

MC/ServiceGuard 관리



제품 번호: B3936-90068

2002년 3월

알림

이 설명서의 내용은 예고 없이 변경될 수 있습니다.

HP는 이 자료에 대해 상업성이나 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 어떤 종류의 보증도 하지 않습니다. HP는 이 자료의 오류나 공급, 수행, 또는 사용과 연관된 부수적 또는 파생적인 손해에 대해 책임을 지지 않습니다.

Copyright © 1995, 1997, 1998, 2000, 2001, 2002 Hewlett-Packard Company.

이 설명서에는 저작권에 따라 보호되는 정보가 포함되어 있습니다. 모든 권리는 HP가 보유합니다. 저작권법에서 허용한 경우를 제외하고 HP의 사전 서면 승인 없이는 이 설명서를 재발행하거나 개작하거나 번역할 수 없습니다.

MC/ServiceGuard, ServiceGuard OPS Edition 및 ServiceGuard Manager는 Hewlett-Packard Company의 제품이며 모두 저작권법의 보호를 받습니다.

HP 본사 주소:

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

이 패키지에서 제공하는 설명서와 플로피 디스크, CD, 테이프 카트리지는 이 제품에서만 사용하도록 제한됩니다. 프로그램의 추가 사본은 보안 및 백업 목적으로만 만들 수 있습니다. 제공된 그대로 또는 부가된 형태로 이 프로그램을 재판매하는 것은 명백히 금지됩니다.

HP 제품에 적용되는 특정 보증서 사본과 교체 부품들은 현지 대리점이나 서비스 센터에서 구할 수 있습니다.

1. MC/ServiceGuard 소개

MC/ServiceGuard란?	22
장애 조치	23
ServiceGuard 클러스터 보기	25
구성 지침	26

2. MC/ServiceGuard 하드웨어 구성 이해

클러스터 구성 요소 중복	28
네트워크 구성 요소의 중복	30
중복 이더넷 구성	30
중복 FDDI 연결 구성	32
이중 부착 FDDI 스테이션 사용	32
직렬(RS232) 하트비트 회선 사용	33
장애가 발생한 네트워크 카드의 교체	34
중복 디스크 저장 장치	35
지원되는 디스크 인터페이스	35
데이터 보호	36
이벤트 모니터링 서비스를 통한 디스크 모니터링	37
장애가 발생한 디스크 교체	37
장애가 발생한 I/O 카드 교체	37
SCSI 디스크 구성 예제	37
Fibre 채널 디스크 구성 예제	39
공유 SCSI 버스의 루트 디스크 제한 사항	40
중복 전원 공급 장치	42
보다 큰 클러스터	43
활성/대기 모델	43
저장 장치에 지점간 연결	44

3. MC/ServiceGuard 소프트웨어 구성 요소 이해

MC/ServiceGuard 아키텍처	48
ServiceGuard 데몬	48
클러스터 관리자 작동 방식	52
클러스터 구성	52
하트비트 메시지	52
전체 클러스터의 수동 시작	53
자동 클러스터 시작	54
동적 클러스터 재편성	54
분할 브레인 상황을 피하기 위한 클러스터 정족수	55
클러스터 잠금	55
클러스터 잠금으로 사용하는 LVM 잠금 디스크	56
클러스터 잠금으로 사용하는 Quorum 서버	58
클러스터 잠금 사용 안 함	59
패키지 관리자 작동 방식	60
패키지 종류	60
장애 조치 패키지	60
패키지 구성	61
패키지를 실행하고 중지하는 시간 및 장소 결정	61
패키지 전환	61
장애 조치 정책	63
장애 복구 정책	66
오래된 패키지 구성 파일 사용	68
이벤트 모니터링 서비스 사용	69
EMS 고가용성 모니터 사용	69
패키지 장애 조치 동작 선택	70
패키지 제어 스크립트의 작동 방식	72
패키지 실행 요인	72
제어 스크립트 시작 전	73
실행 스크립트 실행 중	74

실행 스크립트의 정상 종료 및 비정상 종료	75
cmrunserv를 사용한 서비스 시작	76
서비스 실행 중	76
서비스, 서브넷 또는 모니터된 리소스에서 장애 발생 시	77
명령으로 패키지나 노드 중지 시	78
중지 스크립트 실행 중	78
중지 스크립트의 정상 종료 및 비정상 종료	79
네트워크 관리자의 작동 방식	82
고정 IP 주소와 변동 가능 IP 주소	82
변동 가능 IP 주소 추가 및 삭제	83
LAN 인터페이스 모니터링 및 장애 감지	83
자동 포트 집계	87
데이터 저장을 위한 볼륨 관리자	89
중복 저장 종류	89
미러링된 저장 예제	90
디스크 어레이의 저장 장치 예제	91
볼륨 관리자 유형	93
HP-UX Logical Volume Manager(LVM)	93
VERITAS 볼륨 관리자(VxVM)	94
VERITAS 클러스터 볼륨 관리자(CVM)	95
볼륨 관리자 비교	97
장애 대응	99
노드 장애 시 제어 전송(TOC)	99
하드웨어 장애 대응	100
패키지 및 서비스 장애 대응	100
서비스 재시작	101
네트워크 통신 장애	101
4. HA(고가용성) 클러스터 계획 및 문서화	
전반적인 계획	105

차례

MC/ServiceGuard 메모리 요구 사항	106
확장 계획	106
하드웨어 계획	108
SPU 정보	109
네트워크 정보	109
클러스터의 최대 예상 크기에 맞춰 SCSI 주소 설정	111
디스크 I/O 정보	112
하드웨어 구성 워크시트	115
전원 공급 계획	116
전원 공급 구성 워크시트	117
Quorum 서버 계획	118
LVM 계획	120
5. HA(고가용성) 클러스터 구성 만들기	
시스템 준비	155
파일이 있는 위치에 대한 정보	155
보안 파일 편집	156
루트 논리 볼륨의 미러 만들기	157
클러스터 잠금 디스크 선택	159
비루트 사용자의 cmviewcl 실행 허용	159
이름 확인 서비스의 정의	160
커널 구성의 일관성 확인	162
네트워크 시간 프로토콜 사용	162
클러스터 크기 변경 준비	162
Quorum 서버 설치	164
Quorum 서버 설치	164
Quorum 서버 실행	165
MC/ServiceGuard 설치	166
LVM을 사용하여 저장 하부 구조 만들기	167

미러링된 개별 데이터 디스크의 볼륨 그룹 만들기 167

PV 링크를 사용하여 디스크 어레이의 볼륨 그룹 만들기 170

다른 노드에 대한 볼륨 그룹 배포 172

추가 볼륨 그룹 만들기 175

VxVM을 사용하여 저장 하부 구조 만들기 176

VERITAS 볼륨 관리자 초기화 176

클러스터 구성 181

 CVM을 사용하여 저장 하부 구조 만들기 192

실행 클러스터 관리 197

6. 패키지 및 서비스 구성

패키지 구성 만들기 204

 SAM을 사용한 패키지 구성 204

 MC/ServiceGuard 명령을 사용한 패키지 작성 205

패키지 제어 스크립트 작성 213

 SAM을 사용한 패키지 제어 스크립트 작성 213

 명령을 사용한 패키지 제어 스크립트 작성 213

 데이터베이스 제품의 패키지 만들기 214

 패키지 제어 스크립트 사용자 정의 214

 다수의 저장 장치 단위에 대한 패키지 최적화 217

 패키지 제어 스크립트 템플릿 파일 217

패키지 제어 스크립트에 사용자 정의 함수 추가 222

 사용자 정의 함수에 ServiceGuard 명령 추가 223

 추가 제품 지원 224

패키지 구성 확인 225

구성 배포 226

 SAM을 사용한 구성 파일 및 제어 스크립트 배포 226

 HP-UX 명령을 사용한 패키지 제어 스크립트 복사 226

 HP-UX 명령을 사용한 이진 클러스터 구성 파일 배포 226

 클러스터 및 패키지 작동 테스트 227

7. 클러스터 및 패키지 유지 관리

cmviewcl 명령을 사용하여 클러스터 및 패키지 상태 확인	230
클러스터 및 패키지 상태 종류	230
클러스터 및 패키지 상태의 예제	233
ServiceGuard Manager 사용	243
명령을 사용하여 ServiceGuard Manager 실행	243
ServiceGuard Manager의 작동 방식	244
특정 클러스터를 사용하여 ServiceGuard Manager 시작	244
특정 클러스터를 사용하지 않고 ServiceGuard Manager 시작	245
개체 관리자 연결	247
클러스터 데이터에 저장된 파일 열기	248
클러스터 데이터 보기	248
도움말 보기	251
클러스터 개체 관리	252
클러스터 및 노드 관리	256
모든 노드가 중지되었을 때 클러스터 시작	256
실행 중인 클러스터에 이전에 구성된 노드 추가	257
실행 중인 클러스터에서 노드 제거	258
전체 클러스터 중지	260
클러스터 자동 재시작	260
패키지 및 서비스 관리	261
패키지 시작	261
패키지 중지	262
패키지 이동	263
패키지 전환 동작 변경	264
클러스터 재구성	266
중지된 클러스터 재구성	267
실행 중인 클러스터 재구성	268
패키지 재구성	273

중지된 클러스터에서 패키지 재구성 273
 실행 중인 클러스터에서 패키지 재구성 273
 실행 중인 클러스터에 패키지 추가 274
 실행 중인 클러스터에서 패키지 삭제 275
 서비스 재시작 카운터 재설정 275
 재구성 중에 허용되는 패키지 상태 276
 클러스터 이벤트에 대응 278
 시스템에서 MC/ServiceGuard 제거 279

8. 클러스터 문제점 해결

클러스터 작동 테스트 282
 SAM을 사용하여 클러스터 시작 282
 패키지 관리자 테스트 282
 클러스터 관리자 테스트 283
 네트워크 관리자 테스트 284
 하드웨어 모니터링 285
 이벤트 모니터링 서비스 사용 285
 EMS 하드웨어 모니터 사용 285
 HP Predictive를 사용한 모니터링 286
 디스크 교체 287
 결함 있는 어레이 장치 교체 287
 HA 엔클로저에서 결함 있는 장치 교체 287
 잠금 디스크 교체 288
 인라인 SCSI 터미네이터를 사용하여 온라인 하드웨어 유지 관리 288
 I/O 카드 교체 291
 LAN 카드 교체 292
 오프라인으로 교체 292
 온라인으로 교체 292
 카드 교체 후 293

차례

문제점 해결 방법	294
패키지 IP 주소 검사	294
시스템 로그 파일 검사	295
개체 관리자 로그 파일 검사	296
ServiceGuard Manager 로그 파일 검사	296
구성 파일 검사	296
패키지 제어 스크립트 검사	297
cmcheckconf 명령 사용	297
cmscancl 명령 사용	298
cmviewconf 명령 사용	298
LAN 구성 검사	298
문제점 해결	300
ServiceGuard 명령 중지	300
클러스터 재편성	301
시스템 관리 오류	301
VxVM 디스크 그룹의 문제	304
패키지 이동 오류	305
노드 및 네트워크 장애	306
Quorum 서버 문제점 해결	307
A. Mc/ServiceGuard 명령	
B. Enterprise Cluster Master Toolkit	
C. 고가용성 클러스터 응용 프로그램 설계	
응용 프로그램 작동 자동화	322
사용자 개입의 최소화	322
응용 프로그램 시작과 종료 정의	323
응용 프로그램 장애 조치 속도 제어	324
데이터가 아닌 파일 시스템 복제	324
원시 볼륨 사용	324

JFS 사용 검토	325
데이터 손실 최소화	325
재시작 가능한 트랜잭션 사용	326
검사점 사용	327
다중 서버를 위한 설계	327
복제된 데이터 지점을 위한 설계	328
다중 시스템에서 실행 가능한 응용 프로그램 설계	329
노드에 특정한 정보 최소화	329
SPU ID 또는 MAC 주소 사용 최소화	330
응용 프로그램에 고유한 이름 지정	331
uname(2) 사용 시 주의점	332
고정 포트로 바인딩	332
변동 가능 IP 주소로 바인딩	333
응용 프로그램마다 고유한 볼륨 그룹 부여	334
SNA 응용 프로그램을 위한 복수의 목적지 사용	335
파일 잠금 방지	335
클라이언트 연결 복원	336
응용 프로그램 장애 처리	338
장애를 극복할 수 있는 응용 프로그램 작성	338
응용 프로그램 모니터링 기능	339
계획된 다운시간 최소화	340
응용 프로그램 업그레이드/패치에 요구되는 시간 단축	340
온라인 응용 프로그램 재편성 기능	341
유지 보수 작업 문서화	342
D. HA(고가용성) 응용 프로그램과 MC/ServiceGuard 통합	
HA(고가용성) 응용 프로그램을 통합하기 위한 검사 목록	344
단일 시스템에서 기본 응용 프로그램의 동작 정의	344
다중 시스템에 HA(고가용성) 응용 프로그램 통합	344
클러스터 테스트	345

E. 롤링 소프트웨어 업그레이드

롤링 업그레이드의 단계	348
SAM을 사용한 롤링 업그레이드 수행	348
커널 일관성 유지	349
롤링 업그레이드의 예제	350
1 단계	350
2 단계	351
3 단계	351
4 단계	352
5 단계	352
롤링 업그레이드의 제한 사항	354

F. LAN 에서 OTS/9000 사용

G. 계획 워크시트 양식

하드웨어 워크시트	358
전원 공급 장치 워크시트	359
Quorum 서버 워크시트	360
LVM 볼륨 그룹 및 실제 볼륨 워크시트	361
VxVM 디스크 그룹 및 디스크 워크시트	362
클러스터 구성 워크시트	363
패키지 구성 워크시트	364
패키지 제어 스크립트 워크시트	365

H. LVM 에서 VxVM 데이터 저장으로 변환

VxVM 로드	368
볼륨 그룹 변환	369

VxVM의 패키지 사용자 정의371

CVM의 패키지 사용자 정의373

LVM 볼륨 그룹 제거376

인쇄 정보

표 1

인쇄 날짜	제품 번호	개정판
1995년 1월	B3936-90001	초판(영문본)
1995년 6월	B3936-90003	제2판(영문본)
1995년 12월	B3936-90005	제3판(영문본)
1997년 5월	B3936-90019	제4판(영문본)
1998년 1월	B3936-90024	제5판(영문본)
1998년 10월	B3936-90026	제6판(한글본)
2000년 12월	B3936-90045	제7판(한글본)
2001년 9월	B3936-90053	제8판(한글본)
2002년 3월	B3936-90068	제9판(한글본)

마지막 인쇄 날짜와 제품 번호는 현재 개정판을 나타내며 MC/ServiceGuard의 최신 버전은 A.11.14입니다.

인쇄 날짜는 새 개정판이 인쇄될 때 변경됩니다.(재판 과정에 포함된 사소한 수정 및 업데이트 사항은 날짜 변경에 포함되지 않습니다.) 제품 번호는 광범위한 기술 변경 내용이 통합될 때 개정됩니다.

이 설명서의 개정판은 이전 판 이후에 업데이트된 모든 정보를 통합합니다.

HP 인쇄 본부:

Business Critical Computing
Hewlett-Packard Co.
19111 Pruneridge Ave.
Cupertino, CA 95014

머리말

이 안내서는 HP-UX 운영 체제의 HP 9000 서버에서 실행될 수 있도록 MC/ServiceGuard를 구성하는 방법에 대해 설명합니다. 그 내용은 다음과 같습니다.

- 제1장 “MC/ServiceGuard 소개”는 클러스터를 설명하고 이 안내서를 사용하기 위한 지침을 제공합니다.
- 제2장 “MC/ServiceGuard 하드웨어 구성 이해”는 MC/ServiceGuard에서 사용하는 하드웨어 구성의 일반적인 개념을 제시합니다.
- 제3장 “MC/ServiceGuard 소프트웨어 구성 요소 이해”는 MC/ServiceGuard의 소프트웨어 구성 요소를 설명하고 HP-UX 운영 체제 내에서 구성 요소가 어떻게 기능하는지 보여줍니다.
- 제4장 “HA(고가용성) 클러스터 계획 및 문서화”는 계획 프로세스의 단계를 차례대로 설명하고 클러스터에 대한 정보를 구성하기 위한 일련의 워크시트를 제공합니다.
- 제5장 “HA(고가용성) 클러스터 구성 만들기”는 클러스터 구성을 만드는 작업에 대해 설명합니다.
- 제6장 “패키지 및 서비스 구성”은 고가용성 패키지를 만드는 작업과 이 패키지와 관련된 제어 스크립트를 설명합니다.
- 제7장 “클러스터 및 패키지 유지 관리”는 기본 클러스터 관리 작업을 제공합니다.
- 제8장 “클러스터 문제점 해결”은 클러스터 테스트 및 문제점 해결 방법을 설명합니다.
- 부록 A “Mc/ServiceGuard 명령”은 MC/ServiceGuard에서 사용하는 명령을 나열하고 각 man 페이지의 요약 정보를 다시 제공합니다.
- 부록 B “Enterprise Cluster Master Toolkit”은 특수한 클러스터 구성을 만드는 데 사용할 수 있는 도구 그룹을 설명합니다.
- 부록 C “고가용성 클러스터 응용 프로그램 설계”는 HA 작업을 위한 응용 프로그램을 만들거나 포팅하는 방법에 대해 설명합니다.
- 부록 D “HA(고가용성) 응용 프로그램과 MC/ServiceGuard 통합”은 MC/ServiceGuard와 기존 응용 프로그램의 통합을 제안합니다.
- 부록 E “롤링 소프트웨어 업그레이드”는 응용 프로그램을 종료하지 않고 한 MC/ServiceGuard 또는 HP-UX 릴리즈로부터 다른 프로그램으로 이동하는 방법에 대해 설명합니다.
- 부록 F “LAN에서 OTS/9000 사용”은 전송 서비스 및 기타 프로토콜의 구성에 대해 설명합니다.

- 부록 G “계획 워크시트 양식”은 구성을 준비하기 위한 계획 워크시트들을 제공합니다.
- 부록 H “LVM에서 VxVM 데이터 저장으로 변환”은 LVM 데이터 저장에서 VxVM 데이터 저장으로 변환하는 방법에 대해 설명합니다.

관련 서적

다음 문서에서 유용한 정보를 추가로 얻으실 수 있습니다.

- *Clusters for High Availability: a Primer of HP-UX Solutions*. Hewlett-Packard Professional Books: Prentice Hall PTR, 2001 (ISBN 0-13-089355-2)
- *ServiceGuard OPS Edition*으로 OPS 클러스터 구성(B5158-90039)
- *Using High Availability Monitors*(B5736-90025)
- *Using the Event Monitoring Service*(B7612-90015)
- *Using Advanced Tape Services*(B3936-90032)
- *Designing Disaster Tolerant High Availability Clusters*(B7660-90008)
- *Managing ServiceGuard Extension for SAP/R3*(B7885-90004)
- *시스템 및 작업 그룹 관리*(B2355-90157)
- *Managing Highly Available NFS*(B5125-90001)
- *HP Auto Port Aggregation Release Notes*

ServiceGuard로 VxVM 저장 장치를 사용하기 전에 다음을 참조하십시오.

- *VERITAS Volume Manager 3.1 Administrator's Guide*. 여기에 VERITAS 용어집이 수록되어 있습니다.
- *VERITAS Volume Manager 3.1 Storage Administrator Administrator's Guide*
- *VERITAS Volume Manager 3.1 Reference Guide*
- *VERITAS Volume Manager 3.1 Migration Guide*
- *VERITAS Volume Manager 3.1 for HP-UX Release Notes*

다음 URL을 통해 HP 고가용성 웹 페이지에 액세스하실 수 있습니다.

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

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

- <http://docs.hp.com/hpux>

문제점 보고

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

1 MC/ServiceGuard 소개

이 장은 HP-UX용 MC/ServiceGuard를 소개하고 다른 정보를 이 설명서의 어느 위치에서 찾을 수 있는지 설명합니다. 이 장에서는 다음과 같은 내용을 다룹니다.

- MC/ServiceGuard란?
- ServiceGuard 클러스터 보기
- 구성 지침

MC/ServiceGuard 클러스터 구성을 시작할 준비가 되었으면 이 장을 건너뛰고 4장 “HA(고가용성) 클러스터 계획 및 문서화” 장으로 가십시오. 자세한 설치 단계는 5장 “HA(고가용성) 클러스터 구성 만들기” 장에 제공됩니다.

MC/ServiceGuard란?

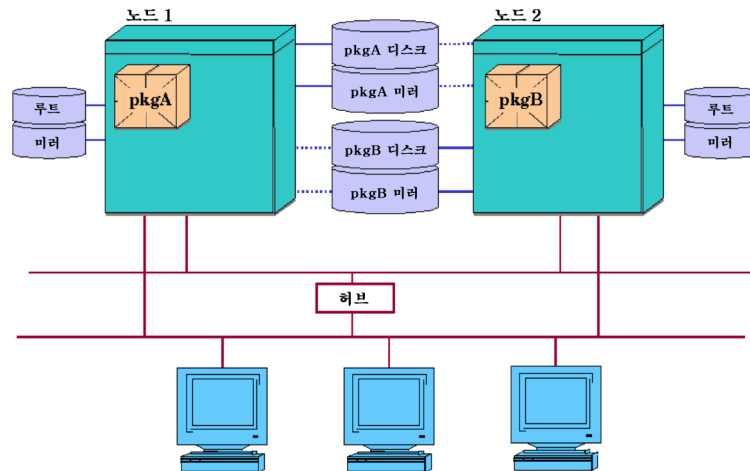
MC/ServiceGuard를 사용하여 HP 9000 시리즈 800 컴퓨터에서 고가용성 클러스터를 만들 수 있습니다. **고가용성** 컴퓨터 시스템은 하드웨어나 소프트웨어 장애가 발생해도 응용 프로그램 서비스를 계속 실행할 수 있습니다. 고가용성 시스템은 소프트웨어 장애뿐만 아니라 SPU(System Processing Unit), 디스크 또는 근거리 네트워크(LAN) 구성 요소의 장애로부터 사용자를 보호합니다. 구성 요소 중 하나에서 장애가 발생하면 다른 구성 요소가 대신 작업을 수행합니다. MC/ServiceGuard와 다른 고가용성 하위 시스템은 서로 협조하여 구성 요소 사이의 전송을 처리합니다.

MC/ServiceGuard 클러스터는 네트워크로 연결된 HP 9000 시리즈 800 서버 그룹으로(호스트 시스템을 **노드**라고 함) 소프트웨어와 하드웨어에 충분히 여유가 있어서 **한 곳에서 장애가 발생**해도 서비스에 심각한 장애가 일어나지 않습니다. 응용 프로그램 서비스(각 HP-UX 프로세스)는 **패키지**로 그룹화되어 한 서비스, 노드, 네트워크 또는 다른 리소스에 장애가 발생해도 MC/ServiceGuard가 패키지의 제어를 클러스터에 속한 다른 노드로 자동으로 전송할 수 있으며 최소한의 중단으로 서비스를 계속 사용할 수 있는 상태로 유지할 수 있습니다.

그림 1-1은 2개의 노드로 구성된 일반적인 MC/ServiceGuard 클러스터를 나타냅니다.

그림 1-1

일반적인 클러스터 구성



그림에서 노드 1(두 개의 SPU 중 하나)은 패키지 A를 실행하고 노드 2는 패키지 B를 실행합니다. 각 패키지는 패키지 응용 프로그램에서 필요로 하는 데이터와 데이터의 미러 복사본이 들어 있는 별도의 디스크 그룹을 포함합니다. 두 노드는 두 그룹의 미러링 디스크에 실제적으로 연결되어 있습니다. 그러나, 한 번에 하나의 노드만 주어진 디스크 그룹의 데이터에 액세스할 수 있습니다. 그림에서 노드 1은 위에 있는 두 디스크에 독점적으로 액세스하고(실선) 노드 2는 위에 있는 디스크에 액세스하지 않고 연결되어 있습니다(점선). 마찬가지로, 노드 2는 아래에 있는 두 디스크에 독점적으로 액세스하고(실선) 노드 1은 아래에 있는 디스크에 액세스하지 않고 연결되어 있습니다(점선).

데이터의 미러 사본은 디스크 장애를 대비한 중복 기능을 제공합니다. 또한, 노드 1과 노드 2에 연결된 디스크용으로 총 4개의 데이터 버스가 있습니다. 이 구성은 각 패키지가 서로 다른 버스를 사용하기 때문에 최대 중복 기능과 최적의 I/O 성능을 제공합니다.

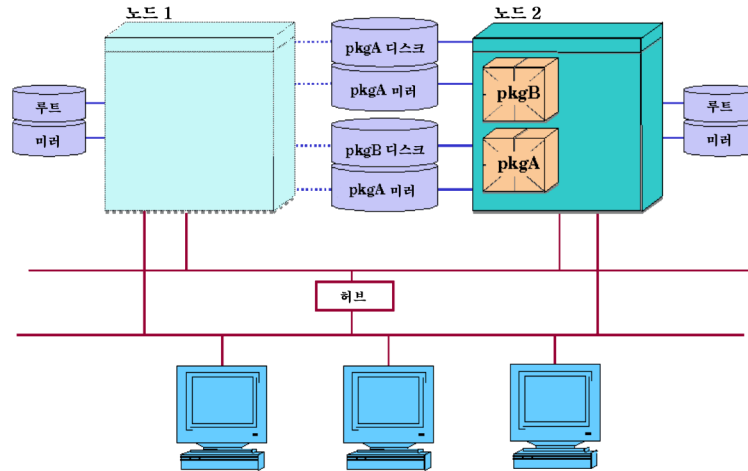
각 노드에서 네트워크 하드웨어가 연결되어 중복된 LAN 인터페이스를 제공합니다. MC/ServiceGuard는 클러스터 노드들 사이의 안정적인 통신을 위해 TCP/IP 네트워크 서비스를 사용하며, 여기에는 각 작동 노드가 보내는 신호로서 클러스터 운영의 핵심 역할을 하는 **하트비트 메시지**의 전송도 포함됩니다. TCP/IP 서비스는 다른 종류의 노드 간 통신에도 사용됩니다. 하트비트는 “MC/ServiceGuard 소프트웨어의 이해” 장에 자세히 설명되어 있습니다.

장애 조치

일반적인 조건에서 완전히 운영되고 있는 MC/ServiceGuard 클러스터는 패키지가 각 노드에서 실행되는 동안 클러스터 구성 요소의 이상 여부만 감시합니다. MC/ServiceGuard 클러스터에서 실행되는 호스트 시스템을 **활성 노드(Active Node)**라고 합니다. 패키지를 만들 때 하나의 **주 노드(Primary Node)**와 하나 이상의 **대체 노드(Adoptive Node)**를 지정합니다. 노드나 노드 간 네트워크 통신에 장애가 발생하면 MC/ServiceGuard는 패키지 제어 다음으로 사용 가능한 대체 노드로 전달할 수 있습니다. 그림 1-2는 이러한 상황을 나타냅니다.

그림 1-2

장애 조치 후의 일반적인 클러스터



제어가 전달된 다음에 패키지는 대체 노드가 실행되는 동안 계속해서 대체 노드에 남아 있습니다. 그러나 원하는 경우 주 노드가 다시 온라인 상태가 될 때 패키지가 원래 주 노드로 되돌아가도록 구성할 수 있습니다. 또는 원하는 시간에 수동으로 패키지 제어를 주 노드로 다시 가져올 수 있습니다.

그림 1-2는 클러스터의 전원 연결이 표시되지 않았지만 전원 연결도 중요한 요소 중 하나입니다. 클러스터에서 단순 장애 요인을 모두 제거하려면 클러스터 상의 노드, 디스크 및 디스크 미러의 단순 장애 요인을 방지할 수 있도록 충분한 수의 개별 전원 회로를 갖추어야 합니다. 그리고 모든 전원 회로를 무정전 전원으로 보호해야 합니다. 자세한 내용은 4장 “HA 클러스터 계획 및 문서화”의 “전원 공급 계획” 절을 참조하십시오.

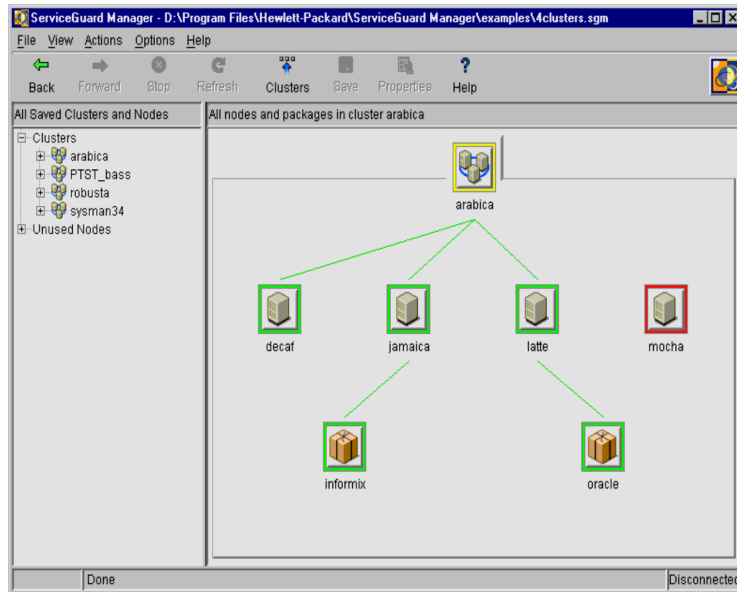
MC/ServiceGuard는 다른고가용성 제품과 함께 사용할 수 있도록 설계되었으며, 이 제품에는 디스크 하위 시스템의 단순 장애 요인을 제거할 수 있도록 디스크 중복 기능을 제공하는 MirrorDisk/UX 또는 VERITAS 볼륨 관리자, MC/ServiceGuard가 직접 처리하지 않는 장애를 감시하여 찾아낼 수 있는 이벤트 모니터링 서비스(EMS), 데이터 보호를 위해 다양한 RAID 레벨을 사용하는 디스크 어레이, HP PowerTrust처럼 정전과 관련된 장애를 제거하는 HP 지원의 무정전 전원 공급 장치(UPS) 등이 있습니다.고가용성을 확보할 수 있도록 MC/ServiceGuard와 함께 이러한 제품을 사용하는 것이 좋습니다.

ServiceGuard 클러스터 보기

ServiceGuard Manager는 MC/ServiceGuard 클러스터를 보고, 클러스터, 노드 및 패키지과 같은 개체의 상태를 관찰하는 그래픽 사용자 인터페이스입니다. ServiceGuard Manager로 클러스터 상태에 대한 스냅샷을 만들어 저장함으로써 클러스터 구성을 문서화하고 그 내용을 나중에 비교 자료로 사용할 수 있습니다. 그림 1-3은 예제 화면으로 클러스터 맵과 한 클러스터에 있는 개체들의 계층 구조를 볼 수 있습니다.

그림 1-3

ServiceGuard Manager 클러스터 상태



ServiceGuard Manager를 실행하려면 HP-UX에서 `sgmgr` 명령을 사용하거나 Windows에서 ServiceGuard Manager 아이콘을 누릅니다. ServiceGuard Manager에 대한 자세한 내용은 7장의 “ServiceGuard Manager 사용”을 참조하십시오.

구성 지침

이 설명서는 MC/ServiceGuard를 사용하여 실제 HA(고가용성) 클러스터를 만드는 데 필요한 작업에 대해 설명합니다. 이러한 작업은 그림 1-4에서 볼 수 있습니다.

그림 1-4

MC/ServiceGuard 클러스터 구성 작업

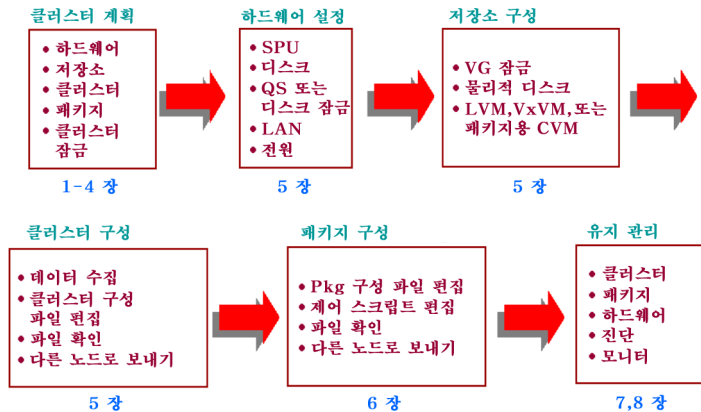


그림 1-4의 작업은 4장부터 7장까지 차례로 자세히 다룹니다. **작업을 시작하기 전에** 구성에 필요한 모든 데이터를 수집하는 것이 좋습니다. 데이터 준비에 대한 자세한 내용은 4장, “HA(고가용성) 클러스터 계획 및 문서화”를 참조하십시오.

2 MC/ServiceGuard 하드웨어 구성 이해

이 장에서는 MC/ServiceGuard 하드웨어 구성 요소가 작동하는 방식을 개괄적으로 설명합니다. 이 장에서는 다음과 같은 내용을 다룹니다.

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

MC/ServiceGuard 소프트웨어 구성 요소에 대해서는 다음 장을 참조하십시오.

클러스터 구성 요소 중복

고가용성을 제공하기 위해 클러스터는 일반적으로 둘 이상의 SPU 및 독립 디스크 등 중복 시스템 구성 요소를 사용합니다. 이러한 중복 시스템으로 단순 장애 요인을 제거할 수 있습니다. 일반적으로 중복 구성 요소가 많을수록 장애가 발생했을 때 응용 프로그램, 데이터, 지원 서비스에 대한 액세스 가능성이 높아집니다. 하드웨어 중복에 추가하여 장애 발생 시 응용 프로그램을 다른 SPU 또는 네트워크로 전송하고 제어할 수 있는 소프트웨어의 지원도 필요합니다. MC/ServiceGuard는 다음과 같은 소프트웨어 지원을 제공합니다.

- LAN에 장애가 발생하면 MC/ServiceGuard는 대기 LAN으로 전환하거나 영향을 받은 패키지를 대기 노드로 이동시킵니다.
- SPU에 장애가 발생하면 응용 프로그램은 지체 없이 장애가 발생한 SPU에서 사용 가능한 SPU로 자동 전송됩니다.
- 디스크 인터페이스 같이 모니터링 중인 다른 자원에 장애가 발생하면 패키지를 다른 노드로 이동시킬 수 있습니다.
- 소프트웨어 장애의 경우 응용 프로그램을 현재 노드에서 다시 시작하거나 최소한의 중지로 다른 노드에서 다시 시작할 수 있습니다.

또한 MC/ServiceGuard는 시스템 관리, 유지 보수, 버전 업그레이드 등의 이유로 SPU를 중지해야 할 때 다른 SPU로 응용 프로그램의 제어를 쉽게 전송할 수 있도록 합니다.

현재 MC/ServiceGuard 클러스터에서 지원되는 최대 노드 수는 16개입니다. 공유(다중 초기자) 버스에서 Fast/Wide SCSI 디스크 또는 디스크 어레이는 동시에 최대 4개의 노드에 연결할 수 있습니다. Fibre 채널을 사용하지만 공유 버스를 사용하지 않는 디스크 어레이(HP SureStore EXP 시리즈 및 EMC Symmetrix 등)는 동시에 모두 16개의 노드에 연결할 수 있습니다.

패키지 장애 조치에 대한 지침은 클러스터에 사용된 디스크 기술 종류에 따라 다릅니다. 예를 들어, Fast/Wide SCSI 디스크나 디스크 어레이의 데이터에 액세스하는 패키지는 장애 발생 시 최대 4개 노드로 전송될 수 있습니다. Fibre 채널이나 HP SureStore XP 또는 EMC Symmetrix 디스크 기술을 사용하는 클러스터에 있는 디스크의 데이터에 액세스하는 패키지는 장애 발생 시 16개의 노드로 전송되도록 구성할 수 있습니다.

공유 버스에 있는 디스크의 데이터에 액세스**하지 않는** 패키지는 디스크 기술에 관계없이 클러스터에서 구성된 모든 노드로 전송되도록 구성할 수 있습니다. 예를 들어, 로컬 서비스만 실행하는 패키지인 경우 장애 발생 시 디스크 연결 방식의 종류에 관계 없이 로컬 서비스의 사본이 포함된 클러스터의 모든 노드로 전송되도록 구성할 수 있습니다.

네트워크 구성 요소의 중복

네트워크에서 단순 장애 요인을 제거하려면 클러스터 노드가 액세스하는 각 서브넷은 중복된 네트워크 인터페이스를 사용해야 합니다. 또한, 케이블 장애에 대비하기 위해 케이블도 중복되도록 구성해야 합니다. 각 인터페이스 카드는 서로 다른 케이블로 연결하고 각 케이블은 허브나 브리지 등의 구성 요소로 연결합니다. FDDI 네트워크의 경우 각 인터페이스 카드는 케이블을 통해 서로 다른 집중 장치로 연결합니다. 이렇게 실제 케이블을 브리지, 집중 장치, 스위치 등을 통해 다른 케이블과 연결하는 구성 방식을 **브리지 연결 네트워크(Bridged Net)**라고 합니다.

브리지 연결 네트워크의 인터페이스에 IP 주소를 부여할 수 있습니다. 관련된 IP 주소가 있는 인터페이스를 **기본 인터페이스(Primary Interface)**라고 하며 IP 주소가 부여되지 않은 인터페이스를 **대기 인터페이스(Standby Interface)**라고 합니다. 대기 인터페이스는 기본 인터페이스에 장애가 발생했을 때 MC/ServiceGuard에 의해 대신 사용되는 인터페이스입니다. MC/ServiceGuard는 기본 인터페이스에서 장애를 발견하면 장애가 발생한 인터페이스 카드의 IP 주소와 관련 연결을 사용 가능한 대기 인터페이스 카드로 전환합니다.

네트워크 구성 선택에 대한 내용은 다음 절에서 자세히 설명합니다. 지원되는 네트워크의 전체 목록은 사용하는 제품의 **MC/ServiceGuard 릴리즈 노트**를 참조하십시오.

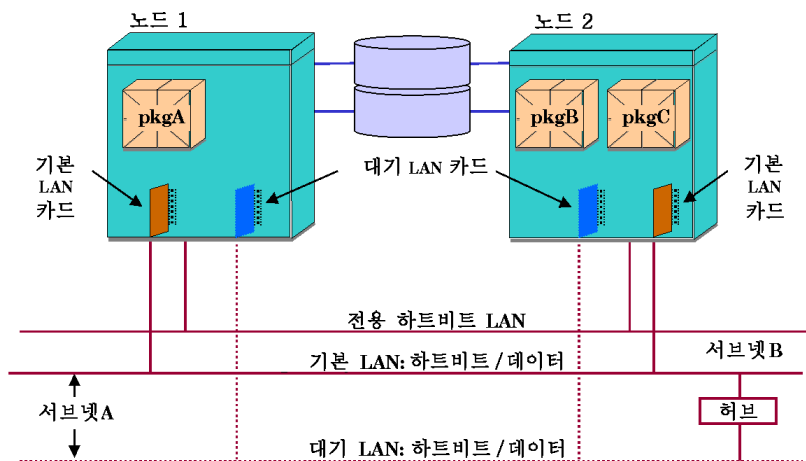
주

Fibre 채널은 더 이상 하트비트나 데이터 LAN으로 지원되지 않습니다.

중복 이더넷 구성

그림 2-1은 이더넷으로 구성된 중복 네트워크 구성 요소의 사용을 보여줍니다. 토론링도 비슷한 방법으로 구성됩니다.

그림 2-1 중복 LAN



그림에서 2개의 노드를 사용하는 MC/ServiceGuard 클러스터는 데이터/하트비트 서브넷(서브넷 A)을 위해 기본 및 대기 LAN 카드를 사용하여 구성한 하나의 브리지 연결 네트워크를 포함합니다. 또 다른 LAN 카드는 선택적으로 하트비트 전용 LAN을 제공합니다. 기본 LAN과 대기 LAN이 허브로 연결되어 중복 데이터/하트비트 서브넷을 구성하고 있습니다. 각 노드는 서브넷에 대해 고유한 IP 주소를 갖습니다. 데이터/하트비트 서브넷을 위한 기본 LAN 카드에 장애가 발생하면 MC/ServiceGuard에 의해 동일한 노드에 있는 대기 LAN 카드로 전환됩니다.

중복 하트비트는 하트비트가 전송되는 기본 LAN과 하트비트 전용 LAN을 통해서 구성됩니다. 그림 2-1에서 하트비트 전용 LAN은 다른 서브넷을 통한 중복 경로가 있기 때문에 로컬 스위칭이 필요 없습니다. 하트비트 전용 LAN은 기본 LAN에서 데이터가 정체되어 하트비트 장애로 잘못 진단되는 일을 방지합니다. 각 노드는 전용 하트비트 LAN에 대해 고유한 IP 주소를 갖습니다.

주

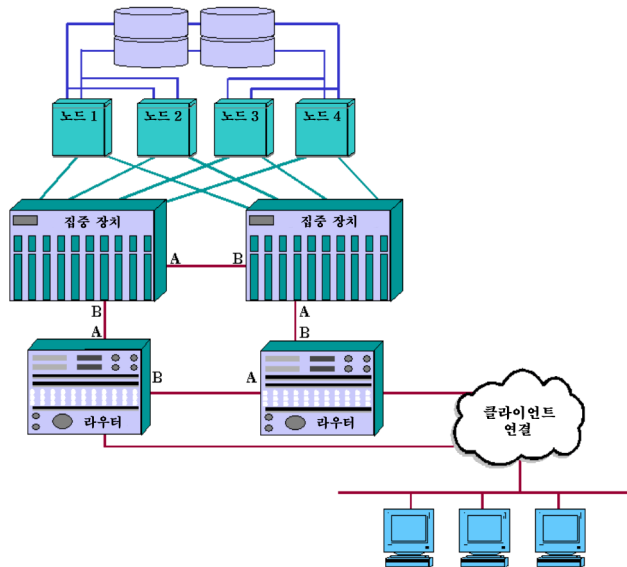
하트비트/데이터 LAN에서 네트워크 통신량이 너무 많지 않은지 확인해야 합니다. 통신량이 너무 많으면 하트비트 전용 LAN에 장애가 발생할 경우 LAN에서 하트비트가 제대로 전송되지 않을 수 있습니다.

중복 FDDI 연결 구성

FDDI는 고속 광섬유 연결 매체입니다. FDDI를 사용하면 스타 토폴로지를 사용하여 모든 노드를 두 개의 집중 장치에 연결하고 이 집중 장치를 다시 클러스터 외부의 네트워크와 통신하는 두 개의 라우터에 연결하여 중복 구성을 만들 수 있습니다. 이 경우 노드마다 두 개의 FDDI 카드를 사용합니다. 그림 2-2는 이 구성을 보여줍니다. 그림에서 집중 장치는 포트 A와 포트 B에 교차 연결된 이중 케이블로 서로 연결됩니다. 라우터는 양쪽 집중 장치 모두에 패킷을 보내도록 구성되어야 합니다.

그림 2-2

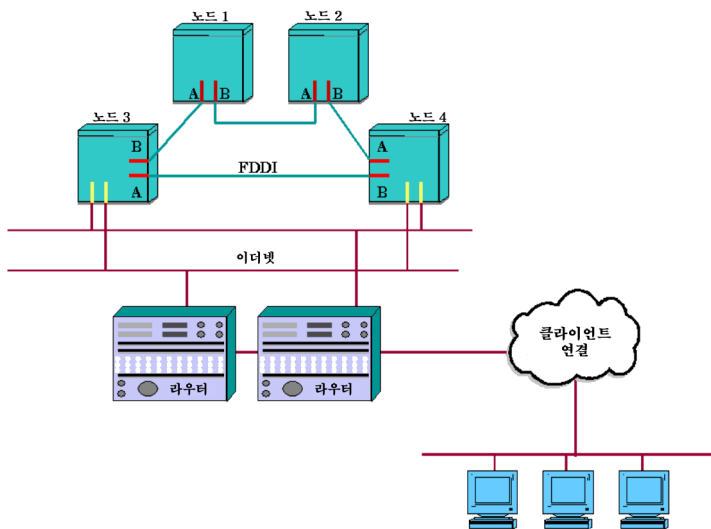
중복 FDDI 구성



이중 부착 FDDI 스테이션 사용

중복 FDDI 연결을 만드는 또 다른 방법은 그림 2-3과 같이 각 노드에 이중 부착 스테이션을 구성하여 FDDI 연결을 만드는 것입니다. 이 구성은 시스템 캐비닛의 슬롯을 하나만 사용한다는 장점이 있습니다. 또한, 그림 2-3에서 볼 수 있듯이 노드 3과 노드 4는 클러스터 외부의 네트워크에 연결하기 위해 이더넷도 사용합니다.

그림 2-3 이중 부착 FDDI 스테이션을 사용한 구성



이중 부착 카드를 사용하면 케이블과 커넥터의 장애로부터는 보호할 수 있지만 카드의 장애로부터는 보호할 수 없습니다. LAN 카드에서 장애가 발생하면 패키지는 다른 노드로 전송됩니다.

직렬(RS232) 하트비트 회선 사용

직렬(RS232) 하트비트 회선 MC/ServiceGuard는 하트비트 전용으로 직렬(RS232) 통신을 사용하는 2개로 구성된 노드를 지원합니다. 중복 하트비트 데이터를 제공하는 대체 인터페이스로 이 구성을 선택할 수 있습니다.

주

직렬(RS232) 하트비트 회선은 2개로 노드를 사용하는 클러스터 구성에서만 사용할 수 있습니다. 직렬 하트비트 회선은 1개의 하트비트 LAN을 통해 2개의 노드를 사용하는 클러스터에서 **사용할 수 있습니다**. 2개 이상의 하트비트 LAN이 있거나 1개의 하트비트 LAN과 1개의 대기 LAN이 있는 경우에는 직렬(RS232) 하트비트를 **사용할 수 없습니다**.

한 노드에서 하트비트 네트워크 카드에 장애가 발생하는 경우, 하트비트를 사용하면 LAN 컨트롤러 카드의 상태를 검사할 동안 클러스터를 계속 실행할 수 있고, 사용 가능한 노드로 전환하여 모든 패키지를 다시 실행하는 동안 장애가 발생한 네트워크에 연결된 노드를 중지시킬 수 있습니다.

MC/ServiceGuard 하드웨어 구성 이해 네트워크 구성 요소의 중복

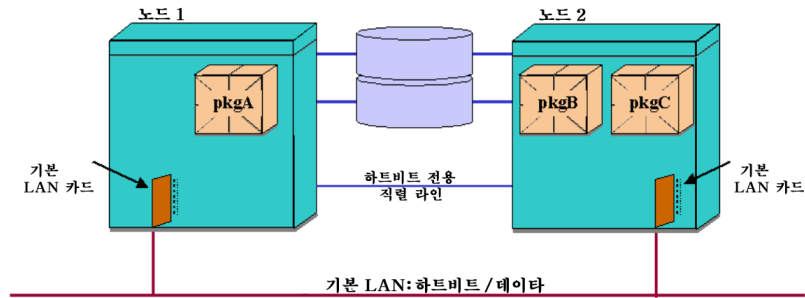
중복 하트비트를 위해 직렬(RS232) 회선을 구성한 경우에도 하트비트 신호를 전달할 하나의 LAN이 필요합니다. MC/ServiceGuard는 클러스터 구성원 사이의 통신에 TCP/IP를 사용하기 때문에 직렬 라인 하트비트는 네트워크 통신량의 증가로 인한 정체로부터는 보호할 수 있지만 네트워크 장애로부터는 보호할 수 없습니다.

직렬(RS232) 회선은 이더넷이나 FDDI 같은 TCP/IP 프로토콜을 실행하는 네트워크 카드보다 불안정합니다. TCP/IP와는 달리 직렬 회선 프로토콜은 오류 수정이나 다시 시도 장치가 없습니다. 또한 표준이 부족하여 구성이 복잡하고 어렵습니다.

그림 2-4는 직렬(RS232) 하트비트 회선을 보여줍니다.

그림 2-4

직렬(RS232) 하트비트 회선



장애가 발생한 네트워크 카드의 교체

시스템 구성에 따라, 장애가 발생한 네트워크 카드를 클러스터의 실행 중에 교체할 수 있습니다. 교체 과정에 대해서는 “클러스터 문제 해결” 장의 “LAN 카드 교체” 절을 참조하십시오.

중복 디스크 저장 장치

클러스터의 각 노드에는 자체적인 루트 디스크가 있지만 각 노드는 둘 이상의 노드가 패키지와 관련된 데이터 및 프로그램에 액세스하는 것과 같은 방법으로 일부 다른 디스크에도 실제적으로 연결되어 있습니다. 이러한 액세스는 논리 볼륨 관리자(LVM), VERITAS 볼륨 관리자(VxVM) 또는 VERITAS 클러스터 볼륨 관리자와 같은 저장 관리자에서 제공합니다. 한 디스크 저장 그룹은 한 번에 단 하나의 노드에서만 활성화될 수 있지만 패키지가 이동하는 경우에는 대체 노드가 저장 그룹을 활성화시킬 수 있습니다. 패키지에 속하는 저장 그룹에 있는 모든 디스크는 원래의 노드 및 사용 가능한 모든 대체 노드에 연결되어야 합니다. 디스크 저장 장치는 RAID나 소프트웨어 미러링을 사용하여 중복 구성합니다.

지원되는 디스크 인터페이스

둘 이상의 노드에 연결되는 디스크(공유 데이터 디스크)를 위해 MC/ServiceGuard가 지원하는 인터페이스의 종류는 다음과 같습니다.

- SE(Single-ended) SCSI
- Fast/Wide SCSI
- Fibre 채널

모든 SCSI 디스크가 지원되지는 않습니다. 현재 지원되는 디스크 목록은 *HP 9000 Servers Configuration Guide*를 참조하십시오(HP 대리점을 통해 구할 수 있음).

주

PCI SCSI 어댑터를 사용하는 시스템이 설치된 클러스터에서는 PCI와 NIO SCSI 어댑터 모두를 같은 공유 SCSI 버스에 동시에 부착할 수 없습니다.

공유 버스에 연결된 디스크를 사용할 경우 외부 공유 Fast/Wide SCSI 버스에 반드시 인라인 터미네이터를 장착해야 합니다. 자세한 내용은 “문제점 해결” 장을 참조하십시오.

SCSI 버스 우선 순위를 계획하고 할당할 때는 공유 버스의 각 노드에 대해 컨트롤러에 할당된 SCSI 주소에 따라 여러 노드가 공유하는 버스를 한 노드가 독점할 수 있다는 점에 주의하십시오. 인터페이스 카드의 주소를 포함하여 모든 SCSI 주소는 공유 버스 상의 다른 장치에 대해서 고유해야 합니다. SCSI 버스 주소 지정 및 우선 순위에 대한 자세한 내용은 **주변 장치를 위한 HP-UX 구성** 설명서를 참고하십시오.

데이터 보호

다음의 방법 중 하나를 사용하여 고가용성 시스템의 데이터를 보호할 것을 권장합니다.

- 디스크 미러링
- RAID 레벨 및 다중 데이터 경로를 사용한 디스크 어레이

디스크 미러링

디스크 미러링은 데이터 보호를 위한 방법 중 하나입니다. MC/ServiceGuard 패키지를 위해 사용하는 논리 볼륨은 반드시 미러링되어야 합니다. MC/ServiceGuard는 디스크의 데이터를 보호하지 않습니다. 이 기능은 LVM 저장의 경우 HP의 MirrorDisk/UX 제품에 제공되며, VxVM 및 CVM의 경우 VERITAS 볼륨 관리자(B9116AA)에 제공됩니다. 소프트웨어 미러링을 사용하여 논리 볼륨을 구성하면 각 미러링된 세트의 볼륨은 완전히 똑같은 데이터를 수록하게 됩니다. 디스크 중 하나에 장애가 발생하면 저장 관리자가 자동으로 다른 디스크 미러에 액세스하여 데이터의 가용성을 유지합니다. 온라인 백업이나 더 높은 수준의 가용성을 얻기 위해 LVM(또는 VxVM이 있는 추가 플렉스)의 3중 미러링을 사용할 수도 있습니다.

Fibre 채널이나 SCSI 버스 장애로부터 보호하려면 별도의 SCSI 버스에서 데이터의 각 사본을 액세스해야 합니다. 즉, 디스크 드라이브에 있는 데이터의 모든 사본을 동일한 버스에 연결하면 안됩니다.

루트 디스크의 미러링도 좋은 방법이지만 데이터 디스크의 미러링은 고가용성을 위해 매우 중요합니다. 한 SPU의 루트 디스크에 장애가 발생하면 클러스터의 다른 SPU가 응용 프로그램의 제어를 이어받습니다. 하지만 미러링되어 있지 않은 데이터 디스크에 장애가 발생하면 이 디스크를 사용하는 모든 응용 프로그램은 디스크를 수리하기 전까지 사용할 수 없게 됩니다. 최악의 경우 고장난 디스크의 데이터를 복구하지 못할 수도 있습니다.

RAID 레벨 및 다중 데이터 경로를 사용한 디스크 어레이

데이터를 보호하기 위한 다른 방법에는 RAID 레벨 1 또는 RAID 레벨 5 같이 데이터 중복을 제공하는 하드웨어 RAID 레벨로 디스크 어레이를 사용하는 방법이 있습니다. 디스크 어레이는 디스크의 데이터를 중복 구성합니다. 이러한 보호는 각 노드와 어레이 사이에 중복 호스트 버스 인터페이스(SCSI 또는 Fibre 채널)를 사용하여 보완해야 합니다. LVM의 PV 링크 기능이나 VxVM의 동적 다중 경로 지정(DMP) 기능을 사용하여 구성되는 중복 인터페이스는 I/O 채널에서 발생하는 단순 장애 요인으로부터 보호하며, RAID 1 또는 5 구성은 저장 매체를 중복 구성합니다. PV 링크를 LVM에서는 대체 링크, VxVM에서는 다중 경로라고도 합니다.

DMP는 VxVM의 별도 구성 요소로 사용할 수 있습니다. 활성/활성 장치용 DMP는 B9116AA가 필요하지만, 활성/비활성 장치용 DMP는 기본 제품인 B7961AA 없이도 사용할 수 있습니다.

이벤트 모니터링 서비스를 통한 디스크 모니터링

LVM을 사용하면 별도 제품(B5735AA)으로 제공되는 EMS HA Monitor의 디스크 모니터링 기능을 사용하여 장애가 있는 장치를 찾아내는 디스크 모니터링을 구성할 수 있습니다. 모니터링은 패키지 장애 조치를 시작하거나 ClusterView 같은 응용 프로그램에 디스크 고장 이벤트를 보고하도록 설정할 수 있습니다. 자세한 내용은 *Using HA Monitors* 설명서(제품 번호 B5736-90006)를 참조하십시오.

장애가 발생한 디스크 교체

미러링은 데이터를 보호하지만 디스크에 장애가 발생한 다음에는 이 디스크를 교체해야 합니다. 일반 디스크의 경우 클러스터를 중지한 다음에 장치를 교체합니다. 특수한 HA 디스크 엔클로저를 사용하는 디스크 어레이의 경우에는 클러스터가 작동 중이고 응용 프로그램이 온라인인 상태에서 디스크를 교체할 수 있습니다. 교체 과정에 대한 자세한 내용은 “클러스터 문제점 해결” 장의 “디스크 교체” 절을 참조하십시오.

장애가 발생한 I/O 카드 교체

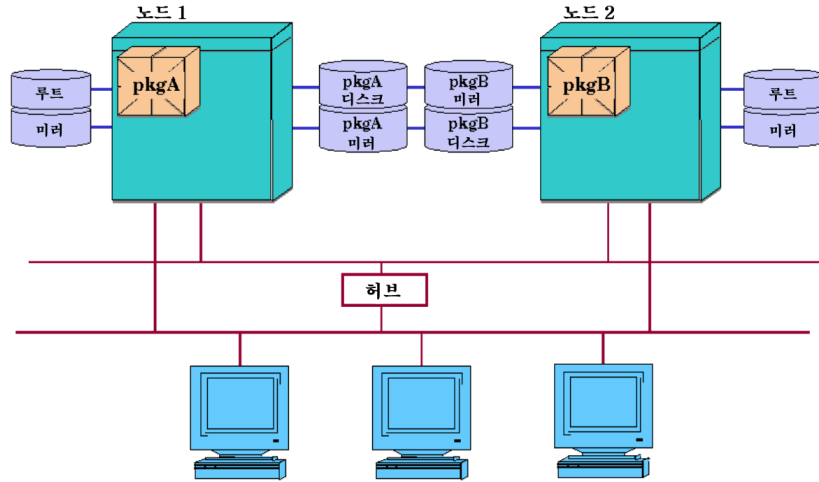
시스템 구성에 따라, 장애가 발생한 디스크 I/O 카드를 시스템이 온라인인 상태에서 교체할 수 있습니다. 교체 과정에 대한 자세한 내용은 “클러스터 문제점 해결” 장의 “I/O 카드 교체” 절을 참조하십시오.

SCSI 디스크 구성 예제

그림 2-5는 두 개의 노드를 사용하는 클러스터를 보여줍니다. 각 노드는 하나의 미러링된 루트 디스크와 주 노드로서 하나의 패키지를 포함합니다. 다른 노드로부터 패키지를 받을 수 있도록 각 노드에는 자원이 할당되어 있습니다. 각 패키지에는 할당된 하나의 디스크 볼륨 그룹이 있으며 볼륨 그룹에 있는 논리 볼륨은 미러링되어 있습니다. 패키지 A의 디스크와 패키지 B의 디스크 미러가 한 인터페이스에 있는 반면, 패키지 B의 디스크와 패키지 A의 디스크 미러는 별도의 버스에 있습니다. 이러한 배치는 단순 장애 요인을 제거하며 버스 중 하나에서 장애가 발생할 경우 디스크 또는 디스크 미러를 사용할 수 있게 합니다.

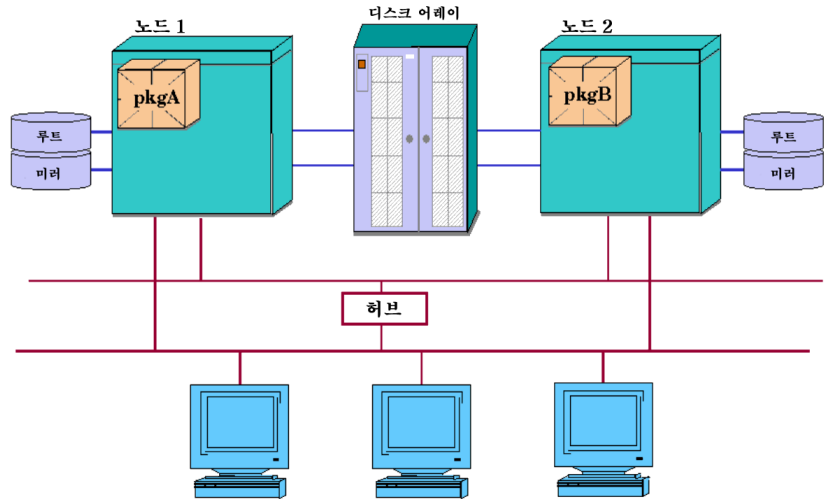
그림 2-5

고가용성을 위해 연결된 미러링 디스크



아래 그림 2-6은 두 개의 I/O 채널의 각 노드에 연결된 디스크 어레이를 사용하는 유사한 클러스터를 보여줍니다. 이러한 구성 방식은 LVM의 PV 링크 또는 VERITAS 동적 다중 경로(DMP)나 EMC PowerPath와 같은 다른 다중 경로 지정 소프트웨어를 사용합니다.

그림 2-6 고가용성 디스크 어레이를 사용한 클러스터



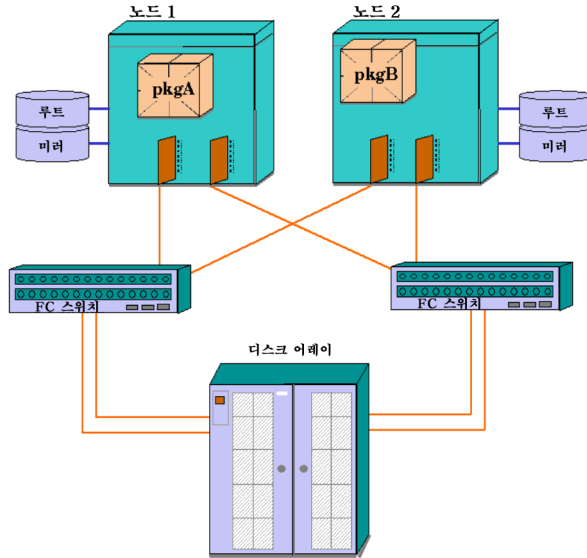
PV 링크를 비롯하여 MC/ServiceGuard의 논리 볼륨 구성에 대한 자세한 내용은 “HA(고가용성) 클러스터 구성 만들기” 장을 참조하십시오.

Fibre 채널 디스크 구성 예제

아래 그림 2-7에서 루트 디스크는 간단한 미러링으로 표시되어 있지만 공유 저장 장치는 디스크 어레이에 부착된 중복 Fibre 채널 스위치를 통해 액세스됩니다. 케이블이 연결되어 각 노드는 양쪽 스위치에 연결되고 두 스위치는 중복 어레이가 있는 디스크 어레이에 연결됩니다.

그림 2-7

Fibre 채널 전환 디스크 어레이가 있는 클러스터



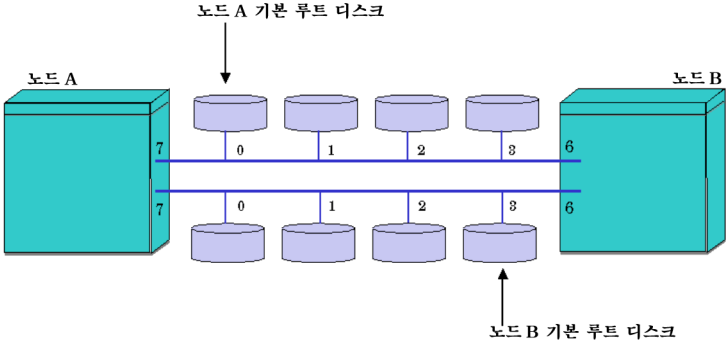
이러한 구성 방식은 PV 링크 또는 VERITAS 동적 다중 경로(DMP)나 EMC PowerPath와 같은 다른 다중 경로 지정 소프트웨어를 사용합니다.

공유 SCSI 버스의 루트 디스크 제한 사항

IODC 펌웨어는 동일한 SCSI 버스에서 동시에 둘 이상의 노드가 시작하는 것을 지원하지 않습니다. 따라서 각 클러스터에서 동일한 SCSI 버스에 둘 이상의 루트 디스크를 부착하면 안됩니다.

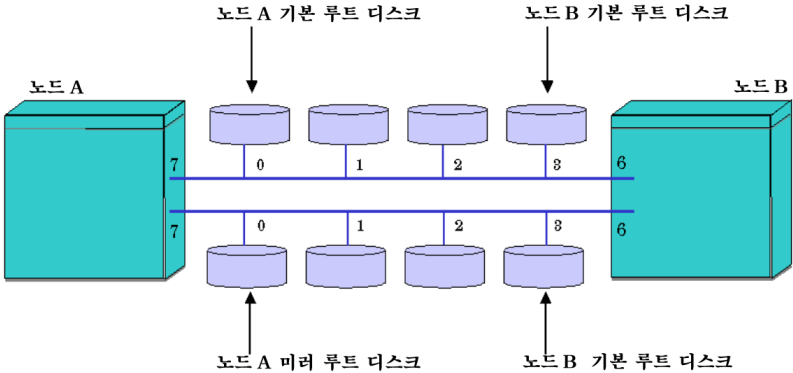
그림 2-8은 올바른 구성 예로써, 두 개의 노드는 외부 SCSI 버스를 공유하며 노드 A는 버스에 연결된 기본 루트 디스크를 포함하고 노드 B는 다른 버스에 연결된 기본 루트 디스크를 포함합니다. 0번에서 3번 그리고 6번과 7번은 서로 다른 버스의 SCSI 주소입니다.

그림 2-8 서로 다른 공유 버스 상의 루트 디스크



두 노드의 기본 루트 디스크를 동일한 버스에 연결하는 구성은 지원되지 않습니다. 양쪽 시스템을 동시에 시작하려면 두 시스템에 대해 세가지 장애가 발생해야 하는데 그 가능성은 희박하기 때문에 노드 B의 루트 디스크 미러 사본을 노드 A의 기본 루트 디스크와 동일한 SCSI 버스에 위치시키는 것은 가능합니다. 그 가능한 경우를 예를 들면, 노드 B의 기본 루트 디스크에서 장애가 발생하여 다시 시작할 때 노드 A 역시 다시 시작한다면 IODC 펌웨어에 문제가 발생합니다. 그림 2-9는 이 구성을 보여줍니다.

그림 2-9 서로 다른 공유 버스에 존재하는 기본 디스크와 미러 디스크



어레이가 공유 디스크에 존재하면 디스크 어레이 내의 디스크를 루트 디스크로 사용할 수 없습니다.

중복 전원 공급 장치

노드와 디스크의 배터리를 백업하여 하드웨어의 가용성을 확장할 수 있습니다. HP PowerTrust 같이 HP가 지원하는 무정전 전원 공급 장치(UPS)를 사용하면 정전 피해로부터 보호됩니다.

미러 사본을 서로 다른 전원에 연결하는 방법과 동일한 방법으로 디스크를 전원 회로에 연결해야 합니다. 부팅 디스크는 해당 노드와 동일한 회로에 연결합니다.

특히, 클러스터 잠금 디스크(클러스터 재생성 시 타이 브레이커(tie-breaker)로 사용됨)에는 중복 전원 공급 장치가 있어야 합니다. 그렇지 않으면 클러스터에 있는 노드가 아닌 다른 전원 공급 장치의 전원이 사용될 수 있습니다. 클러스터의 전원 공급 장치, 디스크 및 LAN 하드웨어에 관한 자세한 내용은 HP 대리점에 문의하십시오.

여러 개의 디스크 어레이와 다른 랙 시스템에는 장치에 있는 다른 전원 공급 장치가 별도의 전원 회로에 연결되어 구축된 여러 개의 전원 공급 장치가 있습니다. 일반적으로 두 개 또는 세 개의 전원 공급 장치가 있는 장치는 전원 회로 중 하나에 장애가 발생해도 계속 작동합니다. 그러므로, 클러스터에 있는 모든 하드웨어에 두 개 또는 세 개의 전원 공급 장치가 있을 경우 적어도 별도로 세 개의 전원 회로가 있어야 클러스터의 전원 회로 설계의 한 곳에서도 장애가 발생하지 않습니다.

보다 큰 클러스터

MC/ServiceGuard에서 최대 16개의 노드로 구성된 클러스터를 만들 수 있습니다. 16개의 노드로 구성된 클러스터는 이더넷을 통해 각 SPU를 연결하여 만들 수 있으며 FDDI 네트워크를 사용하면 최대 8개의 시스템을 MC/ServiceGuard 클러스터로 구성할 수 있습니다.

하지만 16개의 노드로 구성되는 클러스터를 만들 수 있다고 해서 모든 종류의 클러스터 구성이 16개의 노드 구성과 동일한 방식으로 동작한다는 의미는 아닙니다. 예를 들어, 공유 F/W SCSI 버스인 경우 버스의 부하와 케이블의 길이 제한으로 인해 동일한 공유 버스에 연결할 수 있는 노드 수의 실제 한계는 4개입니다. 그러나, 이 경우에도 16개의 노드로 구성된 클러스터를 관리 단위로 설정하고 다른 대용량 저장 장치에 부착된 다른 SCSI 버스에 4개의 노드로 구성되는 하위 그룹을 설정할 수 있습니다.

XP 시리즈나 EMC 디스크 어레이에 연결하는 비공유 F/W SCSI 연결의 경우에는 4개의 노드 제한이 적용되지 않습니다. 각 노드는 PV 링크를 지원하는 두 개의 F/W SCSI 버스를 통해 XP 또는 EMC에 직접 연결할 수 있습니다. 패키지는 16개의 모든 노드에서 장애 조치를 수행하도록 구성할 수 있습니다. 이 구성 방식에 대한 자세한 내용은 아래에서 “저장 장치에 지점간 연결”을 참조하십시오.

주

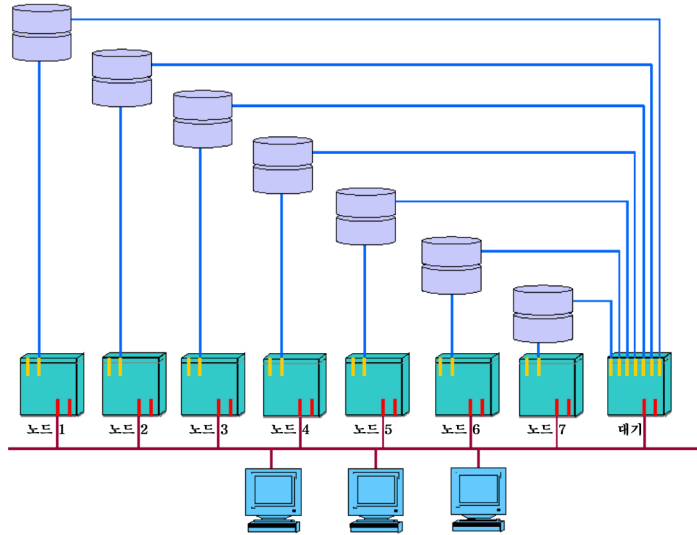
큰 규모의 클러스터를 구성할 때는 클러스터 및 패키지 구성 시간뿐만 아니라 `cmviewcl` 같은 명령의 실행 시간도 길어질 수 있다는 점에 주의하십시오. 일부 명령 옵션을 사용하면 이 문제를 해결할 수 있습니다. 자세한 내용은 온라인 도움말에서 `cmquerycl`을 참조하십시오.

활성/대기 모델

대기 노드가 포함된 클러스터를 만들 수도 있습니다. 예를 들어, 8개의 노드를 사용하는 구성에서 한 노드가 다른 7개 노드에 대해 대기 노드의 역할을 할 경우, 각 활성 노드에 별도의 연결을 허용하는 7개의 공유 버스를 사용하는 백업 노드를 구성하여 쉽게 설정할 수 있습니다. 그림 2-10은 이 구성을 보여줍니다.

그림 2-10

8개의 노드로 구성된 활성/대기 클러스터

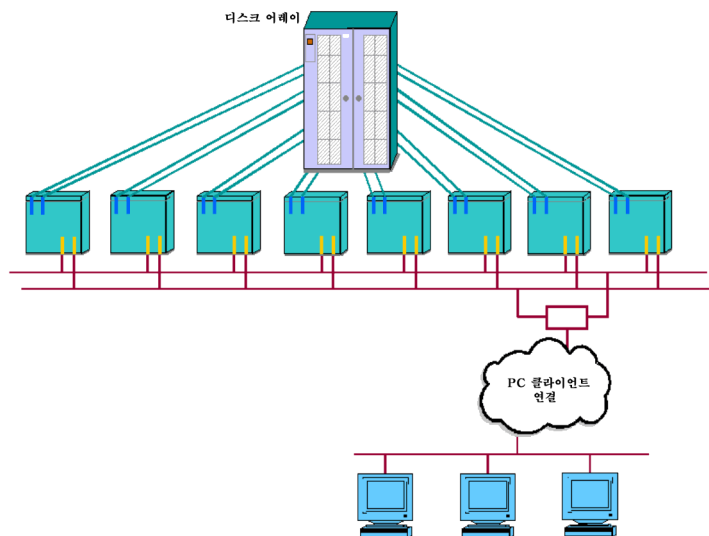


저장 장치에 지점간 연결

일부 저장 장치는 공유 SCSI 버스를 사용하지 않고도 다수의 호스트 노드로의 지점간 연결을 허용합니다. 그림 2-11은 Fibre 채널로 연결되어 있는 8개의 노드로 구성된 클러스터의 예제를 보여줍니다. 클라이언트 연결은 이더넷으로 제공됩니다. 노드는 16개의 I/O 포트가 구성된 XP 256 또는 EMC 디스크 어레이 구성에 있는 공유 데이터에 액세스합니다. 각 노드는 PV 링크로 구성된 두 개의 분리된 F/W SCSI 채널을 사용하여 어레이에 연결됩니다. 각 채널은 전용 버스이며 데이터 체인으로 연결되지 않습니다.

그림 2-11

XP 또는 EMC 디스크 어레이를 사용한 8개의 노드로 구성된 클러스터



Fibre 채널 전환 구성도 조정 루프나 패브릭 로그인 토폴로지를 사용하여 지원됩니다. 지원되는 클러스터 구성에 대한 자세한 내용은 *HP 9000 Servers Configuration Guide*를 참조하십시오. 이 설명서는 HP 대리점을 통해 구할 수 있습니다.

MC/ServiceGuard 하드웨어 구성 이해
보다 큰 클러스터

3 MC/ServiceGuard 소프트웨어 구성 요소 이해

이 장에서는 MC/ServiceGuard 소프트웨어 구성 요소가 어떻게 동작하는지 개괄적으로 설명합니다. 이 장에서는 다음과 같은 내용을 다룹니다.

- MC/ServiceGuard 아키텍처
- 클러스터 관리자 작동 방식
- 패키지 관리자 작동 방식
- 패키지 제어 스크립트의 작동 방식
- 네트워크 관리자의 작동 방식
- 데이터 저장을 위한 볼륨 관리자
- 장애 대응

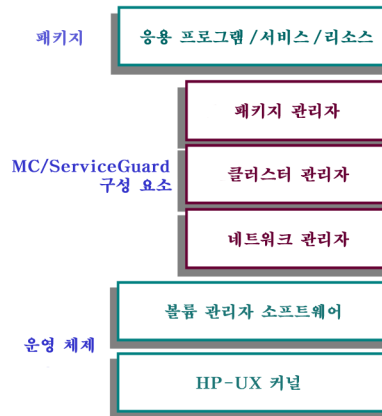
MC/ServiceGuard 클러스터 구성을 시작할 준비가 되었으면 이 장을 건너뛰고 4장, “HA(고가용성) 클러스터 계획 및 문서화”로 가십시오.

MC/ServiceGuard 아키텍처

다음 그림은 MC/ServiceGuard의 주요 소프트웨어 구성 요소를 나타냅니다. 이 장에서는 이러한 구성 요소에 대해 설명합니다.

그림 3-1

MC/ServiceGuard 소프트웨어 구성 요소



ServiceGuard 데몬

ServiceGuard와 관련된 9개의 데몬 프로세스는 다음과 같습니다.

- /usr/sbin/cmclconfd— ServiceGuard 구성 데몬
- /usr/sbin/cmclcd—ServiceGuard 클러스터 데몬
- /usr/sbin/cmlogd—ServiceGuard Syslog 로그 데몬
- /usr/sbin/cmlvmd—클러스터 논리 볼륨 관리자 데몬
- /opt/cmom/sbin/cmomd—클러스터 개체 관리자 데몬
- /usr/sbin/cmsnmpd—클러스터 SNMP 하위 에이전트(선택적으로 실행)
- /usr/sbin/cmsrvassistd—ServiceGuard 서비스 지원 데몬
- /usr/sbin/cmtaped—ServiceGuard 공유된 테이프 데몬
- /usr/sbin/qs—ServiceGuard Quorum 서버 데몬

표준 출력 파일(출력 파일 이름을 /var/adm/qs/qs.log로 재지정하는 것이 좋음)에 기록되는 quorum 서버와 /var/opt/cmom/cmomd.log에 기록되는 /usr/sbin/cmomd를 제외한 이들 데몬은 각각 /var/adm/syslog/syslog.log 파일에 기록됩니다.

구성 데몬: cmclconfd

이 데몬은 ServiceGuard 명령과 SAM 인터페이스를 사용하여 모든 클러스터 노드로부터 정보를 모읍니다. 즉, 네트워크 및 볼륨 그룹 정보와 같은 구성 정보를 모아서 클러스터 이진 구성 파일을 클러스터의 모든 노드로 배포합니다. 이 데몬은 inetd(1M)으로 시작되며, 항목은 /etc/inetd.conf 파일에 위치합니다.

클러스터 데몬: cmcld

이 데몬은 하트비트 메시지를 ServiceGuard 클러스터의 다른 노드에 있는 다른 cmcld 데몬으로 보내어 클러스터 구성원을 결정할 때 사용됩니다. 이 데몬은 실시간 우선 순위로 실행되어 메모리에 잠겨 있습니다. cmcld 데몬은 커널 중지를 검사하는 데 사용되는 **안전 타이머**를 커널에 설정합니다. 이 데몬이 정기적으로 타이머를 다시 설정하지 않으면 커널에서 시스템 제어 전송(TOC)이 실행되어 CPU를 재설정하고 크래시 덤프 파일을 생성합니다. 이 상황은 cmcld가 클러스터 구성원 과반수와 통신할 수 없는 경우 또는 cmcld가 예기치 않게 종료 또는 중단되거나 상당한 시간 동안 실행 불가능인 상태에서 커널 중지를 표시하는 커널 타이머를 업데이트할 수 없을 때에 발생할 수 있습니다. 안전 타이머의 시간 초과로 인한 TOC가 발생하기 전에 /var/adm/syslog/syslog.log와 커널의 메시지 버퍼에 메시지가 기록됩니다.

또한, cmcld 데몬은 시스템 네트워크 상태를 검사하고 로컬 lan 장애 조치를 수행합니다. 끝으로 ServiceGuard 패키지를 관리하고 패키지 실행 위치 및 시작 시간을 결정합니다.

주

MC/ServiceGuard의 주요 구성 요소는 패키지 관리자, 클러스터 관리자, 네트워크 관리자이며 이들은 cmcld 데몬의 일부로 실행됩니다. 이 데몬은 모든 클러스터 노드에서 우선 순위 20으로 실행됩니다. 따라서 사용자 프로세스의 우선 순위는 20보다 낮아야 합니다. 그렇지 않으면 MC/ServiceGuard가 커널 안전 타이머를 업데이트하지 못하기 때문에 노드에서 제어 전송(TOC)이 일어날 수 있습니다.

Syslog 로그 데몬: cmlogd

cmcltd 데몬은 cmcltd 데몬이 syslog에 메시지를 기록할 때 사용됩니다. cmcltd 데몬에 의해 syslog에 기록되는 모든 메시지는 cmlogd 데몬을 실행하여 기록됩니다. 이렇게 하면 syslog에 기록하는 작업이 지연되더라도 cmcltd 타이밍에 영향을 미치지 않습니다.

클러스터 논리 볼륨 관리자 데몬: cmlvmd

이 데몬은 클러스터를 인식하는 모든 볼륨 그룹을 추적합니다. 볼륨 그룹이 클러스터를 인식하면, 클러스터 노드는 단독 모드에서 볼륨 그룹을 활성화할 수 있습니다. 이렇게 하면 동시에 2개 이상의 노드에서 볼륨 그룹이 쓰기 모드로 활성화되지 않습니다.

클러스터 개체 관리자 데몬: cmomd

이 데몬은 클러스터 정보를 클라이언트(클러스터 개체 상태 정보에 의존하는 ServiceGuard Manager 같은 외부 제품이나 도구)에 제공합니다. 클라이언트는 개체 관리자에게 질의를 보내고 응답을 받습니다. 이 데몬은 시스템에서 실행되지 않으며 개체 관리자의 클라이언트에서만 사용할 수 있습니다.

cmomd 데몬은 클라이언트와의 연결을 허용하고 질의를 검사합니다. 질의는 여러 공급자가 제공하는 클래스 범주로 나뉩니다. 공급자는 연결된 모든 노드에 구성되어 있는 cmclconfd 데몬 등 여러 소스에서 데이터를 모은 다음, 정확한 클라이언트 질의의 요구를 충족시키기 위해 필터링된 중앙의 해당 위치로 데이터를 반환합니다. 이 데몬은 inetd(1M)으로 시작되며, 항목은 /etc/inetd.conf 파일에 위치합니다.

클러스터 SNMP 에이전트 데몬: cmsnmpd

이 데몬은 클러스터에 대한 정보를 얻기 위해 클러스터 보기(ClusterView)에서 사용됩니다. 이 데몬은 클러스터 MIB를 생성합니다. 클러스터 MIB에 관한 자세한 내용은 <http://docs.hp.com/hpux/ha>의 질문과 대답을 참조하십시오. 이 데몬은 /etc/rc.config.d/cmsnmpagt 파일을 편집하여 이 하위 에이전트를 자동으로 시작할 수 있을 경우에만 실행됩니다. 올바르게 실행하려면 ServiceGuard 클러스터를 시작하기 전에 cmsnmpd를 시작해야 합니다.

서비스 지원 데몬: cmsrvassistd

이 데몬은 cmcltd 클러스터 데몬이 요청하는 스크립트나 프로세스를 포크하고 실행합니다. 이 데몬이 수행하는 포크에는 다음 2가지 종류가 있습니다.

- 실행 패키지 실행 및 중지 스크립트
- 시작 서비스

서비스와 관련하여, `cmclsd` 데몬은 서비스 프로세스를 모니터링하고, 서비스 재시도 횟수에 따라 `cmsrvassistd`를 통해 서비스를 다시 시작하거나, 패키지를 중지하고 대체 노드로 이동시킵니다.

공유된 테이프 데몬: `cmtaped`

공유된 테이프 데몬은 클러스터의 일부인 공유된 모든 테이프 장치를 추적합니다. 공유된 테이프 장치는 `stapplyconf` 명령을 사용하여 구성할 수 있습니다.

Quorum 서버 데몬: `qs`

`quorum` 서버 데몬은 클러스터 재편성 도중 필요할 때 서비스를 재편성합니다. `quorum` 서버는 클러스터 외부 시스템에서 실행되며 `ServiceGuard`가 아닌 사용자에게 의해 시작됩니다. 이 데몬은 `/etc/inittab`에서 정상적으로 시작됩니다. 즉, 이 데몬에 오류가 발생하거나 중지되면 자동으로 다시 편성됩니다.

클러스터의 모든 구성원이 `quorum` 서버에 대한 연결을 시작하고 유지합니다. `quorum` 서버가 중단되면 `ServiceGuard` 노드가 이를 감지하고 `quorum` 서버가 정상적으로 작동할 때까지 정기적으로 `quorum` 서버에 연결을 시도합니다. 클러스터 재구성 중에 `quorum` 서버가 중지되고 재구성해야 하는 클러스터에 파티션이 있으면 재구성되지 않습니다.

클러스터 관리자 작동 방식

클러스터 관리자(Cluster Manager)는 클러스터를 초기화하고 클러스터의 이상 유무를 감시하고 노드의 장애를 감시하고 노드가 클러스터를 벗어나거나 클러스터에 합류할 때 클러스터의 재편성을 관리합니다. 클러스터 관리자는 각 노드에서 실행되는 데몬 프로세스로서 작동합니다. 클러스터가 시작되거나 재편성될 때 한 노드가 선택되어 **클러스터 코디네이터(Cluster Coordinator)**의 역할을 담당합니다. 모든 노드가 일부 클러스터 관리 기능을 수행하지만 클러스터 코디네이터가 노드간 통신의 중심 역할을 수행합니다.

클러스터 구성

시스템 관리자가 클러스터 구성 매개 변수를 설정하고 클러스터를 최초로 시작합니다. 그런 다음에는 정상적으로 작동되는 동안은 사용자의 조작이 없이 클러스터가 자동으로 작동합니다. 클러스터 구성 매개 변수에는 클러스터 이름과 노드, 클러스터 하트비트에 대한 네트워크 매개 변수, 클러스터 잠금 정보, 타이밍 매개 변수 등이 있습니다. i계 획i 장에 자세하게 설명되어 있습니다. 클러스터 매개 변수는 SAM을 사용하여 입력하거나 **클러스터 ASCII 구성 파일**(5장 참조)을 편집하여 입력합니다. 입력된 매개 변수는 이진 구성 파일을 작성하는 데 사용되며 작성된 이진 구성 파일은 클러스터에 속하는 모든 노드로 배포됩니다. 이진 클러스터 구성 파일은 클러스터 내의 모든 노드에서 동일해야 합니다.

하트비트 메시지

클러스터 관리자의 중심 역할은 클러스터에 속한 노드 간에 **하트비트 메시지(Heartbeat Message)**를 보내고 받는 것입니다. 클러스터의 각 노드는 하트비트 장치로 구성된 모니터된 TCP/IP 네트워크나 RS232 직렬 회선을 통해 클러스터 코디네이터와 하트비트 메시지를 교환합니다. LAN 모니터링에 대해서는 다음에 소개되는 “LAN 인터페이스 모니터링 및 장애 감지”를 참조하십시오.

클러스터 노드에 다른 클러스터 노드가 보내는 하트비트 메시지가 지정된 시간 내에도 도착하지 않으면 클러스터 재편성이 시작됩니다. 재편성이 끝났을 때 새로운 노드 집합이 클러스터를 구성하면 이 정보는 **패키지 코디네이터(Package Coordinator)**에 전달됩니다(자세한 내용은 “패키지 관리자의 작동 방식” 참조). 더 이상 새 클러스터에 존재하지 않는 노드에서 실행되고 있던 패키지는 대체 노드로 전송됩니다. 하트비트가 전송 중 중지될 경우에는 클러스터가 이전과 같은 노드로 재편성됩니다. 이 경우에는 패키지가 중지되거나 전환되지는 않지만 재편성 과정에서 응용 프로그램 성능에 일부 영향을 미칠 수 있습니다.

