2)` 는 응용 프로그램의 IP 주소로 전달됩니다. 이 IP 주소는 응용 프로그램과 함께 새 노드로 이동할 수 있습니다.

그러나 DNS에 응용 프로그램 이름이 구성되어 있는 경우에만 `gethostbyname(2)` 를 사용하여 응용 프로그램의 IP 주소를 찾아야 합니다. 각각의 독립된 HA 서비스에 서로 다른 응용 프로그램 이름을 연관시키는 것이 가장 바람직할 수 있습니다. 이렇게 하면 각 응용 프로그램과 응용 프로그램의 IP 주소가 다른 응용 프로그램에 영향을 미치지 않으면서 다른 노드로 이동할 수 있습니다. DNS의 호스트 이름에는 고정 IP 주소만 연관되어야 합니다.

uname(2) 사용 시 주의점

앞 절에서는 호스트 이름과 관련된 내용 중에 응용 프로그램에서 정식 시스템 이름을 보여 주는 `uname(2)` 를 사용하는 방법에 대해 설명했습니다. 시스템 이름은 시스템의 LAN 카드 수에 관계없이 특정 시스템에 대해 고유합니다. 기본적으로 `uname` 과 `hostname` 은 동일하지만 반드시 같아야 하는 것은 아닙니다. 보안상의 이유로 일부 응용 프로그램에서는 시스템에 연결한 다음 `uname(2)` 를 호출하여 올바른 시스템에 연결되었는지 확인할 수도 있습니다. 이러한 응용 프로그램은 HA 환경에서는 적합하지 않습니다. HA 환경에서는 서비스가 한 시스템에서 다른 시스템으로 이동하지만 `uname`

이나 hostname은 이동하지 않기 때문입니다. 응용 프로그램에서는 실행되고 있는 위치를 확인하기 위한 다른 방법을 찾아야 합니다. 예를 들어, 응용 프로그램은 구성 파일에서 제공되는 호스트 이름 목록을 확인할 수 있습니다.

고정 포트에 바인딩

소켓을 바인딩할 때 포트 주소를 지정하거나 포트를 동적으로 할당할 수 있습니다. 임의의 포트에 바인딩할 때의 한 가지 문제는 나중에 응용 프로그램이 다른 클러스터 노드에서 다시 시작되었을 때 다른 포트가 할당될 수 있다는 점입니다. 이것은 응용 프로그램에 액세스하는 클라이언트를 혼란스럽게 할 수 있습니다.

권장되는 방법은 포트 번호를 동적으로 할당하는 대신 응용 프로그램이 실행될 모든 노드에서 동일한 고정 포트를 사용하는 것입니다. 그러면 응용 프로그램은 현재 실행 중인 노드에 관계없이 항상 같은 포트 번호를 반환합니다. 응용 프로그램 포트 할당 정보는 `/etc/services`에 있어야 추적이 가능하고 다른 사람이 동일한 포트를 선택하지 않도록 할 수 있습니다.

변동 가능 IP 주소에 바인딩

소켓이 바인딩될 때 포트 번호와 함께 IP 주소가 지정됩니다. 이 주소는 통신에 사용할 IP 주소를 나타내며 응용 프로그램이 클라이언트와 통신할 수 있는 인터페이스를 제한할 수 있도록 합니다. 응용 프로그램은 어떤 인터페이스로도 메시지가 도착할 수 있음을 나타내는 `INADDR_ANY`에 바인딩할 수 있습니다.

네트워크 응용 프로그램은 고정 IP 주소, 변동 가능 IP 주소 또는 `INADDR_ANY`에 바인딩할 수 있습니다. 고정 IP 주소를 지정하면 응용 프로그램이 다른 노드에서 다시 시작될 때 고정 IP 주소가 새 시스템으로 이동하지 않기 때문에 응용 프로그램이 작동하지 않을 수 있습니다. 응용 프로그램을 변동 가능 IP 주소에 바인딩하면 다른 시스템으로 이동했을 때 응용 프로그램이 올바르게 작동합니다.

많은 서버 형식의 응용 프로그램은 `INADDR_ANY`에 바인딩하므로 어떤 인터페이스로도 요청을 받을 수 있습니다. 따라서 클라이언트는 고정 IP 주소나 변동 가능 IP 주소로 전송할 수 있습니다. 그러나, 이 경우 네트워킹 코드는 응답에 가장 적합한 송신자 IP 주소를 찾기 못하기 때문에 항상 고정 IP 주소를 선택합니다.

TCP 스트림 소켓의 경우 TCP가 연결 기반 프로토콜이기 때문에 프로토콜 스택의 TCP 수준에서 클라이언트에 대한 이 문제를 해결합니다. 클라이언트에서 TCP는 고정 IP 주소를 무시하고 클라이언트가 원래 사용하던 이전에 바인딩된 변동 가능 IP 주소를 계속 사용합니다.

하지만 UDP 데이터그램 소켓을 사용하면 문제가 있습니다. 클라이언트는 변동 가능 IP 주소를 사용하는 여러 서버에 연결되고 서버 응답 메시지에 있는 송신자 IP 주소를 기초로 응답을 정렬합니다. 그러나 이 응답의 송신자 IP 주소는 변동 가능한 응용 프로그램 IP 주소가 아니라 고정 IP 주소가 됩니다. 따라서 수신을 위해 UDP 소켓을 구성할 때 응용 프로그램은 INADDR_ANY가 아니라 적절한 변동 가능한 응용 프로그램 IP 주소를 사용하여 bind(2)를 항상 호출해야 합니다.

위에서 권장한 방식으로 응용 프로그램을 수정할 수 없는 경우 이 문제를 해결할 수 있는 방법은 주어진 LAN 카드에서 고정 IP 주소를 전혀 사용하지 않고 하나의 변동 가능한 응용 프로그램 IP 주소만 사용하는 것입니다. 이 방법의 한계는 다음과 같습니다.

- 로컬 LAN 장애 조치가 작동하지 않습니다.
- 장애 발생 시 변동 가능 응용 프로그램 IP 주소를 재배치하는 데 사용되는 각 백업 노드에 대기 LAN 카드가 하나씩 있어야 합니다.

Connect() 전에 bind() 호출

응용 프로그램은 연결을 초기화할 때 먼저 bind(2)를 호출하여 connect(2)를 호출하기 전에 응용 프로그램의 IP 주소를 지정해야 합니다. 그렇지 않으면 연결 요청이 변동 가능한 응용 프로그램 IP 주소가 아닌 시스템 아웃바운드 LAN 인터페이스의 고정 IP 주소를 사용하여 전송됩니다. 클라이언트는 accept(2) 호출에서 이 IP 주소를 받으며, 이로 인해 클라이언트 소프트웨어를 혼동시켜 정상적으로 작동하지 않을 수 있습니다.

응용 프로그램마다 고유한 볼륨 그룹 부여

데이터를 사용하는 각 응용 프로그램에 별도의 볼륨 그룹을 사용하십시오. 디스크를 사용하지 않는 응용 프로그램에는 별도의 볼륨 그룹을 할당하지 않아도 됩니다. 볼륨 그룹(디스크 그룹)은 노드 사이에서 이동할 수 있는 저장 장치의 단위입니다. 각 응용 프로그램을 자체 볼륨 그룹으로 제한하면(즉, 두 응용 프로그램이 동일 디스크 드라이브 집합을 공유하지 않으면) 로드 균형 조정에 대한 유연성이 높아집니다. 두 응용 프로그램이 동일한 볼륨 그룹을 사용하여 데이터를 저장하면 두 응용 프로그램을 함께 이동해야 합니다. 두 응용 프로그램의 데이터가 별도의 볼륨 그룹에 저장되면 장애 조치 시 서로 다른 노드로 전환될 수 있습니다.

응용 프로그램 데이터는 서로 다른 디스크 드라이브에 설정되어야 하며, 가능한 경우에는 서로 다른 마운트 지점에 설정되어야 합니다. 응용 프로그램은 서로 다른 디스크 및 별도의 마운트 지점을 허용하도록 설계되어야 합니다. 가능하면 응용 프로그램은 특정 마운트 지점을 가정하지 않아야 합니다.

한 노드가 실수로 다른 노드의 응용 프로그램이 사용하는 디스크에 액세스하는 것을 방지하기 위해 HA 소프트웨어는 단독 액세스 메커니즘을 사용하여 한 번에 하나의 노드만 액세스하도록 합니다. 이 단독 액세스는 볼륨 그룹에 전체적으로 적용됩니다.

SNA 응용 프로그램을 위한 복수 목적지 사용

SNA는 지점 간 연결을 중심으로 합니다. 즉, 시스템에는 메인프레임에서 시작된 다른 지점 간 연결이 있으므로 서비스가 다른 시스템으로 이동할 수 없습니다. 따라서 SNA가 단순 장애 요인(Single Point of Failure)이 되지 않도록 노드의 백업 링크나 다른 노드의 백업 링크가 구성되어야 합니다. SNA 링크를 위한 구성은 한 번에 하나만 활성화될 수 있습니다. 따라서 다른 용도로 사용되는 백업 링크는 장애 조치 시 주요 업무 목적을 위해 재구성되어야 합니다.

파일 잠금 방지

NFS 환경에서 응용 프로그램은 잠길 파일이 NFS 서버에 위치하는 파일 잠금 메커니즘을 사용하지 않아야 합니다. 그리고 로컬 시스템 및 원격 시스템의 응용 프로그램 모두가 파일 잠금을 사용하지 않아야 합니다. 로컬 파일 잠금을 사용하면 백업 시스템의 역할을 하는 시스템은 장애가 발생한 시스템에서 관리한 잠금에 대한 전혀 알 수 없습니다. 이것은 응용 프로그램이 재시작될 때 문제를 일으킬 수 있습니다.

원격 파일 잠금은 잠금을 수행하는 시스템이 장애가 발생한 시스템일 수 있기 때문에 두 경우 중 가장 나쁜 경우입니다. 이 경우 잠금이 절대 해제되지 않기 때문에 응용 프로그램의 다른 부분이 데이터에 액세스할 수 없게 됩니다. NFS 환경에서는 NFS 클라이언트 시스템 장애 시 파일 잠금에 의해 시간이 오래 지연될 수 있으며 장애 조치 자체를 지연시킬 수도 있습니다.

클라이언트 연결 복원

장애 발생 후 클라이언트가 어떻게 서버에 다시 연결됩니까?

반환될 수도 있는 다른 응용 프로그램에서 시작된 오류와 서버에 대한 연결 손실이 명확하게 구별되도록 클라이언트 응용 프로그램을 작성하는 것이 중요합니다. 응용 프로그램은 연결이 손실된 경우 특별한 조치를 취해야 합니다.

한 가지 고려할 문제는 장애 발생 후 응용 프로그램이 새로 시작된 서버에 다시 연결되는 시기를 클라이언트가 알 수 있는 방법입니다. 일반적인 경우 클라이언트가 세션을 다시 시작하거나 다시 로그인해야 하지만 이 방법은 크게 자동화되어 있지는 않습니다. 예를 들어, 잘 구성된 하드웨어 및 응용 프로그램 시스템은 5분 안에 장애 조치될 수 있습니다. 하지만 사용자가 장애 발생 후 응답이 없자 2분 뒤에 포기하고 커피를 마시러 나가 28분 동안 돌아오지 않는다고 가정하면 인식되는 중단 시간은 5분이 아니라 30분이 됩니다. 고려해야 할 요소는 재연결 시도 횟수, 재연결 시도 빈도 및 사용자에게 연결 손실을 알릴지 여부입니다.

클라이언트 재연결에 사용할 수 있는 방법은 많습니다.

- 장애가 발생한 서버로 계속 재연결을 시도하는 클라이언트 설계

사용자가 재연결하게 하지 말고 클라이언트 응용 프로그램이 재연결하게 하십시오. 5분 안에 서버가 백업되어 실행되고 클라이언트가 계속 재시도하면 5분 후에 클라이언트 응용 프로그램은 서버와 재연결하여 트랜잭션을 다시 시작하거나 계속합니다. 사용자의 개입은 필요하지 않습니다.

- 다른 서버에 재연결하는 클라이언트 설계

여러 활성 서버가 포함된 서버 설계를 사용하는 경우 클라이언트는 두 번째 서버에 연결될 수 있으므로 지연 시간이 짧아집니다.

이 설계의 문제는 클라이언트가 두 번째 서버로 전환되어야 하는 시기를 알아야 한다는 것입니다. 클라이언트가 첫 번째 서버를 포기하고 두 번째 서버로 이동하기 전에 첫 번째 서버에서 재시도를 얼마나 오래 해야 하는지에 대한 정답은 없습니다. 정답은 서버 응용 프로그램의 설계에 따라 달라집니다. 서버 응용 프로그램에서 장애 발생 후 같은 노드에서 다시 시작될 수 있으면(다음의 “응용 프로그램 장애 처리” 참조) 로컬로 서버를 다시 시작할 수 있을 때까지 현재 서버에서 재시도를 계속해야 합니다. 이렇게 하면 응용 프로그램 장애가 발생했을 때 클라이언트가 두 번째 서버로 전환하지 않아도 됩니다.

- 트랜잭션 처리 모니터 또는 메시지 대기열 소프트웨어를 사용한 안정성 향상

Tuxedo나 **DCE/Encina** 같이 서버와 클라이언트 사이에서 인터페이스를 제공하는 트랜잭션 처리 모니터를 사용하십시오. **TPM**(트랜잭션 처리 모니터)은 가용성이 더 높은 응용 프로그램을 만드는 데 유용할 수 있습니다. 트랜잭션은 대기열에서 대기될 수 있기 때문에 클라이언트는 서버의 장애를 감지하지 못합니다. 많은 **TPM**은 선택적으로 대체 서버로의 자동 경로 재설정이나 트랜잭션의 자동 재시도를 제공합니다. **TPM**은 또한 트랜잭션의 안정적인 완료를 보장합니다. 물론, **TPM**이 트랜잭션의 안정적 완료를 보장하는 유일한 메커니즘은 아닙니다. 서버가 다시 온라인 상태가 되면 트랜잭션 모니터는 새 서버로 재연결되어 트랜잭션 라우팅을 계속합니다.

- 요청을 대기열에 기록

TPM을 사용하는 대신 서버를 사용할 수 없을 때 요청을 대기열에 기록할 수 있습니다. 서버를 사용할 수 없을 때 사용자에게 알리는 대신 사용자의 요청은 대기열에 기록되어 서버를 다시 사용할 수 있을 때 전송됩니다. 메시지 대기열 소프트웨어는 트랜잭션뿐 아니라 모든 종류의 메시지가 전달되고 승인되도록 보장합니다.

메시지 대기열은 사용자가 요청이 완료되었다는 응답을 필요로 하지 않거나 기대하지 않는 경우에만 유용합니다(즉, 응용 프로그램이 대화식이 아닌 경우).

응용 프로그램 장애 처리

응용 프로그램의 일부나 전체에 장애가 발생하면 어떤 일이 발생합니까?

앞에서 다루었던 장애 상황들은 응용 프로그램이 아닌 클러스터의 다른 구성 요소에서 장애가 발생한 상황이었습니다. 이 절에서는 응용 프로그램의 문제를 다룹니다. 예를 들어, 소프트웨어의 버그로 인해 응용 프로그램에 장애가 발생하거나 시스템 리소스 문제(스왑/메모리 공간 부족 등)로 인해 응용 프로그램이 정지할 수 있습니다. 이 절에서는 이런 종류의 장애를 복구할 수 있는 응용 프로그램의 설계 방법에 대해 설명합니다.

장애를 극복할 수 있는 응용 프로그램 만들기

응용 프로그램은 단일 구성 요소의 장애를 극복할 수 있어야 합니다. 많은 응용 프로그램이 한 노드에서 여러 프로세스를 실행합니다. 한 프로세스에 장애가 발생하는 경우 다른 프로세스에도 장애가 발생하거나 장애가 발생한 프로세스가 응용 프로그램의 나머지 부분에 영향을 미치지 않고 같은 노드에서 다시 시작될 수 있습니다.

이상적인 처리 방법은 한 프로세스에 장애가 발생한 후 다시 온라인 상태로 돌아올 때까지 다른 프로세스가 대기할 수 있게 하는 것입니다. 이것은 장애를 일으킨 구성 요소가 같은 시스템에 있거나 원격 시스템에 있는 경우에 모두 해당됩니다. 장애가 발생한 구성 요소가 같은 시스템에서 자동으로 다시 시작되어 대기 상태인 작업에 다시 참가하고 작업을 계속할 수 있습니다. 이런 종류의 장애는 몇 초 안에 감지되어 다시 시작될 수 있으며 일반 사용자는 장애가 발생한 사실을 전혀 알아차리지 못합니다.