하트비트와 데이터가 같은 LAN 서브넷을 거쳐서 전송되는 경우 데이터 정체로 인해 하트비트의 제한 시간이 초과되면 MC/ServiceGuard가 하트비트를 잘못 인식하여 불필요한 클러스터 재편성을 시작할 수 있습니다. 이런 상황이 발생하지 않도록 하트비트 전송에 데이터 네트워크뿐만 아니라 하트비트 전용 네트워크를 사용할 수 있도록 구성하거나 직렬(RS232) 회선으로 하트비트를 실행할 수 있도록 구성하는 것이 좋습니다. 전용 LAN이 필수적인 요소는 아니지만 네트워크 상태를 확인하여 클러스터에서 하트비트가 손실될 가능성이 있으면 전용 LAN을 사용해야 합니다.

주

하트비트는 직렬 회선에서 직접 실행할 수는 없습니다. MC/ServiceGuard의 직렬 회선에 대한 자세한 내용은 2장의 “직렬(RS232) 하트비트 회선 사용” 절을 참조하십시오.

중요

다중 하트비트는 병렬로 전송됩니다. 클러스터 노드를 상호 연결하는 모든 서브넷을 하트비트 네트워크로 구성하는 것이 좋습니다. 이렇게 하면 추가 비용 없이 여러 장에 대한 보호를 향상시킬 수 있습니다. 하지만, VERITAS 클러스터 볼륨 관리자(CVM)를 사용할 경우에는 단일 하트비트 서브넷만 사용할 수 있습니다. 이 경우 하트비트는 대기 LAN을 사용하여 구성하거나 집계된 포트 그룹으로 구성해야 합니다. 아래의 “CVM에 필요한 단일 하트비트 서브넷”을 참조하십시오.

각 노드는 클러스터 하트비트 주기에 의해 결정되는 빈도로 하트비트 메시지를 보냅니다. 클러스터 하트비트 주기는 클러스터 구성 파일의 일부로서 작성한 **클러스터 구성 파일(Cluster Configuration File)**에 설정되어 있습니다. 자세한 내용은 “HA(고가용성) 클러스터 구성 파일 만들기”를 참조하십시오.

전체 클러스터의 수동 시작

수동 시작은 클러스터 구성의 모든 노드 중 하나의 클러스터를 편성합니다. 수동 시작은 일반적으로 최초로 클러스터를 시작할 때나 클러스터 전반에 걸친 유지 보수 또는 업그레이드, 구성 변경 후에 수행합니다.

시작하기 전에 동일한 내용의 이전 클러스터 구성 파일이 클러스터 내의 모든 노드에 존재해야 합니다. 시스템 관리자는 SAM에서 클러스터를 시작하거나 한 노드에서 `cmrunc1` 명령을 실행하여 클러스터를 시작합니다. `cmrunc1` 명령은 클러스터가 실행 중이지 않을 때, 즉 어떤 노드에서도 `cmc1d` 데몬이 실행되고 있지 않을 때만 사용할 수 있습니다.

클러스터 관리자 소프트웨어는 시작할 때 시작 명령에 지정된 노드가 모두 유효한 클러스터 구성원인지, 명령이 모두 시작되어 실행 중인지, 클러스터 편성을 시도하고 있는지, 서로 통신할 수 있는지를 검사합니다. 이상이 없으면 클러스터 관리자가 클러스터를 편성합니다.

자동 클러스터 시작

노드가 다시 시작되어 클러스터에 참여하면 클러스터가 자동으로 시작됩니다. 이 상황은 개별 노드가 다시 시작될 때, 또는 정전으로 인한 모든 SPU의 중지처럼 클러스터의 모든 노드에서 장애가 발생할 때 일어날 수 있습니다.

/etc/rc.config.d/cmcluster 파일에서 AUTOSTART_CMCLD 플래그가 1로 설정되어 있으면 클러스터가 자동으로 시작됩니다. 이 매개 변수가 1로 설정된 상태에서 노드가 다시 시작되면 노드는 기존 클러스터에 다시 참여하게 됩니다. 만약 기존 클러스터가 없다면 새 클러스터를 편성하려고 시도합니다.

동적 클러스터 재편성

동적 재편성은 노드가 실행 중인 클러스터에 참여하거나 클러스터를 떠날 때 발생하는 클러스터 소속의 일시적인 변경입니다. 재편성과 재구성은 다릅니다. 재구성은 구성 파일을 영구적으로 변경하는 것입니다. 다음(완전한 목록은 아님)과 같은 경우에 클러스터가 재편성됩니다.

- 활성 노드에서 SPU 또는 네트워크 장애가 감지된 경우
- 비활성 노드가 클러스터에 참여하려고 할 경우. 클러스터 관리자 데몬이 이 노드에서 시작된 경우
- 클러스터 구성에서 노드가 추가되거나 삭제된 경우
- 시스템 관리자가 노드를 중지시킨 경우
- 패키지 장애로 인해 노드가 중지된 경우
- 서비스 장애로 인해 노드가 중지된 경우
- 과도한 네트워크 통신량으로 인해 클러스터가 하트비트 신호를 받지 못한 경우
- 하트비트 네트워크에 장애가 발생했지만 다른 네트워크가 하트비트를 전송하도록 구성되지 않은 경우

일반적으로 재편성 후에는 클러스터 구성이 변경됩니다. 새 클러스터는 이전의 클러스터보다 노드 수가 적거나 많습니다.

분할 브레인 상황을 피하기 위한 클러스터 정족수

일반적으로 클러스터 재편성 알고리즘은 이전에 실행된 노드의 정확한 과반수(50% 이상)를 클러스터 정족수로 요구합니다. 이전에 실행 중이던 클러스터의 절반(정확히 50%) 모두 재편성이 허용되면 동일한 클러스터에서 2개의 인스턴스가 실행되는 **분할 브레인(split-brain)** 상황이 발생할 수 있습니다. 분할 브레인 상황에서는 응용 프로그램의 서로 다른 인카네이션이 동일한 디스크에 동시에 액세스하게 될 수 있습니다. 따라서 한 인카네이션은 복구 작업을 시작하는 한편 다른 인카네이션은 디스크 상태를 수정하는 상황이 생길 수 있습니다. 이러한 분할 브레인 상황을 방지하기 위해 ServiceGuard의 정족수 요건을 만들었습니다.

클러스터 잠금

일반적으로 클러스터 정족수로 50% 이상이 요구되지만, **이전에 실행 중이던 노드의 나머지 50% 역시 재편성되지 않는 경우**에는 이전에 실행 중이던 노드의 50%가 정확히 새 클러스터로 재편성될 수 있습니다. 2개의 동일한 크기의 노드 그룹을 선택하여 한 그룹은 클러스터를 편성하게 하고 다른 그룹은 종료하도록 하려면 반드시 과반수 이상이 필요합니다. 이러한 타이 브레이커(tie-breaker)를 **클러스터 잠금(Cluster Lock)**이라고 합니다. 클러스터 잠금은 **잠금 디스크** 또는 **quorum 서버**를 통해 구현됩니다.

클러스터 잠금은 실행 중이던 클러스터에 장애가 발생하여 ServiceGuard가 새로운 클러스터를 편성하려고 하는데, 클러스터가 동일한 크기의 두 하위 클러스터로 나뉘는 경우에 과반수를 만들기 위해서만 사용됩니다. 하위 클러스터는 모두 클러스터 잠금을 획득하려고 시도합니다. 클러스터 잠금을 획득한 하위 클러스터가 새로운 클러스터를 편성하여 두 개의 하위 클러스터가 동시에 실행되지 않도록 합니다. 두 하위 클러스터의 크기가 같지 않은 경우에는 노드 수가 50%를 넘는 하위 클러스터가 새로운 클러스터로 편성되며 클러스터 잠금은 사용되지 않습니다.

만약 2개의 노드를 사용하는 클러스터라면 클러스터 잠금을 구성해야 합니다. 두 노드 사이에서 통신이 중지되는 경우에는 클러스터 잠금을 사용하는 노드가 클러스터를 제어하고 다른 노드는 중지되거나 TOC를 실행합니다. 클러스터 잠금이 없다면 클러스터의 어느 한 노드에서 장애가 발생했을 때 다른 노드에도 장애가 발생하여 결과적으로 클러스터가 중지될 수 있습니다. 클러스터 잠금을 획득하려고 시도하는 중에 클러스터 잠금이 실패하면 클러스터는 중지됩니다.

잠금 요구 사항

1개의 노드로 구성된 클러스터는 클러스터 잠금이 필요하지 않습니다. 2개의 노드로 구성된 클러스터는 클러스터 잠금이 **필요합니다**. 4개 이상의 노드로 구성된 클러스터에서는 클러스터 잠금을 사용하는 것이 좋습니다. 5개 이상의 노드로 구성된 클러스터에서는 클러스터 잠금을 사용할 수 없지만 quorum 서버는 사용할 수 있습니다.

클러스터 잠금으로 사용하는 LVM 잠금 디스크

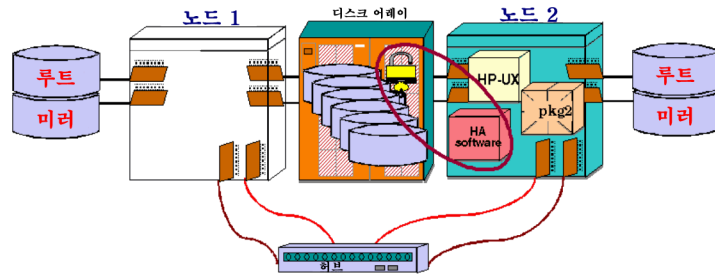
잠금 디스크는 4개 이하의 노드로 구성된 클러스터에 사용할 수 있습니다. 클러스터 잠금 디스크는 클러스터의 모든 노드가 공유하는 볼륨 그룹에 위치한 LVM 디스크의 특수 영역입니다. 한 노드에서 클러스터 잠금을 획득하면 잠금을 “사용 중”임을 다른 노드가 인식할 수 있도록 해당 영역이 표시됩니다.

잠금 디스크는 클러스터 잠금에만 사용하지 않으며, 사용자 데이터가 포함된 정상적 볼륨 그룹의 일부로 사용할 수 있습니다. 클러스터 잠금 볼륨 그룹 및 물리 볼륨 이름은 클러스터 구성 파일에서 식별됩니다.

그림 3-2는 잠금 디스크의 작동을 보여줍니다.

그림 3-2

잠금 디스크 작동



ServiceGuard는 잠금 디스크의 정상 작동을 정기적으로 검사하고, 잠금 디스크에서 정상 작동 검사를 하지 못하는 경우에는 syslog 파일에 메시지를 기록합니다. 잠금 디스크 문제를 조기에 검사하기 위해서는 이 파일을 모니터링해야 합니다.

선택한 고가용성 구성에 따라 단일 잠금 디스크 또는 이중 잠금 디스크를 선택합니다. **가능하다면 단일 잠금 디스크를 권장합니다.** 그러나, 단일 잠금과 이중 잠금의 경우 한 노드에 전원이 공급되지 않더라도 클러스터 잠금을 사용할 수 있어야 합니다. 따라서 잠금 구성을 선택할 때는 사용 가능한 전원 회로 수를 고려해야 합니다. 선택한 잠금에 상관 없이 고가용성을 유지하기 위해서 클러스터의 모든 노드가 클러스터 잠금에 액세스해야 합니다.

단일 잠금 디스크

권장되는 방식은 단일 잠금 디스크입니다. 단일 잠금 디스크는 클러스터에 속한 노드의 전원 회로와 분리된 전원 회로로 구성해야 합니다. 예를 들어, 2개의 노드로 구성된 클러스터의 경우에는 클러스터 잠금을 위한 하나의 분리된 전원을 사용하는 디스크를 포함하여 세 개의 전원 회로를 사용하는 것이 좋습니다. 2개의 노드로 구성된 클러스터에서 이 단일 잠금 디스크는 어떤 노드와도 전원 회로를 공유하지 않는 외부 디스크여야 합니다. 3개 또는 4개의 노드로 구성된 클러스터의 경우에 잠금 디스크는 50% 이상의 노드와 전원 회로를 공유하지 않아야 합니다.

이중 잠금 디스크

동일한 캐비닛에 내부적으로 마운트된 디스크를 클러스터 노드로 사용하면 잠금 디스크를 포함하는 노드에서 정전이 될 경우 클러스터 잠금에도 영향을 주기 때문에 단일 잠금 디스크는 이 종류의 클러스터에서 단순 장애 요인이 될 수 있습니다. 이와 유사하게, 두 개의 별도 데이터 센터에서 노드를 실행 중인 클러스터인 캠퍼스 클러스터에 치명적 장애가 발생하는 경우에 단일 잠금 디스크는 단순 장애 요인이 될 수 있습니다. **이 두 경우에만** 전원이 서로 분리된 두 개의 클러스터 디스크를 사용하는 이중 클러스터 잠금을 사용하여 잠금 디스크에서 단순 장애 요인을 제거합니다. 이중 클러스터 잠금에서 두 디스크가 전원 회로나 노드 새시를 공유해서는 안 됩니다. 이중 잠금을 사용하면 한 노드와 디스크에 전원 장애가 발생해도 나머지 노드와 디스크는 사용할 수 있으므로 남은 노드로 클러스터를 재편성할 수 있습니다. 캠퍼스 클러스터의 각 데이터 센터에는 잠금 디스크가 하나씩 있어야 하며, 모든 노드는 두 잠금 디스크 모두에 액세스해야 합니다. 한 데이터 센터에 장애가 발생하는 경우, 남은 데이터 센터의 노드에서 로컬 잠금 디스크를 획득하여 새 클러스터를 재편성할 수 있습니다.

주

이중 잠금 디스크는 중복 클러스터 잠금을 제공하지 않습니다. 실제로 이중 잠금이란 복합 잠금입니다. 이것은 단일 잠금 디스크에 필요한 한 개의 잠금 디스크보다는 클러스터 편성 시에 두 개의 잠금 디스크를 사용할 수 있어야 한다는 의미입니다. 그러므로 장애 발생 시에 클러스터 노드의 정확히 절반으로부터 단일 클러스터 잠금을 격리시킬 수 없을 때는 이중 클러스터 잠금을 사용하는 것이 좋습니다.

이중 잠금 디스크 중 하나에 장애가 발생하면 ServiceGuard는 정기 검사 시 해당 잠금 디스크를 검사하고 메시지를 `syslog` 파일에 기록합니다. 잠금 디스크 중 하나가 손실되면 클러스터 노드의 장애로 인해 클러스터가 중지될 수 있습니다.

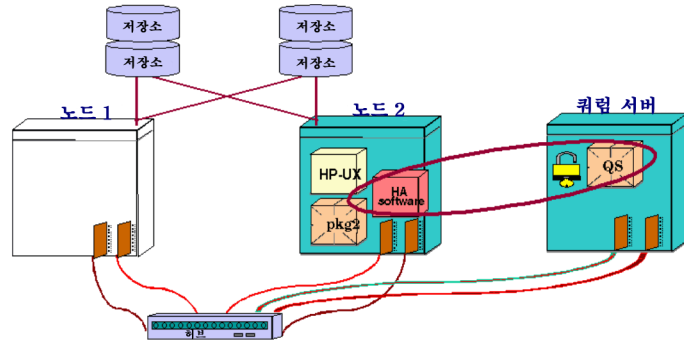
클러스터 잠금으로 사용하는 Quorum 서버

quorum 서버는 모든 규모의 클러스터에 사용할 수 있습니다. quorum 서버 프로세스는 quorum 서비스를 제공하는 클러스터 외부의 시스템에서 실행됩니다. quorum 서버는 알려진 포트의 ServiceGuard 노드에서 연결 요청을 수신합니다. 이 서버는 각 클러스터의 메모리에 특정 영역을 유지하고 한 노드에서 클러스터 잠금을 획득하면 잠금을 “사용 중”임을 다른 노드가 인식할 수 있도록 해당 영역이 표시됩니다. 같은 크기의 두 노드 그룹 사이에서 통신이 중지되면 Quorum 서버에서 클러스터 잠금을 획득한 그룹이 클러스터를 제어하고 다른 노드는 TOC를 실행합니다. 클러스터 잠금이 없다면 노드 그룹의 한 노드에서 장애가 발생했을 때 다른 그룹에서 장애가 발생하여 결과적으로 클러스터가 중지될 수 있습니다. 클러스터 잠금에 액세스하려고 시도하는 중에 quorum 서버를 사용할 수 없으면 클러스터가 중지됩니다.

그림 3-3은 quorum 서버의 작동을 보여줍니다. 노드 1과 노드 2 사이의 통신이 중지되면 quorum 서버는 클러스터에서 실행을 계속할 한 노드(예제에서는 노드 2)를 선택합니다. 다른 노드는 중지됩니다.

그림 3-3

Quorum 서버의 작동



quorum 서버는 별도 시스템에서 실행되며 여러 클러스터에 quorum 서비스를 제공할 수 있습니다.

클러스터 잠금 사용 안 함

일반적으로 3개 이하의 노드로 구성된 클러스터에는 클러스터 잠금을 사용해야 합니다. 2개의 노드로 구성된 클러스터에서는 반드시 클러스터 잠금을 사용해야 합니다. 3개 이상의 노드로 구성된 클러스터의 경우에는 클러스터 잠금을 사용하지 않을 수도 있습니다. 하지만 어떤 클러스터도 재편성될 수 있다는 사실을 염두에 두고 사용 여부를 결정해야 합니다. 예를 들어, 3개의 노드로 구성된 클러스터에서 유지 관리를 위해 노드 하나를 제거했다면 이 클러스터는 2개의 노드로 구성된 클러스터로 재편성됩니다. 그런데 노드 또는 통신에 장애가 발생하여 재편성이 진행되면 전체 클러스터를 사용 수 없게 됩니다.

4개 이상의 노드로 구성된 클러스터에서는 클러스터가 동일한 크기로 분할될 가능성이 매우 적기 때문에 클러스터 잠금이 필요 없을 수 있습니다. 하지만 노드의 절반에서 동시에 장애가 발생하지 않도록 클러스터를 구성해야 합니다. 예를 들어, 동일한 수의 노드 사이를 연결하는 단일 LAN 같은 단순 장애 요인을 제거해야 하며, 또한 노드의 절반이 하나의 전원 회로를 사용하지 않도록 해야 합니다.

패키지 관리자 작동 방식

클러스터에 속한 각 노드는 패키지 관리자의 사본을 실행합니다. 클러스터 코디네이터에 위치한 패키지 관리자는 **패키지 코디네이터(Package Coordinator)**라고 합니다.

패키지 코디네이터는 다음과 같은 작업을 수행합니다.

- 패키지를 실행하고, 중지하고, 이동할 시간과 장소를 결정합니다.

모든 노드에서 패키지 관리자는 다음과 같은 작업을 수행합니다.

- 사용자 정의 제어 스크립트를 실행하여 패키지 및 패키지 서비스를 실행하거나 중지합니다.
- 모니터링 중인 자원의 상태 변화에 대응합니다.

패키지 종류

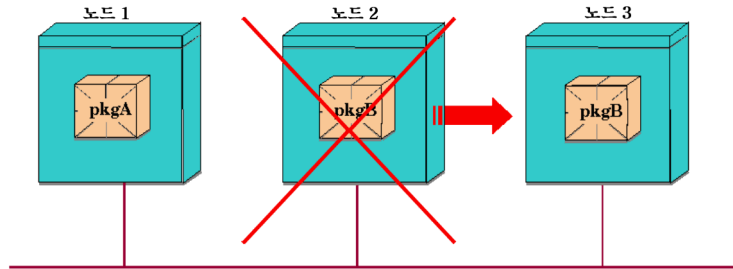
클러스터에서는 **장애 조치(Failover) 패키지**와 **시스템 다중 노드 패키지**를 실행할 수 있습니다. 시스템 다중 노드 패키지는 VERITAS 클러스터 볼륨 관리자(CVM)를 저장 관리자로 사용하는 시스템에서만 사용됩니다. 이 패키지는 `VxVM-CVM-pkg`라고도 하며, 클러스터에서 활성 상태인 모든 노드에서 실행되고 클러스터 소속 정보를 볼륨 관리자 소프트웨어에 제공합니다. 이러한 종류의 패키지는 저장소 관리에 CVM을 사용할 때만 구성되고 사용됩니다. CVM의 시스템 다중 노드 패키지를 작성하는 프로세스는 5장에 설명되어 있습니다. 이 절의 나머지 부분에서는 표준 패키지 장애 조치에 대해 설명합니다.

장애 조치 패키지

클러스터가 시작되면 노드에서 장애 조치 패키지가 시작됩니다. 패키지 **장애 조치(Failover)**는 패키지 코디네이터가 새 노드에서 패키지를 시작할 때 일어납니다. 패키지 장애 조치가 발생하면, 기존 패키지(서비스, 네트워크 또는 리소스 장애 발생 시)가 중지되고 패키지의 새 인스턴스가 시작됩니다.

다음 그림은 장애 조치를 보여줍니다.

그림 3-4 장애 조치 중 패키지 이동



패키지 구성

각 패키지는 개별적으로 구성됩니다. SAM을 사용하거나 패키지 ASCII 구성 파일(6장 참조)을 편집하여 패키지를 만듭니다. 그런 다음 `cmapplyconf` 명령을 사용하여 패키지를 검사하고 클러스터 구성 데이터베이스에 적용할 수 있습니다. 또한, 패키지 서비스 실행을 관리하는 패키지 제어 스크립트를 생성하여 패키지를 실행할 수 있습니다.

패키지를 실행하고 중지하는 시간 및 장소 결정

패키지 구성 파일은 패키지에 이름을 부여하고 패키지를 실행할 노드의 목록을 우선 순위(목록의 첫 번째 노드가 가장 높은 우선 순위)에 따라서 열거합니다. 또한, 이 파일에는 장애 조치 동작을 결정하는 3개의 매개 변수가 포함됩니다. 이 매개 변수는 `AUTO_RUN`, `FAILOVER_POLICY`, `FAILBACK_POLICY`입니다.

패키지 전환

`AUTO_RUN` 매개 변수(이전 ServiceGuard 버전에서는 `PKG_SWITCHING_ENABLED` 매개 변수) 클러스터가 시작될 때 패키지의 기본적인 전체 전환 속성을 정의합니다. 즉, 장애가 발생했을 때 새 노드에서 패키지가 자동으로 재시작되는지와 클러스터가 시작될 때 패키지도 자동으로 시작되는지를 정의합니다. 클러스터가 실행되면 각 패키지의 패키지 전환 속성은 `cmmodpkg` 명령으로 설정할 수 있습니다.

이 매개 변수는 다음과 같이 ASCII 형식의 패키지 구성 파일에 기록됩니다.

MC/ServiceGuard 소프트웨어 구성 요소 이해 패키지 관리자 작동 방식

```
# 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 패키지의 IP 주소를 새 시스템으로 이동하는 과정입니다. 패키지가 이동하는 새 시스템은 동일한 하위 네트워크로 구성되어야 하며 올바르게 작동하고 있어야 합니다. 그렇지 않으면 패키지가 시작되지 않습니다. 패키지 장애 조치가 발생하면 TCP 연결이 손실됩니다. TCP 응용 프로그램은 연결을 설정하기 위해 다시 연결되어야 하는데 이 과정은 자동으로 처리되지 않습니다. 패키지가 다수의 하위 네트워크에 의존하고 있으면 모든 하위 네트워크 역시 패키지가 이동된 노드에서 사용할 수 있어야 패키지를 시작할 수 있습니다.

그림 3-5와 그림 3-6은 변경 가능한 IP 주소의 전환을 보여줍니다. 그림 3-5는 2개의 노드로 구성된 클러스터의 최초 상태를 나타낸 것으로, 패키지 1이 노드 1에서 실행되고 패키지 2는 노드 2에서 실행되고 있습니다. 사용자는 패키지의 IP 주소를 사용하여 원하는 노드에 연결합니다. 각 노드에는 고정 IP 주소가 할당되어 있으며 각 패키지는 자체 IP 주소가 할당되어 있습니다.

그림 3-5

패키지 전환 이전

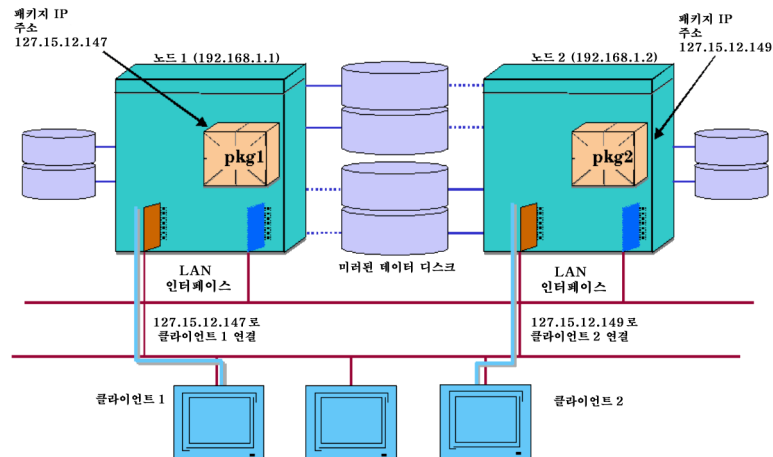
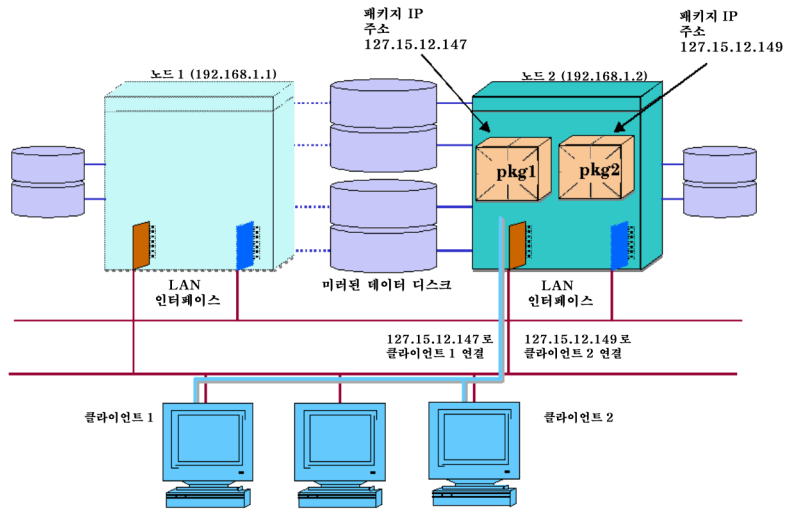


그림 3-6은 노드 1에 장애가 발생하여 패키지 1이 노드 2로 이동된 상태를 나타냅니다. 패키지 1의 IP 주소는 패키지와 함께 노드 2로 전송되었습니다. 패키지 1은 계속 사용할 수 있으며 현재 노드 2에서 실행 중입니다. 또한, 노드 2는 패키지 A의 디스크와 패키지 B의 디스크 모두에 액세스할 수 있습니다.

그림 3-6 패키지 전환 이후



장애 조치 정책

패키지 관리자는 패키지 구성 파일에 포함된 우선 순위 목록과 함께 이 파일에 포함되어 있거나 SAM을 사용하여 설정된 FAILOVER_POLICY 매개 변수에 기초하여 실행할 패키지에 대한 노드를 선택합니다. 장애 조치 정책은 특정한 노드가 확인되지 않았는데 패키지를 시작해야 할 때 패키지 관리자가 패키지를 실행할 노드를 선택하는 방법을 제어합니다. 이 정책은 장애 조치 때에만 적용되는 것이 아니라 최초 시작을 포함한 패키지의 시작에도 적용됩니다. 두 가지 장애 조치 정책은 CONFIGURED_NODE(기본값)와 MIN_PACKAGE_NODE입니다. 이 매개 변수는 다음과 같이 ASCII 형식의 패키지 구성 파일에 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 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
```

```
FAILOVER_POLICY          CONFIGURED_NODE
```

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

장애 조치 정책 값으로 MIN_PACKAGE_NODE를 사용하면 현재 실행 중인 패키지의 수가 가장 적은 노드에서 패키지가 시작됩니다. 이 때 선택되는 노드는 패키지의 로드가 가장 적은 노드가 아니라 가장 적은 수의 패키지를 실행하는 노드입니다.

자동 순환 대기 노드

MIN_PACKAGE_NODE 장애 조치 정책을 사용하면 클러스터를 구성하여 하나의 노드를 클러스터에 대한 **자동 순환 대기 노드**로 사용할 수 있습니다. 4개의 노드로 구성된 클러스터에 대해 다음과 같은 패키지 구성을 생각할 수 있습니다. 모든 패키지는 모든 노드에서 실행될 수 있으며 동일한 NODE_NAME 목록을 가질 수 있습니다. 예제에서는 각 패키지에서 순서가 다른 노드 이름을 보여주고 있지만, 반드시 필요한 것은 아닙니다.

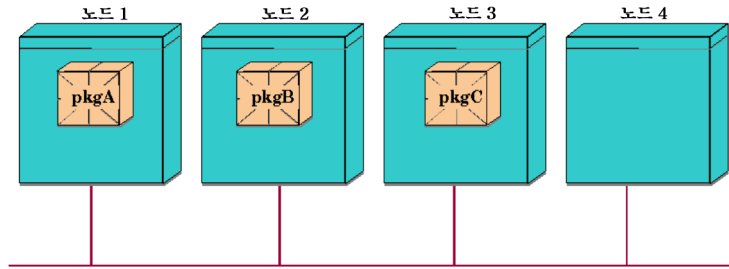
표 3-1

패키지 구성 데이터

패키지 이름	NODE_NAME 목록	FAILOVER_POLICY
pkgA	노드1, 노드2, 노드3, 노드4	MIN_PACKAGE_NODE
pkgB	노드2, 노드3, 노드4, 노드1	MIN_PACKAGE_NODE
pkgC	노드3, 노드4, 노드1, 노드2	MIN_PACKAGE_NODE

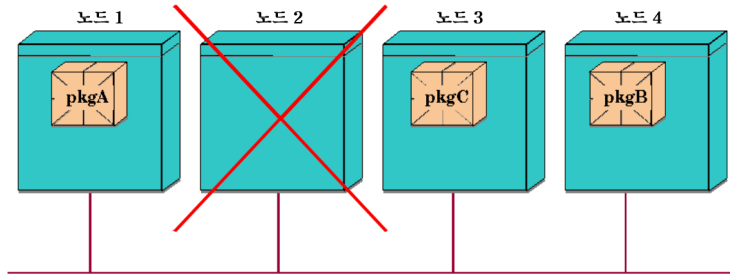
그림 3-7과 같이 클러스터가 시작되면 각 패키지가 시작됩니다.

그림 3-7 장애 조치 이전의 순환 대기 노드 구성



장애가 발생하면 패키지는 그림 3-8(노드 2에서의 장애를 나타냄) 실행 중인 패키지 수가 가장 작은 노드로 장애 조치가 수행됩니다.

그림 3-8 장애 조치 후의 순환 대기 노드 구성



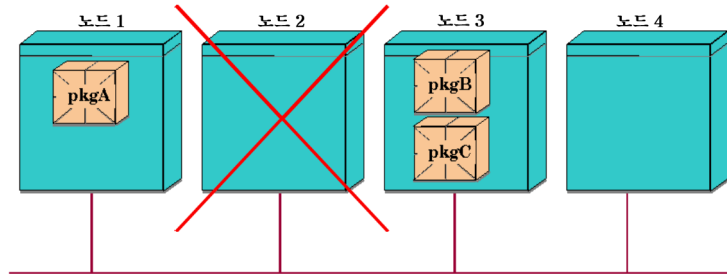
주

MIN_PACKAGE_NODE 정책을 사용하면 노드 2의 장애가 해결되어 다시 클러스터로 가져왔을 때 이 노드가 가장 적은 수의 패키지를 실행하기 때문에 새로운 대기 노드가 됩니다.

이러한 패키지가 CONFIGURED_NODE 장애 조치 정책을 사용하여 설정되면 처음에는 그림 3-7과 같이 시작되지만 노드 2에서 장애가 발생한 후에는 그림 3-9와 같이 노드 3에서 패키지가 시작될 것입니다.

그림 3-9

장애 조치 후의 CONFIGURED_NODE 정책 패키지



장애 조치 정책 값으로 CONFIGURED_NODE를 사용할 경우 이 노드가 클러스터의 구성원으로 실행 중이라고 가정하면 노드 목록에서 우선 순위가 가장 높은 노드에서 패키지가 시작됩니다. 장애 조치가 발생하면 패키지는 목록에서 다음으로 우선 순위가 높은 사용 가능한 노드로 이동합니다.

장애 복구 정책

FAILBACK_POLICY 매개 변수를 사용하면 주 노드가 다시 사용 가능한 상태가 되었을 때 패키지가 현재 주 노드에서 실행되고 있지 않으면 패키지가 다시 주 노드로 돌아갈 것인지 여부를 결정할 수 있습니다. 구성된 주 노드는 패키지의 노드 목록에서 우선 순위가 가장 높은 노드입니다.

이 정책에 사용할 수 있는 값은 AUTOMATIC과 MANUAL 두 가지입니다. 이 매개 변수는 다음과 같이 ASCII 형식의 패키지 구성 파일에 기록됩니다.

```
# Enter the failback policy for this package. This policy will be used
# to determine what action to take during failover when a 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, FAILOVER_POLICY는 CONFIGURED_NODE로 설정하고 FAILBACK_POLICY는 AUTOMATIC으로 설정한 다음과 같은 4개의 노드를 사용한 구성을 가정합니다.

그림 3-10 장애 조치 이전의 자동 장애 복구 구성

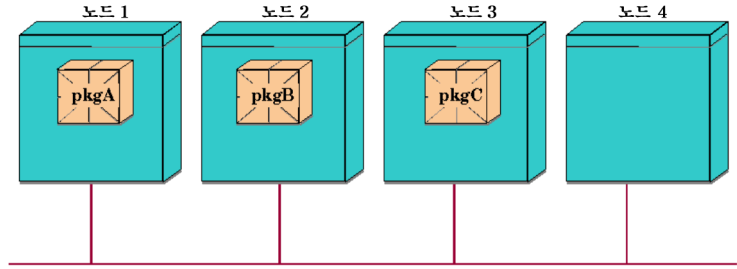
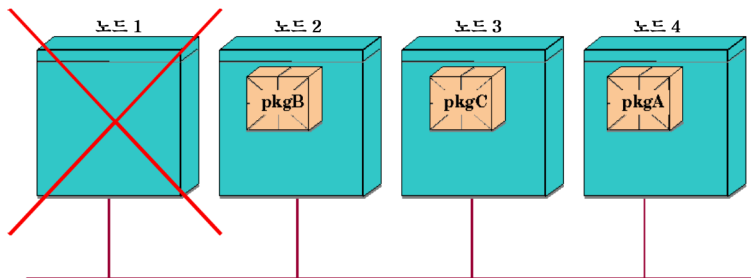


표 3-2 샘플 클러스터의 노드 목록

패키지 이름	NODE_NAME 목록	장애 조치 정책	FAILOVER POLICY
pkgA	노드1, 노드4	CONFIGURED_NODE	AUTOMATIC
pkgB	노드2, 노드4	CONFIGURED_NODE	AUTOMATIC
pkgC	노드3, 노드4	CONFIGURED_NODE	AUTOMATIC

노드 1에서 장애가 발생하여 클러스터가 재편성된 후 pkgA는 노드 4에서 실행됩니다.

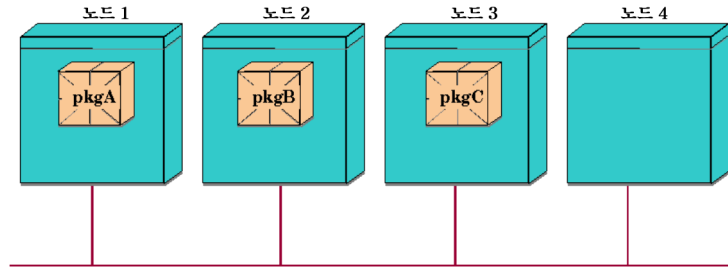
그림 3-11 장애 조치 이후의 자동 장애 복구 구성



다시 시작한 다음, 노드 1이 클러스터에 다시 참여합니다. 이 때 pkgA는 자동으로 노드 4에서 중지되고 노드 1에서 다시 시작합니다.

그림 3-12

노드 1의 재시작 후 자동 복구 구성



주

FAILBACK_POLICY를 AUTOMATIC으로 설정하면 패키지 장애 복구가 발생하여 중요한 순간에 응용 프로그램이 일시적으로 중단될 수 있습니다. 자동 장애 복구를 사용하는 경우에는 장애 조치된 패키지의 주 노드를 패키지가 주 노드로 전환되는 동안 일시적으로 사용할 수 없어도 되는 적절한 시간이 될 때까지 클러스터로 복구시키지 말아야 합니다.

장애 조치 정책과 장애 복구 정책의 결합

MIN_PACKAGE_NODE의 FAILOVER_POLICY와 AUTOMATIC의 FAILBACK_POLICY를 결합하면 장애 조치가 일어날 때마다 가장 적은 수의 패키지를 실행하는 노드가 다를 수 있으므로 패키지가 예상치 않았던 노드에서 실행될 수 있습니다.

오래된 패키지 구성 파일 사용

이전 버전의 MC/ServiceGuard에서 생성된 패키지 구성 파일을 사용한다면 FAILOVER_POLICY는 CONFIGURED_NODE의 기본 패키지 동작이 되고 FAILBACK_POLICY는 MANUAL의 기본 패키지 동작이 됩니다. 이 정책을 변경하려면 패키지 구성 파일을 편집하여 매개 변수를 수정하거나 cmmakepkg를 사용하여 새로운 패키지 구성 파일을 만듭니다.

MC/ServiceGuard 버전 A.11.12로 시작하면 PKG_SWITCHING_ENABLED 매개 변수 이름은 AUTO_RUN으로 바뀌고, NET_SWITCHING_ENABLED 매개 변수 이름은 LOCAL_LAN_FAILOVER_ALLOWED로 바뀝니다. 기존 이름은 구성 파일에서 계속 유지되지만 키워드를 변경하는 것이 좋습니다.

이벤트 모니터링 서비스 사용

기본 패키지 리소스에는 응용 프로그램의 개별 프로세스인 클러스터 노드, LAN 인터페이스 및 서비스가 포함됩니다. 모든 리소스는 **ServiceGuard**에서 직접 모니터링됩니다. 또한, 추가 모니터를 구성하는 방식으로 이벤트 모니터링 서비스(EMS) 레지스트리를 사용할 수도 있습니다. 이 레지스트리에서는 다른 소프트웨어 구성 요소가 **ServiceGuard** 리소스 모니터링을 보완합니다. 다른 소프트웨어 제품을 사용하여 구성할 수 있는 모니터로는 EMS 고가용성 모니터, OTS/9000 모니터, ATM 모니터가 있습니다.

등록된 리소스가 패키지로 구성된 경우 패키지 관리자는 리소스 등록자에게 리소스의 외부 모니터를 시작할 것을 요청합니다. 노드가 클러스터에 참여할 때 또는 패키지 시작 작업이 끝날 때 리소스가 시작되도록 구성할 수 있습니다. 그러면 모니터는 **ServiceGuard**로 메시지를 다시 보내어, 패키지가 시작되기 전에 리소스가 사용 가능한지 점검합니다. 또한, 패키지가 시작된 후에 리소스를 사용할 수 없게 되는 경우에 패키지 관리자는 패키지를 다른 노드로 보내지 않거나 다른 조치를 취할 수 있습니다.

SAM 패키지 구성 영역에 표시된 사용 가능 리소스 목록에서 리소스를 선택하여 패키지의 등록 리소스를 지정할 수 있습니다. SAM 영역에 표시된 목록의 크기는 시스템에 등록된 리소스 모니터에 따라 다릅니다. 또는, /opt/resmon/bin/resls 명령을 사용하여 시스템에 등록된 리소스에 대한 정보를 얻을 수 있습니다. 자세한 내용은 resls(1m)의 온라인 도움말을 참조하십시오.

EMS 고가용성 모니터 사용

별도 제품(B5736DA)으로 제공되는 EMS HA 모니터는 디스크 및 기타 리소스의 모니터링을 패키지 종속 관계로 설정할 때 사용됩니다. 다음은 EMS를 사용하여 모니터링할 수 있는 리소스 특성의 예입니다.

- 논리 볼륨 상태
- 실제 볼륨 상태
- 시스템 로드
- 사용자 수
- 파일 시스템 사용률
- LAN 정상 작동

패키지 관리자 작동 방식

모니터가 패키지 종속 관계로 구성되면 모니터는 리소스가 중지되었음을 나타내는 이벤트가 발생했을 때 패키지 관리자에게 알립니다. 그러면 패키지는 대체 노드로 장애 조치를 수행합니다.

또한, EMS HA 모니터는 그래픽을 표시하거나 운영자에게 알리기 위해 모니터된 이벤트를 OpenView IT/Operations와 같은 대상 응용 프로그램에 보고하는 데도 사용됩니다. 자세한 내용은 설명서 *Using High Availability Monitors(B5736-90022)*를 참조하십시오.

패키지 장애 조치 동작 선택

장애 조치 동작을 결정하기 위해 현재 실행되고 있지 않은 패키지를 자동으로 실행할 노드를 통제하는 패키지 장애 조치 정책을 정의할 수 있습니다. 또한, 주 노드를 다시 사용할 수 있게 되면 자동으로 패키지가 주 노드로 돌아가는지 여부를 결정하는 장애 복구 정책을 정의할 수 있습니다.

아래 표에서는 장애 조치 동작의 종류에 대해 설명하고, 각 동작을 결정하는 SAM 또는 ASCII 패키지 구성 파일의 설정을 설명합니다.

표 3-3

패키지 장애 조치 동작

전환 동작	SAM 옵션	ASCII 파일 매개 변수
서비스, 네트워크 또는 EMS 장애를 감지한 후에 패키지가 전환됩니다. 패키지가 전환되기 전에 중지 스크립트가 실행됩니다(기본).	<ul style="list-style-type: none"> 패키지 초기 차단을 Disabled로 설정합니다(기본). 모든 서비스에 대해 서비스 초기 차단을 Disabled로 설정합니다(기본). 패키지에 대해 자동 전환을 Enabled로 설정합니다(기본). 	<ul style="list-style-type: none"> NODE_FAIL_FAST_ENABLED를 NO로 설정합니다(기본). SERVICE_FAIL_FAST_ENABLED를 모든 서비스에 대해 NO로 설정합니다(기본). AUTO_RUN을 패키지에 대해 YES로 설정합니다(기본).
패키지는 실행 중인 패키지 수가 가장 적은 노드로 장애 조치가 수행됩니다.	<ul style="list-style-type: none"> 장애 조치 정책을 Minimum Package Node로 설정합니다. 	<ul style="list-style-type: none"> FAILOVER_POLICY를 MIN_PACKAGE_NODE로 설정합니다.
패키지는 노드 목록에서 다음 순위에 있는 노드로 장애 조치가 수행됩니다(기본).	<ul style="list-style-type: none"> 장애 복구 정책을 Configured Node로 설정합니다(기본). 	<ul style="list-style-type: none"> FAILOVER_POLICY를 CONFIGURED_NODE로 설정합니다(기본).

표 3-3 패키지 장애 조치 동작

전환 동작	SAM 옵션	ASCII 파일 매개 변수
주 노드가 사용 가능한 상태에서 패키지가 주 노드가 아닌 다른 노드에서 실행 중이면 패키지는 자동으로 중지되고 주 노드에서 다시 시작됩니다.	<ul style="list-style-type: none"> 장애 조치 정책을 Automatic으로 설정합니다. 	<ul style="list-style-type: none"> FAILBACK_POLICY를 AUTOMATIC으로 설정합니다.
패키지가 주 노드가 아닌 다른 노드에서 실행 중이면 패키지를 반드시 수동으로 주 노드로 되돌려 보내야 합니다.	<ul style="list-style-type: none"> 장애 복구 정책을 Manual로 설정합니다(기본). 장애 복구 정책을 Configured Node로 설정합니다(기본). 	<ul style="list-style-type: none"> FAILBACK_POLICY를 MANUAL로 설정합니다(기본). FAILOVER_POLICY를 CONFIGURED_NODE로 설정합니다(기본).
특정 서비스의 장애 발생 시에 노드에서 TOC(제어 전송을 의미하며, 충분한 종료 시간 없이 즉시 중지시키는 방법)가 발생한 다음 모든 패키지가 전환됩니다. TOC가 발생하기 전에 우선 시스템을 다시 시작하려고 시도합니다. 중지 스크립트는 실행되지 않습니다.	<ul style="list-style-type: none"> 특정 서비스에 대해 서비스 조기 차단을 Enabled로 설정합니다. 모든 패키지에 대해 자동 전환을 Enabled로 설정합니다. 	<ul style="list-style-type: none"> SERVICE_FAIL_FAST_ENABLED를 특정 서비스에 대해 YES로 설정합니다. AUTO_RUN을 모든 패키지에 대해 YES로 설정합니다.
서비스의 장애 발생 시 노드에서 TOC가 발생한 다음, 모든 패키지가 전환됩니다. TOC가 발생하기 전에 우선 시스템을 다시 시작하려고 시도합니다.	<ul style="list-style-type: none"> 모든 서비스에 대해 서비스 조기 차단을 Enabled로 설정합니다. 모든 패키지에 대해 자동 전환을 Enabled로 설정합니다. 	<ul style="list-style-type: none"> SERVICE_FAIL_FAST_ENABLED를 모든 서비스에 대해 YES로 설정합니다. AUTO_RUN을 모든 패키지에 대해 YES로 설정합니다.

패키지 제어 스크립트의 작동 방식

패키지는 ServiceGuard가 구성된 응용 프로그램을 시작하고 중지할 때 사용되는 수단입니다. 또한, ServiceGuard의 장애 조치 동작의 단위이기도 합니다. 패키지는 ServiceGuard가 관리하는 서비스, 디스크 볼륨 및 IP 주소의 모음입니다. 클러스터 당 최대 60개의 패키지와 총 900개의 서비스가 가능합니다.

패키지 실행 요인

현재 실행 중이지 않은 패키지가 시작되면 패키지 관리자는 클러스터의 노드에서 패키지가 사용 가능한 상태를 인식합니다. 패키지를 사용할 수 있는 노드가 여러 개 있는 경우, 패키지 관리자는 장애 조치 정책을 사용하여 패키지를 시작할 노드를 결정할 수 있습니다. 이 때 반드시 `cmrunpkg` 명령을 사용해야 하는 것은 아닙니다. 그러나 대부분의 경우 패키지를 시작하는 가장 좋은 방법은 하나 이상의 노드에서 패키지를 사용 가능하게 하는 `cmrunpkg` 명령을 실행하는 것입니다.

패키지가 노드에서 제대로 실행되지 않으면 패키지 관리자는 이 패키지를 계속 실행하려고 시도합니다. 패키지가 실행되지 않는 주요 원인으로는 특정 노드에서 패키지에 대해 `AUTO_RUN`이 사용 불능이거나 `NODE_SWITCHING`이 사용 불능으로 설정되어 있기 때문입니다. 한 노드에서는 패키지에 장애가 발생하고 다른 노드에서는 활성 상태이면 패키지는 자동으로 새 위치에서 시작됩니다. 이 프로세스를 **패키지 전환(package switching)** 또는 **원격 전환(remote switching)**이라고 합니다.

패키지를 생성할 때는 패키지를 실행할 수 있는 노드의 목록을 지정해야 합니다. 표준 패키지는 한 번에 단 하나의 노드에서 실행될 수 있으며, 노드 목록의 다음 순위인 노드에서 실행됩니다. `AUTO_RUN` 매개 변수가 `YES`로 설정되어 있으면 클러스터가 시작될 때 패키지가 자동으로 시작됩니다. 이와 반대로 `AUTO_RUN` 매개 변수가 `NO`로 설정되어 있으면 클러스터가 시작될 때 패키지가 자동으로 시작되지 않습니다. `cmmodpkg` 명령을 사용하여 이러한 종류의 패키지를 명시적으로 설정해야 합니다.

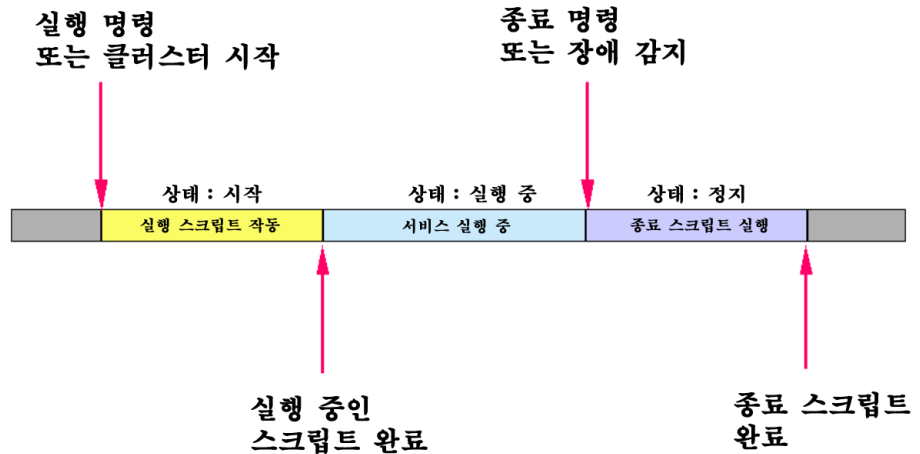
주

클러스터 실행 중에 패키지를 구성할 경우에는 `cmapplyconf` 명령이 완료되자마자 패키지가 시작되지 않습니다. 클러스터를 중지하고 다시 시작하는 과정 없이 패키지를 시작하려면 `cmrunpkg` 또는 `cmmodpkg` 명령을 사용해야 합니다.

패키지는 어떻게 시작되며, 패키지가 실행되는 동안에는 어떤 동작이 발생합니까? 그림 3-13은 패키지 수명의 여러 단계 중 일부를 보여줍니다.

그림 3-13

중요 이벤트를 표시하는 패키지 타임 라인



다음은 패키지 수명에서 가장 중요한 순간입니다.

1. 제어 스크립트 시작 전
2. 실행 스크립트 실행 중
3. 서비스 실행 중
4. 서비스, 서버넷 또는 모니터링 리소스에서 장애 발생 시
5. 중지 스크립트 실행 중
6. 명령으로 패키지나 노드 중지 시
7. 노드에 장애 발생 시

제어 스크립트 시작 전

우선, 노드를 선택합니다. 이 노드는 반드시 패키지의 노드 목록에 있어야 하며, 패키지의 장애 조치 정책을 따라야 합니다. 또한, 패키지에서 요구하는 리소스는 선택된 노드에서 사용 가능해야 합니다. 한 리소스 유형은 패키지를 위해 모니터링되는 서버넷입니다. 단, 서버넷이 사용 불가능인 경우 이 노드에서 패키지가 시작되지 않습니다. 또 다른 유형의 리소스는 모니터링된 외부 리소스의 종속 리소스입니다. 허용 범위를 벗어난 구성 리소스

스 값을 모니터링에서 표시하면 패키지가 시작되지 않습니다.

노드가 선택되면, 선택된 노드에서 패키지를 시작할 수 있는지 확인하기 위해 검사합니다. 그러면 제어 스크립트에 의해 해당 노드에서 패키지에 대한 서비스가 시작됩니다. 엄밀히 말하자면, 패키지를 시작하려면 선택한 노드에서 실행 스크립트를 실행해야 합니다.

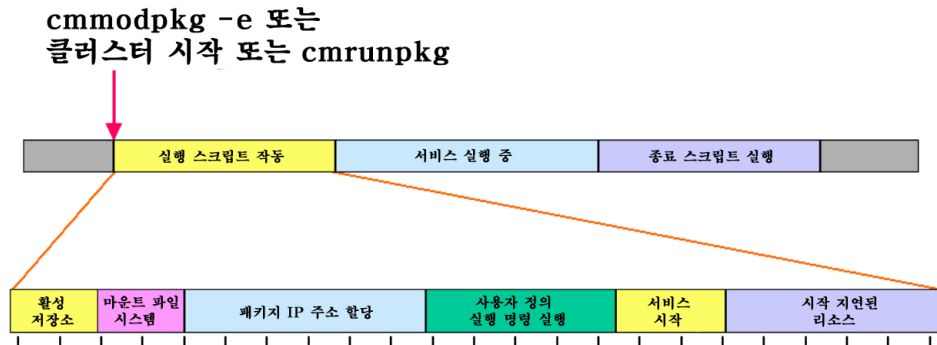
실행 스크립트 실행 중

패키지 관리자는 패키지를 시작할 수 있는 노드를 결정한 후 실행 스크립트를 실행합니다(즉, 'start'를 매개 변수로 하여 제어 스크립트가 실행됩니다.). 그림 3-14의 스크립트는 다음 단계를 수행합니다(그림 3-14 참조).

1. 볼륨 그룹 또는 디스크 그룹을 활성화합니다.
2. 파일 시스템을 마운트합니다.
3. 노드의 LAN 카드에 패키지 IP 주소를 할당합니다.
4. 사용자 정의된 실행 명령을 실행합니다.
5. 각 패키지 서비스를 시작합니다.
6. 시작 지연으로 표시된 패키지에서 요구하는 리소스를 시작합니다.
7. 종료 코드 0을 사용하여 종료합니다.

그림 3-14

실행 스크립트 실행을 위한 패키지 타임 라인



어느 단계에든 오류가 발생하면 종료 코드 1을 사용하여 비정상적으로 스크립트가 종료됩니다. 예를 들어, 패키지 서비스를 시작할 수 없으면 오류가 발생하고 제어 스크립트가 종료됩니다.

또한, `RUN_SCRIPT_TIMEOUT`에서 지정한 시간 내에 실행 스크립트 실행이 완료되지 않은 경우 패키지 관리자는 스크립트를 중지합니다. 실행 스크립트 실행 중 메시지는 실행 스크립트와 같은 디렉토리에 위치한 로그 파일에 기록됩니다. 이 로그의 이름은 실행 스크립트 및 `extension` .로그의 이름과 같습니다. 정상적인 시작은 패키지 시작과 관련된 오류 메시지나 경고와 함께 로그에 기록됩니다.

주

패키지 실행 스크립트가 작업을 완료하면 스크립트가 종료됩니다. 즉, 패키지가 정상적으로 실행되면 스크립트는 더 이상 실행되지 않습니다. 스크립트가 종료되면 스크립트에 의해 시작된 서비스의 PID를 패키지 관리자가 직접 모니터링합니다. 서비스가 중지되면 패키지 관리자는 중지 스크립트를 실행하고 `SERVICE_FAILFAST_ENABLED` 매개 변수가 `YES`로 설정되어 있으면 패키지를 실행 중인 노드를 중지합니다. 패키지 제어 스크립트에 서비스의 다시 시작 횟수가 지정되어 있는 경우, 지정된 다시 시작 횟수 범위 내에서는 패키지 제어 스크립트를 다시 실행하지 않고도 서비스를 다시 시작할 수 있습니다.

실행 스크립트의 정상 종료 및 비정상 종료

실행 스크립트를 종료하는 종료 코드는 다음 패키지에 발생할 작업을 결정하게 됩니다. 정상 종료란 패키지가 성공적으로 시작된 상태를 의미하고, 이외의 다른 종료란 시작 작업이 성공적으로 완료되지 않은 상태를 의미합니다.

- 0—정상 종료. 패키지가 정상적으로 시작되었으며, 이 노드에서 모든 서비스가 실행됩니다.
- 1—비정상 종료 또는 `NO_RESTART` 종료. 패키지가 정상적으로 시작되지 않았습니다. 서비스가 중지되며, 패키지는 다른 노드로 장애 조치를 수행할 수 없습니다.
- 2—대체 종료 또는 `RESTART` 종료. 오류가 발생했지만 다른 노드에서 패키지를 시작할 수 있습니다. 오류 발생 시 사용자가 정의한 절차를 통해 이러한 방식으로 종료한 다음 다른 노드에서 패키지를 시작할 수 있습니다. `RESTART` 종료된 패키지는 로컬 노드에서 실행할 수 없지만 다른 노드에서는 실행할 수 있습니다.
- 시간 제한—`RUN_SCRIPT_TIMEOUT`을 초과하면 종료됩니다. 이러한 상황에서는 패키지가 전체적으로 중지되고 사용할 수 없게 됩니다. 그러나, 현재 노드에서는 사용할 수 있습니다.

cmrunserv를 사용한 서비스 시작

패키지 제어 스크립트에서 cmrunserv 명령을 사용하여 개별 서비스를 시작할 수 있습니다. 이 명령은 파일에 구성된 각 서비스에 대해 한 번 실행됩니다. 각 서비스에는 이에 대한 다시 시작 횟수가 지정되어 있습니다. cmrunserv 명령은 이 횟수를 패키지 관리자에게 제공하여 서비스 장애 발생 시에 적절한 횟수 내에서 서비스를 다시 시작하도록 합니다. 다음은 일반적인 설정의 예입니다.

```
SERVICE_RESTART[0]=" " ; 다시 시작하지 않습니다.  
SERVICE_RESTART[0]="-r <n>" ; <n>번 다시 시작합니다.  
SERVICE_RESTART[0]="-R" ; 무한대로 다시 시작합니다.
```

주

<n>번 다시 시작을 설정하고 SERVICE_FAILFAST_ENABLED를 YES로 설정하면 다시 시작 시도 횟수가 <n>번 실패한 후에는 조기 차단이 발생합니다. 서비스에 대해 SERVICE_RESTART를 "-R"로 설정하고 SERVICE_FAILFAST_ENABLED를 YES로 설정하는 것은 바람직하지 **않습니다**.

서비스 실행 중

클러스터 서비스의 정상 작동 중에 패키지 관리자는 다음 사항을 지속적으로 모니터링합니다.

- 서비스의 프로세스 ID
- 패키지 구성 파일에서 모니터링하기 위해 구성된 서버넷
- 패키지과 종속 관계인 구성된 리소스

일부 장애 발생 시에는 로컬 전환이 생길 수도 있습니다. 예를 들어, 특정 LAN 카드에 장애가 발생하고 대기 LAN이 서버넷에 구성되어 있으면 네트워크 관리자는 정상적인 LAN 카드로 전환시킵니다. 서비스에 장애가 발생했지만 해당 서비스의 RESTART 매개 변수가 0보다 큰 값으로 설정되어 있으면 패키지를 중지하지 않고 지정된 다시 시작 횟수 이내에서 서비스를 다시 시작할 수 있습니다.

EMS 리소스 종속 관계가 구성되어 있고 이벤트를 발생시키는 트리거가 있으면 패키지는 중지됩니다.

모든 서비스가 실행되는 정상 작동 중에는 cmviewcl 명령의 출력 결과 중 “스크립트 매개 변수” 서비스 상태를 볼 수 있습니다.

서비스, 서브넷 또는 모니터된 리소스에서 장애 발생 시

장애가 발생하면 어떠한 일이 발생합니까? 서비스에 장애가 발생하고 더 이상 다시 시작할 수 없는 경우, 서브넷에 장애가 발생했는데 대기 서브넷이 없는 경우, 구성된 리소스에 장애가 발생하는 경우, 패키지는 현재 실행 중인 노드에서 중지되고 패키지 전환 플래그의 설정에 따라 다른 노드에서 다시 시작됩니다.

정상적으로 패키지를 중지하는 작업은 패키지 중지 스크립트를 실행하는 작업을 의미합니다(다음 부분 참조). 그러나, `SERVICE_FAILFAST_ENABLED` 매개 변수가 서비스에 대해 `YES`로 설정되어 있고 해당 서비스에 장애가 발생하면 장애가 감지되는 즉시 노드가 중지됩니다. 이 플래그가 설정되어 있지 않은 경우, 서비스가 손실되면 중지 스크립트를 실행하여 패키지를 중지하게 됩니다.

`AUTO_RUN` 매개 변수가 `YES`로 설정되어 있으면 패키지는 시작 요건을 충족하는 다른 노드에서 시작됩니다. `AUTO_RUN` 매개 변수가 `NO`로 설정되어 있으면 패키지는 다른 노드에서 시작되지 않고 중지됩니다.

명령으로 패키지나 노드 중지 시

ServiceGuard `cmhaltpkg` 명령은 패키지 중지 스크립트 실행에 영향을 주어 특정 패키지에 대해 실행 중인 서비스를 중지합니다. 이 명령은 자동 패키지 시작(AUTO_RUN)을 비활성화하여 패키지를 종료합니다.

주

`cmhaltpkg` 명령이 `-n <nodename>` 옵션과 함께 사용되는 경우, 해당 노드에서 패키지가 실행 중인 때에만 중지됩니다.

`cmmodpkg` 명령은 패키지를 중지하는 데 사용할 수 없지만, 특정 노드나 모든 노드에서 전환 설정을 해제할 수는 있습니다. 전환 설정이 해제되면 패키지를 계속 실행할 수 있지만, 현재 노드에서 실행을 중지하는 경우에 다른 노드에서 시작할 수 없습니다.

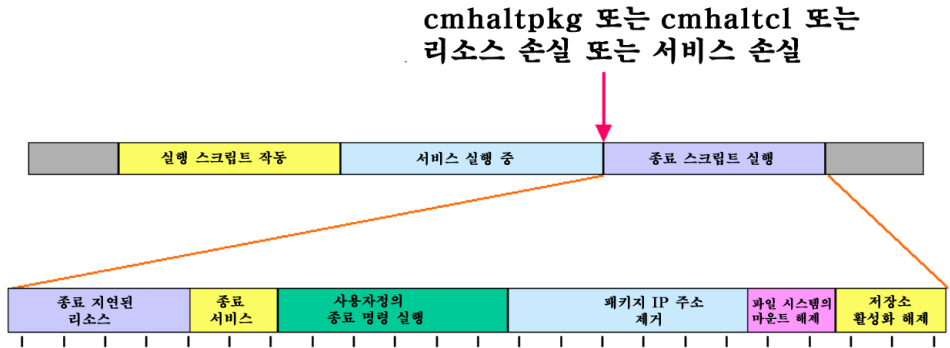
중지 스크립트 실행 중

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

1. 이전에 시작한 지연된 리소스를 중지합니다.
2. 모든 패키지 서비스를 중지합니다.
3. 사용자가 정의한 중지 명령을 실행합니다.
4. 노드의 LAN 카드에서 패키지 IP 주소를 삭제합니다.
5. 파일 시스템을 언마운트합니다.
6. 볼륨 그룹을 비활성화합니다.
7. 종료 코드 0을 사용하여 종료합니다.

그림 3-15

중지 스크립트 실행을 위한 패키지 타임 라인



어느 단계에든 오류가 발생하면 종료 코드 1을 사용하여 비정상적으로 스크립트가 종료됩니다. 또한, `HALT_SCRIPT_TIMEOUT`에 지정된 시간이 되기 전에 중지 스크립트 실행이 완료되지 않으면 패키지 관리자는 이 스크립트를 중지합니다. 중지 스크립트의 실행 중 메시지는 중지 스크립트와 같은 디렉토리에 있는 로그 파일에 기록됩니다. 이 로그의 이름은 중지 스크립트 및 `extension` 로그의 이름과 같습니다. 정상적인 시작은 패키지 중지 및 관련된 오류 메시지나 경고와 함께 로그에 기록됩니다.

중지 스크립트의 정상 종료 및 비정상 종료

패키지가 다른 노드로 이동하는 기능은 중지 스크립트에서 나가는 종료 조건에 의해 결정됩니다. 다음은 가능한 종료 코드입니다.

- 0— 정상 종료. 패키지가 정상적으로 중지되었으므로 이 노드에서 모든 서비스가 중지됩니다.
- 1— 비정상 종료 또는 `NO_RESTART` 종료. 패키지가 정상적으로 종료되지 않았습니다. 서비스가 중지되며, 패키지는 전체적으로 사용할 수 없게 됩니다. 그러나, 현재 노드에서는 사용할 수 있습니다.
- 시간 제한— `HALT_SCRIPT_TIMEOUT`을 초과하면 종료됩니다. 이러한 상황에서는 패키지가 전체적으로 중지되고 사용할 수 없게 됩니다. 그러나, 현재 노드에서는 사용할 수 있습니다.

패키지 제어 스크립트 오류 및 종료 조건

표 3-1에서는 장애 조치 패키지의 오류 조건, 조기 차단 설정 및 패키지 이동의 가능한 조합을 나타냅니다.

표 3-4 오류 조건 및 패키지 이동

패키지 오류 조건			결과			
오류 또는 종료 코드	노드 조기 차단 사용 설정	서비스 조기 차단 사용 설정	오류 이후 주 노드의 HP-UX 상태	오류 또는 종료 이후 중지 스크립트 실행	오류 이후 주 노드에서 실행할 수 있는 패키지	대체 노드에서 실행할 수 있는 패키지
서비스 장애	YES	YES	TOC	실행 안 함	N/A(TOC)	있음
서비스 장애	NO	YES	TOC	실행 안 함	N/A(TOC)	있음
서비스 장애	YES	NO	실행 중	있음	실행 안 함	있음
서비스 장애	NO	NO	실행 중	있음	실행 안 함	있음
실행 스크립트 종료 1	YES나 NO로 설정	YES나 NO로 설정	실행 중	실행 안 함	변경 없음	실행 안 함
실행 스크립트 종료 2	YES	YES나 NO로 설정	TOC	실행 안 함	N/A(TOC)	있음
실행 스크립트 종료 2	NO	YES나 NO로 설정	실행 중	실행 안 함	실행 안 함	있음
실행 스크립트 시간 제한	YES	YES나 NO로 설정	TOC	실행 안 함	N/A(TOC)	있음
실행 스크립트 시간 제한	NO	YES나 NO로 설정	실행 중	실행 안 함	변경 없음	실행 안 함
중지 스크립트 종료 1	YES	YES나 NO로 설정	실행 중	N/A	있음	실행 안 함
중지 스크립트 종료 1	NO	YES나 NO로 설정	실행 중	N/A	있음	실행 안 함

표 3-4 오류 조건 및 패키지 이동

패키지 오류 조건			결과			
오류 또는 종료 코드	노드 조기 차단 사용 설정	서비스 조기 차단 사용 설정	오류 이후 주 노드의 HP-UX 상태	오류 또는 종료 이후 중지 스크립트 실행	오류 이후 주 노드에서 실행할 수 있는 패키지	대체 노드에서 실행할 수 있는 패키지
중지 스크립트 시간 제한	YES	YES나 NO로 설정	TOC	N/A	N/A(TOC)	cmhaltpkg 명령을 실행한 후에 시간 제한이 발생하지 않는 경우, 있음
중지 스크립트 시간 제한	NO	YES나 NO로 설정	실행 중	N/A	있음	실행 안 함
서비스 장애	YES나 NO로 설정	YES	TOC	실행 안 함	N/A(TOC)	있음
서비스 장애	YES나 NO로 설정	NO	실행 중	있음	실행 안 함	있음
네트워크 손실	YES	YES나 NO로 설정	TOC	실행 안 함	N/A(TOC)	있음
네트워크 손실	NO	YES나 NO로 설정	실행 중	있음	있음	있음
모니터된 리소스 손실	YES	YES나 NO로 설정	TOC	실행 안 함	N/A(TOC)	있음
모니터된 리소스 손실	NO	YES나 NO로 설정	실행 중	있음	리소스가 지연되지 않은 경우, 있음. 리소스가 지연되는 경우, 없음	있음

네트워크 관리자의 작동 방식

네트워크 관리자의 역할은 네트워크 카드와 케이블의 장애를 찾아 복구하여 클라이언트를 위한 네트워크 서비스의 지속적인 고가용성을 유지하는 것입니다. 실제적으로는 각 패키지에 대한 IP 주소를 패키지가 실행 중인 노드의 기본 LAN 인터페이스 카드에 할당하고, 모든 인터페이스의 상태를 모니터링하고, 필요할 때 전환하는 역할을 합니다.

고정 IP 주소와 변동 가능 IP 주소

각 노드(호스트 시스템)는 각각의 활성 네트워크 인터페이스를 위한 IP 주소가 있어야 합니다. 이 주소는 **고정 IP 주소(Stationary IP Address)**라고 하며 노드의 `/etc/rc.config.d/netconf` 파일에서 구성됩니다. 고정 IP 주소는 다른 노드로 전송할 수 없지만 대기 LAN 인터페이스 카드로는 전송할 수 있습니다. 이 고정 IP 주소는 패키지에 연결되지 않습니다. 고정 IP 주소는 하트비트 메시지(“클러스터 관리자 작동 방식” 절 참조) 및 기타 데이터를 전송하는 데 사용됩니다.

고정 IP 주소 외에 일반적으로 각 패키지마다 하나 이상의 고유 IP 주소를 할당합니다. 이 패키지 IP 주소는 패키지 제어 스크립트에 있는 `cmmodnet` 명령에 의해 패키지가 시작할 때 주 LAN 인터페이스 카드에 할당됩니다. 패키지와 연결되어 있는 IP 주소는 한 클러스터 노드에서 다른 클러스터 노드로 이동할 수 있기 때문에 **변동 가능 IP 주소(Relocatable IP Address)**, **패키지 IP 주소(Package IP Address)** 또는 **유동 IP 주소(Floating IP Address)**라고 합니다. 변동 가능 IP 주소는 최대 30개의 패키지에 분산된 클러스터에서 최대 200개까지 사용할 수 있습니다.

변동 가능 IP 주소는 패키지에 할당된 가상 호스트 IP 주소와 비슷합니다. DNS(Domain Name System)를 통해서 각 패키지의 이름을 구성하는 것이 좋습니다. 그러면, 프로그램이 패키지의 이름을 호스트명처럼 `gethostbyname()`에 입력하여 패키지의 변동 가능 IP 주소를 얻을 수 있습니다.

LAN에 카드 장애가 발생하면 고정 IP 주소와 변동 가능 IP 주소 모두 대기 LAN 인터페이스로 전환됩니다. 또한, 패키지 제어가 전송되면 변동 가능 주소(고정 주소 제외)는 대체 노드로 이동될 수 있습니다. 따라서 현재 패키지가 존재하는 노드의 위치를 몰라도 응용 프로그램이 패키지의 변동 가능 주소를 통해 패키지에 액세스할 수 있습니다.

변동 가능 IP 주소 추가 및 삭제

패키지가 시작되면 지정된 IP 서브넷에 변동 가능 IP 주소를 추가할 수 있습니다. 또한, 패키지가 정지되면 지정된 서브넷에서 변동 가능 IP 주소를 삭제할 수 있습니다. 변동 가능 IP 주소의 추가 및 삭제는 **패키지 제어 스크립트**에 있는 `cmmodnet` 명령을 통해 처리됩니다. 자세한 내용은 “패키지 및 서비스 구성” 장을 참조하십시오.

IP 주소는 각각의 주 네트워크 인터페이스 카드에만 할당되며 대기 카드에는 할당되지 않습니다. 동일한 네트워크 카드의 다중 IP 주소는 반드시 같은 IP 서브넷에 속해야 합니다.

로드 분배

하나의 패키지에서 여러 서비스를 같은 IP 주소에 연결할 수 있습니다. 한 서비스가 새 시스템으로 이동하면 같은 IP 주소를 사용하는 다른 서비스도 함께 이동됩니다. 로드 분배는 각 서비스가 고유한 패키지에 포함되고 각 패키지에 고유한 IP 주소를 할당하여 구현할 수 있습니다. 이렇게 하면 관리자가 특정 서비스를 로드가 적은 시스템으로 이동시킬 수 있습니다.

LAN 인터페이스 모니터링 및 장애 감지

MC/ServiceGuard는 주기적으로 클러스터 구성 파일에 지정된 모든 네트워크 인터페이스 카드를 폴링합니다. 네트워크 장애는 각각의 단일 노드 내에서 다음과 같은 방식으로 감지됩니다. 노드의 한 인터페이스가 폴러로 지정됩니다. 이 폴러는 브리지 연결 네트워크에 있는 다른 기본 인터페이스 및 대기 인터페이스를 폴링하여 장애 유무를 검사합니다. 일반적으로, 대기 인터페이스가 폴러가 됩니다. 대기 인터페이스가 없는 브리지 연결 네트워크의 경우 기본 인터페이스가 폴러의 역할을 수행합니다. 브리지 연결 네트워크에 대해서는 2장 “네트워크 구성 요소의 중복” 절을 참조하십시오.

폴링 인터페이스는 동일한 브리지 연결 네트워크에 있는 다른 모든 인터페이스에 LAN 패킷 메시지를 보내고 응답을 받습니다. LAN 드라이버에 오류가 발생하고 인터페이스의 송수신 패킷 수가 일정 시간 동안 증가하지 않으면 이 인터페이스에 장애가 있는 것으로 간주됩니다.

로컬 전환

로컬 네트워크 전환은 로컬 네트워크 인터페이스의 장애를 감지하여 로컬 백업 LAN 카드로 장애 조치하는 과정입니다. 백업 LAN 카드에는 IP 주소가 할당되어서는 안 됩니다. 로컬 네트워크 전환이 일어날 때 이더넷에 대한 TCP/IP 연결은 중지되지 않지만 IEEE 802.3 연결은 중지됩니다. 이더넷, 토큰링, FDDI는 ARP 프로토콜을 사용하며 HP-UX는 요청되지 않은 ARP를 보내서 MAC(링크 수준) 주소와 IP 수준 주소 사이의 주소 매핑을 원격 시스템에 알립니다. IEEE 802.3에는 `rearp` 기능이 없습니다.

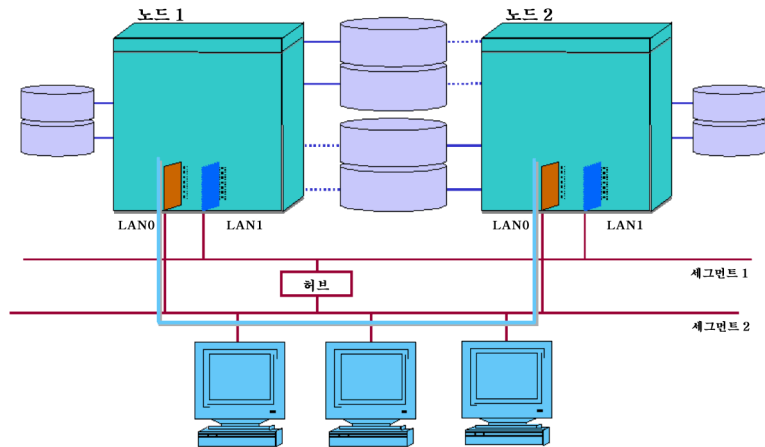
전송 중에 IP 패킷이 손실되면 TCP(전송 제어 프로토콜)는 패킷을 다시 전송합니다.

MC/ServiceGuard 소프트웨어 구성 요소 이해 네트워크 관리자의 작동 방식

UDP(사용자 데이터그램 프로토콜)인 경우에는 패킷이 프로토콜에 의해 자동으로 재전송되지 않습니다. UDP는 신뢰성이 높지 않은 서비스이기 때문에 UDP 응용 프로그램은 네트워크 패킷 손실과 그에 따른 적절한 복구에 대비해야 합니다. 로컬 전환은 종류가 같은 두 LAN 사이에서만 지원됩니다. 예를 들어, 이더넷 인터페이스와 FDDI 인터페이스 사이의 로컬 전환은 지원되지 않지만 10BT 이더넷과 100BT 이더넷 사이의 로컬 전환은 지원됩니다.

그림 3-16에서는 하나의 브리지 연결 네트워크에 두 노드가 연결되어 있습니다. LAN 세그먼트 1과 세그먼트 2는 허브로 연결되어 있습니다.

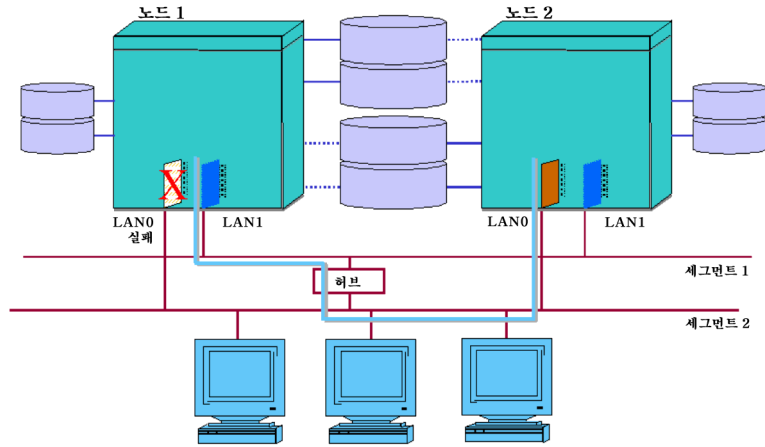
그림 3-16 로컬 네트워크 전환 이전의 클러스터



노드 1과 노드 2는 LAN 세그먼트 2에서 통신하고 있으며 LAN 세그먼트 1은 대기 상태입니다.

그림 3-17에서는 노드 1의 LAN 세그먼트 2 네트워크 인터페이스 카드에 장애가 발생하는 경우를 보여줍니다.

그림 3-17 로컬 네트워크 전환 이후의 클러스터



로컬 네트워크 전환 이후에 클러스터 장애가 발생하면 대기 인터페이스로 전환되기 때문에 대기 인터페이스의 하드웨어 경로로 IP 주소가 전환됩니다. 이 전환은 TCP/IP 수준에서 직접 수행됩니다. 모든 응용 프로그램은 원노드에서 계속 실행됩니다. 이 때, 전송으로 인해 노드 1의 IP 소통이 지연됩니다. 그러나, TCP/IP 연결은 계속 유지되며 응용 프로그램 역시 계속 실행됩니다. 노드 1의 패키지 제어는 영향을 받지 않습니다.

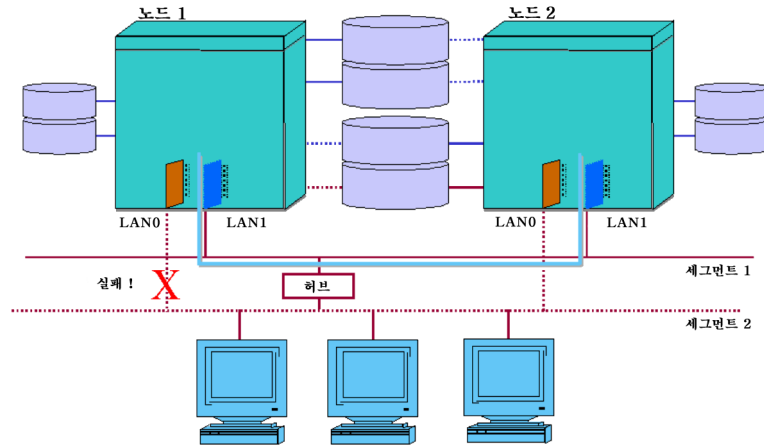
주

이더넷 네트워크에서 MC/ServiceGuard는 “이더넷 프로토콜”로 구성된 네트워크 인터페이스 또는 “IEEE 802.3 내에서 캡슐화된 SNAP”로 구성된 네트워크 인터페이스 사이의 로컬 장애 조치를 지원합니다. 한 인터페이스에 두 프로토콜을 모두 사용할 수는 없으며 서로 다른 프로토콜을 사용하는 인터페이스 사이에서는 로컬 장애 조치를 수행할 수 없습니다.

그림 3-18은 또 다른 로컬 전환의 예를 보여줍니다. 그림에서 세그먼트 2에 영향을 미치는 장애가 발생하여 두 노드가 모두 세그먼트 1에 부착된 LAN 카드로 전환되었습니다.

그림 3-18

케이블 장애 이후의 로컬 전환



로컬 네트워크 전환은 노드가 하나 이상인 클러스터에서 수행됩니다. 노드가 하나만 필요하고 더 복잡한 클러스터 구성을 필요로 하지 않는 경우, 단일 노드 클러스터를 구성하여 로컬 네트워크 전환 기능을 사용할 수 있습니다.

로컬 전환 이후에 주 LAN 인터페이스로 다시 전환

노드가 중지되면, 클러스터 데몬(cmc1d)은 대기 인터페이스에서 실행 중인 ServiceGuard에 구성된 서브넷을 주 인터페이스로 다시 전환하려고 시도합니다. 이 작업은 주 인터페이스의 연결 상태와는 상관 없이 수행됩니다. 이렇게 다시 전환하려는 목적은 클러스터가 시작되기 전의 원래 네트워크 구성을 유지하기 위해서입니다. cmhaltnode 명령이 실행되면 지정된 노드에서 다시 전환되고, cmhaltcl 명령이 실행되면 클러스터의 모든 노드에서 다시 전환됩니다.

원격 전환

원격 전환(패키지 전환이라고도 함)은 패키지과 패키지의 IP 주소를 새로운 시스템으로 이동하는 과정입니다. 패키지가 이동하는 새 시스템은 동일한 하위 네트워크로 구성되어야 하며 올바르게 작동하고 있어야 합니다. 그렇지 않으면 패키지가 시작되지 않습니다. 원격 전환에서는 TCP 연결이 손실됩니다. TCP 응용 프로그램은 연결을 설정하기 위해 다시 연결되어야 하는데 이 과정은 자동으로 처리되지 않습니다. 패키지가 다수의 하위 네트워크에 의존하고 있으면 모든 하위 네트워크 역시 패키지가 이동된 노드에서 사용할 수 있어야 패키지를 시작할 수 있습니다.

원격 전환은 동일한 종류의 LAN 사이에서만 지원됩니다. 예를 들어, 한 시스템의 이더넷과 장애 조치 시스템의 FDDI 인터페이스간에는 원격 전환이 지원되지 않습니다. 변경 가능 IP 주소의 원격 전환은 그림 3-5와 그림 3-6에 나타난 대로입니다.

전환 이후의 ARP 메시지

유동 IP 주소가 새 인터페이스로 로컬 또는 원격 전환되었을 때 IP 주소와 링크 수준 주소의 새로운 매핑을 알리는 ARP 메시지가 브로드캐스트됩니다. ARP 메시지는 이동된 각 IP 주소로 전달됩니다. 브로드캐스트를 받는 모든 시스템은 변경 사항이 반영되도록 ARP 캐시를 업데이트해야 합니다. 현재는 IP 주소가 새 시스템에 추가되는 시점에 ARP 메시지가 전달됩니다. ARP 메시지는 ARP 요청 형식으로 전달됩니다. ARP 요청 메시지의 송신자 및 수신자 프로토콜 주소 필드는 모두 동일한 유동 IP 주소로 설정됩니다. 이렇게 하면 메시지를 받은 노드가 응답 메시지를 보내지 않습니다.

자동 포트 집계

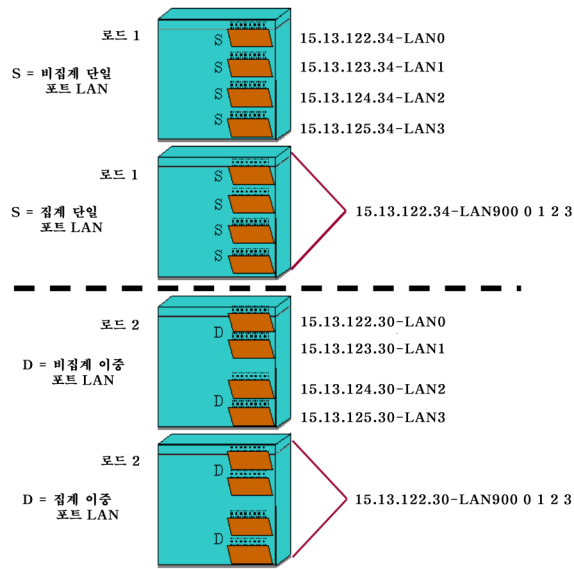
MC/ServiceGuard는 HP-APA(자동 포트 집계, HP 제품 번호 J4240AA)를 통해 자동 포트 집계의 사용을 지원합니다. HP-APA는 다중 실제 빠른 이더넷이나 다중 실제 기가 바이트 이더넷 포트를 논리 링크 집계로 집계하는 새로운 네트워킹 기술입니다. HP-APA는 다중 100 메가 바이트 빠른 이더넷 링크나 다중 1 기가 바이트 이더넷 링크(각각 200 메가 바이트 및 2 기가 바이트 전이중)를 토대로 유연하고 확장 가능한 대역폭을 허용합니다. HP-APA가 제공하는 또 다른 이점은 실제 링크 간의 로드 균형 조정, 자동 장애 감지, 고가용성을 요구하는 환경의 복구 등이 있습니다. 포트 집계 기능은 링크 집계 또는 트렁킹(trunking)이라고 합니다.

사용 가능하다면 각 링크 집계는 하나의 IP와 MAC 주소를 가지는 여러 실제 포트의 단일한 논리 링크로 볼 수 있습니다. HP-APA는 최대 4개의 실제 포트를 하나의 링크 집계로 집계할 수 있습니다. 링크 집계는 한 시스템에 50개까지 허용됩니다. 빈 링크 집계는 0이라는 MAC 주소를 갖게 됩니다.

다중 포트의 네트워킹 카드(현재 최대 4개의 포트가 사용 가능한 카드)에서 포트를 집계할 수 있습니다. 또는, 다른 카드에서 포트를 집계할 수 있습니다. 그림 3-19에서는 두 가지 예제를 보여줍니다.

그림 3-19

집계된 네트워킹 포트



19

집계되지 않은 구성의 단일 및 이중 포트된 LAN은 4개의 LAN 카드를 보유하며, 각 카드는 별도의 집계되지 않은 IP 주소와 MAC 주소를 할당 받고 자체 LAN 이름(lan0, lan1, lan2, lan3)을 가지고 있습니다. 이들 포트가 집계되면 4개의 포트는 단일한 IP 주소와 MAC 주소를 할당 받습니다. 이 예에서 집계된 포트는 lan900으로 알려져 있으며 이는 클러스터 구성 중에 어그리게이트를 지칭하는 이름입니다.

이더넷 카드 종류(단일 또는 이중 포트)와 집계 그룹을 다양하게 조합할 수 있습니다. 그러나, 하트비트 연결 시 단일 장애 요인을 피하기 위해서는 APA를 조합할 때 2개 이상의 실제 카드를 사용해야 합니다. 현재 HP-APA는 링크 집계를 자동과 수동으로 구성합니다.

데이터 저장을 위한 볼륨 관리자

볼륨 관리자는 저장 그룹이라고 하는 디스크 저장 단위를 만들 수 있는 도구입니다. 저장 그룹에는 단일 시스템에서 사용할 논리 볼륨과 고가용성 클러스터에서 사용할 논리 볼륨이 있습니다. ServiceGuard 클러스터에서 저장 그룹은 패키지 제어 스크립트에 의해 활성화됩니다.

중복 저장 종류

MC/ServiceGuard에서 지원되는 공유 데이터 저장 종류는 미러링된 **개별 디스크**(JBOD라고도 함)와 하드웨어에 중복 저장을 구성하는 외부 **디스크 어레이**가 있습니다. 미러링은 RAID1이라고 하며, 디스크 어레이에 사용할 수 있는 중복은 RAID5라고 합니다. 이 두 가지 저장 방법에는 몇 가지 차이가 있습니다.

- JBOD를 사용할 경우 기본 저장 요소는 개별 디스크입니다. 이 디스크는 다른 디스크와 쌍이 되어야 미러(RAID1)를 만들 수 있습니다. ServiceGuard 구성은 일반적으로 다른 저장 장치에 별도의 미러를 갖습니다.
- 디스크 어레이를 사용할 경우 기본 저장 요소는 LUN으로, 이 요소는 RAID5를 통해 저장소 중복성을 제공합니다.

미러링된 저장 예제

그림 3-20은 HA 저장 랙을 사용하여 미러링된 저장을 나타낸 것입니다. 예제에서 노드1과 노드2는 병렬 구성으로 케이블로 연결되어 있고 각각 두 개의 공유 저장 장치에 중복된 경로를 갖고 있습니다. 두 노드에는 각각 루트 파일 시스템, 스왑 등에 사용된 두 개의 (비 공유) 내부 디스크가 있습니다. 각각의 공유 저장 단위에는 세 개의 디스크가 있습니다. 두 저장 장치 중 하나에 있는 세 디스크의 장치 파일 이름은 c0t0d0, c0t1d0, c0t2d0입니다. 나머지 한 장치에 있는 장치 파일 이름은 c1t0d0, c1t1d0, c1t2d0입니다.

그림 3-20 공유 저장 단위 내의 실제 디스크

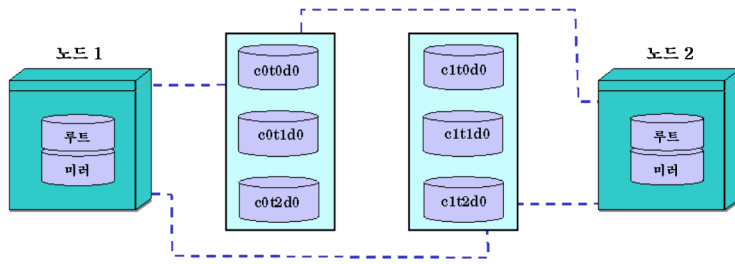


그림 3-21은 여러 디스크가 미러링된 구성에 결합된 개별 디스크를 나타낸 것입니다.

그림 3-21 미러링된 실제 디스크

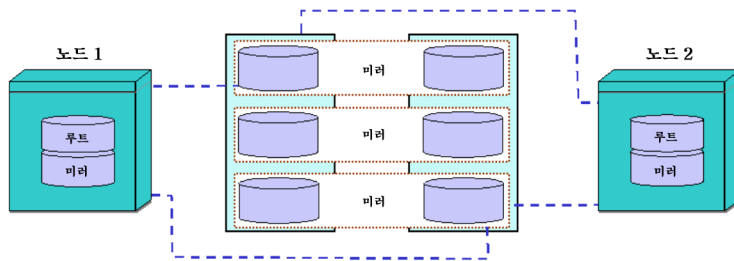
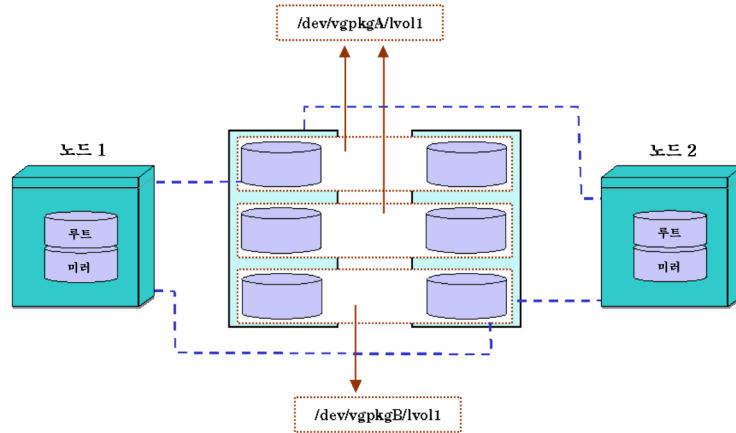


그림 3-22는 그림에 /dev/vgpkgA와 /dev/vgpkgB로 표시되어 있는 LVM 볼륨 그룹에 구성된 미러를 나타낸 것입니다. 볼륨 그룹은고가용성 응용 프로그램에 사용할 ServiceGuard 패키지에 의해 활성화됩니다.

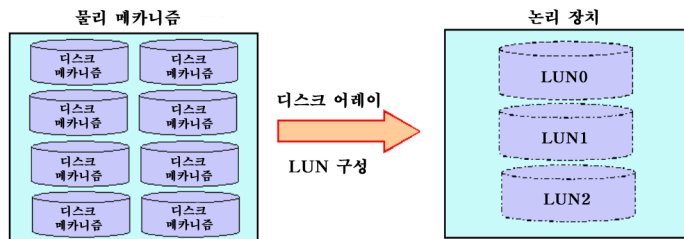
그림 3-22 볼륨 그룹에 구성된 여러 장치



디스크 어레이의 저장 장치 예제

그림 3-23은 디스크 어레이에 구성된 저장 장치를 나타낸 것입니다. 실제 디스크는 어레이 유틸리티 프로그램에 의해 논리 단위나 LUN에 구성되며 운영 체제를 통해 볼 수 있습니다.

그림 3-23 LUN에 결합된 실제 디스크

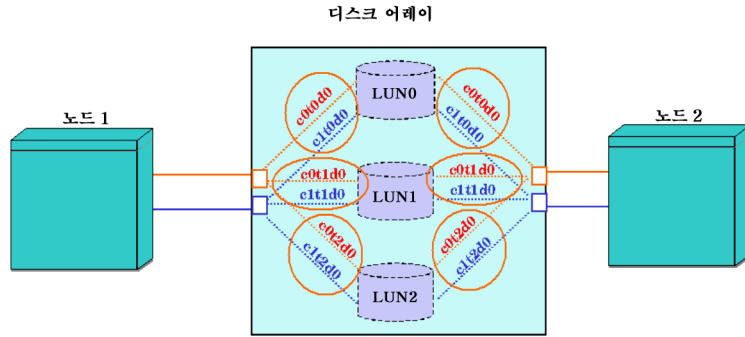


주

LUN 정의는 일반적으로 디스크 어레이 제조업체에서 제공한 유틸리티 프로그램을 사용하여 수행됩니다. 어레이는 아주 다양하므로 저장 장치 단위에 제공된 설명서를 참조하십시오.

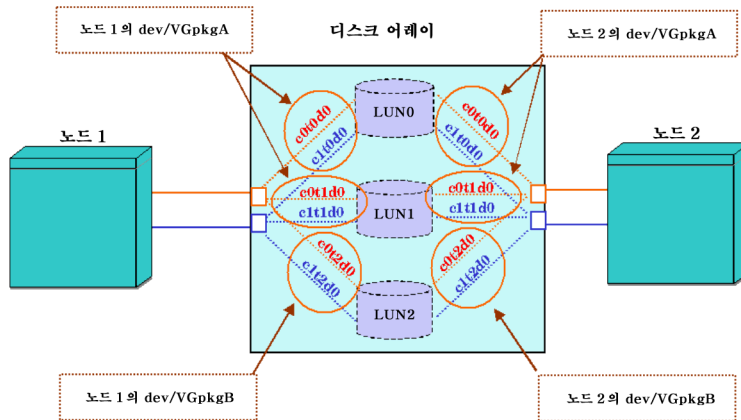
그림 3-24는 데이터에 중복 경로를 제공할 수 있도록 다중 경로(링크)로 구성된 LUN을 나타냅니다.

그림 3-24 LUN에 대한 다중 경로



마지막으로, 다중 경로는 그림 3-25에 나타난 대로 볼륨 그룹에 구성됩니다.

그림 3-25 볼륨 그룹에 구성된 다중 경로



볼륨 관리자 유형

ServiceGuard를 통해 데이터 저장을 위한 볼륨 관리자를 선택할 수 있습니다.

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

이러한 볼륨 관리자를 모두 사용하여 클러스터 저장을 구성하는 방법에 대해서는 5장과 6장에서 따로 설명합니다. 이 절의 나머지 부분에서는 이러한 볼륨 관리자의 몇 가지 차이점 및 현재 클러스터 환경에 적합한 관리자를 선택하는 방법에 대해 설명합니다.

주

HP-UX Logical Volume Manager는 **시스템 및 작업 그룹 관리**에 설명되어 있습니다. VERITAS 볼륨 관리 제품에 대한 자세한 설명은 *VERITAS Volume Manager for HP-UX Release Notes*를 참조하십시오.

HP-UX Logical Volume Manager(LVM)

논리 볼륨 관리자(LVM)는 HP-UX의 레거시 저장 관리 제품입니다. LVM은 운영 체제에 포함되어 있기 때문에 모든 클러스터 노드에서 사용할 수 있습니다. LVM은 MirrorDisk/UX 사용을 지원합니다. MirrorDisk/UX는 총 3개의 데이터 복사본에 대해 두 개의 미러까지 디스크 미러링을 허용하는 추가 제품입니다.

현재, HP-UX 루트 디스크는 LVM 볼륨 그룹으로 구성해야 합니다. HP-UX 루트 디스크는 VERITAS 루트 디스크 그룹인 rootdg와 같지 않습니다. rootdg는 VERITAS 볼륨 관리자 제품을 사용하는 노드에 HP-UX 루트 디스크를 추가로 구성해야 합니다. ServiceGuard 클러스터 잠금 디스크도 LVM 볼륨 그룹에 구성된 디스크를 사용하여 구성됩니다.

LVM은 HP-UX 단일 시스템과 ServiceGuard 클러스터에서도 지원됩니다.

VERITAS 볼륨 관리자(VxVM)

Base VERITAS Volume Manager for HP-UX(Base-VXVM)는 HP-UX 11i에 추가하여 무료로 제공되며, vmsa라고도 하는 Java 기반의 GUI 등 기본 볼륨 관리자 기능이 포함되어 있습니다. Base-VXVM만 사용하여 ServiceGuard용 클러스터 저장을 구성할 수 있으나 이렇게 하면 제한된 기능만 사용이 가능합니다.

추가 제품인 VERITAS Volume Manager for HP-UX(B9116AA)는 기본 볼륨 관리 외에 향상된 볼륨 관리자 기능을 제공합니다. 이 제품은 미러링, 활성/활성 저장 장치에 대한 동적 다중 경로 지정 및 활성 재배치와 같은 기능이 있습니다. 또 다른 추가 제품인 VERITAS FastResync Option for HP-UX(B9118AA)는 추가 비용을 부담해야 하며, 볼륨에 분할 미러를 동기화하는 데 걸리는 시간을 줄여줍니다. 이 제품을 사용하려면 VERITAS Volume Manager for HP-UX(B9116AA)가 있어야 합니다.

VxVM은 다음과 같은 클러스터에서 사용할 수 있습니다.

- 최대 16개의 노드로 구성된 모든 크기의 클러스터
- 빨리 클러스터를 시작해야 하는 클러스터
- 공유 저장 그룹을 활성화하지 않아도 되는 클러스터
- 모든 노드가 모든 디스크에 연결되지 않은 클러스터
- RAID 5 또는 스트라이프(strip) 미러링을 사용해야 하는 클러스터
- 다중 하트비트 서버넷이 구성된 클러스터

VxVM에서 디스크 그룹 배포

VxVM을 사용하면 클러스터를 실행 여부에 관계 없이 디스크 그룹을 모든 노드에 만들 수 있습니다. 그러면, 사용자는 각 노드로 이동하여 해당 디스크 그룹 가져오기를 시도하여 디스크 그룹을 검사해야 합니다. VxVM을 사용하면 CVM를 사용할 때보다 디스크 그룹 배포를 위한 단계가 추가로 필요하지만 어느 노드에서나 마음대로 디스크 그룹을 만들 수 있습니다.

VxVM을 사용한 패키지 시작 시간

VxVM을 사용하면 디스크 그룹을 사용하는 패키지 제어 스크립트가 각 디스크 그룹을 가져옵니다. 이것은 VxVM은 패키지가 시작할 때 디스크 그룹을 가져오기 때문에 클러스터 시작 시간은 영향을 받지 않지만, 개별 패키지 시작 시간은 늘어남을 의미합니다.

VERITAS 클러스터 볼륨 관리자(CVM)

볼륨 관리자(VxVM) 대신 VERITAS 클러스터 볼륨 관리자(CVM)를 사용하여 클러스터 저장을 구성할 수도 있습니다. Base-VXVM은 ServiceGuard를 설치할 때 몇 가지 기본 클러스터 기능을 제공합니다. 4개까지 노드가 지원되지만 소프트웨어 미러링, 동적 다중 경로 지정(활성/활성 저장 장치의 경우)이나 추가 라이선스를 필요로 하는 여러 가지 다른 기능은 지원되지 않습니다. VERITAS Cluster Volume Manager, CVM(B9117AA)은 클러스터 사용을 위해 특별하게 설계된 VxVM 볼륨 관리자의 향상된 버전입니다.

VERITAS Volume Manager(B9116AA)를 설치하면 CVM 추가 제품은 클러스터 환경에서 향상된 VxVM 기능을 대부분 제공합니다. CVM은 실제로 클러스터를 인식하는 볼륨 관리자로, ServiceGuard에서 직접 클러스터 구성원에 대한 정보를 얻습니다. 클러스터 정보는 VxVM-CVM-pkg라고도 하는 특수한 시스템 다중 노드 패키지를 통해 제공됩니다. 이 패키지는 클러스터의 모든 노드에서 실행됩니다. CVM과 함께 사용할 VxVM 디스크 그룹을 구성하기 위해 클러스터를 실행하고 이 패키지를 실행해야 합니다.

CVM을 통해 한 번에 한 노드에서 저장을 활성화하거나, 한 노드에서는 쓰기 활성화를 수행하고 동시에 다른 노드에서는 읽기 활성화를 수행할 수 있습니다(예: 백업 허용). CVM은 클러스터에 미러링과 동적 다중 경로 지정(DMP)을 제공합니다.

CVM은 다음과 같은 클러스터에서 사용할 수 있습니다.

- 패키지 장애 조치 이후 빠른 디스크 그룹 활성화를 필요로 하는 응용 프로그램을 실행할 클러스터
- 한 번에 둘 이상의 노드에서 활성화를 필요로 하는 클러스터. 한 노드에서는 볼륨을 사용하여 패키지를 활성화하는 동안 다른 노드에서 백업 수행하는 경우를 예로 들 수 있습니다. 이 경우, 디스크 그룹을 사용하는 패키지가 단독 쓰기 모드에서 디스크 그룹을 활성화하는 동안 백업을 수행하는 노드는 공유 읽기 모드에서 디스크 그룹을 활성화합니다.
- 단일 하트비트 서브넷으로만 구성된 클러스터

CVM은 이 릴리즈에서 4개 이하의 노드에서 지원됩니다. 노드가 공유 저장 장치에 있는 데이터에 액세스하는 여부에 관계 없이 이 장치를 클러스터 있는 모든 노드에 연결해야 합니다.

CVM을 사용한 클러스터 시작 시간

CVM을 사용하면 VxVM-CVM-pkg 제어 스크립트가 완료될 때 공유 저장 그룹(DG)을 모두 가져옵니다. DG 수, 노드 수 및 노드 구성(예: 디스크 수, 볼륨)에 따라 공유 저장 그룹을 가져오는 데 걸리는 시간이 다릅니다(이 패키지의 현재 시간 제한 값은 3분이지만 구성이 큰 경우에는 시간이 오래 걸림). CVM DG를 사용하는 모든 장애 조치 패키지는 VxVM-CVM-pkg가 실행될 때까지 시작되지 않습니다. 이러한 지연은 패키지 장애 조치 시간에 영향을 미치지 않으며, 클러스터가 시작될 때 오버헤드에 영향을 미칩니다.

CVM을 사용하여 디스크 그룹 배포

CVM을 사용하면 CVM 주 노드라고도 하는 한 클러스터 노드에서 디스크 그룹이 만들어집니다. 공유 디스크 그룹을 만들려면 클러스터를 실행해야 합니다. CVM은 각 노드에서 각 디스크를 볼 수 있는지 확인하고 잘못된 DG가 만들어지지 않도록 합니다.

CVM에 필요한 단일 하트비트 서브넷

클러스터 노드를 상호 연결하는 모든 서브넷을 하트비트 네트워크로 구성하는 것이 좋습니다. 이렇게 하면 추가 비용 없이 여러 장애에 대한 보호를 향상시킬 수 있습니다. 그러나, VERITAS 클러스터 볼륨 관리자(CVM)를 사용할 경우에는 한 개의 하트비트 서브넷만 사용할 수 있습니다. VxVM-CVM-pkg가 추가될 때(5장 참조) cmcheckconf 및 cmapplyconf가 하나의 하트비트 서브넷만 구성되는지 확인합니다. 두 개 이상의 하트비트 서브넷이 구성되면 cmcheckconf 및 cmapplyconf는 중단됩니다.

이러한 제한은 VxVM-CVM-pkg 내에서 시작하는 VERITAS 클러스터 데몬(vxclustd) 설계 때문입니다. 이 데몬은 단일 서브넷의 단일 IP 연결에 있는 클러스터 노드에만 정보를 전달할 수 있습니다. 또한, Veritas 클러스터 데몬에서 클러스터 노드 간의 통신은 ServiceGuard 클러스터 데몬 cmcld의 경우와 같습니다. 따라서, cmcld는 단일 하트비트 서브넷만 사용해야 합니다.

단일 서브넷을 사용하면 여러 가지 효과가 있습니다. 하트비트가 NODE_TIMEOUT 간격 이상으로 수신되지 않으면 클러스터가 재편성됩니다. 동작 이상이 지속되는 정도에 따라 한 개 또는 모든 클러스터 노드를 TOC에 발생시킬 수 있습니다. 이것은 하트비트 패킷이 전송되지 않도록 방지하는 서브넷에 트래픽이 많을 때 가장 많이 나타납니다. 또한, ServiceGuard나 APA에서 감지하지 못하거나 쉽게 복구할 수 없는 네트워크 하드웨어 문제 또는 네트워크 구성 문제가 있을 때에도 나타납니다. 이러한 상황에서 문제가 발생하지 않도록 하는 일반적인 방법은 다중 하트비트 네트워크를 사용하는 것입니다.

CVM의 경우 단일 하트비트 서브넷을 대기 LANS를 사용하여 구성하거나 집계된 포트 그룹으로 구성해야 시스템이나 클러스터 오류를 최소화할 수 있습니다.

볼륨 관리자 비교

아래 표는 현재 사용 가능한 볼륨 관리자의 몇 가지 장점과 단점을 요약한 것입니다.

표 3-5

ServiceGuard를 사용한 볼륨 관리자의 장점과 단점

제품	장점	단점
논리 볼륨 관리자 (LVM)	<ul style="list-style-type: none"> 레거시 시스템이 강력하며 HP-UX 사용자에게 친숙함 기존 패키지를 변경할 필요가 없음 클러스터마다 최대 16개 노드 지원 다중 데이터 경로에 PV 링크 사용 지원 여러 노드에서 읽기 전용 활성화뿐만 아니라 단독 활성화 지원 클러스터 잠금 디스크 구성에 사용 가능 다중 하트비트 서브넷 지원 	<ul style="list-style-type: none"> 유연성 및 다른 볼륨 관리자의 확장 기능 부족 PV 링크는 활성화/대기 전용으로, 한 번에 한 개의 링크만 활성화 가능
기본-VxVM	<ul style="list-style-type: none"> HP-UX 11i 이상의 제품에 무료로 제공 간단한 기본 기능 제공 다중 하트비트 서브넷 지원 	<ul style="list-style-type: none"> 제한된 기능 설정 클러스터 잠금 서비스로 사용 불가능

표 3-5

ServiceGuard를 사용한 볼륨 관리자의 장점과 단점

제품	장점	단점
VERITAS Volume Manager - B9116AA(VxVM)	<ul style="list-style-type: none"> 모든 노드에서 디스크 그룹 구성 활성/활성 저장 장치에 대한 소프트웨어 미러링, RAID 0/1, RAID 5 및 동적 다중 경로 지정을 포함하여 여러 가지 볼륨 관리 기능 향상 클러스터 시작 시간이 CVM을 사용할 때보다 빠름 클러스터마다 최대 16개 노드 지원 다중 하트비트 서브넷 지원 	<ul style="list-style-type: none"> 공유 모드나 읽기 전용 모드에서는 다중 모드에서 활성화 지원 안 함 vxdg 가져오기에 시간이 많이 걸리기 때문에 패키지를 시작할 때 지연될 수 있음 추가 라이선스를 구입해야 향상된 기능 사용 가능 클러스터 잠금 서비스로 사용 불가능
VERITAS 클러스터 볼륨 관리자 - B9117AA(CVM)	<ul style="list-style-type: none"> 소프트웨어 미러링 및 RAID 0/1을 비롯하여 볼륨 관리자 기능 향상 패키지 시작 시간이 VxVM을 사용할 때보다 빠름 단독 활성화 지원 동시에 여러 노드에서 다른 모드로 활성화 지원 	<ul style="list-style-type: none"> 디스크 그룹은 주 노드에서 구성해야 함 최대 4개까지 클러스터 노드 사용 가능 클러스터 시작 시간이 VxVM을 사용할 때보다 오래 걸림 추가 라이선스를 구입해야 함 스트라이프(stripe) 미러나 RAID 5 지원 안 함 단일 하트비트 서브넷만 지원

장애 대응

MC/ServiceGuard는 장애의 종류에 따라 각기 다른 방식으로 대응합니다. 대부분의 하드웨어 장애에 대한 대응은 사용자 설정이 불가능하지만 패키지 및 서비스 장애인 경우는 사용자가 시스템의 대응을 어느 정도 설정할 수 있습니다.

노드 장애 시 제어 전송(TOC)

MC/ServiceGuard 클러스터에서 가장 극단적인 장애 대응은 HP-UX TOC(Transfer of Control)로서 아무런 준비 없이 SPU를 즉각 정지시키는 방법입니다. 이러한 TOC는 데이터의 무결성을 보호하기 위해서 수행합니다.

클러스터 노드가 클러스터 구성원 과반수와 지정된 시간 동안 통신을 할 수 없거나, 커널 중지, 커널 스핀, 런어웨이 실시간 프로세스가 발생하거나, MC/ServiceGuard 클러스터 데몬인 `cmclld`에 장애가 발생한 경우에 TOC가 수행됩니다. TOC가 진행되는 동안 시스템 덤프가 수행되고 콘솔로 다음과 같은 메시지가 전달됩니다.

```
MC/ServiceGuard: Unable to maintain contact with cmclld daemon.  
Performing TOC to ensure data integrity.
```

또한, MC/ServiceGuard는 특정한 조건에 부합되었을 때 TOC가 시작됩니다. 패키지 구성 파일에 서비스 조기 차단 매개 변수를 사용한 경우 이 서비스에 장애가 발생할 때마다 노드 전체는 TOC와 함께 장애 상태가 됩니다. 패키지 구성 파일에서 `NODE_FAIL_FAST_ENABLED` 매개 변수가 YES로 설정된 경우 시간 제한이나 장애가 발생하여 패키지 제어 스크립트가 0이나 1 이외의 다른 값으로 종료할 때마다 노드 전체가 TOC와 함께 장애 상태가 됩니다. 또한, 노드 수준의 장애는 패키지 및 패키지 서비스와 독립적인 이벤트에 의해서도 발생할 수 있습니다. 하트비트의 손실 또는 클러스터 데몬(`cmclld`)의 손실 또는 다른 중요한 데몬의 손실로 인해 패키지 및 패키지 서비스가 정상적으로 작동하는 경우에도 노드에 장애가 발생할 수 있습니다.

때로는 TOC에 앞서 시스템의 재부팅을 시도할 수 있습니다. 안전 타이머가 종료되기 전에 재부팅이 완료될 수 있는 경우에는 TOC가 시작되지 않습니다. 어떤 경우에서도 패키지는 다른 노드로 신속하게 이동할 수 있습니다.

하드웨어 장애 대응

시스템이 중지되거나 SPU 회로가 물리적으로 손상되는 등의 심각한 문제가 발생하면 MC/ServiceGuard는 노드에 장애가 발생했음을 인식하고 현재 실행 중인 패키지를 클러스터 내의 다른 대체 노드로 전송합니다. 각 패키지의 새로운 위치는 패키지의 주 노드와 대체 노드가 설정되어 있는 패키지 구성 파일에 의해 결정됩니다. 패키지가 다른 노드로 전송될 때 프로그램 카운터는 전송되지 않습니다. 전송된 패키지의 프로세스는 처음부터 다시 시작합니다. 응용 프로그램이 장애 발생 이후 즉시 다시 시작되도록 하려면 “충돌 극복” 능력을 갖추고 있어야 합니다. 즉, 패키지에 있는 모든 프로세스가 재시작을 인식할 수 있도록 작성되어야 합니다. 심각하지 않은 시스템 충돌 이후의 재시작을 위해서도 동일한 응용 프로그램 조건이 요구됩니다.

LAN 인터페이스에 장애가 발생했을 때 대기 LAN 인터페이스가 있으면 대기 LAN 인터페이스로 코걸 전환이 수행됩니다. 하트비트 LAN 인터페이스에서 장애가 발생했을 때 대기 인터페이스나 중복 하트비트가 구성되어 있지 않으면 그 노드는 TOC와 함께 장애 상태가 됩니다. 모니터링된 데이터 LAN 인터페이스에서 장애가 발생했을 때 대기 인터페이스가 없으면 해당 패키지에 대해 `NODE_FAILFAST_ENABLED`(“패키지 구성 계획”에서 “계획” 참조)가 YES로 설정된 경우에만 노드가 TOC와 함께 장애 상태가 됩니다.

디스크 보호는 LVM의 MirrorDisk/UX나 VxVM 및 CVM의 VERITAS 미러링과 같은 별도 제품을 통해 제공됩니다. 또한, 별도로 제공되는 EMS 디스크 모니터를 이용하면 잠금 디스크 장애 같은 특정 장애가 발생했을 때 담당자가 이 사실을 통보받을 수 있습니다. 자세한 내용은 *Using High Availability Monitors* (HP 제품 번호 B5736-90025) 설명서를 참조하십시오.

클러스터 구성 요소의 전원에 장애가 발생하면 MC/ServiceGuard는 이 장애를 인식하여 적절한 전환 동작을 수행하지만 MC/ServiceGuard는 전원 장애에 직접 대응하지 않습니다. 전원은 HP PowerTrust 같은 HP의 무정전 전원 공급 장치(UPS)에 의해 보호됩니다.

패키지 및 서비스 장애 대응

일반적인 경우 패키지나 패키지 내의 서비스에서 장애가 발생하면 'stop' 매개 변수를 사용한 제어 스크립트가 실행되어 패키지를 중지한 다음 대체 노드에서 패키지를 다시 시작합니다. EMS 모니터가 구성된 리소스의 종속성 요건이 충족되지 않았음을 알리는 메시지를 패키지 관리자에게 보내면 패키지에 장애가 발생하고 대체 노드에서 다시 시작하려고 시도하게 됩니다.

필요하다면 기본 설정을 변경하여 노드가 먼저 충돌(TOC)한 경우에만 전송되도록 지정할 수 있습니다. 이렇게 지정하면 MC/ServiceGuard는 TOC에 앞서 시스템을 다시 시작하려고 시도합니다. 버퍼 캐시에 있는 버퍼를 플러시할 수 있을 만큼 시간이 충분하면 성공적으로 재부팅할 수 있으며 TOC는 발생하지 않습니다. 어떤 경우에도 시스템은 지정된 몇 초 내에 중지됩니다.

패키지가 중지되어 노드가 중지되고 알 수 없는 상태가 되는 모든 경우에 조기 차단 옵션을 사용하면 다시 시작할 때 노드가 정리된 후 신속한 장애 조치를 수행합니다. 그러나, 노드가 충돌하면 그 노드에 있는 **모든** 패키지도 즉시 중지된다는 점에 주의하십시오.

패키지 구성 중 설정한 노드 및 서비스 조기 차단 매개 변수는 장애가 발생했을 때 패키지 및 노드가 수행하는 동작을 결정합니다. 적절한 장애 조치 동작을 선택하는 방법은 “계획” 장의 “패키지 구성 매개 변수” 절을 참조하십시오.

서비스 재시작

장애 발생 후 로컬로 서비스를 다시 시작할 수 있습니다. 서비스를 다시 시작하려면 패키지 제어 스크립트에서 각 서비스의 재시작 횟수를 지정합니다. 서비스가 시작될 때 서비스 환경에 RESTART_COUNT 변수가 설정됩니다. 그러면 서비스가 실행될 때 이 변수를 검사하여 장애 이후에 서비스가 재시작되었는지 확인할 수 있고, 장애 후 재시작한 경우라면 초기화 같은 필요한 조치를 취할 수 있게 됩니다.

네트워크 통신 장애

클러스터에서 중요한 요소 중 하나는 네트워크의 정상 작동입니다. 클러스터의 이상 유무를 지속적으로 모니터링할 때 모든 노드는 다른 노드의 하트비트 메시지를 수신하여 모든 노드가 서로 통신 가능한 상태인지 확인합니다. 지정된 시간 내에 하트비트 메시지를 수신하지 못하면 노드 시간 제한에 의해 클러스터가 재편성되고, 그럼에도 불구하고 하트비트 메시지를 수신하지 못하면 TOC가 발생합니다. 2개의 노드로 구성된 클러스터의 경우 RS-232 회선을 사용하면 LAN의 정체 때문에 발생하는 일시적인 하트비트 손실로 인한 TOC를 방지할 수 있습니다. RS232 회선은 네트워크 장애를 신속하게 감지하는 데에도 도움이 됩니다.

MC/ServiceGuard 소프트웨어 구성 요소 이해
장애 대응

4 HA(고가용성) 클러스터 계획 및 문서화

MC/ServiceGuard 클러스터를 만드는 작업은 클러스터 구성의 모든 하드웨어 및 소프트웨어 구성 요소에 관한 정보를 수집하여 기록하는 계획 단계부터 시작합니다. 계획은 하드웨어와 네트워크 구성 요소의 간단한 목록을 만드는 것으로 시작합니다. 설치와 구성을 진행하면서 이 목록을 확장하고 수정하게 됩니다. 하드웨어를 설치한 다음에는 SAM 고가용성 옵션이나 다양한 HP-UX 명령을 사용하여 현재 구성에 관한 정보를 얻을 수 있습니다. 이 정보는 이 장에서 제공하는 워크시트에 기록됩니다. 클러스터를 만드는 과정에서 이 계획 워크시트를 참고하여 SAM에 값을 입력하거나 구성 파일과 제어 스크립트의 값을 편집하십시오.

구성이 완료되면 ServiceGuard Manager를 사용하여 .sgm 파일의 모든 구성 데이터를 캡처할 수 있습니다. 또한, 이후에 사용할 수 있도록 .gif 파일에 저장된 클러스터의 맵을 표시할 수 있습니다. 자세한 내용은 7장의 “ServiceGuard Manager 사용” 절을 참조하십시오.

이 장은 다음과 같은 계획 작업을 설명합니다.

- 전반적인 계획
- 하드웨어 계획
- 전원 공급 계획
- Quorum 서버 계획
- LVM 계획
- CVM 및 VxVM 계획
- 클러스터 구성 계획
- 패키지 구성 계획

이 장에서는 각 계획 단계를 설명하면서 성공적인 설치 및 유지 보수를 위해 선택적으로 매개 변수와 데이터를 기록할 수 있는 워크시트를 제공합니다. 각 단계를 진행하면서 구성의 중요한 세부 사항들을 빠짐 없이 기록하여 구성 시스템을 문서화하십시오. 그리고 실제 클러스터를 구성할 때 이 워크시트의 정보를 참조하십시오. 전체 워크시트 양식은 부록 G에 수록되어 있습니다.

주

계획과 설치의 상당 부분이 서로 겹치기 때문에 실제로 구성을 진행하기 전까지는 워크시트의 내용을 완성하기가 어렵습니다. 워크시트가 완성되지 않았다면 실제 구성을 진행하면서 빠진 항목을 기입하여 시스템을 문서화하십시오.

이후에 제공되는 장에서는 구성 및 유지 관리 작업에 대해 자세히 설명합니다.

전반적인 계획

고가용성 목표를 명확하게 설정하면 신속하게 하드웨어 요건을 정의하고 시스템을 설계하는 데 많은 도움이 됩니다. 다음 질문에 따라 전반적인 계획을 수립하십시오.

1. 장애가 발생하더라도 계속 가용성을 유지해야 하는 응용 프로그램은 무엇입니까?
2. 이 응용 프로그램을 지원하는 데 필요한 시스템 리소스(처리 성능, 네트워킹, SPU, 메모리, 디스크 공간)는 무엇입니까?
3. 정상 작동 중에 이 리소스가 클러스터 노드에 어떻게 분산됩니까?
4. 특히 노드에 고장이 발생하는 경우를 포함하여, 장애가 발생하는 모든 경우에 리소스가 클러스터 노드에 어떻게 분산됩니까?
5. 클러스터의 일상적인 유지 관리 중에 리소스가 어떻게 분산됩니까?
6. 네트워킹 요구 사항은 무엇입니까? 모든 네트워크와 서브넷이 사용 가능합니까?
7. 단순 장애 요인을 모두 제거했습니까? 예를 들면 다음과 같습니다.
 - 네트워크 단순 장애 요인
 - 디스크 단순 장애 요인
 - 전원 단순 장애 요인
 - 응용 프로그램 단순 장애 요인

MC/ServiceGuard 메모리 요구 사항

MC/ServiceGuard가 필요로 하는 잠금 가능한 메모리의 총량은 클러스터에 구성된 패키지 리소스 수에 따라 달라집니다. 다음 공식을 사용하여 필요한 메모리 양을 대략적으로 추정할 수 있습니다.

$$\text{총 메모리} = 6\text{MB} + 80\text{KB}/\text{클러스터의 패키지}$$

이 총량은 해당 패키지 또는 리소스가 노드에 있는지 여부에 상관 없이 **클러스터에 속한 모든 노드**에서 요구됩니다.

확장 계획

처음에 클러스터를 설정할 때는 일련의 노드를 지정하고 최초 구성을 위한 패키지 그룹을 정의합니다. 이후에 노드와 패키지를 추가하거나 공유 데이터 저장을 위해 디스크를 추가해야 할 수 있습니다. 클러스터를 **중지하지 않고** 클러스터를 확장하려면 주의하여 최초 구성을 계획하십시오. 다음 지침을 따르십시오.

- 이후에 패키지를 추가할 수 있도록 Maximum Configured Packages 매개 변수(다음에 나오는 “클러스터 구성 계획” 절에서 설명)를 충분히 크게 설정하십시오. 패키지 용량을 늘리면 더 많은 메모리 리소스(패키지 당 80K)가 사용된다는 점에 주의하십시오.
- 필요하면 공유 데이터 저장을 위한 디스크 하드웨어를 추가할 수 있도록 SCSI 버스 케이블 연결을 계획하십시오. 추가 장치로 사용할 SCSI 포트에는 인라인 SCSI 터미네이터 케이블을 부착해야 합니다. 이렇게 하면 버스가 활성 상태일 때 장치를 추가할 수 있습니다.
- 확장을 고려할 때는 클러스터 잠금 규칙을 반드시 따르십시오. 2개의 노드로 구성된 클러스터에는 하나의 클러스터 잠금이 필요하지만 5개 이상의 노드로 구성된 클러스터에는 클러스터 잠금을 사용할 수 **없습니다**. 즉, 클러스터가 실행 중일 때 4개 이상의 노드로 클러스터를 확장하려면 최소한 3개의 노드로 확장을 시작하고 클러스터 잠금은 구성하면 안 됩니다.
- 나중에 클러스터가 실행 중일 때 추가할 패키지에 네트워크가 필요한 경우 사전에 클러스터 구성에 포함시켜야 합니다.

- EMS가 모니터링하는 리소스가 나중에 클러스터가 실행 중일 때 추가할 패키지에서 필요한 경우 사전에 클러스터 구성에 포함시켜야 합니다. 일단 클러스터의 패키지에 의존하는 리소스가 구성되면 이후에 패키지가 시작할 때 이 리소스를 항상 사용할 수 있습니다. 그러나, 클러스터가 실행 중일 때 이전에 구성된 적이 없는 리소스를 패키지에 추가할 수는 없습니다.

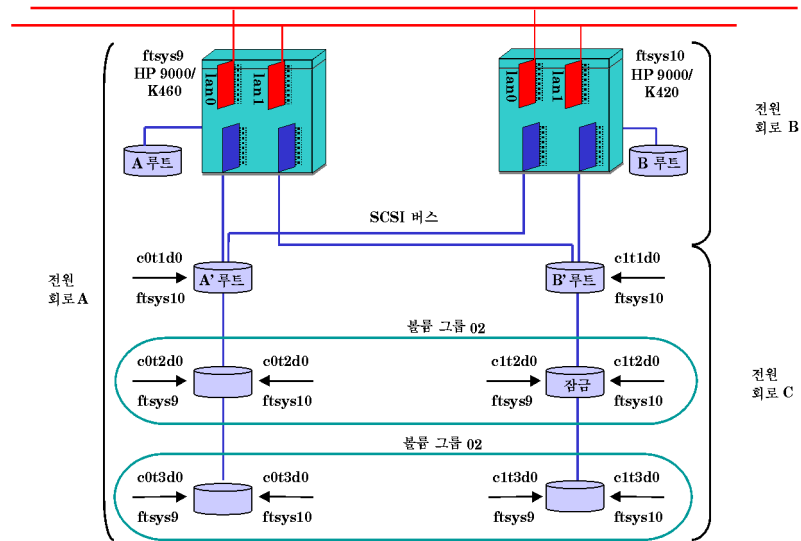
동적으로 클러스터 구성을 변경하는 방법 즉, 클러스터가 실행 중일 때 구성을 변경하는 방법에 대한 자세한 내용은 “클러스터 및 패키지 유지 관리”를 참조하십시오.

하드웨어 계획

하드웨어 계획 단계에서는 실제 하드웨어를 확인하는 작업이 필요합니다. 어댑터 카드, 버스, 케이블 연결, 디스크, 주변 장치를 포함한 하드웨어 구성을 도표로 만드는 방법도 유용합니다. 그림 4-1은 2개의 노드로 구성된 클러스터의 예제 도표입니다.

그림 4-1

클러스터 구성 예제



구성할 클러스터에 대해 이와 같은 도표를 작성하고 하드웨어 워크시트에 그 내용을 기록하십시오. 장치 어댑터를 설치할 슬롯을 지정하고 각 어댑터의 버스 주소를 지정하십시오. 그리고 클러스터 구성을 진행하면서 이 내용을 업데이트하십시오(5장 참조). **SPU 마다 하나의 양식을 사용하십시오.** 이 양식은 다음과 같은 세 부분으로 구성됩니다.

- SPU 정보
- 네트워크 정보
- 디스크 I/O 정보

SPU 정보

SPU 정보에는 클러스터에서 사용하는 S800 시스템의 기본 특성이 포함됩니다. 여러 가지 모델의 HP 9000 시리즈 800 HP-PA 컴퓨터를 동일한 클러스터에서 함께 사용할 수 있습니다. 이 컴퓨터에는 단일 프로세서 컴퓨터와 다중 프로세서 컴퓨터도 포함됩니다. HP 9000 시리즈 700 SPU는 MC/ServiceGuard에서 지원되지 않습니다.

노드 당 한 워크시트에 다음과 같은 사항을 기록하십시오.

S800 시리즈 번호

시리즈 번호를 기입하십시오(예: V2200).

호스트 이름

시스템에서 호스트 이름으로 사용할 이름을 기입하십시오.

메모리 용량

MB 단위로 메모리를 기입하십시오.

I/O 슬롯 번호

슬롯 번호를 기입하십시오.

디스크 없는 NFS 클러스터와 NetLS 서버는 지원되지 않습니다.

네트워크 정보

MC/ServiceGuard는 LAN 인터페이스뿐만 아니라 클러스터 하트비트 전송 전용으로 구성된 직렬(RS232) 회선도 모니터링합니다.

LAN 정보

한 서브넷에 요구되는 최소 LAN 인터페이스는 한 개이지만 네트워크 단순 장애 요인을 제거하기 위해서는 적어도 두 개(주 인터페이스와 하나 이상의 대기 인터페이스)의 LAN 인터페이스가 필요합니다.

클라이언트 데이터에 사용하는 것을 포함하여 모든 서브넷에서 하트비트를 구성하는 것이 좋습니다. 워크시트에서 각 LAN 인터페이스에 대해 다음 사항을 기입하십시오.

서브넷 이름

서브넷의 IP 주소 마스크를 기입하십시오. 하트비트 IP 주소는 각 노드에서 같은 서브넷에 위치해야 합니다.

인터페이스 이름

이 노드가 서브넷에 액세스할 때 사용할 LAN 카드의 이름을 기입하십시오. 이 이름은 카드를 설치한 후에 lanscan을 사용하여 볼 수 있습니다.

HA(고가용성) 클러스터 계획 및 문서화 하드웨어 계획

IP 주소

인터페이스에서 사용할 노드의 호스트 IP 주소를 기입하십시오. IP 주소는 'nnn.nnn.nnn.nnn'과 같이 점으로 구분된 숫자열입니다. 인터페이스가 대기 인터페이스이기 때문에 IP 주소가 없다면 'Standby'라고 기입하십시오.

LAN 통신량 종류

이 서브넷의 용도를 기록하십시오. 이 용도에는 다음과 같은 종류가 있습니다.

- 하트비트
- 클라이언트 통신량
- 대기

목록에 설명을 첨부하여 브리지 연결 네트워크에 속하는 서브넷을 표시하십시오.

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

RS232 정보

직렬 회선(RS232)을 구성하려면 각 노드에서 직렬 포트에 해당하는 직렬 장치 파일을 찾아야 합니다.

1. MUX 패널을 사용하면 MUX에 해당하는 시스템 슬롯 번호를 기록하고 패널에서 선택한 포트 옆에 표시된 포트 번호를 기록합니다.
2. 각 노드에서 `ioscan -fnC tty`를 사용하여 하드웨어 주소와 장치 파일 이름을 확인합니다. 예를 들면 다음과 같습니다.

```
# ioscan -fnC tty
```

이 명령은 특정 노드의 각 RS232 장치와 관련된 모든 장치 파일 표시합니다.

3. 장치 파일을 확인한 다음에는 다음과 같은 방법으로 연결을 검사합니다. 노드 1이 `/dev/tty0p0`을 사용하고 노드 2가 `/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
```

그러면 암호 파일의 내용이 노드 1의 터미널에 표시됩니다.

4. 워크시트에 다음 내용을 기록합니다.

RS232 장치 파일

각 노드의 직렬 인터페이스에 해당하는 장치 파일 이름을 기입합니다. 이 매개 변수는 ASCII 구성 파일에서 SERIAL_DEVICE_FILE입니다.

두 번째 노드 이름

RS232를 통해 이 노드에 연결할 노드 이름을 기입하십시오.

클러스터의 최대 예상 크기에 맞춰 SCSI 주소 설정

SCSI 표준은 SCSI 주소에 따라 우선 순위를 정합니다. SPU에서 컨트롤러가 정상 작동될 수 있도록 하려면 SCSI 인터페이스 카드는 최고 우선 순위로 구성되어야 합니다. 따라서, 고가용성 클러스터를 구성할 때는 노드에 가장 높은 SCSI 주소를 부여하고 디스크에는 이것보다 낮은 우선 순위의 주소를 부여해야 합니다.

Fast/Wide SCSI에서 우선 순위는 7부터 시작해서 0까지이고 그 다음으로 15부터 시작해서 8까지입니다. 따라서 7이 우선 순위가 가장 높고 8이 우선 순위가 가장 낮습니다. 예를 들어, 클러스터에 최대 4개의 노드가 존재하고 4개의 시스템 모두 일련의 디스크를 공유하려면 각 시스템에서 인터페이스 카드의 SCSI 주소는 고유하며 높은 우선 순위 주소로 설정되어야 합니다. 그러므로 시스템과 디스크의 주소 지정은 다음과 같습니다.

표 4-1 클러스터 구성에서 Fast/Wide SCSI 주소 지정

시스템 또는 디스크	호스트 인터페이스 SCSI 주소
주 시스템 A	7
주 시스템 B	6
주 시스템 C	5
주 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 부트/루트 디스크의 공유 SCSI 버스에 우선 순위가 낮은 주소를 부여하면 부트/루트 장치에 액세스할 때 시간이 초과될 경우 시스템 오류가 발생할 수 있습니다. 동일한 버스에 여러 노드와 디스크를 구성하면 이런 오류가 발생할 수 있습니다.

정확한 방법은 클러스터 노드의 인터페이스 카드에 우선 순위가 가장 높은 SCSI 주소를 부여하고 그 공유 버스에 있는 부트/루트 디스크와 공유 버스에 있는 다른 디스크에 각각 그 다음 우선 순위 주소를 부여하는 것입니다.

디스크 I/O 정보

이 부분의 워크시트에 디스크 장치 어댑터를 설치할 위치를 지정합니다. 노드의 각 디스크 장치 어댑터에 연결되는 각각의 디스크에 대해 다음 사항을 워크시트에 기록하십시오.

버스 종류

버스 종류를 기록하십시오. 지원되는 버스는 Fibre 채널인 단일 SCSI와 F/W SCSI입니다.

슬롯 번호

컴퓨터의 backplane에 삽입된 SCSI 인터페이스 카드의 슬롯 번호를 기록하십시오.

주소

버스 하드웨어의 경로 번호를 기록하십시오. 이 값은 나중에 시스템에서 ioscan을 사용하여 하드웨어를 표시했을 때 볼 수 있습니다.

디스크 장치 파일

디스크 장치 파일 이름을 기록하십시오. 장치 파일 이름을 보려면 ioscan -fnC disk 명령을 사용하십시오.

이 부분의 워크시트 정보는 논리 볼륨 관리자를 사용하여 미러링 디스크 구성을 만들 때 사용됩니다. 디스크 구성에 대하여 가능한 많은 정보를 수집하는 것이 좋습니다. 사용 가능한 디스크에 대한 정보는 다음 명령을 사용하여 얻을 수 있습니다.

- diskinfo
- ioscan -fnC disk
- lssf /dev/*dsk/c*
- bdf
- mount
- swapinfo
- vgdisplay -v
- lvdisplay -v
- lvlnboot -v
- vxvg list(VxVM 및 CVM)
- vxprint(VxVM 및 CVM)

HA(고가용성) 클러스터 계획 및 문서화 하드웨어 계획

이 명령은 표준 **HP-UX** 명령입니다. 특수한 사용 방법에 대한 자세한 내용은 온라인 도움말을 참조하십시오. 이 명령은 하드웨어를 설치하고 시스템을 다시 시작한 다음 **모든 노드**에서 실행해야 합니다. 수집된 정보는 저장 그룹과 클러스터 구성을 수행할 때 유용하게 사용됩니다. `lsssf` 명령의 출력 결과를 인쇄한 목록에서 디스크에 할당해야 하는 실제 볼륨 그룹을 확인할 수 있습니다.

하드웨어 구성 워크시트

다음 워크시트는 클러스터 하드웨어 구성을 기록하고 정리하는 데 도움을 줍니다. 필요한 만큼 복사하십시오. 워크시트를 완성하고 나중에 참고할 수 있도록 잘 보관하십시오.

그림 4-2

하드웨어 구성 워크시트

```

SPU Information:

S800 Host Name ___ftsys9_____ S800 Series No ___892_____

Memory Capacity ___128 MB_____ Number of I/O Slots ___12_____
=====
LAN Information:

Name of      Name of      Node IP      Traffic
Subnet ___Blue___ Interface ___lan0___ Addr___35.12.16.10___ Type ___HB___

Name of      Name of      Node IP      Traffic
Subnet ___Blue___ Interface ___lan2___ Addr_____ Type ___standby___

Name of      Name of      Node IP      Traffic
Subnet ___Red___ Interface ___lan1___ Addr___35.12.15.12___ Type ___HB, client___
=====
Serial (RS232) Heartbeat Interface Information:

RS232 Device File ___/dev/tty0p0___

Second Node Name ___ftsys10_____

=====
X.25 Information

OTS subnet _____ OTS subnet _____
=====
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 회선 전원과 무정전 전원(UPS)을 고려해야 합니다. 전원 회로의 손상으로 클러스터가 중지되면 안됩니다. 또한, 노드의 절반 이상이 하나의 전원을 사용하면 안됩니다. 한 전원이 정확하게 노드의 절반에 전원을 공급하는 경우 클러스터 잠금 디스크에 이 전원을 사용하면 안됩니다. 그렇지 않으면 장애가 발생했을 때 클러스터가 재편성될 수 없습니다. 자세한 내용은 “클러스터 구성 및 계획”에서 “클러스터 잠금”에 대한 절을 참조하십시오.

전원 장애 시에도 높은 가용성을 유지하려면 각 노드의 SPU와 클러스터 잠금 디스크에 별도의 UPS를 사용하고, 서로 다른 실제 볼륨 그룹의 구성원으로 구성된 디스크는 서로 다른 전원 공급 장치에 연결되도록 구성해야 합니다. 이 규칙은 서로 다른 전원 공급 장치에 연결되어 서로 다른 I/O 버스를 사용하는 실제 디스크 사이에서 디스크 미러링을 사용하기 위한 것입니다.

혼동되지 않도록 각각의 하드웨어 장치와 전원 공급 장치에 서로 다른 장치 번호의 레이블을 붙이는 것이 좋습니다. 전원 공급 워크시트에 사용하는 특정 하드웨어 장치와 이 장치를 연결할 전원 공급 장치에 대해 기록하십시오. 워크시트에 다음과 같은 레이블 정보를 기록하십시오.

S800 호스트 이름	각 SPU에 대한 호스트 이름을 기록하십시오.
디스크 장치	각 디스크의 디스크 장치 번호를 기록하십시오.
테이프 장치	각 백업 장치의 테이프 장치 번호를 기록하십시오.
기타 장치	기타 장치의 번호를 기록하십시오.
전원 공급 장치	호스트 및 기타 장치가 연결되는 UPS의 전원 공급 장치 번호를 기록하십시오.

UPS와 캐비닛의 전력 한계 및 SPU의 전력 한계를 초과하지 않도록 주의하십시오.

주

891/892/893 (T 클래스) 시스템의 배터리 백업 기능은 MC/ServiceGuard에서 지원되지 않습니다.

전원 공급 구성 워크시트

다음 워크시트는 전원 공급 장치 구성을 기록하고 정리하는 데 도움을 줍니다. 필요한 만큼 복사하십시오. 워크시트를 완성하고 나중에 참고할 수 있도록 잘 보관하십시오.

그림 4-3

전원 공급 구성 워크시트

```

=====
SPU Power:

S800 Host Name ___ftsys9_____ Power Supply ___1_____
S800 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:

Unit Name _____ Power Supply _____
Unit Name _____ Power Supply _____
    
```

Quorum 서버 계획

quorum 서버(QS)는 클러스터의 서비스를 재편성합니다. QS에 대해서는 3장의 “클러스터 잠금으로 사용하는 quorum 서버” 절을 참조하십시오.

quorum 서버는 다음과 같습니다.

- 둘 이상의 클러스터에서 사용할 수 있습니다.
- 지원되는 모든 노드와 함께 클러스터를 지원할 수 있습니다.

주

quorum 서버가 실행 중인 노드는 서비스를 제공하는 클러스터와 같은 서브넷에 있어야 합니다. 이렇게 하면 네트워크 작업이 지연되더라도 quorum 서버 작업에 영향을 미치지 않습니다. 다른 서브넷을 사용하면 네트워크 작업이 지연되어 quorum 서버 시간 제한이 발생할 수 있습니다. 이러한 시간 제한이 발생하지 않도록 하려면 클러스터 ASCII 파일에 QS_TIMEOUT_EXTENSION 매개 변수를 사용하여 quorum 서버 시간 제한 간격을 늘리십시오.

서버 접속에 클러스터 하트비트 네트워크가 사용된 경우에는 적어도 한 개의 다른 네트워크가 하트비트 네트워크여야 두 네트워크에 동시에 장애가 발생하지 않습니다.

Quorum 서버 워크시트를 사용하여 한 개 또는 여러 개의 클러스터에 사용할 quorum 서버를 구분할 수 있습니다. 클러스터 구성 워크시트에 quorum 서버 호스트와 타이밍 매개 변수를 입력해야 합니다.

QS 워크시트에 다음을 입력합니다.

Quorum 서버 호스트

quorum 서버의 호스트 이름을 입력합니다.

IP 주소

quorum 서버에서 액세스할 IP 주소를 입력합니다.

지원되는 노드 이름

이 quorum 서버에서 지원할 모든 클러스터 노드 이름을 입력합니다. 이러한 항목은 quorum 서버 프로세스를 실행하는 시스템의 qs_authfile에 입력됩니다.

Quorum 서버 워크시트

다음 워크시트는 quorum 서버 하드웨어 구성을 기록하고 정리하는 데 도움을 줍니다. 필요한 만큼 복사하십시오. 워크시트를 완성하고 나중에 참고할 수 있도록 잘 보관하십시오.

그림 4-4

Quorum 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 계획

HP-UX Logical Volume Manager(LVM)를 사용하거나 VERITAS VxVM 및 CVM 소프트웨어를 사용하여 저장 그룹을 만들 수 있습니다. 저장 그룹에 대해서는 다음 절에서 설명합니다.

LVM을 사용하여 디스크 어레이를 설계할 때는 다음과 같은 사항을 고려해야 합니다.

- 루트 디스크는 루트 디스크의 볼륨 그룹에 속해야 합니다.
- 고가용성 응용 프로그램, 서비스 또는 데이터를 포함하는 볼륨 그룹은 주 노드 및 모든 대체 노드에서 액세스할 수 있는 버스에 있어야 합니다.
- 고가용성 응용 프로그램, 서비스 및 데이터는 가용성이 높지 않은 응용 프로그램, 서비스 및 데이터와 분리된 별도 볼륨 그룹에 있어야 합니다.
- 제어를 위해 함께 전송해야 하는 고가용성 응용 프로그램, 서비스 및 데이터들은 그룹으로 묶어 하나의 단일 볼륨 그룹 또는 연속된 볼륨 그룹에 두어야 합니다.
- 제어를 위해 별도로 전송해야 하는 두 개의 다른 고가용성 응용 프로그램, 서비스, 데이터를 하나의 볼륨 그룹에 두지 않아야 합니다.
- 루트 디스크는 다른 노드에서 활성화될 수 있는 볼륨 그룹에 속하지 않아야 합니다.
- 볼륨 그룹 이름은 기본 볼륨 그룹 이름(예: `vg01`, `vg02`)이 아닌 다른 이름을 사용하는 것이 좋습니다. 볼륨 그룹 이름은 그 볼륨 그룹과 관련된 고가용성 응용 프로그램을 의미하는 이름을 선택하십시오. 예를 들어, `/dev/vgdatabase`와 같은 이름을 사용하면 클러스터를 관리하기가 더 쉬워질 것입니다.

EMS HA Disk Monitor를 사용할 계획이라면 *Using High Availability Monitors*(B5736-90025) 설명서에서 “Rules for Using EMS Disk Monitor with MC/ServiceGuard”를 참조하십시오.

LVM 워크시트

다음 워크시트는 디스크 구성을 기록하고 정리하는 데 도움을 줍니다. 필요한 만큼 복사하십시오. 워크시트를 완성하고 나중에 참고할 수 있도록 잘 보관하십시오. 이 워크시트는 볼륨 그룹과 실제 볼륨 그룹만 포함합니다. 패키지 구성 워크시트(이 장의 뒷부분에서 설명)에는 각 볼륨 그룹을 구성하는 논리 볼륨 및 파일 시스템에 관한 자세한 정보를 기록할 수 있는 공간이 있습니다.

그림 4-5

볼륨 그룹 및 실제 볼륨 워크시트

```
=====
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

HP-UX Logical Volume Manager(LVM, 이전 절에서 설명)을 사용하거나 VERITAS VxVM 및 CVM 소프트웨어를 사용하여 저장 그룹을 만들 수 있습니다.

CVM 또는 VxVM 디스크 그룹을 사용하여 저장 그룹을 구성할 때는 다음 사항을 고려하십시오.

- VxVM 저장을 사용할 각 클러스터 노드에 *rootdg* 디스크 그룹을 만들어야 합니다. 이 디스크 그룹은 HP-UX 루트 디스크와 다르기 때문에 HP-UX 루트 디스크로 사용할 수 없습니다. VxVM 루트 디스크 그룹을 만든 노드에만 이 그룹을 가져올 수 있습니다. 이 디스크 그룹은 각 클러스터 노드에 **한 번만** 만들어집니다.
- CVM 디스크 그룹은 클러스터가 구성된 후에 만들어지지만 VxVM 디스크 그룹은 클러스터가 구성되기 전에 만들어집니다.
- 고가용성 응용 프로그램, 서비스 및 데이터는 가용성이 높지 않은 응용 프로그램, 서비스 및 데이터와 분리된 별도 디스크 그룹에 있어야 합니다.
- 제어를 위해 별도로 전송해야 하는 두 개의 다른 고가용성 응용 프로그램, 서비스 또는 데이터를 하나의 디스크 그룹에 두지 않아야 합니다.
- 사용하는 HP-UX 루트 디스크는 클러스터 노드 간에 공유된 LVM 볼륨 그룹에 속하지 **않아야** 합니다.
- 클러스터 잠금 디스크는 LVM 볼륨 그룹으로만 구성할 수 있습니다.
- VxVM 디스크 그룹 이름은 클러스터 구성 ASCII 파일에 입력하지 않아야 합니다. 이 이름은 `cmqueryc1` 명령으로 클러스터 구성 ASCII 파일에 삽입되지 않습니다.

CVM 및 VxVM 워크시트

다음 워크시트는 디스크 구성을 기록하고 정리하는 데 도움을 줍니다. 필요한 만큼 복사하십시오. 워크시트를 완성하고 나중에 참고할 수 있도록 잘 보관하십시오. 이 워크시트는 볼륨 그룹과 실제 볼륨 그룹만 포함합니다. 패키지 구성 워크시트(이 장의 뒷부분에서 설명)에는 각 볼륨 그룹을 구성하는 논리 볼륨 및 파일 시스템에 관한 자세한 정보를 기록할 수 있는 공간이 있습니다.

그림 4-6

VxVM 디스크 그룹 및 디스크 워크시트

```
=====
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 장애로부터 최대한 빠르게 복구될 수 있도록 설계해야 합니다. 장애로부터 복구하는 데 소요되는 실제 시간은 여러 가지 요인에 따라 다릅니다.

- 클러스터 하트비트 주기와 노드 시간 제한의 길이. 이 값들은 가능한 짧게 설정해야 하지만 각각 1000000(1초)와 2000000(2초)보다 짧으면 안됩니다. 하트비트 주기 값은 1000000(1초)으로, 노드 시간 제한 값은 5 내지 8초 범위(5000000 내지 8000000) 내로 설정하는 것이 좋습니다.
- 패키지 제어 스크립트의 실행 및 중지 명령 설계. 이들 명령이 신속하게 실행될 수 있도록 작성해야 합니다.
- 원시 디스크의 액세스 가능 여부. 원시 디스크에 액세스하는 응용 프로그램은 충돌 복구 서비스를 사용하여 설계해야 합니다.
- 응용 프로그램과 데이터베이스 복구 시간. 이 시간은 최소화되도록 설계해야 합니다.

또한, 다음과 같이 클러스터 전체에 걸쳐 일관성을 유지해야 합니다.

- 모든 노드에서 사용자 이름이 동일해야 합니다.
- 모든 노드에서 UID가 동일해야 합니다.
- 모든 노드에서 GID가 동일해야 합니다.
- 모든 노드에서 시스템 영역의 응용 프로그램이 동일해야 합니다.
- 클러스터 전체에 걸쳐서 시스템 시간이 동일해야 합니다.
- /usr 파일과 같이 하나 이상의 노드에서 사용할 수 있는 파일은 모든 노드에서 동일해야 합니다.

하트비트 서브넷과 재편성 시간

사용하는 하트비트 네트워크의 종류도 클러스터 재편성 시간에 부분적으로 영향을 미칩니다. 이더넷은 다른 종류보다 장애 조치 속도가 느립니다. 하트비트 서브넷을 둘 이상 사용 중인 경우에는 장애 조치 시간이 가장 빠른 서브넷이 사용됩니다.

클러스터 잠금 정보

클러스터 잠금은 이전에 클러스터를 구성했던 노드의 정확히 절반에 해당하는 노드가 새로운 클러스터를 구성하려고 할 때 단 하나의 새 클러스터만 구성되도록 하는 역할을 합니다. 새로운 클러스터가 오직 하나만 구성되고, 패키지에서 지정된 디스크에 유일하게 액세스할 수 있도록 하는 것은 매우 중요합니다. 잠금 디스크나 quorum 서버를 클러스터 잠금으로 지정할 수 있습니다. 2개의 노드로 구성된 클러스터는 클러스터 잠금을 사용해야 하지만 클러스터가 클 경우에는 잠금을 사용하는 것이 좋습니다. 5개 이상의 노드로 구성된 클러스터는 클러스터 잠금으로 quorum 서버만 사용할 수 있습니다.

클러스터 잠금 디스크와 재편성 시간

디스크 잠금을 사용할 경우 클러스터 잠금을 얻는 데 걸리는 시간은 사용하는 디스크 I/O 인터페이스에 따라 달라집니다. 디스크 하드웨어가 모두 구성된 다음에, 그러나 클러스터가 구성되기 전에 클러스터의 모든 노드에서 `cmquerycl` 명령을 실행하여 사용 가능한 디스크 목록과 각 디스크와 관련된 재편성 시간을 표시할 수 있습니다. 예를 들면 다음과 같습니다.

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

또는, SAM을 사용하여 클러스터 잠금의 실제 볼륨 목록과 재편성 시간을 볼 수 있습니다.

기본적으로 MC/ServiceGuard는 재편성 시간이 가장 빠른 디스크를 선택합니다. 하지만, 전원을 고려하여 다른 디스크를 선택해야 할 수도 있습니다. 클러스터 잠금 디스크에는 가능하면 별도의 전원을 공급해야 한다는 점에 주의하십시오.

클러스터 잠금 디스크 및 확장 계획

클러스터가 시작되어 실행 중일 때 클러스터 노드를 추가할 수 있습니다. 그러나, 클러스터를 중지하지 않고 노드를 추가하려면 다음과 같은 규칙을 따라야 합니다. 4개 이상의 노드로 구성된 클러스터는 잠금 디스크가 없습니다. 그러므로, 총 4개 이상의 노드를 가져올 수 있을 정도의 충분한 노드를 추가할 계획이라면 quorum 서버를 사용해야 합니다.

클러스터 구성 매개 변수

클러스터 관리자의 작동과 관련된 일련의 클러스터 매개 변수를 정의해야 합니다. 이 매개 변수는 클러스터의 모든 노드에 위치한 이진 클러스터 구성 파일에 저장됩니다. 이 매개 변수는 `cmquerycl` 명령을 실행했을 때 생성되는 클러스터 구성 템플릿 파일을 편집하여 입력할 수 있습니다(“HA 클러스터 구성 만들기” 참조). 아래에 제공된 매개 변수 이름은 클러스터 ASCII 구성 파일에서 사용하는 이름입니다.

HA(고가용성) 클러스터 계획 및 문서화 Quorum 서버 계획

다음 매개 변수는 반드시 정의해야 합니다.

CLUSTER_NAME

cmviewcl 같은 명령의 출력 결과에 나타나고 클러스터 구성 파일에 표시되는 이름입니다.

클러스터 이름에는 공백, 슬래시(/), 백슬래시(\) 및 별표(*) 같은 문자를 사용할 수 없습니다. 이외의 다른 문자는 모두 사용할 수 있습니다. 클러스터 이름은 최대 40자까지 지정할 수 있습니다.

QS_HOST

quorum 서버 기능을 제공하는 현재 클러스터 외부에 있는 호스트 시스템의 이름이거나 IP 주소입니다. 이 매개 변수는 클러스터에서 서비스를 재편성하기 위해 quorum 서버를 사용할 때만 사용됩니다.

QS_POLLING_INTERVAL

quorum 서버가 실행되고 있는지 확인하기 위해 연락을 시도하는 시간 간격(밀리초)입니다. 기본값은 300,000,000밀리초(5분)입니다.

QS_TIMEOUT_EXTENSION

quorum 서버 시간 제한은 quorum 서버가 클러스터와 통신하지 않는 시간입니다. 이 시간이 지나면 클러스터는 quorum 서버를 DOWN으로 표시합니다. 이 시간은 ServiceGuard 매개 변수를 기준으로 계산되지만 밀리초를 추가하여 늘릴 수도 있습니다.

QS_TIMEOUT_EXTENSION은 선택적 매개 변수입니다.

FIRST_CLUSTER_LOCK_VG, SECOND_CLUSTER_LOCK_VG

클러스터 잠금이 기록되어 있는 실제 디스크 볼륨을 포함하는 볼륨 그룹입니다. 2개의 노드로 구성된 클러스터에서는 클러스터 잠금 볼륨 그룹을 식별하는 작업이 중요합니다. 두 개의 클러스터 잠금을 만들려면 각 잠금에 대한 볼륨 그룹 이름을 모두 입력해야 합니다. 이 매개 변수는 클러스터에서 서비스를 재편성하기 위해 잠금 디스크를 사용할 때 사용됩니다.

첫 번째 잠금 볼륨 그룹의 `FIRST_CLUSTER_LOCK_VG`를 사용합니다. 두 번째 잠금 볼륨 그룹이 있다면 `SECOND_CLUSTER_LOCK_VG` 매개 변수가 별도 행으로 파일에 포함됩니다.

주

잠금 볼륨 그룹은 클러스터 ASCII 구성 파일의 `VOLUME_GROUP` 매개 변수에도 정의되어 있어야 합니다.

NODE_NAME 클러스터의 노드가 될 각 시스템의 호스트 이름입니다. 노드 이름은 최대 40자까지 지정할 수 있습니다.

NETWORK_INTERFACE

하트비트나 사용자 데이터에 사용되는 각 LAN의 이름입니다. 그 예로 `lan0`을 들 수 있습니다.

HEARTBEAT_IP

클러스터 하트비트를 전송할 서브넷을 의미하는 IP 정보입니다. 하트비트 IP 주소는 각 노드에서 같은 서브넷에 위치해야 합니다.

VERITAS CVM 디스크 그룹을 저장에 사용할 경우에는 **단일** 하트비트 서브넷만 사용할 수 있습니다. 이 경우 하트비트는 대기 LAN을 사용하여 구성하거나 집계된 포트 그룹으로 구성해야 합니다.

주

원격 프로시저 통화(RPC) 프로토콜과 서비스를 사용할 경우에는 개인 하트비트 네트워크를 사용하지 않는 것이 좋습니다. RPC는 각 네트워크 어댑터 장치나 I/O 카드가 라우팅 가능한 네트워크에 연결되어 있다고 가정합니다. 분리된 또는 개인 하트비트 LAN은 라우팅할 수 없으므로, 이러한 LAN으로 이동한 RPC 요청 응답은 결국 시간 제한을 받게 됩니다.

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

STATIONARY_IP

클러스터 하트비트를 전송하지 않는 각각의 모니터링되는 서버넷의 IP 주소입니다. 원하는 만큼 모니터링할 서버넷 수를 지정할 수 있습니다. 응용 프로그램 데이터와 하트비트 메시지를 분리시키려면 여기에서 모니터링되는 하트비트 이외의 서버넷을 정의하십시오.

FIRST_CLUSTER_LOCK_PV, SECOND_CLUSTER_LOCK_PV

클러스터 잠금이 기록되어 있는 잠금 볼륨 그룹에 포함되는 실제 볼륨의 이름입니다. *FIRST_CLUSTER_LOCK_PV*는 첫 번째 물리 잠금 볼륨의 매개 변수이고, *SECOND_CLUSTER_LOCK_PV*는 두 번째 물리 잠금 볼륨의 매개 변수입니다. 두 번째 실제 잠금 볼륨이 있다면 *SECOND_CLUSTER_LOCK_PV*가 별도 행으로 파일에 포함됩니다. 이들 매개 변수는 클러스터에서 서비스를 재편성하기 위해 잠금 디스크를 사용할 때 사용됩니다.

클러스터의 두 노드에 표시되는 것과 동일한 실제 볼륨 이름을 입력하십시오. 각 노드에서 실제 동일한 볼륨이 서로 다른 이름을 가질 수 있습니다. 2개의 클러스터 잠금을 만들려면 각 잠금의 실제 볼륨 이름을 입력하십시오. 실제 볼륨 그룹 식별자는 최대 40자까지 지정할 수 있습니다.

SERIAL_DEVICE_FILE

각 노드에서 선택된 직렬(RS232) 포트에 해당되는 장치 파일의 이름입니다. RS232를 하트비트 회선으로 사용하는 경우에 이 매개 변수를 지정합니다.

ASCII 클러스터 구성 파일에서 이 매개 변수는 *SERIAL_DEVICE_FILE*입니다. 장치 파일 이름은 최대 40자까지 지정할 수 있습니다.

HEARTBEAT_INTERVAL

클러스터의 한 노드에서 다른 노드로 하트비트 메시지를 전송하는 기본 주기입니다. 시간(초)을 입력하십시오.

ASCII 클러스터 구성 파일에서 이 매개 변수는 *HEARTBEAT_INTERVAL*이며 값은 마이크로초 단위로 입력합니다. 기본값은 1,000,000밀리초입니다. 이 매개 변수를 기본값보다 큰 값으로 설정하는 것이 좋습니다.

가능하면 기본값을 사용하는 것이 좋습니다. 권장되는 최대값은 15초이며, 지원되는 최대값은 30초입니다. 이 값은 (아래의) *Node Timeout* 값의 절반보다 커야 합니다.

NODE_TIMEOUT

한 노드에서 다른 노드가 사용 불가능임을 결정하고 클러스터 재편성을 시작할 수 있는 시간 이후의 시간입니다. 이 매개 변수는 밀리초 단위로 입력합니다.

기본값은 ASCII 파일에서 2,000,000 밀리초입니다. 최소값은 2*(하트비트 주기)입니다. 권장되는 최대값은 ASCII 파일에서 30,000,000(SAM에서는 30초)입니다. 기본 설정 값은 가장 빠르게 클러스터를 재편성하게 합니다. 그러나, 기본값을 사용하는 경우 일시적인 시스템 중지나 네트워크 로드 스파이크로 인해 비논리적인 재편성이 발생할 가능성이 증가합니다. 설정 시 상당한 부분에 대해서는 5,000,000에서 8,000,000(5초에서 8초)로 설정하는 것이 적합합니다.

권장되는 최대값은 30초이며, 지원되는 최대값은 60초입니다.

AUTO_START_TIMEOUT

자동 클러스터 시작 작업 중에 노드가 클러스터에 참여하려는 시도를 중지할 때까지 대기하는 시간입니다. ASCII 클러스터 구성 파일에서 이 매개 변수는 *AUTO_START_TIMEOUT*입니다. 모든 노드는 클러스터 시작 작업이 완료되기 전에 다른 노드가 시작될 수 있도록 이 시간 동안 대기합니다. 이 시간은 클러스터에서 가장 늦은 부팅 시간을 기준으로 설정해야 합니다. 가장 늦은 부팅 노드의 부팅 시간에서 가장 빠른 부팅 노드의 부팅 시간을 차감한 다음 600초(10분)을 가산한 값을 입력하십시오.

ASCII 파일에서 기본값은 600,000,000 마이크로초(SAM에서 600초)입니다.

NETWORK_POLLING_INTERVAL

MC/ServiceGuard에 구성된 네트워크를 검사하는 주기입니다. ASCII 클러스터 구성 파일에서 이 매개 변수는 NETWORK_POLLING_INTERVAL입니다.

ASCII 파일에서 기본값은 2,000,000 마이크로초(SAM에서 2초)입니다. 따라서 2초마다 네트워크 관리자는 각 네트워크 인터페이스를 폴링하여 정보를 계속 주고 받을 수 있는지 확인해야 합니다. 이 값을 변경하면 네트워크 장애를 신속히 감지하는 방식에 영향을 미칩니다. 최소값은 1,000,000(1초)입니다. 권장되는 최대값은 15초이며, 지원되는 최대값은 30초입니다.

MAX_CONFIGURED_PACKAGES

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

기본값은 0입니다. 즉, 패키지를 사용하려면 반드시 이 매개 변수를 변경해야 합니다. 최소값은 0이고 최대값은 60입니다. 이 매개 변수를 충분히 크게 설정하여, 차후에 패키지를 추가할 때 클러스터를 다운시키고 매개 변수를 재설정하지 않아도 되게 하십시오. 단, 이 매개 변수를 너무 큰 값으로 설정하여 메모리를 낭비해서는 안됩니다. 모든 클러스터 노드에서 패키지는 6MB 외에 80KB의 잠금 가능한 메모리를 필요로 합니다. CVM 디스크 저장을 사용할 경우 CVM-VXVM-PKG용으로 이러한 크기의 메모리를 추가해야 합니다.

주

각 노드에 있는 HP-UX 커널 매개 변수를 조정하여 해당 노드에서 가장 많은 패키지가 동시에 실행될 수 있을 만큼 충분히 크게 설정되어 있는지 확인합니다.

VOLUME_GROUP

디스크가 클러스터에 구성된 두 개 이상의 노드에 부착된 LVM 볼륨 그룹의 이름입니다. 이 디스크들은 클러스터 인식으로 간주됩니다. ASCII 클러스터 구성 파일에서 이 매개 변수는 *VOLUME_GROUP*입니다. 볼륨 그룹 이름은 최대 40자까지 지정할 수 있습니다.

클러스터 구성 워크시트

다음 워크시트는 클러스터 구성을 기록하고 정리하는 데 도움을 줍니다.

그림 4-7

클러스터 구성 워크시트

```
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_
=====
Autostart Delay: __10 min__
=====
Cluster Aware LVM Volume Groups _____
=====
```

패키지 구성 계획

패키지 구성 계획은 각각의 고가용성 서비스 그룹에 관한 정보를 종합하는 과정입니다. 이 정보 가운데 일부는 패키지 구성 파일을 작성하는 데 이용되고 일부는 패키지 제어 스크립트를 편집하는 데 이용됩니다.

주

또한, 패키지가 활성화할 LVM 볼륨 그룹이 클러스터 구성 파일에서 클러스터 인식 볼륨 그룹으로 정의되어야 합니다. 자세한 내용은 앞에서 설명한 “클러스터 구성 계획” 절을 참조하십시오. 패키지가 활성화할 VERITAS 디스크 그룹은 패키지 구성 ASCII 파일에 정의되어야 합니다. 자세한 내용은 아래에 설명되어 있습니다.

논리 볼륨 및 파일 시스템 계획

클러스터에서 패키지가 실행되기 위한 하부 구조의 일부분으로서, 볼륨 그룹에서 논리 볼륨을 사용해야 할 수도 있습니다. 패키지가 한 노드에서 다른 노드로 이동할 때 패키지는 이전 노드와 같은 디스크에서 데이터를 액세스할 수 있어야 합니다. 볼륨 그룹을 활성화하고 이 볼륨 그룹에 상주하는 파일 시스템을 마운트하여 이 액세스를 가능하게 할 수 있습니다.

MC/ServiceGuard에서 고가용성 응용 프로그램, 서비스 및 데이터는 공유 버스 상의 볼륨 그룹에 위치합니다. 노드에서 장애가 발생하면 응용 프로그램, 서비스 및 데이터를 포함하는 볼륨 그룹은 장애 노드에서 비활성화되고 대체 노드에서 다시 활성화됩니다. 이렇게 하려면 볼륨 그룹이 장애 노드에서 대체 노드로 전송될 수 있도록 볼륨 그룹을 구성해야 합니다.

계획의 일부로서 다음 사항을 결정해야 합니다.

- 어떤 볼륨 그룹이 필요합니까?
- 디스크 공간은 얼마나 필요하며, 디스크 공간을 논리 볼륨에 어떻게 할당해야 합니까?
- 각 패키지에서 어떤 파일 시스템이 마운트되어야 합니까?
- 어떤 노드에 어떤 논리 볼륨 구성을 가져와야 합니까?
- 패키지가 대체 노드로 이동될 때 패키지가 성능에 어떤 영향을 미칠 수 있습니까?

HA(고가용성) 클러스터 계획 및 문서화 Quorum 서버 계획

볼륨 그룹, 논리 볼륨, 파일 시스템의 패키지별 목록을 만드십시오. 시간별로 공통 파일 시스템에 액세스해야 하는 노드를 기록하십시오.

기본 논리 볼륨 이름(예: lv01, lv02)이 아닌 사용자가 정의한 논리 볼륨 이름을 사용하는 것이 좋습니다. 논리 볼륨 이름은 그 논리 볼륨과 관련된 고가용성 응용 프로그램을 의미하는 이름을 선택하십시오. 예를 들어, lv01database와 같은 이름을 사용하면 클러스터 관리가 더 쉬워질 것입니다.

각 노드에서 패키지 관련 볼륨 그룹, 논리 볼륨, 파일 시스템을 더 자세히 문서화하기 위해 주석 줄을 `/etc/fstab` 파일에 추가할 수 있습니다. 다음은 데이터베이스 응용 프로그램의 예제입니다.