다른 방법으로 한 구성 요소에 장애가 발생할 때 다른 구성 요소를 정상 종료하는 방법이 있습니다. 데이터베이스 SQL 서버에 장애가 발생한 경우 데이터베이스 복구가 필요하지 않도록 데이터베이스가 정상 종료될 수 있어야 합니다.

더 나쁜 상황은 한 구성 요소의 장애가 전체 시스템의 장애를 유발하는 경우입니다. 한 구성 요소에 장애가 발생할 때 다른 모든 구성 요소가 다시 시작되어야 하면 중단 시간이 길어집니다.

응용 프로그램 모니터링 가능

응용 프로그램을 비롯한 시스템의 모든 구성 요소에 대한 상태를 모니터링할 수 있어야 합니다. 모니터링은 표시 명령처럼 간단하거나 SQL 쿼리처럼 복잡할 수 있습니다. 응용 프로그램이 올바르게 작동하는 것을 보장하는 방법이 있어야 합니다. 응용 프로그램에 장애가 발생했을 때 자동으로 감지되지 않으면 사용자가 장애 원인을 파악하고 복구하는 데 많은 시간이 걸릴 수 있습니다.

계획된 중단 시간 최소화

계획된 중단 시간(계획되지 않은 중단 시간과 반대)의 예는 백업, 시스템 OS 업그레이드, 하드웨어 교체 등입니다. 계획된 중단 시간의 경우 응용 프로그램 설계자는 다음 사항을 고려해야 합니다.

- **응용 프로그램 업그레이드/패치에 필요한 시간 단축**

관리자가 중단 시간을 계획하지 않고 응용 프로그램의 새 버전을 설치할 수 있습니까? 한 시스템에서 서로 다른 버전의 응용 프로그램이 함께 작동할 수 있습니까? 한 시스템에서 서로 다른 버전의 클라이언트와 서버가 함께 작동할 수 있습니까?

- **온라인 응용 프로그램 재구성 제공**

응용 프로그램을 중지하지 않고 응용 프로그램에서 사용하는 구성 정보를 변경할 수 있습니까?

- **유지 관리 작업 문서화**

운영자가 유지 관리 작업을 처리하는 방법을 알고 있습니까?

고가용성 시스템에 대한 논의에서 일반적인 주요 관심사는 계획되지 않은 중단 시간입니다. 그러나 계획된 중단 시간의 경우라도 시스템을 새로운 버전의 소프트웨어로 업그레이드하는 데 2주가 걸린다면 상당한 불만이 제기될 것입니다.

다음 절에서는 여러 유형의 계획된 중단 시간을 처리하는 방법에 대해 설명합니다.

응용 프로그램 업그레이드/패치에 필요한 시간 단축

거의 매년 응용 프로그램의 새 버전이 출시됩니다. 최종 사용자가 새 버전으로 업그레이드하는 데 얼마나 많은 시간이 필요한지에 대한 답은 사용자가 자신의 응용 프로그램을 업그레이드할 때 경험해야 하는 계획된 중단 시간의 양입니다. 다음은 이 시간을 단축하는 방법에 대한 지침입니다.

롤링 업그레이드 제공

클라이언트/서버 환경에서 “롤링 업그레이드”를 제공하십시오. 많은 구성 요소가 있는 시스템의 경우 전형적인 업그레이드 방식은 전체 시스템을 종료하고 모든 노드를 새로운 버전의 소프트웨어로 업그레이드한 다음 모든 노드에서 응용 프로그램을 다시 시작하는 것입니다. 이러한 방식은 대규모 시스템에서 긴 중단 시간을 초래할 수 있습니다.

이에 대한 대안은 롤링 업그레이드를 수행하는 것입니다. 롤링 업그레이드는 한 번에 한 구성 요소만 업그레이드 하는 단계별 접근 방식으로 새로운 소프트웨어를 설치합니다. 예를 들어, 월요일에 데이터베이스 서버를 업그레이드하면서 15분의 중단 시간이 발생합니다. 그런 다음 화요일에는 두 노드의 응용 프로그램 서버를 업그레이드하지만 나머지 노드의 응용 프로그램 서버는 그대로 두었기 때문에 중단 시간이 발생하지 않습니다. 수요일에는 나머지 응용 프로그램 서버를 업그레이드합니다. 이런 접근 방식을 사용하면 모든 것을 한 번에 변경했을 때 일어날 수 있는 문제를 피할 수 있으며 중단 시간을 최소화할 수 있습니다.

롤링 업그레이드를 지원하는 Serviceguard 릴리즈에 대한 자세한 내용은

<http://docs.hp.com> (영문 설명서) 및 <http://docs.hp.com/ko> (한글 설명서)에서 사용자의 버전에 맞는 Serviceguard 릴리즈 노트를 참조하십시오.

이 방법의 문제는 응용 프로그램 소프트웨어의 여러 버전을 함께 운영해야 한다는 것입니다. 위의 예에서 데이터베이스 서버의 버전은 5.0인데 일부 응용 프로그램 서버의 버전은 4.0일 수 있습니다. 응용 프로그램은 이러한 상황을 처리할 수 있도록 설계되어야 합니다.

버전 간 데이터 레이아웃을 변경하지 않음

새로운 형식으로 데이터를 마이그레이션하는 작업에는 상당한 시간이 걸릴 수 있으며, 롤링 업그레이드도 거의 불가능해집니다. 예를 들어, 데이터베이스가 첫 번째 노드에서 실행 중이면 두 번째 노드를 새 버전의 데이터베이스로 업그레이드할 수 있습니다. 업그레이드가 완료되면 짧은 중단 시간을 계획하여 데이터베이스 서버를 첫 번째 노드에서 새로 업그레이드된 두 번째 노드로 이동할 수 있습니다. 그런 다음 데이터베이스 서버를 다시 시작하면 첫 번째 노드는 대기 상태가 되어 업그레이드가 가능한 상태가 됩니다. 그러나 새 버전의 데이터베이스에서 다른 데이터베이스 레이아웃이 필요하면 **이전** 데이터를 새로 업데이트된 데이터베이스에서 읽을 수 없습니다. 데이터를 새로운 레이아웃으로 마이그레이션해야 하기 때문에 중단 시간이 길어집니다.

온라인 응용 프로그램 재구성 제공

대부분의 응용 프로그램에는 응용 프로그램을 시작할 때 읽는 일종의 구성 정보가 있습니다. 이 구성 정보를 변경하면 응용 프로그램을 중단하고 새로운 구성 파일을 읽어야 하므로 중단 시간이 발생합니다.

이러한 중단 시간을 방지하려면 응용 프로그램과 상호 작용하여 온라인 상태에서 동적으로 구성을 변경할 수 있는 구성 도구를 사용해야 합니다. 이상적인 해결책은 응용 프로그램과 상호 작용하는 구성 도구를 사용하는 것입니다. 구성은 온라인 상태에서 변경되고 일반 사용자는 거의 영향을 받지 않습니다. 이 도구는 데이터 크기를 확장하거나, 시스템에 새로운 사용자를 추가하거나, 응용 프로그램에 새로운 사용자를 추가하는 등의 모든 작업을 온라인에서 처리할 수 있어야 합니다. 응용 프로그램 시스템에 대해 관리자가 수행해야 하는 모든 작업이 온라인 상태에서 수행될 수 있어야 합니다.

유지 관리 작업 문서화

표준 절차가 중요합니다. 응용 프로그램 설계자는 고가용성 환경과 일반 환경 모두에 대한 작업을 일반화하는 데 최선을 다해야 합니다. 장애가 발생했을 때 전체 시스템을 종료하는 것에 익숙해진 관리자는 응용 프로그램이 단순 장애를 처리할 수 있도록 재설계된 이후에도 계속해서 전체 시스템을 종료할 것입니다. 응용 프로그램 문서에서는 고가용성과 관련하여 일반적인 유지 관리 작업을 대신할 수 있는 방법에 대해 다루어야 합니다.

D HA 응용 프로그램과 Serviceguard 통합

다음은 응용 프로그램을 Serviceguard 환경에 통합하기 위해 수행해야 하는 단계를 요약한 것입니다.

1. 클러스터 및 패키지 구성에 대한 장과 부록 “고가용성 클러스터 응용 프로그램 설계”를 비롯한 이 설명서의 나머지 부분을 읽습니다.
2. 정상적으로 작동될 때의 클러스터 동작을 정의합니다.
 - 정상적으로 작동될 때 클러스터의 상태는 어떠해야 합니까?
 - 대부분의 사람이 사용하는 표준 구성은 어떤 것입니까? 사용자 요구 사항에 대해 사용할 수 있는 데이터가 있습니까?
 - 데이터베이스나 응용 프로그램 서버와 같은 기능을 별도의 시스템으로 분리할 수 있습니까? 아니면, 모든 기능이 한 시스템에서 실행되고 있습니까?
3. 장애 조치 시 클러스터의 동작을 정의합니다.
 - 장애 조치 시 모든 기능이 함께 대체 노드로 전달됩니까?
 - 장애 조치 시 같은 노드로 전달되는 응용 프로그램을 분리할 수 있습니까?
 - 응용 프로그램에 Serviceguard에서 제공하는 기능 이외의 다른 고가용성 메커니즘이 이미 있습니까?
4. 문제가 있는 부분을 확인합니다.
 - 현재 응용 프로그램에서 시스템 재부팅이나 패닉 상태를 처리하기 위해 수행하는 작업은 무엇입니까?
 - 장애 조치를 위해 다른 시스템으로 전달하는 데 문제가 될 수 있는 `uname()`이나 `gethostname()`, `SPU_ID` 또는 `MAC` 주소 등의 시스템 관련 정보를 응용 프로그램이 사용하고 있습니까?

HA 응용 프로그램을 통합하기 위한 검사 목록

이 절에는 HA 응용 프로그램을 단일 시스템 및 다중 시스템에 통합하기 위한 검사 목록이 포함되어 있습니다.

단일 시스템의 기존 응용 프로그램 동작 정의

1. 독립형 시스템에서 응용 프로그램의 기존 동작을 정의합니다.

- 응용 프로그램, 데이터베이스 및 기타 필요한 리소스를 하나의 시스템에 설치합니다. 이 작업을 수행할 때는 **Serviceguard** 규칙을 따라야 합니다.
 - 모든 공유 데이터를 별도의 외부 볼륨 그룹에 설치합니다.
 - 필요하면 **JFS** 파일 시스템을 사용합니다.
- 몇 가지 표준 테스트를 수행하여 응용 프로그램이 올바르게 실행되고 있는지 확인합니다. 이 테스트는 나중에 **Serviceguard**에서 테스트할 때 사용할 수 있습니다. 가능하면 클라이언트를 통해 응용 프로그램에 연결하십시오.
- 응용 프로그램이 설치된 독립형 시스템에 장애를 일으키고 재부팅한 다음 응용 프로그램이 다시 시작되는 방식을 테스트합니다. 이때 주의해야 할 사항은 다음과 같습니다.
 - 수동 조작이 필요한 단계가 있습니까? 있으면 기록합니다.
 - 모든 항목을 **rc** 스크립트에서 시작할 수 있습니까?
- 키보드로 입력하지 않고도 모든 것을 시작할 수 있는 간단한 스크립트를 만듭니다. 관리자가 키보드로 입력해야 하는 작업을 찾아 스크립트에 포함시킵니다.
- 응용 프로그램을 종료하는 간단한 스크립트를 만듭니다. 여기에서도 관리자가 키보드로 입력해야 하는 작업을 찾아 스크립트에 포함시킵니다.

다중 시스템에 HA 응용 프로그램 통합

1. 두 번째 시스템에 응용 프로그램을 설치합니다.

- 두 번째 시스템에 LVM 하부 구조를 만듭니다.
 - 시스템에 적절한 사용자를 추가합니다.
 - 적절한 실행 파일을 설치합니다.
 - 첫 번째 시스템에서 응용 프로그램이 실행되고 있지 **않은** 상태에서 두 번째 시스템에서 응용 프로그램을 시작합니다. 위의 단계에서 만든 스크립트를 사용할 수도 있습니다. 수행해야 하는 작업에 차이가 있습니까? 응용 프로그램이 실행됩니까?
 - 두 번째 시스템에서 응용 프로그램을 실행할 수 있을 때까지 이 과정을 반복합니다.
2. Serviceguard 클러스터를 구성합니다.
- 클러스터 구성을 만듭니다.
 - 패키지를 만듭니다.
 - 패키지 스크립트를 만듭니다.
 - 이전 단계에서 만든 간단한 스크립트를 패키지 제어 스크립트에서 사용자 정의 함수로 사용합니다.
3. 클러스터를 시작하고 응용 프로그램이 예상대로 실행되는지 확인합니다.

클러스터 테스트

1. 클러스터를 테스트합니다.
- 클라이언트를 연결합니다.
 - 일반적인 시스템 부하 상태를 만듭니다.
 - 첫 번째 노드에서 패키지를 중지하고 두 번째 노드로 이동합니다.

```
# cmhaltpkg pkg1  
# cmrunpkg -n node2 pkg1  
# cmmodpkg -e pkg1
```
 - 다시 첫 번째 노드로 이동합니다.

```
# cmhaltpkg pkg1  
# cmrunpkg -n node1 pkg1  
# cmmodpkg -e pkg1
```
 - 시스템 중 하나에 장애를 일으킵니다. 예를 들어, 노드 1의 전원을 끈 다음 패키지가 노드 2에서 시작되는지 확인합니다.

- 노드 2에서 노드 1로 장애 조치를 반복합니다.
2. 이 테스트 과정에서 응용 프로그램 부하의 모든 조합을 테스트하도록 합니다. 사용자 부하가 많을 때와 적을 때, 일괄 작업과 온라인 트랜잭션 작업 등의 서로 다른 응용 프로그램 상태에서 장애 조치 과정을 반복합니다.
 3. 각 응용 프로그램 상태마다 장애 조치에 걸린 시간을 기록합니다. 예를 들면, 클러스터 재구성에 45초, 파일 시스템에서 fsck를 실행하는 데 15초, 응용 프로그램 시작에 30초, 데이터베이스 복구에 3분이 걸릴 수 있습니다.

E 롤링 소프트웨어 업그레이드

클러스터를 중지하지 않고 한 번에 한 노드씩 HP-UX 운영 체제와 Serviceguard 소프트웨어를 업그레이드할 수 있습니다. 이 과정은 하드웨어 유지 관리나 패치 설치를 위해 한 시스템을 오프라인 상태로 만들어야 하는 경우에도 사용할 수 있습니다. 모든 노드에서 업그레이드 과정이 완료될 때까지는 클러스터 구성 파일을 변경할 수 없으며 새로운 Serviceguard 버전의 기능을 사용할 수 없습니다.

롤링 업그레이드는 Serviceguard의 지원되는 모든 버전에서 수행할 수 있습니다. 이 방법으로 모든 이전 버전을 더 높은 버전으로 업그레이드할 수 있습니다. 예를 들어, HP-UX 10.10의 Serviceguard 버전 A.10.05를 HP-UX 11.11의 A.11.15로 업그레이드할 수 있습니다.

이 부록에 포함된 내용은 다음과 같습니다.

- 롤링 업그레이드 단계
- 롤링 업그레이드 예제
- 롤링 업그레이드의 제한 사항

롤링 업그레이드 단계

다음 단계를 따르십시오.

1. 업그레이드할 노드를 중지합니다. 이렇게 하면 해당 노드의 패키지가 대체 노드에서 시작됩니다. **Serviceguard Manager**에서 중지할 노드를 선택하고 동작 메뉴에서 관리를 선택한 다음 노드 중지를 선택합니다. 또는 **Serviceguard** 명령줄에서 `cmhaltnode` 명령을 실행합니다.
2. `/etc/rc.config.d/cmcluster` 파일을 편집하여 다음 줄을 포함시킵니다.

```
AUTOSTART_CMCLD = 0
```
3. 노드를 사용 가능한 **HP-UX** 버전과 **Serviceguard** 버전으로 업그레이드합니다. **SCSI** 케이블의 연결을 끊지 않는 범위 내에서 필요에 따라 다른 소프트웨어나 하드웨어도 업그레이드할 수 있습니다(예: **VERITAS Volume Manager** 소프트웨어의 설치). 자세한 내용은 “문제 해결” 장에서 하드웨어 유지 관리에 대한 절을 참조하십시오.
4. `/etc/rc.config.d/cmcluster` 파일을 편집하여 다음 줄을 포함시킵니다.

```
AUTOSTART_CMCLD = 1
```
5. 업그레이드된 노드에서 클러스터를 다시 시작합니다. **Serviceguard Manager**에서 노드를 선택하고 동작 메뉴에서 관리를 선택한 다음 노드 실행을 선택합니다. 또는 **Serviceguard** 명령줄에서 `cmrunnode` 명령을 실행합니다.
6. 클러스터의 각 노드에서 이 과정을 반복합니다.

주

이 과정에서 패키지가 한 노드에서 다른 노드로 이동해도 심각한 성능 저하가 발생하지 않도록 시스템 용량을 여유 있게 계획해야 합니다.

심각한 전원 문제 등으로 인해 롤링 업그레이드가 완료되기 전에 클러스터가 중지되면 최신 버전의 소프트웨어로 업그레이드된 노드에서 `cmrunc1` 명령을 입력하여 클러스터를 다시 시작할 수 있습니다.

커널 일관성 유지

롤링 업그레이드 과정에서 커널 매개 변수를 변경하는 경우 장애 조치 상황에서 동일한 패키지를 실행할 수 있는 모든 노드에서도 해당 매개 변수를 변경해야 합니다.

cmclnodelist 항목을 A.11.16으로 마이그레이션

Serviceguard 버전 A.11.16으로 업그레이드하면 **cmclnodelist** 파일이 삭제됩니다.

이 파일의 정보는 새로운 액세스 제어 정책 형식으로 마이그레이션됩니다.

cmclnodelist 파일의 모든 <hostnode> <username> 쌍은 클러스터 구성 파일에서 세 요소가 함께 정의되며 모두 모니터링 역할을 갖습니다. 루트 권한이 없는 사용자에게 관리 역할을 부여하려면 구성 파일에 항목을 추가합니다.

클러스터의 모든 노드가 버전 A.11.16으로 업그레이드되기 전까지 **cmclnodelist**는 현재 상태를 유지합니다.

새 액세스 정책에 대한 자세한 내용은 182페이지의 “보안 파일 편집”을 참조하십시오.

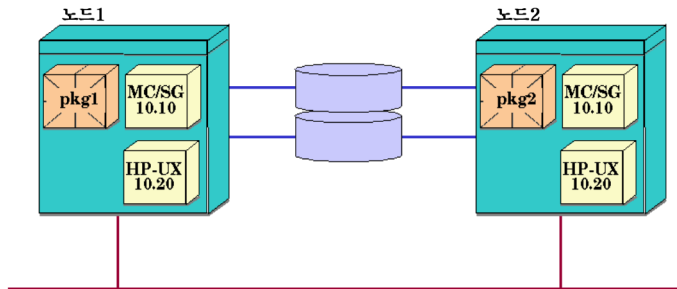
롤링 업그레이드 예제

롤링 업그레이드 수행 중 노드에서 실행되고 있는 소프트웨어 버전을 확인하는 동안 경고 메시지가 표시될 수 있습니다. 이것은 정상적인 상황이며 문제가 되지 않습니다.

다음은 패키지를 하나씩 실행하는 두 노드에서 간단한 롤링 업그레이드를 수행하는 예제입니다(그림 E-1 참조). 이 그림과 다음 그림에서는 예시용으로 Serviceguard 10.10 및 HP-UX 10.20에서 시작하여 Serviceguard 11.13 및 HP-UX 11.00으로 롤링 업그레이드하는 과정을 보여 줍니다. 사용자 시스템에서는 롤링 업그레이드 경로의 실제 버전 번호로 대체하십시오.

그림 E-1

롤링 업그레이드 전의 클러스터 실행



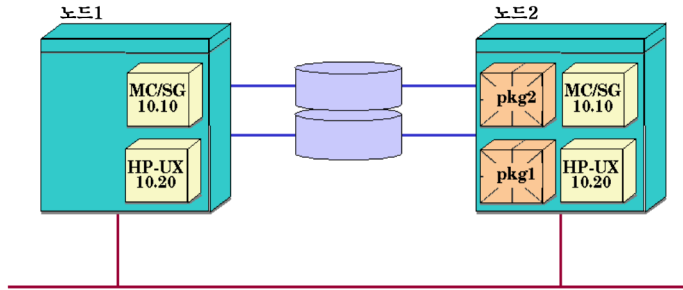
단계 1

다음과 같이 첫 번째 노드를 중지하십시오.

```
# cmhaltnode -f node1
```

이 명령은 PKG1을 정상적으로 중지하고 노드 2로 이동합니다. 그림 E-2와 같이 노드 1의 Serviceguard 데몬은 중지됩니다.

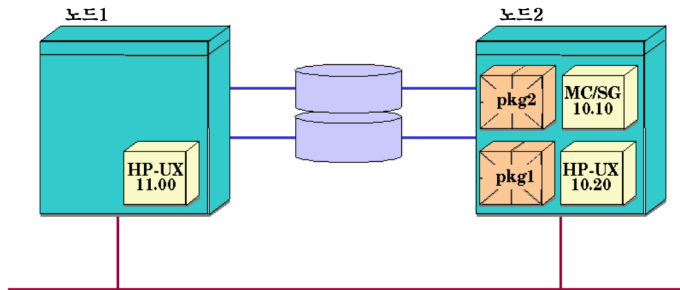
그림 E-2 노드 2로 이동한 패키지를 포함하는 클러스터 실행



단계 2

그림 E-3과 같이 노드 1을 상위 버전의 운영 체제(이 예제에서는 HP-UX 11.00)로 업그레이드한 다음 상위 버전의 Serviceguard(11.13)를 설치합니다.

그림 E-3 노드 1을 HP-UX 11.00으로 업그레이드



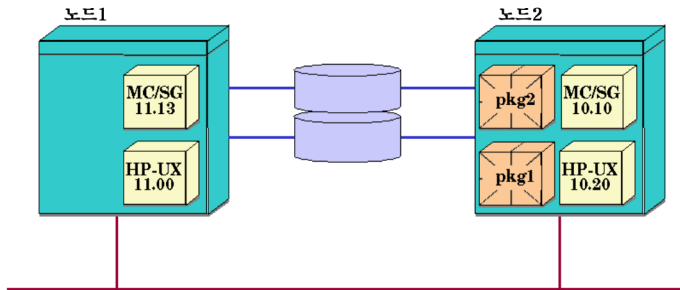
단계 3

업그레이드가 완료되면 노드 1에서 다음 명령을 입력하여 클러스터를 노드 1에서 다시 시작합니다.

```
# cmrunnode -n node1
```

이제 그림 E-4와 같이 두 노드에서 여러 버전의 Serviceguard 데몬(*cmclcd*)이 실행됩니다.

그림 E-4 노드 1이 클러스터에 다시 참가



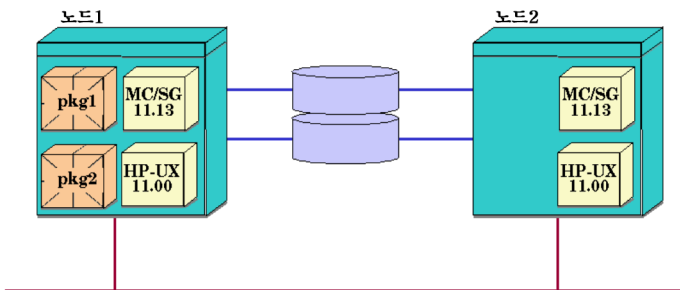
단계 4

노드 2에서 이 과정을 반복합니다. 다음과 같이 노드를 중지합니다.

```
# cmhaltnode -f node2
```

이렇게 하면 두 패키지가 모두 노드 1로 이동합니다. 그런 다음 노드 2를 HP-UX 11.00 및 Serviceguard 11.13으로 업그레이드합니다.

그림 E-5 노드 1로 이동한 패키지를 포함하는 클러스터 실행



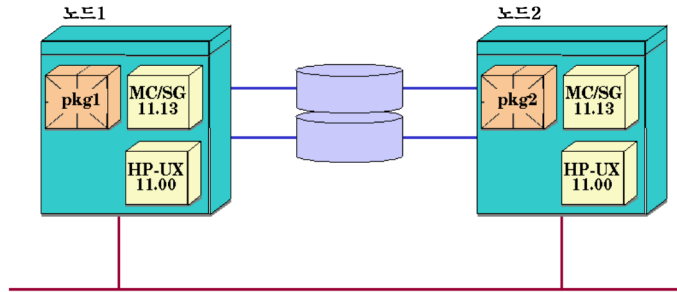
단계 5

PKG2를 다시 원래 노드로 이동합니다. 다음 명령을 수행합니다.

```
# cmhaltpkg pkg2  
# cmrunpkg -n node2 pkg2  
# cmmodpkg -e pkg2
```

cmmodpkg 명령은 cmhaltpkg 명령으로 비활성화된 패키지 전환 기능을 다시 활성화합니다. 그림 E-6에서는 최종 실행 클러스터를 보여 줍니다.

그림 E-6 업그레이드 후의 클러스터 실행



롤링 업그레이드의 제한 사항

롤링 업그레이드에는 다음과 같은 제한 사항이 있습니다.

- 롤링 업그레이드 과정에서는 최신 버전의 소프트웨어가 설치된 노드에서만 **Serviceguard** 명령(`cmrunnode`와 `cmhaltnode`는 제외)을 실행해야 합니다. 이전 버전의 소프트웨어가 설치된 노드에서 작업을 수행하면 올바르게 작동하지 않거나 일치하지 않는 결과를 얻게 됩니다.
- 업그레이드가 완료될 때까지는 클러스터나 패키지 구성을 수정할 수 없습니다. 롤링 업그레이드 도중에는 클러스터의 네트워크 구성을 포함한 하드웨어 구성을 수정할 수 **없습니다**. 즉, 모든 노드를 새 버전으로 업그레이드한 후에 구성 파일을 수정하고 모든 노드에 복사할 수 있습니다.
- 모든 노드가 업그레이드될 때까지는 새로운 **Serviceguard** 버전의 기능을 사용할 수 없습니다.
- 서로 다른 **Serviceguard** 버전 사이에서는 이전 구성 파일이 호환되지 않을 수 있습니다. 노드 간에 구성 파일을 수동으로 복사하지 **마십시오**.
- **Serviceguard** 클러스터에서 롤링 업그레이드가 진행되고 있을 때는 최대 두 버전의 **Serviceguard**만 실행할 수 있습니다.
- 클러스터가 마지막으로 시작된 이후 구성이 수정되지 않은 경우에만 롤링 업그레이드를 수행할 수 있습니다.
- 롤링 업그레이드는 클러스터에서 여러 버전의 **Serviceguard** 또는 **HP-UX**를 함께 사용하기 위한 수단은 아닙니다. 모든 클러스터 노드를 가능하면 빨리 새 버전으로 업그레이드하는 것이 좋습니다.
- 클러스터가 롤링 업그레이드 과정에 있을 때는 노드에서 `swremove` 명령으로 **Serviceguard** 소프트웨어를 삭제할 수 없습니다.

F 계획 워크시트 양식

여기에는 “HA 클러스터 계획 및 문서화” 장에서 설명한 계획 워크시트의 양식이 첨부되어 있습니다. 여기에 있는 워크시트 중 유용한 것을 복사하여 계획 과정에서 작성하여 사용할 수 있습니다.

하드웨어 계획 워크시트

HARDWARE WORKSHEET

Page ____ of ____

=====
Node Information:

Host Name _____ Series No _____

Memory Capacity _____ Number of I/O Slots _____

=====
LAN Information:

| | | | |
|----------------------|-------------------------|---------------|--------------------|
| Name of Subnet _____ | Name of Interface _____ | IP Addr _____ | Traffic Type _____ |
|----------------------|-------------------------|---------------|--------------------|

| | | | |
|----------------------|-------------------------|---------------|--------------------|
| Name of Subnet _____ | Name of Interface _____ | IP Addr _____ | Traffic Type _____ |
|----------------------|-------------------------|---------------|--------------------|

| | | | |
|----------------------|-------------------------|---------------|--------------------|
| Name of Subnet _____ | Name of Interface _____ | IP Addr _____ | Traffic Type _____ |
|----------------------|-------------------------|---------------|--------------------|

=====
Serial Heartbeat Interface Information:

Node Name _____ RS232 Device File _____

Node Name _____ RS232 Device File _____

=====
Disk I/O Information:

| | | |
|----------------|---------------------|------------------------|
| Bus Type _____ | Hardware Path _____ | Device File Name _____ |
|----------------|---------------------|------------------------|

| | | |
|----------------|---------------------|------------------------|
| Bus Type _____ | Hardware Path _____ | Device File Name _____ |
|----------------|---------------------|------------------------|

| | | |
|----------------|---------------------|------------------------|
| Bus Type _____ | Hardware Path _____ | Device File Name _____ |
|----------------|---------------------|------------------------|

Attach a printout of the output from `ioscan -f` and `lssfs /dev/*dsk/*s2` after installing disk hardware and rebooting the system. Mark this printout to indicate which physical volume group each disk belongs to.

전원 공급 장치 워크시트

POWER SUPPLY WORKSHEET

Page ____ of ____

=====

SPU Power:

Host Name _____ Power Supply _____

Host Name _____ Power Supply _____

=====

Disk Power:

Disk Unit _____ Power Supply _____

Disk Unit _____ Power Supply _____

Disk Unit _____ Power Supply _____

Disk Unit _____ Power Supply _____

Disk Unit _____ Power Supply _____

Disk Unit _____ Power Supply _____

=====

Tape Backup Power:

Tape Unit _____ Power Supply _____

Tape Unit _____ Power Supply _____

=====

Other Power:

Unit Name _____ Power Supply _____

Unit Name _____ Power Supply _____

쿼럼 서버 워크시트

Quorum Server Data:

=====

QS Hostname: _____ IP Address: _____

=====

Quorum Services are Provided for:

Cluster Name: _____

Host Names _____

Host Names _____

Cluster Name: _____

Host Names _____

Host Names _____

LVM 볼륨 그룹 및 물리 볼륨 워크시트

PHYSICAL VOLUME WORKSHEET

Page ____ of ____

=====

Volume Group Name: _____

PV Link 1

PV Link2

Physical Volume Name: _____

Physical Volume Name: _____

Physical Volume Name: _____

Physical Volume Name: _____

Physical Volume Name: _____

Physical Volume Name: _____

Physical Volume Name: _____

Physical Volume Name: _____

Volume Group Name: _____

PV Link 1

PV Link2

Physical Volume Name: _____

Physical Volume Name: _____

Physical Volume Name: _____

Physical Volume Name: _____

Physical Volume Name: _____

Physical Volume Name: _____

Physical Volume Name: _____

Physical Volume Name: _____

Physical Volume Name: _____

VxVM 디스크 그룹 및 디스크 워크시트

DISK GROUP WORKSHEET

Page ____ of ____

=====
=====

Disk Group Name:

Physical Volume

Name: _____

Physical Volume

Name: _____

Physical Volume

Name: _____

Physical Volume Name:

Physical Volume Name:

Physical Volume Name:

Physical Volume Name:

Physical Volume Name:

Disk Group Name:

Physical Volume Name:

Physical Volume

Name: _____

Physical Volume Name:

Physical Volume Name:

Physical Volume Name:

Physical Volume Name:

Physical Volume Name:

Physical Volume Name:

Physical Volume Name:

클러스터 구성 워크시트

=====
Name and Nodes:
=====

Cluster Name: _____ OPS Version: _____

Node Names: _____

Volume Groups (for packages): _____
=====

Subnets:
=====

Heartbeat Subnet: _____

Monitored Non-heartbeat Subnet: _____

Monitored Non-heartbeat Subnet: _____
=====

Cluster Lock Volume Groups and Volumes:
=====

First Lock Volume Group:

Physical Volume:

Name on Node 1: _____

Name on Node 2: _____

Disk Unit No: _____

Power Supply No: _____
=====

Timing Parameters:
=====

Heartbeat Interval: _____

Node Timeout: _____

Network Polling Interval: _____

Autostart Delay: _____
=====

Access Policies:

User name:

Host node:

Role:

=====

패키지 구성 워크시트

=====
Package Configuration File Data:
=====

Package Name: _____

Failover Policy: _____

Failback Policy: _____

Primary Node: _____

First Failover Node: _____

Additional Failover Nodes: _____

Package Run Script: _____ Timeout: _____

Package Halt Script: _____ Timeout: _____

Package AutoRun Enabled? _____ Local LAN Failover Allowed? _____

Node Failfast Enabled? _____

CVM Storage Groups:

Additional Package Resource:

Resource Name: _____ Polling Interval _____ Resource UP Value _____

=====
Access Policies:

User name:

Host node:

Role:
=====

패키지 제어 스크립트 워크시트

LVM Volume Groups:

VG[0]_____VG[1]_____VG[2]_____

VGCHANGE: _____

CVM Disk Groups:

CVM_DG[0]_____CVM_DG[1]_____CVM_DG[2]_____

CVM_ACTIVATION_CMD: _____

VxVM Disk Groups:

VXVM_DG[0]_____VXVM_DG[1]_____VXVM_DG[2]_____

=====
Logical Volumes and File Systems:

LV[0]_____FS[0]_____FS_MOUNT_OPT[0]_____

LV[1]_____FS[1]_____FS_MOUNT_OPT[1]_____

LV[2]_____FS[2]_____FS_MOUNT_OPT[2]_____

FS Umount Count: _____FS Mount Retry Count: _____

=====
Network Information:

IP[0] _____ SUBNET _____

IP[1] _____ SUBNET _____

=====
Services:

Service Name: _____ Command: _____ Restart: _____

Service Name: _____ Command: _____ Restart: _____

계획 워크시트 양식

패키지 제어 스크립트 워크시트

Deferred Resources:

Deferred Resource Name _____

G LVM에서 VxVM 데이터 저장 장치로 마이그레이션

여기에서는 LVM 볼륨 그룹을 VERITAS Volume Manager(VxVM)나 Cluster Volume Manager(CVM)에서 사용하기 위해 VxVM 디스크 그룹으로 마이그레이션하는 방법에 대해 설명합니다. 이 부록에 포함된 내용은 다음과 같습니다.