```
# /dev/vg01/lv01db1 /applic1 vxfs defaults 0 1 # These six entries are
# /dev/vg01/lv01db2 /applic2 vxfs defaults 0 1 # for information purposes
# /dev/vg01/lv01db3 raw_tables ignore ignore 0 0 # only. They record the
# /dev/vg01/lv01db4 /general vxfs defaults 0 2 # logical volumes that
# /dev/vg01/lv01db5 raw_free ignore ignore 0 0 # exist for MC/ServiceGuard's
# /dev/vg01/lv01db6 raw_free ignore ignore 0 0 # HA package. Do not uncomment.
```

논리 그룹 각각에 대해 항목을 만들고 논리 볼륨의 파일 시스템이나 원시 장치에 대한 용도를 기록하십시오.

주의

MC/ServiceGuard 패키지가 사용하는 파일 시스템을 마운트할 때 `/etc/fstab`을 사용하지 마십시오.

MC/ServiceGuard에서 볼륨 그룹을 만들고, 내보내고, 가져오는 방법에 대한 자세한 내용은 “HA 클러스터 구성 만들기”를 참조하십시오.

EMS 리소스 구성 매개 변수

MC/ServiceGuard는 EMS 리소스를 구성하는 매개 변수 조합을 제공합니다. 이 매개 변수들은 `RESOURCE_NAME`, `RESOURCE_POLLING_INTERVAL`, `RESOURCE_START` 및 `RESOURCE_UP_VALUE`입니다. 이 매개 변수들은 패키지와 종속 관계인 각 리소스의 패키지 구성 파일에 입력합니다. `DEFERRED_RESOURCE_NAME` 매개 변수는 `RESOURCE_START` 매개 변수가 `DEFERRED`로 설정된 리소스의 패키지 제어 스크립트에 추가됩니다.

`RESOURCE_START` 옵션은 ServiceGuard가 EMS 리소스에 대해 리소스 모니터링을 시작하는 시간을 결정할 때 사용합니다. `RESOURCE_START` 옵션을 `AUTOMATIC` 또는 `DEFERRED`로 설정할 수 있습니다.

AUTOMATIC으로 지정된 경우 ServiceGuard 클러스터 데몬이 노드에서 시작되면 ServiceGuard는 이 리소스에 대한 리소스 모니터링을 자동으로 시작합니다. AUTOMATIC으로 설정된 경우에 패키지 제어 스크립트에서 DEFERRED_RESOURCE_NAME을 정의할 필요가 없습니다.

DEFERRED로 설정된 경우, ServiceGuard는 노드가 시작되는 동안 이 DEFERRED 리소스에 대한 리소스 모니터링 시도하지 않습니다. 그러나, 이 DEFERRED 매개 변수는 패키지 제어 스크립트에 지정되어야 하며, DEFERRED_RESOURCE_NAME 매개 변수를 설정합니다. 그러면 패키지 실행 시간 중에 DEFERRED 매개 변수가 패키지 제어 스크립트에서 시작됩니다.

다음은 DEFERRED 및 AUTOMATIC 리소스를 구성하는 방법의 예제입니다. 패키지 구성 파일에서 리소스를 아래와 같이 지정하십시오.

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

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

```
RESOURCE_NAME           /net/interfaces/lan/status/lan2
RESOURCE_POLLING_INTERVAL 60
RESOURCE_START          AUTOMATIC
RESOURCE_UP_VALUE       = ONLINE
```

패키지 제어 스크립트에서 DEFERRED_RESOURCE_NAME 매개 변수를 사용하여 지연된 리소스만 지정합니다.

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

확장 계획

실행 중인 클러스터에 패키지를 추가할 수 있습니다. 이 과정은 “클러스터 및 패키지 관리” 장에 설명되어 있습니다. 패키지를 추가할 때는 클러스터 구성 파일에 정의되어 있는 MAX_CONFIGURED_PACKAGES 값을 초과하지 않도록 주의하십시오.

패키지 수를 늘릴 계획이라면 클러스터 노드에서 각 패키지는 6MB 외에 약 80KB의 잠금 가능한 메모리 필요로 하고, 구성된 각 EMS 리소스 종속성은 약 300KB의 잠금 가능한 메모리를 필요로 한다는 점에 주의하십시오.

전환 및 장애 조치 동작 선택

SAM에서 Automatic Switching이 Enabled로 설정된 경우(ASCII 패키지 구성 파일에서 LOCAL_LAN_FAILOVER_ALLOWED가 YES로 설정된 경우) 동일한 실제 서버넷의 LAN 카드에서 장애가 발생할 때 대기 LAN 카드로 IP 주소를 전환합니다. 기본값은 Automatic Switching Enabled입니다.

장애 조치 동작을 결정하기 위해 현재 실행되고 있지 않은 패키지를 자동으로 실행할 노드를 통제하는 패키지 장애 조치 정책을 정의할 수 있습니다. 또한, 주 노드를 다시 사용할 수 있게 되면 자동으로 패키지가 주 노드로 돌아가는지 여부를 결정하는 장애 복구 정책을 정의할 수 있습니다.

이전 장의 표 3-3에서는 장애 조치 동작의 종류에 대해 설명하고, 각 동작을 결정하는 SAM 또는 ASCII 패키지 구성 파일의 설정을 설명합니다.

패키지 구성 파일 매개 변수

패키지 구성 파일을 만들기 전에 다음과 같은 패키지 구성 데이터를 정리하십시오. 다음의 매개 변수 이름은 SAM에 나타나는 이름입니다. ASCII 클러스터 구성 파일에서 사용하는 이름은 각 항목 끝에 표시됩니다. 각 패키지별로 다음 매개 변수를 확인하여 워크시트에 기입해야 합니다.

패키지 이름

패키지의 이름입니다. 패키지 이름은 클러스터에서 고유해야 합니다. 패키지를 시작하고, 중지하고, 수정하고, 볼 때 패키지 이름을 사용합니다. ASCII 패키지 구성 파일에서 이 매개 변수는 PACKAGE_NAME입니다.

패키지 이름은 1자부터 40자까지 지정할 수 있습니다. 패키지 이름에는 공백, 슬래시(/), 백슬래시(\) 및 별표(*) 같은 문자를 사용할 수 없습니다. 이외의 다른 문자는 모두 사용할 수 있습니다.

PACKAGE_TYPE

패키지의 종류입니다. 이 매개 변수는 패키지를 동시에 하나의 노드에서 실행할지 또는 여러 노드에서 실행할지 여부를 표시합니다. 설정할 수 있는 형식은 FAILOVER와 SYSTEM_MULTI_NODE입니다. 기본값은 FAILOVER입니다.

SYSTEM_MULTI_NODE 형식은 모든 클러스터 노드 간의 공유 CVM 데이터 저장을 지원하는 CVM-VXVM-PKG 패키지를 정의하는 데만 사용됩니다. SYSTEM_MULTI_NODE 형식으로 사용자 정의 패키지를 만들 수 없습니다.

패키지 장애 조치 정책

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

기본값은 `CONFIGURED_NODE`로, 패키지의 노드 이름 목록에서 다음으로 사용 가능한 노드가 선택됩니다. 이 동작은 이전 버전의 `MC/ServiceGuard`와 동일합니다. 노드 이름 항목의 순서는 노드 선택 시 우선 순위가 됩니다. 다른 정책으로는 `MIN_PACKAGE_NODE`가 있습니다. 이 정책은 이 패키지가 시작될 때에 가장 적은 수의 다른 패키지를 실행 중인 노드를 시작합니다.

패키지 장애 복구 정책

패키지가 해당 주 노드에 실행되고 있지 않지만 그 주 노드에서 이 패키지를 실행할 수는 있을 경우 패키지 관리자가 취할 조치를 결정하는 데 사용하는 정책입니다. ASCII 패키지 구성 파일에서 이 매개 변수는 `FAILBACK_POLICY`입니다.

기본값은 `MANUAL`로, 패키지가 대체 노드에서 실행 중일 때 패키지를 주 노드로 다시 이동하려는 시도를 하지 않습니다. 이 동작은 이전 버전의 `MC/ServiceGuard`와 동일합니다. 다른 정책으로는 `AUTOMATIC`이 있습니다. 이 정책은 `MIN_PACKAGE_NODE`가 패키지 장애 조치 정책으로 선택되었고 패키지의 주 노드가 현재 패키지가 실행 중인 노드보다 적은 수의 패키지를 실행할 경우 주 노드가 다시 패키지를 실행할 수 있게 되었을 때 패키지를 중지하고 주 노드에서 패키지를 다시 시작합니다.

노드 이름

패키지의 주 노드와 대체 노드의 이름입니다. ASCII 구성 파일에서 이 매개 변수는 `NODE_NAME`입니다. 패키지가 실행할 수 있는 각 노드의 노드 이름을 입력합니다.

노드 이름을 지정하는 순서는 중요합니다. 우선 주 노드 이름을 선택한 다음 첫 번째 대체 노드 이름, 그 다음에 두 번째 대체 노드 이름을 순으로 선택하십시오. 장애 조치에서 패키지 제어는 패키지 구성 파일에 나열된 다음 대체 노드 이름으로 전송됩니다.

노드 이름은 최대 40자까지 지정할 수 있습니다.

자동 전환 (Auto Run)

SAM에서 이 매개 변수의 이름은 Automatic Switching입니다. ASCII 패키지 구성 파일에서 이 매개 변수는 `AUTO_RUN`이며, 이전에는 `PKG_SWITCHING_ENABLED`로 명명되었습니다. 이 매개 변수는 패키지 시작 방식을 결정합니다. Automatic Switching이 Enabled로 설정된 경우(`AUTO_RUN`이 YES로 설정된 경우) 패키지는 사용 가능한 노드에서 자동으로 시작되어 다른 노드로 자동으로 장애 조치를 수행하게 됩니다. Automatic Switching이 Disabled(`AUTO_RUN`이 NO)로 설정된 경우 클러스터가 실행되지 않으면 패키지는 자동으로 시작되지 않으며 다른 노드로 자동으로 장애 조치를 수행할 수 없습니다.

기본값은 SAM에서는 Enabled(ASCII 파일에서는 `AUTO_RUN`이 YES)입니다. 기본값으로 지정하면 사용 가능한 노드에서 패키지가 정상적으로 시작됩니다. SAM에서 Enabled 또는 Disabled를 입력합니다(ASCII 파일에서는 `AUTO_RUN YES` 또는 NO).

로컬 전환

Enabled 또는 Disabled를 입력하십시오. 이것은 장애가 발생했을 때 MC/ServiceGuard가 로컬에서 LAN을 전환할 수 있도록 허용합니다. 즉, 장애가 발생하면 패키지 IP 주소를 대기 LAN 카드로 전송할 수 있습니다. 기본값은 Enabled입니다. ASCII 패키지 구성 파일에서 이 매개 변수는 `LOCAL_LAN_FAILOVER_ALLOWED`이며 선택 가능한 값은 YES와 NO입니다.

기본값은 SAM에서 Enabled(ASCII 파일에서 (YES)입니다.

노드 조기 차단

Enabled 또는 Disabled를 입력하십시오. 제어 스크립트에 장애가 발생하거나, 서버넷에 장애가 발생하거나, 리소스 중지를 표시하는 EMS 모니터 이벤트를 통해 보고하는 경우, 이 매개 변수가 Enabled로 설정되어 있다면 MC/ServiceGuard는 제어 스크립트에 장애가 발생한 노드에서 TOC를 실행합니다. TOC 실행 이전에 제일 먼저 노드를 다시 시작해야 합니다. ASCII 패키지 구성 파일에서 이 매개 변수 이름은 NODE_FAIL_FAST_ENABLED이며 YES와 NO를 선택할 수 있습니다.

기본값은 SAM에서 Disabled(ASCII 파일에서 NO)입니다.

NODE_FAIL_FAST_ENABLED가 YES로 설정되어 있는 경우 아래 장애 동작 중 하나가 발생하면 패키지가 실행 중인 노드는 중지됩니다.

- 패키지 서버넷에 장애가 발생하였으며 백업 네트워크를 사용할 수 없는 경우
- EMS 리소스에 장애가 발생한 경우
- 중지 스크립트가 종료되지 않은 경우
- ServiceGuard가 중지 스크립트를 실행할 수 없는 경우
- 중지 스크립트의 시간이 제한되는 경우

그러나 패키지 중지 스크립트에 "exit 1"으로 장애가 발생하면 ServiceGuard가 이 노드를 중지시키지는 않지만 패키지에 대해 NO_RESTART를 설정하여 패키지 전환(AUTO_RUN)을 사용할 수 없도록 하여 패키지가 대체 노드에서 실행되지 않도록 합니다.

제어 스크립트 경로 이름

패키지 제어 스크립트의 전체 경로 이름을 입력하십시오. 스크립트는 "cmcluster"를 포함하는 디렉토리에 있어야 합니다. 이 스크립트는 실행 가능해야 합니다. ASCII 패키지 구성 파일에서 이 매개 변수는 RUN_SCRIPT 매개 변수와 HALT_SCRIPT 매개 변수로 매핑됩니다.

실행 스크립트와 중지 스크립트로 동일한 스크립트를 사용하는 것이 좋습니다. 이 스크립트에는 패키지 실행 명령과 패키지 중지 명령이 포함되어 있습니다. 패키지 실행 명령과 패키지 중지 명령을 별도의 스크립트로 분리하려면, 패키지 구성 파일에서 2개의 스크립트 이름을 지정하면 됩니다. 하지만, 대부분의 경우에는 ASCII 파일에서 단일 제어 스크립트 이름을 RUN_SCRIPT와 HALT_SCRIPT로 사용하는 편이 훨씬 편리합니다. 패키지가 시작될 때 패키지의 실행 스크립트가 실행되고 매개 변수 'start'가 전달됩니다. 마찬가지로 패키지가 중지될 때 중지 스크립트가 실행되고 매개 변수 'stop'이 전달됩니다.

별도의 패키지 실행 및 중지 스크립트를 기록하려면 두 스크립트에 동일한 구성 정보(예: 노드 이름, IP 주소)를 포함시켜야 합니다.

실행 스크립트 시간 제한 및 중지 스크립트 시간 제한

지정된 시간 제한 값에서 스크립트가 완성되지 않으면 MC/ServiceGuard는 스크립트를 종료합니다. ASCII 구성 파일에서 이 매개 변수는 RUN_SCRIPT_TIMEOUT 및 HALT_SCRIPT_TIMEOUT입니다. 값은 초 단위로 입력하십시오.

기본값은 0으로, 시간 제한이 없습니다. 최소값은 10초이지만, 최소 HALT_SCRIPT_TIMEOUT 값은 모든 Service Halt Timeout 값의 합계보다 커야 합니다. 절대적 최대값은 절대적 한계 4,294초에 대한 HP-UX 매개 변수 ULONG_MAX에 의해서만 제한됩니다.

시간 제한을 초과하면 다음과 같은 상황이 발생합니다.

- 패키지의 제어 권한이 전송되지 않습니다.
- 실행 또는 중지 명령이 실행되지 않습니다.
- 전체 전환이 일어나지 않습니다.
- 현재 노드가 해당 패키지를 실행할 수 없게 됩니다.
- 제어 스크립트가 status 1로 종료됩니다.

중지 스크립트 시간 제한이 발생하면, 수동으로 정리 작업을 수행해야 합니다. 자세한 내용은 “패키지 제어 스크립트 중지 또는 장애”를 참고하십시오.

주

VxVM 디스크 그룹은 패키지 실행 시간에 가져오고 패키지 중지 시간에 내보냅니다. 패키지에서 보다 많은 수의 VxVM 디스크를 사용할 경우 이들 디스크가 가져오거나 내보내기 작업을 마칠 수 있을 정도로 시간 제한이 늘어납니다.

저장 그룹

이 매개 변수는 CVM 디스크 그룹에 사용됩니다. 패키지에서 사용할 모든 CVM 디스크 그룹 이름을 입력하십시오.

ASCII 패키지 구성 파일에서 이 매개 변수는 STORAGE_GROUP입니다.

주

패키지 ASCII 구성 파일의 VxVM 디스크 그룹 이름이나 LVM 볼륨 그룹 이름을 입력하지 마십시오.

서비스 이름

각 서비스의 고유 이름을 입력하십시오. ASCII 패키지 구성 파일에서 이 매개 변수는 SERVICE_NAME입니다.

각 서비스 당 하나의 SERVICE_NAME 항목을 정의하십시오. 한 패키지에는 최대 30개의 서비스를 구성하거나 클러스터마다 최대 900개의 서비스를 구성할 수 있습니다. 서비스 이름에는 공백, 슬래시(/), 백슬래시(\) 및 별표(*) 같은 문자를 사용할 수 없습니다. 이외의 다른 문자는 모두 사용할 수 있습니다. 서비스 이름은 최대 40자까지 지정할 수 있습니다.

서비스 조기 차단

각 서비스에 대해 Enabled 또는 Disabled를 입력하십시오. 이 매개 변수는 서비스의 장애를 노드의 장애로 간주할 것인지의 여부를 지정합니다. 이 매개 변수가 Enabled로 설정되면 서비스에서 장애가 발생할 경우 MC/ServiceGuard는 서비스가 실행 중인 노드를 TOC와 함께 중지시킵니다. TOC에 앞서 시스템을 다시 시작하려고 시도합니다. 기본값은 Disabled입니다.

ASCII 패키지 구성 파일에서 이 매개 변수는 `SERVICE_FAIL_FAST_ENABLED`이며 선택 가능한 값은 YES와 NO입니다. 기본값은 NO입니다. 서비스 당 하나의 `SERVICE_FAIL_FAST_ENABLED` 항목을 정의하십시오.

서비스 중지 시간 제한

서비스가 중지되면 MC/ServiceGuard는 우선 SIGTERM 신호를 보내 서비스를 종료합니다. 만약 프로세스가 종료되지 않으면 MC/ServiceGuard는 지정된 시간 제한 값까지 대기한 다음 SIGKILL 신호를 보내 프로세스를 강제로 종료합니다. ASCII 패키지 구성 파일에서 이 매개 변수는 `SERVICE_HALT_TIMEOUT`입니다. 각 서비스 당 하나의 `SERVICE_HALT_TIMEOUT` 항목을 정의하십시오.

기본값은 300초(5분)입니다. 최소값은 1초이며, 최대값은 절대적 한계 4,294초에 대한 HP-UX 매개 변수 `ULONG_MAX`에 의해서만 제한됩니다.

각 서비스 당 하나의 `SERVICE_HALT_TIMEOUT` 항목을 정의하십시오.

서브넷

패키지에 대해 모니터링될 IP 서브넷을 입력하십시오.

ASCII 패키지 구성 파일에서 이 매개 변수는 `SUBNET`입니다.

리소스 이름

패키지가 의존하는 대상으로서 MC/ServiceGuard가 모니터링할 리소스의 이름입니다. ASCII 패키지 구성 파일에서 이 매개 변수는 `RESOURCE_NAME`입니다.

리소스 이름은 특정한 시스템 리소스의 중요한 속성의 이름입니다. 리소스 이름은 시스템에서 리소스가 존재하는 리소스 클래스 및 하위 클래스의 전체 계층 구조를 포함합니다. 리소스 이름은 SAM에서 제공되는 목록 또는 리소스 모니터와 함께 제공되는 문서에서 찾을 수 있습니다.

클러스터 당 최대 60개의 리소스를 정의할 수 있습니다. 다음에 설명하는 리소스 Up 값의 한계도 참고하십시오. 모든 클러스터 노드에서 구성된 각 리소스는 약 300KB의 잠금 가능한 메모리를 요구합니다.

리소스 이름은 최대 1024자까지 지정할 수 있습니다.

리소스 폴링 주기

구성된 패키지 리소스를 모니터링하는 주기입니다. ASCII패키지 구성 파일에서 이 매개 변수는 RESOURCE_POLLING_INTERVAL입니다.

기본값은 60초입니다. 최소값은 1이며, 최대값에는 기능적 한계가 없습니다.

리소스 폴링 주기는 SAM에서 제공되는 목록 또는 리소스 모니터와 함께 제공되는 문서에서 찾을 수 있습니다.

리소스 시작

패키지 시작 이전이나 이후에 시작할지를 결정하는 리소스 속성입니다. ASCII 패키지 구성 파일에서 이 매개 변수는 RESOURCE_START입니다.

기본 설정은 AUTOMATIC로, 노드가 클러스터에 참여하면 리소스가 시작됩니다. 또는, DEFERRED로 설정할 수 있으며, 이 때는 패키지 서비스가 시작된 다음에 리소스가 시작됩니다. 리소스가 DEFERRED로 설정되어 있는 경우에는 리소스 이름을 제어 스크립트의 DEFERRED_RESOURCE_NAME 매개 변수에 추가해야 합니다.

리소스 Up 값

추가 패키지 리소스에 장애가 있는지 판단하는 기준입니다. ASCII 패키지 구성 파일에서 이 매개 변수는 RESOURCE_UP_VALUE입니다. 리소스 Up 값은 SAM에서 제공되는 목록 또는 리소스 모니터와 함께 제공되는 문서에서 찾을 수 있습니다.

패키지 당 최대 15개의 리소스 Up 값을 구성할 수 있습니다. 예를 들어, 패키지에 리소스가 하나뿐이라면 최대 15개의 리소스 Up 값을 정의할 수 있습니다. 두 개의 리소스 이름이 정의되었고 그 중 하나가 10개의 리소스 Up 값을 가지고 있다면 다른 리소스 이름은 5개의 리소스 Up 값만 가질 수 있습니다.

리소스 Up 값 문자열은 최대 1024자까지 지정할 수 있습니다.

패키지 구성 워크시트

아래 예제에 나타난 대로 각 패키지에 대해 패키지 구성 데이터를 별도의 워크시트에 정리하십시오.

그림 4-8

패키지 구성 워크시트

```
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_____
```

패키지 제어 스크립트 변수

일련의 변수에 값을 할당하기 위해 각 패키지의 제어 스크립트도 편집해야 합니다. 다음 변수를 설정할 수 있습니다.

PATH

스크립트에서 사용할 경로를 지정합니다.

VGCHANGE

LVM 볼륨 그룹의 활성화 방법을 지정합니다. 볼륨 그룹을 배타적 모드에서 활성화하려면 기본값(`VGCHANGE="vgchange -a e"`)을 그대로 두십시오. 여기서 볼륨 그룹 작성 시 `vgchange -c y`를 사용하여 볼륨 그룹이 초기화되었다고 가정합니다.

별도의 실제 경로에 디스크가 미리링되었으면 `VGCHANGE="vgchange -a y"`를 사용하십시오.

디스크가 별도의 실제 경로에서 미리링되었고 패키지 시작과 동시에 미리 재동기화 작업을 실행하려면 `VGCHANGE="vgchange -a y"`를 사용하십시오.

비 배타적 활성화 모드를 사용하려면 `VGCHANGE="vgchange -a y"`를 사용하십시오. 단일 노드 클러스터 구성에서는 비 배타적 활성화 모드를 사용해야 합니다.

CVM_ACTIVATION_CMD

VERITAS CVM 디스크 그룹 활성화를 위해 이 명령을 지정하십시오.

배타적 쓰기 모드에서 디스크 그룹을 활성화하려면 기본 `CVM_ACTIVATION_CMD="vxdg -g \${DiskGroup} set activation=exclusivewrite"`를 사용하십시오.

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

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

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

주

VxVM 디스크 그룹을 사용하면 특정 활성화 명령을 선택할 수 없습니다. VxVM 디스크 그룹 활성화에는 항상 동일한 명령을 사용합니다.

VXVOL

미러링된 VxVM 볼륨에 대한 미리 복구 방법을 제어합니다.

복구가 완료될 때까지 패키지 제어 스크립트가 대기하도록 하려면 기본 `VXVOL="vxvol -g \${DiskGroup} startall"`을 사용하십시오.

패키지 시작과 함께 미리 재동기화를 시작하게 하려면 `VXVOL="vxvol -g \${DiskGroup} -o bg startall"`을 사용하십시오.

볼륨 그룹

이 어레이 매개 변수에는 패키지에서 활성화할 LVM 볼륨 그룹이 포함되어 있습니다. 별도 행에 각 VG를 입력하십시오.

CVM 디스크 그룹

이 어레이 매개 변수에는 패키지에서 사용할 VERITAS CVM 디스크 그룹 목록이 포함되어 있습니다. 별도 행에 각 디스크 그룹을 입력하십시오.

VxVM 디스크 그룹

이 어레이 매개 변수에는 패키지에서 활성화할 VERITAS VxVM 디스크 그룹 목록이 포함되어 있습니다. 별도 행에 각 디스크 그룹을 입력하십시오.

논리 볼륨, 파일 시스템 및 마운트 옵션

이 어레이 매개 변수는 어레이 변수에 이 세 변수를 함께 입력합니다. 각 세 변수는 파일 시스템, 논리 볼륨, 패키지에서 사용한 파일 시스템의 마운트 옵션을 지정합니다. 패키지 제어 스크립트 파일에서 이러한 변수는 LV, FS 및 FS_MOUNT_OPT같은 어레이입니다.

패키지가 시작되는 동안 스크립트는 하나 이상의 저장 그룹을 활성화하고 파일 시스템으로 논리 볼륨을 마운트합니다. 패키지가 중지되는 동안 스크립트는 파일 시스템을 언마운트하고 각 저장 그룹을 비활성화합니다. 모든 저장 그룹은 각 대상 노드에서 액세스할 수 있어야 합니다(CVM 디스크 그룹은 클러스터의 모든 노드에 액세스할 수 있어야 함). 각 파일 시스템(FS)에서 논리 볼륨(LV)을 식별해야 합니다.

논리 볼륨은 LVM 볼륨 그룹, VERITAS CVM 디스크 그룹 또는 VERITAS VxVM 디스크 그룹에 만들 수 있습니다. 사용된 저장 그룹 종류에 관계 없이 어느 순서로든 LV를 입력할 수 있습니다.

파일 시스템 언마운트 횟수

패키지가 중지되는 동안 각 파일 시스템에 허용된 언마운트 시도 횟수입니다. 기본값은 1입니다.

파일 시스템 마운트 재시도 횟수

각 파일 시스템의 마운트 재시도 횟수입니다. 기본값은 0입니다. 패키지가 시작되는 동안 마운트 지점이 사용 중이고 FS_MOUNT_RETRY_COUNT가 0이라면 패키지는 시작되지 않고 스크립트는 1로 종료됩니다. 마운트 지점이 사용 중이고 FS_MOUNT_RETRY_COUNT가 0보다 큰 값이라면 스크립트에서는 사용 중인 마운트 지점에 책임이 있는 사용자를 중지시키기 위해 FS_MOUNT_RETRY_COUNT에 지정된 횟수만큼 시도합니다. 지정된 횟수만큼 시도한 후에도 마운트에 오류가 발생하면 스크립트는 1로 종료됩니다.

CONCURRENT_VGCHANGE_OPERATIONS

패키지 시작 및 종료 중에 허용할 동시 볼륨 그룹 활성화 또는 활성화 해제 수를 지정합니다. 기본값은 1입니다. 볼륨 그룹을 여러 개 사용할 경우 이 변수 값을 크게 설정하면 성능이 향상됩니다. 1보다 작은 값을 지정하면 스크립트가 변수를 1로 기본 설정하고 패키지 제어 스크립트 로그 파일에 경고 메시지를 기록합니다.

CONCURRENT_DISKGROUP_OPERATIONS

패키지 시작 및 종료 중에 허용할 동시 VxVM 디스크 그룹 가져오기 또는 내보내기 수를 지정합니다. 기본값은 1입니다. 디스크 그룹을 여러 개 사용할 경우 이 변수 값을 크게 설정하면 성능이 향상됩니다. 1보다 작은 값을 지정하면 스크립트가 변수를 1로 기본 설정하고 패키지 제어 스크립트 로그 파일에 경고 메시지를 기록합니다.

CONCURRENT_FSCK_OPERATIONS

패키지 시작 중에 허용할 동시 fsck 명령 수를 지정합니다. 기본값은 1입니다. 파일 시스템을 여러 개 검사할 경우 이 변수 값을 크게 설정하면 성능이 향상됩니다. 1보다 작은 값을 지정하면 스크립트가 변수를 1로 기본 설정하고 패키지 제어 스크립트 로그 파일에 경고 메시지를 기록합니다.

CONCURRENT_MOUNT_OPERATIONS

패키지 시작 또는 종료 중에 허용할 동시 마운트 및 언마운트 수를 지정합니다. 기본값은 1입니다. 파일 시스템 여러 개를 마운트하거나 언마운트할 경우 이 변수 값을 크게 설정하면 성능이 향상됩니다. 1보다 작은 값을 지정하면 스크립트가 변수를 1로 기본 설정하고 패키지 제어 스크립트 로그 파일에 경고 메시지를 기록합니다.

IP 주소와 서브넷

패키지를 LAN 카드에 매핑하는 IP 주소입니다. 인터페이스 카드에 추가할 각 IP 주소에 대해 IP 주소와 서브넷을 지정하십시오.

패키지 제어 스크립트 파일에서 이 변수들은 한 쌍으로 입력됩니다. 예를 들면, IP[0]=192.10.25.12와 SUBNET[0]=192.10.25.0입니다. 이 경우 서브넷 마스크는 255.255.255.0입니다.

서비스 이름

패키지의 각 특정 서비스에 대한 고유한 이름을 입력하십시오. 모든 서비스는 MC/ServiceGuard에 의해 모니터링됩니다. 한 패키지에는 최대 30개의 서비스를 지정할 수 있습니다. 모든 이름은 클러스터에서 고유해야 합니다. 서비스 이름은 패키지 제어 스크립트에 구성된 cmrunserv와 cmhalt.serv에서 사용하는 이름입니다. 이 서비스 이름은 패키지 ASCII 구성 파일의 서비스 이름과 동일해야 합니다.

패키지 제어 스크립트 파일에서 이 값을 SERVICE_NAME 어레이에 입력합니다. 각 서비스 당 한 개의 서비스 이름을 입력합니다.

SERVICE_NAME, SERVICE_CMD 및 SERVICE_RESTART 매개 변수는 3개 그룹으로 패키지 제어 스크립트에 설정됩니다.

서비스 이름에는 공백, 슬래시(/), 백슬래시(\) 및 별표(*) 같은 문자를 사용할 수 없습니다. 이외의 다른 문자는 모두 사용할 수 있습니다. 서비스 이름은 최대 40자까지 지정할 수 있습니다.

서비스 명령

이름을 지정한 각 서비스에 대해 서비스 명령을 입력하십시오. 제어 스크립트에서 이 명령은 cmrunserv 명령을 이용하여 실행됩니다.

니다.

패키지 구성 스크립트 파일에서 이 값을 `SERVICE_CMD` 어레이에 입력합니다. 각 서비스 당 하나의 서비스 명령 문자열을 입력합니다. `SERVICE_NAME`, `SERVICE_CMD` 및 `SERVICE_RESTART` 매개 변수는 3개 그룹으로 패키지 제어 스크립트에 설정됩니다.

서비스 재시작 매개 변수

재시작 횟수를 입력하십시오. 유효한 매개 변수 형태 중 하나는 `"-r n"`입니다. 여기서 `n`은 재시작 횟수입니다. 값이 `"-r 0"`이면 재시도를 하지 않습니다. 값이 `"-R"` 이면 계속 재시도를 합니다. 기본 값은 0으로, 재시작하지 않습니다.

ASCII 패키지 제어 스크립트에서 이 값은 `SERVICE_RESTART` 어레이에 입력하십시오. 서비스당 하나의 재시작 값을 입력하십시오. `SERVICE_NAME`, `SERVICE_CMD` 및 `SERVICE_RESTART` 매개 변수는 3개 그룹으로 패키지 제어 스크립트에 설정됩니다.

보류된 리소스 이름

패키지 구성 ASCII 파일에 지정된 각 보류 파일에 대해 리소스 이름을 제어 스크립트의 이 어레이에 입력해야 합니다. 이 리소스 이름은 패키지 ASCII 구성 파일의 `RESOURCE_NAME` 매개 변수에 나타난 것과 동일해야 합니다.

패키지 제어 스크립트 파일에서 `DEFERRED_RESOURCE_NAME` 어레이에 값을 입력하십시오. 이 이름은 패키지 ASCII 구성 파일의 `RESOURCE_START` 매개 변수와 나열된 리소스 이름과 동일해야 합니다.

DTC 관리자 데이터

각 DTC에 대한 DTC 이름을 입력하십시오. DTC를 MC/ServiceGuard와 함께 사용하는 방법에 대한 자세한 내용은 *Using the HP DTC Manager/UX* 설명서에서 “Configuring DTC Manager for Operation with MC/ServiceGuard” 장을 참조하십시오.

오류가 발생하면 패키지 제어 스크립트는 환경을 정리하고 작업을 원래대로 되돌립니다. 자세한 내용은 3장의 “패키지 제어 스크립트의 작동 방식” 절을 참조하십시오.

제어 스크립트 워크시트

각 패키지에 대한 패키지 제어 스크립트 데이터는 아래 예에서처럼 별도의 워크시트에 정리해두십시오.

그림 4-9

패키지 제어 스크립트 워크시트

```
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: ___svcl___ Command: ___/usr/bin/MySvc -f___ Restart: ___r 2___

Service Name: _____ Command: _____ Restart: ___

Deferred Resource Name _____
```


5 HA(고가용성) 클러스터 구성 만들기

이 장과 다음 장에서는 MC/ServiceGuard 클러스터를 설치하는 데 필요한 구성 작업을 설명합니다. 이 과정은 구성 노드(Configuration Node)라는 하나의 노드에서 수행되며, 이 결과로 생성되는 이진 파일은 MC/ServiceGuard에 의해 클러스터의 모든 노드에 배포됩니다. 이 장의 예제에서 구성 노드는 *ftsyst9*이며, 예제 대상 노드는 *ftsyst10*입니다. 이 장에서는 다음의 클러스터 구성 작업을 설명합니다.

- 시스템 준비
- Quorum 서버 설치
- MC/ServiceGuard 설치
- LVM을 사용하여 저장 하부 구조 만들기
- VxVM을 사용하여 저장 하부 구조 만들기
- 클러스터 구성
- CVM을 사용하여 저장 하부 구조 만들기
- 실행 클러스터 관리

구성 패키지는 다음 장에서 설명합니다.

대부분의 구성 작업은 SAM(System Administration Manager) 또는 MC/ServiceGuard 명령을 사용하여 수행할 수 있습니다. SAM 고가용성 옵션을 사용하여 클러스터와 패키지를 구성하고 전체 노드 또는 개별 노드에서 클러스터를 시작할 수 있습니다. SAM의 고가용성 옵션을 사용할 때는 다음과 같은 SAM의 사용자 인터페이스 특징을 알아야 합니다.

- SAM은 개체 동작 모델을 사용합니다. 개체를 선택한 다음 해당 개체에 대한 동작을 수행합니다. 사용할 수 있는 작업 메뉴는 개체의 선택 여부에 따라 달라질 수 있습니다.
- 목록에서 항목을 선택할 때 항상 신중하게 선택해야 합니다. 항목을 마우스로 선택하거나 Tab 키로 항목까지 이동한 다음 **OK**를 선택하면 항목을 선택할 수 있습니다. 항목이 강조 표시된 경우에도 기본적으로 항목은 선택되어 있지 않습니다.

HA(고가용성) 클러스터 구성 만들기

- 마우스로 목록에서 둘 이상의 항목을 선택하려면 마우스 왼쪽 단추로 첫째 항목을 선택한 다음 **Ctrl** 키를 누른 상태에서 마우스 왼쪽 단추로 다음 항목을 선택합니다. 마지막으로 **OK**를 눌러 항목을 선택합니다.

구성 및 관리 화면에 항목과 관련된 다양한 도움말이 나타납니다.

MC/ServiceGuard 명령을 사용하여 클러스터와 패키지를 구성할 경우 각 명령의 온라인 도움말에서 구문 및 사용법에 대한 정보를 얻을 수 있습니다.

시스템 준비

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

파일이 있는 위치에 대한 정보

ServiceGuard는 /etc/cmcluster.conf라는 파일을 사용하여 HP-UX 파일 시스템 내에 구성 및 로그 파일 위치를 정의합니다. 다음 위치가 이 파일에 정의되어 있습니다.

```
##### cmcluster.conf#####
#
# Highly Available Cluster file locations
#
# This file must not be edited
#####

SGCONF=/etc/cmcluster
SGSBIN=/usr/sbin
SGLBIN=/usr/lbin
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 정의되어 있지 않으면 사용자 **루트**에 대해 프로필 로그인에 /etc/cmcluster.conf를 포함시킵니다.

이 설명서에서 시스템 파일 이름은 일반적으로 이러한 위치 접두어 중 하나와 함께 제공됩니다. 그러므로 이 파일에 제공된 접두어 정의를 제공하면 \$SGCONF/<FileName>에 대한 참조를 해결할 수 있습니다. 예를 들어, SGCONF가 /etc/cmcluster/conf로 정의 되면 \$SGCONF/cmclconfig 파일에 대한 전체 경로명은 /etc/cmcluster/conf/cmclconfig가 됩니다.

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

보안 파일 편집

MC/ServiceGuard는 ARPA 서비스를 사용하여 클러스터 노드 간의 보안 통신을 유지할 수 있습니다. MC/ServiceGuard를 설치하기 전에 클러스터에서 다른 노드의 루트 사용자가 액세스하도록 허용된 노드를 찾아야 합니다. 그렇지 않으면 MC/ServiceGuard는 구성 중에 노드 간에 파일을 복사할 수 없습니다.

/.rhosts 파일을 사용하여 클러스터 내의 모든 노드 간 통신 보안을 유지하는 대신 MC/ServiceGuard를 사용하면 대체 파일 /etc/cmcluster/cmclnodelist를 지정하여 클러스터 내의 노드 간 액세스 권한을 검사할 수 있습니다. 네트워크 보안 서비스와 관련하여 클러스터 환경에 NIT나 NIS+가 구성되어 있는 경우 /etc/cmcluster/cmclnodelist를 사용하는 것이 좋습니다. MC/ServiceGuard는 먼저 /etc/cmcluster/cmclnodelist 파일이 있는지 확인합니다. /etc/cmcluster/cmclnodelist 파일이 있으면 MC/ServiceGuard는 이 파일만을 사용하여 클러스터내 액세스를 검증합니다. 이 파일 없을 때는 /.rhosts 파일이 사용됩니다.

/etc/cmcluster/cmclnodelist 파일의 입력 형식은 다음과 같습니다.

```
[hostname or IP address]      [rootuser]      [#comment]
```

아래 예제에서는 파일을 편성하는 방법을 보여줍니다.

아래에서 NodeA 및 NodeB.sys.dom.com은 동일한 서버넷에 구성되어 있습니다.

```
NodeA                root                # cluster1
NodeB.sys.dom.com    root                # cluster1
```

아래 예제에서 NodeA와 NodeB는 2개의 공용 서버넷인 192.6.1.0과 192.6.5.0을 공유합니다. NodeA는 서버넷 192.6.1.0에 구성되어 있으며 클라이언트는 192.6.1.10을 공식적인 IP 주소로 알고 있습니다. NodeA는 또 다른 IP 주소인 192.6.1.20도 가지고 있습니다. 이 주소는 서버넷 192.6.1.0의 통신에 사용됩니다. cmclnodelist 파일에는 다른 모든 노드로부터 받은 ServiceGuard 메시지에 대한 사용 권한을 허가하기 위해 이 IP 주소 모두가 포함되어야 합니다.

```
NodeA                root                # cluster1
192.60.50.10         root                # cluster1
NodeB.sys.dom.com    root                # cluster1
192.6.1.20           root                # cluster1
```

`/.rhosts` 또는 `/etc/cmcluster/cmclnodelist` 파일은 모든 클러스터 노드에 복사해야 합니다. `MC/ServiceGuard`는 `/etc/cmcluster/cmclnodelist` 및 `/.rhosts` 파일에서 모두 완전한 도메인 이름을 지원합니다.

주

`/.rhosts` 파일은 `group`이나 `other`에게 액세스 쓰기 권한을 허용하지 않아야 합니다. 이러한 쓰기 권한이 허용되어 있으면 `MC/ServiceGuard` 명령에 오류가 발생하고 “Permission Denied for User” 메시지가 기록됩니다.

`/etc/hosts.equiv` 및 `/var/adm/inetd.sec` 파일을 사용하여 다른 수준의 클러스터 보안을 제공할 수도 있습니다. 자세한 내용은 HP-UX 안내서 **시스템 및 작업 그룹 관리**를 참조하십시오.

루트 논리 볼륨의 미러 만들기

모든 클러스터 노드에서 미러링된 루트 볼륨을 사용하는 것이 바람직합니다. 다음 과정은 별도의 부트 및 루트 볼륨을 사용하고 있다는 가정 하에 부트 볼륨(`/dev/vg00/1vol1`), 기본 스왑(`/dev/vg00/1vol2`) 및 루트 볼륨(`/dev/vg00/1vol3`)의 미러를 만듭니다. 이 과정은 SAM을 사용하여 수행할 수 없습니다. 이 예제와 다음 명령에서 `/dev/dsk/c4t5d0`은 기본 디스크이고 `/dev/dsk/c4t6d0`은 미러입니다. 시스템의 루트 디스크에 올바른 장치 파일 이름을 사용하십시오.

1. 미러에 사용할 부팅 가능한 LVM 디스크를 만듭니다.


```
# pvcreate -B /dev/rdisk/c4t6d0
```
2. 이 디스크를 현재 루트 볼륨 그룹에 추가합니다.


```
# vgextend /dev/vg00 /dev/dsk/c4t6d0
```
3. 새 디스크로 부팅 디스크를 만듭니다.


```
# mkboot -l /dev/rdisk/c4t6d0
```
4. 부트, 기본 스왑 및 루트 논리 볼륨을 부팅 가능한 새로운 디스크로 미러링합니다. `/usr`, `/swap` 같은 `vg00`의 모든 장치가 미러링되었는지 확인합니다

주

부트, 루트 및 스왑 논리 볼륨은 **반드시** 아래 순서대로 미러링되어, 부트 볼륨이 새 디스크의 연속적 확장 세트의 맨 앞에 위치하고, 스왑 볼륨과 루트 볼륨이 그 다음에 위치해야 합니다.

다음은 부트 논리 볼륨의 미러링 예제입니다.

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

다음은 기본 스왑 논리 볼륨의 미러링 예제입니다.

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

다음은 루트 논리 볼륨의 미러링 예제입니다.

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

5. 부트, 루트 및 기본 스왑의 미러 사본에 대해 BDRA에 수록된 부팅 정보를 업데이트합니다.

```
# /usr/sbin/lvlnboot -b /dev/vg00/lvol1  
# /usr/sbin/lvlnboot -s /dev/vg00/lvol2  
# /usr/sbin/lvlnboot -r /dev/vg00/lvol3
```

6. 미러가 제대로 작성되었는지 확인합니다.

```
# lvlnboot -v
```

이 명령의 출력 결과는 다음과 같습니다.

```
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:  lvoll   on:      /dev/dsk/c4t5d0  
      /dev/dsk/c4t6d0  
Root:  lvoll   on:      /dev/dsk/c4t5d0  
      /dev/dsk/c4t6d0  
Swap:  lvoll   on:      /dev/dsk/c4t5d0  
      /dev/dsk/c4t6d0  
Dump:  lvoll   on:      /dev/dsk/c4t6d0, 0
```

클러스터 잠금 디스크 선택

잠금 디스크를 사용할 경우 아래 지침이 적용됩니다. 클러스터 잠금 디스크는 모든 클러스터 노드에 실제로 연결된 볼륨 그룹에 구성되어 있으며, 이 볼륨 그룹은 패키지에서 사용하는 데이터를 포함할 수 있습니다.

이중 클러스터 잠금 디스크를 사용하는 경우에는 클러스터 잠금 실제 볼륨의 기본 IO 시간 제한 값을 사용해야 합니다. 클러스터 잠금 실제 볼륨의 IO 시간 제한 값을 변경하면, 클러스터 재편성을 방지하기 위해 지정한 시간 내에 클러스터 노드에서 장애가 발생한 잠금 디스크를 감지하지 못 할 수도 있습니다. 현재의 IO 시간 제한 값을 기본값을 보려면 다음 명령을 실행합니다.

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

IO 시간 제한값은 “기본값” 으로 설정되어야 합니다. IO 시간 제한값을 기본값으로 설정하려면 다음 명령을 실행합니다.

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

이중 클러스터 잠금은 특정한 하드웨어 구성에서만 사용 가능합니다. 자세한 내용은 3장의 “이중 클러스터 잠금” 절을 참조하십시오.

클러스터 잠금 디스크 정보의 백업

클러스터를 구성하고 클러스터 잠금 볼륨 그룹과 실제 볼륨을 만든 다음에는 각 잠금 볼륨 그룹에서 볼륨 그룹 구성 데이터의 백업을 만들어야 합니다. 구성된 각 잠금 볼륨 그룹에서 `vgcfgbackup` 명령을 사용하고, 디스크 장애 발생 시 `vgcfgrestore` 명령으로 잠금 구성을 새 디스크로 복원할 수 있도록 이 백업 파일을 보관하십시오.

주

잠금 볼륨 그룹을 만들 때 SAM이나 HP-UX 명령 사용 여부에 상관 없이 잠금 볼륨 그룹 구성 데이터를 백업하고 복구하려면 `vgcfgbackup` 및 `vgcfgrestore` 명령을 사용해야 합니다.

비루트 사용자의 `cmviewcl` 실행 허용

MC/ServiceGuard `cmviewcl` 명령은 일반적으로 시스템에 대한 루트 액세스를 요구합니다. 그러나, `cmclodelist` 파일을 간단히 수정하여 비루트 사용자로 하여금 `cmviewcl` 명령을 실행하도록 할 수 있습니다.

HA(고가용성) 클러스터 구성 만들기 시스템 준비

특정한 비루트 사용자가 `cmviewc1` 명령을 실행할 수 있게 하려면 `/etc/cmcluster/cmclnodelist` 파일에서 호스트 이름-사용자 이름 쌍을 추가합니다. 모든 사용자가 `cmviewc1` 명령을 실행할 수 있게 하려면 `/etc/cmcluster/cmclnodelist` 파일 끝에 "+"를 추가합니다. 예를 들어, 2개의 노드로 구성된 클러스터에 대한 아래 항목에서는 `user1`과 `user2`는 `system1`에서 `cmviewc1`을 실행하고, `user3`은 `system2`에서 `cmviewc1`을 실행할 수 있습니다.

```
system1    root
system1    user1
system1    user2
system2    root
system2    user3
```

아래 예제는 비루트 사용자가 `cmviewc1` 명령을 실행할 수 있는 상황을 나타냅니다.

```
system1    root
system2    root
+
```

이름 확인 서비스의 정의

MC/ServiceGuard가 어떻게 이름 확인 서비스를 사용하는지 이해하는 것이 중요합니다. 모든 사용자 수준 ServiceGuard 명령(`cmviewc1` 포함)은 이름 확인 서비스 (`/etc/resolv.conf`에 구성되어 있음)를 사용하여 클러스터 노드 전체의 주소를 연습니다. 이름 서비스를 사용할 수 없는 경우 명령은 중단되거나 예상치 못한 네트워킹 오류 메시지가 표시될 수 있습니다. SAM에서 이름 서비스를 사용할 수 없으면 클러스터 또는 패키지 작업에도 오류가 발생합니다.

주

이러한 중단이나 오류가 발생하여 사용자가 실행한 명령이 중지된 경우에도 MC/ServiceGuard와 모든 보호된 응용 프로그램은 계속 작동합니다. 즉, MC/ServiceGuard 구성 명령과 SAM 함수만 영향을 받으며 클러스터 데몬이나 패키지 서비스는 영향을 받지 않습니다.

이러한 문제가 발생하지 않도록 하려면 DNS 또는 NIS 대신 모든 클러스터 노드의 /etc/hosts 파일을 사용할 수 있습니다. 또한, 다중 DNS 서버를 사용하거나 DNS를 MC/ServiceGuard 패키지로 구성하여 DNS의 고가용성을 확보하는 것이 바람직합니다.

다른 조회 방법을 사용할 수 없을 경우 /etc/hosts 파일을 검색하도록 /etc/nsswitch.conf 파일을 구성하면 일반적인 이름 조회 방법을 계속 사용할 수 있습니다. 이름 조회 서비스를 사용할 수 없을 때 ServiceGuard 명령과 SAM 기능이 로컬 시스템의 /etc/hosts 파일을 사용하여 이름을 확인합니다. 물론 클러스터에 있는 모든 노드의 이름과 IP 주소는 /etc/hosts 파일에 있어야 합니다.

기본 LAN 장애 또는 DNS 손실에 따른 이름 확인

LAN 장애 발생 이후, 장애가 발생한 기본 LAN에 대해 대기 LAN을 구성했을 때에도 cmrunnode 및 cmrunc1 같은 클러스터 구성 명령을 계속 사용할 수 있도록 하려면 몇 가지 특수 구성 단계가 필요합니다. 이들 단계는 또한 DNS 서비스가 손실되지 않도록 함으로써 클러스터 노드가 다른 노드와 통신을 계속할 수 있도록 합니다.

1. 클러스터의 모든 노드에서 /etc/cmcluster/cmclnodelist 파일을 편집하고 노드에 있는 다른 IP 주소 및 클러스터 노드 이름뿐만 아니라 모든 하트비트 IP 주소를 추가합니다. 예를 들면 다음과 같습니다.

```
192.2.1.1      root
197.20.10.2   root
192.20.10.3   root
15.13.1720.231 root
15.13.1720.232 root
15.13.172.233 root
hasupt01      root
hasupt02      root
hasupt03      root
```

이렇게 하면 하트비트 네트워크 또는 공용 네트워크 외에 다른 네트워크에 구성된 모든 IP 주소에 사용 권한이 부여됩니다.

2. 모든 노드에서 /etc/nsswitch.conf 파일을 편집하거나 작성하고 다음 행이 없을 경우 추가합니다.

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

HA(고가용성) 클러스터 구성 만들기 시스템 준비

"hosts:"로 시작하는 행이 있으면 이 문자열 오른쪽에 아래 텍스트가 있는지 확인하십시오.

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

DNS가 중지된 동안 또는 기본 LAN이 중지된 경우 클러스터에 있는 노드에서 호스트 이름을 IP 주소로 확인할 수 있으므로 이 단계는 중요합니다.

3. 모든 노드에서 `/etc/hosts` 파일을 편집하거나 작성하고 모든 노드의 기본 IP 주소와 노드 이름을 추가합니다.

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

커널 구성의 일관성 확인

모든 클러스터 노드의 커널 구성이 장애 조치 중에 예상되는 클러스터의 행동과 일관되는지 확인하십시오. 특히 한 클러스터 노드의 커널 매개 변수를 변경할 경우 이들 매개 변수는 동일한 패키지를 실행할 수 있는 다른 클러스터 노드에서도 변경되어야 합니다.

네트워크 시간 프로토콜 사용

클러스터의 각 노드에서 네트워크 시간 프로토콜(NTP) 서비스를 사용하는 것이 바람직합니다. 각 시스템에서 데몬 프로세스로 실행되는 NTP를 사용하면 모든 노드의 시스템 시간을 일관되게 유지하기 때문에 로그 파일의 시간 스탬프와 메시지 서비스의 행동이 일관되게 나타납니다. 일관된 시간을 유지하면 클러스터에서 실행 중인 응용 프로그램이 올바르게 동기화됩니다. NTP 서비스 데몬 `xntpd`는 클러스터 구성을 시작하기 전에 모든 노드에서 실행되고 있어야 합니다. NTP 구성 파일은 `/etc/ntp.conf`입니다.

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

클러스터 크기 변경 준비

온라인 상태에서 클러스터에 노드를 추가하려면 실행 중에 이 노드를 다른 클러스터 노드와 동일한 하드웨어 서브넷 및 동일한 잠금 디스크에 연결하십시오. 클러스터 잠금 구성을 선택할 때는 이후에 추가 클러스터 노드가 필요한지 신중하게 생각해 보십시오. 5개 이상의 노드로 구성된 클러스터는 잠금 디스크를 **사용하지 않지만** 2개의 노드로 구성된 클러스터는 클러스터 잠금을 **사용해야** 합니다. 그러므로 5개의 노드가 필요한 경우 처음부터 quorum 서버를 사용하여 구성해야 합니다.

클러스터 실행 중에 클러스터 구성에서 노드를 제거하려면 제거한 다음의 클러스터 구성이 위에서 설명한 클러스터 잠금 규칙에 위배되지 않는지 확인합니다.

클러스터 구성에서 쉽게 노드를 추가하고 제거하려면 SCSI 종료를 방해하지 않으면서 버스에서 노드를 제거할 수 있도록 인라인 터미네이터와 함께 SCSI 케이블을 사용할 수 있습니다. 인라인 SCSI 터미네이터에 대한 자세한 내용은 “문제 해결” 장의 “인라인 SCSI 터미네이터를 사용하여 온라인 하드웨어 유지 관리” 절을 참조하십시오.

온라인 상태에서 노드를 추가하고 패키지를 새 노드에서 실행할 계획이면 해당 패키지에 대한 클러스터 연결 볼륨 그룹을 새 노드로 가져왔는지 확인하십시오. 또한, MAX_CONFIGURED_PACKAGES 매개 변수가 사용하고자 하는 전체 패키지 수를 수용할 수 있을 만큼 크게 설정되어 있는지 확인하십시오.

Quorum 서버 설치

quorum 서버 소프트웨어는 클러스터 구성 중에 실행해야 하며, 클러스터가 실행될 노드 이외의 다른 시스템에 설치해야 합니다.

주

quorum 서버가 실행 중인 노드는 서비스를 제공하는 클러스터와 같은 서브넷에 있는 것이 좋습니다. 이렇게 하면 quorum 서버 작업에 영향을 미칠 수 있는 네트워크 지연을 막을 수 있습니다. 다른 서브넷을 사용하는 경우 quorum 서버 시간 제한을 일으킬 수 있는 네트워크 지연이 생길 수 있습니다. 이러한 시간 제한이 발생하지 않도록 하려면 클러스터 ASCII 파일의 QS_TIMEOUT_EXTENSION 매개 변수를 사용하여 quorum 서버 시간 제한 간격을 늘립니다.

서버 접속에 클러스터 하트비트 네트워크가 사용된 경우에는 적어도 한 개의 다른 네트워크가 하트비트 네트워크여야 두 quorum 서버와 하트비트 통신에 동시에 장애가 발생하지 않습니다.

Quorum 서버 설치

HP-UX swinstall 명령을 사용하여 quorum 서버, 제품 번호 T8467BA를 실행할 시스템에 설치합니다. Quorum 서버를 사용하는 ServiceGuard 노드에 이 제품을 설치하지 마십시오. 설치에 대한 자세한 내용은 사용하는 Quorum 서버의 *Quorum Server Release Notes*를 참조하십시오.

quorum 서버 실행 파일 qs는 /usr/sbin 디렉토리에 설치됩니다. 설치가 끝나면 특정 호스트 시스템에서 quorum 서비스를 얻을 수 있도록 QS가 실행할 서버에 인증 파일을 만들어야 합니다. 이 파일은 **반드시** /etc/cmcluster/qs_authfile에 있어야 합니다. 이 quorum 서버에서 클러스터 서비스에 액세스할 모든 클러스터 노드 이름을 파일에 추가합니다. 아래 예제에서처럼 노드마다 한 줄을 사용합니다.

```
ftsys9.localdomain.com  
ftsys10.localdomain.com
```

모든 노드에 액세스를 허용하려면 덧셈 기호(+) 다음에 **CR**을 사용합니다.

Quorum 서버 실행

quorum 서버는 다음 클러스터 작업 중에 실행해야 합니다.

- `cmquerycl` 명령 실행 시
- `cmapplyconf` 명령 실행 시
- 클러스터 재형성 시

기본적으로 quorum 서버 런타임 메시지는 `stdout` 및 `stderr`로 이동됩니다. `/var/adm/qs` 디렉토리를 만든 다음 `stdout` 및 `stderr`을 이 디렉토리의 파일로 재지정하는 것이 좋습니다(예: `/var/adm/qs/qs.log`).

quorum 서버를 실행할 루트 사용 권한이 있어야 합니다. quorum 서버가 설치된 시스템을 언제든지 다시 시작하고 다시 부팅할 수 있도록 quorum 서버를 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 quorum 서버를 시작합니다.

```
# init q
```

명령이 완료되면 메시지가 나타납니다.

`qs.log` 파일을 검사하여 quorum 서버가 실행 중인지 확인합니다.

```
# cat /var/adm/qs/qs.log
```

로그에는 quorum 서버가 시작되었음을 나타내는 다음과 같은 항목이 포함되어 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
```

quorum 서버 작동 방법에 대해서는 3장을 참조하고 `cmquerycl` 명령을 사용하여 클러스터 ASCII 파일에 quorum 서버를 지정하는 방법에 대해서는 이 장의 “클러스터 구성” 절에 있는 “quorum 서버 지정” 절을 참조하십시오.

MC/ServiceGuard 설치

MC/ServiceGuard를 설치하면 Software Distributor를 통해 소프트웨어가 업데이트됩니다. 이미 HP-UX 11.0이 설치되어 있다고 가정합니다.

각 노드에 대해 다음 단계를 따르십시오.

1. 테이프 드라이브나 CD ROM 드라이버에 설치 미디어를 넣습니다.
2. `swinstall` 명령을 사용하여 Software Distributor를 실행합니다.
3. 올바른 입력 장치를 지정합니다.
4. 표시된 목록에서 다음 번들을 선택합니다.

B3935DA

MC/ServiceGuard

5. 번들을 선택한 후 OK를 선택합니다. 그러면 해당 소프트웨어가 로드됩니다.

`swinstall` 명령 실행과 새 사용자 계정 작성에 대한 자세한 내용은 HP-UX 안내서 시스템 및 작업 그룹 관리를 참조하십시오.

LVM을 사용하여 저장 하부 구조 만들기

클러스터를 구성하기 전에 적절한 논리 볼륨 하부 구조를 만들어 여러 노드의 데이터에 액세스할 수 있게 하십시오. 논리 볼륨 관리자(LVM), VERITAS Cluster Volume Manager(CVM) 또는 VERITAS Volume Manager(VxVM)를 사용할 수 있습니다. 필요에 따라 이들 볼륨을 함께 사용할 수 있습니다. LVM 및 VxVM 구성은 클러스터 구성 전에, CVM 구성은 클러스터 구성 후에 수행합니다.

이 절에서는 LVM을 사용한 저장 구성에 대해 설명합니다. 다음 작업의 단계는 별도로 설명합니다.

- 미러링된 개별 디스크의 볼륨 그룹 만들기
- PV 링크를 사용하여 디스크 어레이의 볼륨 그룹 만들기
- 다른 노드에 대한 볼륨 그룹 배포

Event Monitoring Service HA Disk Monitor는 LVM 디스크의 상태를 모니터링하는 기능을 제공합니다. 미러링된 디스크에 이 모니터링 기능을 사용하려면 실제 볼륨 그룹에 해당 디스크를 구성해야 합니다. 자세한 내용은 *Using High Availability Monitors* 설명서를 참조하십시오.

미러링된 개별 데이터 디스크의 볼륨 그룹 만들기

이 절에서 설명하는 과정은 개별 디스크의 미러링을 위한 **실제 볼륨 그룹(Physical Volume Group)**을 사용하여 각 논리 볼륨을 다른 입출력 버스의 디스크로 미러링합니다. 이런 종류의 미러링 방법을 **PVG 완전 미러링(PVG-strict mirroring)**이라고 합니다. 디스크가 미리 사본으로 사용되는 것과 동일한 방법으로 이미 디스크 하드웨어가 구성되어 있고, 이 하드웨어는 다른(기본) 사본에 사용되지 않는 버스의 각 노드에 연결되어 있다고 가정합니다.

LVM 사용에 대한 자세한 내용은 HP-UX 설명서 **시스템 및 작업 그룹 관리**를 참조하십시오.

SAM을 사용하여 볼륨 그룹 및 논리 볼륨 만들기

SAM을 사용하여 HA 패키지에 필요한 볼륨 그룹 및 논리 볼륨 구조를 준비할 수 있습니다. SAM에서 “Disks and File Systems Area”를 선택합니다. 그런 다음 패키지에 사용하는 각 볼륨 그룹 및 파일 시스템에 대해 다음 단계를 수행하십시오.

HA(고가용성) 클러스터 구성 만들기 LVM을 사용하여 저장 하부 구조 만들기

1. Volume Groups 하위 영역을 선택합니다.
2. Actions 메뉴에서 Create 또는 Extend를 선택합니다.
3. 사용 가능한 디스크 목록의 볼륨 그룹에서 사용할 첫 번째 실제 디스크를 선택합니다.
4. 볼륨 그룹 이름(예: vgdatabase)을 입력합니다.
5. Create Volume Group 또는 Extend Volume Group을 선택합니다.
6. Add New Logical Volumes를 선택합니다.
7. 논리 볼륨을 볼륨 그룹에 추가할 때 PVG 완전 할당 방식으로 미리링된 논리 볼륨을 만들고 있는지 확인합니다.
8. 볼륨 그룹에 마운트될 파일 시스템(예: /mnt1)을 지정합니다.
9. 추가 볼륨 그룹, 논리 볼륨 및 파일 시스템에 대해 이 과정을 반복하십시오.
“볼륨 그룹 비활성화” 절로 건너뛰십시오.

LVM을 사용하여 볼륨 그룹 및 논리 볼륨 만들기

볼륨 그룹을 설정하지 않은 경우 다음 절의 과정을 수행하십시오. LVM을 이미 구성했다면 “클러스터 구성” 절로 건너뛰십시오.

볼륨 그룹을 위한 디스크 선택

두 노드의 디스크 목록을 얻어 두 노드의 동일한 디스크에 사용되는 장치 파일을 찾습니다. 각 노드에 대해 다음 명령을 사용하여 각 시스템에서 식별된 사용 가능한 디스크 목록을 표시하십시오.

```
# lsssf /dev/dsk/*
```

다음 예제에서 사용되는 /dev/rdisk/c1t2d0과 /dev/rdisk/c0t2d0은 *ftsys9* 및 *ftsys10* 모두에서 동일한 디스크의 장치 이름으로 사용됩니다. 장치 파일 이름이 다른 노드에서 다르게 사용될 경우 해당 이름을 잘 기록하십시오.

실제 볼륨 만들기 구성 노드(ftsys9)에서 `pvcreate` 명령을 사용하여 디스크를 실제 볼륨으로 정의할 수 있습니다. 이 작업은 구성 노드에서만 수행하면 됩니다. 다음 명령을 사용하여 예제 구성을 위한 두 개의 실제 볼륨을 만듭니다.


```
# 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 만들 수 있습니다. 나중에 다른 노드에 동일한 볼륨 그룹을 만듭니다.

1. 먼저 vgdatabase를 위한 그룹 디렉토리를 만듭니다.

```
# mkdir /dev/vgdatabase
```

2. 그런 다음 아래와 같이 디렉토리 /dev/vgdatabase에 *group*이라는 이름의 제어 파일을 만듭니다.

```
# mknod /dev/vgdatabase/group c 64 0xhh0000
```

중심 숫자는 항상 64이고 16진 하위 숫자는 다음 형식을 갖습니다.

```
0xhh0000
```

여기에서 *hh*는 작성 중인 볼륨 그룹에서 고유해야 합니다. 이미 구성된 볼륨 그룹 다음에는 시스템에서 사용 가능한 그 다음 16진수를 사용합니다. 다음 명령을 사용하여 기존의 볼륨 그룹 목록을 표시합니다.

```
# ls -l /dev/*/group
```

3. 다음 명령을 사용하여 볼륨 그룹을 만든 다음 실제 볼륨을 추가합니다.

```
# vgcreate -g bus0 /dev/vgdatabase /dev/dsk/c1t2d0  
# vgextend -g bus1 /dev/vgdatabase /dev/dsk/c0t2d0
```

첫째 명령은 볼륨 그룹을 만들고 *bus0*라는 이름의 실제 볼륨 그룹에 실제 볼륨을 추가합니다. 둘째 명령은 볼륨 그룹에 둘째 드라이브를 추가하고 *bus1*이라는 이름의 다른 실제 볼륨 그룹에 추가합니다. 실제 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(예제는 /dev/vgdatabase에 대한 논리 볼륨)을 만듭니다.

```
# lvcreate -L 120 -m 1 -s g /dev/vgdatabase
```

이 명령은 *lv011*이라는 이름의 미러링된 120MB의 볼륨을 만듭니다. 명령에서 이름을 지정하지 않았기 때문에 기본적으로 이 이름이 사용됩니다. *-s g* 옵션은 PVG 완전 미러링을 의미하기 때문에 데이터의 미러 사본이 다른 실제 볼륨 그룹에 위치하게 됩니다.

주

RAID 1 또는 RAID 5 모드에서 디스크 어레이를 사용하는 경우 `-m 1` 및 `-s g` 옵션을 사용하지 마십시오.

파일 시스템 만들기

설치 프로그램이 파일 시스템을 사용하면 나중에 파일 시스템을 만드십시오. 다음 명령을 사용하여 위에서 만든 논리 볼륨에 마운트할 파일 시스템을 만드십시오.

1. 새로 만든 논리 볼륨에 파일 시스템을 만듭니다.

```
# newfs -F vxfs /dev/vgdatabase/rlvol1
```

논리 볼륨에 대한 원시 장치 파일의 사용에 주목합니다.

2. 디스크를 마운트할 디렉토리를 만듭니다.

```
# mkdir /mnt1
```

3. 디스크를 마운트하여 작업이 제대로 수행되었는지 확인합니다.

```
# mount /dev/vgdatabase/lvol1 /mnt1
```

mount 명령이 논리 볼륨에 대해 블록 장치 파일을 사용한다는 것에 주목합니다.

4. 구성을 확인합니다.

```
# vgdisplay -v /dev/vgdatabase
```

PV 링크를 사용하여 디스크 어레이의 볼륨 그룹 만들기

HP의 HA 디스크 어레이에서 대용량 저장 장치를 사용하는 볼륨 그룹을 구성할 경우 각 노드의 중복 I/O 채널을 어레이의 별도 포트에 연결하여 사용해야 합니다. 그런 다음 어레이에서 정의한 LUN 또는 논리 디스크로의 대체 링크(PV 링크라고도 함)를 정의할 수 있습니다. SAM에서는 구성하려는 디스크 어레이 종류를 선택한 다음 메뉴에 따라 대체 링크를 정의하십시오.

다음 예제는 LVM 명령을 사용하여 대체 링크를 구성하는 방법을 보여줍니다. 이 예제에서 디스크 구성이 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 */
```

디스크 어레이가 구성되었고 `ioscan` 명령을 실행하면 같은 LUN(논리 디스크)에 대해 다음 두 장치 파일이 나타난다고 가정합니다.

```
/dev/dsk/c0t15d0
/dev/dsk/c1t3d0
```

다음 단계에 따라 이 논리 디스크의 볼륨 그룹을 구성하십시오.

1. 먼저 `vgdatabase`를 위한 그룹 디렉토리를 만듭니다.

```
# mkdir /dev/vgdatabase
```

2. 그런 다음 아래와 같이 디렉토리 `/dev/vgdatabase`에 `group`이라는 이름의 제어 파일을 만듭니다.

```
# mknod /dev/vgdatabase/group c 64 0xhh0000
```

중심 숫자는 항상 64이고 16진 하위 숫자는 다음 형식을 갖습니다.

```
0xhh0000
```

여기에서 `hh`는 작성 중인 볼륨 그룹에서 고유해야 합니다. 이미 구성된 볼륨 그룹 다음에는 시스템에서 사용 가능한 그 다음 16진수를 사용합니다. 다음 명령을 사용하여 기존의 그룹 파일 목록을 표시합니다.

```
# ls -l /dev/*/group
```

3. LUN과 관련된 장치 파일 중 하나에서 `pvcreate` 명령을 사용하여 실제 볼륨으로 LUN을 LVM에 정의합니다.

```
# pvcreate /dev/rdsk/c0t15d0
```

이 작업은 LUN의 장치 파일 이름 중 하나만 사용하여 수행할 수 있습니다.

HA(고가용성) 클러스터 구성 만들기 LVM을 사용하여 저장 하부 구조 만들기

4. 다음 명령을 사용하여 볼륨 그룹 자체를 만듭니다.

```
# vgcreate /dev/vgdatabase /dev/dsk/c0t15d0
# vgextend /dev/vgdatabase /dev/dsk/c1t3d0
```

이제 `vgdisplay -v` 명령을 사용하여 주 링크와 대체 링크를 볼 수 있습니다. LVM은 이제 `/dev/dsk/c0t15d0`에 기록된 I/O 채널을 디스크에 대한 주 링크로 인식합니다. 주 링크에 장애가 발생할 경우 LVM은 `/dev/dsk/c1t3d0`에 기록된 대체 I/O 채널로 자동 전환합니다.

논리 볼륨을 만들려면 앞의 “논리 볼륨 만들기” 절에 설명한 단계를 따르십시오.

다른 노드에 대한 볼륨 그룹 배포

클러스터 데이터의 볼륨 그룹을 만든 다음, 볼륨 그룹을 활성화할 클러스터 노드로 배포해야 합니다. 클러스터 잠금 볼륨 그룹은 모든 노드에서 사용 가능해야 합니다.

볼륨 그룹 비활성화

볼륨 그룹을 작성할 때 볼륨 그룹은 구성 노드(예: `ftsys9`)에서 활성화되어 있습니다. 다른 노드에서 사용할 수 있도록 볼륨 그룹을 설치하기 전에 먼저 볼륨 그룹에 있는 파일 시스템을 언마운트한 다음 비활성화해야 합니다. 실행 시간 중에 패키지 제어 스크립트를 통해 볼륨 그룹이 활성화되고 파일 시스템이 마운트됩니다.

이전 절의 예제를 다시 진행하며 이제 `ftsys9`에서 다음을 수행하십시오.

```
# umount /mnt1
# vgchange -a n /dev/vgdatabase
```

SAM을 사용한 볼륨 그룹 배포

SAM에서 디스크 및 파일 시스템 영역을 선택한 다음 볼륨 그룹을 선택합니다. 하나 이상의 추가 노드에 배포할 볼륨 그룹을 선택합니다. 볼륨 그룹이 위치할 각 노드의 이름을 입력한 다음 “Add”를 선택합니다. 목록이 완성되면 OK를 누릅니다. 그러면 SAM은 다른 노드에서 사용할 수 있도록 볼륨 그룹을 자동으로 구성합니다.

볼륨 그룹을 배포한 다음 각 노드의 `/etc/lvm/vg` 파일을 확인하여 실제 볼륨 그룹이 해당 노드의 올바른 실제 볼륨 이름을 포함하는지 살펴보세요.

LVM 명령을 사용한 볼륨 그룹 배포

다음 명령을 사용하여 다른 클러스터 노드에 동일한 볼륨 그룹을 설정할 수 있습니다. 이 예제의 명령은 *ftsys9*에서 사용했던 것과 동일한 실제 볼륨을 포함할 *ftsys10*에 새로운 볼륨 그룹을 설정합니다. 볼륨 그룹의 패키지가 실행될 수 있는 각 노드에 대해 개별적으로 동일한 과정을 수행해야 합니다.

*ftsys10*에 볼륨 그룹을 설정하려면 다음 단계를 수행하십시오.

1. *ftsys9*에서 볼륨 그룹의 매핑을 지정된 파일로 복사합니다.

```
# vgexport -p -s -m /tmp/vgdatabase.map /dev/vgdatabase
```
2. *ftsys9*에서 *ftsys10*으로 맵 파일을 복사합니다.

```
# rcp /tmp/vgdatabase.map ftsys10:/tmp/vgdatabase.map
```
3. *ftsys10*에서 볼륨 그룹 디렉토리를 만듭니다.

```
# mkdir /dev/vgdatabase
```
4. 또한, *ftsys10*에서 다음과 같이 디렉토리 */dev/vgdatabase*에 *group*이라는 이름의 제어 파일을 만듭니다.

```
# mknod /dev/vgdatabase/group c 64 0xhh0000
```

중심 숫자는 항상 64이고 16진 하위 숫자는 다음 형식을 갖습니다.

```
0xhh0000
```

여기에서 *hh*는 작성 중인 볼륨 그룹에서 고유해야 합니다. 가능하면 *ftsys9*와 동일한 숫자를 사용하십시오. 다음 명령을 사용하여 기존의 볼륨 그룹 목록을 표시합니다.

```
# ls -l /dev/*/group
```

5. 노드 *ftsys9*의 해당 맵 파일을 사용하여 볼륨 그룹 데이터를 가져옵니다. *ftsys10*노드에서 다음을 입력하십시오.

```
# vgimport -s -m /tmp/vgdatabase.map /dev/vgdatabase
```

*ftsys10*의 디스크 장치 이름은 *ftsys9*의 디스크 장치 이름과 다를 수 있습니다. 실제 볼륨 이름이 클러스터 전체에서 올바른지 확인해야 합니다.

HA(고가용성) 클러스터 구성 만들기 LVM을 사용하여 저장 하부 구조 만들기

6. 실제 볼륨 그룹의 미러링된 개별 디스크를 사용하는 경우 `/etc/lvm/pvg` 파일에서 각 실제 볼륨 그룹에 `ftsys10`의 올바른 실제 볼륨 이름이 포함되어 있는지 확인하십시오.

주

PVG 완전 미러링을 사용하면 실제 볼륨 그룹 구성이 구성 노드의 `/etc/lvm/pvg` 파일에 기록됩니다. 이 파일은 미러링의 기본이 되는 실제 볼륨 그룹을 정의하고 각 PVG에 속하는 실제 볼륨을 표시합니다. 각 클러스터 노드에서 `/etc/lvm/pvg` 파일에는 **해당 노드에서 사용한 이름과 동일한 PVG** 디스크의 올바른 실제 볼륨 이름이 포함되어야 합니다. 동일한 디스크의 실제 볼륨 이름이 노드마다 다를 수 있습니다. 다른 노드에 볼륨 그룹을 배포한 다음에는 각 노드의 `/etc/lvm/pvg` 파일이 해당 노드에 있는 모든 실제 볼륨 그룹의 내용을 올바르게 반영하는지 확인해야 합니다. 자세한 내용은 다음에 설명하는 “실제 볼륨 그룹 파일의 일관성 유지”를 참조하십시오.

7. `ftsys9`에서 볼륨 그룹이 비활성화되었는지 확인합니다. 그런 다음 `ftsys10`에서 볼륨 그룹을 사용할 수 있도록 합니다.

```
# vgchange -a y /dev/vgdatabase
```

8. 디스크를 마운트할 디렉토리를 만듭니다.

```
# mkdir /mnt1
```

9. 마운트한 다음 `ftsys10`의 볼륨 그룹을 검사합니다.

```
# mount /dev/vgdatabase/lvol1 /mnt1
```

10. `ftsys10`에서 볼륨 그룹을 언마운트 합니다.

```
# umount /mnt1
```

11. `ftsys10`에서 볼륨 그룹을 비활성화합니다.

```
# vgchange -a n /dev/vgdatabase
```

실제 볼륨 그룹 파일의 일관성 유지

디스크 구성에서 미러링된 개별 디스크의 실제 볼륨 그룹을 사용하지 않을 경우 다음 절로 건너뛰십시오.

다른 볼륨 그룹은 MC/ServiceGuard 클러스터 내의 다른 노드 하위 집합에 의해 활성화될 수 있습니다. 또한, 노드마다 특정 디스크의 실제 볼륨 이름은 다를 수 있습니다. 이러한 이유로 각 노드가 모든 클러스터 인식 디스크와 독립 디스크(클러스터를 인식하지 않는 디스크)를 완전하고 일관되게 파악할 수 있도록 모든 노드에 있는 `/etc/lvm/pvg` 파일을 신중하게 병합해야 합니다. 파일을 보다 쉽게 병합하려면 볼륨 그룹 계획 워크시트에 실제 볼륨 그룹 이름을 잘 기록해 두어야 합니다(“계획” 장 참조).

다음 단계를 따라 구성 노드(*ftsys9*)와 볼륨 그룹을 가져오는 새 노드(*ftsys10*) 사이에서 파일을 병합하십시오.

1. *ftsys9*에서 *ftsys10*의 `/etc/lvm/pvg.new`로 `/etc/lvm/pvg`를 복사합니다.
2. *ftsys10*에 없는 볼륨 그룹이 `/etc/lvm/pvg.new`에 있는 경우 `/etc/lvm/pvg.new`에서 해당 볼륨 그룹의 모든 항목을 제거합니다.
3. *ftsys10*의 `/etc/lvm/pvg`가 `/etc/lvm/pvg.new`에 없는 볼륨 그룹 항목을 포함할 경우 `/etc/lvm/pvg.new`에 볼륨 그룹의 모든 PVG 항목을 복사합니다.
4. *ftsys10*의 올바른 이름을 반영하도록 `/etc/lvm/pvg.new`의 실제 볼륨 이름을 수정합니다.
5. *ftsys10*에서 `/etc/lvm/pvg`를 `/etc/lvm/pvg.old`에 복사하여 백업을 작성합니다. `/etc/lvm/pvg.new`를 *ftsys10*의 `/etc/lvm/pvg`에 복사합니다.

추가 볼륨 그룹 만들기

지금까지 MC/ServiceGuard에서 사용할 볼륨 그룹과 논리 볼륨을 만드는 일반적인 방법을 설명했습니다. 작성해야 하는 수만큼의 볼륨 그룹에 해당 과정을 반복합니다. 이 때, 볼륨 그룹 이름, 논리 볼륨 이름, 실제 볼륨 이름은 해당 이름으로 대체하십시오. 특히 디스크 장치 이름에 주의하십시오. 예를 들어, 한 노드의 `/dev/dsk/c0t2d0`이 다른 노드에서도 `/dev/dsk/c0t2d0`이 될 수 없습니다.

VxVM을 사용하여 저장 하부 구조 만들기

클러스터를 구성하기 전에 적절한 논리 볼륨 하부 구조를 만들어 여러 노드의 데이터에 액세스할 수 있게 하십시오. 논리 볼륨 관리자(LVM), VERITAS Cluster Volume Manager(CVM) 또는 VERITAS Volume Manager(VxVM)를 사용할 수 있습니다. 필요에 따라 이들 볼륨을 함께 사용할 수 있습니다. LVM 및 VxVM 구성은 클러스터 구성 전에, CVM 구성은 클러스터 구성 후에 수행합니다.

LVM에서 VxVM 저장으로 변환하는 방법에 대해서는 부록 G를 참조하십시오.

이 절에서는 VERITAS 볼륨 관리자(VxVM) 명령을 사용하여 새로운 저장을 구성하는 방법에 대해 설명합니다. 루트 디스크 그룹을 만들면(다음 절에서 설명) VxVM 명령이나 Storage Administrator GUI인 vmsa를 사용하여 구성 작업을 수행할 수 있습니다. vmsa를 사용 중이면 GUI를 실행하기 전에 Storage Administrator 서버가 실행 중인지 확인하십시오. 자세한 내용은 *VERITAS Volume Manager for HP-UX Release Notes*에 설명되어 있습니다. 자세한 내용은 *VERITAS VMSA Administrator's Guide*를 참조하십시오. 명령을 사용할 경우에는 VxVM 온라인 도움말을 참조하십시오.

VERITAS 볼륨 관리자 초기화

처음으로 디스크 그룹을 만들 경우에는 볼륨 관리자를 초기화해야 합니다. 이 작업은 적어도 하나의 디스크가 들어 있는 *rootdg* 디스크 그룹을 만들어 수행합니다. 각 노드에 VxVM을 설치한 후에 즉시 다음 명령을 **한 번만** 수행합니다.

```
# vxinstall
```

이 명령은 VxVM 초기화 순서를 안내하는 메뉴 구동 프로그램을 표시합니다. 기본 메뉴에서 “Custom” 옵션을 선택하고 *rootdg*에 포함시킬 디스크를 지정합니다.

중요

VERITAS 볼륨 관리자의 *rootdg*는 HP-UX 루트 디스크와 같지 않으므로 HP-UX 루트 파일 시스템(/)에 사용할 수 없습니다. 또한, 공유 저장 장소에 *rootdg*를 사용할 수 없습니다. 그러나, *rootdg*는 다른 로컬 파일 시스템(예: /export/home)에 사용할 수 있으므로 삭제할 필요가 없습니다.

각 노드에 한 번씩만 루트 디스크 그룹을 만들어야 합니다.

LVM에서 VxVM으로 디스크 변환

vxvmconvert (1m) 유틸리티를 사용하여 LVM 볼륨 그룹을 VxVM 디스크 그룹으로 변환할 수 있습니다. 디스크 그룹으로 변환하기 전에 볼륨 그룹을 비활성화해야 합니다. 즉, 볼륨 그룹을 사용하는 모든 패키지를 중지해야 합니다. *VERITAS Volume Manager Migration Guide*에 나타난 변환 절차대로 수행하십시오. 시작하기 전에 *vgcfgbackup* 명령을 사용하여 각 볼륨 그룹의 구성 백업을 작성하고 볼륨 그룹의 데이터도 백업해두십시오. 변환에 대한 자세한 내용은 부록 H, “LVM에서 VxVM 데이터 저장으로 변환”을 참조하십시오.

VxVM 디스크 초기화

VxVM 디스크 그룹에서 사용할 실제 디스크를 초기화해야 합니다. 디스크를 초기화하려면 클러스터의 한 노드로 로그인한 다음 *vxdiskadm* 프로그램을 사용하여 여러 디스크를 초기화하거나 다음 예제와 같이 *vxdisksetup* 명령을 사용하여 한 번에 한 개의 디스크를 초기화합니다.

```
# /usr/lib/vxvm/bin/vxdisksetup -i c0t3d2
```

LVM에서 이전에 사용한 디스크 초기화

LVM에서 이전에 실제 디스크를 사용했다면 *pvremove* 명령을 사용하여 볼륨 그룹에 있는 모든 디스크에서 LVM 헤더 데이터를 삭제해야 합니다. 또한, LVM 디스크가 이전에 클러스터에 사용되었다면 먼저 *pvcreate -f* 명령으로 디스크를 다시 초기화하여 디스크에서 클러스터 ID를 제거해야 합니다.

주

이들 명령은 디스크와 이 디스크의 데이터를 LVM에서 사용할 수 없도록 하고 VxVM에서 초기화할 수 있도록 합니다. 이 명령은 이전에 LVM에서 디스크를 사용했지만 현재는 데이터를 여기에 저장하지 않을 때만 사용해야 합니다.

다음 예제에서처럼 디스크에서 LVM 헤더 데이터를 제거할 수 있습니다. 이 예제에서

HA(고가용성) 클러스터 구성 만들기 VxVM을 사용하여 저장 하부 구조 만들기

디스크에 있는 데이터는 모두 지워집니다.

```
# pvcreate -f /dev/rdisk/c0t3d2
# pvremove /dev/rdisk/c0t3d2
```

그런 다음, vxdiskadm 프로그램을 사용하여 VxVM의 여러 디스크를 초기화하거나 아래 예제에서처럼 vxdisksetup 명령을 사용하여 한 번에 하나의 디스크를 초기화하십시오.

```
# /usr/lib/vxvm/bin/vxdisksetup -i c0t3d2
```

디스크 그룹 만들기

vxdiskadm 명령을 사용하거나 아래 예제에 나타난 대로 vxdg 명령을 사용하여 디스크 그룹을 만듭니다.

```
# vxdg init logdata c0t3d2
```

다음 명령을 사용하여 구성을 확인하십시오.

```
# vxdg list
```

NAME	STATE	ID
rootdg	enabled	971995699.1025.node1
logdata	enabled	972078742.1084.node1

블록 만들기

다음 예제에서처럼 vxassist 명령을 사용하여 실제 블록을 만드십시오.

```
# vxassist -g logdata make log_files 1024m
```

이 명령은 디스크 그룹 logdata에 1024MB의 log_files 블록을 만듭니다. 이 블록은 블록 장치 파일 /dev/vx/dsk/logdata/log_files 또는 원시(문자) 장치 파일 /dev/vx/rdsk/logdata/log_files로 참조됩니다. 다음 명령을 사용하여 구성을 확인하십시오.

```
# vxprint -g logdata
```

이 명령의 출력 결과는 아래 예제에 표시되어 있습니다.

TY	NAME	ASSOC	KSTATE	LENGTH	PLOFFS	STATE	TUTILO
PUTILO							
v	logdata	fsgen	ENABLED	1024000		ACTIVE	
pl	logdata-01	system	ENABLED	1024000		ACTIVE	

주

VxVM을 사용하여 미러링된 저장 및 다중 경로 저장을 만드는 특정 명령은 *VERITAS Volume Manager Guide*에 설명되어 있습니다.

파일 시스템 만들기

설치 프로그램이 파일 시스템을 사용하면 나중에 파일 시스템을 만드십시오. 다음 명령을 사용하여 위에서 만든 논리 볼륨에 마운트할 파일 시스템을 만드십시오.

1. 새로 만든 논리 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 나타났는지 확인한 다음 파일 시스템을 언마운트하십시오.

```
# umount /logs
```

디스크 그룹 내보내기

ServiceGuard 패키지에서 사용할 디스크 그룹을 만든 다음 각 디스크 그룹에 다음 명령을 사용하여 패키지 제어 스크립트에서 여러 클러스터 노드에 디스크 그룹을 가져올 수 있도록 합니다.

```
# vxdg deport <DiskGroupName>
```

여기서, <DiskGroupName>은 제어 스크립트에서 활성화할 디스크 그룹 이름입니다.

디스크 그룹이 모두 내보내지면 모든 클러스터 노드에서 다음 명령을 실행하여 디스크 그룹에 액세스할 수 있도록 해야 합니다.

```
# vxdctl enable
```

디스크 그룹 다시 가져오기

디스크 그룹을 내보낸 경우에는 패키지 제어 스크립트나 `vx dg import` 명령을 사용하여 이 디스크 그룹을 다시 가져와야 노드에서 사용할 수 있습니다. 유지 보수 또는 기타 목적을 위해 디스크 그룹을 수동으로 가져와야 할 경우에는 디스크 그룹을 가져오고, 해당 논리 볼륨을 모두 시작하고 아래 예제에서처럼 파일 시스템을 마운트하십시오.

```
# vx dg 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 **않**으며, 패키지 ASCII 구성 파일에 입력되지 않습니다.

시스템 재부팅 시 지우고 가져오기

시스템 재부팅 시 클러스터 RC 스크립트는 `noautoimport` 플래그가 설정되어 있고, 다른 실행 노드에서 이들 디스크를 현재 가져오지 않았다고 가정하고 이전에 시스템에서 가져온 모든 디스크에 `vx disk clearimport`를 수행합니다. `clearimport`는 디스크 그룹에 있는 호스트 ID를 지워 패키지가 노드 간에 이동될 때 디스크 그룹에 연결된 모든 노드에서 이 호스트 ID를 가져오도록 합니다.

재부팅 시 지우고 가져오기를 사용하면 ServiceGuard에서 노드 장애(예: 전원 장애 시 시스템 충돌)를 정리할 수 있습니다. 장애 시 가져온 디스크에 노드 ID가 그대로 기록되어 있으면 노드를 재부팅하기 전에 또는 다른 노드에서 패키지 제어 스크립트로 이들 디스크를 가져오기 전에 이 ID를 지워야 합니다.

ServiceGuard가 클러스터에 구성되어 있는지 여부에 관계 없이 ServiceGuard를 설치한 모든 시스템에 `noautoimport` 플래그가 설정되어 이전에 가져온 디스크에 대해 `clearimport`가 수행됩니다.

클러스터 구성

이 절에서는 기본 클러스터 구성을 정의하는 방법을 설명합니다. SAM에서 이 작업을 수행하려면 다음 절을 읽으십시오. MC/ServiceGuard 명령을 사용하려면 “MC/ServiceGuard 명령을 사용한 클러스터 구성” 절로 건너뛰십시오.

SAM을 사용한 클러스터 구성

고가용성 클러스터를 구성하려면 구성 노드(ftsys9)에서 다음 단계를 따르십시오.

1. SAM에서 Clusters를 선택한 다음, >High Availability Clusters를 선택합니다.
2. Cluster Configuration을 선택합니다. SAM은 클러스터 구성 화면을 표시합니다. 아직 구성된 클러스터가 없으면 목록 영역은 빈 상태로 표시됩니다. 하나 이상의 HA(고가용성) 클러스터가 로컬 네트워크에 이미 구성되어 있을 경우 목록에 해당 클러스터가 나타납니다.
3. Actions 메뉴를 선택한 다음 Create Cluster Configuration을 선택합니다. 그러면 단계 메뉴가 표시됩니다.
4. 필요한 각 단계를 순서대로 선택한 다음 대화 상자에서 필요한 정보를 완성하거나 기본값을 그대로 사용합니다. 각 단계에 대한 자세한 내용을 보려면 Help를 선택하십시오.
5. 모든 단계가 완료되면 Step Menu 화면에서 OK를 선택합니다. 이렇게 하면 클러스터 구성 파일이 작성되고 해당 파일이 클러스터의 모든 노드에 복사됩니다. 파일 복사가 완료되면 Cluster Configuration 화면으로 되돌아갑니다.
6. Cluster Configuration 화면에서 나와서 High Availability Clusters 메뉴로 되돌아갑니다.

주

SAM은 이진 클러스터 구성 파일을 작성 및 배포할 뿐 아니라 /etc/cmcluster/cmclconfig.ascii라는 이름의 ASCII 클러스터 구성 파일을 작성합니다. 이 파일은 SAM에 입력된 선택 가능한 레코드 형태로 제공됩니다.

“자동 시작 기능 설정” 절로 건너뛰십시오.

MC/ServiceGuard 명령을 사용한 클러스터 구성

`cmquerycl` 명령을 사용하여 클러스터에 포함되거나 클러스터 구성 템플릿을 생성할 노드 세트를 지정하십시오. 다음은 이 명령을 실행한 예제입니다.

```
# cmquerycl -v -C /etc/cmcluster/lpcluster.config -n lp001 -n lp002
```

이 예제에서는 기본 클러스터 구성 디렉토리 `/etc/cmcluster`에 ASCII 템플릿 파일을 만듭니다. 이 ASCII 파일에는 부분적으로 두 개의 노드 `ftsyst9` 및 `ftsyst10`에 있는 클러스터 구성 요소의 이름과 특성을 수록합니다. 필요할 경우 수록된 클러스터 특성을 편집하여 원하는 클러스터를 정의하십시오. 다음 예제처럼 가능한 모든 네트워크에서 하트비트를 전송하도록 파일을 편집하는 것이 좋습니다.

주

클러스터에 노드, 네트워크 또는 디스크가 많이 연결된 크거나 복잡한 구성에서는 `cmquerycl` 명령이 완료되려면 몇 분이 소요될 수 있습니다. 보다 신속히 구성하려면 `-k` 및 `-w` 옵션을 사용하여 명령이 선택된 정보만 반환하도록 할 수 있습니다.

`-k`는 일부 디스크 진단 과정을 생략하여 잠재적인 클러스터 잠금 볼륨 그룹과 잠금 실제 볼륨에 대한 정보를 반환하지 않습니다.

`-w local`을 사용하면 로컬 네트워크 진단을 지정할 수 있으며, 이 경우 LAN 연결은 각 노드 내의 인터페이스 사이에서만 검사됩니다.

`-w full`을 사용하면 전체 네트워크 진단을 지정할 수 있으며, 이 경우 실제 연결은 클러스터의 모든 노드에 있는 모든 LAN 인터페이스 사이에서 검사됩니다.

자세한 내용은 `cmquerycl (1m)`의 온라인 도움말을 참조하십시오.

클러스터 구성 템플릿 파일

다음은 -w full 옵션과 함께 cmquerycl 명령을 사용하여 생성한 ASCII 구성 파일의 예제입니다.

```
# *****  
# ***** HIGH AVAILABILITY CLUSTER CONFIGURATION FILE *****  
# ***** For complete details about cluster parameters and how to *****  
# ***** set them, consult the cmquerycl(1m) manpage or your manual. *****  
# *****  
  
# Enter a name for this cluster. This name will be used to identify the  
# cluster when viewing or manipulating it.  
  
CLUSTER_NAME          cluster1  
  
# Definition of nodes in the cluster.  
# Repeat node definitions as necessary for additional nodes.  
  
NODE_NAME              ftsys9  
NETWORK_INTERFACE     lan0  
HEARTBEAT_IP          15.130.1710.32  
NETWORK_INTERFACE     lan3  
HEARTBEAT_IP          192.60.70.3  
NETWORK_INTERFACE     lan4  
NETWORK_INTERFACE     lan1  
HEARTBEAT_IP          192.6.1430.10  
FIRST_CLUSTER_LOCK_PV /dev/dsk/c1t2d0  
  
# List of serial device file names  
# For example:  
# SERIAL_DEVICE_FILE   /dev/tty0p0  
  
# Primary Network Interfaces on Bridged Net 1: lan0.  
# Warning: There are no standby network interfaces on bridged net 1.  
# Primary Network Interfaces on Bridged Net 2: lan3.  
# Possible standby Network Interfaces on Bridged Net 2: lan4.  
# Primary Network Interfaces on Bridged Net 3: lan1.  
# Warning: There are no standby network interfaces on bridged net 3.  
  
NODE_NAME              ftsys10  
NETWORK_INTERFACE     lan0  
HEARTBEAT_IP          15.130.1710.30  
NETWORK_INTERFACE     lan3  
HEARTBEAT_IP          192.60.70.4  
NETWORK_INTERFACE     lan4  
NETWORK_INTERFACE     lan1  
HEARTBEAT_IP          192.6.143.20  
FIRST_CLUSTER_LOCK_PV /dev/dsk/c1t2d0  
  
# List of serial device file names  
# For example:  
# SERIAL_DEVICE_FILE   /dev/tty0p0
```

HA(고가용성) 클러스터 구성 만들기 VxVM을 사용하여 저장 하부 구조 만들기

```
# Primary Network Interfaces on Bridged Net 1: lan0.
# Warning: There are no standby network interfaces on bridged net 1.
# Primary Network Interfaces on Bridged Net 2: lan3.
# Possible standby Network Interfaces on Bridged Net 2: lan4.
# Primary Network Interfaces on Bridged Net 3: lan1.
# Warning: There are no standby network interfaces on bridged net 3.

# 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 recommended value for NODE_TIMEOUT is 3000000
# (30 seconds).

HEARTBEAT_INTERVAL      1000000
NODE_TIMEOUT            6000000

# Configuration/Reconfiguration Timing Parameters (microseconds).

AUTO_START_TIMEOUT      600000000
NETWORK_POLLING_INTERVAL 2000000

# 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      10

#
# List of cluster aware Volume Groups. These volume groups
# will be used by package applications via the vgchange -a e command.
# For example:
# VOLUME_GROUP      /dev/vgdatabase
# VOLUME_GROUP      /dev/vg02

VOLUME_GROUP          /dev/vg01
VOLUME_GROUP          /dev/vg02
```

cmquerycl 명령에 대한 온라인 도움말에서 이 파일에 표시된 모든 매개 변수의 정의를 볼 수 있습니다. Many are also described in the ?lanning?chapter. 클러스터 워크시트의 데이터를 사용하여 필요에 따라 /etc/cmcluster/clust1.config 파일을 수정하십시오.

이 파일에서 키워드와 정의는 공백으로 구분됩니다. 주석을 사용할 수 있으며, 주석을 사용할 경우 왼쪽 줄 맨 앞에 파운드 기호(#)를 사용해야 합니다. 자세한 내용은 cmquerycl 명령에 대한 온라인 도움말을 참조하십시오.

잠금 디스크 지정

다음 예제 중 하나에서처럼 클러스터 잠금은 두 개의 노드 클러스터에서 요구됩니다. 이 잠금은 모든 노드에서 액세스할 수 있어야 하고 노드에서 독립 전원을 사용해야 합니다. 자세한 내용은 3장의 “클러스터 잠금” 절을 참조하십시오. 잠금 디스크 정보 다음에 클러스터 이름을 입력하십시오. 잠금 디스크는 클러스터에 있는 모든 노드에 액세스할 수 있는 LVM 볼륨 그룹에 있어야 합니다.

cmquerycl로 작성된 ASCII 템플릿에 제공하는 기본적인 FIRST_CLUSTER_LOCK_VG와 FIRST_CLUSTER_LOCK_PV는 최소로 계산된 장애 조치 시간에 기초하여 선택된 디스크의 볼륨 그룹과 실제 볼륨의 이름입니다. 이 디스크가 전원 배선 요구 사항을 충족하는지 확인해야 합니다. 필요할 경우 클러스터에 있는 노드의 절반 이하에 전원을 공급하는 회로에서 전원을 공급 받는 디스크를 선택하십시오.

디스크의 장애 조치 시간을 나타내려면 cmquerycl 명령을 사용하여 클러스터의 모든 노드들을 지정합니다.

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

명령의 출력 결과에서는 각 노드와 관련된 재편성 시간과 각 노드에 연결된 디스크들을 나열합니다.

주

구성에서 두 번째 잠금 볼륨 그룹이나 실제 볼륨을 특별히 요구하지 않는 한, 이를 구성하지 **마십시오**. 3장의 “클러스터 잠금” 절에 있는 “이중 클러스터 잠금” 부분을 참조하십시오.

또한, 보조 클러스터 잠금을 구성해야 하는 경우 다음 매개 변수를 클러스터 구성 파일에 입력하십시오.

```
SECOND_CLUSTER_LOCK_VG /dev/volume-group  
SECOND_CLUSTER_LOCK_PV /dev/dsk/block-special-file
```

여기에서 /dev/volume-group은 둘째 볼륨 그룹의 이름이고 block-special-file은 선택된 볼륨 그룹에 있는 잠금 디스크의 실제 볼륨 이름입니다. 이 행은 각 노드에 추가해야 합니다.

quorum 서버 지정

잠금 디스크 대신 quorum 서버를 지정하려면 cmquerycl 명령의 -q 옵션을 사용하여 Quorum 서버의 호스트 서버를 지정합니다. 예를 들면 다음과 같습니다.

```
# cmquerycl -n node1 -n node2 -q lp-qs
```

이 경우 생성된 클러스터 ASCII 파일에는 quorum 서버를 정의할 매개 변수가 들어 있습니다. 파일의 이 부분이 아래에 나타나 있습니다.

HA(고가용성) 클러스터 구성 만들기 VxVM을 사용하여 저장 하부 구조 만들기

```
# 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 12000000
# QS_TIMEOUT_EXTENSION 2000000
```

QS_HOST, QS_POLLING_INTERVAL 및 QS_TIMEOUT_EXTENSION을 입력합니다.

하트비트 서브넷 확인

클러스터 ASCII 파일에는 하트비트 서브넷 확인에 IP 주소에 대한 항목이 들어 있습니다. 전용 하트비트 서브넷을 사용하는 것이 좋지만 데이터 서브넷을 비롯하여 다른 서브넷에 하트비트를 구성할 수 있습니다.

주

VERITAS CVM 디스크 그룹을 사용할 경우 **단일** 하트비트 서브넷만 구성할 수 있습니다. 이 서브넷은 전용 서브넷이 됩니다. 사용 가능한 하트비트 경로가 있는지 확인할 수 있도록 이 서브넷의 각 시스템에는 대기 LAN이 구성되어 있어야 합니다.

구성된 패키지의 최대 수 지정

MC/ServiceGuard는 클러스터를 시작할 때 메모리와 스레드를 미리 할당합니다. 또한 클러스터 구성 파일의 MAX_CONFIGURED_PACKAGES 매개 변수에 지정된 패키지 수를 기반으로 이 값을 계산합니다. 이 값은 현재 클러스터에 구성된 패키지의 수와 같거나 커야 합니다. 기본값은 0으로, 이 경우 패키지를 사용하려면 값을 입력해야 합니다. 클러스터 당 최대 패키지 수는 60입니다. MC/ServiceGuard는 6MB 외에 잠금 가능 메모리에 패키지 당 약 80KB를 할당합니다. VERITAS CVM 디스크 저장을 사용할 경우에는

MAX_CONFIGURED_PACKAGES 값을 선택할 때 CVM-VxVM-PKG를 MAX_CONFIGURED_PACKAGES에 포함시켜야 합니다.

주

각 노드의 HP-UX 커널 매개 변수를 조정하여 해당 노드에서 동시에 실행할 최대 패키지 수를 허용할 수 있을 만큼 크게 설정되어 있는지 확인하십시오.

클러스터 타이밍 매개 변수 수정

cmquerycl 명령은 HEARTBEAT_INTERVAL 및 NODE_TIMEOUT에 대한 기본 클러스터 타이밍 매개 변수를 제공합니다. 이러한 매개 변수를 변경하면 클러스터 재편성 시간과 장애 조치 시간에 직접적인 영향을 미칩니다. 클러스터가 과도한 시스템 로드나 네트워크 통신량 때문에 재편성되는 경우 이 매개 변수를 수정하는 것이 좋습니다.

NODE_TIMEOUT에 대한 기본값 2초는 30초의 최적 장애 조치 시간을 제공합니다. NODE_TIMEOUT을 10초로 변경하면 클러스터 관리자는 노드를 종료하기 위해 5배 더 오래 기다려야 하기 때문에 장애 조치 시간은 5배인 대략 150초가 됩니다. NODE_TIMEOUT 값은 2*HEARTBEAT_INTERVAL 이상이어야 합니다. 하나의 NODE_TIMEOUT 내에 최소 2개나 3개의 하트비트를 포함하는 것도 좋은 방법입니다.

직렬 하트비트 연결 식별

하트비트 연결로 직렬(RS232) 회선을 사용하는 경우 SERIAL_DEVICE_FILE 매개 변수를 사용하고 각 노드에서 사용 중인 직렬 포트에 해당하는 장치 파일 이름을 입력하십시오. 구성 중이나 후에 직렬 케이블이 올바르게 연결되어 있는지 확인하십시오.

볼륨 그룹 추가

클러스터에 사용될 각 클러스터 인식 볼륨 그룹에 대한 별도의 VOLUME_GROUP 매개 변수와 함께, 구성된 모든 LVM 볼륨 그룹을 ASCII 클러스터 구성 파일에 추가하십시오. 이러한 볼륨 그룹은 cmapplyconf 명령을 사용하면 클러스터 ID로 초기화됩니다. 또한, 적절한 볼륨 그룹, 논리 볼륨 및 파일 시스템 정보를 볼륨 그룹을 활성화할 각 패키지 제어 스크립트에 추가해야 합니다. 이 프로세스는 6장에서 설명합니다.

주

CVM 디스크 그룹을 사용할 경우 192페이지에 “CVM을 사용하여 저장 하부 구조 만들기”에 설명된 절차대로 클러스터를 구성한 다음 CVM 디스크 그룹을 구성해야 합니다. VERITAS 디스크 그룹은 6장에 설명된 대로 패키지 구성 파일에 추가합니다.

클러스터 구성 확인

SAM은 입력된 구성을 자동으로 확인한 다음 오류를 보고합니다. ASCII 클러스터 구성 파일을 편집한 경우 다음 명령을 사용하면 파일의 내용을 확인할 수 있습니다.

```
# cmcheckconf -k -v -C /etc/cmcluster/clust1.config
```

이 명령이나 SAM의 자동 확인 기능 모두 다음 항목을 검사합니다.

- 네트워크 주소 및 연결
- 클러스터 잠금 연결(잠금 디스크를 구성할 경우)
- 클러스터 및 패키지에 대한 구성 매개 변수의 유효성
- 이름의 고유성
- 명령줄에 지정된 스크립트가 있는지 여부와 사용 권한
- 지정된 모든 노드가 동일한 하트비트 서브넷에 있는지 여부
- 잘못된 구성 파일 이름을 지정했는지 여부
- 모든 노드를 액세스할 수 있는지 여부
- 각각 하나씩의 CLUSTER_NAME, HEARTBEAT_INTERVAL, AUTO_START_TIMEOUT만 지정됩니다.
- 패키지 실행 및 중지 스크립트 시간 제한 값이 4294초 이하인지 여부
- HEARTBEAT_INTERVAL의 값은 적어도 1초 이상입니다.
- NODE_TIMEOUT의 값은 적어도 HEARTBEAT_INTERVAL 값의 두 배 이상입니다.
- AUTO_START_TIMEOUT 변수 값은 >=0입니다.
- 하트비트 네트워크 최소 요구 사항. 클러스터는 대기 기능이 있는 하나의 하트비트 LAN, 두 개의 하트비트 LAN, 하나의 하트비트 LAN과 하나의 RS232 연결, 또는 로컬 LAN 전환 기능은 없으나 2개 이상의 인터페이스의 링크 집계로 구성된 기본 LAN을 가진 하나의 하트비트 네트워크 중에서 하나를 사용해야 합니다.
- 하나 이상의 NODE_NAME이 지정되었는지 여부
- 각 노드가 각 하트비트 네트워크에 연결되어 있는지 여부
- 모든 하트비트 네트워크가 동일한 종류의 LAN인지 여부

- 지정된 네트워크 인터페이스 장치 파일이 유효한 LAN 장치 파일인지 여부
- RS-232가 2개의 노드를 사용하는 클러스터에 구성되어 있고 노드 당 하나의 직렬 (RS232) 포트만 연결되어 있는지 여부
- VOLUME_GROUP 항목이 현재 클러스터 인식 볼륨으로 표시되어 있지 않은지 여부
- CVM 디스크 저장을 사용할 경우 하나의 하트비트 서브넷만 구성됩니다.

클러스터가 온라인 상태에 있을 때 SAM(또는 cmcheckconf 명령)은 구성에서 특정 변경 사항에 대한 모든 조건이 유효한지도 검사합니다.

주

-k 옵션을 사용하면 cmcheckconf가 ASCII 파일에서 식별된 LVM 디스크로의 디스크 연결만 검사합니다. -k 옵션을 생략(기본 설정)하면 cmcheckconf가 모든 노드에 있는 모든 LVM 디스크의 연결을 검사합니다. -k를 사용하면 명령이 보다 신속히 수행될 수 있습니다.

이진 구성 파일 배포

모든 클러스터 매개 변수를 지정한 다음에는 SAM 또는 HP-UX 명령을 사용하여 구성을 적용합니다. 이렇게 하면 클러스터의 모든 노드에 이진 구성 파일이 배포됩니다. 패키지를 구성(다음 장에서 설명)하기 전에 별도로 이 작업을 수행하는 것이 좋습니다. 이 방법으로 실행 중인 클러스터에서 cmviewcl 명령을 사용하여 클러스터 잠금, 하트비트 네트워크 및 다른 클러스터 수준의 작업을 확인할 수 있습니다. 구성을 배포하기 전에 보안 파일이 클러스터 노드 간의 복사를 허용하는지 확인하십시오. 자세한 내용은 이 장의 앞에서 “시스템 준비”를 참조하십시오.

SAM을 사용한 이진 파일 배포

SAM의 Cluster Configuration 하위 영역에서 매개 변수 입력을 완료하면 클러스터의 모든 노드로 파일을 복사할 것인지 확인하는 메시지가 표시됩니다. 이 때 OK를 선택하면 MC/ServiceGuard는 이진 구성 파일과 ASCII 구성 파일을 클러스터의 모든 노드에 복사합니다.

HP-UX 명령을 사용한 이진 파일 배포

다음 단계에 따라 이진 구성 파일을 만들어 클러스터의 모든 노드에 배포하십시오.

- 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 사용하면 cmapplyconf가 ASCII 파일에서 식별된 LVM 디스크로의 디스크 연결만 검사합니다. -k 옵션을 생략하면(기본 설정) cmapplyconf가 모든 노드에 있는 모든 LVM 디스크의 연결을 검사합니다. -k를 사용하면 명령이 보다 신속히 수행될 수 있습니다.

- 클러스터 잠금 볼륨 그룹을 비활성화합니다.

```
# vgchange -a n /dev/vglock
```

cmapplyconf 명령은 클러스터 구성 파일의 이진 버전을 작성하여 클러스터의 모든 노드에 배포합니다. 이렇게 하면 모든 노드에서 파일의 내용이 일관된 상태를 유지합니다. cmapplyconf 명령은 ASCII 구성 파일을 배포하지 않습니다.

주의

잠금 디스크가 초기화될 수 있도록 클러스터 잠금 볼륨 그룹은 cmapplyconf 명령을 실행하는 **노드에서만** 활성화되어야 합니다. 잠금 볼륨 그룹이 다른 노드에서 활성화되어 있는 동안 한 노드에서 SAM을 사용하거나 cmapplyconf 명령을 실행하여 클러스터를 구성하려고 하면 클러스터 잠금은 알 수 없는 상태로 남게 됩니다. 따라서 클러스터를 구성할 때 클러스터 잠금 볼륨 그룹이 해당 구성 노드에서만 활성화 상태이고 다른 모든 노드에 대해서는 비활성화되어 있는지 확인해야 합니다.

반드시 cmapplyconf를 실행한 다음 구성 노드의 클러스터 잠금 볼륨 그룹을 비활성화하십시오.

볼륨 그룹 및 클러스터 잠금 구성 데이터 저장

클러스터를 구성한 다음 만든 각 볼륨 그룹에 대해 `vgcfgbackup` 명령을 사용하여 LVM 볼륨 그룹 구성의 백업 사본을 만드십시오. 볼륨 그룹의 디스크를 교체해야 할 경우 `vgcfgrestore` 명령을 사용하여 디스크의 메타 데이터를 복원할 수 있습니다. 이 과정에 대한 자세한 내용은 “문제 해결” 장의 “디스크 교체”를 참조하십시오.

클러스터 잠금 볼륨 그룹을 포함하여 모든 볼륨 그룹에 대해 반드시 `vgcfgbackup`를 사용해야 합니다.

주

SAM이나 HP-UX 명령 중 어떤 것을 사용해서 볼륨 그룹을 작성했는지에 관계 없이 반드시 `vgcfgbackup` 명령을 사용하여 클러스터 잠금 디스크의 구성 데이터 사본을 저장해야 합니다.

클러스터 잠금 디스크를 클러스터 실행 중에 교체해야 할 경우에는 반드시 `vgcfgrestore` 명령을 사용하여 교체 디스크에 잠금 정보를 복원해야 합니다. 그렇지 않으면 잠금 디스크의 모든 중복 사본에 문제가 발생하고 교체 장치나 LUN에 복원된 잠금 구성이 없을 경우 전체 클러스터에 장애가 발생할 수 있습니다. 클러스터 잠금 디스크가 디스크 어레이에 구성되어 있는 경우, RAID 보호는 클러스터 잠금 데이터의 중복 사본을 제공합니다. MirrorDisk/UX는 클러스터 잠금 정보를 미러링하지 않습니다.

CVM을 사용하여 저장 하부 구조 만들기

클러스터를 구성하기 전에 적절한 논리 볼륨 하부 구조를 만들어 여러 노드의 데이터에 액세스할 수 있게 하십시오. 논리 볼륨 관리자(LVM), VERITAS Cluster Volume Manager(CVM) 또는 VERITAS Volume Manager(VxVM)를 사용할 수 있습니다. 필요에 따라 이들 볼륨을 함께 사용할 수 있습니다. LVM 및 VxVM 구성은 클러스터 구성 전에, CVM 구성은 클러스터 구성 후에 수행합니다.

LVM에서 VxVM 또는 CVM 저장으로 변환하는 방법에 대해서는 부록 G를 참조하십시오.

이 절에서는 VERITAS 클러스터 볼륨 관리자(CVM) 명령을 사용하여 저장을 구성하는 방법에 대해 설명합니다. 시작하기 전에 VxVM 저장된 디렉토리 (/usr/lib/vxvm/bin)가 경로에 있는지 확인하십시오. vxinstall을 사용하여 루트 디스크 그룹을 만들면 VxVM 명령이나 VERITAS Storage Administrator GUI인 vmsa를 사용하여 구성 작업을 수행할 수 있습니다. vmsa를 사용 중이면 GUI를 실행하기 전에 Storage Administrator 서버가 실행 중인지 확인하십시오. vxinstall 실행에 대한 세부 지침은 VERITAS Volume Manager for HP-UX Release Notes에 설명되어 있습니다. 자세한 내용은 VERITAS Volume Manager Storage Administrator Administrator's Guide를 참조하십시오.

다음 작업에 대한 별도의 절차는 아래에 제공됩니다.

- 루트 디스크 그룹 만들기
- CVM에서 사용할 수 있도록 클러스터 준비
- 공유 저장을 위한 디스크 그룹 만들기

플렉스(미러), 다중 경로 및 RAID 구성에 대한 세부 사항을 비롯한 자세한 내용은 VERITAS 볼륨 관리자에 대한 HP-UX 설명서를 참조하십시오.

VERITAS 볼륨 관리자 초기화

처음으로 디스크 그룹을 만들 경우에는 볼륨 관리자를 초기화해야 합니다. 이 작업은 하나의 디스크가 들어 있는 rootdg 디스크 그룹을 만들어 수행합니다. 각 노드에 VxVM/CVM을 설치한 후에 다음 명령을 사용하십시오.

vxinstall

이 명령은 VxVM/CVM 초기화 순서를 안내하는 메뉴 구동 프로그램을 표시합니다. 기본 메뉴에서 “Custom” 옵션을 선택하고 *rootdg*에 포함시킬 디스크를 지정합니다.

중요

VERITAS 볼륨 관리자의 *rootdg*는 HP-UX 루트 디스크와 같지 않으므로 HP-UX 루트 파일 시스템(/)에 사용할 수 **없습니다**. 또한, 공유 저장에 *rootdg*를 사용할 수 없습니다. 그러나, *rootdg*는 다른 로컬 파일 시스템(예: /export/home)에 사용할 수 있으므로 삭제할 필요가 없습니다.

각 노드에 한 번씩만 루트 디스크 그룹을 만들어야 합니다.

CVM에서 사용할 수 있도록 클러스터 준비

VERITAS 클러스터 볼륨 관리자(CVM)를 사용하려면 특수한 CVM 패키지와 함께 실행하는 클러스터가 있어야 합니다. 즉, 디스크 그룹을 만들기 전에 클러스터를 구성하여 실행해야 합니다.

주

클러스터 구성은 이전 절에 설명되어 있습니다.

CVM 디스크 그룹 구성에 클러스터를 준비하려면 클러스터 ASCII 구성 파일에서 `MAX_CONFIGURED_PACKAGES`를 1보다 큰 값으로 설정하고 하나의 하트비트 서브넷만 구성하십시오. 그런 다음 다음 명령을 사용하십시오. 이 명령은 클러스터 정보를 CVM에 알리는 특수 패키지를 만듭니다.

```
# cmapplyconf -P /etc/cmcluster/cvm/VxVM-CVM-pkg.conf
```

경고

`VxVM-CVM-pkg.conf` 파일은 편집하지 마십시오.

이 명령이 제대로 수행되면 다음 절에 설명된 대로 공유할 디스크 그룹을 만들 수 있습니다. 클러스터는 모든 노드에 있는 `VxVM-CVM-pkg`라고 하는 특수한 시스템 다중 노드 패키지로 실행됩니다. 다음 `cmviewcl` 명령 실행 결과는 이 패키지를 보여줍니다.

HA(고가용성) 클러스터 구성 만들기 VxVM을 사용하여 저장 하부 구조 만들기

```
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 주 노드 확인

클러스터가 아직 실행되고 있지 않으면 시작하십시오. 이렇게 하면 특수한 CVM 패키지가 활성화됩니다.

```
# cmruncl
```

CVM를 시작하면 주 노드가 선택됩니다. 이 노드는 디스크 그룹 구성 명령을 실행해야 하는 노드입니다. 주 노드를 결정하려면 클러스터의 각 노드에서 다음 명령을 실행하십시오.

```
# vxdctl -c mode
```

한 노드는 자신을 주 노드로 인식합니다. 이 노드에서 디스크 그룹을 만듭니다.

CVM에 대한 디스크 초기화

CVM 디스크 그룹에 사용할 물리 디스크를 초기화해야 합니다. LVM에서 이전에 물리 디스크를 사용한 경우에는 `pvremove` 명령을 사용하여 볼륨 그룹에 있는 모든 디스크에서 LVM 헤더 데이터를 삭제해야 합니다. 이전에 LVM에서 디스크를 사용하지 않은 경우에는 이 작업이 필요하지 않습니다.

CVM의 디스크를 초기화하려면 주 노드에 로그인한 다음 `vxdiskadm` 프로그램을 사용하여 여러 디스크를 초기화하거나 아래 예제에서처럼 `vxdisksetup` 명령을 사용하여 한 번에 하나의 디스크를 초기화하십시오.

```
# /usr/lib/vxvm/bin/vxdisksetup -i /dev/dsk/c0t3d2
```

디스크 그룹 만들기

다음 단계대로 디스크 그룹을 만들어야 합니다.

1. `vxpdg` 명령을 사용하여 디스크 그룹을 만듭니다. 아래 예제에서처럼 `-s` 옵션을 사용하여 공유 모드를 지정합니다.

```
# vxpdg -s init logdata c0t3d2
```

2. 다음 명령을 사용하여 구성을 확인하십시오.

```
# vxpdg list
```

NAME	STATE	ID
rootdg	enabled	971995699.1025.node1
logdata	enabled,shared	972078742.1084.node2

3. 볼륨을 만들기 전에 다음과 같이 디스크 그룹을 활성화합니다.

```
# vxpdg -g logdata set activation=on
```

볼륨 만들기

아래 예제에서처럼 `vxassist` 명령을 사용하여 논리 볼륨을 만드십시오.

```
# vxassist -g logdata make log_files 1024m
```

이 명령은 디스크 그룹 `logdata`에 1024MB의 `log_files` 볼륨을 만듭니다. 이 볼륨은 블록 장치 파일 `/dev/vx/dsk/logdata/log_files` 또는 원시(문자) 장치 파일 `/dev/vx/rdsk/logdata/log_files`로 참조됩니다.

다음 명령을 사용하여 구성을 확인하십시오.

```
# vxpdg list
```

CVM을 사용한 미러 분리 정책

기본 CVM 디스크 미러 분리 정책은 '전체적'입니다. 즉, 한 노드에서 특정한 미러 사본(플렉스)을 볼 수 없게 되면 모든 노드에서도 이 미러 사본을 볼 수 없게 됩니다. 다른 정책은 '로컬'입니다. 즉, 한 노드에서 특정한 미러 사본을 볼 수 없게 되면 CVM은 이 노드의 볼륨에 대한 액세스만 비활성화합니다. 이 정책은 디스크 그룹을 기준으로 다음과 같이 `vxedit` 명령을 사용하여 재설정할 수 있습니다.

```
# vxedit set diskdetpolicy=[global|local] <DiskGroupName>
```

주

미러링된 저장 및 CVM을 사용한 다중 경로 저장을 만드는 특수 명령에 대해서는 VERITAS 볼륨 관리자에 대한 HP-UX 설명서를 참조하십시오.

파일 시스템 만들기

설치 프로그램이 파일 시스템을 사용하면 나중에 파일 시스템을 만드십시오. 다음 명령을 사용하여 위에서 만든 논리 볼륨에 마운트할 파일 시스템을 만드십시오.

1. 새로 만든 논리 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 나타났는지 확인한 다음 파일 시스템은 언마운트하십시오.

```
#umount /logs
```

5. 다음 명령을 사용하여 디스크 그룹을 비활성화합니다.

```
#vxdg -g logdata set activation=off
```

패키지 구성에 디스크 그룹 추가

VxVM 명령을 사용하여 저장 장치를 만들면 각 패키지 구성 ASCII 파일에 CVM 디스크 그룹을 지정해야 합니다. 패키지에서 사용할 각 디스크 그룹에 DISK_GROUP 매개 변수를 사용하십시오. CVM 디스크 그룹, 파일 시스템, 논리 볼륨 및 마운트 옵션을 확인해야 합니다. 패키지 구성 프로세스는 6장에 설명되어 있습니다.

주

CVM 디스크 그룹은 LVM 볼륨 그룹과 달리 클러스터 ASCII 구성 파일에 입력되지 않습니다.

실행 클러스터 관리

이 절에서는 클러스터의 일상적인 관리에 대한 몇 가지 접근 방법에 대해 설명합니다. 추가 도구 및 제안 사항에 대해서는 7장 “클러스터 및 패키지 유지 관리”에서 설명합니다.

ServiceGuard Manager를 사용한 클러스터 작동 검사

ServiceGuard Manager는 클러스터에 구성된 모든 노드와 패키지 및 현재 상태를 나타냅니다. 자세한 내용은 7장의 “ServiceGuard Manager 사용” 절을 참조하십시오. ServiceGuard Manager를 사용하여 다음과 같은 사항을 확인하십시오.

- 구성된 노드가 모두 실행 중인지 확인합니다.
- 구성된 패키지가 모두 실행 중이고 해당 노드에서 실행되고 있는지 확인합니다.
- 클러스터, 노드 및 패키지의 속성 시트의 설정이 정확인지 확인합니다.

클러스터가 정확하게 구성되어 있으면 보관용으로 구성 데이터 사본을 파일에 저장합니다. 상당 기간의 변경을 이해하기 위해서 이 파일의 데이터를 이후 버전 클러스터와 비교할 수 있습니다.

ServiceGuard 명령을 사용한 클러스터 작동 검사

또한, MC/ServiceGuard는 클러스터의 수동 제어를 위한 몇 가지 명령을 제공합니다.

- `cmrunnode`는 노드를 시작하는 데 사용됩니다.
- `cmhaltnode`는 실행 중인 노드를 수동으로 중지하는 데 사용됩니다. 이 명령은 `shutdown(1m)`에서도 사용됩니다.
- `cmrunc1`은 중지된 클러스터를 수동으로 시작하는 데 사용됩니다.
- `cmhaltc1`은 클러스터를 수동으로 중지하는 데 사용됩니다.

다음과 같이 이러한 명령을 사용하여 클러스터 작동을 검사할 수 있습니다.

HA(고가용성) 클러스터 구성 만들기 VxVM을 사용하여 저장 하부 구조 만들기

1. 클러스터가 아직 온라인 상태가 아닌 경우 다음 방법으로 클러스터를 실행합니다.

```
# cmruncl -v
```

2. 클러스터가 시작된 경우 다음 명령을 사용하여 클러스터 구성 요소가 제대로 작동하고 있는지 확인합니다.

```
# cmviewcl -v
```

모든 노드 및 네트워크가 예상대로 기능하는지 확인하십시오. cmviewcl 사용에 대한 자세한 내용은 7장 “클러스터 및 패키지 유지 관리”를 참조하십시오.

3. 다음 명령을 사용하여 노드가 예상대로 클러스터에서 나가고 들어가는지 확인합니다.

- 클러스터 노드에서 cmhaltnode 명령을 실행합니다.
- cmviewcl 명령을 사용하여 노드가 클러스터에서 나갔는지 확인합니다.
- cmrunnode 명령을 실행합니다.
- cmviewcl 명령을 다시 사용하여 노드가 작동하는지 확인합니다.

4. 다음 명령을 사용하여 클러스터 작동을 중지시킵니다.

```
# cmhaltcl -v -f
```

추가 클러스터 테스트 작업은 “문제점 해결” 장을 참조하십시오. MC/ServiceGuard 명령의 완전한 목록은 부록 A를 참조하십시오.

볼륨 그룹의 자동 활성화 방지

/etc/lvmrc 파일을 사용하여 시스템이 시작될 때 패키지의 LVM 볼륨 그룹이 활성화되지 않게 해야 합니다. 이렇게 하려면 모든 노드에서 /etc/lvmrc 파일을 편집하십시오. AUTO_VG_ACTIVATE를 0으로 설정한 다음 클러스터에 연결되지 않은 모든 볼륨 그룹을 custom_vg_activation 함수에 포함하십시오. 패키지에 사용될 볼륨 그룹은 제어 스크립트에 의해 활성화되고 비활성화되므로 파일의 어느 위치에도 포함하지 **마십시오**.

주

로트 볼륨 그룹은 부팅 시 /etc/lvmrc 파일이 사용되기 전에 자동으로 활성화되므로 custom_vg_activation 함수에 포함시키지 않아도 됩니다.

자동 시작 기능 설정

자동 시작은 각 노드가 클러스터에 개별적으로 참여하는 과정입니다. MC/ServiceGuard는 시작 과정을 제어하는 시작 스크립트를 제공합니다. 클러스터가 이미 존재할 경우 노드는 이 클러스터에 참여하려고 합니다. 실행 중인 클러스터가 없으면 노드는 설정된 모든 노드로 구성되는 클러스터를 편성하려고 합니다. 자동 클러스터 시작을 사용하여 클러스터를 시작하는 것이 좋습니다. 시스템 관리자가 수행해야 하는 작업은 없습니다.

자동 클러스터 시작을 사용하려면 클러스터의 모든 노드에 구성된 `/etc/rc.config.d/cmcluster` 파일에서 플래그 `AUTOSTART_CMCLD`를 1로 설정하십시오. 그러면 시작할 때 노드는 클러스터에 참여합니다.

다음은 `/etc/rc.config.d/cmcluster` 파일의 예제입니다.

```
***** 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

필요하면 다음과 같은 내용을 포함하도록 시스템 로그인 메시지를 수정할 수 있습니다.

```
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.
```

추가적인 클러스터 특정 정보와 함께 이 메시지에 모든 클러스터 노드의 목록을 포함시킬 수 있습니다.

`/etc/issue` 및 `/etc/motd` 파일은 클러스터 관련 정보를 포함하도록 사용자 정의할 수 있습니다.

단일 노드 클러스터 관리

MC/ServiceGuard 클러스터에 필요한 노드의 수는 보호하려는 응용 프로그램의 프로세스 요구 사항에 따라 달라집니다. MC/ServiceGuard의 네트워크 장애 보호 기능을 활용하도록 단일 노드 클러스터를 구성할 수 있습니다.

단일 노드 클러스터에서는 클러스터에 다른 노드가 없기 때문에 클러스터 잠금은 필요하지 않습니다. 노드가 하나만 있을 경우에는 `cmquerc1` 명령의 출력 결과에 클러스터 잠금 정보 영역이 생략됩니다.

중복 네트워크가 필요하지만 하트비트를 전송할 노드가 없기 때문에 하트비트 LAN을 지정하지 않아도 됩니다. 클러스터 구성 ASCII 파일에서 ServiceGuard에서 모니터 할 LAN을 모두 지정하십시오. LAN에 이미 IP 주소가 있을 경우에는 HEARTBEAT_IP 키워드가 아닌 STATIONARY_IP 키워드와 함께 이들 주소를 지정하십시오. 대기 LAN의 경우 LAN 장치 이름과 함께 NETWORK_INTERFACE 키워드만 지정하면 됩니다.

단일 노드 실행

단일 노드 클러스터나 다중 노드 클러스터에서 단일 노드가 실행되고 한 노드를 제외한 모든 노드에서 장애가 발생했거나 종료되어도 응용 프로그램이 실행됩니다.

MC/ServiceGuard 데몬 `cmc1d`가 활성 상태에 있으면 다른 노드들은 나중에 클러스터에 다시 참여할 수 있습니다.

단일 노드 실행에서 MC/ServiceGuard 데몬에 장애가 발생할 경우, 단일 노드와 응용 프로그램은 계속 실행됩니다. 이것은 다중 노드 클러스터에서 MC/ServiceGuard 데몬이 손실되는 것과는 다른데, 이 경우는 TOC와 함께 노드가 중지되고 패키지는 대체 노드로 전환됩니다. 이 시나리오에서는 응용 프로그램이 계속 실행 중이고 패키지가 전환될 다른 노드가 없기 때문에 단일 노드를 중지할 필요가 없습니다. 하지만, 단일 노드에서 계속 실행 중인 응용 프로그램을 다른 노드가 다시 시작하면 데이터가 손상될 수 있기 때문에 MC/ServiceGuard를 다시 시작해서는 안됩니다. 따라서 클러스터를 다시 시작하지 말고 적당한 시간에 노드를 중지한 다음 다시 시작하십시오. 이렇게 하면 응용 프로그램이 종료되고 노드가 다시 시작되었을 때 MC/ServiceGuard가 클러스터를 다시 시작합니다.

클러스터 구성 삭제

SAM을 사용하거나 `cmdleteconf` 명령을 실행하여 모든 클러스터 노드에서 클러스터 구성을 삭제할 수 있습니다. `-f` 옵션을 사용하지 않는 한 이 명령은 파일을 삭제하기 전에 확인 메시지를 표시합니다. 클러스터가 중지된 경우에만 구성을 삭제할 수 있습니다. 이 명령은 클러스터의 모든 노드에서 이전 구성 파일을 삭제하고 모든 클러스터 인식 볼륨 그룹을 클러스터를 인식하지 않는 볼륨 그룹으로 재설정합니다.

주

cmdeleteconf 명령은 클러스터 이진 파일 /etc/cmcluster/cmclconfig만 제거합니다. /etc/cmcluster 디렉터리의 다른 파일을 삭제하지는 **않습니다**.

클러스터를 중지해야 하는 경우에도 클러스터의 모든 노드에 전원이 공급되어야 하며 cmdeleteconf 명령을 사용하기 전에 액세스할 수 있어야 합니다. 노드에 전원이 공급되지 않는 경우 전원을 연결하고 노드를 시작하십시오. 노드를 액세스할 수 없을 경우 다음 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?
```

Yes를 선택하여 구성을 제거합니다. 나중에 사용할 수 없는 노드에 다시 액세스할 수 있게 되면 해당 노드에서 cmdeleteconf 명령을 실행하여 구성 파일을 제거하십시오.

HA(고가용성) 클러스터 구성 만들기
VxVM을 사용하여 저장 하부 구조 만들기

6

패키지 및 서비스 구성

클러스터를 구성한 다음, 패키지로 그룹화하려는 응용 프로그램과 서비스를 찾아야 합니다. 이 장에서는 다음의 **패키지 구성** 작업을 설명합니다.

- 패키지 구성 만들기
- 패키지 제어 스크립트 작성
- 패키지 구성 확인
- 구성 배포

각각의 작업은 아래에 있는 별도의 절에서 설명합니다.

사용자의 해당 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 그 작업을 모니터링하는 실행 가능한 패키지 제어 스크립트에 사용됩니다.

패키지 구성 만들기

패키지 구성은 패키지가 클러스터의 한 노드에서 시작될 때 패키지 관리자에 의해 실행될 일련의 응용 프로그램 서비스를 정의하는 과정입니다. 또한, 이 구성에는 패키지에 대해 허용되는 사용 가능한 장애 조치 유형의 정의에 따라 해당 패키지가 실행될 수 있는 클러스터 노드의 우선 순위 목록도 포함됩니다.

SAM을 사용하거나 HP-UX 명령 및 편집기를 사용하여 패키지를 작성할 수 있습니다. 다음 절은 SAM 구성을 설명합니다. MC/ServiceGuard 명령을 사용하는 경우 “MC/ServiceGuard 명령을 사용한 패키지 작성” 절로 건너뛰십시오.

SAM을 사용한 패키지 구성

고가용성 패키지를 구성하려면 구성 노드(fts9)에서 다음 단계를 수행하십시오.

1. SAM에서 “Clusters” 영역을 선택한 다음 High Availability Clusters 옵션을 선택합니다.
2. Package Configuration 옵션을 선택합니다. SAM은 패키지 구성 화면을 표시합니다. 아직 구성된 패키지가 없으면 목록 영역은 빈 상태로 표시됩니다. 하나 이상의 MC/ServiceGuard 패키지가 네트워크의 클러스터에 이미 구성되어 있을 경우 목록에 해당 패키지가 표시됩니다.
3. Actions 메뉴를 선택한 다음 Create/Add a Package를 선택합니다. 그러면 단계 메뉴가 표시됩니다.
4. 필요한 각 단계를 순서대로 선택한 다음 대화 상자에서 필요한 정보를 완성하거나 기본값을 그대로 사용합니다. 각 단계에 대한 자세한 내용을 보려면 Help를 선택하십시오.
5. 모든 단계가 완료되면 Step Menu 화면에서 OK를 선택합니다. 그러면 패키지 구성 파일이 생성(또는 수정)되어 클러스터의 모든 노드에 복사됩니다. 또한, 패키지 제어 스크립트가 작성되어 모든 노드에 복사됩니다.
6. 파일 복사가 완료되면 Package Configuration 화면으로 되돌아갑니다.
7. Package Configuration 화면에서 High Availability Clusters 메뉴로 되돌아갑니다.

“패키지 제어 스크립트 작성” 절로 건너뛰십시오.

MC/ServiceGuard 명령을 사용한 패키지 작성

다음 단계대로 패키지 구성 파일을 편집하고 처리하여 패키지를 작성할 수 있습니다.

1. 먼저, /etc/cmcluster 디렉토리에 구성할 각 패키지의 하위 디렉토리를 만듭니다.

```
# mkdir /etc/cmcluster/pkg1
```

원하는 다른 디렉토리 이름을 사용할 수 있습니다.

2. 그런 다음 패키지에 대한 패키지 구성 템플릿을 만듭니다.

```
# cmmakepkg -p /etc/cmcluster/pkg1/pkg1.config
```

ASCII 템플릿에 원하는 다른 파일 이름을 사용할 수 있습니다.

3. 이 템플릿 파일을 편집하여 각 패키지에 대한 패키지 이름, 노드의 우선 순위 목록, 제어 스크립트의 위치, 장애 조치 매개 변수를 지정할 수 있습니다. 패키지 구성 워크시트에 기록된 데이터를 포함시키십시오.

단계별 구성

다음과 같이 단계별로 클러스터에 패키지를 구성하는 것이 바람직합니다.

1. 볼륨 그룹과 마운트 지점만 구성합니다.
2. 이 구성을 적용합니다.
3. 모든 노드에 제어 스크립트를 배포합니다.
4. 패키지를 실행한 다음, 패키지가 노드 사이에서 이동할 수 있는지 확인합니다.
5. 패키지를 중지합니다.
6. 제어 스크립트에서 패키지 IP 주소와 응용 프로그램 서비스를 구성합니다.
7. 모든 노드에 제어 스크립트를 배포합니다.
8. 패키지를 실행한 다음, 응용 프로그램이 정상적으로 실행되고 서비스가 중단되었을 때, 패키지의 장애 조치가 올바르게 실행되는지 확인합니다.

패키지 구성 템플릿 파일

다음은 일반적인 패키지에 대해 사용자 정의된 예제 패키지 구성 파일 템플릿입니다

```
# ***** 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 MC/ServiceGuard or ServiceGuard OPS Edition 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                                pkg1

# 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 paramters are ignored:
#
#         FAILOVER_POLICY
#         FAILBACK_POLICY
#
#     Since an IP address cannot be assigned to more than one 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 fallback 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 it primary node when it is running on an adoptive node.
#
# The alternate 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

# 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                ftsys9
NODE_NAME                ftsys10

# 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.

```

패키지 및 서비스 구성

패키지 구성 만들기

```
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 running and halting of the package.
# 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 specified for all services.

RUN_SCRIPT                      /etc/cmcluster/pkg1/control.sh
RUN_SCRIPT_TIMEOUT              NO_TIMEOUT
HALT_SCRIPT                     /etc/cmcluster/pkg1/control.sh
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.

# NOTE: Should only be used by applications provided by
# Hewlett-Packard.
#
# Example:STORAGE_GROUP  vg01
#           STORAGE_GROUP  vg02
#           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 names used by
# cmrunserv and cmhaltserv commands in the run and halt scripts.
#
# 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
# default will be NO.
#
# SERVICE_HALT_TIMEOUT is represented in the 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>

SERVICE_NAME          servicel
SERVICE_FAIL_FAST_ENABLED      NO
SERVICE_HALT_TIMEOUT          300

# 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.

SUBNET    15.16.168.0

# 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"
#

```

패키지 및 서비스 구성 패키지 구성 만들기

```
# 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
# 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>]
```

패키지 구성 워크시트의 정보를 사용하여 파일을 완성할 수 있습니다. 각 매개 변수에 대한 추가적인 설명은 구성 템플릿의 주석을 참조하십시오. 다음 정보를 포함할 수도 있습니다.

- **FAILOVER_POLICY:** CONFIGURED_NODE 또는 MIN_PACKAGE_NODE를 입력하십시오.
- **FAILBACK_POLICY:** MANUAL 또는 AUTOMATIC을 입력하십시오.
- **NODE_NAME:** 클러스터의 각 노드 이름을 별도의 행에 입력하십시오.
- **AUTO_RUN:** 제일 먼저 사용 가능한 노드에서 패키지를 시작하려면 YES를 입력하십시오. 패키지 자동 시작을 실행하지 않으려면 NO를 입력하십시오.
- **LOCAL_LAN_FAILOVER_ALLOWED:** 패키지 IP 주소가 대기 LAN으로 전환되도록 하려면 YES를 입력하십시오. 또는 패키지 로컬 전환을 허용하지 않으려면 NO를 입력하십시오.

- RUN_SCRIPT와 HALT_SCRIPT: 패키지 제어 스크립트의 경로 이름을 지정하십시오(다음 절에서 설명). 기본 경로 이름은 없습니다.
- STORAGE_GROUP: 이 패키지에서 사용할 모든 CVM 저장 그룹 이름을 지정하십시오. 별도의 행에 각 저장 그룹(CVM 디스크 그룹)을 입력하십시오. CVM 저장 그룹은 클러스터 ASCII 구성 파일에 입력되지 **않습니다**.

주

이 파일에 LVM 볼륨 그룹이나 VxVM 디스크 그룹을 입력하지 마십시오.

- 패키지에 서비스가 포함되어 있는 경우: SERVICE_NAME, SERVICE_FAIL_FAST_ENABLED 및 SERVICE_HALT_TIMEOUT 값을 입력하십시오. 각 서비스에 대해 이 세 항목을 입력하십시오. 패키지 당 서비스는 30개까지만 구성할 수 있습니다.
- 패키지에 해당 IP 주소가 있는 경우: SUBNET을 입력하십시오. 이것은 클러스터 구성에 이미 지정된 서브넷이어야 합니다.
- NODE_FAIL_FAST_ENABLED 매개 변수: YES 또는 NO를 입력하십시오.

패키지 내에서 등록된 리소스에 대한 모니터링을 구성하려면 다음 매개 변수의 값을 입력하십시오. SAM의 “Package Configuration” 하위 영역에 있는 “Additional Package Resources” 부분을 사용하여 패키지에 종속되도록 설정할 수 있는 리소스 이름, 폴링 주기 및 사용 가능 상태 값의 목록을 얻을 수 있습니다.

- RESOURCE_NAME: MC/ServiceGuard가 모니터링할 등록된 리소스의 이름을 입력하십시오.
- RESOURCE_POLLING_INTERVAL: 리소스가 사용 가능한 상태인지 확인하는 시간 주기를 입력하십시오.
- RESOURCE_UP_VALUE: 리소스가 사용 가능 상태로 인식될 때의 값을 입력하십시오. 모니터링하는 동안 리소스에 대해 다른 값이 발견되면 패키지에 오류가 발생합니다.
- RESOURCE_START: RESOURCE_START 옵션은 ServiceGuard가 EMS 리소스에 대해 리소스 모니터링을 시작하는 시간을 결정할 때 사용합니다. RESOURCE_START 옵션을 AUTOMATIC 또는 DEFERRED로 설정할 수 있습니다. AUTOMATIC으로 지정된 경우 ServiceGuard 클러스터 데몬이 노드에서 시작되면 ServiceGuard는 이 리소스에 대한 리소스 모니터링을 자동으로 시작합니다. AUTOMATIC으로 설정된 경우에 패키지 제어 스크립트에서 DEFERRED_RESOURCE_NAME을 정의할 필요가 없습니다.

패키지 및 서비스 구성 패키지 구성 만들기

DEFERRED로 지정된 경우, ServiceGuard는 노드가 시작되는 동안 이 DEFERRED 리소스에 대해 리소스 모니터링을 시작하려고 시도하지 않습니다. 그러나, 패키지 제어 스크립트에서 이 DEFERRED 리소스를 지정해야 합니다.

다음은 DEFERRED 및 AUTOMATIC 리소스를 구성하는 방법의 예제입니다. 패키지 구성 파일에서 리소스를 아래와 같이 지정하십시오.

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

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

RESOURCE_NAME           /net/interfaces/lan/status/lan2
RESOURCE_POLLING_INTERVAL 60
RESOURCE_START           AUTOMATIC
RESOURCE_UP_VALUE        = ONLINE
```

이 패키지 구성 데이터는 나중에 이진 클러스터 구성 파일의 클러스터 구성 데이터와 결합됩니다.

실행 중인 클러스터에서 패키지 추가 또는 제거

클러스터가 실행 중일 때 패키지를 추가하거나 제거할 수 있습니다. 이것은 MAX_CONFIGURED_PACKAGES의 제한을 받습니다. 온라인 상태에서 패키지를 추가 또는 제거하는 작업에 대해서는 “클러스터 및 패키지 유지 관리”장을 참고하십시오.

패키지 제어 스크립트 작성

패키지 제어 스크립트는 패키지의 모든 서비스를 실행하고, 실행 중에 모니터링하고, 장애에 대처하고, 필요한 경우 패키지를 중지하는 데 필요한 모든 정보를 포함합니다.

SAM 또는 HP-UX 명령을 사용하여 패키지 제어 스크립트를 작성하거나 수정할 수 있습니다. 보안상의 이유로 제어 스크립트는 경로에서 문자열 `cmcluster`를 포함하는 디렉토리에 위치해야 합니다.

SAM을 사용한 패키지 제어 스크립트 작성

SAM에서 고가용성 옵션을 선택한 다음 “패키지 구성”을 선택하십시오. 동작 메뉴에서 “패키지 작성/추가”를 선택하십시오. 그러면 단계 메뉴가 표시되고 이 메뉴에서 일련의 옵션을 볼 수 있습니다. 메뉴의 마지막 두 단계는 패키지 제어 스크립트의 경로 이름을 선택하기 위한 것입니다. 해당 패키지를 먼저 정의한 다음 각 옵션을 선택하십시오. 자세한 정보를 보려면 도움말 키를 사용하십시오.

이 방법으로 패키지 제어 스크립트를 작성할 때는 추가적으로 편집할 필요는 없지만 원하는 경우 스크립트를 사용자 정의할 수 있습니다.

명령을 사용한 패키지 제어 스크립트 작성

각 패키지에는 반드시 별도의 제어 스크립트가 있어야 하며, 이 스크립트는 실행 가능해야 합니다. 제어 스크립트는 패키지 디렉토리에 위치하며, 패키지 ASCII 구성 파일의 `RUN_SCRIPT`와 `HALT_SCRIPT` 매개 변수에 지정된 이름과 동일한 이름으로 지정됩니다. 패키지 제어 스크립트 템플릿은 패키지에 대한 실행 명령과 중지 명령을 포함합니다. 패키지를 실행 및 중지하기 위해서 혹은 별도 스크립트를 만들기 위해서 단일 스크립트를 사용할 수 있습니다.

다음 단계를 따라 예제 패키지 `pkg1`의 제어 스크립트를 작성하십시오.

먼저 제어 스크립트 템플릿을 만드십시오.

```
# cmmakepkg -s /etc/cmcluster/pkg1/pkg1.sh
```

“패키지 제어 스크립트 사용자 정의” 절에서 설명하는 것처럼 스크립트를 사용자 정의할 수 있습니다.

데이터베이스 제품의 패키지 만들기

별도 구입 가능한 ECM Toolkits 제품(B5139BA)과 함께 제공되는 데이터베이스 템플릿 파일을 사용하여, 클러스터 노드의 시작과 종료를 이용하여 데이터베이스 소프트웨어의 시작과 종료를 조정할 수 있습니다. 이 파일은 /opt/cmcluster/toolkit/DB/에 위치합니다. 별도의 도구 키트는 Oracle, Informix 및 Sybase에서 구입할 수 있습니다. 표준 패키지 제어 스크립트 뿐만 아니라 데이터베이스에 맞춘 특수 스크립트를 사용할 수 있습니다. 이 스크립트를 설정하려면 각 도구 키트와 함께 제공된 README 파일 설명서를 따라 수행하십시오.

패키지 제어 스크립트 사용자 정의

패키지 구성 워크시트의 정보를 사용하여 제어 스크립트의 맨 앞에 나오는 정의 및 선언을 확인하십시오. 다음과 같이 사용자 정의해야 합니다.

- 서비스를 시작하기 위해 필요한 경로와 일치하도록 PATH 문을 업데이트합니다.
- LVM을 사용할 경우 VG[] 어레이 매개 변수를 사용하여 활성화할 볼륨 그룹 이름을 입력하고 파일 시스템 마운트 및 언마운트에 대한 옵션을 비롯하여 저장 활성화 명령에 적절한 옵션을 선택합니다. LVM 볼륨 그룹에 VXVM_DG[] 또는 CVM_DG[] 매개 변수를 사용하지 **마십시오**.
- CVM을 사용할 경우 CVM_DG[] 어레이 매개 변수를 사용하여 활성화할 디스크 그룹 이름을 입력하고 해당 저장 활성화 명령 CVM_ACTIVATION_CMD를 선택합니다. CVM 디스크 그룹에 VG[] 또는 VXVM_DG[] 매개 변수를 사용하지 **마십시오**.
- CVM 없이 VxVM 디스크 그룹을 사용할 경우 VXVM_DG[] 어레이 매개 변수를 사용하여 가져올 VxVM 디스크 그룹 이름을 입력합니다. 어레이 요소마다 디스크 그룹을 입력합니다. CVM 없이 VxVM 디스크 그룹에 CVM_DG[] 또는 VG[] 매개 변수를 사용하지 **마십시오**. 또한, 활성화 명령을 지정하지 **마십시오**.
- 미러된 VxVM 디스크를 사용하는 경우 미러 복구 옵션 VXOL을 지정하십시오.
- 이 LVM 볼륨 그룹에 마운트될 논리 볼륨과 파일 시스템의 이름을 추가합니다.
- 저장 활성화 명령(기본 VxVM 디스크 그룹에는 해당되지 않음)에 대한 옵션을 선택하고 파일 시스템 마운트를 위한 옵션을 추가합니다.
- 파일 시스템 마운트 재시도 및 언마운트 횟수를 지정하십시오.
- 패키지의 해당 IP 서브넷과 IP 주소를 정의합니다.
- 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` 명령은 각 실행 명령을 수행한 다음 실행 명령에 의해 생성된 프로세스의 프로세스 ID를 모니터링합니다.
- `cmrunserv`에 의해 시작된 명령이 종료될 때 `MC/ServiceGuard`는 장애가 발생했음을 인식하고 적절한 조치를 취하는데, 이 장애 조치에 의해 패키지가 대체 노드로 전송될 수도 있습니다.
- 실행 명령이 다른 명령을 실행하는 셸 스크립트인데 종료된 경우 `MC/ServiceGuard`는 이러한 정상 종료를 **오류**로 간주합니다.

제어 스크립트 실행 시 문제가 발생하지 않게 하려면 각 실행 명령이 실제 서비스의 이름과 동일하고, 실제 서비스가 종료될 때까지 프로세스가 활성 상태를 유지하도록 하십시오.

기본 설정에 추가하여 일련의 실행 및 중지 작업을 정의해야 하는 경우 `CUSTOMER_DEFINED_FUNCTIONS`란 이름의 영역 아래에 이 작업에 대한 함수를 작성하십시오.

제어 스크립트로 VxVM 디스크 그룹을 관리하는 방법

CVM 외부에 있는 VxVM 디스크 그룹은 `ServiceGuard` 클러스터 제어 범위를 벗어납니다. 패키지 제어 스크립트는 표준 VxVM 명령을 사용하여 이러한 디스크 그룹을 가져오고 내보냅니다. 디스크 그룹을 가져오기와 내보내기에 대해서는 `vxdg man` 페이지의 `import` 및 `deport` 옵션을 참조하십시오.

제어 스크립트는 `vxdg` 명령과 함께 `-tfc` 옵션을 사용하여 디스크 그룹을 가져옵니다. `-t` 옵션은 `noautoimport` 플래그를 사용하여 디스크를 가져오도록 지정합니다. 즉, 부팅 시 디스크를 자동으로 다시 가져오지 않습니다. `ServiceGuard` 패키지에서만 패키지 제어 스크립트에 포함된 디스크 그룹을 가져오기 때문에 디스크를 자동으로 가져오지 않습니다.

`-f` 옵션을 사용하면 하나 이상의 디스크(예: 미러)를 현재 사용할 수 없는 경우에도 디스크 그룹을 가져올 수 있습니다. `-c` 옵션은 클러스터에 있는 다른 노드에서 활성화하기 전에 디스크에 기록된 기존의 모든 호스트 ID를 지웁니다. `TOC`로 중지된 다른 노드에서 디스크를 사용한 경우에는 해당 호스트 ID는 디스크에 그대로 기록됩니다. 새로운

노드의 ID를 디스크에 기록할 수 있도록 이 호스트 ID를 지워야 합니다. 호스트 ID가 설정되고 장애가 발생하지 않은 노드와 일치하면 디스크 그룹은 지운 호스트 ID를 가져오지 않습니다. 이렇게 하는 이유는 데이터를 손상시킬 수 있는 다중 노드에 디스크 그룹을 실수로 가져오지 않도록 하기 위해서입니다.

경고

ServiceGuard에서 패키지 제어 스크립트 프레임워크 내에 -C 옵션을 사용하더라도 이 옵션은 명령줄에 사용하지 마십시오. “문제 해결” 절에서 명령줄에 -C 옵션을 사용해야 할 상황을 설명합니다.

다음 예제는 명령과 함께 제어 스크립트에서 사용한 동일한 옵션을 나타낸 것입니다.

```
# vxdbg -tfC import dg_01
```

최근 디스크에 다른 호스트 ID가 기록된 경우라도 이 명령을 사용하면 dg_01 디스크 그룹의 모든 디스크에 대한 소유권을 가져옵니다. 이 명령은 dg_01 디스크 그룹에 있는 모든 디스크에 현재 노드의 호스트 ID를 기록하며 이 디스크에 대해 **noautoimport** 플래그를 설정합니다. 이 플래그는 다음 재부팅 시 디스크 그룹이 노드에 의해 자동으로 다시 가져오지 않도록 합니다. 클러스터의 한 노드에 장애가 발생하면 호스트 ID도 디스크 그룹의 각 디스크에 기록됩니다. 그러나, 노드가 ServiceGuard 클러스터에 포함되면 디스크 그룹을 ServiceGuard에서 제어할 수 없어도 재부팅 시 호스트 ID는 자동 가져오기 플래그가 설정된 모든 디스크의 노드에 의해 지워집니다. 이렇게 하면 디스크 그룹에 액세스하는 모든 클러스터 노드에서 디스크를 클러스터 작동의 일부로 가져올 수 있습니다. 제어 스크립트는 또한 vxvol startall 명령을 사용하여 가져온 각 디스크 그룹의 논리 볼륨을 시작합니다.

다수의 저장 장치 단위에 대한 패키지 최적화

다수의 파일 시스템이나 저장 그룹을 사용할 때 성능 향상을 가능하게 하도록 4개의 변수 집합이 제공됩니다. 자세한 내용은 제어 스크립트 템플릿의 설명을 참조하십시오. 다음과 같습니다.

- `CONCURRENT_VGCHANGE_OPERATIONS`—패키지 시작 중에 여러 개의 병렬 LVM 볼륨 그룹 활성화와 패키지 종료 중에 비활성화를 정의합니다.
- `CONCURRENT_DISKGROUP_OPERATIONS`—패키지 시작 중에 여러 개의 병렬 VxVM 볼륨 그룹 활성화와 패키지 종료 중에 비활성화를 정의합니다.
- `CONCURRENT_FSCK_OPERATIONS`—패키지 시작 시 수행될 여러 개의 병렬 fsck 작업을 정의합니다.
- `CONCURRENT_MOUNT_AND_UMOUNT_OPERATIONS`—패키지 시작 중에 여러 개의 병렬 마운트 작업과 패키지 종료 중에 언마운트 작업을 정의합니다.

패키지 제어 스크립트 템플릿 파일

다음은 패키지 제어 스크립트 템플릿 파일에서 인용한 내용입니다. 이 파일에는 VxVM 및 LVM 저장 그룹 활성화에 대한 별도의 절이 포함되어 있습니다.

```
# *****
# *
# *          HIGH AVAILABILITY PACKAGE CONTROL SCRIPT (template)          *
# *
# *          Note:This file MUST be edited before it can be used.          *
# *
# *****

# UNCOMMENT the variables as you set them.

# Set PATH to reference the appropriate directories.
PATH=/sbin:/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
# out the default, if your disks are mirrored on separate physical paths,
# and you want the mirror resynchronization to occur in parallel with
# the package startup.
```

패키지 및 서비스 구성

패키지 제어 스크립트 작성

```
#
# 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"
# VGCHANGE="vgchange -a y"

VGCHANGE='vgchange -a e' # Default

# CVM DISK GROUP ACTIVATION:
# Specify the method of activation for CVM disk groups.
# 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 cvm 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.
#
# FILESYSTEMS
# Specify the filesystems which are used by this package.Uncomment
# LV[0]=""; FS[0]=""; FS_MOUNT_OPT[0]=" and fill in the name of your first
# logical volume, filesystem and mount option for the file system.You must
# begin with LV[0], FS[0] and FS_MOUNT_OPT[0] and increment the list in
# sequence.
#
# For the LVM example, if this package uses the file systems pkg1a and
# pkg1b, which are mounted on the logical volumes lv01 and lv02 with
# read and write options enter:
#     LV[0]=/dev/vg01/lv01; FS[0]=/pkg1a; FS_MOUNT_OPT[0]="-o rw"
#     LV[1]=/dev/vg01/lv02; FS[1]=/pkg1b; FS_MOUNT_OPT[1]="-o rw"
#
#
# For the CVM or VxVM example, if this package uses the file systems
# pkg1a and pkg1b, which are mounted on the volumes lv01 and lv02
# with read and write options enter:
#     LV[0]=/dev/vx/dsk/dg01/lv01; FS[0]=/pkg1a; FS_MOUNT_OPT[0]="-o rw"
#     LV[1]=/dev/vx/dsk/dg01/lv02; FS[1]=/pkg1b; FS_MOUNT_OPT[1]="-o rw"
#
# The filesystems are defined as triplets of entries specifying the logical
# volume, the mount point and the mount options for the file system.Each

```

패키지 및 서비스 구성

패키지 제어 스크립트 작성

```
# 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.
#
#LV[0]=""; FS[0]=""; FS_MOUNT_OPT[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_UNMOUNT_COUNT=1

# FILESYSTEM MOUNT RETRY COUNT.
# Specify the number of mount retries 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

# IP ADDRESSES
# Specify the IP and Subnet address pairs which are used by this package.
# 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 -fi" to see the available subnets in the Network field.
#
```

```

# 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]=" "

# 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]=

```

패키지 및 서비스 구성

패키지 제어 스크립트 작성

위의 제어 스크립트 내용은 일련의 변수에 값을 지정하는 것을 보여줍니다. 나머지 스크립트는 이들 변수를 통해 논리 볼륨 관리자 명령, HP-UX 명령 및 `cmrunserv`, `cmmodnet`, `cmhaltserv`를 포함하는 `MC/ServiceGuard` 명령을 실행하여 패키지를 제어합니다. 논리의 흐름을 보려면 제어 스크립트 템플릿을 확인하십시오. 다음 명령을 사용합니다.

```
# cmmakepkg -s | more
```

주 함수는 스크립트의 맨 마지막에 나타납니다.

개별 변수가 선택 사항입니다. 따라서, 원하는 패키지 작업에 필요한 만큼의 변수만 사용해야 합니다. 예를 들어, 패키지가 볼륨 그룹을 활성화할 필요가 없는 경우 `VG` 변수를 생략하십시오. 그리고 패키지가 서비스를 사용하지 않을 경우 해당 `SERVICE_NAME`, `SERVICE_CMD`, `SERVICE_RESTART` 변수를 생략하십시오.

패키지 구성 파일에서 `EMS` 리소스를 `DEFERRED` 옵션으로 설정했다면 패키지 제어 스크립트에서 `DEFERRED_RESOURCE_NAME`을 정의해야 합니다.

`DEFERRED_RESOURCE_NAME` 매개 변수를 사용하여 지연된 리소스만 지정합니다.

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

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

스크립트를 사용자 정의한 후 `rcp`, `ftp` 또는 원하는 복사 방법을 사용하여 클러스터의 각 노드에 스크립트를 배포하십시오.

패키지 제어 스크립트에 사용자 정의 함수 추가

패키지가 시작되거나 중단될 때마다 실행될 패키지 제어 스크립트에 다른 셸 명령을 추가할 수 있습니다. 추가하려면 단지 스크립트의 `CUSTOMER DEFINED FUNCTIONS` 영역에 원하는 명령을 입력하면 됩니다. 이렇게 하면 제어 스크립트를 보다 폭넓게 사용자 정의할 수 있습니다.

그 아래에는 특수 파일에 패키지의 시작 및 중지를 기록하기 위한 `date` 및 `echo` 명령을 포함한 스크립트의 예제가 나옵니다.

```
# 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 사용자 정의 함수 영역에 ServiceGuard 명령(cmmmodpkg 등)을 추가할 수 있습니다. 하지만 이 명령을 패키지에 직접 사용해서는 안됩니다. 또한 ServiceGuard 명령을 다른 패키지에 직접 사용할 경우 ServiceGuard 명령을 사용하는 모든 패키지에서 명령 루프가 발생할 가능성이 있는지 검사해야 합니다.

예를 들어, 다음 상황에서 명령 루프가 발생할 수 있습니다. Pkg1이 Pkg2의 cmmmodpkg -d를 실행하고 Pkg2가 Pkg1의 cmmmodpkg -d를 실행한다고 가정합니다. Pkg1과 Pkg2가 동시에 시작될 경우 Pkg1은 Pkg2의 cmmmodpkg를 실행하려고 시도합니다. 하지만 cmmmodpkg 명령은 Pkg2의 실행이 끝날 때까지 기다려야 합니다. 마찬가지로 Pkg2는 Pkg1의 cmmmodpkg를 실행하려고 시도하지만 Pkg2는 Pkg1의 실행이 완료될 때까지 기다려야 하기 때문에 명령 루프가 발생합니다.

명령 루프를 방지하려면 특히 제어 스크립트에서 ServiceGuard 명령을 사용하는 패키지를 포함한 모든 패키지에 대해 항상 RUN_SCRIPT_TIMEOUT 및 HALT_SCRIPT_TIMEOUT를 지정하는 것이 좋습니다. 제한 시간을 지정하지 않을 경우 위에서 설명한 것처럼 구성에 명령 루프가 있으면 클러스터 증지를 포함한 불안정한 상태가 발생할 수 있습니다.

추가 제품 지원

패키지 제어 스크립트를 통해 MetroCluster with Continuous Access/CA, MetroCluster with EMC SRDF 및 HA NFS 톨킷 등의 추가 제품을 사용할 수 있습니다. 제어 스크립트에서 제공되는 후크를 사용하여 패키지를 만드는 방법에 대한 자세한 사항은 추가 제품 설명서를 참조하십시오.

패키지 구성 확인

SAM은 입력된 구성을 자동으로 확인한 다음 오류를 보고합니다. ASCII 패키지 구성 파일을 편집하는 경우 다음 명령을 사용하여 파일 내용을 확인합니다.

```
# cmcheckconf -k -v -P /etc/cmcluster/pkg1/pkg1.config
```

표준 출력 결과에 오류가 표시됩니다. 필요하다면 파일을 편집하여 오류를 수정한 다음, 오류 발생 없이 완료될 때까지 명령을 다시 실행합니다.

cmcheckconf 명령을 실행하면 다음 사항을 검사합니다.

- 패키지 이름이 유효한지, 하나 이상의 NODE_NAME 항목이 포함되어 있는지
- 중복된 매개 변수 항목이 없는지
- 매개 변수 값이 허용된 범위 이내에 있는지
- 실행 및 중지 스크립트가 클러스터의 모든 노드에서 종료되는지, 그리고 실행 가능한지
- 실행 및 중지 스크립트 시간 제한이 4294초보다 작은지
- 구성된 리소스가 클러스터 노드에서 사용 가능한지

구성 배포

SAM 또는 HP-UX 명령을 사용하여 이진 클러스터 구성 파일을 클러스터의 노드로 배포할 수 있습니다.

SAM을 사용한 구성 파일 및 제어 스크립트 배포

SAM의 패키지 구성 하위 영역에서 패키지 작성을 완료하면 클러스터의 모든 노드로 파일을 복사할 것인지 묻는 확인 메시지가 표시됩니다. 이 때 OK를 누르면 MC/ServiceGuard는 이진 구성 파일과 패키지 제어 스크립트를 클러스터의 모든 노드에 복사합니다.

HP-UX 명령을 사용한 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 등)을 사용하여 복사하십시오. 예를 들어, 다음과 같이 *ftsyst9*에서 rcp 명령을 실행하여 *ftsyst10*에 패키지 제어 스크립트를 복사할 수 있습니다.

```
# rcp /etc/cmcluster/pkg1/control.sh ftsyst10:/etc/cmcluster/pkg1/control.sh
```

HP-UX 명령을 사용한 이진 클러스터 구성 파일 배포

ASCII 클러스터 및 패키지 구성 파일이 있는 노드에서 다음 단계를 수행합니다.

- 구성 파일이 올바른지 확인합니다. 다음 명령을 사용합니다.

```
# 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 작성하여 각 노드에 배포합니다.

```
# cmapplyconf -v -C /etc/cmcluster/cmcl.config -P \  
/etc/cmcluster/pkg1/pkg1.config
```
- 디스크 잠금을 사용할 경우 클러스터 잠금 볼륨 그룹을 비활성화합니다.

```
# vgchange -a n /dev/vg01
```

`cmapplyconf` 명령은 클러스터 구성 파일의 이전 버전을 작성하여 클러스터의 모든 노드에 배포합니다. 이렇게 하면 모든 노드에서 파일의 내용이 일관된 상태를 유지합니다.

주

`cmcheckconf` 및 `cmapplyconf`는 클러스터 및 패키지 구성 파일이 변경될 때마다 다시 실행해야 합니다.

클러스터 및 패키지 작동 테스트

MC/ServiceGuard 클러스터를 구성하는 동안 장애가 발생하는 경우 클러스터의 여러 구성 요소가 제대로 작동하는지 테스트하는 것이 좋습니다. 패키지, 노드 또는 LAN에 장애가 발생했을 때 클러스터가 올바르게 응답하는지 테스트하는 방법에 대한 자세한 내용은 “클러스터 문제점 해결”을 참조하십시오.

패키지 및 서비스 구성
구성 배포

7

클러스터 및 패키지 유지 관리

이 장에서는 `cmviewcl` 명령을 설명한 다음 클러스터나 개별 노드를 시작하고 중지하는 방법, 영구적인 재구성을 수행하는 방법, 클러스터의 일상적인 유지 관리 중에 패키지를 시작, 중지, 이동, 수정하는 방법을 설명합니다. 다음과 같은 내용을 다룹니다.

- `cmviewcl` 명령을 사용하여 클러스터 및 패키지 상태 확인
- ServiceGuard Manager 사용
- 클러스터 및 노드 관리
- 패키지 및 서비스 관리
- 클러스터 재구성
- 패키지 재구성
- 클러스터 이벤트에 대응
- 시스템에서 MC/ServiceGuard 제거

모든 관리 작업은 ServiceGuard Manager, SAM 또는 HP-UX 명령을 사용하여 수행할 수 있습니다.

cmviewcl 명령을 사용하여 클러스터 및 패키지 상태 확인

클러스터 상태에 대한 정보는 클러스터의 각 개별 노드에서 유지 관리되는 상태 데이터베이스에 저장됩니다. cmviewcl 명령을 실행하여 이 데이터베이스에 포함된 정보를 표시할 수 있습니다.

cmviewcl -v

-v 옵션을 사용하여 이 명령을 실행하면, 실행 중인 클러스터의 모든 노드와 패키지에 대한 정보가 표시되며, 장애 조치 작동을 결정하는 매개 변수의 설정 값이 표시됩니다.

팁

일부 명령은 구성이 큰 경우에는 완료하는 데 오래 걸립니다. 특히, ServiceGuard의 CPU 활용 프로그램을 사용할 경우 패키지 및 서비스 수가 늘어남에 따라 cmviewcl -v 실행 시간이 늘어납니다.

사용자가 원하는 릴리즈 포맷을 표시하는 -r 옵션을 사용하여, 이전에 제공된 특정 릴리즈에서 포맷된 방식으로 출력 결과를 포맷할 수 있습니다. 예를 들면 다음과 같습니다.

cmviewcl -r A.11.12

cmviewcl 옵션에 대한 자세한 내용은 온라인 도움말을 참조하십시오.

클러스터 및 패키지 상태 종류

클러스터나 그 구성 요소 노드는 시간에 따라 여러 다른 상태로 변할 수 있습니다. 다음 절은 클러스터나 패키지의 일반적인 여러 조건에 대해 설명합니다.

클러스터의 상태

클러스터의 **상태**는 다음 중 하나일 수 있습니다.

- **Up:** 하나 이상의 노드에서 클러스터 데몬이 실행 중이고 재구성은 수행되고 있지 않습니다.
- **Down:** 모든 클러스터 노드에서 클러스터 데몬이 실행되고 있지 않습니다.
- **Starting:** 클러스터는 활성 구성원을 식별하는 과정 중에 있습니다. 최소한 하나 이상의 클러스터 데몬이 실행 중입니다.
- **Unknown:** cmviewcl 명령이 실행된 노드는 클러스터의 다른 노드와 통신할 수 없습니다.

노드의 상태 및 상황

클러스터 데몬이 실행 중인지의 여부에 따라 노드의 **상태**는 up(클러스터의 구성원으로서 *active*) 또는 down(클러스터에서 *inactive*)이 됩니다. 클러스터의 관점에서는 노드가 중지된 경우이지만 HP-UX는 계속 실행되고 있음에 주의하십시오.

또한, 노드는 다음 중 하나의 상태일 수 있습니다.

- **Failed:** 노드는 자체를 인식할 수 없습니다. 노드가 현재는 아니지만 이전에 활성 클러스터에 있었고 중지되지 않은 경우 클러스터의 다른 활성 구성원들은 노드를 인식할 수 있습니다.
- **Reforming:** 클러스터 재편성 시의 노드 상태입니다. 이 노드에서는 모든 노드가 활성 클러스터의 새 구성원에 동의하도록 하는 프로토콜을 현재 실행 중입니다. 동의하면 상태 데이터베이스는 새로운 클러스터 구성원을 반영하도록 업데이트됩니다.
- **Running:** 노드는 마지막 재편성에 필요한 모든 작업을 완료하고 정상적으로 작동되고 있습니다.
- **Halted:** 노드는 자체를 인식할 수 없습니다. 노드가 정상적으로 활성 클러스터를 떠난 다음에(예: cmhaltnode 명령을 사용하여) 다른 노드들이 이 노드를 인식할 수 있습니다.
- **Unknown:** 노드는 자체를 인식할 수 없습니다. 이전에 클러스터의 구성원이 된 적이 없는 노드는 다른 노드에 의해 알 수 없는 상태로 지정됩니다.

패키지의 상태 및 상황

패키지는 다음 **상태** 중 하나일 수 있습니다.

- **Up:** 패키지 제어 스크립트가 활성화되어 있습니다.
- **Down:** 패키지 제어 스크립트가 활성화되어 있지 않습니다.
- **Unknown:**

패키지의 **상태**는 다음 중 하나일 수 있습니다.

- **Starting:** 제어 스크립트의 시작 명령이 실행 중입니다.
- **Running:** 서비스가 활성화되어 있고 모니터링 중입니다.
- **Halting:** 제어 스크립트의 중지 명령이 실행 중입니다.

패키지의 전환 속성

또한, 패키지의 전환 속성은 다음과 같습니다.

- 패키지 스위칭: **Enabled**는 장애가 발생할 경우 패키지가 다른 노드로 전환될 수 있음을 표시합니다.
- 노드에서 스위칭 사용 설정: **Enabled**는 패키지가 참조 노드로 전환할 수 있음을 표시합니다. **Disabled**는 `cmmodpkg` 명령을 사용하여 노드에서 패키지를 사용할 수 있게 될 때까지 패키지는 지정된 노드로 전환할 수 없음을 표시합니다.

패키지의 기본 노드이거나 대체 노드인 각 노드에 대해 모든 패키지는 **Enabled** 또는 **Disabled**로 표시됩니다.

서비스의 상태

서비스의 상태는 다음 중 하나입니다.

- **Up**: 서비스를 모니터링 중입니다.
- **Down**: 서비스가 실행되고 있지 않습니다. 서비스가 중지되었거나 장애가 발생했을 수 있습니다.
- **Uninitialized**: 서비스가 클러스터에 구성되어 있으나, 제어 스크립트에서 실행 명령을 실행했는데도 서비스가 시작되지 않습니다.
- **Unknown**:

네트워크의 상태

네트워크 인터페이스의 상태는 다음 중 하나입니다.

- **Up**:
- **Down**:
- **Unknown**: 인터페이스가 실행 중인지 중지되었는지 확인할 수 없습니다. 이 상태는 클러스터가 중지되었을 때 발생할 수 있습니다. 대기 인터페이스의 상태는 **unknown**입니다.

직렬 회선의 상태

직렬 회선의 상태는 다음 중 하나입니다.

- **Up**: 직렬 회선을 통해 하트비트가 전달됩니다.

- **Down:** 하트비트를 NODE_TIMEOUT 값의 두 배인 시간 동안 직렬 회선을 통해 받을 수 없을 때 발생합니다.
- **Recovering:** 직렬 회선을 통해 손상된 메시지를 전달받았으며, 직렬 회선은 재동기화 작업이 진행 중입니다.
- **Unknown:** 직렬 회선의 사용 가능 여부를 확인할 수 없습니다. 이 상태는 원격 노드가 중지되었을 때 발생할 수 있습니다.

장애 조치 및 장애 복구 정책

패키지는 FAILOVER_POLICY 매개 변수의 두 값 중 하나를 사용하여 구성할 수 있습니다.

- **CONFIGURED_NODE:** 패키지는 패키지 구성 파일의 노드 목록에서 그 다음에 위치하는 노드로 장애 조치를 수행합니다.
- **MIN_PACKAGE_NODE:** 패키지는 실행 중인 패키지 수가 가장 작은 클러스터의 노드로 장애 조치를 수행합니다.

패키지는 또한 FAILBACK_POLICY 매개 변수의 두 값 중 하나를 사용하여 구성할 수 있습니다.

- **AUTOMATIC:** 패키지는 장애 조치를 수행한 다음 기본 노드가 다시 사용할 수 있게 되면 기본 노드로 되돌아갑니다.
- **MANUAL:** 장애 조치를 수행한 다음 시스템 관리자가 직접 패키지를 원노드로 다시 이동시켜야 합니다.

장애 조치 및 장애 복구 정책은 `cmviewcl -v` 명령을 사용하여 볼 수 있습니다.

클러스터 및 패키지 상태의 예제

`cmviewcl -v` 명령의 다음 출력 결과는 예제를 위해 구성된 클러스터의 상태를 보여줍니다.

정상 실행 중일 때의 상태

클러스터의 두 노드가 실행 중이며 패키지가 기본 노드에 있는 정상적인 실행 모습을 보여줍니다.

```

CLUSTER      STATUS
example      up
  NODE      STATUS      STATE
  ftsys9    up            running

Network_Parameters:
INTERFACE    STATUS      PATH      NAME
PRIMARY      up          56/36.1   lan0
    
```

클러스터 및 패키지 유지 관리

cmviewcl 명령을 사용하여 클러스터 및 패키지 상태 확인

```
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           service1
Subnet        up          0             0           15.13.168.0

Node_Switching_Parameters:
NODE_TYPE     STATUS      SWITCHING     NAME
Primary       up          enabled       ftsys9      (current)
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     ftsys10

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    (current)
Alternate     up          enabled       ftsys9
```

Quorum 서버 상태

클러스터에서 서비스를 재편성하기 위해 quorum 서버를 사용할 경우 아래 cmviewcl -v 출력 내용에서처럼 각 노드 아래에 서버 이름, 상태 및 상황이 차례로 표시됩니다.

```
CLUSTER      STATUS
example      up

NODE          STATUS      STATE
```

```

ftsys9      up      running

Quorum Server Status:
NAME        STATUS      STATE
lp-gs       up          running
...

NODE        STATUS      STATE
ftsys10     up          running

Quorum Server Status:
NAME        STATUS      STATE
lp-gs       up          running

```

CVM 패키지 상태

클러스터에서 디스크 저장에 VERITAS 클러스터 볼륨 관리자를 사용할 경우 시스템의 다중 노드 패키지 CVM-VxVM-pkg는 CVM 디스크 그룹에 액세스할 응용 프로그램의 모든 활성 노드에서 실행 중이어야 합니다. 다음 cmviewcl 명령 출력은 이 패키지를 보여 줍니다.

```

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 아래에 나타난 대로 클러스터에서 각 활성 노드와 관련된 시스템 다중 노드 패키지가 표시됩니다.

```

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 명령을 사용하여 pkg2를 중지한 이후 cmviewcl -v의 출력 결과는 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          service1
Subnet      up      0              0          15.13.168.0
Resource    up          /example/float

Node_Switching_Parameters:
NODE_TYPE    STATUS    SWITCHING    NAME
Primary      up        enabled      ftsys9      (current)
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     unowned

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:

```
NODE_TYPE  STATUS  SWITCHING  NAME
Primary    up      enabled    ftsys10
Alternate  up      enabled    ftsys9
```

현재 pkg2는 “중지” 상태이고 패키지 전환을 할 수 없도록 소유되지 않은 상황으로 표시 됩니다. pkg2에 의존하도록 구성된 “/example/float” 리소스는 한 노드에서 중지됩니다. 하지만, 두 노드 모두에서 switching은 enabled가 됩니다. 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 nv 명령의 출력 결과는 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          servicel
Subnet    up      0              0          15.13.168.0
```

클러스터 및 패키지 유지 관리

cmviewcl 명령을 사용하여 클러스터 및 패키지 상태 확인

```

Resource      up                                     /example/float

Node_Switching_Parameters:
NODE_TYPE     STATUS      SWITCHING   NAME
Primary       up          enabled     ftsys9      (current)
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      (current)

NODE          STATUS      STATE
ftsys10      up          running

Network_Parameters:
INTERFACE     STATUS      PATH         NAME
PRIMARY      up          28.1         lan0
STANDBY      up          32.1         lan1

```

현재 pkg2는 노드 ftsys9에서 실행 중입니다. 여전히 전환은 사용 해제되어 있습니다.

패키지 전환이 사용 설정된 후의 상태

다음 명령은 패키지의 상태를 다시 Auto Run Enabled로 변경합니다.

```
# cmmodpkg -e pkg2
```

cmviewcl 명령의 출력 결과는 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 모두 ftsys9에서 실행되고 있으며 pkg2는 전환이 가능합니다. Ftsys10은 데몬을 실행 중이고 ftsys10에서 실행 중인 패키지는 없습니다.

노드 중지 후의 상태

다음 명령을 사용하여 *ftsys10*을 중지하십시오.

```
# cmhaltnode ftsys10
```

cmviewcl의 출력 결과는 *ftsys9*에서 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* 및 *ftsys10* 모두에 표시됩니다.

RS232의 상태 보기

하트비트 연결로 직렬(RS232) 회선을 사용할 경우 cmviewcl -v 명령을 사용하여 일련의 구성된 RS232 장치 파일을 볼 수 있습니다. 다음은 정상적인 실행 상태를 보여줍니다.

```
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/tty0p0

      NODE      STATUS    STATE
      ftsys10   down     running

      Network_Parameters:
      INTERFACE  STATUS    PATH      NAME
      PRIMARY   up        28.1     lan0

      Serial_Heartbeat:
```

클러스터 및 패키지 유지 관리

cmviewcl 명령을 사용하여 클러스터 및 패키지 상태 확인

```
DEVICE_FILE_NAME    STATUS    CONNECTED_TO:
/dev/tty0p0         unknown  ftsys9 /dev/tty0p0
```

다음은 노드 *ftsys10*이 중지된 후의 상태를 보여줍니다.