- VxVM 로드
- 볼륨 그룹 마이그레이션
- VxVM 패키지 사용자 정의
- CVM 패키지 사용자 정의
- LVM 볼륨 그룹 제거

여기에서는 마이그레이션 중에 클러스터와 패키지를 관리하기 위해 수행해야 하는 단계를 중점적으로 설명합니다. VxVM 디스크 그룹의 구성 방법에 대한 자세한 내용은 <http://docs.hp.com>에서 *VERITAS Volume Manager Administrator's Guide*와 *VERITAS Volume Manager Migration Guide*를 참조하십시오. 새 디스크에서 시작하여 새 시스템의 기본 저장 장치를 만들려면 5장을 참조하십시오.

아래에서 설명하는 절차는 클러스터가 실행되는 중에 수행할 수 있지만 마이그레이션되고 있는 볼륨 그룹을 사용하는 패키지는 모두 중지되어야 합니다. Cluster Volume Manager(CVM)에서 사용할 디스크 그룹의 경우에는 몇 개의 단계가 더 제공됩니다.

VxVM 로드

데이터 마이그레이션을 시작하려면 먼저 모든 클러스터 노드에서 필요한 모든 VxVM 라이선스와 VERITAS Volume Manager 소프트웨어를 설치해야 합니다. 이 단계에서는 각 시스템이 재부팅되므로 설치 전에 클러스터에서 노드를 제거하고 설치 후에 노드를 다시 시작해야 합니다. 이 작업은 부록 E에서 설명한 롤링 업그레이드 과정을 통해 수행할 수 있습니다.

VxVM 설치에 대한 자세한 내용은 <http://docs.hp.com>에서 *VERITAS Volume Manager Release Notes*를 참조하십시오.

볼륨 그룹 마이그레이션

다음 절차에서는 특정 노드에서 실행되도록 구성된 패키지에 대한 개별 볼륨 그룹의 마이그레이션을 수행하는 방법을 보여 줍니다. 패키지의 모든 볼륨 그룹을 동시에 변환하는 것이 좋습니다.

여기에서는 VxVM 소프트웨어와 HP-UX 및 Serviceguard의 적절한 버전이 노드에 설치되어 있고 노드가 재부팅되어 클러스터에 다시 참가한 것으로 가정합니다. 또한 “루트 디스크 그룹 만들기”에서 설명한 것처럼 노드에서 `rootdg`를 만든 것으로 가정합니다.

1. VxVM으로 변환하려는 볼륨 그룹을 활성화하는 패키지를 중지합니다.

```
# cmhaltpkg PackageName
```

2. LVM 볼륨 그룹을 읽기 전용 모드로 활성화합니다.

```
# vgchange -a r VolumeGroupName
```

3. 해당 볼륨 그룹에 포함된 데이터에 가장 적절한 방법을 사용하여 볼륨 그룹의 데이터를 백업합니다. 예를 들어, Omniback과 같은 백업/복원 유틸리티를 사용하거나 `dd`와 같은 HP-UX 유틸리티를 사용할 수 있습니다.

4. 볼륨 그룹 구성을 백업합니다.

```
# vgcfgbackup
```

5. 새 VxVM 디스크 그룹과 논리 볼륨을 정의합니다. VxVM 버전의 모든 LVM 볼륨 그룹을 만들기 위해 사용 가능한 추가 디스크가 충분히 있어야 합니다. LVM 구성과 일반 레이아웃이 같은 VxVM 논리 볼륨을 만들어야 합니다. 예를 들어, LVM의 미러링되는 볼륨의 경우 SCSI 컨트롤러에 미러 사본을 하나 두고 다른 컨트롤러에 두 번째 미러 사본을 두어서 전체 볼륨의 동작을 중지하는 단일 컨트롤러 장애를 방지할 수도 있습니다. 이러한 분리를 적용하기 위해 물리 볼륨 그룹이 LVM에서 사용되기도 합니다. 동일한 미러링 패턴이 VxVM `plex`를 만들 때도 나타나야 합니다. 각 `plex`는 서로 다른 버스에 연결되는 디스크에서 구성됩니다.

새 디스크 집합에서 VxVM 디스크 그룹을 정의하는 방법 대신 vxvmconvert (1M) 유틸리티를 사용하여 기존 LVM 볼륨 그룹을 VxVM 디스크 그룹으로 변환할 수도 있습니다. 이 유틸리티에 대한 제한 사항과 주의점을 비롯한 자세한 내용은 <http://docs.hp.com>에서 *VERITAS Volume Manager Release Notes*를 참조하십시오. vxconvert (1M) 유틸리티를 사용하는 경우에는 다음 단계를 건너뛰고 다음 절을 읽으십시오.

주

클러스터 잠금 디스크는 LVM 볼륨 그룹과 물리 볼륨에서 구성되어야 합니다. 이 구성은 VxVM으로 이동하려는 데이터가 포함된 잠금 볼륨 그룹이 있으면 가능하지만 LVM 헤더가 잠금 디스크에서 여전히 필요하기 때문에 vxvmconvert를 사용하면 안 됩니다.

-
6. 데이터를 새 VxVM 디스크 그룹으로 복원합니다. 위의 3단계에서 데이터를 백업한 방식에 가장 적절한 방법을 사용합니다.

VxVM 패키지 사용자 정의

VxVM 디스크 그룹을 만든 후에는 저장 장치에 액세스할 Serviceguard 패키지를 사용자 정의해야 합니다. VERITAS Volume Manager(VxVM)에서 사용할 디스크 그룹에 대해 다음 절차를 사용하십시오. Cluster Volume Manager(CVM)를 사용하고 있으면 이 절을 건너뛰고 다음 절을 읽으십시오.

1. 다음과 같이 기존 패키지 제어 스크립트의 이름을 바꿉니다.

```
# mv Package.ct1 Package.ct1.bak
```

2. 기존 이름과 같은 이름으로 새 패키지 제어 스크립트를 만듭니다.

```
# cmmakepkg -s Package.ct1
```

3. 새로운 VxVM 디스크 그룹과 논리 볼륨의 이름을 포함하도록 새 스크립트를 편집합니다.

VxVM에 대해 필요한 패키지 제어 스크립트의 새로운 부분은 다음과 같습니다.

- VXVM_DG[] 배열은 해당 패키지에서 사용하는 VxVM 디스크 그룹을 정의합니다. 다음과 같이 첫 번째 VXVM_DG[] 항목의 인덱스는 0이고 두 번째 항목의 인덱스는 1이 되는 식으로 인덱스가 부여되어야 합니다.

```
VXVM_DG[0]="dg01"
```

```
VXVM_DG[1]="dg02"
```

- LV[], FS[] 및 FS_MOUNT_OPT[] 배열은 LVM에서 사용될 때와 마찬가지로 사용됩니다. LV[]는 논리 볼륨을 정의하고, FS[]는 마운트 지점을 정의하며, FS_MOUNT_OPT[]는 마운트 옵션을 정의합니다. 예를 들어, 위의 두 디스크 그룹에서 두 개의 볼륨을 정의하는 경우 정의된 볼륨이 각각 lvol101, lvol102와 lvol201, lvol202라면 이 볼륨은 각각 /mnt_dg0101, /mnt_dg0102, /mnt_dg0201 및 /mnt_dg0202에 마운트됩니다. /mnt_dg0101과 /mnt_dg0201은 모두 읽기 전용으로 마운트됩니다. 이 경우 LV[], FS[] 및 FS_MOUNT_OPT[] 항목은 다음과 같습니다.

```
LV[0]="/dev/vx/dsk/dg01/lvol101"
```

```
LV[1]="/dev/vx/dsk/dg01/lvol102"
```

```
LV[2]="/dev/vx/dsk/dg02/lvol201"
```

```
LV[3]="/dev/vx/dsk/dg02/lvol202"
```

VxVM 패키지 사용자 정의

```
FS[0]="/mnt_dg0101"  
FS[1]="/mnt_dg0102"  
FS[2]="/mnt_dg0201"  
FS[3]="/mnt_dg0202"  
  
FS_MOUNT_OPT[0]="-o ro"  
FS_MOUNT_OPT[1]="-o rw"  
FS_MOUNT_OPT[2]="-o ro"  
FS_MOUNT_OPT[3]="-o rw"
```

4. 추가될 수 있는 모든 사용자 특정 코드를 기존 스크립트에서 복사해야 합니다.
여기에는 환경 변수와 사용자 정의 함수 등이 있습니다.
5. 새 패키지 제어 스크립트를 클러스터의 모든 노드에 배포합니다.
6. 디스크 그룹과 데이터가 변경되지 않았는지 확인합니다.
7. 디스크 그룹을 내보냅니다.

```
# vxdg deport DiskGroupName
```

8. 다른 모든 노드에서 다음 명령을 실행하여 클러스터의 다른 노드에서도 이 디스크 그룹을 인식할 수 있도록 합니다.

```
# vxdctl enable
```

9. 패키지를 다시 시작합니다.

CVM 패키지 사용자 정의

VxVM 디스크 그룹을 만든 후에는 저장 장치에 액세스할 Serviceguard 패키지를 사용자 정의해야 합니다. Cluster Volume Manager(CVM)에서 디스크 그룹을 사용하고 있는 경우 다음 절차를 수행하십시오. VERITAS Volume Manager(VxVM)를 사용하고 있는 경우에는 앞의 절에서 설명한 절차를 수행하십시오.

1. 다음과 같이 기존 패키지 제어 스크립트의 이름을 바꿉니다.

```
# mv Package.ct1 Package.ct1.bak
```

2. 기존 이름과 같은 이름으로 새 패키지 제어 스크립트를 만듭니다.

```
# cmmakepkg -s Package.ct1
```

3. 새로운 CVM 디스크 그룹과 논리 볼륨의 이름을 포함하도록 새 스크립트를 편집합니다.

CVM에 대해 필요한 패키지 제어 스크립트의 새로운 부분은 다음과 같습니다.

- CVM_DG[] 배열은 해당 패키지에서 사용하는 CVM 디스크 그룹을 정의합니다. 다음과 같이 첫 번째 CVM_DG[] 항목의 인덱스는 0이고 두 번째 항목의 인덱스는 1이 되는 식으로 인덱스가 부여되어야 합니다.

```
CVM_DG[0]="dg01"  
CVM_DG[1]="dg02"
```

- LV[], FS[] 및 FS_MOUNT_OPT[] 배열은 LVM에서 사용될 때와 마찬가지로 사용됩니다. LV[]는 논리 볼륨을 정의하고, FS[]는 마운트 지점을 정의하며, FS_MOUNT_OPT[]는 마운트 옵션을 정의합니다. 예를 들어, 위의 두 디스크 그룹에서 두 개의 볼륨을 정의하는 경우 정의된 볼륨이 각각 lvol101, lvol102와 lvol201, lvol202라면 이 볼륨은 각각 /mnt_dg0101, /mnt_dg0102, /mnt_dg0201 및 /mnt_dg0202에 마운트됩니다. /mnt_dg0101과 /mnt_dg0201은 모두 읽기 전용으로 마운트됩니다. 이 경우 LV[], FS[] 및 FS_MOUNT_OPT[] 항목은 다음과 같습니다.

```
LV[0]="/dev/vx/dsk/dg01/lvol101"  
LV[1]="/dev/vx/dsk/dg01/lvol102"  
LV[2]="/dev/vx/dsk/dg02/lvol201"  
LV[3]="/dev/vx/dsk/dg02/lvol202"
```

CVM 패키지 사용자 정의

```
FS[0]="/mnt_dg0101"  
FS[1]="/mnt_dg0102"  
FS[2]="/mnt_dg0201"  
FS[3]="/mnt_dg0202"  
  
FS_MOUNT_OPT[0]="-o ro"  
FS_MOUNT_OPT[1]="-o rw"  
FS_MOUNT_OPT[2]="-o ro"  
FS_MOUNT_OPT[3]="-o rw"
```

4. 추가될 수 있는 모든 사용자 특정 코드를 기존 스크립트에서 복사해야 합니다. 여기에는 환경 변수와 사용자 정의 함수 등이 있습니다.
5. 패키지가 디스크 그룹에서 수행하려는 가져오기의 종류를 지정하기 위해 적절한 CVM_ACTIVATION_CMD 문의 주석을 제거해야 합니다.
6. 새 패키지 제어 스크립트를 클러스터의 모든 노드에 배포합니다.
7. 패키지 ASCII 구성 파일에서 HALT_SCRIPT_TIMEOUT 매개 변수 바로 다음에 각 디스크 그룹을 입력합니다. 각 디스크 그룹에 대한 STORAGE_GROUP 정의를 하나씩 추가합니다. 앞의 예제에 있는 두 디스크 그룹의 경우 다음 줄을 입력할 수 있습니다.

```
STORAGE_GROUP dg01  
STORAGE_GROUP dg02
```

그런 다음 패키지 구성을 다시 적용합니다.

```
# cmapplyconf -P PackageName.ascii
```

8. 디스크 그룹과 데이터가 변경되지 않았는지 확인합니다.
9. 디스크 그룹을 내보냅니다.

```
# vxvgm deport DiskGroupName
```

10. 클러스터가 아직 실행되고 있지 않으면 클러스터를 시작합니다.

```
# cmruncl
```

이렇게 하면 특별한 CVM 패키지가 활성화됩니다.

11. CVM이 시작되면 마스터 노드가 선택됩니다. 이 노드에서 디스크 그룹 구성 명령을 실행해야 합니다. 마스터 노드를 확인하려면 클러스터의 각 노드에서 다음 명령을 실행합니다.

```
# vxctl -c mode
```

하나의 노드가 마스터로 확인됩니다.

12. 마스터 노드에서 다음 명령을 실행하여 클러스터의 다른 노드에서도 이 디스크 그룹을 인식할 수 있도록 합니다.

```
# vxvg -s import DiskGroupName
```

13. 패키지를 다시 시작합니다.

LVM 볼륨 그룹 제거

새 VxVM 디스크 그룹을 검사한 후에는 표준 LVM 명령인 `lvremove`, `pvremove` 및 `vgremove`를 사용하여 시스템에서 더 이상 필요하지 않은 LVM 볼륨 그룹을 제거합니다. 편리한 시간에 클러스터 ASCII 구성 파일을 편집하여 클러스터에서 더 이상 사용되지 않는 LVM 볼륨 그룹을 참조하는 `VOLUME_GROUP` 문도 제거해야 합니다. 다음에 클러스터 구성을 다시 적용하려면 먼저 이러한 항목을 제거해야 합니다.

H IPv6 네트워크 지원

여기에서는 IPv6 네트워크 주소의 다음과 같은 특성에 대해 설명합니다.