```
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/tty0p0
  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 중지되었을 때의 상태를 보여줍니다.

```
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
```

비소유 패키지의 데이터 보기

다음 예제는 현재 소유되지 않은, 즉 구성된 어떠한 노드에서도 실행되고 있지 않은 패

키지를 보여줍니다. 모니터링 중인 리소스에 대한 정보는 이 리소스를 실행할 수 있는 패키지가 있는 각 노드에 제공됩니다. 이 정보로 장애의 원인을 식별하고 패키지가 다시 시작될 위치를 결정할 수 있습니다.

UNOWNED_PACKAGES

PACKAGE	STATUS	STATE	AUTO_RUN	NODE
PKG3	down	halted	enabled	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 패키지의 데이터 보기

다음 예제는 표준 ServiceGuard 패키지 뿐만 아니라 시스템 다중 노드 패키지를 포함하는 클러스터를 보여줍니다. 시스템 다중 노드 패키지는 클러스터의 모든 노드에서 실행되지만 표준 패키지는 한 번에 하나의 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

클러스터 및 패키지 유지 관리

cmviewcl 명령을 사용하여 클러스터 및 패키지 상태 확인

```
SYSTEM_MULTI_NODE_PACKAGES:
```

PACKAGE	STATUS	STATE
VxVM-CVM-pkg	up	running

ServiceGuard Manager 사용

ServiceGuard Manager는 ServiceGuard 클러스터를 관리하는 그래픽 사용자 인터페이스입니다. ServiceGuard Manager를 사용하여 실행 중인 클러스터에서 받은 데이터를 표시하거나 또는 저장된 데이터 파일을 볼 수 있습니다. ServiceGuard Manager는 HP-UX 및 Windows NT 시스템에서 구현됩니다. 옵션은 대화 상자나 명령줄에서 선택할 수 있습니다.

ServiceGuard Manager 설치에 대한 자세한 내용은 최신 *ServiceGuard Manager Release Notes*를 참조하십시오. ServiceGuard Manager는 <http://software.hp.com>에서 무료로 다운로드할 수 있습니다.

명령을 사용하여 ServiceGuard Manager 실행

HP-UX 또는 Windows NT DOS 창의 명령줄에서 ServiceGuard Manager를 시작하려면 다음 구문을 사용하십시오.

```
SGMgr [-s server [-l username [-p password]] | -f filename ]
      [-c clustername...]
```

표 7-1은 다양한 옵션에 대한 사용 설명을 나타냅니다.

표 7-1

ServiceGuard Manager 명령줄 옵션

옵션	설명
-s server	개체 관리자를 실행하고 있는 서버의 이름을 지정합니다. 이 개체 관리자는 사용자가 관찰하려는 클러스터에 대한 데이터를 모읍니다. 이 옵션이나 -f filename 옵션 중 하나를 선택할 수 있습니다. 그러나 두 옵션 모두를 선택해서는 안됩니다.
-l username	-s server 옵션을 사용하여 지정된 서버의 로그인 이름
-p password	위에서 지정된 사용자 이름과 서버의 암호. 먼저 서버 이름을 지정해야만 이 옵션을 사용할 수 있습니다. 서버 이름을 입력하고 이 옵션은 생략하면, 로그인 이름과 암호를 묻는 메시지가 나타납니다.
-c clustername...	사용자가 보고 싶어하는 클러스터의 이름
-f filename.	서버에 연결된 파일이 아니라 사용자가 보고 싶어하는 저장된 데이터를 포함하는 파일의 이름. 이 옵션이나 -s server 옵션 중 하나를 선택할 수 있습니다.

클러스터 및 패키지 유지 관리 ServiceGuard Manager 사용

예를 들어, ServiceGuard Manager를 기본 디렉터리에 설치할 때, 저장된 개체(.sgm) 파일인 4clusters.sgm 파일을 열려면 다음 명령을 입력합니다.

```
sgmgr -f C:\Program Files\Hewlett  
Packard\ServiceGuardManager\examples\4clusters.sgm
```

자세한 구문 정보에 대해서는 sgmgr (1M) 온라인 도움말을 참조하십시오.

ServiceGuard Manager의 작동 방식

ServiceGuard Manager는 관리 중인 모든 클러스터 노드와 연결된 개체 관리자 데몬으로부터 클러스터 데이터를 받아 구성 데이터를 획득합니다. 개체 관리자 데몬은 ServiceGuard Manager와 다른 시스템에 상주할 수 있습니다.

ServiceGuard Manager를 사용하려면 먼저 개체 관리자 서버에 로깅한 다음 표시할 클러스터나 노드를 지정합니다. 만약 보안 요구 사항이 충족된다면 개체 관리자는 지정된 클러스터나 노드로부터 데이터를 획득할 수 있습니다. 성공적으로 개체 관리자를 연결하기 위해서는 ServiceGuard Manager에서 표시할 노드의 보안 파일(/etc/cmclnodelist 또는 .rhosts)에 개체 관리자 호스트 이름을 입력해야 합니다. 이 파일을 편집하는 방법에 대한 자세한 내용은 5장의 “시스템 준비” 절을 참조하십시오.

특정 클러스터를 사용하여 ServiceGuard Manager 시작

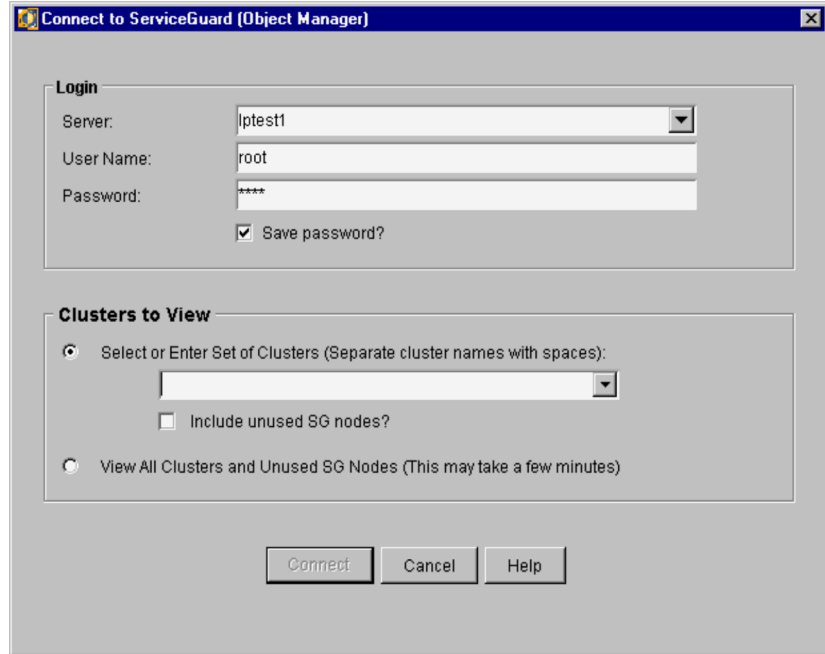
ServiceGuard Manager를 시작하고 특정 클러스터를 지정하려면 다음 단계에 따라 진행하십시오. 가장 신속하게 표시할 수 있는 방법으로 이 방식을 권장합니다.

- HP-UX에서는 터미널 창에 다음 명령을 입력합니다.

```
# /opt/sgmgr/bin/sgmgr -c <clustername>
```
- Windows NT에서는
 - 바탕 화면의 ServiceGuard Manager 아이콘을 마우스 오른쪽 단추로 누릅니다.
 - 등록 정보를 누릅니다.
 - 명령줄의 끝에 “-c <clustername>” 문자열을 추가합니다.
 - 등록 정보 대화 상자를 종료합니다.
 - ServiceGuard Manager 아이콘을 두 번 누릅니다.

ServiceGuard Manager가 시작되면, ServiceGuard Manager 스플래쉬 화면에 프로그램 초기화 화면이 나타납니다. 그러면 그림 7-1과 같은 개체 관리자 로그인 화면이 나타납니다.

그림 7-1 개체 관리자 로그인 화면



아래에 있는 “개체 관리자 연결” 부분으로 가십시오.

특정 클러스터를 사용하지 않고 ServiceGuard Manager 시작

특정 클러스터를 지정하지 않고 ServiceGuard Manager를 시작하려면 다음 단계를 수행하십시오. 이 방법은 서브넷을 통해 클러스터를 검색합니다. 클러스터가 다수 구성되어 있는 경우에는 검색하는 데 시간이 걸립니다.

- HP-UX에서는 터미널 창에 다음 명령을 입력합니다.

```
# /opt/sgmgr/bin/sgmgr
```

- Windows NT에서는 다음 중 하나를 선택합니다.
 - 바탕 화면의 sgmgr 아이콘을 두 번 누릅니다.
 - PC 시작 메뉴에서 다음 순서대로 선택합니다.

클러스터 및 패키지 유지 관리 ServiceGuard Manager 사용

프로그램 -> ServiceGuard Manager -> sgmgr

— “내 컴퓨터”에서

1. 다음 파일로 이동합니다.

C:\Program Files\Hewlett-Packard\ServiceGuard Manager\bin

2. sgmgr.exe를 두 번 누릅니다.

— DOS 창에서 다음 명령을 입력합니다.

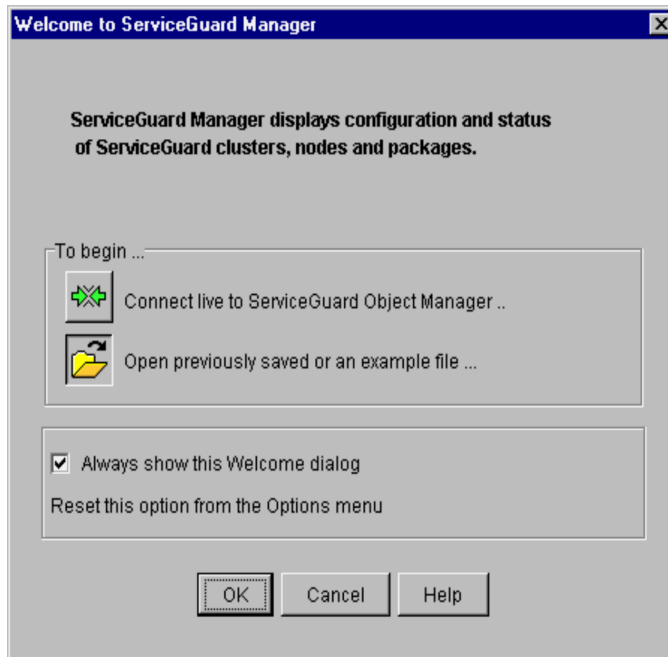
```
C: > CD C:\Program Files\Hewlett-Packard\ServiceGuard Manager\bin
```

```
C: > sgmgr
```

ServiceGuard Manager가 시작되면, ServiceGuard Manager 스플래시 화면이 간단히 나타납니다. 그러면 그림 7-2와 같이 시작 화면이 나타납니다.

그림 7-2

ScrServiceGuard Manager 시작 화면



이 화면에서는 다음 중 하나를 선택할 수 있습니다.

- ServiceGuard 개체 관리자에 연결하여 실행 중인 클러스터에서 받은 데이터를 봅니다.
- 저장된 파일을 열어, 이전에 저장된 데이터를 봅니다.

개체 관리자 연결

클러스터 노드의 ServiceGuard 개체 관리자 데몬은 실행 중인 클러스터 정보를 ServiceGuard Manager에게 제공합니다. 실행 중인 클러스터의 데이터를 보려면, MC/ServiceGuard 11.12를 실행 중인 노드에 로그인하거나 개체 관리자 데몬을 실행한 다음, 개체 관리자에 연결해야 합니다. 그림 7-3에서는 로그인 화면을 보여줍니다.

그림 7-3

개체 관리자 로그인 화면

The screenshot shows a dialog box titled "Connect to ServiceGuard (Object Manager)". It contains a "Login" section with fields for "Server" (set to "lptest1"), "User Name" (set to "root"), and "Password" (masked with "****"). There is a checked checkbox for "Save password?". Below this is a "Clusters to View" section with two radio button options. The first option, "Select or Enter Set of Clusters (Separate cluster names with spaces):", is selected and has a dropdown menu below it. There is also an unchecked checkbox for "Include unused SG nodes?". The second option is "View All Clusters and Unused SG Nodes (This may take a few minutes)". At the bottom of the dialog are three buttons: "Connect", "Cancel", and "Help".

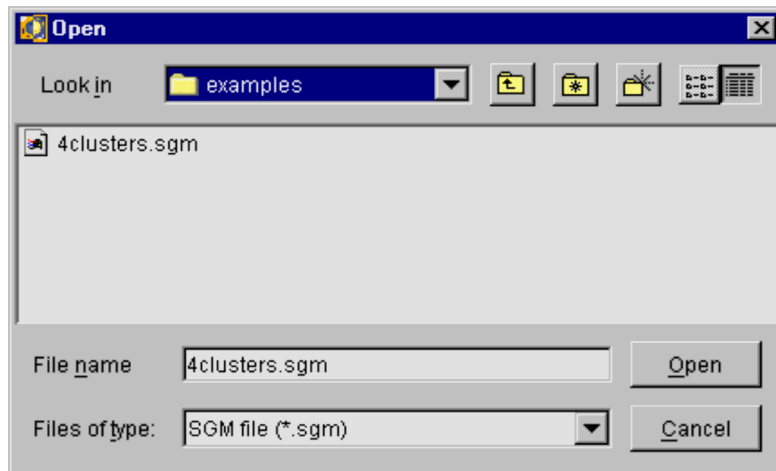
개체 관리자 노드 이름, 사용자 이름 및 암호를 입력하십시오. 유효한 사용자는 개체 관리자 서버의 `/etc/cmclnodelist` 또는 `.rhosts` 파일에 표시됩니다.

주 관리 기능을 수행하기 위해 **루트**로 로그인해야 합니다.

클러스터 데이터에 저장된 파일 열기

시작 화면의 두 번째 옵션을 선택하면, 저장된 데이터의 파일을 열 수 있습니다. 그림 7-4에서처럼 저장된 파일 목록을 볼 수 있습니다.

그림 7-4 ServiceGuard Manager 파일 열기 화면



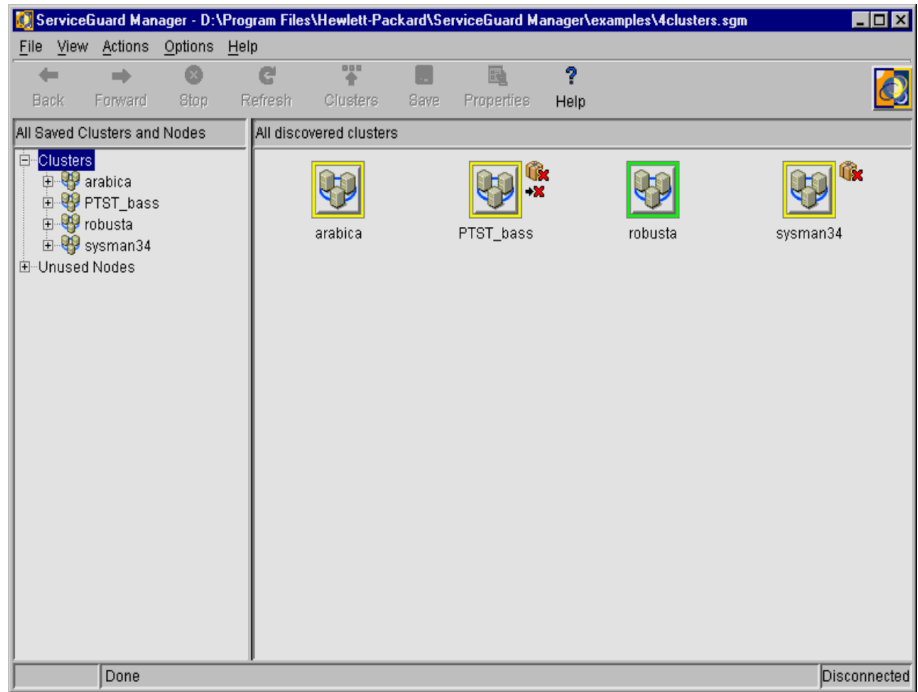
원하는 파일을 선택한 다음 “Open”을 누르십시오.

클러스터 데이터 보기

개체 관리자 노드에 연결되어 있는지 여부나 저장된 파일에서 받은 데이터를 표시하는지 여부에 관계 없이 초기 맵과 트리 계층 구조 표시를 볼 수 있습니다. 이 화면은 그림 7-5에서 설명합니다.

그림 7-5

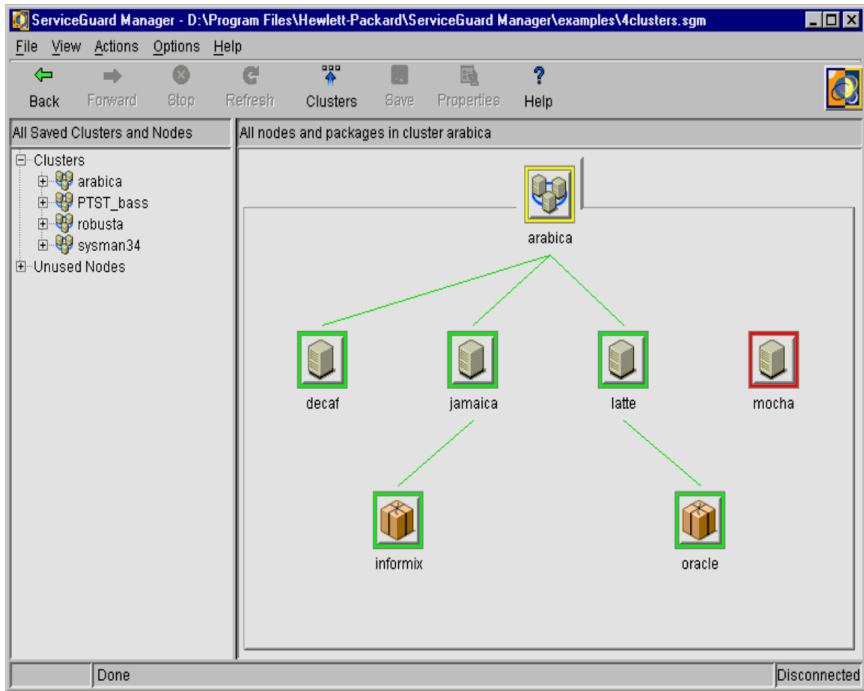
맵과 클러스터 목록 화면



특정 클러스터에 있는 개체를 표시하려면 클러스터 아이콘을 두 번 누릅니다. 그러면 그림 7-6의 화면과 같은 화면이 나타납니다.

그림 7-6

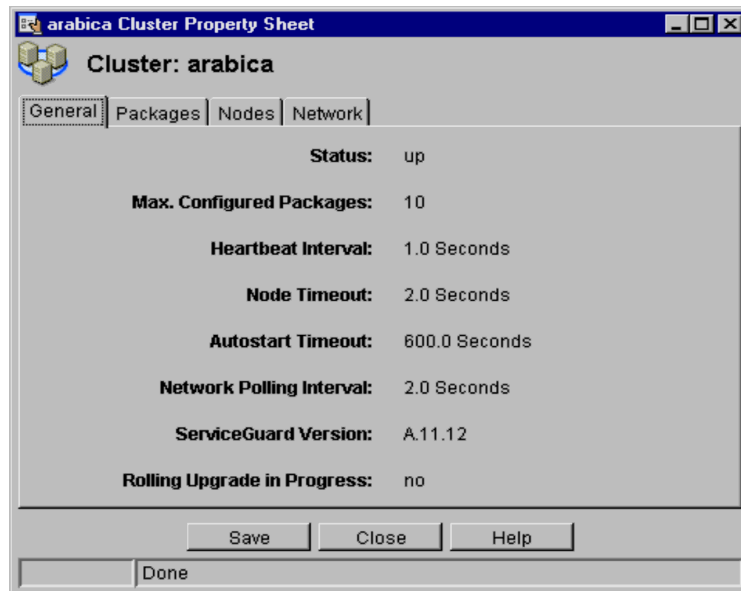
클러스터 세부 정보 화면



개체 속성 시트를 표시하려면 개체를 누르고 도구 모음에서 “Properties”를 누릅니다. 클러스터 속성 시트가 그림 7-7에 나옵니다.

그림 7-7

클러스터 속성 시트 화면

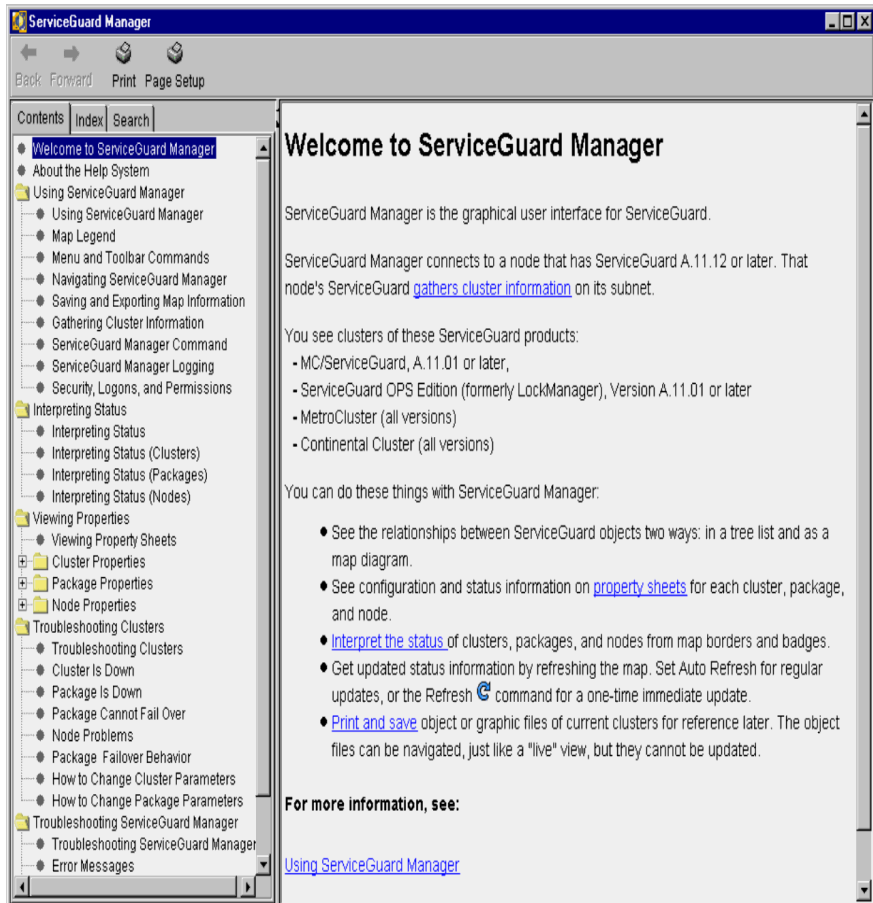


도움말 보기

온라인 도움말을 보려면 화면 상단의 메뉴 항목 "Help" 메뉴를 누릅니다. 그림 7-8에서처럼 처음 도움말 화면이 나타납니다.

그림 7-8

도움말 화면



이 인터페이스를 처음 사용하는 사용자에게는 “ServiceGuard Manager 사용” 절에 언급된 다음 도움말 항목이 특히 유용할 것입니다.

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

클러스터 개체 관리

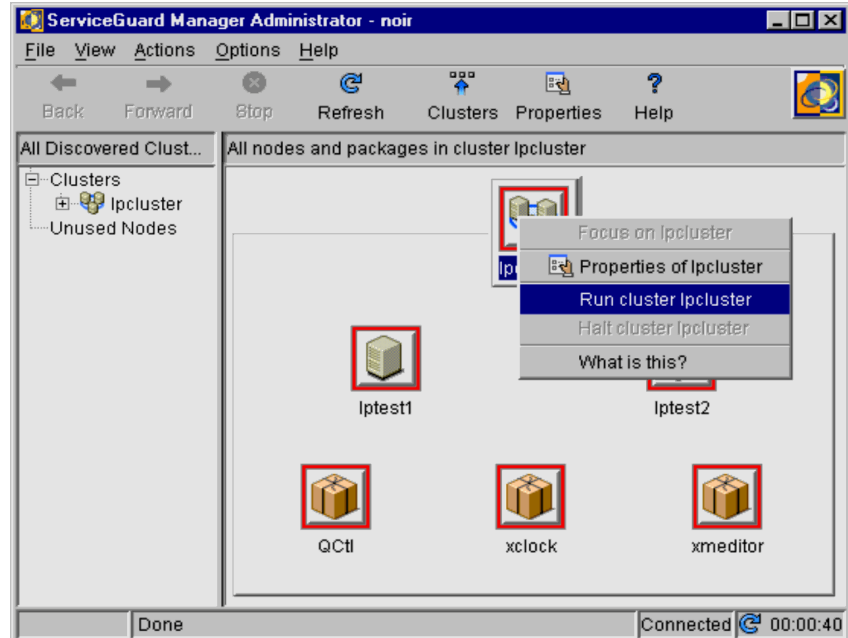
ServiceGuard Manager를 사용하여 클러스터, 노드 및 패키지를 조작할 수 있습니다. 모든 패키지를 비롯하여 클러스터를 시작 및 중지하고, 클러스터 작업으로 각 노드를 가져오

거나 가져나가고, 개별 패키지를 시작 또는 중지하고, 노드 간에 패키지를 이동할 수 있습니다. 관리 작업을 수행하려면 클러스터 개체를 선택하고 Actions 메뉴를 표시하거나 클러스터 개체를 마우스 오른쪽 단추로 눌러 가능한 작업을 포함한 메뉴를 표시합니다. 특정 작업이 클러스터 개체에 적합하지 않으면 그 작업은 회색으로 표시됩니다.

그림 7-9는 클러스터 관리 예를 나타낸 것입니다. 클러스터 아이콘을 마우스 오른쪽 단추로 누르면 작업 메뉴가 표시됩니다. **Start the Cluster**를 누르면 ServiceGuard Manager는 클러스터를 시작하는 명령을 실행합니다.

그림 7-9

클러스터 맵에 동작 메뉴 표시



GUI는 클러스터를 시작하기 위해 실행되고 있는 ServiceGuard 명령을 표시하는 진행 상태를 표시합니다(그림 7-10).

그림 7-10

ServiceGuard Manager의 클러스터 시작 명령

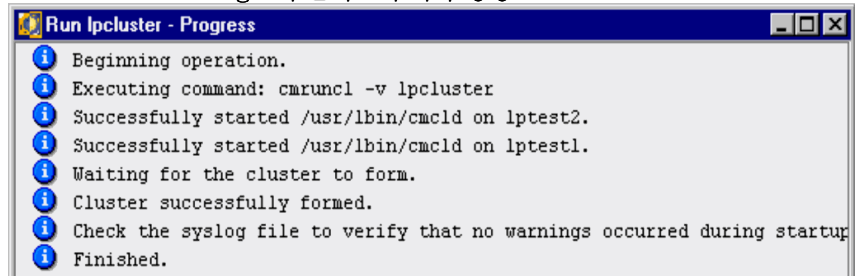
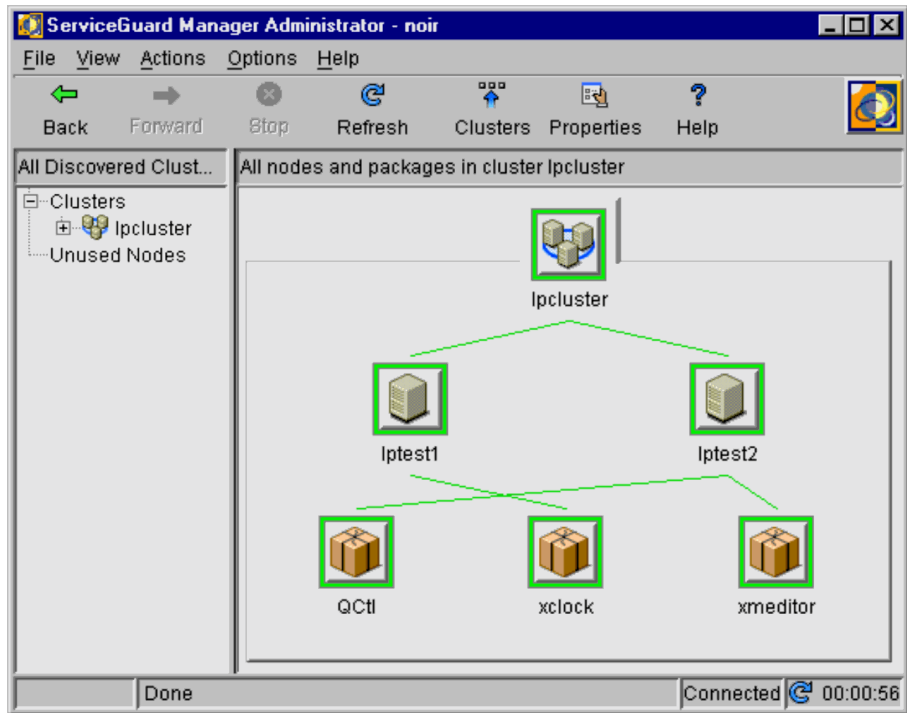


그림 7-11은 결과 즉, 실행 클러스터를 나타낸 것입니다.

그림 7-11

실행 클러스터



개체를 선택하고 동작 메뉴를 활성화하여 ServiceGuard Manager로 다음과 같은 클러스터 관리 작업을 수행할 수 있습니다.

- 클러스터 시작
- 클러스터 중지
- 이전에 구성한 노드를 클러스터에 추가
- 클러스터 노드 중지
- 패키지 시작
- 패키지 중지
- 노드 간에 패키지 이동
- 패키지와 노드에 대한 전환 매개 변수 재설정

주의

ServiceGuard Manager 클라이언트 사용자가 관리 작업을 수행하려면 루트 암호를 사용하여 루트로 개체 관리자 시스템에 로그인해야 합니다. 이렇게 하면 개체 관리자가 제공한 모든 클러스터를 관리하는 기능이 효율적으로 클라이언트에게 부여됩니다. 일부 사용자가 클러스터를 관리할 필요가 없다면 그러한 사용자는 개체 관리자 시스템에 일반 사용자로 로그인하도록 할 수 있습니다. 이 경우 이들 사용자는 클러스터를 모니터링할 수 있지만 관리할 수는 없습니다.

클러스터 및 노드 관리

클러스터 관리에는 다음 작업이 포함됩니다.

- 모든 노드가 중지되었을 때 클러스터 시작
- 실행 중인 클러스터에 이전에 구성된 노드 추가
- 실행 중인 클러스터에서 노드 제거
- 전체 클러스터 중지

ServiceGuard Manager, SAM 또는 MC/ServiceGuard 명령을 사용하여 클러스터를 시작 또는 중지하거나 노드를 추가 또는 중지시킬 수 있습니다. SAM 또는 MC/ServiceGuard 명령을 사용하여 클러스터 구성을 수정할 수 있습니다. 클러스터를 시작한다는 것은 클러스터의 하나 이상의 노드에서 클러스터 데몬을 실행하는 것을 의미합니다. 모든 노드가 현재 중지되었는지(즉, 클러스터 데몬이 실행되고 있지 않음) 또는 개별 노드에서 클러스터 데몬을 시작할 것인지에 따라 다른 MC/ServiceGuard 명령을 사용하여 클러스터를 시작합니다.

이 장의 설명에서 이미 구성된 노드를 클러스터에 추가하는 것과 새 노드를 클러스터 구성에 추가하는 것과의 차이점에 주의하십시오. 이미 구성된 노드는 클러스터 구성 파일에 이미 입력된 노드이며 새 노드는 클러스터 구성 파일을 수정하여 클러스터에 추가합니다.

모든 노드가 중지되었을 때 클러스터 시작

ServiceGuard Manager, SAM 또는 MC/ServiceGuard 명령을 사용하여 클러스터를 시작할 수 있습니다.

ServiceGuard Manager를 사용하여 클러스터 시작

클러스터 아이콘을 선택한 다음 마우스 오른쪽 단추를 눌러 동작 메뉴를 표시합니다. “Start the Cluster”를 선택합니다. 동작이 발생할 때 진행 창에 메시지가 표시됩니다. 여기에는 각 노드와 패키지 시작에 대한 메시지가 포함됩니다. 작업이 끝나면 진행 창에서 OK를 누릅니다.

SAM을 사용하여 클러스터 시작

SAM에서 Clusters를 선택한 다음 High Availability Clusters 옵션을 선택하십시오. Cluster Administration 영역에서 시작할 클러스터를 선택한 다음 Actions 메뉴에서 “Start Cluster”를 선택합니다. 클러스터를 모든 노드 또는 특정 노드로 가져올지 지정합니다. 특정 노드를 선택할 경우 목록에서 노드를 선택해야 합니다. 이 옵션은 클러스터가 노드에서 아

직 실행되고 있지 않은 것이 확실할 때만 사용하십시오.

확인 메시지가 나타나면 OK를 선택하여 클러스터를 시작합니다.

MC/ServiceGuard 명령을 사용하여 클러스터 시작

모든 클러스터 노드가 중지되었을 때 `cmruncl` 명령을 사용하여 클러스터를 시작하십시오. 특정 노드의 그룹을 지정하려면 `-n` 옵션을 사용하십시오. 이 옵션을 사용하지 않으면 모든 노드가 시작됩니다. 다음 명령은 클러스터에 구성된 모든 노드를 시작합니다.

```
# cmruncl -v
```

다음 예제는 `ftsys9` 및 `ftsys10`에서만 로컬에서 구성된 클러스터를 시작합니다. 이 명령은 클러스터가 어떤 노드에서도 아직 실행되고 있지 않을 때에만 사용해야 합니다.

```
# cmruncl -v -n ftsys9 -n ftsys10
```

경고

특정 클러스터 노드가 이미 클러스터를 실행 중인 동안 `cmruncl -n` 명령을 사용하여 클러스터를 시작하면 MC/ServiceGuard는 데이터의 무결성을 보장할 수 없습니다. 노드 사이에서 네트워크 연결이 끊어진 경우 `cmruncl -n`을 사용하면 또 다른 클러스터가 형성되고, 이 클러스터는 다른 클러스터에서 이미 실행 중인 동일한 응용 프로그램을 시작하게 됩니다. 그 결과, 두 응용 프로그램이 디스크에 있는 서로의 데이터를 덮어쓸 수 있습니다.

실행 중인 클러스터에 이전에 구성된 노드 추가

ServiceGuard Manager, SAM 또는 MC/ServiceGuard 명령을 사용하여 실행 클러스터 내에 구성된 노드를 나타낼 수 있습니다.

ServiceGuard Manager를 사용하여 클러스터에 노드 추가

노드 아이콘을 선택한 다음 마우스 오른쪽 단추를 눌러 동작 메뉴를 표시합니다. “Start the Node”를 선택합니다. 동작이 발생할 때 진행 창에 메시지가 표시됩니다. 여기에는 노드에서 실행에 적합한 모든 패키지 시작에 대한 메시지가 포함됩니다. 작업이 끝나면 진행 창에서 OK를 누릅니다.

SAM을 사용하여 실행 중인 클러스터에 이전에 구성된 노드 추가

Cluster Administration 영역에서 노드를 추가할 클러스터를 강조 표시한 다음 “Specify Nodes to Join the Cluster”를 선택합니다. 추가할 노드를 선택한 다음 OK를 선택합니다. 현재 클러스터 구성의 구성원에 해당하는 노드만 선택할 수 있습니다.

MC/ServiceGuard 명령을 사용하여 실행 중인 클러스터에 이전에 구성된 노드 추가

cmrunnode 명령을 사용하여 이미 실행 중인 클러스터에 하나 이상의 노드를 추가할 수 있습니다. 추가할 노드는 이미 클러스터 구성의 일부이어야 합니다. 다음 예제는 노드 *ftsys9* 및 *ftsys10*과 함께 시작한 클러스터에 노드 *ftsys8*을 추가합니다.

```
# cmrunnode -v ftys8
```

노드의 클러스터가 이미 실행 중이기 때문에 노드는 클러스터에 참여하고 패키지가 시작됩니다. 노드가 실행 중인 해당 클러스터를 찾지 못하거나 노드가 클러스터 구성의 일부가 아니면 이 명령은 실패합니다.

실행 중인 클러스터에서 노드 제거

SAM 또는 HP-UX 명령을 사용하여 클러스터에서 노드를 제거할 수 있습니다. 이 명령은 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 눌러 동작 메뉴를 표시합니다. “Halt the Node”를 선택합니다. 동작이 발생할 때 진행 창에 메시지가 표시됩니다. 여기에는 노드의 모든 패키지를 대체 노드로 이동하는 것이 대한 메시지가 포함됩니다. 작업이 끝나면 진행 창에서 OK를 누릅니다.

SAM을 사용하여 노드 제거

Cluster Administration 영역에서 원하는 클러스터를 강조 표시한 다음 Actions 메뉴에서 “Specify Nodes to Leave the Cluster”를 선택합니다. 노드를 선택한 다음 OK를 선택하십시오.

중지하려는 노드에서 패키지가 실행 중일 경우에는 노드 중지를 확인하는 메시지가 표시됩니다. 중지하려면 OK를 선택하십시오. 중지된 다음 대체 노드로 전환될 수 있는 패키지는 대체 노드에서 시작됩니다.

MC/ServiceGuard 명령을 사용하여 작업에서 노드 제거

`cmhaltnode` 명령을 사용하면 클러스터에서 하나 이상의 노드를 중지할 수 있습니다. 지정된 노드의 클러스터 데몬이 중지된 다음 클러스터에 참여한 활성 노드가 제거됩니다.

실행 중인 패키지를 포함한 노드를 중지하려면 `-f` 옵션을 사용하십시오. 대체 노드로 전환될 수 있는 패키지가 실행 중이면 대체 노드로 전환되어 이 노드에서 패키지가 시작됩니다. 예를 들어, 다음 명령은 예제 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 패키지를 중지합니다. `ftsys9`는 중지되고 패키지는 대체 노드 `ftsys10`에서 시작됩니다.

`cmhaltnode`를 사용하면 편리하게 다른 노드에서 그 패키지를 계속 사용할 수 있도록 하면서 시스템 유지 관리를 위해 노드를 중지할 수 있습니다. 유지 관리가 끝나면 패키지는 기본 노드로 되돌아갈 수 있습니다. 아래 “패키지 이동”을 참조하십시오.

노드를 클러스터로 다시 보내려면 `cmrunnode`를 사용하십시오.

주

특히 중지할 때 패키지 응용 프로그램에 문제가 발생하여 확실히 중지되지 않는 경우를 대비하여 HP-UX의 `shutdown` 명령을 실행하기 전에 `cmhaltnode`를 실행하는 것이 좋습니다.

전체 클러스터 중지

ServiceGuard Manager, SAM 또는 MC/ServiceGuard 명령을 사용하여 실행 클러스터를 중지시킬 수 있습니다.

ServiceGuard Manager를 사용하여 클러스터 중지

클러스터를 선택한 다음 마우스 오른쪽 단추를 눌러 동작 메뉴를 표시합니다. “Halt the Cluster”를 선택합니다. 동작이 발생할 때 진행 창에 메시지가 표시됩니다. 여기에는 각 패키지와 노드 중지에 대한 메시지가 포함됩니다. 작업이 끝나면 진행 창에서 OK를 누릅니다.

SAM을 사용하여 클러스터 중지

Cluster Administration 영역에서 시작할 클러스터를 선택한 다음 Actions 메뉴에서 “Shut Down Cluster”를 선택하여 모든 노드에서 선택한 클러스터를 중지합니다. 확인 메시지가 나타나면 OK를 선택하십시오.

MC/ServiceGuard 명령을 사용하여 클러스터 중지

cmhaltcl 명령을 사용하여 전체 클러스터를 중지할 수 있습니다. 이 명령은 구성된 클러스터의 모든 노드에서 MC/ServiceGuard 데몬을 중지합니다. -f 옵션을 사용하면 패키지가 실행 중일 때에도 강제로 클러스터를 중지할 수 있습니다. 이 명령은 실행 중인 어떤 노드에서도 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 시작하도록 구성할 수 있습니다. 이 작업은 /etc/rc.config.d/cmcluster 파일에서 AUTOSTART_CMCLD를 1로 설정하여 수행합니다.

패키지 및 서비스 관리

패키지 및 서비스 관리에는 다음 작업이 포함됩니다.

- 패키지 시작
- 패키지 중지
- 패키지 이동
- 패키지 전환 동작 변경

ServiceGuard Manager, SAM 또는 MC/ServiceGuard 명령을 사용하여 이러한 작업을 수행할 수 있습니다.

패키지 시작

일반적으로 클러스터의 일부로 구성된 패키지는 클러스터가 시작되면 기본 노드에서 시작됩니다. 패키지를 수동으로 중지했으면 수동으로 시작해야 합니다. SAM 또는 MC/ServiceGuard 명령으로 이 작업을 수행할 수 있습니다.

ServiceGuard Manager를 사용하여 패키지 시작

시작할 패키지를 선택하고 마우스 오른쪽 단추를 눌러 동작 목록을 표시합니다. 패키지의 기본 구성 노드나 패키지 노드 목록의 노드에서 패키지를 시작할 수 있습니다. “Start Package on Configured Node” 또는 “Start Package on Specified Node”를 선택합니다. “Start Package on Specified Node”를 선택한 경우 패키지를 시작할 노드에서 선택할 수 있는 적합한 실행 노드 목록이 표시됩니다.

동작이 발생할 때 진행 창에 메시지가 표시됩니다. 여기에는 패키지 시작에 대한 메시지가 포함됩니다.

클러스터를 실행해야 패키지를 시작할 수 있습니다.

SAM을 사용하여 패키지 시작

SAM에서 “Package Administration”을 선택한 다음 시작할 패키지를 선택하십시오. “Actions” 메뉴에서 “Start Package”를 선택합니다. 특정 노드에서 패키지를 시작하려면 “Start a Package on a Specific Node”를 선택합니다. 그렇지 않으면 “Start Package”를 선택한 다음 확인 메시지가 나타날 때 Yes를 선택하십시오.

MC/ServiceGuard 명령을 사용하여 패키지 시작

cmrunpkg 명령을 사용하여 특정 노드의 패키지를 실행한 다음 cmmmodpkg 명령을 사용하여 패키지 전환을 설정하십시오. 예를 들면 다음과 같습니다.

```
# cmrunpkg -n ftsys9 pkg1  
# cmmmodpkg -e pkg1
```

이 명령은 *ftsys9*에서 패키지를 시작한 다음 패키지 전환을 사용 설정합니다. 패키지를 중지하면 전환할 수 없기 때문에 일부 노드에서 이전에 패키지가 중지된 경우에 이 과정이 필요합니다.

패키지 중지

패키지를 중지하는 경우에도 노드가 계속 실행되도록 하려면 MC/ServiceGuard 패키지를 중지해야 합니다. SAM이나 MC/ServiceGuard 명령을 사용하여 패키지를 중지할 수 있습니다. 패키지 중지의 결과와 노드 중지의 결과는 서로 다릅니다. 노드를 중지하면 그 패키지는 대체 노드로 전환됩니다(패키지 전환이 사용 설정된 경우). 반면 패키지를 중지하면 패키지가 다른 노드로 전환될 수 없기 때문에 다른 노드나 동일한 노드에서 수동으로 패키지를 재시작해야 합니다.

ServiceGuard Manager로 패키지 중지

중지할 패키지를 선택하고 마우스 오른쪽 단추를 눌러 동작 목록을 표시합니다. “Halt the package”를 선택합니다. 패키지가 실행되고 있어야 합니다.

동작이 발생할 때 진행 창에 메시지가 표시됩니다. 여기에는 패키지 중지 관련 메시지가 포함됩니다.

SAM을 사용하여 패키지 중지

SAM에서 “Package Administration” 영역의 목록에서 패키지를 선택한 다음 Actions 메뉴에서 “Halt Package”를 선택하십시오. 확인 메시지가 나타나면 OK를 선택하십시오. 이 방법으로 패키지를 중지하면 패키지가 다른 노드로 전환되지 않습니다.

MC/ServiceGuard 명령을 사용하여 패키지 중지

다음과 같이 cmhaltpkg 명령을 사용하여 패키지를 중지할 수 있습니다.

```
# cmhaltpkg pkg1
```

이 명령은 pkg1을 중지하고 다른 노드로 전환될 수 없게 합니다.

패키지 이동

SAM 또는 MC/ServiceGuard 명령을 사용하여 한 노드에서 다른 노드로 패키지를 이동할 수 있습니다.

ServiceGuard Manager를 사용하여 패키지 이동

중지할 패키지를 선택하고 마우스 오른쪽 단추를 눌러 동작 목록을 표시합니다. “Move package to node”를 선택합니다. 패키지가 실행되고 있어야 합니다.

가능한 대상 목록이 표시됩니다. 패키지를 실행할 노드를 누릅니다.

동작이 발생할 때 진행 창에 메시지가 표시됩니다. 여기에는 패키지 중지 및 대상 노드에서 시작할 다른 패키지에 대한 메시지가 포함됩니다.

SAM을 사용하여 실행 중인 패키지 이동

SAM의 Package Administration 화면에서 패키지를 선택한 다음 Action 메뉴에서 “Move a Package”를 선택하십시오. 패키지가 이동할 대상 노드를 선택한 다음 OK를 선택하십시오. 확인 메시지가 나타나면 Yes를 선택하십시오.

MC/ServiceGuard 명령을 사용하여 실행 중인 패키지 이동

패키지를 이동하기 전에 `cmhaltpkg` 명령을 사용하여 원래의 노드에서 해당 패키지를 중지하십시오. 이 작업은 패키지를 중지할 뿐 아니라 패키지를 중지한 노드로 패키지가 다시 전환될 수 없게 합니다.

패키지를 이동한 다음에는 패키지를 재시작하고 전환을 사용 설정해야 합니다. SAM에서 또는 `cmrunpkg` 명령과 함께 `cmmodpkg -e package_name`을 실행하여 이 작업을 수행할 수 있습니다. `-n` 옵션과 함께 `cmmodpkg`를 사용하면 패키지가 일부 오류로 인해 노드에서 실행되지 않은 경우 패키지를 실행할 수 있습니다. 노드를 지정하지 않으면 명령이 실행 중인 노드가 선택됩니다.

예를 들면 다음과 같습니다.

```
# cmhaltpkg pkg1
# cmrunpkg -n ftsys10 pkg1
# cmmodpkg -e pkg1
```

패키지 전환 동작 변경

패키지에 적용되는 전환에는 두 가지가 있습니다.

- 패키지 전환 또는 노드 간에 이동하는 기능
- 노드 전환 또는 특정 노드로 이동하는 기능

패키지 전환에 대한 초기 설정은 `AUTO_RUN` 매개 변수로 결정됩니다. 이 매개 변수는 패키지 ASCII 구성 파일에 설정되어 있습니다. `AUTO_RUN` 매개 변수가 `YES`로 설정되어 있으면 패키지가 처음 클러스터에서 시작될 때 패키지 전환이 설정됩니다. 노드 전환에 대한 최초의 설정은 패키지를 실행하도록 구성된 모든 노드로 전환을 허용하는 것입니다. 노드 전환과 패키지 전환은 클러스터가 실행될 때 둘 다 동적으로 변경할 수 있습니다.

ServiceGuard Manager로 패키지 전환 변경

ServiceGuard Manager에서 패키지 전환을 변경하려면 패키지 아이콘을 선택한 다음 마우스 오른쪽 단추를 눌러 동작 메뉴를 표시합니다. 이렇게 하면 “Package Switching” 확인란이 표시됩니다. 패키지 전환을 설정하려면 이 확인란을 눌러 확인 표시가 나타나도록 합니다. 패키지 전환 기능을 해제하려면 이 확인란을 눌러 확인 표시를 제거합니다.

특정 노드에서 시작하도록 패키지 기능을 설정하거나 변경하려면 동작 메뉴에서 “Node Switching”을 선택합니다. 이렇게 하면 패키지를 실행하는 데 적합한 각 노드에 대해 확인란이 표시됩니다. 패키지를 실행할 노드의 기능을 토글하려면 해당 노드 확인란을 누릅니다. 패키지가 노드에서 실행되고 있는 데 해당 노드로 패키지가 전환되는 설정을 해제한 경우에는 패키지는 계속 실행되지만 나중에 그 노드로 다시 전환할 수 없습니다.

MC/ServiceGuard 명령으로 패키지 전환 변경

MC/ServiceGuard 명령을 사용하여 패키지 전환 동작을 임시 또는 영구적으로 변경할 수 있습니다. 실행 중인 패키지의 다른 노드로의 전환 기능을 임시로 사용 해제하려면 `cmmodpkg`를 사용합니다. 예를 들어, `pkg1`이 현재 실행 중일 때 이 패키지가 다른 노드에서 시작되지 않도록 하려면 다음을 입력합니다.

```
# cmmodpkg -d pkg1
```

이 명령은 패키지를 중지시키지는 않습니다. 그러나 패키지가 다른 곳에서 시작되지 않도록 합니다.

`cmmodpkg` 명령의 `-n` 옵션을 사용하여 특정 노드로 패키지 전환 설정을 해제할 수도 있습니다. 다음은 `lpctest3` 노드로 전환하는 `pkg1`의 기능 설정을 해제합니다.

```
# cmmodpkg -d -n lpctest3 pkg1
```

전환을 영구적으로 사용 불능으로 설정하여 클러스터가 다시 시작된 다음에도 변경 내용이 그대로 적용되도록 하려면, 패키지 구성 파일에서 `AUTO_RUN` 플래그를 변경하고 구성을 다시 적용해야 합니다. 이렇게 변경된 사항은 다음 번에 클러스터가 시작될 때에도 적용됩니다.

재구성에 관한 자세한 내용은 “실행 중인 클러스터에서 패키지 재구성” 절을 참조하십시오.

클러스터 재구성

클러스터가 중지되었거나 실행되고 있을 때 클러스터를 재구성할 수 있습니다. 일부 작업은 클러스터가 중지되었을 때만 수행할 수 있습니다. 표 7-2는 여러 종류의 변경에 필요한 클러스터 상태를 나타낸 것입니다.

표 7-2

영구적인 클러스터 구성 변경의 종류

클러스터 구성 변경	요구되는 클러스터 상태
새 노드 추가	모든 클러스터 노드가 실행되고 있어야 합니다. 직렬 하트비트가 구성되어 있으면 안 됩니다.
노드 삭제	사용할 수 없거나 액세스할 수 없는 경우에도 노드를 삭제할 수 있습니다.
볼륨 그룹 추가	클러스터가 실행 중이거나 중지 중입니다.
볼륨 그룹 삭제	클러스터가 실행 중이거나 중지 중입니다. 볼륨 그룹을 사용하는 패키지는 제어 스크립트가 변경될 때까지는 다시 시작되지 않습니다.
구성된 패키지의 최대 수 변경	클러스터가 실행되고 있으면 안 됩니다.
타이밍 매개 변수 변경	클러스터가 실행되고 있으면 안 됩니다.
Quorum 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 모니터된 서브넷의 IP 주소 변경	클러스터가 실행되고 있으면 안 됩니다.

중지된 클러스터 재구성

클러스터가 실행 중이지 않을 때 클러스터 구성을 영구적으로 변경할 수 있습니다. 이 과정은 quorum 서버나 잠금 디스크 구성 변경, 타이밍 매개 변수 변경 및 최대 패키지 수 매개 변수 변경 시 **반드시** 수행해야 하지만 다른 클러스터 구성 변경 시에도 사용할 수 있습니다.

다음 단계를 수행하십시오.

1. 모든 노드에서 클러스터를 중지합니다.
2. “고가용성 클러스터 구성 만들기”에서 설명한 것처럼 하나의 노드에서 클러스터를 재구성합니다. SAM을 사용하거나 cmquerycl 명령으로 ASCII 파일을 만들고 편집하여 이 작업을 수행할 수 있습니다.
3. 클러스터 구성 파일에 나열된 모든 노드에 전원이 공급되고 액세스할 수 있는지 확인합니다. 모든 노드에 이진 클러스터 구성 파일을 복사하려면 SAM 또는 cmapplyconf 명령을 사용합니다. 이 파일은 기존의 이진 클러스터 구성 파일을 덮어씁니다.
4. SAM 또는 cmrunc1 명령을 사용하여 모든 노드나 특정 노드에서 클러스터를 시작합니다.

SAM을 사용하여 최대 패키지 매개 변수 변경

클러스터를 중지한 다음 Cluster Configuration 영역으로 이동하여 Actions 메뉴에서 “Modify Cluster Configuration”을 선택합니다. 그런 다음 “Modify Cluster Package Requirements”를 선택하십시오. 지정된 영역에 새로운 패키지의 최대 수를 입력한 다음 OK를 선택하십시오.

MC/ServiceGuard 명령을 사용하여 MAX_CONFIGURED_PACKAGES 변경

cmgetconf 명령을 사용하여 클러스터의 기존 구성에 대한 현재 사본을 만드십시오. 예를 들면 다음과 같습니다.

```
# cmgetconf -C clconfig.ascii
```

원하는 MAX_CONFIGURED_PACKAGES의 값을 포함하도록 clconfig.ascii 파일을 편집하십시오. 그런 다음 cmcheckconf 명령을 사용하여 새로운 구성을 확인하십시오. cmapplyconf 명령을 사용하여 변경 사항을 구성에 적용하고 모든 클러스터 노드에 새로운 구성 파일을 전송하십시오.

실행 중인 클러스터 재구성

클러스터가 시작되어 실행 중일 때 클러스터 구성에서 새 노드를 추가하거나 삭제할 수 있습니다. 하지만 다음 사항에 주의하십시오.

- 클러스터가 실행되고 있을 때는 quorum 서버나 잠금 디스크 구성을 변경할 수 없습니다.
- 클러스터에서 활성 노드를 제거할 수 없습니다. 먼저 노드를 중지해야 합니다.
- 클러스터 구성에서 활성 볼륨 그룹을 삭제할 수 없습니다. 볼륨이 비활성화된 상태인지 확인한 다음 볼륨 그룹을 사용하는 패키지를 중지해야 합니다.
- 클러스터 타이밍 매개 변수를 변경할 수 없습니다.
- 노드에 접근할 수 없을 때(예: 네트워크에서 완전히 연결이 끊어진 경우) 허용되는 유일한 구성 변경은 클러스터 구성에서 이 노드를 삭제하는 것뿐입니다. 또한, 해당 노드에 의존하는 패키지가 있으면 패키지 구성 역시 노드를 삭제할 수 있도록 수정되어야 합니다. 이 작업은 모두 하나의 구성 요청(`cmapplyconf` 명령)으로 수행되어야 합니다.

패키지 구성 변경에 대한 자세한 내용은 뒤에서 설명합니다.

다음 절에서는 SAM이나 MC/ServiceGuard 명령을 사용하여 동적 재구성 작업을 수행하는 방법을 설명합니다.

클러스터 실행 중에 SAM을 사용하여 구성에 노드 추가

SAM의 Cluster Configuration 영역으로 이동하여 Actions 메뉴에서 “Modify Cluster Configuration”을 선택하십시오. 그런 다음 “Modify Cluster Name and Nodes”를 선택하십시오. 화면의 오른쪽에 있는 목록에서 클러스터에 추가할 노드의 이름을 강조 표시하고 Add 단추를 누르십시오. 노드 목록의 변경이 끝나면 OK를 선택하십시오.

클러스터 실행 중에 MC/ServiceGuard 명령을 사용하여 구성에 노드 추가

다음 단계를 따라 HP-UX 명령을 사용하여 노드를 추가할 수 있습니다. 이 예제에서 노드 `ftsyst8` 및 `ftsyst9`는 `cluster1`이라는 이름의 실행 중인 클러스터에 이미 구성되어 있으며, 사용자는 노드 `ftsyst10`을 추가하려고 합니다.

1. 다음 명령을 사용하여 임시 파일에 기존의 클러스터 구성에 대한 현재 사본을 저장할 수 있습니다.

```
# cmgetconf -C temp.ascii
```

2. 구성할 새 노드 집합을 지정하고 새 구성을 적용한 템플리트를 만듭니다.

```
# cmquerycl -C clconfig.ascii -c cluster1 \  
-n ftsys8 -n ftsys9 -n ftsys10
```

3. 파일 clconfig.ascii를 편집하여 새 노드에 대한 정보를 확인합니다.
4. 다음과 같이 새 구성을 확인합니다.

```
# cmcheckconf -C clconfig.ascii
```

5. 구성에 변경 사항을 적용하고 모든 클러스터 노드에 새 이진 구성 파일을 전송합니다.

```
# cmapplyconf -C clconfig.ascii
```

cmrunnode를 사용하여 새 노드를 시작하고, 필요하면 /etc/rc.config.d/cmcluster 파일에서 AUTOSTART_CMCLD 매개 변수를 1로 설정하여 재부팅할 때마다 자동으로 새 노드가 클러스터에 참여하게 하십시오.

주

CVM 디스크 그룹을 사용하는 실행 클러스터에 노드를 추가하면 이 노드가 클러스터에 연결되어 있을 때 디스크 그룹에서 가져올 수 있습니다.

클러스터 실행 중에 SAM을 사용하여 구성에서 노드 삭제

SAM의 Cluster Administration 영역에서 클러스터 구성에서 제거할 노드를 중지하십시오. 그런 다음 Cluster Configuration 영역으로 이동하여 Actions 메뉴에서 “Modify Cluster Configuration”을 선택하십시오. 그런 다음 “Modify Cluster Name and Nodes”를 선택하십시오. 구성된 노드 목록의 클러스터에서 삭제할 노드의 이름을 강조 표시하고 Remove 단추를 누르십시오. 노드 목록의 변경이 끝나면 OK를 선택하십시오.

삭제하려는 노드에 접근할 수 없는 경우(예: LAN에서 연결이 끊어진 경우) 접근할 수 없는 해당 노드를 지정하는 패키지가 없을 때에만 SAM을 사용하여 노드를 삭제할 수 있습니다. 접근할 수 없는 해당 노드의 영향을 받는 패키지가 있으면 다음 절에서 설명하는 것처럼 MC/ServiceGuard 명령을 사용하십시오.

클러스터 실행 중에 MC/ServiceGuard 명령을 사용하여 구성에서 노드 삭제

다음 단계대로 HP-UX 명령을 사용하여 노드를 삭제할 수 있습니다. 이 예제에서 노드 *ftsyst8*, *ftsyst9* 및 *ftsyst10*은 *cluster1*이라는 이름의 실행 중인 클러스터에 이미 구성되어 있으며, 사용자는 노드 *ftsyst10*을 삭제하려고 합니다.

1. 다음 명령을 사용하여 임시 파일에 기존의 클러스터 구성에 대한 현재 사본을 저장할 수 있습니다.

```
# cmgetconf -C temp.ascii
```

2. 구성할 새 노드 집합을 지정하고(*ftsyst10*은 생략) 새 구성을 적용한 템플릿을 만듭니다.

```
# cmquerycl -C clconfig.ascii -c cluster1 -n ftsyst8 -n ftsyst9
```

3. 파일 *clconfig.ascii*를 편집하여 클러스터에 남아 있는 노드에 대한 정보를 확인합니다.

4. 다음과 같이 새 구성을 확인합니다.

```
# cmcheckconf -C clconfig.ascii
```

5. 구성에 변경 사항을 적용하고 모든 클러스터 노드에 새 이진 구성 파일을 전송합니다.

```
# cmapplyconf -C clconfig.ascii
```

주

접근할 수 없으나 많은 패키지가 의존하고 있는 노드를 제거할 경우, 특히 종속 패키지가 많은 EMS 리소스를 사용하고 있으면 다음 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 SAM을 사용하여 LVM 구성 변경

SAM에서 Cluster Configuration 영역으로 이동하여 Actions 메뉴에서 “Modify Cluster Configuration”을 선택합니다. 그런 다음 “Modify Cluster Package Requirements”를 선택합니다. 볼륨 그룹 이름을 강조 표시하고 이 이름을 클러스터 인식 볼륨 그룹 목록에 추가하거나 목록에서 제거하여 볼륨 그룹을 추가하거나 삭제할 수 있습니다.

클러스터 실행 중에 클러스터 잠금 볼륨 그룹이나 실제 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 그룹에서 LVM vgexport 명령을 사용해야 합니다.

클러스터 실행 중에 MC/ServiceGuard 명령을 사용하여 LVM 구성 변경

cmgetconf 명령을 사용하여 클러스터의 기존 구성에 대한 현재 사본을 만드십시오. 예를 들면 다음과 같습니다.

```
# cmgetconf -C clconfig.ascii
```

clconfig.ascii 파일을 편집하여 볼륨 그룹을 추가하거나 삭제하십시오. 그런 다음 cmcheckconf 명령을 사용하여 새로운 구성을 확인하십시오. cmapplyconf 명령을 사용하여 변경 사항을 구성에 적용하고 모든 클러스터 노드에 새로운 구성 파일을 전송하십시오.

주

패키지에 의해 현재 활성화된 볼륨 그룹을 클러스터에서 삭제하는 경우 구성은 변경되지만 패키지를 중지할 때까지 삭제되지 않습니다. 따라서 패키지 제어 스크립트에서 볼륨 그룹을 제거하는 것과 같은 추가 수정 작업이 없으면 더 이상 패키지를 실행할 수 없습니다.

VxVM 또는 CVM 저장 구성 변경

클러스터 실행 중에 클러스터 구성에 VxVM 디스크 그룹을 추가할 수 있습니다. 새로운 CVM 디스크 그룹을 추가하려면 클러스터를 실행하고 **있어야 합니다**.

새로운 CVM 디스크 그룹을 만들 경우에는 다음 명령을 사용하여 만들 주 노드를 결정해야 합니다.

```
# vxdctl -c mode
```

한 노드는 자신을 주 노드로 인식합니다. 이 노드에서 디스크 그룹을 만듭니다. 저장 그룹을 클러스터 ASCII 파일의 STORAGE_GROUP 문에 추가한 다음 cmapplyconf 명령을 실행해야 합니다.

마찬가지로, 한 번에 한 클러스터 노드에서 VxVM 또는 CVM 디스크 그룹을 사용 중이라고 가정하고 이들 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 항목도 제거해야 합니다.

패키지 재구성

패키지 재구성 과정은 6장에 설명된 기본 구성과 거의 같습니다. 편성 과정에 대한 자세한 내용은 이 장을 참조하십시오.

패키지를 재구성하는 중에 클러스터가 중지되거나 실행되고 있을 수 있습니다. 변경할 수 있는 패키지 유형과 적용 시기는 패키지 실행 여부에 따라 다릅니다.

중지된 클러스터에서 패키지 재구성

클러스터가 실행 중이지 않을 때 패키지 구성을 영구 변경할 수도 있습니다. 다음 단계를 수행하십시오.

- 이 장의 앞에서 설명한 것처럼 하나의 노드에서 패키지를 재구성합니다. SAM을 사용하거나 패키지의 ASCII 파일을 편집하여 이 작업을 수행할 수 있습니다.
- 패키지 제어 스크립트를 직접 편집하거나 SAM에서 “Edit a Package Control Script” 옵션을 사용합니다. 서비스 이름을 변경할 경우 패키지 구성 파일에서도 동일하게 이름을 변경해야 합니다.
- 모든 노드에 이진 클러스터 구성 파일을 복사하려면 SAM 또는 `cmapplyconf` 명령을 사용합니다. `-P` 옵션을 사용하여 변경할 패키지를 지정하십시오. `-C` 옵션은 사용하지 마십시오. 이 파일은 기존의 이진 클러스터 구성 파일을 덮어씁니다.
- 패키지를 실행할 수 있는 모든 노드로 수정된 제어 스크립트를 복사합니다.
- SAM 또는 `cmrunc1` 명령을 사용하여 모든 노드나 특정 노드에서 클러스터를 시작합니다. 패키지는 노드가 온라인 상태가 될 때 시작됩니다.

실행 중인 클러스터에서 패키지 재구성

클러스터 실행 중에 패키지를 재구성할 수 있고, 일부 경우에는 패키지 자체가 실행되는 동안 패키지를 재구성할 수도 있습니다. 패키지 실행 중에는 특정 사항만 변경할 수 있습니다.

패키지를 수정하려면 6장에서 설명한 것처럼 SAM의 “Package Configuration” 하위 영역에서 옵션을 선택하십시오. 또는, HP-UX 명령을 사용하여 다음 과정을 따르십시오 (`pkg1`을 예제로 사용).

클러스터 및 패키지 유지 관리

패키지 재구성

1. 필요하면 패키지를 중지합니다.

```
# cmhaltpkg pkg1
```

이 단계가 필요한지 확인하려면 표 7-1을 참조하십시오.

2. 아직 패키지의 ASCII 구성 파일의 사본이 없는 경우 `cmgetconf` 명령에서 패키지 이름을 지정하여 이 사본을 만들 수 있습니다.

```
# cmgetconf -P pkg1.ascii
```

3. ASCII 패키지 구성 파일을 편집합니다.
4. 다음 명령으로 변경 사항을 확인합니다.

```
# cmcheckconf -v -P pkg1.ascii
```

5. 모든 노드에 변경 사항을 배포합니다.

```
# cmapplyconf -v -P pkg1.ascii
```

6. 패키지를 실행할 수 있는 모든 노드로 패키지 제어 스크립트를 복사합니다.

실행 중인 클러스터에 패키지 추가

클러스터가 작동 중이고 다른 패키지가 실행 중일 때 새로운 패키지를 작성하여 클러스터 구성에 추가할 수 있습니다. 추가할 수 있는 패키지 수는 SAM의 *Maximum Configured Packages*에 지정된 값이나 클러스터 구성 파일에 정의된 값에 따라 달라집니다.

패키지를 만들려면 “패키지 및 서비스 구성” 장에서 설명한 절차를 따르면 되는 데 다음과 같은 차이가 있습니다. HP-UX 명령을 사용하여 구성을 확인하고 배포할 때는 클러스터 ASCII 파일을 지정하지 마십시오. 예를 들어, 다음과 같이 HP-UX 명령을 사용하여 실행 중인 클러스터에서 새로 작성된 `pkg1`의 구성을 확인할 수 있습니다.

```
# cmcheckconf -P /etc/cmcluster/pkg1/pkg1conf.ascii
```

다음과 같은 HP-UX 명령을 사용하여 클러스터의 모든 노드에 새로운 패키지 구성을 배포할 수 있습니다.

```
# cmapplyconf -P /etc/cmcluster/pkg1/pkg1conf.ascii
```

패키지를 실행할 수 있는 모든 노드의 `/etc/cmcluster/pkg1` 디렉토리로 제어 스크립트를 복사해야 합니다.

실행 중인 클러스터에서 패키지 삭제

`cmdelleteconf` 명령을 사용하여 모든 클러스터 노드에서 패키지를 삭제할 수 있습니다. 이 명령은 패키지가 실행되고 있지 않을 때만 실행할 수 있습니다. 하지만 클러스터는 실행되고 있어도 됩니다. 이 명령은 클러스터의 모든 노드에 있는 이진 구성 파일에서 패키지 정보를 제거합니다.

다음 예제는 패키지 `mypkg`를 중지하고 클러스터에서 패키지 구성을 제거합니다.

```
# cmhaltpkg mypkg
# cmdelleteconf -p mypkg
```

`-f` 옵션을 사용하지 않는 한 이 명령은 파일을 삭제하기 전에 확인 메시지를 표시합니다. `/etc/cmcluster/mypkg` 디렉토리는 이 명령으로 삭제되지 않습니다.

서비스 재시작 카운터 재설정

서비스 재시작 카운터는 패키지 서비스가 자동으로 다시 시작한 횟수입니다. 이 값은 패키지 서비스가 허용되는 최대 자동 재시작 횟수를 초과한 때를 확인하기 위해 사용됩니다.

주

특정 서비스에 허용되는 최대 재시작 횟수는 패키지 제어 스크립트 매개 변수 `SERVICE_RESTART []`에서 설정됩니다. 이 매개 변수는 패키지 관리자가 개별적으로 관리하는 재시작 카운터와는 다릅니다.

여러 차례의 재시도 끝에 패키지 서비스가 성공적으로 재시작되면 패키지 관리자는 재시작 카운터를 자동으로 재설정하지 않습니다. 그러나 `cmmodpkg -R -s` 명령을 사용하여 카운터를 온라인 상태로 재설정할 수 있기 때문에 이후에는 `SERVICE_RESTART`에 설정된 최대 횟수까지 재시작을 시도할 수 있습니다. 예를 들면 다음과 같습니다.

```
# cmmodpkg -R -s myservice pkg1
```

현재 재시작 카운터의 값은 `cmviewcl -v` 명령을 사용하여 볼 수 있습니다.

재구성 중에 허용되는 패키지 상태

구성을 변경할 때 클러스터의 모든 노드는 전원이 공급되고 액세스할 수 있어야 합니다.

특정 종류의 변경을 수행하는 동안 패키지를 실행할 수 있는지 확인하려면 표 7-1을 참조하십시오. 다음과 같은 경우에는 클러스터가 실행될 수 있으며 재구성되는 패키지 이외의 패키지들도 실행될 수 있습니다.

표 7-3

패키지 변경 유형

패키지 변경	필요한 패키지 상태
새 패키지 추가	다른 패키지는 어떠한 상태도 가능합니다.
패키지 삭제	패키지를 실행하고 있지 않아야 합니다.
서비스 추가	패키지를 실행하고 있지 않아야 합니다.
서비스 제거	패키지를 실행하고 있지 않아야 합니다.
서브넷 추가	패키지를 실행하고 있지 않아야 합니다. 서브넷이 이미 클러스터에 구성되어 있어야 합니다.
서브넷 제거	패키지를 실행하고 있지 않아야 합니다.
리소스 추가	패키지를 실행하고 있지 않아야 합니다. 리소스가 이미 클러스터에 구성되어 있어야 합니다.
리소스 제거	패키지를 실행하고 있지 않아야 합니다.
볼륨 그룹 추가	클러스터가 실행되는 동안 볼륨 그룹을 클러스터에 구성할 수 있습니다. 제어 스크립트는 변경 가능하므로 패키지는 어떠한 상태일 수도 있습니다. 그러나 변경 사항을 적용하려면 패키지를 중지한 다음 다시 시작해야 합니다.
볼륨 그룹 제거	패키지를 실행하고 있지 않아야 합니다.
실행 스크립트 내용 변경	패키지를 중지하는 것이 좋습니다. 패키지가 실행 중일 때 패키지의 실행 스크립트를 수정하면 타이밍 문제가 발생할 수 있습니다.
중지 스크립트 내용 변경	패키지를 중지하는 것이 좋습니다. 패키지가 실행 중일 때 패키지의 실행 스크립트를 중지하면 타이밍 문제가 발생할 수 있습니다.
스크립트 시간 제한	패키지는 실행 중이거나 중지되어도 됩니다.

표 7-3

패키지 변경 유형

패키지 변경	필요한 패키지 상태
서비스 시간 제한	패키지를 실행하고 있지 않아야 합니다.
서비스 조기 차단	패키지를 실행하고 있지 않아야 합니다.
패키지 자동 실행	패키지는 실행 중이거나 중지되어도 됩니다.
로컬 LAN 장애 조치	패키지는 실행 중이거나 중지되어도 됩니다.
패키지에서 실행할 수 있는 노드의 순서 변경	패키지는 실행 중이거나 중지되어도 됩니다.
패키지 장애 조치 정책 변경	패키지는 실행 중이거나 중지되어도 됩니다.
패키지 장애 복구 정책 변경	패키지는 실행 중이거나 중지되어도 됩니다.

클러스터 이벤트에 대응

MC/ServiceGuard는 지속적인 시스템 관리 작업이 필요 없습니다. 장애가 없으면 클러스터는 모니터링되고 보호됩니다. 장애가 발생하면 다른 노드로 전송되도록 지정된 패키지들이 자동적으로 전송됩니다. 시스템 관리자가 계속 수행해야 할 책임은 클러스터를 모니터링하고 패키지 전송이 발생했는지 확인하는 것입니다. 패키지가 전송되면 그 원인을 알아내어 해결해야 합니다.

이벤트 모니터링 서비스와 HA 모니터는 디스크, LAN 카드 및 일부 시스템 이벤트에 대한 모니터링 기능을 제공할 수 있습니다. 자세한 내용은 설명서 *Using HA Monitors*를 참조하십시오.

패키지 전송이 발생한 경우에 수행해야 할 일반적인 수정 작업은 다음과 같습니다.

- 전송이 발생한 시간 확인
- 전송 원인 확인
- 하드웨어 장애 수리
- 소프트웨어 문제점 해결
- 노드 재시작
- 패키지를 원래의 노드로 다시 전송

시스템에서 MC/ServiceGuard 제거

MC/ServiceGuard에서 노드를 제거하려면 `swremove` 명령을 사용하여 소프트웨어를 삭제하십시오. 다음 사항에 주의하십시오.

- MC/ServiceGuard를 삭제할 노드에는 실행 중인 클러스터가 없어야 합니다.
- MC/ServiceGuard를 삭제할 노드가 클러스터 구성에 포함되어 있으면 안됩니다.
- 두 개 이상의 노드에서 MC/ServiceGuard를 제거하는 경우 `swremove`는 한 번에 하나의 노드에서만 실행해야 합니다.

클러스터 및 패키지 유지 관리
시스템에서 MC/ServiceGuard 제거

8

클러스터 문제점 해결

이 장에서는 클러스터 작동을 확인하고, 클러스터 상태를 확인하고, 하드웨어를 추가 또는 교체하고, 몇 가지 일반적인 클러스터 문제를 해결하는 방법을 설명합니다. 다음과 같은 내용을 다룹니다.

- 클러스터 작동 테스트
- 하드웨어 모니터링
- 디스크 교체
- I/O 카드 교체
- LAN 카드 교체
- 문제점 해결 방법
- 문제점 해결

클러스터 작동 테스트

일단 MC/ServiceGuard 클러스터를 구성했으면 장애가 발생할 경우 클러스터의 여러 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. 결과적으로 노드 및 응용 프로그램을 사용하지 못할 수도 있습니다.

SAM을 사용하여 클러스터 시작

클러스터 구성을 완료했으면 계속하기 전에 클러스터를 실행하십시오.

클러스터를 시작하려면 SAM에서 다음을 수행하십시오.

1. Clusters를 선택한 다음 High Availability Clusters를 선택합니다.
2. Cluster Administration 영역에서 시작할 클러스터를 선택한 다음 Actions 메뉴에서 “Start Cluster”를 선택합니다.
3. 클러스터를 모든 노드 또는 특정 노드로 가져올지 지정합니다. 특정 노드를 선택할 경우 목록에서 노드를 선택해야 합니다. 이 옵션은 클러스터가 노드에서 아직 실행되고 있지 않은 것이 확실할 때만 사용하십시오.
4. 확인 메시지가 나타나면 OK를 선택하여 클러스터를 시작합니다.

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>
```

여기에서 *service_cmd*는 매개 변수 SERVICE_CMD를 사용하여 패키지 제어 스크립트에 지정한 실행 파일입니다. 선택된 서비스는 SERVICE_RESTART를 지정해서는 안 됩니다.

2. `service_cmd` PID를 종료하려면 다음을 입력합니다.

```
# kill PID
```

3. 패키지 상태를 보려면 다음을 입력합니다.

```
# cmviewcl -v
```

패키지는 지정된 대체 노드에서 실행되고 있어야 합니다.

4. SAM을 사용하여 패키지를 기본 노드로 다시 이동합니다.

SAM의 Package Administration 화면에서 패키지를 선택한 다음 Action 메뉴에서 “Move a Package”를 선택하십시오. 패키지가 이동할 대상 노드를 선택한 다음 OK를 선택하십시오. 확인 메시지가 나타나면 Yes를 선택하십시오.

실행 중인 특정 데이터베이스에 따라 적절한 데이터베이스 복구를 수행하십시오.

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 검사하려면 구성된 다른 노드에 다음 명령을 입력하십시오.

```
# cmviewcl -v
```

전원을 끄는 노드가 중지되었고 그 패키지가 다른 노드로 올바르게 전환되었는지 확인할 수 있어야 합니다.

3. 노드 SPU의 전원을 켭니다.
4. 노드가 클러스터에 다시 참여하고 있는지 확인하려면 구성된 노드에 대해 다음 명령을 입력합니다.

```
# cmviewcl -v
```

해당 노드를 클러스터가 인식해야 하지만 노드의 패키지가 실행되고 있으면 **안됩니다**.

5. SAM을 사용하여 패키지를 원래 노드로 다시 이동합니다.

SAM의 Package Administration 화면에서 패키지를 선택한 다음 Action 메뉴에서 “Move a Package”를 선택하십시오. 패키지가 이동할 대상 노드를 선택한 다음 OK를 선택하십시오. 확인 메시지가 나타나면 Yes를 선택하십시오.

실행 중인 특정 데이터베이스에 따라 적절한 데이터베이스 복구를 수행하십시오.

6. 한 번에 하나씩 클러스터의 모든 노드에 대해 이 절차를 반복합니다.

네트워크 관리자 테스트

네트워크 관리자가 올바르게 작동 중인지 테스트하려면 클러스터의 각 노드에 대해 다음을 수행하십시오.

1. 노드의 기본 및 대기 LAN 카드를 확인하려면 다음을 입력합니다.

```
# lanscan
```

그런 후 다음을 입력합니다.

```
# cmviewcl -v
```

2. 기본 카드에서 LAN 연결을 끊습니다. ThinLAN 케이블을 사용할 경우 서브넷을 중단하지 않도록 주의하십시오.
3. `cmviewcl -v`를 사용하여 대기 카드가 기본 카드가 되도록 로컬 전환이 발생했는지 확인합니다.
4. 원래의 기본 카드로 LAN을 다시 연결한 다음 `cmviewcl -v`를 사용하여 상태를 확인합니다.

하드웨어 모니터링

고가용성 시스템의 권장되는 운용 방식에는 가능하면 장애를 막고, 장애가 발생할 때 최소한 신속하게 대처할 수 있도록 주의 깊은 결함 모니터링이 포함됩니다. 모든 종류의 오류나 경고에 대해 다음 항목을 모니터링해야 합니다.

- 디스크
- CPU
- 메모리
- LAN 카드
- 전원
- 모든 케이블
- 디스크 인터페이스 카드

일부 모니터링 작업은 간단한 실제 검사를 통해 수행할 수 있으나 포괄적인 모니터링을 위해서는 주기적으로 시스템 로그 파일(/var/adm/syslog/syslog.log)을 검사하여 구성된 모든 HA 장치에 대한 보고서를 작성해야 합니다. 장치와 관련된 오류가 발견되면 유지 관리 작업이 필요합니다.

이벤트 모니터링 서비스 사용

이벤트 모니터링 서비스(EMS)를 통해 특정 장치 및 시스템 자원의 모니터를 구성할 수 있습니다. 문제가 발생하면 관리자가 이 사실을 통보 받고 추가 조치를 수행할 수 있는 관리 워크스테이션으로 경고를 보낼 수 있습니다. 예를 들어, 클러스터에서 사용하는 미러링된 볼륨 그룹에서 미러가 손상되었을 때 디스크 모니터가 이를 보고하도록 구성할 수 있습니다.

자세한 내용은 *Using HA Monitors*를 참조하십시오.

EMS 하드웨어 모니터 사용

메모리, CPU, 기타 시스템 값에 대한 모니터링 및 보고를 위해 하드웨어 모니터를 사용할 수 있습니다. 일부 모니터는 특정 하드웨어 제품에서 제공됩니다.

자세한 내용은 *EMS Hardware Monitors User's Guide(B6191-90020)*를 참조하십시오.

HP Predictive를 사용한 모니터링

로그는 실제 장치의 장애 보고 메시지뿐만 아니라 시간이 지남에 따라 보다 덜 중요한 메시지도 기록하기 때문에 로그를 확인하여 발생할 수도 있는 장애를 미리 파악할 수 있습니다. HP Predictive는 모니터링을 자동화합니다. 이 제품은 모니터링된 시스템의 상태 대기열에서 정보를 수집하여 누적되는 오류를 확인합니다. 이 도구는 장애를 보고하고 시간이 지남에 따라 특정한 심각하지 않은 오류가 발생하는 장치에 대한 통계를 바탕으로 장애를 예측하기도 합니다. MC/ServiceGuard 클러스터의 경우, 모든 노드에서 HP Predictive를 실행해야 합니다.

또한, HP Predictive는 HP Response Center에 직접 오류 상황을 보고하고 기술 지원 담당자에게 잠재적인 문제점을 보고합니다. 다양한 지원 계약을 통해 HP Predictive를 사용할 수 있습니다. 자세한 내용은 HP 담당자에게 문의하십시오.

디스크 교체

결함 있는 디스크 장치의 교체 과정은 사용하는 디스크 구성 종류에 따라 다릅니다. 고가용성 엔클로저에서 어레이 장치와 디스크를 교체하기 위한 과정은 별도로 설명합니다.

결함 있는 어레이 장치 교체

RAID 1 또는 RAID 5에 HA 디스크 어레이를 구성한 경우 결함 있는 장치의 교체 방법은 어레이의 설명서를 참조하십시오. 교체 후에 장치는 새 디스크에 누락된 데이터를 자동으로 다시 구성합니다. LVM은 필요 없습니다. 이 과정을 디스크 **핫 스와핑**이라고 합니다.

HA 엔클로저에서 결함 있는 장치 교체

MirrorDisk/UX와 함께 소프트웨어 미러링을 사용하고 있으며 미러링된 디스크가 고가용성 디스크 엔클로저에 마운트된 경우 다음 단계에 따라 디스크 장치를 **핫 플러그**할 수 있습니다.

1. 장애가 발생한 디스크의 실제 볼륨 이름과 이 볼륨이 구성된 볼륨 그룹의 이름을 확인합니다. 다음 예제에서 볼륨 그룹 이름은 /dev/vg_sg01이고 실제 볼륨 이름은 /dev/dsk/c2t3d0입니다. 이 이름을 사용자 시스템의 볼륨 그룹 및 실제 볼륨 이름으로 대체하십시오.
2. 장애가 발생한 실제 볼륨에서 확장 정의된 논리 볼륨의 이름을 확인합니다.
3. 볼륨 그룹이 현재 활성화되어 있는 노드에서 **장애가 발생한 실제 볼륨에 확장된 각 논리 볼륨에 대해** 다음 명령을 실행합니다.

```
# lvreduce -m 0 /dev/vg_sg01/lvolname /dev/dsk/c2t3d0
```

4. 이때 장애 디스크를 제거하고 새 디스크를 설치합니다. 새 디스크는 이전 디스크와 동일한 HP-UX 이름을 갖게 됩니다.

5. lvreduce 명령을 실행한 노드에서 다음 명령을 실행하여 볼륨 그룹 구성 데이터를 새로 설치한 디스크에 복원하십시오.

```
# vgcfgrestore /dev/vg_sg01 /dev/dsk/c2t3d0
```

6. 다음 명령을 실행하여 논리 볼륨을 새로 설치한 디스크로 확장합니다.

```
# lvextend -m 1 /dev/vg_sg01 /dev/dsk/c2t3d0
```

클러스터 문제점 해결 디스크 교체

7. 마지막으로 장애가 발생한 실제 볼륨에서 확장된 각 논리 볼륨에 대해 `lvsync` 명령을 **사용합니다**. 이 명령은 새 디스크의 확장 볼륨을 다른 미러의 확장 볼륨과 동기화시킵니다.

```
# lvsync /dev/vg_sg01/lvolname
```

잠금 디스크 교체

장애가 발생한 잠금 디스크 장치의 교체는 데이터 디스크의 교체와 동일합니다. **전용** 잠금 디스크(사용자 데이터가 없는)를 사용하는 경우 다음 예제처럼 하나의 LVM 명령만 실행합니다.

```
# vgcfgrestore /dev/vg_lock /dev/dsk/c2t3d0
```

이 명령을 수행한 다음 적어도 한 시간 후에 `syslog` 파일의 메시지를 검토하여 잠금 디스크가 다시 사용 가능한지 확인하십시오.

인라인 SCSI 터미네이터를 사용하여 온라인 하드웨어 유지 관리

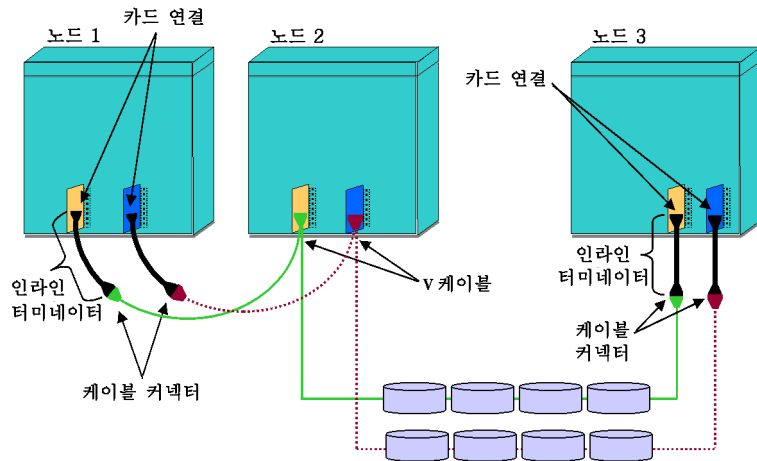
MC/ServiceGuard는 공유 FW/SCSI 버스의 한쪽 끝에 연결된 노드에서 HP의 **인라인 터미네이터(C2980A)**를 사용하는 경우 모든 클러스터 노드에 대해 온라인 SCSI 디스크 컨트롤러 하드웨어의 수리를 허용합니다. 인라인 터미네이터는 컨트롤러 카드에 부착된 터미네이션 팩 **대신** 사용되며, 버스의 터미네이션 없이 F/W SCSI 버스의 끝에서 노드를 실제적으로 분리할 수 있도록 합니다. Y 케이블을 사용하여 버스 중간에 부착된 노드도 손상시키지 않고 버스에서 분리할 수 있습니다. 인라인 터미네이터와 Y 케이블을 사용할 때는 컨트롤러 카드에서 모든 오렌지색 소켓 터미네이션 팩이 **제거**되었는지 확인하십시오.

주

D 및 K 계열 시스템의 내부 FW/SCSI 버스 및 단중점 SCSI 버스에서는 인라인 터미네이터를 사용할 수 없습니다. 노드를 Y 케이블에 연결하기 위해 인라인 터미네이터를 사용하지 **마십시오**.

그림 8-1은 두 개의 F/W SCSI 버스가 있는 3개의 노드로 구성된 클러스터를 보여줍니다. 실선과 점선은 다른 버스를 나타내며 이 두 버스에는 노드 1과 3에 부착된 인라인 터미네이터가 있습니다. Y 케이블도 노드 2에 부착되어 있습니다.

그림 8-1 인라인 터미네이터를 사용하는 F/W SCSI 버스



인라인 SCSI 터미네이터를 사용하면 하드웨어를 중지하지 않고 일시적으로 패키지를 다른 노드로 이동한 다음 원래 노드를 중지하여 특정 노드의 하드웨어를 유지 관리할 수 있습니다. 교체한 다음 패키지는 원래 노드로 다시 이동할 수 있습니다.

인라인 SCSI 터미네이터나 Y 케이블을 사용하여 버스에 부착된 노드를 버스에서 분리하려면 다음 단계를 수행하십시오.

1. 유지 관리가 필요한 노드의 패키지를 다른 노드로 이동합니다.
2. 유지 관리가 필요한 노드를 중지합니다. 클러스터는 재편성되어 다른 노드에서 계속 실행됩니다. 중지된 노드의 패키지는 전환 설정된 경우 사용 가능한 다른 노드로 전환됩니다.
3. 노드의 전원을 끕니다.
4. 필요하면 인라인 터미네이터 케이블이나 Y 케이블에서 노드를 분리합니다. 버스에 액세스하는 다른 노드의 경우 인라인 터미네이터나 Y 케이블이 버스에 연결되어 있는 한 문제가 발생하지 않습니다.
5. 필요하면 노드의 하드웨어를 교체하거나 업그레이드합니다.
6. 필요하면 인라인 터미네이터 케이블이나 Y 케이블로 노드를 다시 연결합니다.

클러스터 문제점 해결
디스크 교체

7. 전원을 다시 연결하고 노드를 재시작합니다. `/etc/rc.config.d/cmcluster` 파일에서 `AUTOSTART_CMCLD`가 1로 설정되면 노드는 클러스터에 다시 참여합니다.
8. 필요하다면 패키지를 대체 위치에서 노드로 다시 이동한 다음 재시작합니다.

I/O 카드 교체

I/O 카드에 장애가 발생하여 이 카드를 교체해야 할 경우에는 다음 단계대로 교체합니다. 각 노드에서 인라인 SCSI 터미네이터 또는 Y 케이블을 사용중이라면 이 작업을 실행하기 위해 클러스터를 중지하지 않아도 됩니다.

1. SAM 또는 `cmhaltnode` 명령을 사용하여 노드를 중지합니다. 패키지는 일반적으로 다른 노드에 대한 장애 조치를 수행해야 합니다.
2. 카드에서 I/O 케이블을 제거합니다. 인라인 SCSI 터미네이터를 사용하면 버스 상의 디스크나 다른 노드에게 영향을 미치지 않고도 실행할 수 있습니다.
3. SAM을 사용하여 I/O 카드의 온라인 교체 실행 옵션을 선택합니다.
4. 결함 있는 I/O 카드를 제거합니다.
5. 새 I/O 카드를 설치합니다. 새 카드는 정확하게 동일한 종류의 카드여야 하며 카드를 제거했던 동일한 슬롯에 설치해야 합니다.
6. SAM에서 새 I/O 카드 연결 옵션을 선택합니다.
7. SAM 또는 `cmrunnode` 명령을 사용하여 클러스터에 노드를 다시 추가합니다.

LAN 카드 교체

LAN 카드에 장애가 발생하여 LAN 카드를 교체해야 할 경우 실행하는 하드웨어 종류와 운영 체제에 따라 온라인이나 오프라인으로 교체할 수 있습니다. 이 작업을 위해 클러스터 중지할 필요는 없습니다.

오프라인으로 교체

다음 단계는 오프라인으로 LAN 카드를 교체하는 방법을 보여줍니다. 이 단계는 HP-UX 11.0와 11i에 모두 적용하십시오.

1. `cmhaltnode` 명령을 사용하여 노드를 중지합니다.
2. `/etc/shutdown`을 사용하여 시스템을 종료한 다음 시스템 전원을 끕니다.
3. 결합 있는 LAN 카드를 제거합니다.
4. 새 LAN 카드를 설치합니다. 새 카드는 정확하게 동일한 종류의 카드이어야 하며 카드를 제거했던 동일한 슬롯에 설치해야 합니다.
5. 시스템을 켭니다.
6. 필요하다면 `cmrunnode` 명령을 사용하여 클러스터에 노드를 다시 추가합니다. 노드가 자동으로 클러스터에 참여하도록 구성한 경우 이 단계를 생략할 수 있습니다.

온라인으로 교체

시스템 하드웨어에서 핫스왑 I/O 카드를 지원하고 시스템에서 HP-UX 11i(B.11.11 이상)를 실행 중인 경우 결합 있는 LAN 카드를 온라인으로 교체할 수 있는 옵션이 생깁니다. 이렇게 하면 전반적인 시스템 가용성이 향상됩니다. 이렇게 하려면 **주변 장치를 위한 HP-UX 구성** 설명서에서 “SAM을 사용하여 PCI 카드를 온라인으로 교체하는(OLR) 방법” 절에 설명된 단계를 수행하십시오. 네트워크 드라이버가 잘못 작동되지 않도록 하려면 OLR 절차에서 새 카드가 제거한 카드와 같은 종류인지 확인해야 합니다. ServiceGuard는 LAN 카드가 교체되고 네트워크에 연결되면 이 카드를 자동으로 복구합니다.

카드 교체 후

LAN 카드를 온라인이나 오프라인으로 교체하면 ServiceGuard는 클러스터 이진 구성 파일에 저장한 값에서 카드의 MAC 주소(LLA)가 변경되었는지 검사하고 클러스터의 다른 노드에 새로운 MAC 주소를 알립니다. 그 이후에 클러스터는 정상적으로 작동됩니다.

또한, 클러스터 구성을 다시 적용하여 클러스터의 이진 구성 파일에서 새로운 MAC 주소를 업데이트하는 것도 좋습니다. 다음 단계대로 온라인 재구성을 수행하십시오.

1. 다음과 같이 `cmgetconf` 명령을 사용하여 최신 ASCII 구성 파일을 얻습니다.

```
# cmgetconf config.ascii
```

2. `cmapplyconf` 명령을 사용하여 구성을 적용하고 모든 클러스터 노드에 새 이진 구성 파일을 복사합니다.

```
# cmapplyconf -C config.ascii
```

이 과정은 새로운 MAC 주소로 이진 파일을 업데이트하여 `cmviewconcl` 및 `lanscan` 명령의 출력 결과 간에 데이터가 일치하도록 만듭니다.

문제점 해결 방법

다음 절에서는 실행 중인 시스템의 상태를 검사하고 클러스터 상태 데이터, 로그 파일 및 구성 파일을 조사하여 문제를 해결하는 몇 가지 방법을 설명합니다. 다음과 같은 내용을 설명합니다.

- 패키지 IP 주소 검사
- 시스템 로그 파일 검사
- 구성 파일 검사
- 패키지 제어 스크립트 검사
- `cmquerycl` 및 `cmcheckconf` 사용
- `cmscancl` 및 `cmviewcl` 사용
- LAN 구성 검사

주

클러스터의 현재 상태를 관찰하고 클러스터 개체의 속성을 보기 위해서는 **ServiceGuard Manager**를 사용하는 편이 좋습니다. **ServiceGuard Manager**의 실행에 대해서는 7장의 “**ServiceGuard Manager 사용**” 절을 참조하십시오.

패키지 IP 주소 검사

`netstat -in` 명령은 LAN 구성을 검사하는 데 사용할 수 있습니다. 노드 `ftsys10`을 중지한 다음 `ftsys9`에 대해 이 명령을 실행하면 다음과 같이 패키지의 IP 주소가 하트 비트의 IP 주소와 함께 `ftsys9`의 `lan0`에 지정됩니다.

Name	Mtu	Network	Address	Ipkts	Opkts
ni0*	0	none	none	0	0
nil*	0	none	none	0	0
lo0	4608	127	127.0.0.1	10114	10114
lan0	1500	15.13.168	15.13.1710.14	959269	305189
lan0	1500	15.13.168	15.13.1710.23	959269	305189
lan0	1500	15.13.168	15.13.171.20	959269	305189
lan1*	1500	none	none	418623	41716

시스템 로그 파일 검사

클러스터 관리자 및 패키지 관리자의 메시지는 시스템 로그 파일에 기록됩니다. 로그 파일의 기본 위치는 `/var/adm/syslog/syslog.log`입니다. `vi` 같은 텍스트 편집기나 `more` 명령을 사용하여 클러스터의 기록된 정보에 대한 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

MC/ServiceGuard 및 HP-UX에서 실행 중인 기타 다른 제품은 `syslog.log` 파일을 사용하여 메시지를 저장합니다. HP-UX의 **시스템 및 작업 그룹 관리** 설명서는 시스템 로그 사용에 대한 추가 정보를 제공합니다.

예제 시스템 로그 항목

`/var/adm/syslog/syslog.log` 파일의 다음 항목은 `pkg5_run` 스크립트의 문제로 인해 실행되지 못한 패키지를 보여줍니다. 자세한 내용은 `pkg5_run.log`를 참조하십시오.

```
Dec 14 14:33:48 star04 cmclld[2048]: Starting cluster management protocols.
Dec 14 14:33:48 star04 cmclld[2048]: Attempting to form a new cluster
Dec 14 14:33:53 star04 cmclld[2048]: 3 nodes have formed a new cluster
Dec 14 14:33:53 star04 cmclld[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 successfully.
Dec 14 14:34:44 star04 CM-CMD[2054]: cmrunpkg -v pkg5
Dec 14 14:34:44 star04 cmclld[2048]: Request from node star04 to 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 exited with
    NO_RESTART.
Dec 14 14:34:45 star04 cmclld[2048]: Examine the file
    /etc/cmcluster/pkg5/pkg5_run.log for more details.
```

다음은 성공적으로 시작하는 패키지의 예제입니다.

```
Dec 14 14:39:27 star04 CM-CMD[2096]: cmruncl
Dec 14 14:39:27 star04 cmclld[2098]: Starting cluster management protocols.
Dec 14 14:39:27 star04 cmclld[2098]: Attempting to form a new cluster
Dec 14 14:39:27 star04 cmclconfd[2097]: Command execution message
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 membership is:
    star04(id=1), star05(id=2), star06(id=3)
Dec 14 17:39:33 star04 cmlvmd[2099]: Clvmd initialized successfully.
```

클러스터 문제점 해결 문제점 해결 방법

```
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]: cmmmodnet -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 node star04.
```

개체 관리자 로그 파일 검사

ServiceGuard 개체 관리자 데몬 cmomd 메시지는 `/var/opt/cmom/cmomd.log` 파일에 로그됩니다. 다음과 같이 `cmreadlog` 명령을 사용하여 이들 메시지를 검사할 수 있습니다.

```
# cmreadlog /var/opt/cmom/cmomd.log
```

cmomd 메시지는 데이터 종류, 시간 스탬프 같이 개체 관리자의 데이터를 요청하는 프로세스 정보를 포함합니다. 개체 관리자의 데이터를 요청하는 클라이언트의 예로서 ServiceGuard Manager를 들 수 있습니다.

ServiceGuard Manager 로그 파일 검사

ServiceGuard Manager는 사용자 작업 로그 파일을 관리합니다. 이 파일은 HP-UX 디렉토리 `/var/opt/sgmgr` 또는 Windows 디렉터리

`X:\Program Files\Hewlett-Packard\ServiceGuard Manager\log`(여기서 X는 ServiceGuard Manager를 설치한 드라이브를 나타냄)에 저장됩니다. 아래 HP-UX 예제에서 보듯이 `cmreadlog` 명령을 사용하여 이 메시지를 검사할 수 있습니다.

```
# cmreadlog /var/opt/sgmgr/929917sgmgr.log
```

ServiceGuard Manger의 메시지에는 로그인 날짜와 시간, 개체 관리자 서버 시스템, 시간 스탬프 등에 대한 정보가 포함됩니다.

구성 파일 검사

다음의 ASCII 구성 파일을 검사하십시오.

- 클러스터 구성 파일
- 패키지 구성 파일

구성 계획 워크시트에 따라 파일이 완전하고 올바른지 확인하십시오.

패키지 제어 스크립트 검사

패키지에서 실행할 수 있는 모든 노드에 패키지 제어 스크립트가 있는지, 이 파일이 모든 노드에서 동일한지 확인하십시오. 이 스크립트가 모든 노드에서 실행 가능한지 확인하십시오. 제어 스크립트의 이름이 패키지 구성 파일에 나타나는지 확인하고 패키지 구성 파일에서 명명된 모든 서비스도 패키지 제어 스크립트에 나타나는지 확인하십시오.

각 패키지의 시작 및 중지 관련 정보를 패키지의 제어 스크립트 로그에서 찾을 수 있습니다. 이 로그는 패키지 제어 스크립트의 실행 내역을 제공합니다. 이 내역은 `/etc/cmcluster/package_name/control_script.log`에서 볼 수 있습니다. 이 로그는 실행 중인 모든 패키지와 중지된 작업을 기록합니다. 패키지에 대해 별도의 실행 및 중지 스크립트를 작성했다면 각 스크립트는 자신의 로그를 갖습니다.

cmcheckconf 명령 사용

또한, 구성을 확인할 때 사용했던 대로 `cmcheckconf` 명령을 사용하여 클러스터 문제를 해결할 수 있습니다.

다음 예제는 `ftsys9` 및 `ftsys10`의 기존 클러스터 구성을 확인하는 데 사용된 명령을 보여줍니다.

```
# cmquerycl -v -C /etc/cmcluster/verify.ascii -n ftsys9 -n ftsys10
# cmcheckconf -v -C /etc/cmcluster/verify.ascii
```

`cmcheckconf` 명령은 다음 사항을 확인합니다.

- 네트워크 주소 및 연결
- 클러스터 잠금 디스크 연결
- 클러스터 및 패키지의 구성 매개 변수에 대한 유효성
 - 이름의 고유성
 - 스크립트의 존재 여부와 사용 권한

다음 사항을 확인하지 않습니다.

- 전원 회로의 올바른 설치
- 패키지 구성 스크립트의 정확성

cmsscanc1 명령 사용

cmsscanc1 명령은 IP 주소 또는 서브넷, 디스크의 실제 볼륨 이름, 클러스터의 모든 노드에 있는 노드에 특정한 기타 항목을 비교할 수 있도록 체계적인 보고서에 클러스터의 모든 노드에 대한 정보를 표시합니다. cmsscanc1은 실제로 모든 노드에 대해 몇 가지 다른 HP-UX 명령을 실행하고 그 출력 결과를 명령을 실행한 노드의 보고서에 기록합니다.

다음은 cmsscanc1가 각 노드에 대해 표시하는 구성 데이터 종류입니다.

표 8-1

cmsscanc1 명령에 의해 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 실제 볼륨 그룹 데이터	/etc/lvmpvg 파일
모든 링크의 링크 수준 연결	linkloop 명령
이진 구성 파일	cmviewconf 명령

cmviewconf 명령 사용

cmviewconf를 사용하면 클러스터가 실행되고 있지 않을 때에도 이진 클러스터 구성 파일을 검사할 수 있습니다. 이 명령은 명령을 실행 중인 노드에 있는 구성 파일의 내용을 표시합니다.

LAN 구성 검사

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.
- lanscan도 LAN 구성을 검사하는 데 사용할 수 있습니다. 이 명령은 노드에 있는 각 LAN 인터페이스 카드의 MAC 주소와 상태를 나열합니다.
- arp -a는 arp 테이블을 확인하는 데 사용할 수 있습니다.
- landiag는 LAN 카드 정보를 표시, 진단, 재설정하는 데 유용합니다.

- linkloop는 MAC 주소 수준에서 LAN 카드 사이의 통신을 검사합니다. 예를 들면 다음과 같습니다.

```
# linkloop -i4 0x08000993AB72
```

그 결과로 다음과 같은 메시지를 볼 수 있습니다.

```
Link Connectivity to LAN station: 0x08000993AB72 OK
```

- cmscancl은 기본 및 대기 LAN이 동일한 브리지 연결 네트워크에 있는지 확인하는데 사용할 수 있습니다.
- cmviewcl -v는 기본 및 대기 LAN의 상태를 보여줍니다.

모든 노드에 대해 이 명령을 사용하십시오.

문제점 해결

MC/ServiceGuard의 문제점은 여러 가지 종류일 수 있습니다. 다음은 일반적인 문제 범주 목록입니다.

- MC/ServiceGuard 명령 중지
- 클러스터 재편성
- 시스템 관리 오류
- 패키지 제어 스크립트 중지
- VxVM 디스크 그룹의 문제
- 패키지 이동 오류
- 노드 및 네트워크 장애
- Quorum 서버 문제

처음 두 가지 문제의 범주는 잘못된 MC/ServiceGuard의 구성으로 인해 발생합니다. 마지막 범주에는 MC/ServiceGuard가 반응하여 응용 프로그램의 가용성을 확보할 수 있도록 설계된 “보통” 장애가 포함됩니다.

ServiceGuard 명령 중지

cmviewcl을 포함한 많은 MC/ServiceGuard 명령은 클러스터 노드의 주소를 확인하기 위해 이름 확인 서비스를 사용합니다. 이름 서비스를 사용할 수 없으면(예를 들어, 이름 서버가 중지된 경우) ServiceGuard 명령은 중지되거나 네트워크 관련 오류 메시지를 반환할 수 있습니다. 이러한 경우 각 클러스터 노드에 대해 nslookup 명령을 사용하여 이름 확인이 올바른지 확인할 수 있습니다. 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 출력 결과에 노드의 올바른 IP 주소가 포함되어 있지 않은 경우 이름 확인 서비스를 보다 자세히 확인하십시오.

클러스터 재편성

현재의 클러스터 상황에 따라 클러스터가 재편성될 수도 있습니다. 그 원인 중 일부는 다음과 같습니다.

- 클러스터 `NODE_TIMEOUT` 값보다 전환이 오래 걸릴 경우 이더넷 LAN에서의 로컬 전환. 이러한 문제가 발생하지 않도록 하기 위해 클러스터의 `NODE_TIMEOUT` 값을 늘리거나 다른 LAN 종류를 사용할 수 있습니다.
- 하트비트 LAN의 과도한 네트워크 통신량. 이 문제가 발생하지 않도록 하기 위해 전용 하트비트 LAN을 사용하거나 통신량이 보다 적은 LAN을 사용할 수 있습니다.
- 전체 I/O 및 네트워크 통신량에 의한 시스템 과부하
- 매우 큰 라우팅 테이블을 사용하는 경우처럼 부적절하게 구성된 네트워크

이 경우 응용 프로그램은 클러스터 재편성 중에 다소의 성능상 문제가 있을 수 있으나 계속 실행됩니다.

시스템 관리 오류

클러스터를 시작할 때 표시되지는 않지만 MC/ServiceGuard를 구성할 때 상당한 오류가 발생할 수 있습니다. 하드웨어 또는 소프트웨어 장애가 발생하고 예상대로 패키지 제거가 다른 노드로 전송되기 전까지는 클러스터가 실행되고 아무런 문제가 없는 것처럼 보일 수 있습니다.

이러한 오류는 특히 클러스터 구성 파일과 패키지 구성 스크립트의 오류 때문에 발생합니다. 이러한 오류의 예제는 다음과 같습니다.

- 대체 노드에 볼륨 그룹이 정의되어 있지 않습니다.
- 대체 노드에 마운트 지점이 없습니다.
- 대체 노드의 네트워크 오류(구성 오류)
- 대체 노드에서 사용자 정보가 올바르지 않습니다.

다음 명령을 사용하여 디스크의 상태를 검사할 수 있습니다.

- `bdf` - 패키지 볼륨 그룹이 마운트되었는지 확인
- `vgdisplay -v` - 모든 볼륨이 존재하는지 확인
- `lvdisplay -v` - 미러가 동기화되었는지 확인
- `strings /etc/lvmtab` - 구성이 올바른지 확인
- `ioscan -fnC disk` - 실제 디스크 확인
- `diskinfo -v /dev/rdisk/cxtydz` - 디스크에 대한 정보 표시

- `lssfs /dev/dsk/*d0 -LV` 및 경로 확인
- `vxdg list - VERITAS` 디스크 그룹 나열
- `vxpriint - VERITAS` 디스크 그룹 세부 사항 표시

패키지 제어 스크립트 중지 또는 장애

`RUN_SCRIPT_TIMEOUT` 또는 `HALT_SCRIPT_TIMEOUT` 값이 설정되어 있을 때 제어 스크립트가 중지되어 제한 시간이 초과되면 MC/Serviceguard는 스크립트를 삭제하고 패키지를 “Halted”를 표시합니다. 마찬가지로 패키지 제어 스크립트에 오류가 발생하면 MC/Serviceguard는 스크립트를 삭제하고 패키지를 “Halted”를 표시합니다. 두 경우 모두 다음과 같은 상황이 발생합니다.

- 패키지의 제어 권한이 전송되지 않습니다.
- 실행 또는 중지 명령이 완료되지 않습니다.
- 전체 전환이 일어나지 않습니다.
- 현재 노드가 해당 패키지를 실행할 수 없게 됩니다.

이러한 장애 이후에 제어 스크립트가 종료되므로 일부 패키지의 자원은 계속 활성 상태를 유지할 수 있습니다. 이 경우는 다음과 같습니다.

- 볼륨 그룹이 활성화되어 있습니다.
- 파일 시스템이 계속 마운트되어 있습니다.
- IP 주소가 계속 설정되어 있습니다.
- 서비스가 계속 실행됩니다.

이러한 상황에서 수동으로 시작하지 않는 한 MC/Serviceguard는 패키지를 재시작하지 않습니다. 패키지를 재시작하기 전에 수동으로 정리해야 합니다. 지침 EOFH 다음 단계를 수행하십시오.

1. 응용 프로그램에 특정한 정리 작업을 수행합니다. 대체 노드에서 패키지를 성공적으로 시작하기 위해 제어 스크립트가 수행했을 가능성이 있는 응용 프로그램에 특정한 작업은 실행 취소해야 합니다. 여기에는 응용 프로그램 프로세스 종료, 잠금 파일 제거, 임시 파일 제거와 같은 작업이 포함됩니다.
2. 패키지 IP 주소가 시스템에서 제거되었는지 확인합니다. 이 단계는 `cmmodnet (1m)` 명령을 통해 수행됩니다. 먼저 `netstat -in` 명령을 사용하여 설정된 패키지 IP 주소를 확인하십시오. 패키지 제어 스크립트에 지정된 IP 주소가 “Address” 열 아래의 `netstat` 출력에 나타날 경우 `cmmodnet`를 사용하여 제거하십시오.

```
# cmmodnet -r -i <ip-address> <subnet>
```

여기서 <ip-address>는 “Address” 열의 주소이고 <subnet>은 “Network” 열의 해당 항목입니다.

- 패키지 볼륨 그룹이 비활성화되어 있는지 확인합니다. 먼저, 파일 시스템에 사용할 패키지 논리 볼륨을 언마운트하십시오. 이 작업은 `bf -l` 명령을 실행하여 확인할 수 있습니다. 패키지 제어 스크립트의 `LV[]` 어레이 변수에 의해 지정된 것과 같은 패키지 논리 볼륨이 “Filesystem” 열 아래에 나타날 경우 `umount`를 사용하여 언마운트하십시오.

```
# fuser -ku <logical-volume>
# umount <logical-volume>
```

다음에는 패키지 볼륨 그룹을 비활성화하십시오. 이들 그룹은 패키지 제어 스크립트의 `VG[]` 어레이 항목에 의해 지정됩니다.

```
# vgchange -a n <volume-group>
```

- 마지막으로 패키지 전환을 다시 사용 설정합니다.

```
# cmmodpkg -e <package-name>
```

시간 제한이 발생한 노드를 정리한 다음에는 대체 노드를 구성하여 패키지를 실행하는 것이 좋습니다. 노드를 실행하려면 패키지를 다시 사용 설정해야 합니다.

```
# cmmodpkg -e -n <node-name> <package-name>
```

기본 ServiceGuard 제어 스크립트는 응용 프로그램의 실행 및 중지에는 필요한 간단한 과정을 수행하기 위해 제공됩니다. 패키지 관리자가 이러한 단계를 수행하기 위해 필요한 시간 제한을 지정한 다음 이 제한이 어떤 이유로 초과된 경우 ServiceGuard는 제어 스크립트의 논리가 중지되거나 손상된 경우에만 조치를 취합니다. 이때 제어 스크립트는 정리 작업을 제대로 수행할 것으로 기대되지 않기 때문에 스크립트는 종료되고 패키지 관리자는 필요한 정리 과정을 계획해야 합니다.

제어 스크립트 시간 제한에 도달했을 때 자동으로 패키지를 전환하려면

`NODE_FAIL_FAST_ENABLED` 매개 변수를 `YES`로 설정하십시오. `SAM`을 사용하는 경우 `Package Failfast`를 `Enabled`로 설정하십시오. 이 경우 ServiceGuard는 제어 스크립트 시간 제한이 발생한 노드에서 `TOC`를 시작합니다. 결과적으로 패키지의 실행 또는 중지 시도에 따른 부작용이 효과적으로 제거됩니다. 이 경우 패키지가 구성된 사용 가능한 대체 노드에서 패키지가 자동으로 재시작됩니다.

VxVM 디스크 그룹의 문제

이 절에서는 클러스터 환경에서 VxVM 디스크 그룹에 발생한 문제를 해결하는 몇 가지 방법에 대해 설명합니다. 대부분의 문제의 경우 `vxdg list` 명령을 사용하여 특정 노드에 가져온 디스크 그룹을 표시하는 것이 유용합니다. 또한, 특정 노드에 디스크 그룹 가져오기오하 내보내기에 대한 메시지에 대해 패키지 제어 스크립트 로그 파일을 확인하십시오.

노드 장애 이후 가져오기와 내보내기 지정

장애가 발생하면 VxVM 디스크 그룹으로 구성된 패키지를 시작하지 못하고 다음과 같은 오류가 패키지 로그 파일에 나타납니다.

```
vxdg: Error dg_01 may still be imported on ftsys9
      ERROR: Function check_dg failed
```

이 패키지를 한 노드에서 실행 중이고 패키지 제어 스크립트에서 디스크 그룹을 내보내기 전에 장애가 발생한 경우 이러한 상황이 발생할 수 있습니다. 이런 경우 장애가 발생한 노드의 호스트 이름은 디스크 그룹 헤더에 여전히 기록됩니다.

패키지가 클러스터의 다른 노드에서 시작되면 아래 예제에서처럼 메시지가 표시됩니다. 이 예제에서 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 `vxdg(1M)`.

```
*****
```


To force import dg_01, execute the following commands on the local system:

```
vx dg -tfC import $vg
vx dg deport $vg
```

메시지에 나타난 지침대로 가져오기 옵션(-C)을 사용하여 현재 노드에서 디스크 그룹을 가져옵니다. 그런 다음, 디스크 그룹을 내보냅니다. 이렇게 하면 패키지에서 다시 이 디스크 그룹을 사용할 수 있습니다. 예를 들면 다음과 같습니다.

```
# vx dg -tfC import dg_01
# vx dg deport dg_01
```

가져오기를 지정하면 현재 디스크 그룹의 디스크에 기록된 호스트 이름을 지운 후에 오류 발생 없이 디스크 그룹을 내보내면 다른 노드에서 실행 중인 패키지에서 가져올 수 있습니다.

주의

이러한 강제 가져오기 절차는 현재 다른 노드에서 디스크를 액세스하고 있지 않은 경우에만 사용해야 합니다. 다른 노드에서 액세스 중인 디스크를 강제로 가져오면 데이터가 손상될 수 있습니다.

패키지 이동 오류

이러한 오류는 특히 패키지 제어 스크립트 오류에 의해 발생했다는 점을 제외하고 시스템 관리 오류와 유사합니다. 이러한 오류를 방지하는 가장 좋은 방법은 고가용성 응용 프로그램을 온라인 상태로 바꾸기 전에 패키지 제어 스크립트를 테스트하는 것입니다.

제어 스크립트의 두 번째 줄에 "set -x" 문을 추가하면 스크립트 오류가 발생할 수 있는 상황에 대한 자세한 정보를 볼 수 있습니다.

노드 및 네트워크 장애

이런 장애가 발생하면 MC/ServiceGuard는 패키지의 제어를 다른 노드로 전송합니다. 이것은 MC/ServiceGuard의 일반적인 동작이지만 사용자는 전송이 발생한 때를 알 수 있어야 하고 클러스터를 계속 현재 상태로 둘 것인지 또는 원래 상태로 복원할 것인지를 결정해야 합니다.

다음 상황에서 노드 장애가 발생할 수 있습니다.

- HPMC(High Priority Machine Check). 높은 우선 순위 시스템 검사로, 하드웨어 오류로 인한 시스템 중지를 의미합니다.
- TOC
- 시스템 중지
- 중지
- 전원 장애

TOC의 경우에는 실패한 노드에서 시스템 덤프가 수행되고 콘솔에 다양한 메시지가 표시됩니다.

다음 명령을 사용하여 네트워크 및 서브넷의 상태를 확인할 수 있습니다.

- netstat -in - LAN 상태를 표시하고 해당 패키지 IP가 LAN 카드에 지정되었는지 확인
- lanscan - LAN이 기본 인터페이스에 있는지 또는 대기 인터페이스로 전환되었는지 확인
- arp -a - arp 테이블 확인
- lanadmin - LAN 카드 표시, 테스트, 재설정

사용하는 클러스터가 모두 동일하지 않기 때문에 모든 문제에 대한 완전한 해결책은 없습니다. 그러나 이러한 검사와 명령들을 수행하고 로그 파일을 참조하여 작업하면 성공적으로 문제를 찾아 해결할 수 있을 것입니다.

Quorum 서버 문제점 해결

인증 파일 문제

ServiceGuard 노드의 syslog 파일 또는 cmviewc1 -v 명령 실행 결과에 다음과 같은 종류의 메시지가 나타난다면 인증 문제가 있는 것입니다.

```
Access denied to quorum server 192.6.7.4
```

그 이유는 인증 파일을 업데이트하지 않았기 때문입니다. 노드가 파일에 포함되어 있는지 확인하고 /usr/sbin/qs -update를 사용하여 인증 파일을 다시 읽어 보십시오.

시간 제한 문제

ServiceGuard 노드의 syslog 파일에 다음과 같은 종류의 메시지가 나타난다면 시간 제한 문제가 있는 것입니다.

```
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
```

이 메시지가 나타나는 이유는 네트워크 연결이 자주 끊어지거나 기본 quorum 서버 시간 제한이 충분히 설정되지 않았기 때문입니다. QS_TIMEOUT_EXTENSION을 설정하여 시간 제한을 늘리거나 하트비트 또는 노드 시간 제한 값을 늘릴 수 있습니다.

ServiceGuard 노드의 syslog 파일에 다음과 같은 종류의 메시지가 나타난다면 노드가 잠금 요청에 대한 응답을 수신하지 않은 것입니다. 이것은 노드와 qs 사이에 또는 qs와 클러스터의 다른 노드 사이에 통신 지연이 발생했기 때문입니다.

```
Attempt to get lock /sg/cluser1 unsuccessful. Reason: request_timedout
```

메시지

ServiceGuard의 코디네이터 노드는 잠금 상태를 설정하는 데 quorum 서버로 요청을 보내기도 합니다. (이 요청은 재편성 잠금을 얻기 위한 요청과는 다릅니다.) quorum 서버가 클러스터 노드 중 하나에 연결되지 않으면 설정 요청은 실패하며 quorum 서버의 로그 파일에 다음과 같이 두 행으로 된 메시지가 나타납니다.

```
Oct 008 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.
```

클러스터 문제점 해결 문제점 해결

이러한 상태는 무시할 수 있습니다. 요청이 몇 초 후에 재시도되어 성공적으로 수행되었습니다. 다음과 같은 메시지가 기록됩니다.

```
Oct 08 16:10:06: Request for lock /sg/lockTest1  
succeeded. New lock owners: 1,2.
```

A Mc/ServiceGuard 명령

다음은 ServiceGuard 클러스터 구성 및 유지 보수에 사용되는 명령을 알파벳 순서로 정렬한 목록입니다. 설치가 끝난 후 시스템에서 이 명령들의 온라인 도움말을 사용할 수 있습니다.

표 A-1 MC/ServiceGuard 명령

온라인 도움말	설명
cmapplyconf	<p>ServiceGuard 클러스터 구성 및 패키지 구성 파일을 검사하여 적용합니다.</p> <p>cmapplyconf는 cluster_ascii_file 및 관련된 pkg_ascii_file에 지정된 클러스터 구성과 패키지 구성을 검사하고, 이진 구성 파일 cmclconfig를 작성 또는 업데이트한 다음, 전체 노드에 배포합니다. 이 이진 구성 파일에는 클러스터 구성 정보와 함께 지정된 모든 패키지의 패키지 구성 정보가 수록됩니다. 이 파일은 클러스터 데몬이 전체 클러스터와 패키지 환경을 관리하는 데 사용하며 /etc/cmcluster 디렉토리에 저장됩니다.</p> <p>클러스터 구성이나 패키지 구성 파일 가운데 하나를 변경해야 할 때는 먼저 해당 ASCII 파일(클러스터 또는 패키지)을 수정하고 cmcheckconf 명령을 사용하여 변경 사항을 검사한 다음, cmapplyconf를 다시 사용하여 이진 파일을 검사하고 모든 노드에 배포합니다. 클러스터 및 패키지 구성은 클러스터가 다운되었을 경우에만 수정할 수 있습니다. 처음 클러스터를 구성하는 경우나 클러스터에서 노드를 추가 또는 삭제하는 경우에는 클러스터 ASCII 파일만 지정하면 됩니다. 패키지를 추가하거나 패키지 구성을 수정하는 경우에는 패키지 ASCII 파일만 지정하면 됩니다.</p>

표 A-1

MC/ServiceGuard 명령

온라인 도움말	설명
cmapplyconf (계속)	기존 구성을 변경할 때는 cmgetconf 명령을 실행하여 클러스터 ASCII 구성 파일 또는 패키지 ASCII 구성 파일을 준비하는 것이 좋습니다. cmapplyconf 명령이 클러스터 구성 또는 패키지 파일을 검사하고 배포합니다. 이 명령은 클러스터 구성에서 클러스터 데몬을 시작하거나 제거하지 않습니다. 패키지 구성에서도 같은 방법으로 패키지 노드, 패키지 서브넷 등을 추가 또는 삭제할지 결정합니다. 패키지 구성을 변경할 때는 언제나 패키지를 중지해야 합니다.
cmcheckconf	고가용성 클러스터 구성 및 패키지 구성 파일을 검사합니다. cmcheckconf은 cluster_ascii_file에 의해 지정된 클러스터 구성 및/또는 명령의 각 pkg_ascii_file에 의해 지정된 패키지 구성 파일을 검사합니다. 이전에 클러스터가 구성되었다면 cmcheckconf 명령은 cluster_ascii_file의 구성과 이전 구성 파일에 저장되어 있는 이전의 구성 정보를 비교하여 변경 사항을 검사합니다. 이와 동일한 규칙이 pkg_ascii_file에도 적용됩니다. cmcheckconf 명령을 실행하기 위해서는 클러스터를 중지해야 합니다.

표 A-1

MC/ServiceGuard 명령

온라인 도움말	설명
cmdeleteconf	<p>클러스터 구성 또는 패키지 구성을 삭제합니다.</p> <p>cmdeleteconf는 클러스터의 모든 패키지를 포함하여 클러스터 구성 전체를 삭제하거나 지정된 패키지 구성만 삭제합니다. <i>cluster_name</i>과 <i>package_name</i>을 모두 지정하지 않으면 cmdeleteconf는 로컬 클러스터의 구성과 이 구성의 모든 패키지를 삭제합니다. <i>package_name</i>만 지정하면 로컬 클러스터에 있는 <i>package_name</i>의 구성만 삭제됩니다.</p> <p><i>cluster_name</i>과 <i>package_name</i>을 모두 지정하면 지정된 패키지는 <i>cluster_name</i>에 구성되어 있어야 하며 <i>package_name</i>에 해당하는 패키지만 삭제됩니다. 로컬 클러스터는 cmdeleteconf 명령이 실행되는 노드를 포함하는 클러스터입니다.</p>
cmgetconf	<p>클러스터 또는 패키지 구성 정보를 얻습니다.</p> <p>cmgetconf는 클러스터 구성(패키지 구성 제외)이나 지정된 패키지의 구성 정보를 받아 <i>output_filename</i> 파일로 출력하거나 stdout로 출력합니다. 이 명령은 클러스터 실행 여부에 상관 없이 실행할 수 있습니다. <i>cluster_name</i>과 <i>package_name</i>이 지정되어 있지 않으면 cmgetconf는 로컬 클러스터의 구성을 얻습니다. <i>cluster_name</i>과 <i>package_name</i>이 모두 지정되면 지정된 패키지는 <i>cluster_name</i>에 구성되어 있어야 하며 <i>package_name</i>의 패키지 구성만 <i>output_filename</i> 또는 stdout로 출력됩니다.</p>
cmhaltcl	<p>고가용성 클러스터를 중지합니다.</p> <p>cmhaltcl은 구성된 클러스터의 모든 노드가 각자의 클러스터 데몬을 중지하게 합니다. 선택적으로 실행 중인 모든 패키지 또는 응용 프로그램을 중지할 수 있습니다.</p> <p>이 명령은 현재 실행 중인 모든 시스템에서 데몬을 모두 중지시킵니다. 데몬 중 일부만 종료하려면 이 명령 대신 cmhaltnode 명령을 사용해야 합니다.</p>

표 A-1

MC/ServiceGuard 명령

온라인 도움말	설명
cmhaltnode	<p>고가용성 클러스터의 노드를 중지합니다.</p> <p>cmhaltnode는 노드가 자신의 클러스터 데몬을 중지하고 기존 클러스터에서 스스로를 제거하도록 합니다.</p> <p>노드에서 cmhaltnode가 실행되면 클러스터 데몬이 중지되며, 가능한 경우 선택적으로 노드에서 실행 중인 모든 패키지를 다른 노드로 이동할 수 있습니다.</p> <p>node_name이 지정되지 않으면 로컬 노드에서 실행되는 클러스터 데몬은 기존 데몬에서 중지되고 제거됩니다.</p>
cmhaltpkg	<p>고가용성 패키지를 중지합니다.</p> <p>cmhaltpkg는 ServiceGuard 클러스터에서 실행 중인 고가용성 패키지의 수동 중지를 실행합니다. 이 명령은 클러스터의 모든 노드에서 실행할 수 있으며 클러스터 내의 모든 패키지에 대해 수행할 수 있습니다.</p>
cmhaltserv	<p>고가용성 패키지 중지 스크립트의 서비스를 중지합니다. 이 명령은 실행 명령이 아니라, 패키지 제어 스크립트에서만 실행됩니다.</p> <p>cmhaltserv는 서비스를 중지하기 위해 고가용성 패키지 중지 스크립트에 사용됩니다. 패키지의 일부에 표시되면 패키지 중지 스크립트는 복구 과정의 일부로 실행됩니다.</p> <p>이 명령은 SIGTERM를 PID 및 모니터링 서비스의 해당 프로세스 그룹으로 보냅니다. 실행 응용 프로그램이 이 신호를 받는다면 응용 프로그램이 프로세스를 종료해야 합니다.</p>

표 A-1

MC/ServiceGuard 명령

온라인 도움말	설명
cmmakepkg	<p>고가용성 패키지 템플릿 파일을 생성합니다.</p> <p>cmmakepkg는 옵션 선택으로 지정된 템플릿 ASCII 패키지 구성 파일이나 패키지 제어 스크립트를 생성합니다. <i>output_file_name</i> 파일을 특정 클러스터 환경에 대해 사용자 정의해야 합니다. 사용자 정의를 한 다음, cmcheckconf 명령을 사용하여 이 파일을 검사합니다. <i>output_file_name</i> 파일이 없으면 stdout로 출력됩니다.</p>
cmmodnet	<p>고가용성 클러스터에서 주소를 추가 또는 삭제합니다.</p> <p>cmmodnet 는 <i>subnet_name</i> 파일을 실행하는 현재 네트워크 인터페이스에 <i>IP_address</i>를 추가 또는 삭제하기 위해 고가용성 패키지 제어 스크립트에 사용됩니다.</p> <p>패키지 제어 스크립트의 컨텍스트 이외에서 이 명령을 실행할 때는 매우 주의해야 합니다. 이 명령은 장애가 발생하여 “중지” 상태인 변동 가능 IP 주소를 삭제할 때만 사용해야 합니다. 패키지 실행중에 사용하면 클라이언트 연결이 해제될 수 있습니다.</p>
cmmodpkg	<p>고가용성 패키지의 전환 속성을 활성화하거나 비활성화 합니다.</p> <p>cmmodpkg는 패키지 장애 발생 시 패키지가 다른 노드로 전환되는 기능을 사용 설정하거나 해제합니다. 그리고 특정 노드가 특정한 패키지를 실행하는 것을 허용하거나 금지합니다. 패키지 전환은 전체적으로 사용 설정하거나 해제할 수 있습니다. 예를 들어, 전체적으로 사용 해제된 패키지에서 장애가 발생하면 패키지는 다른 노드로 전환되지 않으며, 전체적으로 사용 설정된 패키지에서 장애가 발생하면 패키지가 실행될 수 있도록 구성된 사용 가능한 첫 번째 노드로 전환을 시도합니다.</p>

표 A-1

MC/ServiceGuard 명령

온라인 도움말	설명
cmquerycl	<p>클러스터 또는 노드 구성 정보에 대해 질의합니다.</p> <p>cmquerycl은 모든 노드에서 클러스터 구성과 논리 볼륨 관리자(LVM) 정보를 찾습니다. 클러스터 구성 정보에는 LAN 인터페이스, IP 주소, 브리지 연결 네트워크, 가능한 하트비트 네트워크 등의 네트워크 정보가 포함됩니다. LVM 정보에는 볼륨 그룹(VG) 상호 연결과 파일 시스템 마운트 지점 정보가 포함됩니다. 클러스터 구성을 준비하는 첫 단계로서 이 명령을 실행해야 합니다. 이 명령은 현재 클러스터 구성을 파악하기 위한 문제 해결 도구로 사용할 수도 있습니다.</p>
cmreadlog	<p>보기 쉽도록 개체 관리자 로그 파일을 형성합니다.</p> <p>이 명령은 개체 관리자가 관리 개체 파일(MOF) 형식으로 생성한 로그 파일을 읽고, 한 줄에 한 항목씩 보고서에 표시합니다. 이 명령은 개체 관리자 작업의 문제를 해결 또는 검토할 때 사용합니다.</p>
cmruncl	<p>고가용성 클러스터를 실행합니다.</p> <p>cmruncl은 구성된 클러스터의 모든 노드 또는 지정된 모든 노드가 자신의 클러스터 데몬을 실행하고 새로운 클러스터를 편성하게 합니다. 이 명령은 구성된 모든 노드에서 클러스터가 활성화되어 있지 않을 때에만 실행할 수 있습니다. 만약 일부 노드에서 클러스터가 이미 실행 중이면 cmrunnode 명령을 사용하여 나머지 노드를 시작한 다음 기존 클러스터에 참여하도록 해야 합니다.</p>
cmrunnode	<p>고가용성 클러스터의 노드를 실행합니다.</p> <p>cmrunnode는 노드가 자신의 클러스터 데몬을 시작한 다음 기존 클러스터에 참여하게 합니다.</p> <p>노드를 시작해도 활성 패키지가 새 노드로 이동하지는 않습니다. 그렇지만 패키지가 중지되었을 때 전환 속성이 활성화되어 있고 그 패키지가 새 노드에서 실행될 수 있으면 패키지는 새 노드에서 자동으로 실행됩니다.</p>

표 A-1

MC/ServiceGuard 명령

온라인 도움말	설명
cmrunpkg	<p>고가용성 패키지를 실행합니다.</p> <p>cmrunpkg는 이전에 중지된 고가용성 패키지를 실행합니다. 이 명령은 클러스터의 모든 노드에서 실행할 수 있으며 클러스터 내의 모든 패키지에 대해 수행할 수 있습니다. 노드를 지정하지 않으면 명령이 실행 중인 노드가 선택됩니다. 이때, 현재 노드가 패키지를 실행할 수 없거나 패키지의 가능한 소유자 목록에 없으면 오류가 발생합니다. 패키지가 새 노드에서 시작될 때는 패키지의 실행 스크립트가 실행됩니다.</p>
cmrunserv	<p>고가용성 패키지 실행 스크립트의 서비스를 실행합니다. 이 명령은 실행 명령이 아니라, 패키지 제어 스크립트제어 스크립트에서만 실행됩니다.</p> <p>cmrunserv은 서비스를 실행하기 위해 고가용성 패키지 실행 스크립트에 사용됩니다. 서비스 프로세스의 종료 시, cmrunserv 명령은 서비스 상태를 업데이트하여 다운시킵니다. 클러스터 소프트웨어는 상태의 변경을 인식하고 일반적인 패키지 복구 시퀀스를 실행합니다. 여기에는 패키지 중지 스크립트를 실행하고, 서로 다른 노드에서 패키지를 실행할 수 있는지 여부를 결정하고, 패키지 실행이 가능한 경우에는 새 노드에서 패키지 실행 스크립트를 실행하는 것이 포함됩니다.</p> <p>cmhaltserv 명령으로 <i>service_command</i>가 중지되면, SIGTERM 신호를 프로세스로 보내게 됩니다. 이 실행 스크립트 또는 셸 스크립트는 SIGTERM 신호를 처리할 수 있어야 하며, 종료를 실행하여 필요한 모든 정리 작업을 수행할 수 있어야 합니다. 프로세스에서 SIGTERM 신호를 무시하면, SIGKILL 신호를 프로세스로 보냅니다. SIGKILL 신호를 보내면, 프로세스는 즉시 종료되고 정리 작업을 수행할 수 없게 됩니다.</p>

표 A-1

MC/ServiceGuard 명령

온라인 도움말	설명
<p>cmsscanc1</p>	<p>ServiceGuard가 설치된 노드에서 시스템 구성 정보를 수집합니다.</p> <p>cmsscanc1은 구성 보고 및 진단 도구로서, 노드 목록 또는 클러스터에 있는 모든 노드에서 시스템 소프트웨어 및 하드웨어 구성 정보를 수집합니다. 이 명령이 표시하는 정보에는 LAN 장치 구성, 네트워크 상태 및 인터페이스, 파일 시스템, LVM 구성, 링크 수준 연결, 이진 클러스터 구성 파일의 데이터가 포함됩니다. 이 명령은 문제 해결 도구 또는 데이터 수집 도구로 사용할 수 있습니다.</p> <p>output_file이 지정되어 있지 않으면 stdout로 출력됩니다. 출력 파일은 아래를 포함합니다.</p> <ul style="list-style-type: none"> • LAN 장치 구성(lanscan의 출력물) • 네트워크 상태 및 인터페이스(netstat의 출력물) • 파일 시스템(mount의 출력물) • LVM 구성(/etc/lvm/tab 파일의 콘텐츠) • LVM 물리 볼륨 그룹 정보(/etc/lvm/vg 파일의 콘텐츠) • 링크 수준 연결(linkloop의 출력물) • 이진 구성 파일 데이터(cmviewconf의 출력물)
<p>cmstartres</p>	<p>ServiceGuard 패키지에 구성되어 있는 EMS 리소스에 대해 로컬 노드에서의 리소스 모니터링을 시작합니다.</p> <p>cmstartres는 로컬 노드에서 EMS 리소스에 대한 리소스 모니터링을 시작합니다. 이 리소스는 지정된 package_name에 구성되어 있어야 합니다.</p>

표 A-1

MC/ServiceGuard 명령

온라인 도움말	설명
cmstopres	<p>ServiceGuard 패키지에 구성되어 있는 EMS 리소스에 대해 로컬 노드에서의 리소스 모니터링을 중지합니다.</p> <p>cmstopres는 로컬 노드에서 EMS 리소스에 대한 리소스 모니터링을 중지합니다. 이 리소스는 지정된 <code>package_name</code>에 구성되어 있어야 합니다.</p>
cmviewcl	<p>현재고가용성 클러스터에 대한 정보를 표시합니다.</p> <p>cmviewcl은 클러스터의 현재 상태 정보를 표시합니다. 이 명령은 전체 클러스터를 출력하거나 특정 노드 또는 패키지에 한정된 정보를 출력할 수 있습니다.</p>
cmviewconf	<p>ServiceGuard 클러스터 구성 정보를 표시합니다.</p> <p>cmviewconf는 기존 클러스터의 이전 구성 파일로부터 클러스터 구성 정보를 수집하여 ASCII 형식으로 표시합니다. 선택적으로 출력 결과를 파일에 기록할 수 있습니다. 이 명령은 클러스터 구성을 파악하기 위한 문제 해결 도구로 사용할 수도 있습니다.</p>

Mc/ServiceGuard 명령

B Enterprise Cluster Master Toolkit

Enterprise Cluster Master Toolkit(제품 번호 B5121)은 여러 주요 데이터베이스 제품과 인터넷 소프트웨어 제품을 위한 ServiceGuard 패키지를 만드는 데 사용할 수 있는 예제 스크립트와 패키지 구성 파일의 모음을 제공합니다. 각각의 툴킷에는 사용자의 필요에 따라 패키지를 사용자 정의하는 방법을 설명한 README 파일이 수록되어 있습니다.

다음과 같은 HP 도메인 서버 구성 요소를 위한 툴킷이 포함되어 있습니다.

- FastTrack Server
- Enterprise Server
- Enterprise Server Pro
- Proxy Server
- Directory Server
- Messaging Server
- Collabra Server
- Calendar Server

다음과 같은 데이터베이스 제품을 위한 툴킷이 포함되어 있습니다.

- Informix
- Oracle
- Oracle 8i Standby Database
- Progress
- Sybase

가용성을 측정하는 툴킷 또한 포함되어 있으며, 툴킷 이름은 **Foundation Monitor** 툴킷입니다.

별도로 NFS 툴킷을 구할 수 있습니다. 자세한 내용은 *Managing Highly Available NFS*(제품 번호 B5125-90001)를 참조하십시오.

이외에도 HP를 통해 다양한 응용 프로그램 통합 스크립트를 구할 수 있습니다.

C

고가용성 클러스터 응용 프로그램 설계

이 부록에서는 고가용성을 구현하도록 응용 프로그램을 작성하거나 이식하는 방법을 다음과 같은 주제에 중점을 두어 설명합니다.

- 응용 프로그램 작동 자동화
- 응용 프로그램 장애 조치 속도 제어
- 다중 시스템에서 실행 가능한 응용 프로그램 설계
- 클라이언트 연결 복원
- 응용 프로그램 장애 처리
- 계획된 다운시간 최소화

고가용성 설계는 계획되지 않은 다운시간 및 계획된 다운시간을 줄이는 작업입니다. 계획된 다운시간에는 정전, 시스템 장애, 네트워크 장애, 디스크 고장, 응용 프로그램 장애 등과 같은 계획되지 않은 사건들이 포함됩니다. 계획된 다운시간에는 정기적인 백업, 시스템 OS 업그레이드, 하드웨어 교체 등과 같은 계획된 사건들이 포함됩니다.

항상 염두에 두어야 할 핵심 전략이 두 가지 있습니다.

1. 시스템의 재부팅 또는 중지를 처리할 수 있도록 응용 프로그램을 설계합니다. 기존 응용 프로그램을 고가용성 환경에 맞게 수정하려면 시스템 중지 이후에 응용 프로그램에 어떤 일이 발생하는지 관찰하십시오. 고가용성 환경에서는 이런 경우에 응용 프로그램을 재시작하는 정의된(스크립트화된) 절차가 있어야 합니다. 응용 프로그램을 시작하거나 중지하는 절차는 자동으로 수행되어야 하며 사용자의 개입을 필요 없어야 합니다.
2. 응용 프로그램은 다음과 같은 시스템 관련 정보를 사용하면 안됩니다. 그렇지 않으면 응용 프로그램이 다른 시스템으로 장애 조치되지 않으며 정상적으로 실행되지 않습니다.
 - 응용 프로그램은 `uname()` 또는 `gethostname()`를 참조해서는 안됩니다.
 - 응용 프로그램은 SPU ID를 참조해서는 안됩니다.
 - 응용 프로그램은 MAC 주소(링크 수준의 주소)를 참조해서는 안됩니다.

응용 프로그램 작동 자동화

응용 프로그램이 자동으로 시작되고 중지될 수 있습니까? 또는 사용자의 개입이 필요합니까?

이 절에서는 사용자 개입이 필요 없도록 응용 프로그램의 작동을 자동화하는 방법을 설명합니다. 고가용성의 우선 원칙 중 하나는 수동 개입을 최소화하는 것입니다. 만약 응용 프로그램의 하위 시스템을 시작하기 위해 사용자가 터미널, 콘솔 또는 GUI 인터페이스로 명령을 입력한다면 이 시스템에서는 사용자가 핵심적인 요소가 됩니다. 필요한 작업을 위해 사용자가 시스템 콘솔 앞에 도착하기까지 때로는 몇 시간이 걸릴 수도 있습니다. 문제가 발생한 하드웨어가 혼란된 직원이 없는 멀리 떨어진 장소에 있을 수도 있습니다. 또는 시스템이 접근이 통제된 데이터 센터에 있거나 업무 시간 이후에는 모뎀을 통해 접속해야 할 수도 있습니다.

응용 프로그램 재배포의 자동화를 위해 염두에 두어야 할 두 가지 원칙이 있습니다.

- 사용자의 개입이 없어야 합니다.
- 응용 프로그램의 시작 및 종료 절차가 정의되어야 합니다.

응용 프로그램이 실행되고 있는 시스템이 재부팅되었을 때 어떤 일이 발생하는지 파악해야 합니다. 그리고 고가용성을 위해서 응용 프로그램의 반응을 개선해야 하는지 결정해야 합니다.

사용자 개입의 최소화

가능한 사용자의 개입을 최소화하십시오. 여기에는 다음 사항이 포함됩니다.

- 서버의 장애로 연결이 끊겼을 때 사용자가 직접 연결하지 않도록 합니다.
- 가능하면 사용자에게 장애 조치로 인해 약간의 지연이 있음을 알립니다.
- 데이터 재입력을 최소화합니다.
- 사용자의 생산성 저하를 최소화할 수 있도록 시스템에 충분한 여유를 두고 설계합니다.

응용 프로그램 시작과 종료 정의

응용 프로그램은 사용자 개입 없이 다시 시작할 수 있어야 합니다. 하드웨어 장치에서 직접 스위치를 켜야 하는 응용 프로그램은 재시작을 자동화하는 것이 불가능합니다. HA 소프트웨어가 응용 프로그램 시작, 종료, 모니터링을 자동으로 수행하도록 하려면 이 작업에 대한 절차를 만들어야 합니다.

반응을 자동화하기 위해서는 응용 프로그램을 시작하고 종료하는 절차가 정의되어야 합니다. MC/ServiceGuard에서 이 절차는 패키지 제어 스크립트에 기록되어 있습니다. 이 절차는 오류를 검사하여 HA 제어 소프트웨어에 그 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 자원(특정 디스크 드라이브에 대한 액세스 등)이 있는 클러스터의 사용 가능한 시스템에서 응용 프로그램을 재시작합니다. 응용 프로그램은 다음 두 상황에서 재시작할 수 있어야 합니다.

- 응용 프로그램은 백업 시스템(또는 응용 프로그램 재시작 옵션이 선택된 경우 동일한 시스템에서)에서 재시작하여 복구할 수 있어야 합니다.
- 응용 프로그램은 시작 중 장애가 발생할 경우 장애 원인이 해결되면 재시작할 수 있어야 합니다.

응용 프로그램 관리자는 HA 명령으로 응용 프로그램을 시작하고 종료하는 방법을 숙지해야 합니다. 부주의한 응용 프로그램 종료는 직접적으로 원치 않는 장애를 발생시킬 수 있습니다. 또한 응용 프로그램 관리자는 개발 환경에서 테스트를 위한 응용 프로그램 인스턴스 대신 실수로 응용 프로그램의 실제 인스턴스를 종료하지 않도록 주의해야 합니다.

HA 소프트웨어가 응용 프로그램의 장애가 발생한 시간을 알 수 있도록 응용 프로그램의 활성 여부를 모니터링할 수 있는 기능이 필요합니다. 이 기능은 응용 프로그램의 모든 프로세스에 대해 `ps -ef | grep xxx` 명령을 실행하는 간단한 스크립트로 구현할 수 있습니다.

응용 프로그램을 중지하게 되면 백업 시스템으로 불필요한 장애 조치가 수행되기 때문에, 사용자의 영향을 최소화하기 위해서는 오류가 발생했을 때 응용 프로그램을 단순히 중지하면 안 됩니다. 응용 프로그램을 단순히 중지하지 말고 정확한 오류를 파악하여 복구하기 위한 조치를 취해야 합니다.

응용 프로그램 장애 조치 속도 제어

신속한 장애 조치 테스트를 위해 수행해야 하는 절차는 무엇입니까?

장애로 인해 응용 프로그램이 다른 노드로 이동하는 경우(장애 조치) 응용 프로그램이 다시 실행되기까지 소요되는 시간을 줄일 수 있는 여러 가지 방법이 있습니다. 다음과 같은 내용을 다룹니다.

- 데이터가 아닌 파일 시스템 복제
- 원시 볼륨 사용
- JFS 사용 검토
- 데이터 손실 최소화
- 재시작 가능한 트랜잭션 사용
- 검사점 사용
- 다중 서버를 위한 설계
- 복제된 데이터 사이트를 위한 설계

데이터가 아닌 파일 시스템 복제

데이터가 아닌 파일 시스템은 공유하지 말고 복제해야 합니다. 응용 프로그램 데이터의 사본은 하나만 있을 수 있습니다. 데이터는 응용 프로그램을 실행하는 시스템이 액세스할 수 있는 일련의 디스크에 위치할 것입니다. 장애 조치 후 이 데이터 디스크가 파일 시스템이라면 먼저 파일 시스템 복구(fsck)를 수행해야 데이터에 액세스할 수 있습니다. 데이터 파일 시스템이 작을수록 복구가 신속히 수행되기 때문에 이 크기를 줄여 복구 시간을 줄일 수 있습니다. 그러므로, 복제할 수 있는 것들은 모두 데이터 파일 시스템 밖으로 이동하는 것이 좋습니다. 예를 들어, 응용 프로그램의 실행 파일을 하나의 공유 파일 시스템에 위치시키는 것보다는 각 시스템에 모두 위치시키는 것이 좋습니다. 또한 응용 프로그램의 실행 파일을 복제하면 필요할 때 응용 프로그램을 롤링 업그레이드할 수 있습니다.

원시 볼륨 사용

응용 프로그램이 데이터를 사용하면 파일 시스템 대신 원시 볼륨을 사용하십시오. 원시 볼륨은 파일 시스템의 fsck를 요구하지 않기 때문에 장애 조치 시 시간이 많이 걸릴 수 있는 단계 하나가 줄어듭니다.

JFS 사용 검토

파일 시스템을 꼭 사용해야 한다면 HFS에 비해 JFS에서 파일 시스템 복구가 훨씬 빠르다는 점을 참고하십시오. 그렇지만 JFS의 성능은 응용 프로그램에 따라 많은 차이가 있습니다.

데이터 손실 최소화

계획되지 않은 중지가 발생할 경우에 손실될 수 있는 데이터의 양을 최소화하십시오. 장애 발생 시 일부 데이터가 손실되는 것을 피할 수는 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 상주 데이터(메모리 상주 구문)가 손실됩니다. 데이터를 쉽게 다시 계산할 수 있는 경우가 아니면 응용 프로그램은 메모리 상주 데이터의 크기를 최소화하도록 설계되어야 합니다. 응용 프로그램은 대기 노드에서 재시작될 때 메모리에 있어야 하는 정보들을 다시 계산하거나 디스크에서 다시 읽어들이어야 합니다.

장애 조치 속도를 측정하는 한 방법은 응용 프로그램이 일반 시스템에서 재부팅 후 얼마나 빨리 시작되는지 측정하는 것입니다. 응용 프로그램이 즉시 시작됩니까? 또는 응용 프로그램이 여러 단계를 거쳐야만 최종 사용자가 응용 프로그램에 연결할 수 있습니까? 이상적인 경우라면 응용 프로그램은 메모리 내의 데이터 구조나 테이블을 다시 초기화하지 않고 신속하게 시작되어야 합니다.

성능을 높이려면 데이터를 디스크가 아닌 메모리에 유지해야 합니다. 하지만 데이터를 디스크에 유지할 때 저하되는 성능 문제만큼 메모리에 데이터를 유지할 때 발생할 수도 있는 데이터 손실 가능성도 고려해야 합니다.

공유 디스크에서 메모리로 가져와서 읽기 전용으로 사용되는 데이터는 손실에 대해 걱정할 필요가 없습니다.

로그 크기 최소화

일부 데이터베이스는 온라인 성능을 향상하기 위해 로그를 메모리에 버퍼링하기도 합니다. 물론, 장애가 발생하면 진행 중인 트랜잭션은 손실됩니다. 그러나 이 메모리 상에 있는 로그 크기를 최소화하면 장애 발생 시 완료된 트랜잭션 데이터의 손실을 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

로컬 데이터의 필요성을 최소화하십시오. 3계층 모델의 클라이언트/서버 환경에서 중간 층에는 데이터가 없어도 됩니다.(즉, 클라이언트에 특정하거나 수정해야 하는 로컬 데이터가 없습니다.) “응용 프로그램 서버” 층은 더 높은 수준의 가용성, 로드 균형 조정, 장애 조치를 제공할 수 있습니다. 하지만 이 경우 모든 데이터가 클라이언트(1계층)나 데이터베이스 서버(3계층)에 저장되어야 합니다.

재시작 가능한 트랜잭션 사용

서버에 장애가 발생하여 응용 프로그램이 다른 시스템에서 시작되었을 때 클라이언트가 트랜잭션을 다시 입력하거나 처음부터 다시 시작하지 않도록 하기 위해 트랜잭션은 재시작할 수 있어야 합니다. 즉, 트랜잭션 도중에 장애가 발생하더라도 처음부터 다시 시작하지 않도록 해야 합니다. 이러한 기능은 응용 프로그램을 더욱 튼튼하게 만들며 장애 조치 시 사용자에게 피해를 덜 주게 됩니다.

일반적인 예로 인쇄 작업을 들 수 있습니다. 프린터 응용 프로그램은 일반적으로 작업 스케줄러를 사용합니다. 스케줄러는 한 작업이 완료되면 다음 작업으로 진행합니다. 예 들 들어, 만약 긴 작업(예를 들어, 3시간 동안 급여 명세서를 인쇄하는 경우) 도중에 시스템이 중지된 후 다시 정상으로 돌아왔을 때의 결과를 가정할 수 있습니다. 가능한 결과는 작업을 처음부터 다시 시작해서 명세표를 모두 다시 인쇄하거나, 작업이 중단된 부분부터 다시 시작하거나 또는 작업 스케줄러가 작업이 완료된 것으로 간주하고 남은 시간의 인쇄 작업을 수행하지 않는 방법이 있을 것입니다. 고가용성 환경에서 이 경우 올바른 처리 방법은 중단된 부분에서 인쇄를 다시 시작하여 모두가 정확하게 급여 명세표를 받을 수 있도록 하는 것입니다.

또 다른 예로 신입 사원에 대한 데이터를 입력하는 응용 프로그램을 들 수 있습니다. 이 응용 프로그램은 고유한 직원 번호를 요구하는데, 신입 사원의 이름과 번호를 입력하는 도중에 장애가 발생했다고 가정합니다. 이 경우 장애가 발생하기 전에 직원 번호를 입력했기 때문에 응용 프로그램에 이 데이터를 다시 입력할 필요가 없을까요? 또는 부분 입력된 정보를 먼저 삭제해야 할까요? 고가용성 환경에서는 이 경우 응용 프로그램에 해당 항목을 쉽게 다시 입력하거나 다음 데이터 항목에서 입력을 계속할 수 있도록 합니다.

검사점 사용

복잡한 트랜잭션에는 검사점을 사용하도록 응용 프로그램을 설계하십시오. 사용자 관점에서 하나로 보이는 트랜잭션은 여러 개의 실제 데이터베이스 트랜잭션으로 나뉠 수 있습니다. 이 설명이 재시작 가능한 트랜잭션과 관련되기는 하지만, 여기에서는 시스템 장애로 중단된 트랜잭션을 장애 조치 후 완료할 수 있도록 진행 상황을 클라이언트에서 로컬로 기록하는 방법만 설명합니다.

예를 들어, PI 값을 계산하는 응용 프로그램을 가정합니다. 원래 시스템에서 이 응용 프로그램은 소숫점 1000자리까지 계산했는데 디스크에 아직 아무 것도 기록하지 않았습니다. 이때 이 노드에서 장애가 발생했습니다. 응용 프로그램은 두 번째 노드에서 재시작되지만, 응용 프로그램은 처음부터 다시 시작해야 합니다. 따라서 응용 프로그램은 소수점 1000자리까지 다시 계산해야 합니다. 하지만 만약 응용 프로그램이 주기적으로 디스크에 소수점을 기록했다면 응용 프로그램은 중단된 부분에서 다시 시작할 수 있을 것입니다.

검사점 빈도와 성능의 균형

검사점 빈도와 성능의 균형을 맞추는 것이 중요합니다. 디스크의 검사점 빈도와 성능은 반비례합니다. 검사점 빈도가 너무 많으면 응용 프로그램이 느려지며, 검사점이 충분하지 않으면 장애 조치 후 응용 프로그램이 현재 상태로 돌아오는 데 시간이 더 오래 걸립니다. 이상적인 방법은 최종 사용자가 검사점 빈도를 결정할 수 있게 하는 것입니다. 응용 프로그램은 사용자가 정의할 수 있는 매개 변수를 제공하여 직접 검사점 빈도를 조절할 수 있게 해야 합니다.

다중 서버를 위한 설계

여러 활성화된 서버를 사용하는 경우 분산된 서비스가 클라이언트에게 보다 직접적인 서비스를 제공할 수 있습니다. 그러나 이렇게 하려면 클라이언트가 복수의 서버를 인식하여 서버에 우선 순위를 매길 수 있어야 합니다. 또한 장애가 있는 서버의 데이터 또는 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 단순히 기본 시스템의 작업을 이어받으면 됩니다. 이렇게 하면 응용 프로그램을 보다 빠르게 시작할 수 있습니다. 이러한 구조를 설계하는 방법은 여러 가지가 있으며, 이것과 관련하여 알아야 할 많은 내용들이 있습니다. 여기에서는 자세히 설명하지 않고 몇 가지 예제를 들어 설명합니다.

가장 단순한 방법은 두 응용 프로그램을 마스터/슬레이브 관계로 실행하는 것입니다. 여기서 슬레이브는 단순히 마스터에 대한 활성 대기 응용 프로그램입니다. 마스터에 장애가 발생하면 두 번째 시스템 역시 슬레이브는 데이터의 최종 상태(즉, 데이터 복구 가능 여부 등)를 알아야 합니다. 그러나 응용 프로그램을 로드하여 처음 시작하는 시간은 절약됩니다.

또 다른 방법은 두 응용 프로그램을 모두 활성화하는 것입니다. 예를 들어, 데이터베이스에 데이터를 제공하는 두 개의 응용 프로그램 서버를 가정합니다. 클라이언트의 절반은 첫 번째 응용 프로그램 서버에, 나머지 절반은 두 번째 응용 프로그램 서버에 연결되어 있습니다. 한 서버에서 장애가 발생하면 모든 클라이언트가 남은 응용 프로그램 서버에 연결하게 됩니다.

복제된 데이터 지점을 위한 설계

데이터 지점 복제는 신속한 장애 조치 및 재해 복구 모두에 도움이 됩니다. 데이터를 복제하면 데이터 디스크는 시스템 사이에서 공유되지 **않습니다**. 따라서 데이터 복구의 필요성이 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 데이터베이스가 클라이언트 응용 프로그램에 직접적인 데이터 복제를 제공합니다. 표준 데이터베이스를 사용하도록 응용 프로그램을 설계하면 최종 사용자가 데이터 복제의 사용 여부를 결정할 수 있게 됩니다.

다중 시스템에서 실행 가능한 응용 프로그램 설계

응용 프로그램이 백업 노드로 장애 조치될 수 있으면 다른 시스템에서 어떻게 이 작업을 수행할 수 있습니까?

앞에서 응용 프로그램을 자동으로 재시작하는 방법에 대하여 설명했습니다. 이 절에서는 응용 프로그램을 복수의 시스템에서 실행할 수 있는 방법에 대해서 설명합니다. 다음과 같은 내용을 다룹니다.

- 노드에 특정한 정보 최소화
- 응용 프로그램에 고유한 이름 지정
- Uname (2) 사용 시 주의점
- 고정 포트로 바인딩
- 변동 가능 IP 주소로 바인딩
- 응용 프로그램마다 고유한 볼륨 그룹 부여
- SNA 응용 프로그램을 위한 복수의 목적지 사용
- 파일 잠금 최소화

노드에 특정한 정보 최소화

일반적으로 새 시스템을 설치하면 모든 활성 네트워크 인터페이스에 IP 주소를 할당해야 합니다. 이 IP 주소는 항상 노드에 지정되며 **고정(Stationary) IP** 주소라고 합니다.

고가용성 응용 프로그램을 포함하는 패키지를 사용하면 응용 프로그램에 할당할 추가적인 IP 주소가 필요하게 됩니다. 이것은 **변동 가능한(relocatable) 응용 프로그램 IP** 주소라고 합니다. MC/ServiceGuard의 네트워크 감지 장치는 이 변동 가능한 응용 프로그램 IP 주소가 위치하는 서버넷에 액세스하는 노드를 모니터링합니다. MC/ServiceGuard에서 패키지가 구성되면 관련된 서버넷 주소가 패키지 의존 주소로 지정되며 패키지가 실행될 수 있는 노드의 목록이 제공됩니다. 패키지를 원격 노드로 장애 조치할 때는 대상 노드에서 서버넷이 이미 활성화되어 있어야 합니다.

고가용성 클러스터 응용 프로그램 설계 다중 시스템에서 실행 가능한 응용 프로그램 설계

모든 응용 프로그램 또는 패키지에는 고유한 이름과 함께 변동 가능한 IP 주소를 부여해야 합니다. 이렇게 하여 응용 프로그램과 응용 프로그램이 실행되는 시스템이 분리되며, 따라서 응용 프로그램이 어떤 시스템에서 실행되고 있는지 알아야 할 필요가 없어집니다. 또한 로드 균형 조정 등의 목적으로 응용 프로그램을 클러스터에 있는 다른 시스템으로 이동하는 작업도 더 쉬워집니다. 만약 두 응용 프로그램이 한 IP 주소를 공유한다면 둘을 함께 이동해야 하지만 서로 다른 이름과 주소를 사용하면 따로 이동할 수 있습니다.

외부에서 클러스터에 액세스할 때 클라이언트는 응용 프로그램을 참조하는 방법을 알아야 합니다. 한 가지 방법은 클라이언트에게 응용 프로그램과 관련된 변동 가능 IP 주소를 알려주는 것입니다. 다른 방법은 응용 프로그램의 이름을 호스트 이름처럼 다루어서 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 이 IP주소도 함께 이동하고, 클라이언트는 응용 프로그램의 현재 위치를 몰라도 응용 프로그램을 사용할 수 있게 됩니다. 모든 네트워크 인터페이스에는 고정 IP 주소가 할당되어야 한다는 점에 주의하십시오. 네트워크 장애가 발생했을 때 이 IP 주소는 원격 시스템으로 이동하지 **않습니다**.

충분한 IP 주소 획득

시스템 자체에 할당되는 고정 IP와는 별도로 각 응용 프로그램은 변동 가능 IP 주소를 할당 받습니다. 따라서 한 시스템에는 시스템 IP 주소 하나와 시스템에서 실행되는 각 응용 프로그램에 하나씩, 여러 개의 IP 주소가 있을 수 있습니다. 그러므로 주어진 서브넷 범위의 IP 주소는 고가용성 기능이 없는 시스템보다 빨리 소비됩니다. 이 경우 추가 IP 주소가 필요할 수도 있습니다.

하나의 네트워크 인터페이스에 여러 IP 주소를 사용하려면 이 주소들이 모두 동일한 서브넷에 위치해야 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. 특정 응용 프로그램을 실행할 때 현재 실행 중인 인스턴스를 보여주는 매개 변수를 사용할 필요가 있습니다. 이렇게 하면 정상적인 상황에서 다수의 시스템으로 사용자를 분산할 수 있을 뿐만 아니라 한 시스템에서 장애가 발생하더라도 모든 사용자의 연결을 유지할 수 있습니다.

SPU ID 또는 MAC 주소 사용 최소화

SPU ID나 MAC 주소(링크 수준의 주소)에 의존하지 않도록 응용 프로그램을 설계하십시오. SPU ID는 비활성 메모리에 저장되는 하드웨어의 고유 ID로서 변경이 불가능합니다.

MAC 주소(LANIC ID라고도 함)는 LAN 하드웨어와 관련된 링크에 특정한 주소입니다. 그런데 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 ID 또는 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. 이것은 MC/ServiceGuard에서 더 이상 허용되지 않습니다.

다음과 같이 MAC 주소를 사용해야 하는 이유가 몇 가지 있습니다.

- 라우터처럼 송신측과 수신측 사이에 위치하는 기존의 네트워크 장치들은 MAC 주소와 IP 주소 쌍을 사용하여 수동으로 프로그램해야 합니다. 이 문제의 해결 방법은 장애 조치 시 MAC 주소도 IP 주소와 함께 이동하는 것입니다.
- 시스템과 관련된 시간 제한으로 인해 네트워크 장치 캐시가 갱신되는 데 최대 20분까지 지연될 수 있습니다. 현재 버전의 HA 소프트웨어에서 이 작업은 이전 IP 주소의 새로운 ARP 해석을 새 MAC 주소와 함께 브로드캐스팅하여 처리됩니다.

응용 프로그램에 고유한 이름 지정

각 응용 프로그램에는 고유한 이름을 지정해야 합니다. 그리고 이 이름은 다음에 설명하는 것처럼 `gethostbyname()`의 입력으로 사용할 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 `telnet`과 같은 BSD 소켓 응용 프로그램에 유용합니다. 이런 응용 프로그램들은 이름을 IP 주소로 먼저 매핑한 다음에 연결할 수 있습니다. 여기에서 몇 가지 주의해야 할 사항이 있습니다.

응용 프로그램은 정식 호스트 이름 또는 IP 주소를 참조하면 **안됩니다**. 정식 호스트 이름과 호스트 이름의 해당 IP 주소는 주 LAN 카드와 이 카드의 고정 IP 주소를 참조합니다. 따라서 호스트 이름이나 주 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 확인하기 위한 한 방법은 응용 프로그램에서 `gethostname(2)` 로의 호출을 검사하는 것입니다. 응용 프로그램이 이동할 때 응답이 변할 수 있기 때문에 HA 서비스에서 `gethostname()` 를 사용할 때는 주의해야 합니다. 같은 이유에서 `gethostbyname(2)` 로의 호출을 위한 이름을 확인하기 위해 `gethostname()` 를 사용하는 응용 프로그램도 피해야 합니다. 또한, `gethostbyaddr()` 를 고정 IP 주소로 호출하면 시간이 지나면서 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 서비스에서 서로 다른 응용 프로그램 이름을 매핑하는 것이 가장 바람직합니다. 이렇게 하면 각각의 응용 프로그램과 응용 프로그램의 IP 주소가 다른 응용 프로그램에 영향을 미치지 않으면서 다른 노드로 이동할 수 있습니다. DNS의 호스트 이름에는 오직 고정 IP 주소만 연결되어야 합니다.

uname(2) 사용 시 주의점

앞에서 호스트 이름과 관련된 내용 중에 응용 프로그램에서 정식 시스템 이름을 보여주는 `uname(2)` 를 사용하는 방법에 대해 설명했습니다. 시스템 이름은 시스템의 LAN 카드 수에 상관없이 주어진 시스템에 대해 고유합니다. 일반적으로 `uname` 과 `hostname` 은 동일하지만 반드시 같아야 하는 것은 아닙니다. 일부 응용 프로그램은 보안 상의 이유로 시스템에 연결한 다음 `uname(2)` 를 호출하여 올바른 시스템에 연결했는지 검사하기도 합니다. HA 환경에서는 서비스가 한 시스템에서 다른 시스템으로 이동하지만 `uname` 과 `hostname` 은 이동하지 않으므로 이러한 응용 프로그램은 적합하지 않습니다. 응용 프로그램은 자신이 실행되고 있는 위치를 파악하는 다른 수단을 찾아야 합니다. 예를 들어, 응용 프로그램은 구성 파일에서 제공되는 호스트 이름 목록을 검사할 수 있습니다.

고정 포트로 바인딩

소켓을 바인딩할 때 포트 주소를 지정하거나 임의의 포트가 동적으로 할당되게 할 수 있습니다. 임의의 포트로 바인딩할 때의 한 가지 문제는 나중에 응용 프로그램이 다른 클러스터 노드에서 재시작되었을 때 다른 포트가 할당될 수 있다는 점입니다. 이것은 응용 프로그램에 액세스하는 클라이언트를 혼란스럽게 합니다.

권장되는 방법은 포트 번호를 동적으로 할당하지 말고 응용 프로그램이 실행될 모든 노

드에서 동일한 고정 포트를 사용하는 것입니다. 그러면 응용 프로그램은 현재 어떤 노드에서 실행되고 있는지에 관계 없이 항상 같은 포트 번호를 제공할 것입니다. 응용 프로그램 포트 할당 정보는 관리와 다른 사람이 동일한 포트를 선택하지 않도록 하기 위해 /etc/services에 기록해야 합니다.

변동 가능 IP 주소로 바인딩

소켓이 바인딩될 때 포트 번호와 함께 IP 주소가 지정됩니다. 이 주소는 통신에 사용할 IP 주소를 나타내며 응용 프로그램이 클라이언트와 통신하는 데 사용할 인터페이스를 제한할 수 있게 합니다. 응용 프로그램은 어떤 인터페이스로도 메시지가 도착할 수 있음을 나타내는 INADDR_ANY로 바인딩할 수 있습니다.

네트워크 응용 프로그램은 고정 IP 주소, 변동 가능 IP 주소 또는 INADDR_ANY로 바인딩할 수 있습니다. 고정 IP 주소를 지정하면 응용 프로그램이 다른 노드에서 재시작될 때 고정 IP 주소가 새 시스템으로 이동하지 않기 때문에 응용 프로그램이 작동하지 않을 수 있습니다. 응용 프로그램을 변동 가능 IP 주소로 바인딩하면 다른 시스템으로 이동될 때 응용 프로그램이 올바르게 작동합니다.

많은 서버 형식의 응용 프로그램은 INADDR_ANY로 바인딩하기 때문에 어떤 인터페이스로도 요청을 받을 수 있습니다. 따라서 클라이언트는 고정 IP 주소 또는 변동 가능 IP 주소로 통신할 수 있습니다. 그러나 이 경우 네트워크 코드는 응답할 IP 주소를 찾지 못하기 때문에 항상 고정 IP 주소를 선택합니다.

TCP 스트림 소켓의 경우 이것은 연결 기반 프로토콜이기 때문에 프로토콜 스택의 TCP 수준에서 이 문제를 해결합니다. 클라이언트에서 TCP는 고정 IP를 무시하고 클라이언트가 원래 사용하던 이전에 바인딩된 변동 가능 IP 주소를 계속 사용합니다.

하지만 UDP 데이터그램 소켓에는 문제가 있습니다. 클라이언트는 변동 가능 IP 주소를 사용하는 여러 서버에 연결하고 서버 응답 메시지에 있는 송신자 IP 주소를 기준으로 응답을 구분합니다. 그러나 이 응답의 송신자 IP 주소는 변동 가능한 응용 프로그램 IP 주소가 아니라 고정 IP 주소가 될 것입니다. 그러므로 수신을 위해 UDP 소켓을 구성할 때 응용 프로그램은 INADDR_ANY가 아니라 항상 올바른 변동 가능한 응용 프로그램 IP 주소를 사용하여 bind(2)를 호출해야 합니다.

위에서 권장한 방식으로 응용 프로그램을 수정할 수 없는 경우, 이 문제를 해결할 수 있는 방법은 LAN 카드에 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 재배치할 수 있도록 각각의 백업 노드에 사용하지 않는 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 시스템의 아웃바운드 LAN 인터페이스의 고정 IP 주소를 사용하여 보내질 것입니다. 클라이언트는 accept(2) 호출로부터 이 IP 주소를 받으며, 이로 인해 클라이언트 소프트웨어가 혼란에 빠져 정상적으로 작동하지 않을 수 있습니다.

응용 프로그램마다 고유한 볼륨 그룹 부여

데이터를 사용하는 각각의 응용 프로그램마다 별도의 볼륨 그룹을 사용하십시오. 디스크를 사용하지 않는 응용 프로그램에는 별도의 볼륨 그룹을 할당하지 않아도 됩니다. 볼륨 그룹(디스크의 그룹)은 노드 사이에서 이동할 수 있는 저장 장치의 단위입니다. 각각의 응용 프로그램을 자신의 볼륨 그룹으로 제한하면, 즉 두 응용 프로그램이 동일 디스크 드라이브 집합을 공유하지 않으면 로드 균형 조정에 유연성이 높아집니다. 만약 두 응용 프로그램이 동일한 볼륨 그룹을 사용하여 데이터를 저장하면 그 두 응용 프로그램을 함께 이동해야 합니다. 두 응용 프로그램의 데이터가 별도의 볼륨 그룹에 저장된다면 장애 발생 시 서로 다른 노드로 전환될 수 있습니다.

응용 프로그램의 데이터는 서로 다른 디스크 드라이브에, 그리고 가능하면 서로 다른 마운트 지점에 설정되어야 합니다. 응용 프로그램은 서로 다른 디스크 및 별도의 마운트 지점을 허용하도록 설계되어야 합니다. 가능하면 응용 프로그램은 특정 마운트 지점을 요구하지 않아야 합니다.

한 노드가 실수로 다른 노드의 응용 프로그램이 사용하는 디스크에 액세스하는 것을 방지하기 위해 HA 소프트웨어는 배타적인 액세스 메커니즘을 사용하여 한 번에 하나의 노드만 액세스하도록 해야 합니다. 이 배타적인 액세스는 볼륨 그룹 전체에 적용됩니다.

SNA 응용 프로그램을 위한 복수의 목적지 사용

SNA는 지점간 연결에 기초합니다. 즉, 시스템에는 메인프레임에서 시작된 다른 지점간 연결이 있기 때문에 서비스가 단순히 다른 시스템으로 이동할 수 없습니다. 따라서 SNA가 단순 장애 요인이 되지 않도록 노드의 백업 링크 및 다른 노드의 백업 링크가 구성되어야 합니다. SNA 링크를 위한 구성은 한 번에 오직 하나만 활성화될 수 있습니다. 따라서 다른 용도로 사용되는 백업 링크는 장애 조치 시 주 업무 목적을 위해 재편성되어야 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 연결합니까?

서버로의 연결 손실과 다른 응용 프로그램에서 시작된 오류를 특별히 구별하도록 클라이언트 응용 프로그램을 작성하는 것이 중요합니다. 응용 프로그램은 연결이 손실된 경우 특별한 조치를 취해야 합니다.

한 가지 고려할 의문점은 장애 발생 이후에 응용 프로그램이 새로 시작된 서버에 다시 연결할 수 있는 상황을 어떻게 알 수 있는지는 것입니다. 일반적인 경우에는 클라이언트가 단순히 세션을 재시작하거나 다시 로그인해야 하지만 이 방법은 수동으로 조작해야 합니다. 예를 들어, 잘 구성된 하드웨어 및 응용 프로그램 시스템은 5분 내에 장애 조치될 수 있습니다. 하지만 사용자는 장애 중 응답이 없자 2분 뒤에 포기하고 커피를 마시러 나가 28분 동안 돌아오지 않는다고 가정하면 실제 다운시간은 5분이 아니라 30분이 됩니다. 고려해야 할 요소는 재시도 회수, 재시도의 빈도 그리고 사용자에게 연결 장애 여부를 알릴 것인가 하는 점입니다.

클라이언트 재연결에 사용할 수 있는 몇 가지 전략이 있습니다.

- 장애가 발생한 서버로 계속 재연결을 시도하도록 클라이언트 설계

사용자의 재연결 작업에 의존하지 않고 클라이언트 응용 프로그램이 재연결하도록 만드십시오. 5분 안에 서버가 백업되어 실행되고 클라이언트가 계속 재시도하면 5분 후에 클라이언트 응용 프로그램은 서버와 재연결하여 트랜잭션을 다시 시작하거나 계속할 것입니다. 사용자의 개입은 필요 없습니다.

- 다른 서버에 재연결하도록 클라이언트 설계

여러 활성 서버를 포함하도록 설계된 서버를 사용하면 클라이언트는 두 번째 서버에 연결할 수 있기 때문에 다운시간이 길지 않습니다.

이 설계의 문제는 클라이언트가 언제 두 번째 서버로 전환해야 하는지를 알 수 있는지는 것입니다. 클라이언트가 첫 번째 서버를 포기하고 두 번째 서버로 이동하기까지 얼마나 오래 재시도해야 합니까? 이 문제에 대한 정답은 없습니다. 이 시간은 서버 응용 프로그램의 설계에 따라 달라집니다. 서버 응용 프로그램이 장애 발생 후 같은 노드에서 재시작될 수 있다면(다음의 “응용 프로그램 장애 처리” 참조) 서버가 로컬에서 재시작할 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

Tuxedo, DCE/Encina 같이 서버와 클라이언트 사이에서 인터페이스를 제공하는 트랜잭션 처리 모니터를 사용하십시오. 트랜잭션 처리 모니터(TPM)는 더 높은 가용성의 응용 프로그램을 만드는 데 유용하게 사용할 수 있습니다. 트랜잭션은 대기열을 사용하기 때문에 클라이언트는 서버의 장애를 감지하지 못합니다. 많은 TPM은 선택적으로 대체 서버로의 자동 경로 재설정 또는 트랜잭션의 자동 재시도를 제공합니다. TPM에만 있는 기능은 아니지만 TPM은 또한 트랜잭션의 신뢰성 있는 수행을 보장합니다. 서버가 다시 온라인이 되면 트랜잭션 모니터는 새 서버로 재연결하여 트랜잭션 라우팅을 계속합니다.

- 요청을 대기열에 기록

TPM 사용하는 대신 서버를 사용할 수 없을 때 요청을 대기열에 기록할 수 있습니다. 서버를 사용할 수 없을 그 사실을 사용자에게 알리는 대신 사용자의 요청은 대기열에 기록되어 서버를 다시 사용할 수 있을 때 전송됩니다. 메시지 대기열 소프트웨어는 단순히 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 한 부분 또는 전체에 장애를 발생하면 어떤 일이 일어납니까?

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 프로그램이 장애를 일으키거나 시스템 자원의 문제(스왑/메모리 공간 부족 등)로 인해 응용 프로그램이 정지할 수 있습니다. 이 절에서는 이런 종류의 장애를 복구할 수 있는 응용 프로그램 설계 방법에 대해 설명합니다.

장애를 극복할 수 있는 응용 프로그램 작성

응용 프로그램은 단일 구성 요소로서 장애를 극복할 수 있어야 합니다. 많은 응용 프로그램들은 한 노드에서 여러 프로세스를 실행합니다. 만약 한 프로세스에 장애가 발생한다면 다른 프로세스는 어떻게 됩니까? 다른 프로세스도 같이 장애를 일으킵니까? 장애를 일으킨 프로세스가 응용 프로그램의 나머지 부분에 영향을 미치지 않고 같은 노드에서 다시 시작할 수 있습니까?

이상적인 처리 방법은 한 프로세스가 장애를 일으켰을 때 다른 프로세스들이 장애를 일으킨 구성 요소가 다시 온라인이 될 때까지 일정 시간 대기할 수 있게 하는 것입니다. 이것은 장애를 일으킨 구성 요소가 같은 시스템에 있거나 원격 시스템에 있는 경우 모두 해당됩니다. 장애를 일으킨 구성 요소가 같은 시스템에서 자동으로 재시작되어 대기 상태인 작업에 다시 참여하여 작업을 계속할 수 있습니다. 이런 종류의 장애는 몇 초 내에 감지되어 재시작될 수 있으며, 최종 사용자는 장애가 발생한 사실을 전혀 알지 못할 것입니다.

다른 방법으로 한 구성 요소가 장애를 일으켰을 때 다른 구성 요소들을 정상 종료하는 방법이 있습니다. 만약 데이터베이스 SQL 서버에 장애가 발생했을 때 데이터베이스는 정상 종료되면 데이터베이스 복구가 필요 없을 것입니다.

더 나쁜 상황은 한 구성 요소의 장애가 전체 시스템의 장애를 유발하는 경우입니다. 만약 한 구성 요소가 장애를 일으켰을 때 다른 모든 구성 요소가 재시작되어야 한다면 다운시간이 길어질 것입니다.

응용 프로그램 모니터링 기능

응용 프로그램을 포함하여 시스템의 모든 구성 요소는 이상 유무를 모니터링할 수 있어야 합니다. 모니터는 명령을 표시하는 정도로 단순하거나 SQL 질의 수준으로 복잡할 수 있습니다. 응용 프로그램이 올바르게 작동하는 것을 보장할 수 있는 수단이 있어야 합니다. 응용 프로그램에 장애가 발생했을 때 그 사실이 자동으로 감지되지 않는다면 사용자가 장애 원인을 파악하고 복구하는 데 여러 시간이 걸릴 수 있습니다.

계획된 다운시간 최소화

계획된 다운시간(계획되지 않은 다운시간과 반대)에는 정기적인 백업, 시스템 OS 업그레이드, 하드웨어 교체 등과 같은 계획된 중지입니다. 계획된 다운시간을 위해 응용 프로그램 설계자는 다음 사항을 고려해야 합니다.

- **응용 프로그램 업그레이드/패치에 요구되는 시간 단축**

관리자가 다운시간을 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 재편성 기능**

응용 프로그램을 중지하지 않고 응용 프로그램이 사용하는 구성 정보를 변경할 수 있습니까?

- **유지 보수 작업 문서화**

관리자가 유지 보수 방법을 알고 있습니까?

고가용성 시스템에 대한 논의에서 일반적인 주 관심사는 계획되지 않은 다운시간에 대한 것입니다. 하지만 계획된 다운시간의 경우라도 시스템을 새로운 버전의 소프트웨어로 업그레이드하는 데 2주가 걸린다면 상당한 불편할 것입니다.

다음 절에서는 여러 가지 종류의 계획된 다운시간을 처리하는 방법을 설명합니다.

응용 프로그램 업그레이드/패치에 요구되는 시간 단축

매년 응용 프로그램의 새 버전이 출시됩니다. 최종 사용자가 새 버전으로 업그레이드하는 데 얼마나 많은 시간이 필요합니까? 이 질문의 답은 사용자가 자신의 응용 프로그램을 업그레이드할 때 겪어야 하는 계획된 다운시간의 양입니다. 다음은 이 시간을 축소시키는 방법에 관한 지침입니다.

롤링 업그레이드 구현

클라이언트/서버 환경에서 “롤링 업그레이드”를 구현하십시오. 많은 요소로 구성된 시스템인 경우, 전형적인 업그레이드 방식은 전체 시스템을 종료하고 모든 노드를 새로운 버전의 소프트웨어로 업그레이드한 다음, 모든 노드에서 응용 프로그램을 다시 시작하는 것입니다. 대규모 시스템에서 이런 방식은 아주 긴 다운시간을 초래할 수 있습니다. 이에 대한 대안은 롤링 업그레이드를 구현하는 것입니다. 롤링 업그레이드는 한 번에 한

구성 요소만 업그레이드 하는 단계별 접근 방식으로 새로운 소프트웨어를 설치합니다. 예를 들어, 월요일에 데이터베이스 서버를 업그레이드하면서 15분의 다운시간이 발생합니다. 그런 다음 화요일에는 두 노드의 응용 프로그램 서버를 업그레이드하지만 나머지 노드의 응용 프로그램 서버는 그대로 두었기 때문에 다운시간이 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 다운시간을 최소화할 수 있게 됩니다.

이 방법의 문제는 응용 프로그램 소프트웨어는 반드시 이 소프트웨어의 다른 버전과 함께 사용해야 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 작업은 상당한 시간이 걸릴 수 있습니다. 또한 롤링 업그레이드도 불가능할 것입니다. 예를 들어, 데이터베이스가 첫 번째 노드에서 실행 중이라면 이상적인 경우는 두 번째 노드를 새 버전의 데이터베이스로 업그레이드하는 것입니다. 업그레이드가 완료되면 약간의 다운시간을 계획하여 데이터베이스 서버를 첫 번째 노드에서 새로 업그레이드된 두 번째 노드로 이동합니다. 그런 다음 데이터베이스 서버를 재시작하면 첫 번째 노드는 대기 상태가 되어 업그레이드가 가능하게 됩니다. 하지만 새 버전의 데이터베이스가 다른 데이터베이스 레이아웃을 필요로 하면 **이전의** 데이터는 새로 업데이트된 데이터베이스에서 사용할 수 없습니다. 데이터를 새로운 레이아웃으로 마이그레이션해야 하기 때문에 다운시간이 더 길어질 것입니다.

온라인 응용 프로그램 재편성 기능

대부분의 응용 프로그램은 응용 프로그램이 시작될 때 사용되는 약간의 구성 정보를 가지고 있습니다. 이 구성 정보를 변경하면 응용 프로그램을 중지하고 새로운 구성 파일에서 이 정보를 읽어야 하기 때문에 다운시간이 발생합니다.

이런 다운시간을 방지하려면 응용 프로그램과 상호 작용하여 온라인 상태에서 동적으로 구성을 변경할 수 있는 구성 도구를 사용해야 합니다. 이상적인 해결책은 응용 프로그램과 상호 작용하는 이러한 구성 도구를 갖추는 것입니다. 구성은 온라인 상태에서 변경되고 최종 사용자는 방해받지 않습니다. 이 도구는 데이터 크기를 확장하거나, 시스템에 새로운 사용자를 추가하거나, 응용 프로그램에 새로운 사용자를 추가하는 등의 모든 작업을 온라인에서 처리할 수 있어야 하며 응용 프로그램 시스템에 관하여 관리자가 필요로 하는 모든 작업이 온라인 상태에서 수행될 수 있어야 합니다.

유지 보수 작업 문서화

표준 절차가 중요합니다. 응용 프로그램 설계자는 고가용성 환경과 일반 환경 모두에 대해 작업을 일반화할 수 있도록 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 프로그램의 설명서에는 일반적인 유지 보수에서 고가용성을 제공할 수 있는 방법을 다루어야 합니다.

D

HA(고가용성) 응용 프로그램과 MC/ServiceGuard 통합

다음은 응용 프로그램을 MC/ServiceGuard 환경과 통합하기 위해 수행해야 하는 단계를 요약 설명합니다.

1. 클러스터 구성 및 패키지 구성의 설명과 부록 C, “고가용성 클러스터 응용 프로그램 설계” 포함하여 이 책의 나머지 부분을 읽습니다.
2. 정상 작동 시 클러스터의 동작을 정의합니다.
 - 정상 작동 시 클러스터의 상태는 어떻게 됩니까?
 - 일반적으로 사용할 표준 구성은 무엇입니까?(사용자들의 요구 사항에 관한 자료가 있습니까?)
 - 데이터베이스, 응용 프로그램 서버 등과 같은 기능을 별도의 시스템으로 분리시킬 수 있습니까? 또는 모든 기능이 한 시스템에서 실행되고 있습니까?
3. 장애 조치 시 클러스터의 동작을 정의하십시오.
 - 모두 함께 대체 노드로 장애 조치됩니까?
 - 같은 노드로 장애 조치되는 응용 프로그램을 분리시킬 수 있습니까?
 - 응용 프로그램에 MC/ServiceGuard에서 제공하는 기능 이외의 다른 고가용성 기능이 이미 있습니까?
4. 문제가 있는 영역을 확인하십시오.
 - 응용 프로그램이 시스템 재부팅 또는 중지를 처리하기 위해 하는 일은 무엇입니까?
 - 다른 시스템으로 장애 조치되는 데 문제가 될 수 있는 `uname()`, `gethostname()`, `SPU_ID`, `MAC` 주소 등과 같은 시스템에 특정한 정보를 응용 프로그램이 사용하고 있습니까?

HA(고가용성) 응용 프로그램을 통합하기 위한 검사 목록

이 절은 HA(고가용성) 응용 프로그램을 단일 시스템 및 다중 시스템에 통합하기 위한 검사 목록을 설명합니다.

단일 시스템에서 기본 응용 프로그램의 동작 정의

1. 독립형 시스템에서 응용 프로그램의 기본 동작을 정의합니다.

- 응용 프로그램, 데이터베이스, 기타 필요한 자원을 하나의 시스템에 설치합니다. 이 작업을 수행할 때는 MC/ServiceGuard 규칙에 따라야 합니다.
 - 모든 공유 데이터는 별도의 외부 볼륨 그룹에 설치합니다.
 - 필요하면 JFS 파일 시스템을 사용합니다.
- 몇 가지 일반적인 테스트를 수행하여 응용 프로그램이 올바르게 실행되고 있는지 확인합니다. 이 테스트는 나중에 MC/ServiceGuard를 테스트할 때 사용할 수 있습니다. 가능하면 클라이언트를 통해 응용 프로그램에 연결해 보십시오.
- 응용 프로그램이 설치된 독립형 시스템에 장애를 일으키고, 재부팅하고, 응용 프로그램이 어떻게 다시 시작하는지 테스트합니다. 다음 사항을 관찰하십시오.
 - 수동 조작이 필요한 단계가 있습니까? 있으면 기록합니다.
 - rc 스크립트에서 모든 것을 시작할 수 있습니까?
- 키보드로 아무 것도 입력하지 않고 모든 것을 시작할 수 있는 간단한 스크립트를 작성해 봅니다. 관리자가 키보드로 입력해야 하는 작업을 찾아 이 작업을 스크립트에 포함하십시오.
- 응용 프로그램을 종료하는 간단한 스크립트를 작성해 봅니다. 다시 관리자가 키보드로 입력해야 하는 작업을 찾아 이 작업을 스크립트에 포함하십시오.

다중 시스템에 HA(고가용성) 응용 프로그램 통합

1. 두 번째 시스템에 응용 프로그램을 설치합니다.

- 두 번째 시스템에 LVM 하부 구조를 만듭니다.
- 시스템에 해당 사용자를 추가합니다.

- 필요한 실행 파일들을 설치합니다.
 - 첫 번째 시스템에서 응용 프로그램이 실행되고 있지 않은 상태에서 두 번째 시스템에서 응용 프로그램을 시작합니다. 앞 단계에서 작성했던 스크립트를 사용하십시오. 수동 조작과 다른 점이 있습니까? 응용 프로그램이 실행됩니까?
 - 두 번째 시스템에서 응용 프로그램을 실행할 수 있을 때까지 이 과정을 반복합니다.
2. MC/ServiceGuard 클러스터를 구성합니다.
- 클러스터 구성을 만듭니다.
 - 패키지를 만듭니다.
 - 패키지 스크립트를 만듭니다.
 - 앞 단계에서 작성했던 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 부하를 전달합니다.
 - 첫 번째 노드에서 패키지를 중지하여 두 번째 노드로 이동시킵니다.