- IPv6 주소 유형
- 네트워크 구성 제한 사항
- 로컬 기본/대기 LAN 패턴
- 중복 주소 감지 기능

IPv6 주소 유형

주소 지정 방식에 대한 몇 가지 IPv6 유형이 RFC 2373(IPv6 주소 지정 아키텍처)에 지정되어 있습니다. IPv6 주소는 인터페이스 및 인터페이스 집합에 대한 128비트 식별자입니다. RFC 2373에서 정의한 IPv6의 주소 형식은 다양합니다. IPv6 주소는 크게 다음과 같이 분류됩니다.

다음 표에서는 IPv6 주소 유형 중에서 유니캐스트, 애니캐스트 및 멀티캐스트에 대해 설명합니다.

표 H-1 IPv6 주소 유형

| | |
|-------|---|
| 유니캐스트 | 단일 인터페이스에 대한 주소입니다. 유니캐스트 주소로 전송되는 패킷은 해당 주소로 식별되는 인터페이스로 전달됩니다. |
| 애니캐스트 | 인터페이스 집합에 대한 주소입니다. 대부분의 경우 인터페이스는 서로 다른 노드에 속합니다. 애니캐스트 주소로 전송되는 패킷은 해당 주소로 식별되는 인터페이스 중 하나로 전달됩니다. 애니캐스트 주소 사용에 대한 표준은 개발 단계에 있으므로 HP-UX에서는 아직 지원되지 않습니다. |
| 멀티캐스트 | 일반적으로 서로 다른 노드에 속한 인터페이스 집합에 대한 주소입니다. 멀티캐스트 주소로 전송되는 패킷은 해당 주소로 식별되는 모든 인터페이스로 전달됩니다. |

브로드캐스트 주소의 기능이 멀티캐스트로 대체되었기 때문에 IPv4와 달리 IPv6에는 브로드캐스트 주소가 없습니다.

IPv6 주소의 텍스트 표현

IPv6 주소를 텍스트 문자열로 나타내는 일반적인 형태에는 다음과 같이 세 가지가 있습니다.

- 첫 번째 형태는 `x:x:x:x:x:x:x`입니다. 여기서 'x'는 128비트 주소에서 여덟 개로 구성된 16비트 단위의 16진수 값입니다. 예를 들면 다음과 같습니다.

`2001:fecd:ba23:cd1f:dcb1:1010:9234:4088`

- IPv6 주소 중에는 0비트가 연달아서 길게 붙어 있는 것도 있습니다. 이러한 주소를 텍스트로 쉽게 표현하기 위해 특수 구문을 사용할 수 있습니다. “::”를 사용하면 0으로만 되어 있는 16비트의 그룹이 여러 개 있다는 것을 나타낼 수 있습니다. “::”는 주소에서 한 번만 나타날 수 있고 주소에서 처음, 끝 또는 인접한 16비트의 0을 압축해서 표현할 때 사용될 수 있습니다. 예를 들면 다음과 같습니다.

fec0:1:0:0:0:0:1234는 fec0:1::1234로 표현될 수 있습니다.

- IPv4 및 IPv6 노드가 혼합된 환경에서 사용할 수 있는 IPv6 주소의 형태는 x:x:x:x:d.d.d.d입니다. 여기에서 ‘x’는 IPv6 주소의 상위 96비트의 16진수 값이고 ‘d’는 하위 32비트의 십진수 값입니다. 일반적으로 IPv4 대응 IPv6 주소와 IPv4 호환 IPv6 주소는 이 표기법으로 표현됩니다. 이러한 주소에 대해서는 다음 절에서 설명합니다.

예를 들면 다음과 같습니다.

0:0:0:0:0:10.1.2.3

및

::10.11.3.123

IPv6 주소 접두부

IPv6 주소 접두부는 IPv4의 CIDR과 비슷하며 CIDR 표기법으로 기록됩니다. IPv6 주소 접두부는 다음과 같은 표기법으로 표현됩니다.

IPv6 주소/접두부 길이

여기에서 “IPv6 주소”는 위에서 나열된 모든 표기법의 IPv6 주소이고 “접두부 길이”는 주소의 가장 왼쪽에 있는 연속 비트 몇 개가 접두부를 구성하는지를 나타내는 십진수 값입니다. 예를 들면 다음과 같습니다.

fec0:0:0:1::1234/64

주소의 처음 64비트에 해당하는 “fec0:0:0:1”이 주소 접두부입니다. 주소 접두부는 IPv6 주소에서 몇 비트가 서브넷을 나타내는지를 표시하기 위해 사용됩니다.

유니캐스트 주소

IPv6 유니캐스트 주소는 여러 유형으로 분류됩니다. 유니캐스트 주소에는 결합 가능한 글로벌 유니캐스트 주소, 사이트 로컬 주소 및 링크 로컬 주소가 있습니다. 일반적으로 유니캐스트 주소는 다음과 같이 논리적으로 구분됩니다.

표 H-2

| n 비트 | 128-n 비트 |
|---------|----------|
| 서브넷 접두부 | 인터페이스 ID |

IPv6 유니캐스트 주소의 인터페이스 식별자는 링크에서 인터페이스를 식별하는 데 사용됩니다. 인터페이스 식별자는 해당 링크에서 고유해야 합니다. 일반적으로 링크는 서브넷 접두부로 식별됩니다.

유니캐스트 주소의 모든 비트가 0이면 지정되지 않은 주소라고 합니다. 이것은 “::”를 사용하여 텍스트로 표현됩니다.

유니캐스트 주소 “::1” 또는 “0:0:0:0:0:0:0:1”을 루프백 주소라고 합니다. 이 주소는 노드가 패킷을 노드 자신에게 보낼 때 사용됩니다.

IPv4 및 IPv6 호환성

IPv6 주소 지정 체계에서 IPv4 주소를 사용할 수 있는 방법에는 여러 가지가 있습니다.

IPv4 호환 IPv6 주소

IPv6 변환 메커니즘에서는 기존 IPv4 하부 구조를 통해 IPv6 패킷을 터널링하는 기법을 사용합니다. 이러한 메커니즘을 지원하는 IPv6 노드는 하위 32비트로 IPv4 주소를 전달하는 특수한 유형의 IPv6 주소를 사용합니다. 이러한 주소를 IPv4 호환 IPv6 주소라고 하며, 다음과 같이 표현합니다.

표 H-3

| 80비트 | 16비트 | 32비트 |
|------|------|---------|
| 모두 0 | 0000 | IPv4 주소 |

예를 들면 다음과 같습니다.

::192.168.0.1

IPv4 대응 IPv6 주소

IPv4 주소를 포함하는 특별한 유형의 IPv6 주소가 있습니다. 이 주소는 IPv6 주소로 IPv4만 사용하는 노드의 주소를 나타내는 데 사용됩니다. 이러한 주소는 특히 IPv6과 IPv4를 모두 지원하는 응용 프로그램에서 사용됩니다. 이러한 주소를 IPv4 대응 IPv6 주소라고 하며, 형식은 다음과 같습니다.

표 H-4

| 80비트 | 16비트 | 32비트 |
|------|------|---------|
| 모두 0 | FFFF | IPv4 주소 |

예를 들면 다음과 같습니다.

```
::ffff:192.168.0.1
```

결합 가능한 글로벌 유니캐스트 주소

글로벌 유니캐스트 주소는 전역적으로 고유한 IPv6 주소입니다. 이 주소 형식은 RFC 2374(IPv6 결합 가능한 글로벌 유니캐스트 주소 형식)에서 매우 잘 정의되어 있습니다. 형식은 다음과 같습니다.

표 H-5

| 3 | 13 | 8 | 24 | 16 | 64비트 |
|----|--------|-----|--------|--------|----------|
| FP | TLA ID | RES | NLA ID | SLA ID | 인터페이스 ID |

여기서

FP(Format Prefix) = 형식 접두부. 결합 가능한 글로벌 유니캐스트 주소의 경우 이 값은 "001"입니다.

TLA ID(Top-level Aggregation Identifier) = 최상위 수준 집합 식별자

RES = 이후 사용을 위해 예약됨

NLA ID(Next-Level Aggregation Identifier) = 다음 수준 집합 식별자

SLA ID(Site-Level Aggregation Identifier) = 사이트 수준 집합 식별자

인터페이스 ID = 인터페이스 식별자

링크 로컬 주소

링크 로컬 주소의 형식은 다음과 같습니다.

표 H-6

| 10비트 | 54비트 | 64비트 |
|------------|------|----------|
| 1111111010 | 0 | 인터페이스 ID |

링크 로컬 주소는 단일 링크에서 노드에 주소를 지정하는 데 사용됩니다. 링크 로컬 송신지나 수신지 주소를 가진 패킷은 라우터에 의해 전달되지 않습니다.

사이트 로컬 주소

사이트 로컬 주소의 형식은 다음과 같습니다.

표 H-7

| 10비트 | 38비트 | 16비트 | 64비트 |
|------------|------|--------|----------|
| 1111111011 | 0 | 서브넷 ID | 인터페이스 ID |

사이트 로컬 주소는 사이트 안에서 사용됩니다. 사이트 외부에 있는 사이트 로컬 송신지나 수신지 주소를 가진 패킷은 라우터에 의해 전달되지 않습니다.

멀티캐스트 주소

멀티캐스트 주소는 노드 그룹에 대한 식별자이며 형식은 다음과 같습니다.

표 H-8

| 8비트 | 4비트 | 4비트 | 112비트 |
|----------|-----|-----|-------|
| 11111111 | 플래그 | 범위 | 그룹 ID |

주소의 처음에 나오는 “FF”는 해당 주소를 멀티캐스트 주소로 식별합니다.

“플래그” 필드는 네 개의 플래그 집합인 “000T”입니다. 상위 3비트는 예약되었으므로 0이 되어야 합니다. 마지막 비트 “T”는 영구적으로 할당되었는지 여부를 나타냅니다. 0 값은 영구적으로 할당된 것을 나타내고, 0이 아닌 값은 임시 할당을 나타냅니다.

“범위” 필드는 멀티캐스트 그룹의 범위를 제한하는 데 사용되는 4비트 필드입니다. 예를 들어, “1” 값은 노드 로컬 멀티캐스트 그룹을 나타냅니다. “2” 값은 범위가 링크 로컬임을 나타냅니다.

“그룹 ID” 필드는 멀티캐스트 그룹을 식별합니다. 자주 사용되는 멀티캐스트 그룹은 다음과 같습니다.

모든 노드 주소 = FF02:0:0:0:0:0:0:1(링크 로컬)

모든 라우터 주소 = FF02:0:0:0:0:0:0:2(링크 로컬)

모든 라우터 주소 = FF05:0:0:0:0:0:0:2(사이트 로컬)

네트워크 구성 제한 사항

Serviceguard에서는 현재 데이터 링크에 대해서만 IPv6을 지원합니다. 하트비트 IP는 여전히 IPv4여야 하지만 패키지 IP는 IPv4 또는 IPv6일 수 있습니다.

IPv6을 구성하려면 IPv6 제품 번들(IPv6NCF11i B.11.11.0109.5C)이 설치되어야 하는 이중 스택 구성에서 시스템을 설정해야 합니다. 이 번들을 설치한 후에 필요한 추가 패치 목록은 이 설명서의 “필수 패치 및 권장 패치” 부분에 나와 있습니다.

Serviceguard에서의 IPv6 지원에 대한 제한 사항은 다음과 같습니다.

- 하트비트 IP 주소는 IPv4여야 합니다. 따라서 IPv6만 있는 작업 노드 또는 IPv6만 있는 노드는 Serviceguard 환경에서 지원되지 않습니다.
- Serviceguard 구성에서 호스트 이름은 IPv4여야 합니다. Serviceguard에서는 IPv6 호스트 이름을 인식하지 않습니다.
- 자동 구성된 IPv6 주소는 Serviceguard에서 STATIONARY_IP 주소로 지원되지 **않습니다**. Serviceguard 클러스터 구성에 속한 모든 IPv6 주소는 라우터 알립 등을 통해 자동 구성되어서는 **안 되며**, /etc/rc.config.d/netconf-ipv6에서 수동으로 구성해야 합니다.
- 링크 로컬 IP 주소는 패키지 IP나 STATIONARY_IP로 지원되지 **않습니다**. 패키지 IP는 요구 사항에 따라 사이트 로컬 또는 글로벌 유형일 수 있습니다.
- Serviceguard에서는 각 네트워크 인터페이스(제한된 멀티 네팅)에 대해 각 범위 유형(사이트 로컬 및 글로벌)에 속한 하나의 IPv6 주소만 지원합니다. 따라서 NETWORK_INTERFACE에 대한 클러스터 ASCII 파일에는 최대 두 개의 IPv6 STATIONARY_IPS를 포함할 수 있습니다. 이 중 하나는 사이트 로컬 IPv6 주소가 되고 다른 하나는 글로벌 IPv6 주소가 됩니다.
- 퀘림 서버를 사용하는 경우 퀘림 서버는 IPv4 네트워크에서만 구성되어야 합니다. IPv6은 사용할 수 없습니다. IPv4 네트워크에서 구성된 퀘림 서버는 IPv6 네트워크를 해당 클러스터 구성의 일부로 갖는 Serviceguard IPv6 클러스터에서도 사용할 수 있습니다.
- Serviceguard에서는 10BT, 100BT 및 기가비트 이더넷을 포함한 이더넷 네트워크에서 IPv6만 지원합니다.

주

링크 로컬 IP 주소가 Serviceguard 클러스터 구성에서 지원되지 않지만 Serviceguard 기본 인터페이스에서 기본 링크 로컬 주소는 로컬 전환 중에 대기 인터페이스로 전환됩니다. 이러한 현상은 두 가지 요구 사항 때문에 발생합니다. 첫 번째 요구 사항은 이중 스택(IPv4/IPv6) 커널에서 인터페이스와 관련된 기본 IP 주소는 항상 링크 로컬 주소여야 한다는 것이고, 두 번째 요구 사항은 Serviceguard에서는 사이트 로컬 및 글로벌 IP가 대기 네트워크 인터페이스로 전환되어야 한다는 것입니다.

IPv6 변동 가능 주소 및 중복 주소 감지 기능

IPv6 네트워크 스택에는 이전에 IPv4에서는 사용할 수 없었던 DAD(중복 주소 감지)라는 새로운 기능이 있습니다. 주소가 추가되면 이 DAD가 네트워크에서 이미 사용되고 있는 중복 주소를 감지합니다. DAD는 네트워크 환경에 멀티캐스트 메시지를 보내고, 다른 노드로부터 응답을 수신하는 데 1초 이상 걸립니다. 해당 시간 안에 응답이 없으면 변동 가능 IPv6 주소를 사용할 수 있다고 간주합니다. 이 기능에 대한 자세한 내용은 RFC 2462를 참조하십시오.

Serviceguard에서는 이 기능으로 인해 각 IPv6 재배치 가능 주소를 추가하는 데 걸리는 시간이 해당 IPv4 주소를 추가하는 시간보다 1초 이상 더 걸리게 됩니다. 패키지에 구성되는 IPv6 주소의 개수에 따라 패키지 시작 시간이 느려질 수 있습니다.

중복 주소 감지가 필요하지 않으면 커널 매개 변수 `ip6_nd_dad_solicity_count`를 0으로 설정하여 DAD 기능을 해제할 수 있습니다. 이 커널 매개 변수는 전체 시스템에 적용되므로 DAD 기능을 해제하면 시스템에 있는 모든 응용 프로그램에 대해 이 기능을 사용할 수 없습니다. DAD가 필요하지 않은 시스템의 경우 이 기능을 해제하면 많은 수의 IPv6 변동 가능 주소가 포함된 패키지의 시작 시간이 상당히 빨라질 수 있습니다.

시스템의 현재 DAD 상태를 보려면 `ndd -get` 명령을 사용하여 커널 매개 변수의 현재 값을 확인합니다.

```
# ndd -get /dev/ip6 ip6_nd_dad_solicit_count
```

결과가 1이면 이 기능이 설정되어 있는 것이고 결과가 0이면 해제되어 있는 것입니다.

컴퓨터에서 DAD의 상태를 일시적으로 변경하려면 `ndd -set` 명령을 사용하여 커널 매개 변수를 변경합니다.

```
# ndd -set /dev/ip6 ip6_nd_dad_solicit_count n
```

여기서 n 은 숫자를 나타내며 1은 기능을 설정하고 0은 기능을 해제합니다.

컴퓨터에서 DAD의 상태를 변경하고 컴퓨터를 다시 부팅한 후에도 변경된 상태를 유지하도록 하려면 `/etc/rc/config.d/nddconf` 파일에 다음 항목을 추가합니다.

```
# TRANSPORT_NAME[index]=ip6  
  
# NDD_NAME[index]=ip6_nd_dad_solicit_count  
  
# NDD_VALUE[index]=n
```

여기서 $index$ 는 `nddconf` 파일에서 다음으로 사용 가능한 정수 값이고 n 은 숫자입니다. 1은 기능을 설정하고 0은 기능을 해제합니다.

로컬 기본/대기 LAN 패턴

IPv6을 사용하면 클러스터에 구성된 LAN 카드 중에서 여러 가지 장애 조치 패턴이 가능합니다. 이것은 이중 IPv4/IPv6 구성이 사용될 때 각 LAN 카드가 몇 가지 IP 주소를 지원할 수 있기 때문에 가능합니다. 이 절에서는 대기 LAN으로의 로컬 장애 조치를 구성할 수 있는 몇 가지 방법에 대해 설명합니다.

대기 네트워크 인터페이스는 두 주소 계열(IPv4 또는 IPv6)의 IP 주소가 없는 인터페이스이며 노드에서 주 네트워크 인터페이스와 브리지 연결되어 있습니다.

로컬 장애 조치 상황에서 IPv4 및 IPv6 주소를 사용할 때는 다음과 같이 명심해야 할 두 가지 지침이 있습니다.

- 네트워크 인터페이스 카드는 기본 IP로 IPv4 및 IPv6 주소를 모두 사용할 수 있으므로 대기 네트워크 인터페이스 카드는 두 유형의 기본 인터페이스에 대한 대기 인터페이스의 역할을 할 수 있습니다.

그러나 IPv4 및 IPv6 주소가 두 가지 다른 네트워크 인터페이스에서 구성되어 있으면 로컬 장애 조치 중 대기 인터페이스는 네트워크 인터페이스 하나에서만 IP 주소를 넘겨받을 수 있습니다.

즉, 두 가지 다른 네트워크 인터페이스에서 IPv4 및 IPv6 주소는 장애 조치 상황에서 상호 배타적입니다.

- Serviceguard에서는 기본 네트워크 인터페이스에서 구성된 링크 로컬 주소와 클러스터 구성의 일부로 구성된 다른 모든 IP 주소를 대기 네트워크 인터페이스로 전환합니다. 여기에는 Serviceguard에 의해 추가된 패키지 IP(IPv4 및 IPv6), 고정 IP(IPv4 및 IPv6) 및 하트비트가 모두 포함됩니다.

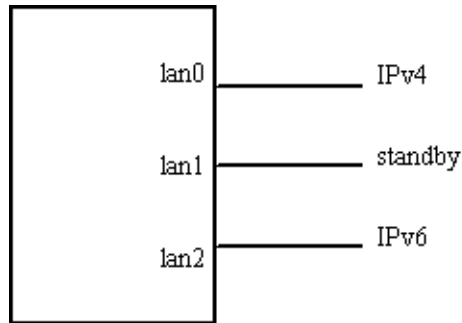
이것을 나타내는 예제는 다음과 같습니다.

구성 예제

IPv4 및 IPv6 주소를 모두 사용하는 클러스터 노드의 LAN 구성 예제는 다음과 같습니다.

그림 H-1

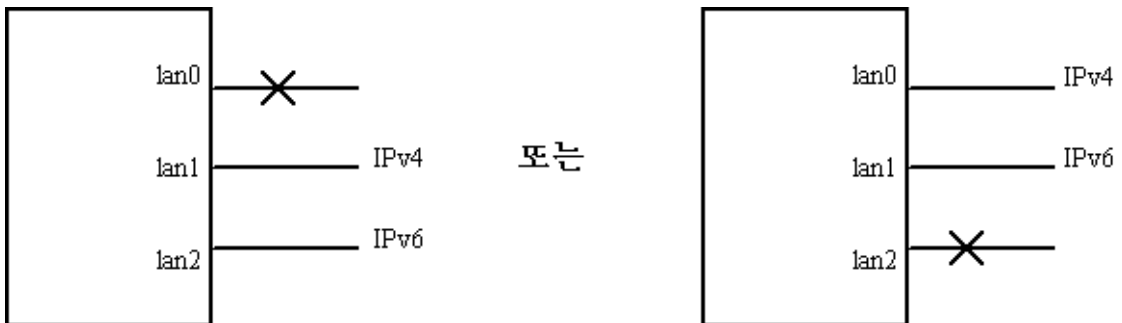
예제 1: 대기 구성의 IPv4 및 IPv6 주소



lan0 또는 lan2가 손실되고 나면 아래와 같이 lan1에서 주소를 받을 수 있습니다.

그림 H-2

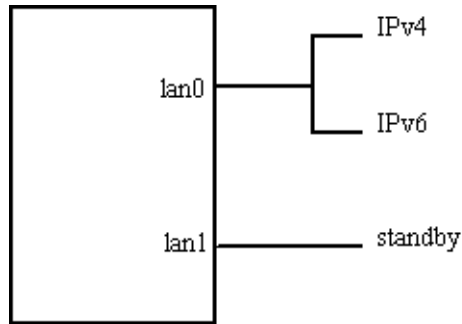
예제 1: 대기 LAN으로 장애 조치된 후의 IPv4 및 IPv6 주소



아래와 같이 IPv4 및 IPv6 주소를 모두 사용하도록 동일한 LAN 카드를 구성할 수 있습니다.

그림 H-3

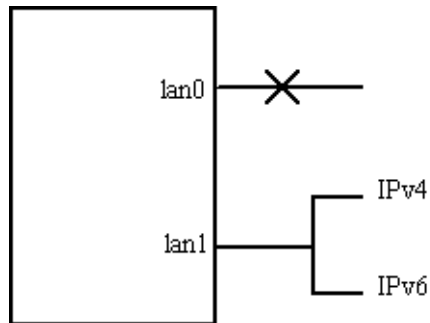
예제 2: 대기 구성의 IPv4 및 IPv6 주소



이러한 구성에서는 두 주소가 대기 LAN으로 장애 조치될 수 있습니다. 이에 대한 그림이 아래에 있습니다.

그림 H-4

예제 2: 대기 LAN으로 장애 조치된 후의 IPv4 및 IPv6 주소



중복 주소 감지 기능

IPv6 네트워킹 스택에는 이전에 IPv4에서는 사용할 수 없었던 DAD(중복 주소 감지)라는 새로운 기능이 있습니다. 주소가 추가되면 이 DAD가 네트워크에서 이미 사용되고 있는 중복 주소를 감지합니다. DAD는 네트워크 환경에 멀티캐스트 메시지를 보내고, 다른 노드로부터 응답을 수신하는 데 1초 이상 걸립니다. 해당 시간 안에 응답이 없으면 변동 가능 IPv6 주소를 사용할 수 있다고 간주합니다. 이 기능에 대한 자세한 내용은 RFC 2462를 참조하십시오.

Serviceguard에서 이 기능을 사용하면 각각의 IPv6 변동 가능 주소를 추가하는 데 필요한 시간이 해당 IPv4 주소를 추가할 때보다 적어도 1초 이상 더 필요합니다. 패키지에 구성된 IPv6 주소의 개수에 따라 패키지 시작 시간이 느려질 수 있습니다.

중복 주소 감지가 필요하지 않으면 커널 매개 변수 `ip6_nd_dad_solicity_count`를 0으로 설정하여 DAD 기능을 해제할 수 있습니다. 이 커널 매개 변수는 전체 시스템에 적용되므로 DAD 기능을 해제하면 시스템에 있는 모든 응용 프로그램에 대해 이 기능을 사용할 수 없습니다. DAD가 필요하지 않은 시스템의 경우 이 기능을 해제하면 많은 수의 IPv6 변동 가능 주소가 포함된 패키지의 시작 시간이 상당히 빨라질 수 있습니다.

시스템의 현재 DAD 상태를 보려면 `ndd -get` 명령을 사용하여 커널 매개 변수의 현재 값을 확인합니다.

```
# ndd -get /dev/ip6 ip6_nd_dad_solicit_count
```

결과가 1이면 이 기능이 설정되어 있는 것이고 결과가 0이면 해제되어 있는 것입니다.

시스템에서 DAD의 상태를 변경하려면 `ndd -set` 명령을 사용하여 커널 매개 변수를 변경합니다.

```
# ndd -set /dev/ip6 ip6_nd_dad_solicit_count n
```

여기서 `n`은 숫자를 나타내며 1은 기능을 설정하고 0은 기능을 해제합니다.

컴퓨터에서 DAD의 상태를 영구적으로 변경하려면 `/etc/rc/config.d/nddconf` 파일에 다음 항목을 추가합니다.

```
# TRANSPORT_NAME[index]=ip6
```

```
# NDD_NAME[index]=ip6_nd_dad_solicit_count
```

```
# NDD_VALUE [index]=n
```

여기서 *index* 는 `nddconf` 파일에서 다음으로 사용 가능한 정수 값이고 *n*은 숫자입니다. **1**은 기능을 설정하고 **0**은 기능을 해제합니다.

IPv6 네트워크 지원
중복 주소 감지 기능

숫자

8개의 노드로 구성된 활성/대기 클러스터
그림, 55

ㄱ

개입

사용자 개입 최소화, 366

검사점, 371

계획

개요, 123

고가용성 목표, 125

디스크 그룹 및 디스크, 144

디스크 입출력 정보, 133

물리 볼륨 계획용 워크시트, 404

볼륨 그룹 및 물리 볼륨, 141

워크시트, 134

전원, 136

쿼럼 서버, 139

클러스터 관리자 구성, 148

클러스터 구성, 146

클러스터 잠금 및 클러스터 확장, 148

패키지 구성, 157

하드웨어 구성, 127

확장 고려, 160

LAN 정보, 128

SCSI 주소, 132

SPU 정보, 128

계획 워크시트

양식, 399

계획된 중단 시간 최소화, 384

계획용 워크시트

양식, 399

고가용성, 24

계획 중인 목표, 125

정의된 HA 클러스터, 36

고가용성 디스크 어레이를 사용하는 클러스터

그림, 49, 50

고가용성을 위해 연결된 미러링된 디스크

그림, 48

고정 IP 주소, 96

공유 디스크

계획, 133

공유 버스에서 루트 디스크 제한, 50

관리

구성 파일 검토, 338

구성 파일 확인, 338

모든 노드가 중지되었을 때 클러스터 시작, 292

문제 해결, 335

실행 중인 클러스터에 노드 추가, 294

실행 중인 클러스터에서 노드 제거, 294

전체 클러스터 중지, 295

클러스터, 292

클러스터 및 패키지 상태, 278

클러스터 오프라인 상태에서 패키지 재구성, 310

클러스터 이벤트에 대응, 316

클러스터 재구성, 303

클러스터가 실행 중일 때 패키지 재구성, 311

패키지 및 서비스, 297

패키지 시작, 297

패키지 이동, 298

패키지 중지, 298

구성

기본 작업 및 단계, 33

서비스, 243

클러스터, 62

클러스터 계획, 146

패키지, 243

패키지 계획, 157

ASCII 클러스터 구성 파일 템플릿, 215

ASCII 패키지 구성 파일 템플릿, 246

구성 파일

문제 해결, 338

클러스터 관리자, 62

구성원 변경

이유, 64

그림

8개의 노드로 구성된 활성/대기 클러스터, 55

고가용성 디스크 어레이를 사용하는 클러스터, 49,
50

고가용성을 위해 연결된 미러링된 디스크, 48

노드 1로 이동한 패키지를 포함하는 클러스터 실행
, 396

노드 1을 HP-UX 10.01로 업그레이드, 395

노드 1이 클러스터에 다시 참가, 396

노드 2로 이동한 패키지를 포함하는 클러스터 실행
, 395

다른 공유 버스의 루트 디스크, 51

다른 버스의 기본 디스크 및 미러, 52

롤링 업그레이드 전의 클러스터 실행, 394

업그레이드 후의 클러스터 실행, 397

이중 접속 FDDI 스테이션으로 구성, 41

- 일반적인 클러스터 구성, 23
- 장애 조치 후의 일반적인 클러스터, 25
- 중복 FDDI 구성, 40
- 중복 LAN, 39
- 직렬(RS232) 하트비트 회선, 42
- 클러스터 구성 예, 127
- EMC 디스크 어레이를 사용한 8개의 노드로 구성된 클러스터, 56
- MC/ServiceGuard 소프트웨어 구성 요소, 58
- MC/ServiceGuard 클러스터 구성 작업, 33
- 기본 네트워크 인터페이스, 38
- 기본 노드, 25
- 기본 LAN 인터페이스 정의, 38

ㄴ

- 네트워크
 - 네트워크 응용 프로그램을 HA 서비스로 작성, 367
 - 노드 및 패키지 IP 주소, 96
 - 로컬 인터페이스 전환, 98
 - 로컬 전환, 100
 - 원격 시스템 전환, 102
 - 장애, 100
 - 중복, 38, 42
 - 패키지 IP 주소 추가 및 삭제, 97
 - 포트 주소에 바인딩, 377
 - IP 주소 및 이름 지정, 373
 - IP 주소로 로드 공유, 97
 - IP 주소를 사용하는 패키지, 375
 - IP 주소에 바인딩, 377
 - MC/ServiceGuard에서 지원되는 종류, 38
- 네트워크 계획
 - 서브넷, 139
- 네트워크 관리자
 - 기본 기능, 96
 - 테스트, 321
 - 패키지 IP 주소 추가 및 삭제, 97
 - LAN 인터페이스 모니터링, 97
- 네트워크 관리자의 작동 방식, 96
- 네트워크 구성 요소
 - MC/ServiceGuard, 38
- 네트워크 시간 프로토콜(NTP) 클러스터, 193
- 네트워크 인터페이스의 중복, 38
- 네트워크 장애 감지 매개 변수, 37

- 네트워크 통신 장애, 121
- 네트워크 폴링 간격 (NETWORK_POLLING_INTERVAL) 클러스터 관리자 구성의 매개 변수, 152
- 네트워킹
 - 중복 서브넷, 129
- 네트워킹 계획
 - 서브넷, 129

- 노드
 - 기본 개념, 36
 - IP 주소, 96
 - MC/ServiceGuard 클러스터, 24

- 노드 종류
 - 주, 25
 - 활성, 25
- 논리 볼륨
 - 계획, 141
 - 계획 워크시트 양식, 407
 - 워크시트, 142, 144
 - 클러스터에 대해 만들기, 202, 211, 233
 - 하부 구조 만들기, 199, 209

ㄷ

- 다른 공유 버스의 루트 디스크
 - 그림, 51
- 다른 버스의 기본 디스크 및 미러
 - 그림, 52
- 다수의 저장 장치를 위한 패키지 최적화, 260
- 다중 시스템
 - 응용 프로그램 설계, 373
- 다중 시스템에서 실행할 응용 프로그램 설계, 373
- 단순 장애 요인(Single Point of Failure)
 - 방지, 24
- 단일 노드 작동, 240
- 단일 클러스터 잠금
 - 선택, 67
- 대기 네트워크 인터페이스, 38
- 대기 LAN 인터페이스
 - 정의, 38
- 대응
 - 클러스터 이벤트, 316
 - 패키지 및 서비스 장애, 120
- 대체 노드, 25
- 대체 링크
 - 볼륨 그룹 만들기, 203

- 데이터, 108
 - 디스크, 44
- 데이터 미리 사본
 - 디스크 장애로부터 보호, 24
- 데이터 정제, 62
- 데이터베이스
 - 도구 키트, 363
- 도구 키트
 - 데이터베이스 용, 363
- 동적 클러스터 재편성, 64
- 디스크
 - 교체, 325
 - 구성 예제, 47, 49
 - 데이터, 44
 - 루트, 44
 - 미러링, 45
 - 볼륨 그룹에 포함하기 위해 선택, 200
 - 인터페이스, 44
 - MC/ServiceGuard, 44
 - MC/ServiceGuard에서 지원되는 종류, 44
- 디스크 관리, 108
- 디스크 교체, 325
- 디스크 구성 예제, 47, 49
- 디스크 그룹
 - 계획, 144
- 디스크 그룹 및 디스크 계획, 144
- 디스크 논리 단위
 - 하드웨어 계획, 133
- 디스크 레이아웃
 - 계획, 141
- 디스크 모니터, 46
- 디스크 모니터(EMS), 82
- 디스크 미러링, 45
- 디스크, 미러링, 45
- 디스크 어레이
 - PV 링크를 사용하여 볼륨 그룹 만들기, 203
- 디스크 어레이를 사용한 8개의 노드로 구성된 클러스터
 - 그림, 56
- 디스크 엔클로저
 - 고가용성, 46
 - 고장난 메커니즘 교체, 325
- 디스크 입출력
 - 하드웨어 계획, 133
- 디스크 장애
 - 미러링을 통한 보호, 24