```
# cmhaltpkg pkg1  
# cmrunpkg -n node2 pkg1  
# cmmodpkg -e pkg1
```
 - 다시 가져옵니다.

```
# 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 여러 응용 프로그램을 함께 사용하여 테스트합니다. 사용자 로드가 많을 때와 적을 때, 일괄 작업과 온라인 트랜잭션 작업 등, 서로 다른 응용 프로

HA(고가용성) 응용 프로그램과 MC/ServiceGuard 통합 HA(고가용성) 응용 프로그램을 통합하기 위한 검사 목록

그럼 상태에서 장애 조치 과정을 반복하십시오.

3. 각각의 응용 프로그램 상태마다 장애 조치에 걸린 시간을 기록합니다. 예를 들면, 클러스터 재편성에 45초, 파일 시스템에서 fsck 실행에 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 MC/ServiceGuard 소프트웨어를 업그레이드할 수 있습니다. 이 과정은 하드웨어 유지 보수나 패치 설치 등을 위해 한 시스템을 오프라인시켜야 하는 경우에도 사용할 수 있습니다. 모든 노드에서 업그레이드 과정이 완료될 때까지는 클러스터 구성 파일을 변경할 수 없으며 새로운 MC/ServiceGuard 버전의 기능을 사용할 수 없습니다.

롤링 업그레이드는 MC/ServiceGuard의 지원되는 모든 버전에서 수행할 수 있습니다. 이 방법으로 모든 이전 버전을 상위 버전으로 업그레이드할 수 있습니다. 예를 들어, HP-UX 10.10의 MC/ServiceGuard 버전 A.10.05를 HP-UX 11.00의 A.11.12로 업그레이드할 수 있습니다.

이 장에서 다음과 같은 내용을 다룹니다.

- 롤링 업그레이드의 단계
- 롤링 업그레이드의 예제
- 롤링 업그레이드의 제한 사항

롤링 업그레이드의 단계

다음 단계를 수행하십시오.

1. `cmhaltnode`를 사용하여 업그레이드할 노드를 중지합니다. 이렇게 하면 노드의 패키지 키가 대체 노드에서 시작됩니다.
2. `/etc/rc.config.d/cmcluster` 파일을 편집하여 다음과 같은 줄을 포함시킵니다.

```
AUTOSTART_CMCLD = 0
```
3. 노드를 사용할 HP-UX 버전과 MC/ServiceGuard 버전으로 업그레이드합니다. SCSI 케이블의 연결을 해제하지 않는 범위 내에서 필요에 따라(예: VERITAS 볼륨 관리자 소프트웨어) 기타 소프트웨어 또는 하드웨어를 업그레이드할 수 있습니다. 자세한 내용은 "문제점 해결" 장에서 하드웨어 유지 보수에 대한 절을 참조하십시오.
4. `/etc/rc.config.d/cmcluster` 파일을 편집하여 다음과 같은 줄을 포함시킵니다.

```
AUTOSTART_CMCLD = 1
```
5. `cmrunnode` 명령으로 업그레이드된 노드에서 클러스터를 재시작합니다.
6. 클러스터의 모든 노드에서 이 과정을 반복합니다.

주

이 과정에서 패키지가 한 노드에서 다른 노드로 이동하더라도 심각한 성능 저하가 발생하지 않도록 시스템 성능을 여유 있게 계획하십시오.

심각한 전원 문제 등으로 인해 롤링 업그레이드가 완료되기 전에 클러스터가 중지되면 최신 버전의 소프트웨어로 업그레이드된 노드에서 `cmrunc1` 명령을 입력하여 클러스터를 다시 시작할 수 있습니다.

SAM을 사용한 롤링 업그레이드 수행

HP-UX 명령 대신 SAM을 사용하여 롤링 업그레이드를 수행할 수 있습니다. 그러나, SAM은 최신 버전의 소프트웨어가 있는 노드에서 실행해야 합니다. 이전 버전의 소프트웨어가 있는 노드에서 작업을 수행하면 올바르게 작동하지 않거나 일관성이 유지되지 않습니다.

SAM을 사용하여 롤링 업그레이드를 수행하려면 SAM의 기본 메뉴에서 Clusters를 선택하고 High Availability Clusters를 선택한 다음 클러스터 관리와 패키지 관리 중에서 적합

한 관리 종류를 선택하십시오. 앞의 각 단계에서 설명한 작업을 HP-UX 명령이 아닌 SAM을 사용하여 수행하십시오.

커널 일관성 유지

롤링 업그레이드 과정에서 커널 매개 변수를 변경할 경우, 장애 조치 상황에서 동일한 패키지를 실행할 수 있는 모든 노드에서도 마찬가지로 매개 변수를 변경해야 합니다.

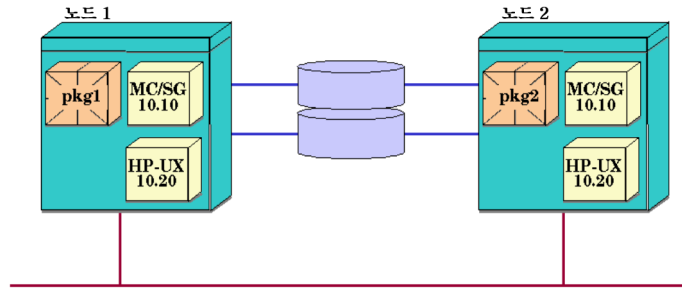
롤링 업그레이드의 예제

롤링 업그레이드를 수행하면서 노드가 실행 중인 소프트웨어의 버전을 검사하는 동안 경고 메시지가 표시될 수 있습니다. 이것은 정상적인 상황이며 문제가 되지 않습니다.

그림 E-1은 하나의 패키지를 실행하는 두 노드에서 간단한 롤링 업그레이드를 수행하는 예제를 보여줍니다. 이 그림과 이후 그림에서는 MC/ServiceGuard 10.10 및 HP-UX 10.20에서 업그레이드를 시작합니다. 이 버전에서 MC/ServiceGuard 11.13 및 HP-UX 11.00으로 롤링 업그레이드를 수행합니다. 사용자의 시스템에서는 롤링 업그레이드를 수행할 실제 버전 번호로 대체하십시오.

그림 E-1

롤링 업그레이드 전에 클러스터 실행



1 단계

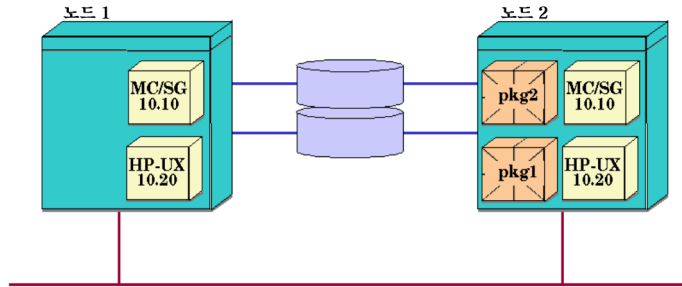
다음과 같이 첫 번째 노드를 중지하십시오.

```
# cmhaltnode -f node1
```

이 명령은 PKG1을 정상 종료하고 노드 2로 이동시킵니다. 그림 E-2와 같이 노드 1의 MC/ServiceGuard 데몬은 중지됩니다.

그림 E-2

노드 2로 이동한 패키지를 포함하는 클러스터 실행

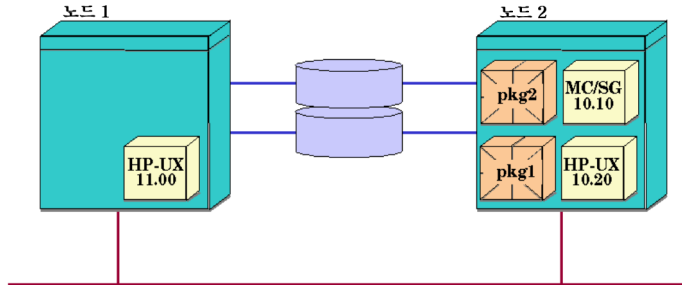


2 단계

노드 1을 상위 버전의 운영 체제(이 예제에서는 HP-UX 11.00)로 업그레이드한 다음, 그림 E-3에서처럼 상위 버전의 MC/ServiceGuard(11.13)를 설치하십시오.

그림 E-3

노드 1을 HP-UX 11.00으로 업그레이드



3 단계

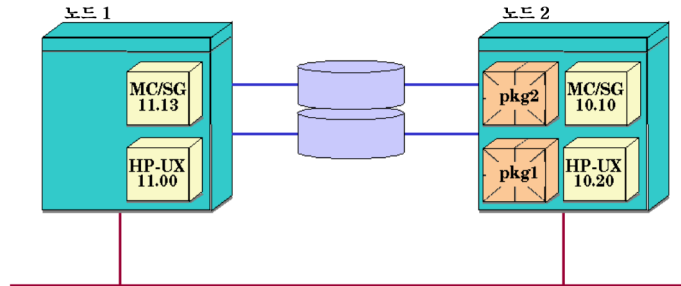
업그레이드가 완료되면 노드 1에서 다음 명령을 입력하여 클러스터를 노드 1에서 재시작하십시오.

```
# cmrunnode -n node1
```

그림 E-4에서처럼, 이때 두 노드에서 서로 다른 버전의 MC/ServiceGuard 데몬(*cmc1d*)이 실행되고 있습니다.

그림 E- 4

노드 1이 클러스터에 다시 참여



4 단계

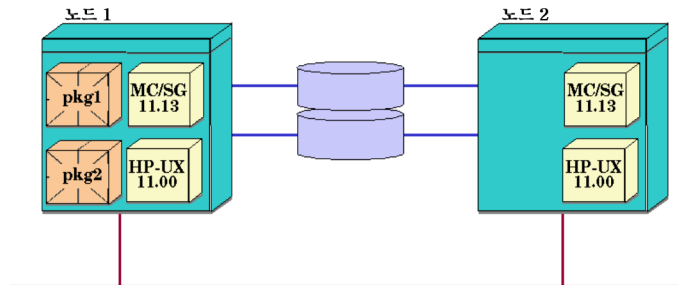
다음과 같이 노드 2에서 이 과정을 반복하십시오.

```
# cmhaltnode -f node2
```

이렇게 하면 두 패키지가 모두 노드 1로 이동합니다. 이제 노드 2를 HP-UX 11.00 및 MC/ServiceGuard 11.13으로 업그레이드하십시오.

그림 E- 5

노드 1로 이동한 패키지를 포함하는 클러스터 실행



5 단계

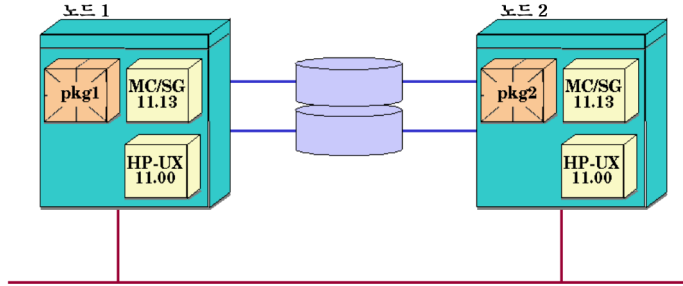
PKG2를 다시 원래 노드로 이동하십시오. 다음 명령을 사용하십시오.

```
# cmhaltpkg pkg2
# cmrunpkg -n node2 pkg2
# cmmodpkg -e pkg2
```

cmmodpkg 명령은 cmhaltpkg 명령으로 사용 해제되었던 패키지의 전환을 다시 사용 설정합니다. 그림 E-6은 최종적으로 실행 중인 클러스터를 보여줍니다.

그림 E-6

업그레이드 이후에 클러스터 실행



롤링 업그레이드의 제한 사항

롤링 업그레이드에는 다음과 같은 제한 사항이 있습니다.

- 롤링 업그레이드가 진행 중인 동안에는 최신의 소프트웨어 버전이 있는 노드에서만 MC/ServiceGuard 명령(`cmrunnode`와 `cmhaltnode`를 제외한 나머지 명령)을 실행해야 합니다. 이전 버전의 소프트웨어가 있는 노드에서 작업을 수행하면 올바른 작동하지 않거나 일관성이 유지되지 않습니다.
- 업그레이드가 완료될 때까지는 클러스터나 패키지 구성을 수정할 수 없습니다. 롤링 업그레이드 동안에는 클러스터의 네트워크 구성을 포함한 하드웨어 구성을 수정할 수 **없습니다**. 즉, 먼저 모든 노드를 새 버전으로 업그레이드한 이후에 구성 파일을 수정하고 모든 노드에 복사할 수 있습니다.
- 모든 노드가 업그레이드될 때까지 새로운 MC/ServiceGuard 버전의 기능은 사용할 수 없습니다.
- 서로 다른 MC/ServiceGuard 버전 사이에서 이진 구성 파일이 호환되지 않을 수 있습니다. 노드 사이에서 구성 파일을 수동으로 복사하지 **마십시오**.
- MC/ServiceGuard 클러스터에서 롤링 업그레이드가 진행 중인 동안 동시에 실행할 수 있는 MC/ServiceGuard의 버전은 두 개입니다.
- 클러스터가 마지막으로 시작된 이후 구성이 수정되지 않은 경우에만 롤링 업그레이드를 수행할 수 있습니다.
- 롤링 업그레이드는 클러스터 내에서 서로 다른 버전의 MC/ServiceGuard 또는 HP-UX를 함께 사용하기 위한 수단은 아닙니다. 모든 클러스터 노드를 가능한 신속히 새 버전으로 업그레이드하는 것이 좋습니다.
- 클러스터의 롤링 업그레이드가 진행 중인 동안에 노드에서 `swremove` 명령을 사용하여 MC/ServiceGuard 소프트웨어를 삭제할 수 없습니다.

F LAN에서 OTS/9000 사용

MC/ServiceGuard는 HP의 OSI Transport Services 및 관련 프로토콜 기술인 OTS/9000을 지원합니다. MC/ServiceGuard에서 OTS/9000을 구성하려면 다음 단계를 수행하십시오.

1. 모든 클러스터 노드에서 OTS/9000을 구성합니다. 자세한 내용은 OTS/9000 설명서를 참조하십시오.
2. OTS를 포함하는 각각의 LAN 카드에 대해 IP 서브넷을 구성합니다. 실제로 MC/ServiceGuard는 구성된 IP 서브넷을 통해 실제로 LANICS(LAN 인터페이스 카드)를 모니터링하기 때문에 이 과정이 필요합니다. MC/ServiceGuard에서 IP 서브넷을 지정하는 방법은 3장을 참조하십시오.

클러스터가 실행되는 동안 MC/ServiceGuard는 IP 서브넷의 장애를 모니터링하여 네트워크 장애를 감지합니다.

주

MC/ServiceGuard 환경에서 OTS/9000 프로토콜은 하트비트가 아닌 데이터 전송에만 사용할 수 있습니다. 그러나 OTS/9000이 실행되고 있는 동일한 LAN에 하트비트를 위한 TCP/IP 네트워크를 구성할 수 있습니다.

MC/ServiceGuard 내에서 OTS 모니터를 패키지에 의존하도록 구성할 수 있습니다. SAM 인터페이스를 사용하여 “Package Configuration” 하위 영역에서 “Additional Package Dependencies”를 선택하십시오.

LAN에서 OTS/9000 사용

G**계획 워크시트 양식**

이 부록은 “HA(고가용성) 클러스터 계획 및 문서화” 장에서 설명한 계획 워크시트 양식들이 있습니다. 필요한 워크시트를 복사하여 계획 과정에서 항목을 입력하는 데 사용하십시오.

하드웨어 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
_____	_____	_____	_____

_____	_____	_____	_____
-------	-------	-------	-------

_____	_____	_____	_____
-------	-------	-------	-------

=====
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
_____	_____	_____

_____	_____	_____
-------	-------	-------

_____	_____	_____
-------	-------	-------

Attach a printout of the output from `ioscan -f` and `lsssf /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 _____

Quorum 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 실제 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: _____
```

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 _____
```

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: _____

=====
Deferred Resources:

Deferred Resource Name _____

계획 워크시트 양식
패키지 제어 스크립트 워크시트

H LVM에서 VxVM 데이터 저장으로 변환

이 부록에서는 VERITAS 볼륨 관리자(VxVM) 또는 클러스터 볼륨 관리자(CVM)에서 사용할 VxVM 디스크 그룹으로 LVM 볼륨 그룹을 변환하는 방법에 대해 설명합니다. 다음과 같은 내용을 다룹니다.

- VxVM 로드
- 볼륨 그룹 변환
- VxVM의 패키지 사용자 정의
- CVM의 패키지 사용자 정의
- LVM 볼륨 그룹 제거

변환 중에 클러스터와 패키지를 관리하기 위해 취할 조치에 대해 중점적으로 설명합니다. VxVM 디스크 그룹에 대한 세부 지침은 *VERITAS Volume Manager Administrator's Guide*를 참조하십시오. 새 디스크로 시작하여 새 시스템에 기본 저장을 만들려면 5장을 참조하십시오.

클러스터 실행 중에 아래에 설명된 절차를 수행할 수 있지만 변환 중인 볼륨 그룹을 사용하는 패키지는 모두 중지해야 합니다. 클러스터 볼륨 관리자(CVM)에서 사용할 디스크 그룹에 대해서는 몇 가지 추가 단계가 제공됩니다.

VxVM 로드

데이터 변환을 시작하기 전에 VERITAS 볼륨 관리자 소프트웨어와 필요한 모든 VxVM 라이선스를 모든 클러스터 노드에 설치해야 합니다. 이 단계에서는 각 시스템을 재부팅해야 합니다. 설치 전에 클러스터에서 노드를 제거하고 설치 후에 노드를 다시 시작해야 합니다. 이 단계는 부록 E에 설명된 롤링 업그레이드 절차에서 수행됩니다.

VxVM 설치에 대한 세부 사항은 *VERITAS Volume Manager 3.1 for HP-UX Release Notes*를 참조하십시오.

볼륨 그룹 변환

다음 절차는 제공된 노트에서 실행하기 위해 구성한 패키지의 개별 볼륨 그룹을 변환하는 방법을 설명합니다. 한 패키지의 볼륨 그룹을 모두 동시에 변환하는 것이 좋습니다.

VxVM 소프트웨어와 해당 수준의 HP-UX 및 ServiceGuard가 노트에 설치되었고 이 노트를 다시 시작하여 클러스터에 다시 연결했다고 가정합니다. 또한, "루트 디스크 그룹 만들기"에 설명된 대로 노트에 rootdg를 만들었다고 가정합니다.

1. VxVM으로 변환할 볼륨 그룹을 활성화하는 패키지를 중지합니다.

```
# cmhaltpkg PackageName
```

2. 읽기 전용 모드에서 LVM 볼륨 그룹을 활성화합니다.

```
# vgchange -a r VolumeGroupName
```

3. 이 볼륨 그룹에 포함된 데이터에 가장 적합한 방법을 사용하여 볼륨 그룹의 데이터를 백업합니다. 예를 들면, Omniback처럼 백업/복원 유틸리티를 사용하거나 dd 같은 HP-UX 유틸리티를 사용합니다.

4. 볼륨 그룹 구성을 백업합니다.

```
# vgcfgbackup
```

5. 새로운 VxVM 디스크 그룹과 논리 볼륨을 정의합니다. 모든 LVM 볼륨 그룹의 VxVM 버전을 만들 수 있는 충분한 디스크가 추가로 필요합니다. LVM 구성과 동일한 일반 레이아웃을 갖는 VxVM 논리 볼륨을 만들어야 합니다. 예를 들어, LVM 미러 볼륨에는 SCSI 컨트롤러에 하나의 미러 사본이 있고 전체 볼륨을 사용할 수 없는 단일 컨트롤러 장애에 대비하여 다른 컨트롤러에 두 번째 사본이 있어야 합니다. 때로는 LVM에서 실제 볼륨 그룹을 사용하여 이렇게 구분합니다. VxVM 플렉스를 만들 때 동일한 미러링 패턴을 따라야 합니다. 다른 버스에 연결된 디스크에 다른 플렉스를 구성합니다.

VxVM 디스크 그룹을 새로운 디스크 집합에 정의하는 다른 방법으로는 vxvmconvert (1M) 유틸리티를 사용하여 기존 LVM 볼륨 그룹을 VxVM 디스크 그룹으로 변환하는 것입니다. 이 유틸리티의 제한 사항 및 주의 사항은 VERITAS *Volume Manager 3.1 Administrator's Guide*에 설명되어 있습니다. vxconvert (1M) 유틸리티를 사용할 경우 다음 단계는 건너뛰고 다음 절을 참조하십시오.

주

클러스터 잠금 디스크는 반드시 LVM 볼륨 그룹 및 물리 볼륨에서 구성되어야 합니다. VxVM으로 이동시키려는 데이터를 포함하는 잠금 볼륨 그룹이 있으면 해당 작업을 수행할 수 있습니다. 그러나 잠금 디스크에는 여전히 LVM 헤더가 필요하므로 `vxvmconvert`를 사용하지 마십시오.

-
6. 데이터를 새로운 VxVM 디스크 그룹으로 복원합니다. 3 단계에서 데이터를 백업한 방법에 가장 적합한 방법을 사용하십시오.

VxVM의 패키지 사용자 정의

VxVM 디스크 그룹을 만들면 이 디스크 그룹에 액세스할 ServiceGuard 패키지를 사용자 정의해야 합니다. VERITAS 볼륨 관리자(VxVM)에서 사용할 디스크 그룹에 다음 절차를 수행하십시오. 클러스터 볼륨 관리자(CVM)를 사용할 경우 다음 절로 건너뛰십시오.

1. 다음과 같이 기존 패키지 제어 스크립트로 이름을 바꿉니다.

```
#mv Package.ct1 Package.ct1.bak
```

2. 기존 패키지 제어 스크립트와 동일한 이름의 패키지 제어 스크립트를 새로 만듭니다.

```
#cmmakepkg -s Package.ct1
```

3. 새 스크립트를 편집하여 새 VxVM 디스크 그룹 및 논리 볼륨 이름을 포함시킵니다.

VxVM 사용에 필요한 새로운 패키지 제어 스크립트 부분은 다음과 같습니다.

- VXVM_DG[] 어레이. 이 어레이는 이 패키지에 사용된 VxVM 디스크 그룹을 정의합니다. 첫 번째 VXVM_DG[] 항목은 index 0에 있어야 하고, 두 번째 항목은 index 1에 있어야 합니다. 예를 들면 다음과 같습니다.

```
VXVM_DG[0]="DG01"  
VXVM_DG[1]="DG02"
```

- LV[], FS[] 및 FS_MOUNT_OPT[] 어레이는 LVM에서와 동일하게 사용됩니다. LV[]는 논리 볼륨을 정의하고, FS[]는 마운트 지점을 정의하고, FS_MOUNT_OPT[]는 마운트 옵션을 정의합니다. 예를 들어 위에서 두 디스크 그룹 각각에 대한 두 개의 볼륨인 lvol101, lvol102와 lvol201, lvol202가 정의되어 있다고 가정합니다. 이들 볼륨은 각각 /mnt_dg0101, /mnt_dg0102, /mnt_dg0201 및 /mnt_dg0202에 마운트됩니다. /mnt_dg0101 와 /mnt_dg0201 는 모두 읽기 전용으로 마운트됩니다. 이들에 대한 LV[], FS[] 및 FS_MOUNT_OPT[] 항목은 다음과 같습니다.

LVM에서 VxVM 데이터 저장으로 변환 VxVM의 패키지 사용자 정의

```
LV [0] = "/dev/vx/dsk/dg01/lvol101"  
LV [1] = "/dev/vx/dsk/dg01/lvol10102"  
LV [2] = "/dev/vx/dsk/dg02/lvol10201"  
LV [3] = "/dev/vx/dsk/dg02/lvol10202"
```

```
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. 기존 스크립트에서 환경 변수 및 클러스터 정의 함수를 비롯하여 추가된 사용자 특정 코드를 복사합니다.
5. 새로운 패키지 제어 스크립트를 클러스터의 모든 노드에 배포합니다.
6. 디스크 그룹과 데이터가 손상되지 않았는지 검사합니다.
7. 디스크 그룹을 내보냅니다.

```
# vxvg deport DiskGroupName
```

8. 다른 모든 노드에서 다음 명령을 실행하여 클러스터의 다른 노드에 이 디스크 그룹이 보이도록 합니다.

```
# vxctl enable
```

9. 패키지를 다시 시작합니다.

CVM의 패키지 사용자 정의

VxVM 디스크 그룹을 만들면 이 디스크 그룹에 액세스할 ServiceGuard 패키지를 사용자 정의해야 합니다. 클러스터 볼륨 관리자(CVM)에서 디스크 그룹을 사용할 경우 다음 절차를 수행하십시오. VERITAS 볼륨 관리자(VxVM)를 사용할 경우 이전 절의 절차를 수행하십시오.

1. 다음과 같이 기존 패키지 제어 스크립트로 이름을 바꿉니다.

```
#mv Package.ct1 Package.ct1.bak
```

2. 기존 패키지 제어 스크립트와 동일한 이름의 패키지 제어 스크립트를 새로 만듭니다.

```
# cmmakepkg -s Package.ct1
```

3. 새 스크립트를 편집하여 새 CVM 디스크 그룹 및 논리 볼륨 이름을 포함시킵니다.

CVM 사용에 필요한 새로운 패키지 제어 스크립트 부분은 다음과 같습니다.

- CVM_DG[] 어레이. 이 어레이는 이 패키지에 사용된 CVM 디스크 그룹을 정의합니다. 첫 번째 CVM_DG[] 항목은 index 0에 있어야 하고, 두 번째 항목은 index 1에 있어야 합니다. 예를 들면 다음과 같습니다.

```
CVM_DG[0]="DG01"  
CVM_DG[1]="DG02"
```

- LV[], FS[] 및 FS_MOUNT_OPT[] 어레이는 LVM에서와 동일하게 사용됩니다. LV[]는 논리 볼륨을 정의하고, FS[]는 마운트 지점을 정의하고, FS_MOUNT_OPT[]는 마운트 옵션을 정의합니다. 예를 들어 위에서 두 디스크 그룹 각각에 대한 두 개의 볼륨인 lvol101, lvol102와 lvol201, lvol202가 정의되어 있다고 가정합니다. 이들 볼륨은 각각 /mnt_dg0101, /mnt_dg0102, /mnt_dg0201 및 /mnt_dg0202에 마운트됩니다. /mnt_dg0101 and /mnt_dg0201 are both mounted read only. 이들에 대한 LV[], FS[] 및 FS_MOUNT_OPT[] 항목은 다음과 같습니다.

LVM에서 VxVM 데이터 저장으로 변환 CVM의 패키지 사용자 정의

```
LV[0]="/dev/vx/dsk/dg01/lvol101"  
LV[1]="/dev/vx/dsk/dg01/lvol10102"  
LV[2]="/dev/vx/dsk/dg02/lvol10201"  
LV[3]="/dev/vx/dsk/dg02/lvol10202"
```

```
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. 기존 스크립트에서 환경 변수 및 클러스터 정의 함수를 비롯하여 추가된 사용자 특정 코드를 복사합니다.
5. 적절한 CVM_ACTIVATION_CMD 문을 언마운트하여 디스크 그룹에 수행할 패키지 가져오기 종류를 지정합니다.
6. 새로운 패키지 제어 스크립트를 클러스터의 모든 노드에 배포합니다.
7. 패키지 ASCII 구성 파일에서 HALT_SCRIPT_TIMEOUT 매개 변수 바로 뒤에 각 디스크 그룹을 입력합니다. 각 디스크 그룹에 STORAGE_GROUP 정의를 추가합니다. 이전에 예제에서 두 디스크 그룹의 경우 다음 줄을 추가하면 됩니다.

```
STORAGE_GROUP dg01  
STORAGE_GROUP dg02
```

그런 다음 패키지 구성을 다시 적용합니다.

```
# cmapplyconf -P PackageName.ascii
```

8. 디스크 그룹과 데이터가 손상되지 않았는지 검사합니다.
9. 디스크 그룹을 내보냅니다.

```
# vxvg deport DiskGroupName
```

10. 클러스터가 실행 중이 아니면 실행합니다.

```
# cmruncl
```

이 명령은 특수한 CVM 패키지를 활성화합니다.

11. CVM이 실행되면 주 노드를 선택합니다. 이 노드에서 디스크 그룹 구성 명령을 실행해야 합니다. 주 노드를 결정하려면 클러스터의 각 노드에서 다음 명령을 실행하십시오.

```
# vxdctl -c mode
```

한 노드는 자신을 주 노드로 인식합니다.

12. 주 노드에서 다음 명령을 실행하여 클러스터의 다른 노드에 이 디스크 그룹이 보이도록 합니다.

```
# vxdg -s import DiskGroupName
```

13. 패키지를 다시 시작합니다.

LVM 볼륨 그룹 제거

새로운 VxVM 디스크 그룹을 테스트한 후에 표준 LVM 명령인 `lvremove`, `pvremove` 및 `vgremove`를 사용하여 시스템에서 더 이상 사용하지 않을 LVM 볼륨 그룹을 제거합니다. 아무 때나 클러스터 ASCII 구성 파일을 편집하여 클러스터에서 더 이상 사용하지 않을 LVM 볼륨 그룹을 참조하는 `VOLUME_GROUP` 문을 제거해야 합니다. 이들 항목은 클러스터 구성을 다시 적용하기 전에 제거해야 합니다.

Numerics

8개의 노드로 구성된 활성/대기 클러스터 그림, 44

A**APA**

자동 포트 집계, 87

ARP 메시지

전환 이후, 87

ASCII 클러스터 구성 파일 템플리트, 182

ASCII 패키지 구성 파일 템플리트, 206

AUTO_RUN

예제 ASCII 패키지 구성 파일에서, 206

패키지 구성의 매개 변수, 138

AUTO_RUN 매개 변수, 214

AUTO_START_TIMEOUT

예제 구성 파일에서, 183

클러스터 관리자 구성의 매개 변수, 129

C

CLUSTER_NAME(클러스터 이름)
예제 구성 파일에서, 183

cmapplyconf, 190, 226

cmcheckconf, 188, 225

문제 해결, 297

cmclnodelist 파일

보안을 위한, 156

cmdeleteconf

클러스터 구성 삭제, 200

패키지 구성 삭제, 275

cmmodnet

제어 스크립트에서 IP 주소 할당, 82

cmquerycl

문제 해결, 297

CONCURRENT_DISKGROUP_OPERATIONS

패키지 제어 스크립트의 매개 변수, 148

CONCURRENT_FSCK_OPERATIONS

패키지 제어 스크립트의 매개 변수, 149

CONCURRENT_MOUNT_OPERATIONS

패키지 제어 스크립트의 매개 변수, 149

CONCURRENT_VGCHANGE_OPERATIONS

패키지 제어 스크립트의 매개 변수, 148

CVM, 93, 95

VxVM-CVM-pkg 사용, 193

계획, 122

저장 하부 구조 만들기, 192

CVM_ACTIVATION_CMD, 146, 147

패키지 제어 스크립트에서, 215

CVM_DG

패키지 제어 스크립트에서, 215

D

DNS 서비스, 160

DTC

MC/ServiceGuard와 함께 사용, 151

E**EMS**

EMS HA 모니터 사용, 69

디스크 모니터링을 위한, 37

문제 해결 모니터링, 285

패키지 리소스 모니터링, 69

F

FAILBACK_POLICY 매개 변수

패키지 관리자가 사용, 66

패키지 구성 파일에서, 137

FAILOVER_POLICY 매개 변수

패키지 관리자가 사용, 63

패키지 구성 파일에서, 137

FIRST_CLUSTER_LOCK_PV

예제 구성 파일에서, 183

클러스터 관리자 구성의 매개 변수, 127, 128

FIRST_CLUSTER_LOCK_VG

예제 구성 파일에서, 183

FS

예제 패키지 제어 스크립트에서, 215

FS_MOUNT_OPT

예제 패키지 제어 스크립트에서, 215

G**ghostbyname**

패키지 IP 주소, 82

ghostbyname(), 331

H**HA**

디스크 엔클로저, 37

HA 모니터(EMS), 69

HA 클러스터 계획 및 문서화, 103

HA(고가용성) 응용 프로그램과

MC/ServiceGuard 통합, 343

HALT_SCRIPT

예제 ASCII 패키지 구성 파일에서, 206

- 패키지 구성의 매개 변수, 139
HALT_SCRIPT_TIMEOUT(스크립트 제한 시간 중지)
예제 ASCII 패키지 구성 파일에서, 206
패키지 구성의 매개 변수, 140
HEARTBEAT_INTERVAL
예제 구성 파일에서, 183
HEARTBEAT_INTERVAL(하트비트 시간 제한)
클러스터 관리자 구성의 매개 변수, 128
HEARTBEAT_IP
예제 구성 파일에서, 183
클러스터 관리자 구성의 매개 변수, 127
HP, 93
HP Predictive를 사용한 모니터링 문제 해결에서, 286
HP-UX의 OPS용 하드웨어 전원 공급 장치, 42
- I**
I/O 버스 주소
하드웨어 계획, 113
I/O 슬롯
하드웨어 계획, 109, 113
IP
예제 패키지 제어 스크립트에서, 215
패키지 제어 스크립트의 IP 주소 어레이 변수, 149
IP 주소
노드 및 패키지를 위한, 82
이동 가능한, 82
전환, 62, 87
패키지에 대해 검사, 294
패키지에 추가 및 삭제, 83
하드웨어 계획, 109, 118
IP 주소 전환, 62, 87
IP 주소로 로드 분배, 83
- J**
JFS, 325
- L**
LAN
계획 정보, 109
인터페이스 이름, 109, 118
하트비트, 52
LAN 계획
소통량 종류, 110
호스트 IP 주소, 109, 118
LAN 인터페이스
기본 인터페이스 및 대기 인터페이스, 30
네트워크 관리자를 사용한 모니터링, 83
LAN 인터페이스 모니터링
네트워크 관리자에서, 83
LAN 장애
MC/ServiceGuard의 동작, 28
LOCAL_LAN_FAILOVER_ALLOWED
예제 ASCII 패키지 구성 파일에서, 206
패키지 구성의 매개 변수, 138
lssf
디스크 목록을 얻기 위해 사용, 168
LV
예제 패키지 제어 스크립트에서, 215
lvextend
루트 미리 만들기, 158
LVM, 93
VxVM으로 변환, 367
- 계획, 120
논리 볼륨 만들기, 133
다른 노드에 볼륨 그룹 설정, 172, 173
디스크, 35
루트 미리 만들기, 157
볼륨 그룹 만들기, 133
클러스터 사용 명령, 167
파일 시스템 만들기, 133
LVM 구성
워크시트, 121, 123
LVM_ACTIVATION_CMD, 146
- M**
MAC 주소, 330
man 페이지, 309
MC/ServiceGuard에 대한 페이지 목록, 309
MAX_CONFIGURED_PACKAGES
클러스터 관리자 구성의 매개 변수, 130
MC/LockManager
설치, 166
MC/ServiceGuard
소개, 21, 22
MC/ServiceGuard 구성 작업 그림, 26
MC/ServiceGuard 네트워크 구성 요소 이해, 30
MC/ServiceGuard 명령
패키지 구성에 사용, 205
MC/ServiceGuard 소프트웨어 구성 요소
그림, 48
MC/ServiceGuard란?, 22
MC/ServiceGuard에서 지원되는 네트워크, 30
MC/ServiceGuard에서 지원되는 디스크, 35

- MC/ServiceGuard의 동작
LAN 장애 시, **28**
모니터링 중인 자원의 장애 시, **28**
소프트웨어 장애 시, **28**
- MirrorDisk/UX, **36**
- mkboot
루트 미리 만들기, **157**
- N**
- NETWORK_INTERFACE
예제 구성 파일에서, **183**
- NETWORK_POLLING_INTERVAL
(네트워크 폴링 간격)
예제 구성 파일에서, **183**
- NODE_FAIL_FAST_ENABLED
예제 ASCII 패키지 구성 파일에서,
206
패키지 구성의 매개 변수, **139**
- NODE_FAILFAST_ENABLED 매개
변수, **214**
- NODE_NAME
예제 ASCII 패키지 구성 파일에서,
206
클러스터 관리자 구성의 매개 변
수, **126, 127**
- NODE_TIMEOUT(노드 시간 제한)
클러스터 관리자 구성의 매개 변
수, **129**
- NODE_TIMEOUT(하트비트 제한
시간)
예제 구성 파일에서, **183**
- NTP
클러스터의 시간 프로토콜, **162**
- O**
- OPS
시작 및 종료 인스턴스, **214**
- OTS/9000 지원, **355**
- P**
- PACKAGE_NAME
예제 ASCII 패키지 구성 파일에서
, **206**
패키지 ASCII 구성 파일의 매개
변수, **136**
- PACKAGE_TYPE
패키지 ASCII 구성 파일의 매개
변수, **136**
- PATH, **146**
- Predictive를 사용한 모니터링, **286**
- PV 링크, **36**
볼륨 그룹 만들기, **170**
- pvccreate
루트 미리 만들기, **157**
- PVG 완전 미리링
볼륨 그룹 만들기, **169**
- Q**
- QS_HOST
클러스터 관리자 구성의 매개 변
수, **126**
- QS_POLLING_INTERVAL
클러스터 관리자 구성의 매개 변
수, **126**
- QS_TIMEOUT_EXTENSION
클러스터 관리자 구성의 매개 변
수, **126**
- quorum server
계획 워크시트 양식, **360**
상황 및 상태, **234**
새로 생성된 클러스터 사용, **58**
설치, **164**
- quorum 서버
계획, **118**
워크시트, **119**
클러스터 관리자 구성의 매개 변
수, **126**
- R**
- RAID
데이터 보호를 위한, **36**
- RAID 디스크, **36**
- README
데이터베이스 툴킷, **319**
- RESOURCE_NAME
예제 ASCII 패키지 구성 파일에서,
206
패키지 구성의 매개 변수, **142,**
143
- RESOURCE_POLLING_INTERVAL
예제 ASCII 패키지 구성 파일에서,
206
패키지 구성의 매개 변수, **143**
- RESOURCE_UP_VALUE
예제 ASCII 패키지 구성 파일에서,
206
패키지 구성의 매개 변수, **143**
- rhosts 파일
보안을 위한, **156**
- RS232 연결
하트비트에 대한, **111**
- RS232 직렬 하트비트 회선, **34**
- RS232 하트비트 회선, 구성, **110**
- RS232의 상태 보기, **239**
- RS232의 상태, 보기, **239**
- RUN_SCRIPT
예제 ASCII 패키지 구성 파일에서,
206
패키지 구성의 매개 변수, **139**
- RUN_SCRIPT_TIMEOUT
예제 ASCII 패키지 구성 파일에서,
206
- RUN_SCRIPT_TIMEOUT(실행 스
크립트 시간 제한)
패키지 구성의 매개 변수, **140**

S

S800 시리즈 번호
하드웨어 계획, 109

SAM
클러스터 구성에 사용, 181
패키지 구성에 사용, 204

SCSI 주소 지정, 111, 125

SECOND_CLUSTER_LOCK_PV
클러스터 관리자 구성의 매개 변수, 127, 128

SERIAL_DEVICE_FILE(RS232)
클러스터 관리자 구성의 매개 변수, 128

SERVICE_CMD
예제 패키지 제어 스크립트에서, 215
패키지 제어 스크립트의 어레이 변수, 149, 150, 151

SERVICE_FAIL_FAST_ENABLED
예제 ASCII 패키지 구성 파일에서, 206
패키지 구성의 매개 변수, 141

SERVICE_HALT_TIMEOUT
예제 ASCII 패키지 구성 파일에서, 206
패키지 구성의 매개 변수, 142

SERVICE_NAME
예제 ASCII 패키지 구성 파일에서, 206
예제 패키지 제어 스크립트에서, 215
패키지 구성의 매개 변수, 141
패키지 제어 스크립트의 어레이 변수, 149

SERVICE_RESTART
예제 패키지 제어 스크립트에서, 215
패키지 제어 스크립트의 어레이 변수, 150

ServiceGuard Manager
개요, 243
클러스터 개체 관리, 252
클러스터 개체 보기, 248

SGCONF, 155

SNA 응용 프로그램, 335

SPU 정보
계획, 109

STATIONARY_IP
클러스터 관리자 구성의 매개 변수, 128

SUBNET
예제 ASCII 패키지 구성 파일에서, 206
예제 패키지 제어 스크립트에서, 215
패키지 제어 스크립트의 어레이 변수, 148, 149

T

TOC
노드 장애 시, 99

U

uname(2), 332

UPS
HP-UX의 OPS용 전원 공급 장치, 42
전원 계획에서, 116

V

VERITAS, 93

VG
예제 패키지 제어 스크립트에서, 215

vgcfsbackup
및 클러스터 잠금 데이터, 191

VGCHANGE

패키지 제어 스크립트에서, 215

vgxextend
루트 미러 만들기, 157

vgimport
다른 노드에 볼륨 그룹을 설정하는 데 사용, 173

VOLUME_GROUP
예제 구성 파일에서, 183
클러스터 관리자 구성의 매개 변수, 131

VxVM, 93, 94
LVM에서 VxVM으로 변환, 367
계획, 122
저장 하부 구조 만들기, 176

VXVM_DG
패키지 제어 스크립트에서, 215

VxVM-CVM-pkg, 193

ㄱ

개입
사용자 개입 최소화, 322

검사점, 327

계획
LAN 정보, 109
quorum 서버, 118
SCSI 주소, 111
SPU 정보, 109
개요, 103
고가용성 목표, 105
디스크 I/O 정보, 112
디스크 그룹 및 디스크, 122
볼륨 그룹 및 실제 볼륨, 120
실제 볼륨 워크시트, 361
워크시트, 115
전원, 116
클러스터 관리자 구성, 125
클러스터 구성, 124
클러스터 잠금과 클러스터 확장, 125

- 패키지 구성, 133
- 하드웨어 구성, 108
- 확장, 135
- 계획 워크시트
 - 양식, 357
- 계획된 다운시간 최소화, 340
- 고가용성, 22
 - HA(고가용성) 클러스터 정의, 28
 - 계획 목표, 105
 - 고가용성 디스크 어레이, 36
 - 고가용성 디스크 어레이를 사용한 클러스터
 - 그림, 39, 40
 - 고가용성을 위해 연결된 미러링 디스크
 - 그림, 38
 - 고정 IP 주소, 82
 - 공유 디스크
 - 계획, 112
 - 공유 버스의 루트 디스크 제한 사항, 40
- 관리
 - 구성 파일 검사, 296
 - 모든 노드가 중지되었을 때 클러스터 시작, 256
 - 문제 해결, 294
 - 실행 중인 클러스터에 노드 추가, 257
 - 실행 중인 클러스터에서 노드 제거, 258
 - 전체 클러스터 중지, 260
 - 클러스터 및 패키지 상태, 230
 - 클러스터 실행 중에 패키지 재구성, 273
 - 클러스터 이벤트에 대응, 278
 - 클러스터 재구성, 267
 - 클러스터가 오프라인 상태에서 패키지 재구성, 273
 - 클러스터의, 256
- 패키지 및 서비스, 261
- 패키지 시작, 261
- 패키지 이동, 263
- 패키지 중지, 262
- 구성
 - ASCII 클러스터 구성 파일 템플리트, 182
 - ASCII 패키지 구성 파일 템플리트, 206
 - 기본 작업 및 단계, 26
 - 서비스, 203
 - 클러스터 계획, 124
 - 클러스터의, 52
 - 패키지, 203
 - 패키지 계획, 133
- 구성 파일
 - 문제 해결, 296
 - 클러스터 관리자를 위한, 52
- 그림
 - 8개의 노드로 구성된 활성/대기 클러스터, 44
 - EMC 디스크 어레이를 사용한 8개의 노드로 구성된 클러스터, 45
 - MC/ServiceGuard 소프트웨어 구성 요소, 48
 - MC/ServiceGuard 클러스터 구성 작업, 26
 - 고가용성 디스크 어레이를 사용한 클러스터, 39, 40
 - 고가용성을 위해 연결된 미러링 디스크, 38
 - 노드 1로 이동한 패키지를 포함하는 클러스터 실행, 352
 - 노드 1을 HP-UX 10.01로 업그레이드, 351
 - 노드 1이 클러스터에 다시 참여, 352
- 노드 2로 이동한 패키지를 포함하는 클러스터 실행, 351
- 롤링 업그레이드 전에 클러스터 실행, 350
- 서로 다른 공유 버스 상의 루트 디스크, 41
- 서로 다른 공유 버스에 존재하는 기본 디스크와 미러 디스크, 41
- 업그레이드 이후에 클러스터 실행, 353
- 이중 부착 FDDI 스테이션을 사용한 구성, 33
- 일반적인 클러스터 구성, 22
- 장애 조치 후의 클러스터, 24
- 중복 FDDI 구성, 32
- 중복 LAN, 31
- 직렬(RS232) 하트비트 회선, 34
- 클러스터 구성 예제, 108
- 기본 LAN 인터페이스
 - 정의, 30
- 기본 네트워크 인터페이스, 30
- L
- 네트워크
 - HA 서비스로서 네트워크 응용 프로그램 작성, 323
 - IP 주소로 로드 분배, 83
 - IP 주소로 바인딩, 333
 - IP 주소를 사용하는 패키지, 331
 - IP 주소와 이름, 329
 - MC/ServiceGuard에서 지원되는 종류, 30
 - OTS/9000 지원, 355
 - 노드 및 패키지 IP 주소, 82
 - 로컬 인터페이스 전환, 83
 - 로컬 전환, 84
 - 원격 시스템 전환, 86
 - 장애, 84

- 중복, 30, 34
- 패키지 IP 주소 추가 및 삭제, 83
- 포트 주소로 바인딩, 332
- 네트워크 계획
 - 서브넷, 109, 118
- 네트워크 관리자
 - LAN 인터페이스 모니터링, 83
 - 주요 기능, 82
 - 테스트, 284
 - 패키지 IP 주소 추가 및 삭제, 83
- 네트워크 관리자의 작동 방식, 82
- 네트워크 구성 요소
 - MC/ServiceGuard에서, 30
- 네트워크 시간 프로토콜(NTP)
 - 클러스터에 대하여, 162
- 네트워크 인터페이스의 중복, 30
- 네트워크 통신 장애, 101
- 네트워크 폴링 주기
 - (NETWORK_POLLING_INTERVAL)
 - 클러스터 관리자 구성의 매개 변수, 130
- 네트워킹
 - 중복 서브넷, 109
- 노드
 - IP 주소, 82
 - MC/ServiceGuard 클러스터에서, 22
 - 기본 개념, 28
- 노드 종류
 - 주, 23
 - 활성, 23
- 논리 볼륨
 - 계획, 120
 - 계획 워크시트 양식, 362
 - 워크시트, 121, 123
 - 클러스터에 대해 만들기, 169, 178, 195
 - 하부 구조 만들기, 167, 176
- ㄷ
 - 다운시간
 - 최소화 계획된, 340
 - 다중 시스템
 - 응용 프로그램 설계, 329
 - 다중 시스템에서 실행 가능한 응용 프로그램 설계, 329
 - 단일 노드 실행, 200
 - 단일 클러스터 잠금
 - 선택, 57
 - 대기 LAN 인터페이스
 - 정의, 30
 - 대기 네트워크 인터페이스, 30
 - 대기 노드 순환
 - 장애 조치 정책으로 구성, 64
 - 패키지 정책 설정, 64
 - 대응
 - 클러스터 이벤트, 278
 - 패키지 및 서비스 장애, 100
 - 대체 노드, 23
 - 대체 링크
 - 볼륨 그룹 만들기, 170
 - 데이터, 89
 - 디스크, 35
 - 데이터 정체, 52
 - 데이터베이스
 - 틀킷, 319
 - 데이터의 미러 사본
 - 디스크 장애 대비, 23
 - 동적 클러스터 재편성, 54
 - 디스크
 - MC/ServiceGuard에서, 35
 - MC/ServiceGuard에서 지원되는 종류, 35
 - 교체, 287
 - 구성 예제, 37, 39
 - 데이터, 35
 - 루트, 35
 - 미러링, 36
 - 볼륨 그룹을 위한 선택, 168
 - 인터페이스, 35
 - 디스크 I/O
 - 하드웨어 계획, 113
 - 디스크 관리, 89
 - 디스크 교체, 287
 - 디스크 구성 예제, 37, 39
 - 디스크 그룹
 - 계획, 122
 - 디스크 그룹 및 디스크 계획, 122
 - 디스크 논리 단위
 - 하드웨어 계획, 113
 - 디스크 모니터, 37
 - 디스크 모니터(EMS), 69
 - 디스크 어레이
 - PV 링크를 사용하여 볼륨 그룹 만들기, 170
 - 계획, 120
 - 디스크 어레이, 고가용성, 36
 - 디스크 어레이를 사용한 8개의 노드로 구성된 클러스터
 - 그림, 45
 - 디스크 엔클로저
 - 고가용성, 37
 - 디스크 장애
 - 미러링을 통해 보호, 23
 - 디스크 저장
 - CVM을 사용하여 하부 구조 만들기, 192
 - 디스크의 엔클로저
 - 결함 있는 장치 교체, 287
- ㄹ
 - 로컬 전환, 83
 - 패키지 구성의 매개 변수, 138
 - 롤링 소프트웨어 업그레이드, 347
 - 단계, 348
 - 예제, 350
 - 롤링 업그레이드

- 제한 사항, 354
- 루트 미러
LVM을 사용하여 만들기, 157
- 리소스 Up 주기
패키지 구성의 매개 변수, 143
- 리소스 이름
패키지 구성의 매개 변수, 142, 143
- 리소스 풀링 주기
패키지 구성의 매개 변수, 143
- 링크 수준 주소, 330
- ▣
- 마운트 옵션
제어 스크립트에서, 147
- 매개 변수
AUTO_RUN, 214
NODE_FAILFAST_ENABLED, 214
장애 조치에 대한, 70
- 메모리 요구 사항
MC/ServiceGuard의 잠금 가능 메모리, 106
- 메모리 용량
하드웨어 계획, 109
- 모니터링 중인 자원의 장애
MC/ServiceGuard의 동작, 28
- 모니터링된 하트비트 이외의 서버넷
클러스터 관리자 구성의 매개 변수, 128
- 문제 해결
cmquerycl 및 cmcheckconf 사용, 297
디스크 교체, 287
방법, 294
시스템 로그 파일 검사, 295
제어 스크립트 검사, 297
패키지 IP 주소 검사, 294
- 하드웨어 모니터링, 285
- 문제점 해결, 300
- 물리 볼륨
클러스터 잠금 구성의 매개 변수, 127
- 클러스터 잠금의, 56
- 미러링
디스크, 36
- ≡
- 마인딩
네트워크 응용 프로그램에서, 332, 333
- 버스 종류
하드웨어 계획, 113
- 변동 가능 IP 주소
MC/ServiceGuard 패키지에서, 82
- 정의, 82
- 보다 큰 클러스터, 43
- 보류된 리소스 이름, 150, 151
- 보안
파일 편집, 156
- 복구 시간, 124
- 볼륨 관리자, 89
- CVM, 95
- LVM, 93
- LVM에서 VxVM으로 변환, 367
- VxVM, 94
- 비교, 97
- 볼륨 그룹
LVM 명령을 사용하여 다른 노드에 설정, 173
- SAM을 사용하여 다른 노드에 설정, 172
- 계획, 120
- 다른 노드로 내보내기 전에 비활성화, 172
- 워크시트, 121, 123
- 제어 스크립트에서, 147
- 클러스터 잠금의, 56
- 클러스터에 대해 만들기, 169, 170
- 클러스터의 실제 볼륨 만들기, 168
- 볼륨 그룹 및 실제 볼륨 계획, 120
- 볼륨 그룹 비활성화, 172
- 브리지 연결 네트워크
네트워크 인터페이스의 중복을 위한, 30
- 정의, 30
- ∧
- 사용자 정의 함수
제어 스크립트에 추가, 222
- 상태
시스템 로그 파일, 295
- 클러스터 및 패키지, 230
- 패키지 IP 주소, 294
- 서로 다른 공유 버스 상의 루트 디스크
그림, 41
- 서로 다른 공유 버스에 존재하는 기본 디스크와 미러 디스크
그림, 41
- 서버넷
하드웨어 계획, 109, 118
- 서비스 관리, 261
- 서비스 구성
단계별, 203
- 서비스 명령
패키지 제어 스크립트의 변수, 149, 150, 151
- 서비스 이름, 149
- 패키지 구성의 매개 변수, 141
- 패키지 제어 스크립트의 변수, 149
- 서비스 장애
대응, 100

- 서비스 재시작, **101**
 - 서비스 재시작 매개 변수
 - 패키지 제어 스크립트의 변수, **150**
 - 서비스 조기 차단
 - 패키지 구성의 매개 변수, **141**
 - 서비스 중지 시간 제한
 - 패키지 구성의 매개 변수, **142**
 - 소개
 - MC/ServiceGuard 소개, **22**
 - MC/ServiceGuard 소프트웨어 이해, **47**
 - MC/ServiceGuard 하드웨어 이해, **27**
 - 소속 변경
 - 이유, **54**
 - 소통량 종류
 - LAN 하드웨어 계획, **110**
 - 소프트웨어 계획
 - CVM 및 VxVM, **122**
 - LVM, **120**
 - 소프트웨어 설치
 - MC/LockManager, **166**
 - quorum 서버, **164**
 - 소프트웨어 장애
 - MC/ServiceGuard의 동작, **28**
 - 수동 클러스터 시작, **53**
 - 시간 프로토콜(NTP)
 - 클러스터에 대하여, **162**
 - 시스템 다중 노드 패키지
 - CVM에서 사용, **193**
 - 시스템 로그 파일
 - 문제 해결, **295**
 - 시스템 메시지
 - 클러스터에 대해 변경, **199**
 - 시스템에서 MC/ServiceGuard 제거, **279**
 - 시작과 종료
 - 응용 프로그램에 대해 정의된, **323**
 - 실제 볼륨 계획, **120**
 - 계획 워크시트에 입력, **361**
 - 워크시트, **121, 123**
 - 클러스터에 대해 만들기, **168**
 - 실행 중인 클러스터 재구성, **268**
 - 실행 중인 클러스터에 노드 추가, **257**
 - 실행 중인 클러스터에 패키지 추가, **274**
 - 실행 중인 클러스터에서 노드 제거, **258**
 - 실행 중인 클러스터에서 패키지 삭제, **275**
 - ㅇ
 - 어레이
 - 결함 있는 장치 교체, **287**
 - 데이터 보호를 위한 디스크 어레이, **36**
 - 언마운트 횟수, **148**
 - 엔클로저
 - 고가용성, **37**
 - 온라인 하드웨어 유지 관리
 - 인라인 SCSI 터미네이터를 사용하여, **288**
 - 워크시트
 - quorum 서버 구성, **119**
 - 계획에서 사용, **103**
 - 볼륨 그룹 및 실제 볼륨, **121, 123**
 - 실제 볼륨 계획, **361**
 - 전원 공급 장치 구성, **117**
 - 클러스터 구성, **132**
 - 패키지 구성 데이터, **145**
 - 패키지 제어 스크립트, **152**
 - 하드웨어 구성, **115**
 - 원격 전환, **86**
 - 원시 볼륨, **324**
 - 유동 IP 주소
 - MC/ServiceGuard 패키지에서, **82**
 - 정의, **82**
 - 응용 프로그램
 - MC/ServiceGuard와 통합하기 위한 단계의 검사 목록, **343**
 - 네트워크를 위한 HA 서비스 작성, **323**
 - 자동화, **322**
 - 장애 처리, **338**
 - 응용 프로그램 작동 자동화, **322**
 - 응용 프로그램 장애 조치 속도 제어, **324**
 - 응용 프로그램 장애 처리, **338**
 - 응용 프로그램에서 클라이언트 연결 복원, **336**
- 이더넷
 - 중복 구성, **30**
 - 이름 확인 서비스, **160**
 - 이벤트 모니터링 서비스
 - 디스크 모니터링을 위한, **37**
 - 문제 해결에서, **285**
 - 사용, **69**
 - 이중 부착 FDDI 스테이션, **32**
 - 이중 부착 FDDI 스테이션을 사용한 구성
 - 그림, **33**
 - 이중 클러스터 잠금
 - 선택, **57**
 - 인라인 터미네이터
 - 온라인 하드웨어 유지 관리 허용, **288**
 - 인터넷
 - 툴킷, **319**
 - 일반적인 클러스터 구성
 - 그림, **22**

ㅈ

자동 시작 지연
클러스터 구성 파일의 매개 변수, 129

자동 장애 복구
장애 조치 정책으로 구성, 66

자동 전환
패키지 구성의 매개 변수, 138

자동 포트 집계
정의, 87

자동으로 클러스터 재시작, 260

자원
디스크, 35

잠금
quorum 서버 사용, 58
클러스터 잠금 디스크 사용, 56
클러스터 잠금 장치와 전원 공급 장치, 42

잠금 디스크
4개 이상의 노드, 57

잠금 볼륨 그룹
계획, 124, 125
구성 파일에서 식별, 185

잠금 볼륨 그룹 재구성, 267

잠금 볼륨 그룹, 재구성, 267

장애
네트워크 통신, 101
대응 종류, 99
응용 프로그램, 338
장애 발생 이후의 서비스 재시작, 101
패키지 및 서비스 장애 대응, 100
하드웨어 장애 대응, 100

장애 감지
네트워크 관리자에서, 83

장애 대응, 99

장애 복구 정책
패키지 관리자가 사용, 66
패키지 구성 파일 매개 변수, 137

장애 요인
네트워크에서, 30, 34

장애 조치
응용 프로그램의 속도 제어, 324
정의, 23

장애 조치 동작
패키지에서, 70

장애 조치 정책
패키지 관리자가 사용, 63
패키지 구성 매개 변수, 137

장애 조치 후의 클러스터
그림, 24

재시도 횟수, 148

재시작
장애 발생 후, 101
클러스터 자동 재시작, 54
패키지 제어 스크립트의 SERVICE_RESTART 변수, 150

재시작 가능한 트랜잭션, 326

재편성
클러스터, 54

재편성 시간, 125

저장 관리, 89

저장 장치에 지점간 연결, 44

전반적인 계획, 105

전원 계획
워크시트, 117
전원 공급, 116

전원 공급 장치
HP-UX의 OPS용 UPS, 42
계획 워크시트 양식, 358, 359
및 클러스터 잠금, 42

전체 클러스터 재구성, 267

전체 클러스터 중지, 260

전환
로컬 인터페이스 전환, 83
원격 시스템 전환, 86
전환 이후의 ARP 메시지, 87

제어 스크립트
SAM을 사용한 작성, 212, 213
명령을 사용한 작성, 213
문제 해결, 297
사용자 정의 함수 추가, 222
패키지 구성에서, 213
패키지 구성의 경로 이름 매개 변수, 139

종료와 시작
응용 프로그램에 대해 정의된, 323

주 노드, 23

주소 지정, SCSI, 111

중복
네트워크에서, 30, 34
클러스터 구성 요소, 28

중복 FDDI 구성
그림, 32

중복 FDDI 연결, 32

중복 LAN
그림, 31

중복 네트워크
하트비트, 23

중복 이더넷 구성, 30

직렬 포트
하트비트에 사용, 111

직렬 하트비트 연결
식별, 187

직렬(RS232) 하트비트 회선, 34
그림, 34

ㅊ

최대 노드 수, 28

추가 패키지 리소스
모니터링, 69
패키지 구성의 매개 변수, 142, 143

ㄴ

커널 일관성

클러스터 구성에서, 162

클라이언트 연결

응용 프로그램에서 복원, 336

클러스터

MC/ServiceGuard, 22

SAM을 사용한 구성, 181

구성 요소 이해, 28

구성 요소 중복, 28

명령을 사용한 구성, 182

보다 큰 크기, 43

일반적인 구성, 22

활성/대기 종류, 43

클러스터 관리, 256

문제점 해결, 300

클러스터 관리자

quorum 서버 시간 제한 확장 매개 변수, 126

quorum 서버 폴링 주기 매개 변수, 126

계획 워크시트 양식, 363

구성 계획, 125

네트워크 폴링 주기 매개 변수, 130

노드 시간 제한 매개 변수, 129

동적 재편성, 54

모니터링된 하트비트 이외의 서브넷, 128

물리 잠금 볼륨 매개 변수, 127 정의, 52

주요 기능, 52

직렬 장치 파일 매개 변수, 128

최대 구성 패키지 매개 변수, 130

클러스터 노드 매개 변수, 126, 127

클러스터 볼륨 그룹 매개 변수, 131

클러스터 자동 재시작, 54

클러스터의 최초 구성, 52

테스트, 283

하트비트 서브넷 매개 변수, 127

하트비트 주기 매개 변수, 128

클러스터 관리자 매개 변수

최초 구성, 52

클러스터 관리자 작동 방식, 52

클러스터 구성

SAM 또는 명령을 사용하여 만들기, 181

계획, 124

계획 워크시트, 132

모든 노드에 있는 파일, 52

예제 도표, 108

클러스터 구성 확인, 188

클러스터 인식 볼륨 그룹 식별, 186

클러스터 잠금 볼륨 그룹 식별, 185

클러스터 구성 삭제

cmddeleteconf 사용, 200

클러스터 구성 예제

그림, 108

클러스터 구성 파일, 183

자동 시작 지연 매개 변수

(AUTO_START_TIMEOUT), 129

클러스터 구성 확인, 188

클러스터 노드

OPS 인스턴스 시작 및 종료, 214

클러스터 관리자 구성의 매개 변수, 126, 127

클러스터 노드 추가

사전 계획, 162

클러스터 만들기

ASCII 클러스터 구성 파일 템플릿, 182

CVM 하부 구조, 192

quorum 서버 식별, 185

VxVM 하부 구조, 176

논리 볼륨 하부 구조, 167

클러스터 구성 단계, 181

클러스터 구성 확인, 188

클러스터 잠금 볼륨 그룹 식별, 185

하트비트 서브넷 확인, 186

클러스터 매개 변수

최초 구성, 52

클러스터 문제 해결, 281

클러스터 및 노드 관리, 256

클러스터 및 패키지 유지 관리, 229

클러스터 볼륨 그룹

실제 볼륨 만들기, 168

클러스터 관리자 구성의 매개 변수, 131

클러스터 소속 변경, 54

클러스터 시작

모든 노드가 중지되었을 때, 256

수동, 53

클러스터 실행 중에 노드 삭제,

269, 271

클러스터 실행 중에 노드 추가, 268

클러스터 실행 중에 볼륨 그룹 구성 변경, 270

클러스터 인식 볼륨 그룹 식별, 186

클러스터 자동 시작

설정, 199

클러스터 자동 재시작, 54, 260

클러스터 작동 테스트, 282

클러스터 잠금

2개의 노드, 56

4개 이상의 노드, 59

구성 데이터 저장, 191

구성 파일에서 식별, 185

단일 잠금 디스크, 57

및 전원 공급 장치, 42

새로 생성된 클러스터 사용, 56

이중 잠금 디스크, 57

- 잠금 데이터 백업, 159
- 잠금 사용 안 함, 59
- 클러스터 재편성 시간, 124, 125
- 클러스터 잠금 디스크 정보의 백업, 159
- 클러스터 잠금 사용, 56, 58
- 클러스터 잠금 사용 안 함 선택, 59
- 클러스터 재편성 시간, 125
- 클러스터 중단, 260
- 클러스터 중지, 260
- 클러스터 코디네이터 정의, 52
- 클러스터 크기 변경 준비, 162
- 클러스터 확장 사전 계획, 106
- 클러스터 확장 계획, 106
- 클러스터가 실행되는 동안 허용되는 클러스터 변경, 266
- 클러스터가 실행되는 동안 허용되는 패키지 변경, 276
- 클러스터가 오프라인 상태에서 패키지 재구성, 273
- 클러스터와 패키지 구성 배포, 225, 226
- 클러스터와 패키지 구성 확인, 225, 226
- 테스트
 - 네트워크 관리자, 284
 - 클러스터 관리자, 283
 - 패키지 관리자, 282
- 템플릿
 - ASCII 클러스터 구성 파일, 182
 - ASCII 패키지 구성 파일, 206
- 툴킷
 - 데이터베이스용, 319
- ㅁ
 - 파일 시스템
 - 계획, 120
 - 제어 스크립트에서, 147
 - 클러스터에 대해 만들기, 170, 179, 196
 - 파일 시스템 마운트 재시도 횟수, 148
 - 파일 시스템 언마운트 횟수, 148
 - 파일 잠금, 335
 - 패키지
 - MC//ServiceGuard 클러스터에서, 22
 - OPS 인스턴스 시작, 214
 - 기본 개념, 28
 - 데이터베이스용 툴킷, 319
 - 로컬 인터페이스 전환, 83
 - 시작, 261
 - 원격 전환, 86
 - 이동, 263
 - 중지, 262
 - 클러스터 실행 중에 재구성, 273
 - 클러스터가 실행되는 동안 허용되는 변경, 276
 - 클러스터가 오프라인 상태에서 패키지 재구성, 273
 - 패키지 IP 주소 추가 및 삭제, 83
 - 패키지를 실행하고 중지하는 시간 및 장소 결정, 61
 - 패키지 IP 주소
 - 검사, 294
 - 정의, 82
 - 패키지 관리, 261
 - 문제점 해결, 300
 - 패키지 관리자
 - 계획 워크시트 양식, 364, 365
 - 테스트, 282
 - 패키지 구성
 - HP-UX 명령 사용, 205
 - SAM에서, 204
 - 계획, 133
 - 구성 파일 배포, 225, 226
 - 구성 확인, 225, 226
 - 단계별, 203
 - 로컬 전환 매개 변수, 138
 - 리소스 Up 매개 변수, 143
 - 리소스 폴링 주기 매개 변수, 143
 - 서브넷 매개 변수, 142
 - 서비스 이름 매개 변수, 141
 - 서비스 조기 차단 매개 변수, 141
 - 서비스 중지 시간 제한 매개 변수, 142
 - 실행 및 중지 스크립트 시간 제한 매개 변수, 140
 - 자동 전환 매개 변수, 138
 - 장애 복구 정책 매개 변수, 137
 - 장애 조치 정책 매개 변수, 137
 - 제어 스크립트 경로 이름 매개 변수, 138
 - 추가 패키지 리소스 매개 변수, 142, 143
 - 패키지 구성의 매개 변수, 136
 - 패키지 이름 매개 변수, 136
 - 패키지 제어 스크립트 작성, 213
 - 패키지 조기 차단 매개 변수, 139
 - 패키지 종류 매개 변수, 136
 - 패키지 구성 만들기, 204
 - 패키지 구성 삭제
 - cmddeleteconf 사용, 275
 - 패키지 구성 파일, 206
 - 패키지 및 서비스 구성, 203
 - 패키지 및 클러스터 유지 관리, 229
 - 패키지 시작, 261
 - 패키지 이동, 263
 - 패키지 장애
 - 대응, 100
 - 패키지 장애 조치 동작, 70
 - 패키지 재구성

- 클러스터 실행 중, 273
- 패키지 전환 동작
 - 변경, 264
- 패키지 제어 스크립트
 - IP 주소, 148, 149
 - 명령을 사용하여 만들기, 213
 - 서브넷, 148, 149
 - 서비스 명령, 149
 - 서비스 이름, 149
 - 서비스 재시작 변수, 150
 - 워크시트, 152
- 패키지 제어 스크립트의 성능 변수,
148, 149
- 패키지 초기 차단
 - 패키지 구성의 매개 변수, 139
- 패키지 종류
 - 패키지 구성의 매개 변수, 136
- 패키지 중지, 262
- 패키지 코디네이터
 - 정의, 52
- 패키지를 실행하고 중지하는 시간
및 장소 결정, 61
- 포트
 - 이중 및 단일 집계, 88
- ≡
- 하드웨어
 - 계획 워크시트 양식, 358
 - 모니터링, 285
- 하드웨어 계획
 - I/O 버스 주소, 113
 - I/O 슬롯 개수, 109
 - I/O 슬롯 번호, 113
 - LAN 소통량 종류, 110
 - LAN 인터페이스 이름, 109, 118
 - LAN 정보, 109
 - RS232 하트비트 회선, 110
 - S800 시리즈 번호, 109
 - SPU 정보, 109
 - 공유 디스크의 디스크 I/O 정보,
112
 - 구성 계획, 108
 - 디스크 I/O 버스 종류, 113
 - 메모리 용량, 109
 - 서브넷, 109, 118
 - 워크시트, 115
 - 호스트 IP 주소, 109, 118
 - 호스트 이름, 109
- 하드웨어 모니터링, 285
- 하드웨어 장애
 - 대응, 100
 - 하드웨어 장애 대응, 100
- 하트비트
 - RS232 회선, 111
 - 하트비트 메시지, 23
 - 정의, 52
 - 하트비트 서브넷 주소
 - 클러스터 관리자 구성의 매개 변
수, 127
 - 하트비트 주기
 - 클러스터 관리자 구성의 매개 변
수, 128
- 하트비트 회선
 - RS232 구성, 110
 - 하트비트 회선, 직렬, 34
- 한 곳에서 장애가 발생
 - 예방, 22
- 호스트 IP 주소
 - 하드웨어 계획, 109, 118
- 호스트 이름
 - 하드웨어 계획, 109
- 확장
 - 계획, 135
- 활성 노드, 23