- 디스크 저장 공간
 - CVM을 사용하여 하부 구조 만들기, 230

크

- 로컬 전환, 98
 - 패키지 구성의 매개 변수, 163
- 롤링 소프트웨어 업그레이드, 391
 - 단계, 392
 - 예제, 394
- 롤링 업그레이드
 - 제한 사항, 398
- 루트 미리
 - LVM으로 만들기, 190
- 리소스
 - 디스크, 44
- 리소스 이름
 - 패키지 구성의 매개 변수, 167, 168
- 리소스 작동 간격
 - 패키지 구성의 매개 변수, 168
- 리소스 폴링 간격
 - 패키지 구성의 매개 변수, 168
- 링크 수준 주소, 375

크

- 마운트 옵션
 - 제어 스크립트, 171, 172
- 마운트 해제 횟수, 172
- 매개 변수
 - 장애 조치, 83
 - AUTO_RUN, 257
 - NODE_FAILFAST_ENABLED, 257
- 맨페이지
 - MC/ServiceGuard에서 사용 가능한 페이지 목록, 351
- 메모리 요구 사항
 - MC/ServiceGuard용 잠금 가능 메모리, 125
- 메모리 용량
 - 하드웨어 계획, 128
- 모니터링되는 비 하트비트 서브넷
 - 클러스터 관리자 구성의 매개 변수, 150
- 모니터링된 리소스 장애
 - MC/ServiceGuard 동작, 36
- 문제 해결, 342
 - 디스크 교체, 325
 - 방법, 335

- 시스템 로그 파일 확인, 336
- 제어 스크립트 확인, 338
- 패키지 IP 주소 확인, 335
- 하드웨어 모니터링, 323
- cmquerycl 및 cmcheckconf 사용, 339
- 물리 볼륨
 - 계획, 141
 - 계획 워크시트 채우기, 404
 - 워크시트, 142, 144
 - 클러스터 잠금, 66
 - 클러스터 잠금 구성의 매개 변수, 149
 - 클러스터에 대해 만들기, 201
- 미러링
 - 디스크, 45
- ㅂ
- 바인딩
 - 네트워크 응용 프로그램, 377
- 버스 종류
 - 하드웨어 계획, 133
- 변동 가능 IP 주소
 - 정의, 96
 - MC/ServiceGuard 패키지, 96
- 보안
 - editing files, 182
- 복구 시간, 146
- 볼륨 관리자, 108
 - 비교, 116
 - CVM, 114
 - LVM, 113
 - LVM에서 VxVM으로 마이그레이션, 413
 - VxVM, 113
- 볼륨 그룹
 - 계획, 141
 - 다른 노드로 내보내기 전에 비활성화, 205
 - 워크시트, 142, 144
 - 제어 스크립트, 171, 172
 - 클러스터 잠금, 66
 - 클러스터에 대해 만들기, 201, 203
 - 클러스터의 물리 볼륨 만들기, 201
 - LVM 명령을 사용하여 다른 노드에 설정, 205
- 볼륨 그룹 및 물리 볼륨 계획, 141
- 볼륨 그룹 비활성화, 205
- 브리지 연결 네트워크
 - 네트워크 인터페이스의 중복, 38
- 정의, 38
- ㅅ
- 사용자 정의 함수
 - 제어 스크립트에 추가, 269
- 상태
 - 시스템 로그 파일, 336
 - 클러스터 및 패키지, 278
 - 패키지 IP 주소, 335
- 서브넷
 - 패키지 구성의 매개 변수, 167
 - 하드웨어 계획, 129, 139
- 서비스 관리, 297
- 서비스 구성
 - 단계별, 243
- 서비스 명령
 - 패키지 제어 스크립트의 변수, 175, 176
- 서비스 이름, 174
 - 패키지 구성의 매개 변수, 166
 - 패키지 제어 스크립트의 변수, 174
- 서비스 장애
 - 내용, 120
- 서비스 장애 조치
 - 패키지 구성의 매개 변수, 166
- 서비스 재시작, 121
 - 패키지 제어 스크립트의 변수, 175
- 서비스 중지 시간 종료
 - 패키지 구성의 매개 변수, 167
- 성능
 - 다수의 저장 장치를 위한 패키지 최적화, 260
- 세부 상태
 - 클러스터 및 패키지, 278
- 소개
 - MC/ServiceGuard 소개, 24
 - MC/ServiceGuard 소프트웨어 이해, 57
 - MC/ServiceGuard 하드웨어 이해, 35
- 소프트웨어 계획
 - CVM 및 VxVM, 144
 - LVM, 141
- 소프트웨어 설치
 - 쿼럼 서버, 196
 - MC/LockManager, 198
- 소프트웨어 장애
 - MC/ServiceGuard 동작, 36

순환 대기

- 장애 조치 정책으로 구성, 76
- 패키지 정책 설정, 76
- 시스템 다중 노드 패키지, 71
- CVM과 함께 사용, 231
- 시스템 로그 파일
- 문제 해결, 336
- 시스템 메시지
- 클러스터에 대해 변경, 239
- 시스템에서 MC/ServiceGuard 제거, 317
- 시작 및 종료
- 응용 프로그램에 대해 정의, 367
- 실행 중인 클러스터
- 패키지 추가 또는 제거, 255
- 실행 중인 클러스터 재구성, 304
- 실행 중인 클러스터에 노드 추가, 294
- 실행 중인 클러스터에 패키지 추가, 255, 311
- 실행 중인 클러스터에서 노드 제거, 294
- 실행 중인 클러스터에서 패키지 삭제, 312
- 실행 중인 클러스터에서 패키지 제거, 255

o

- 액세스 역할, 153
- 액세스 제어 정책, 153, 169
- 어레이
- 고장난 메커니즘 교체, 325
- 데이터 보호를 위한 디스크 어레이, 45
- 엔클로저
- 고가용성, 46
- 온라인 하드웨어 유지 관리
- 인라인 SCSI 단자 사용, 326
- 워크시트
- 계획에 사용, 123
- 물리 볼륨 계획, 404
- 볼륨 그룹 및 물리 볼륨, 142, 144
- 전원 공급 구성, 137
- 쿼럼 서버 구성, 140
- 클러스터 구성, 155
- 패키지 구성 데이터, 169
- 패키지 제어 스크립트, 176
- 하드웨어 구성, 134
- 원격 전환, 102
- 원시 볼륨, 369
- 유동 IP 주소
- 정의, 96

MC/ServiceGuard 패키지, 96

- 응용 프로그램
 - 네트워크용 HA 서비스 작성, 367
 - 자동화, 366
 - 장애 처리, 382
 - MC/ServiceGuard와 통합하기 위한 단계의 검사 목록, 387
 - 응용 프로그램 작동 자동화, 366
 - 응용 프로그램 장애 처리, 382
 - 응용 프로그램에서 클라이언트 연결 복원, 380
 - 응용 프로그램의 장애 조치 속도 제어, 368
 - 이더넷
 - 중복 구성, 38
 - 이름 확인 서비스, 188
 - 이벤트 모니터링 서비스
 - 사용, 81
 - 이중 접속 FDDI 스테이션, 41
 - 이중 접속 FDDI 스테이션으로 구성
 - 그림, 41
 - 이중 클러스터 잠금
 - 선택, 68
 - 인라인 단자
 - 온라인 하드웨어 유지 관리 허용, 326
 - 인터넷
 - 도구 키트, 363
 - 일반 계획, 125
 - 일반적인 클러스터 구성
 - 그림, 23
 - 입출력 버스 주소
 - 하드웨어 계획, 133
 - 입출력 슬롯
 - 하드웨어 계획, 128, 133
- ## ㅈ
- 자동 시작 지연
 - 클러스터 구성 파일의 매개 변수, 152
 - 자동 장애 복구
 - 장애 조치 정책으로 구성, 78
 - 자동 전환
 - 패키지 구성의 매개 변수, 162
 - 자동 포트 통합
 - 정의, 103
 - 자동으로 클러스터 재시작, 296
 - 잠금
 - 쿼럼 서버 사용, 68

클러스터 잠금 디스크 사용, 66
클러스터 잠금 및 전원 공급 장치, 53

잠금 디스크
4개 이상의 노드, 67

잠금 볼륨 그룹
계획, 147
구성 파일에서 지정, 221

잠금 볼륨 그룹 재구성, 303

잠금 볼륨 그룹, 재구성, 303

장애
네트워크 통신, 121
대응 유형, 119
응용 프로그램, 382
장애 후 서비스 재시작, 121
패키지 및 서비스 장애에 대한 대응, 120
하드웨어 장애에 대한 대응, 120

장애 감지
네트워크 관리자, 97

장애 복구 정책
패키지 관리자가 사용, 78
패키지 구성 파일 매개 변수, 162

장애 요인
네트워킹, 38, 42

장애 조치
응용 프로그램의 속도 제어, 368
정의, 25

장애 조치 동작
패키지, 83

장애 조치 정책
패키지 관리자가 사용, 75
패키지 구성 매개 변수, 161

장애 조치 패키지, 71

장애 조치 후의 일반적인 클러스터
그림, 25

장애에 대한 대응, 119

재시도 횟수, 172

재시작
장애 후, 121
클러스터 자동 재시작, 64
패키지 제어 스크립트의 SERVICE_RESTART 변
수, 175

재시작 가능한 트랜잭션, 370

재편성
클러스터, 64

재편성 시간, 147

저장 장치 관리, 108

저장 장치에 지점 간 연결, 55

전원 계획
워크시트, 137
전원 장치, 136

전원 공급 장치
계획 워크시트 양식, 401, 402
클러스터 잠금, 53
HP-UX의 OPS용 UPS, 53

전체 클러스터 재구성, 303

전체 클러스터 중지, 295

전환
로컬 인터페이스 전환, 98
윌격 시스템 전환, 102
전환 후의 ARP 메시지, 103

제어 스크립트
명령으로 작성, 256
문제 해결, 338
사용자 정의 함수 추가, 269
추가 제품 지원, 271
패키지 구성, 256
패키지 구성의 경로 이름 매개 변수, 164
SAM으로 만들기, 256

종료 및 시작
응용 프로그램에 대해 정의, 367

주소 지정, SCSI, 132

중단 시간
계획된 최소화, 384

중복
네트워킹, 38, 42
클러스터 구성 요소, 36

중복 네트워크
하트비트, 24

중복 이더넷 구성, 38

중복 FDDI 구성
그림, 40

중복 FDDI 연결, 40

중복 LAN
그림, 39

지연 시작 리소스 이름, 175, 176

직렬 포트
하트비트에 사용, 131

직렬 하트비트 연결 수
지정, 224

직렬(RS232) 하트비트 회선, 42
그림, 42

ㅈ

최대 노드 수, 36

추가 패키지 리소스

모니터링, 81

패키지 구성의 매개 변수, 167, 168

ㅋ

커널 일관성

클러스터 구성, 193

클러스터 구성에서, 186, 187

쿼럼 서버

계획, 139

계획 워크시트 양식, 403

상태 및 세부 상태, 282

설치, 196

워크시트, 140

클러스터 관리자 구성의 매개 변수, 148

클러스터 재편성 과정에 사용, 68

큰 규모의 클러스터, 54

클라이언트 연결

응용 프로그램에서 복원, 380

클러스터

구성 요소 이해, 36

구성 요소의 중복, 36

명령을 사용하여 구성, 215

일반적인 구성, 23

큰 규모, 54

활성/대기 종류, 54

MC/ServiceGuard, 24

클러스터 관리, 292

문제 해결, 342

클러스터 관리자

계획 워크시트 양식, 408

구성 가능한 최대 패키지 수 매개 변수, 153

구성 계획, 148

기본 기능, 62

네트워크 폴링 간격 매개 변수, 152

노드 시간 종료 매개 변수, 151

동적 재편성, 64

모니터링되는 비 하트비트 서버넷, 150

물리 잠금 볼륨 매개 변수, 149

정의, 62

직렬 장치 파일 매개 변수, 151

쿼럼 서버 매개 변수, 148

쿼럼 서버 시간 종료 확장 매개 변수, 148

쿼럼 서버 폴링 간격 매개 변수, 148

클러스터 노드 매개 변수, 148, 149

클러스터 볼륨 그룹 매개 변수, 153

클러스터 자동 재시작, 64

클러스터 초기 구성, 62

테스트, 321

하트비트 간격 매개 변수, 151

하트비트 서버넷 매개 변수, 149

클러스터 관리자 매개 변수

초기 구성, 62

클러스터 관리자의 작동 방식, 62

클러스터 구성

계획, 146

계획 워크시트, 155

다이어그램 예, 127

모든 노드의 파일, 62

클러스터 구성 확인, 225

클러스터 인식 볼륨 그룹 지정, 223

클러스터 잠금 볼륨 그룹 지정, 221

SAM 또는 명령으로 만들기, 214

클러스터 구성 삭제

cmdeleteconf 사용, 241

클러스터 구성 예

그림, 127

클러스터 구성 파일, 216

자동 시작 지연 매개 변수

(AUTO_START_TIMEOUT), 152

클러스터 구성 확인, 225

클러스터 구성원 변경, 64

클러스터 노드

클러스터 관리자 구성의 매개 변수, 148, 149

OPS 인스턴스 시작 및 종료, 257

클러스터 노드 추가

사전 계획, 194

클러스터 만들기

논리 볼륨 하부 구조, 199

쿼럼 서버 지정, 222

클러스터 구성 단계, 214

클러스터 구성 확인, 225

클러스터 잠금 볼륨 그룹 지정, 221

하트비트 서버넷 지정, 223

ASCII 클러스터 구성 파일 템플릿, 215

CVM 하부 구조, 230

VxVM 하부 구조, 209

클러스터 매개 변수

초기 구성, 62

- 클러스터 문제 해결, 319
- 클러스터 및 노드 관리, 292
- 클러스터 및 패키지 구성 배포, 272, 273
- 클러스터 및 패키지 구성 확인, 272, 273
- 클러스터 및 패키지 유지 관리, 275
- 클러스터 볼륨 그룹
 - 물리 볼륨 만들기, 201
 - 클러스터 관리자 구성의 매개 변수, 153
- 클러스터 수동 시작, 64
- 클러스터 시작
 - 모든 노드가 중지되었을 때, 292
 - 수동, 64
- 클러스터 실행 중에 노드 삭제, 306, 308
- 클러스터 실행 중에 노드 추가, 305
- 클러스터 실행 중에 볼륨 그룹 구성 변경, 307
- 클러스터 오프라인 상태에서 패키지 재구성, 310
- 클러스터 인식 볼륨 그룹 지정, 223
- 클러스터 자동 재시작, 64, 296
- 클러스터 작동 테스트, 320
- 클러스터 잠금
 - 4개 이상의 노드, 69
 - 구성 데이터 저장, 228
 - 구성 파일에서 지정, 221, 222
 - 단일 잠금 디스크, 67
 - 두 노드, 66
 - 이중 잠금 디스크, 68
 - 잠금 데이터 백업, 192
 - 잠금 없음, 69
 - 전원 공급 장치, 53
 - 클러스터 재편성 시간, 147
 - 클러스터 재편성에서 사용, 66
- 클러스터 잠금 사용, 66, 68
- 클러스터 잠금 없음
 - 선택, 69
- 클러스터 잠금 정보 백업, 192
- 클러스터 재편성 시간, 147
- 클러스터 중지, 296
- 클러스터 코디네이터
 - 정의, 62
- 클러스터 크기
 - 변경 준비, 194
- 클러스터 확장
 - 사전 계획, 126
- 클러스터 확장을 위한 계획, 126
- 클러스터가 실행 중일 때 허용되는 패키지 변경, 313

- 클러스터가 실행되는 동안 허용되는 클러스터 변경, 302

- 클러스터의 자동 시작
 - 설정, 239

ㅅ

테스트

- 네트워크 관리자, 321
- 클러스터 관리자, 321
- 패키지 관리자, 320

템플릿

- ASCII 클러스터 구성 파일, 215
- ASCII 패키지 구성 파일, 246

트래픽 종류

- LAN 하드웨어 계획, 129

ㅇ

파일 시스템

- 계획, 141
- 제어 스크립트, 171, 172
- 클러스터에 대해 만들기, 202, 212, 234

- 파일 시스템 마운트 재시도 횟수, 172

- 파일 시스템 마운트 해제 횟수, 172

- 파일 잠금, 379

패키지

- 기본 개념, 36
- 로컬 인터페이스 전환, 98
- 시작, 297
- 원격 전환, 102
- 이동, 298
- 중지, 298

- 클러스터 오프라인 상태에서 재구성, 310

- 클러스터가 실행 중일 때 재구성, 311

- 클러스터가 실행 중일 때 허용되는 패키지 변경, 313

- 패키지 실행 시기 및 위치 결정, 72

- 패키지 IP 주소 추가 및 삭제, 97

- MC/ServiceGuard 클러스터, 24

- OPS 인스턴스 시작, 257

- 패키지 관리, 297

- 문제 해결, 342

- 패키지 관리 액세스, 169

- 패키지 관리자

- 계획 워크시트 양식, 410, 411

- 테스트, 320

- 패키지 구성
 - 계획, 157
 - 구성 파일 배포, 272, 273
 - 구성 확인, 272, 273
 - 단계별, 243
 - 로컬 전환 매개 변수, 163
 - 리소스 작동 매개 변수, 168
 - 리소스 폴링 간격 매개 변수, 168
 - 서브넷 매개 변수, 167
 - 서비스 이름 매개 변수, 166
 - 서비스 장애 조치 매개 변수, 166
 - 서비스 중지 시간 종료 매개 변수, 167
 - 실행 및 중지 스크립트 시간 종료 매개 변수, 165
 - 자동 전환 매개 변수, 162
 - 장애 복구 정책 매개 변수, 162
 - 장애 조치 정책 매개 변수, 161
 - 제어 스크립트 경로 이름 매개 변수, 162
 - 추가 패키지 리소스 매개 변수, 167, 168
 - 패키지 빠른 장애 조치 매개 변수, 163
 - 패키지 유형 매개 변수, 161
 - 패키지 이름 매개 변수, 161
 - 패키지 제어 스크립트 작성, 256
 - HP-UX 명령 사용, 245
 - SAM, 244
- 패키지 구성 만들기, 244
- 패키지 구성 삭제
 - cmdelateconf 사용, 312
- 패키지 구성 파일, 246
- 패키지 및 서비스 구성, 243
- 패키지 및 클러스터 유지 관리, 275
- 패키지 빠른 장애 조치
 - 패키지 구성의 매개 변수, 163
- 패키지 시작, 297
- 패키지 실행 시기 및 위치 결정, 72
- 패키지 유형
 - 패키지 구성의 매개 변수, 161
- 패키지 이동, 298
- 패키지 이름
 - 패키지 구성의 매개 변수, 161
- 패키지 장애
 - 대응, 120
- 패키지 장애 조치 동작, 83
- 패키지 재구성
 - 클러스터가 실행 중일 때, 311
- 패키지 전환 동작
 - 변경, 300
- 패키지 제어 스크립트
 - 명령으로 생성, 256
 - 서브넷, 172, 174
 - 서비스 명령, 175
 - 서비스 이름, 174
 - 서비스 재시작 변수, 175
 - 워크시트, 176
 - IP 주소, 172, 174
- 패키지 제어 스크립트의 성능 변수, 173
- 패키지 중지, 298
- 패키지 코디네이터
 - 정의, 62
- 패키지 IP 주소
 - 정의, 96
 - 확인, 335
- 포트
 - 이중 및 단일 통합, 104
- ㅎ
 - 하드웨어
 - 계획 워크시트 양식, 400
 - 모니터링, 323
 - 하드웨어 계획
 - 공유 디스크의 디스크 입출력 정보, 133
 - 구성 계획, 127
 - 디스크 입출력 버스 종류, 133
 - 메모리 용량, 128
 - 서브넷, 129, 139
 - 워크시트, 134
 - 입출력 버스 주소, 133
 - 입출력 슬롯 번호, 133
 - 입출력 슬롯 수, 128
 - 호스트 이름, 128
 - 호스트 IP 주소, 129, 139, 140
 - LAN 인터페이스 이름, 129, 139
 - LAN 정보, 128
 - LAN 트래픽 종류, 129
 - RS232 하트비트 회전, 131
 - S800 시리즈 번호, 128
 - SPU 정보, 128
 - 하드웨어 모니터링, 323
 - 하드웨어 장애
 - 대응, 120
 - 하드웨어 장애에 대한 대응, 120
 - 하트비트

RS232 회선, 131
하트비트 간격
클러스터 관리자 구성의 매개 변수, 151
하트비트 메시지, 25
정의, 62
하트비트 서브넷 주소
클러스터 관리자 구성의 매개 변수, 149
하트비트 회선
RS232 구성, 131
하트비트 회선, 직렬, 42
호스트 이름
하드웨어 계획, 128
호스트 IP 주소
하드웨어 계획, 129, 139, 140
확장
계획, 160
활성 노드, 25

A

APA
자동 포트 통합, 103
ARP 메시지
전환 후, 103
ASCII 클러스터 구성 파일 템플릿, 215
ASCII 패키지 구성 파일 템플릿, 246
AUTO_RUN
샘플 ASCII 패키지 구성 파일, 246
패키지 구성의 매개 변수, 162
AUTO_RUN 매개 변수, 257
AUTO_START_TIMEOUT
샘플 구성 파일, 216
클러스터 관리자 구성의 매개 변수, 152

C

CLUSTER_NAME(클러스터 이름)
샘플 구성 파일, 216
cmapplyconf, 227, 273
cmcheckconf, 226, 272
문제 해결, 339
cmdeleteconf
클러스터 구성 삭제, 241
패키지 구성 삭제, 312
cmmodnet
제어 스크립트에서 IP 주소 할당, 96
cmquerycl

문제 해결, 339
CONCURRENT_DISKGROUP_OPERATIONS
패키지 제어 스크립트의 매개 변수, 173
CONCURRENT_FSCK_OPERATIONS
패키지 제어 스크립트의 매개 변수, 173
CONCURRENT_MOUNT_OPERATIONS
패키지 제어 스크립트의 매개 변수, 173
CONCURRENT_VGCHANGE_OPERATIONS
패키지 제어 스크립트의 매개 변수, 173
CVM, 112, 114
계획, 144
저장 장치 하부 구조 만들기, 230
VxVM-CVM-pkg 사용, 231
CVM_ACTIVATION_CMD, 171
패키지 제어 스크립트, 258
CVM_DG
패키지 제어 스크립트, 258

D

DNS 서비스, 188
DTC
MC/ServiceGuard에서 사용, 176

E

EMS
디스크 모니터링, 46
예방 모니터링, 323
패키지 리소스 모니터링, 81
EMS HA 모니터 사용, 82
Event Monitoring Service
디스크 모니터링, 46
문제 해결, 323

F

FAILBACK_POLICY 매개 변수
패키지 관리자가 사용, 78
패키지 구성 파일, 162
FAILOVER_POLICY 매개 변수
패키지 관리자가 사용, 75
패키지 구성 파일, 161
FIRST_CLUSTER_LOCK_PV
샘플 구성 파일, 216
클러스터 관리자 구성의 매개 변수, 149, 150
FIRST_CLUSTER_LOCK_VG

샘플 구성 파일, 216

FS

샘플 패키지 제어 스크립트, 258

FS_MOUNT_OPT

샘플 패키지 제어 스크립트, 258

G

gethostbyname

패키지 IP 주소, 96

gethostbyname(), 375

H

HA

디스크 엔클로저, 46

HA 모니터(EMS), 82

HA 응용 프로그램과 MC/ServiceGuard 통합, 387

HA 클러스터 계획 및 문서화, 123

HALT_SCRIPT

샘플 ASCII 패키지 구성 파일, 246

패키지 구성의 매개 변수, 164

HALT_SCRIPT_TIMEOUT(중지 스크립트 시간 종료)

샘플 ASCII 패키지 구성 파일, 246

패키지 구성의 매개 변수, 165

HEARTBEAT_INTERVAL

샘플 구성 파일, 216

HEARTBEAT_INTERVAL(하트비트 시간 종료)

클러스터 관리자 구성의 매개 변수, 151

HEARTBEAT_IP

샘플 구성 파일, 216

클러스터 관리자 구성의 매개 변수, 149

HP, 112

HP Predictive 모니터링

문제 해결, 324

HP-UX의 OPS용 하드웨어

전원 공급 장치, 53

I

IP

샘플 패키지 제어 스크립트, 258

패키지 제어 스크립트의 IP 주소 배열 변수, 174

IP 주소

노드 및 패키지, 96

변동 가능, 96

전환, 73, 74, 103

패키지에 대해 확인, 335

패키지에서 추가 및 삭제, 97

하드웨어 계획, 129, 139, 140

IP 주소 전환, 73, 74, 103

IP 주소로 로드 공유, 97

J

JFS, 369

L

LAN

계획 정보, 128

인터페이스 이름, 129, 139

하트비트, 62

LAN 계획

트래픽 종류, 129

호스트 IP 주소, 129, 139, 140

LAN 인터페이스

기본 및 보조, 38

네트워크 관리자를 사용한 모니터링, 97

LAN 인터페이스 모니터링

네트워크 관리자, 97

LAN 장애

MC/ServiceGuard 동작, 36

LOCAL_LAN_FAILOVER_ALLOWED

샘플 ASCII 패키지 구성 파일, 246

패키지 구성의 매개 변수, 163

lssf

디스크 목록을 구하는 데 사용, 200

LV

샘플 패키지 제어 스크립트, 258

lvextend

루트 미러 만들기, 191

LVM, 112, 113

계획, 141

논리 볼륨 만들기, 157

다른 노드에 볼륨 그룹 설정, 205

디스크, 44

루트 미러 만들기, 190

볼륨 그룹 만들기, 157

클러스터 사용 명령, 199

파일 시스템 만들기, 157

VxVM으로 마이그레이션, 413

LVM 구성

워크시트, 142, 144

LVM_ACTIVATION_CMD, 170

M

MAC 주소, 375

MAX_CONFIGURED_PACKAGES

클러스터 관리자 구성의 매개 변수, 153

MC/LockManager

설치, 198

MC/ServiceGuard

소개, 23, 24

MC/ServiceGuard 구성 작업

그림, 33

MC/ServiceGuard 동작

모니터링된 리소스 장애 시, 36

소프트웨어 장애 시, 36

LAN 장애 시, 36

MC/ServiceGuard 명령

패키지 구성에 사용, 245

MC/ServiceGuard 소프트웨어 구성 요소

그림, 58

MC/ServiceGuard에서 지원되는 네트워크, 38

MC/ServiceGuard에서 지원되는 디스크, 44

MC/ServiceGuard의 개념, 24

MC/ServiceGuard의 네트워크 구성 요소 이해, 38

MirrorDisk/UX, 45

mkboot

루트 미리 만들기, 190

N

NETWORK_INTERFACE

샘플 구성 파일, 216

NETWORK_POLLING_INTERVAL(네트워크 폴링 간격)

샘플 구성 파일, 216

NODE_FAIL_FAST_ENABLED

샘플 ASCII 패키지 구성 파일, 246

패키지 구성의 매개 변수, 163

NODE_FAILFAST_ENABLED 매개 변수, 257

NODE_NAME

샘플 ASCII 패키지 구성 파일, 246

클러스터 관리자 구성의 매개 변수, 148, 149

NODE_TIMEOUT(노드 시간 종료)

클러스터 관리자 구성의 매개 변수, 151

NODE_TIMEOUT(하트비트 시간 종료)

샘플 구성 파일, 216

NTP

클러스터의 시간 프로토콜, 193

NTP(네트워크 시간 프로토콜)

클러스터, 193

O

OPS

인스턴스 시작 및 종료, 257

P

package

데이터베이스 용 도구 키트, 363

PACKAGE_NAME

샘플 ASCII 패키지 구성 파일, 246

패키지 ASCII 구성 파일의 매개 변수, 161

PACKAGE_TYPE

패키지 ASCII 구성 파일의 매개 변수, 161

PATH, 170

Predictive 모니터링, 324

PV 링크

볼륨 그룹 만들기, 203

pvcreate

루트 미리 만들기, 190

PVG 완전 미러링

볼륨 그룹 만들기, 201

Q

QS_HOST

클러스터 관리자 구성의 매개 변수, 148

QS_POLLING_INTERVAL

클러스터 관리자 구성의 매개 변수, 148

QS_TIMEOUT_EXTENSION

클러스터 관리자 구성의 매개 변수, 148

R

RAID

데이터 보호, 45

README

데이터베이스 도구 키트, 363

RESOURCE_NAME

샘플 ASCII 패키지 구성 파일, 246

패키지 구성의 매개 변수, 167, 168

RESOURCE_POLLING_INTERVAL

샘플 ASCII 패키지 구성 파일, 246

패키지 구성의 매개 변수, 168
RESOURCE_UP_VALUE
샘플 ASCII 패키지 구성 파일, 246
패키지 구성의 매개 변수, 168
rhosts 파일
보안용, 182
RS232 상태 보기, 288
RS232 상태, 보기, 288
RS232 연결
하트비트, 131
RS232 직렬 하트비트 회선, 42
RS232 하트비트 회선, 구성, 131
RUN_SCRIPT
샘플 ASCII 패키지 구성 파일, 246
패키지 구성의 매개 변수, 164
RUN_SCRIPT_TIMEOUT
샘플 ASCII 패키지 구성 파일, 246
RUN_SCRIPT_TIMEOUT(실행 스크립트 시간 종료)
패키지 구성의 매개 변수, 165

S

SAM

패키지 구성에 사용, 244

SCSI 주소 지정, 132, 147

SECOND_CLUSTER_LOCK_PV

클러스터 관리자 구성의 매개 변수, 149, 150

SERIAL_DEVICE_FILE(RS232)

클러스터 관리자 구성의 매개 변수, 151

SERVICE_CMD

샘플 패키지 제어 스크립트, 258

패키지 제어 스크립트의 배열 변수, 175, 176

SERVICE_FAIL_FAST_ENABLED

샘플 ASCII 패키지 구성 파일, 246

패키지 구성의 매개 변수, 166

SERVICE_HALT_TIMEOUT

샘플 ASCII 패키지 구성 파일, 246

패키지 구성의 매개 변수, 167

SERVICE_NAME

샘플 패키지 제어 스크립트, 258

샘플 ASCII 패키지 구성 파일, 246

패키지 구성의 매개 변수, 166

패키지 제어 스크립트의 배열 변수, 174

SERVICE_RESTART

샘플 패키지 제어 스크립트, 258

패키지 제어 스크립트의 배열 변수, 175

ServiceGuard 명령을 사용하여 클러스터 모니터링, 236

ServiceGuard 명령줄을 사용하여 클러스터 구성, 215

ServiceGuard Manager, 31

개요, 27

ServiceGuard Manager를 사용하여 클러스터 구성, 214

ServiceGuard Manager를 사용하여 클러스터 모니터링, 236

SGCONF, 180

SMN 패키지, 71

SNA 응용 프로그램, 379

SPU 정보

계획, 128

STATIONARY_IP

클러스터 관리자 구성의 매개 변수, 150

SUBNET

샘플 패키지 제어 스크립트, 258

샘플 ASCII 패키지 구성 파일, 246

패키지 구성의 매개 변수, 167

패키지 제어 스크립트의 배열 변수, 172, 174

T

TOC

노드에 장애가 발생할 때, 119

U

uname(2), 376

UPS

전원 계획, 136

HP-UX의 OPS용 전원 공급 장치, 53

USER_HOST, 153

USER_NAME, 153

USER_ROLE, 153

V

VG

샘플 패키지 제어 스크립트, 258

vgcfgbackup

클러스터 잠금 데이터, 228

VGCHANGE

패키지 제어 스크립트, 258

vgextend

루트 미러 만들기, 190

vgimport

- 다른 노드에 볼륨 그룹을 설정하는 데 사용, 206

VOLUME_GROUP

- 샘플 구성 파일, 216
- 클러스터 관리자 구성의 매개 변수, 153

VxVM, 112, 113

- 계획, 144
- 저장 장치 하부 구조 만들기, 209
- LVM에서 VxVM으로 마이그레이션, 413

VXVM_DG

- 패키지 제어 스크립트, 258

VxVM-CVM 패키지, 71

VxVM-CVM-pkg, 231