

Service Guard Configuration guide.

1) INTRODUCTION	3
2) HARDWARE LAYOUT	3
3) CHANGING INSTANCE NUMBERS OF CONTROLLER ID'S	6
PROCEDURE 1.....	7
PROCEDURE 2.....	8
PROCEDURE 3.....	9
4) SOME SOFTWARE REQUIREMENTS BEFORE YOU START.....	11
CONFIGURE NTP	11
CONFIGURE DNS.....	11
MIRRORING BOOT DISKS ON BOTH MACHINES.....	12
5) READY TO INSTALL SERVICE GUARD.	13
6) CREATING A CLUSTER	14
7) CREATING A BASIC PACKAGE.	17
8) CREATING AN NFS PACKAGE	21
9) CREATING AN ORACLE PACKAGE.....	24
10) OTHER ADMINISTRATION STUFF.....	28
STARTING AND STOPPING A PACKAGE.....	28
DELETING A PACKAGE.....	28
STOPPING AND STARTING THE CLUSTER.....	28
STOPPING AND STARTING A NODE IN THE CLUSTER.....	29
DELETING A CLUSTER.....	29
OTHER USEFUL COMMANDS.....	29
APPENDIX 1	30
TUNE KERNEL OR ORACLE INSTALL.....	30
CREATE "DBA" GROUP	30
CREATE "ORACLE" USER.....	30
EDIT "ORACLE'S" .PROFILE.....	30
AS ROOT , DO THE FOLLOWING: -	30
AS ORACLE, DO THE FOLLOWING: -	31
AFTER INSTALL AS ORACLE: -	31
ADMINISTRATION OF DATABASES: -	31

1) Introduction

This document is intended to give a step by step configuration guide for MC/Service guard. It will use 3 packages, Xclock, NFS, and Oracle, configured in a 2 node cluster. This document was created using MC/SG version A11.12 on HPUX 11.00.

2) Hardware layout

The hardware layout has to remove all SPOF's (Single points of failure). This is done by means of duplication of hardware kit. For example if a lan card fails, then there needs to be a spare lan available to take its place. If a scsi controller dies, then there needs to be another one to take its place. (In this layout though, there is a SPOF in the switch. There should be 2 switches to allow for a failure.)

Fig 2.1 shows how our 2 machines are laid out for network resilience.

On machine "pch1test"

Lan0 = 192.168.100.101

Lan1 = 164.39.8.92

Lan2 = not configured.

On Machine "pch2test"

Lan0 = 192.168.100.102

Lan1 = 164.39.8.93

Lan2 = not configured.

On both machines lan1 and lan2 are connected into the switch, and both lan0's are connected by a crossovercable.

Fig 2.2 shows how our 2 machines are laid out for disk and scsi resilience.

Both machine have got 2 scsi channels to the disks, so that each disk can be reached via 2 routes, in order to remove any SPOF. Because both machines see each individual disk from different scsi controllers, they will assign different controller ID's to the same disk. (This is shown in the red and blue controller numbers.) The controller ID's have been changed for our setup by changing the instance numbers of the controllers ID's. This is covered in the next section.

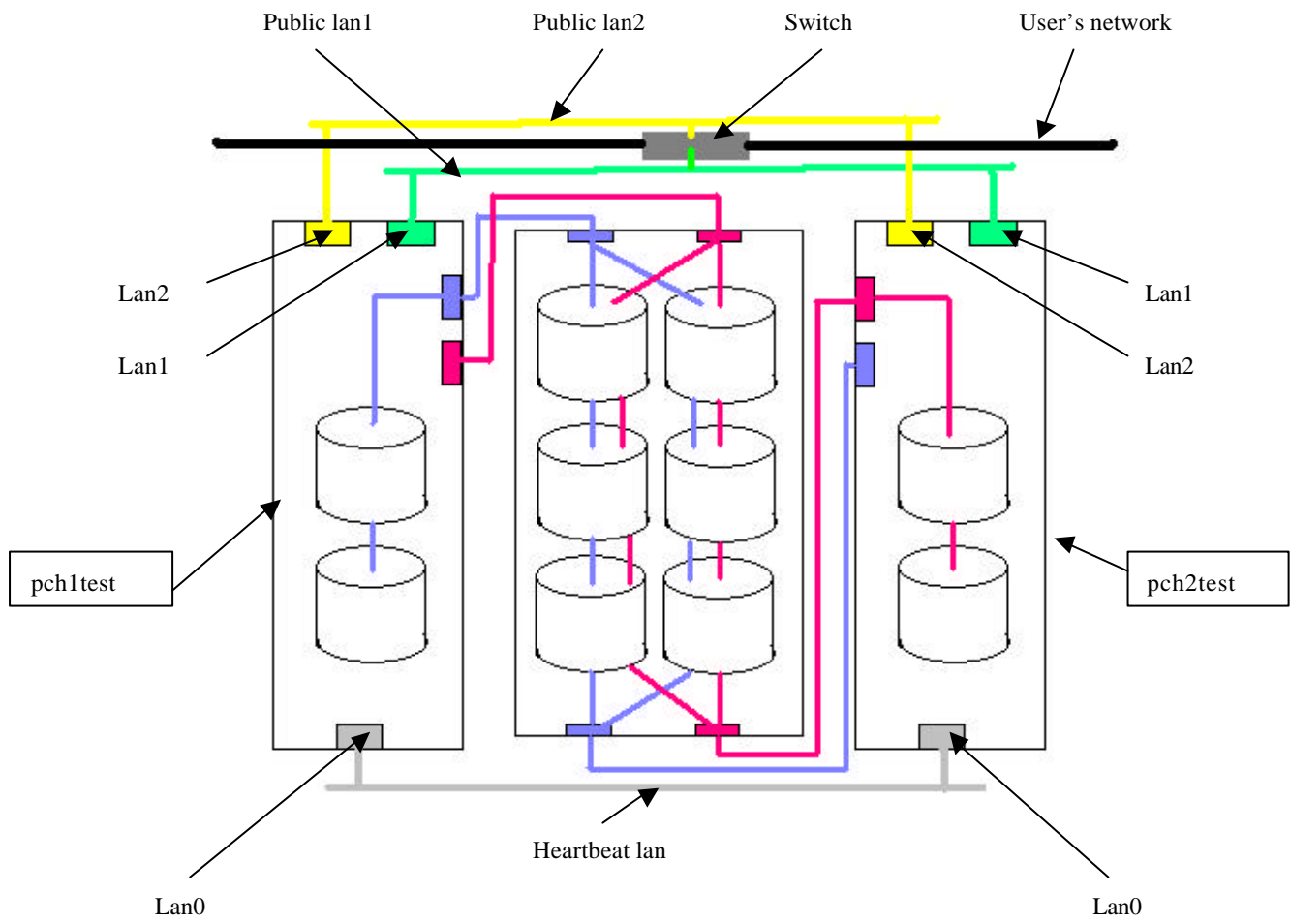


Fig 2.1

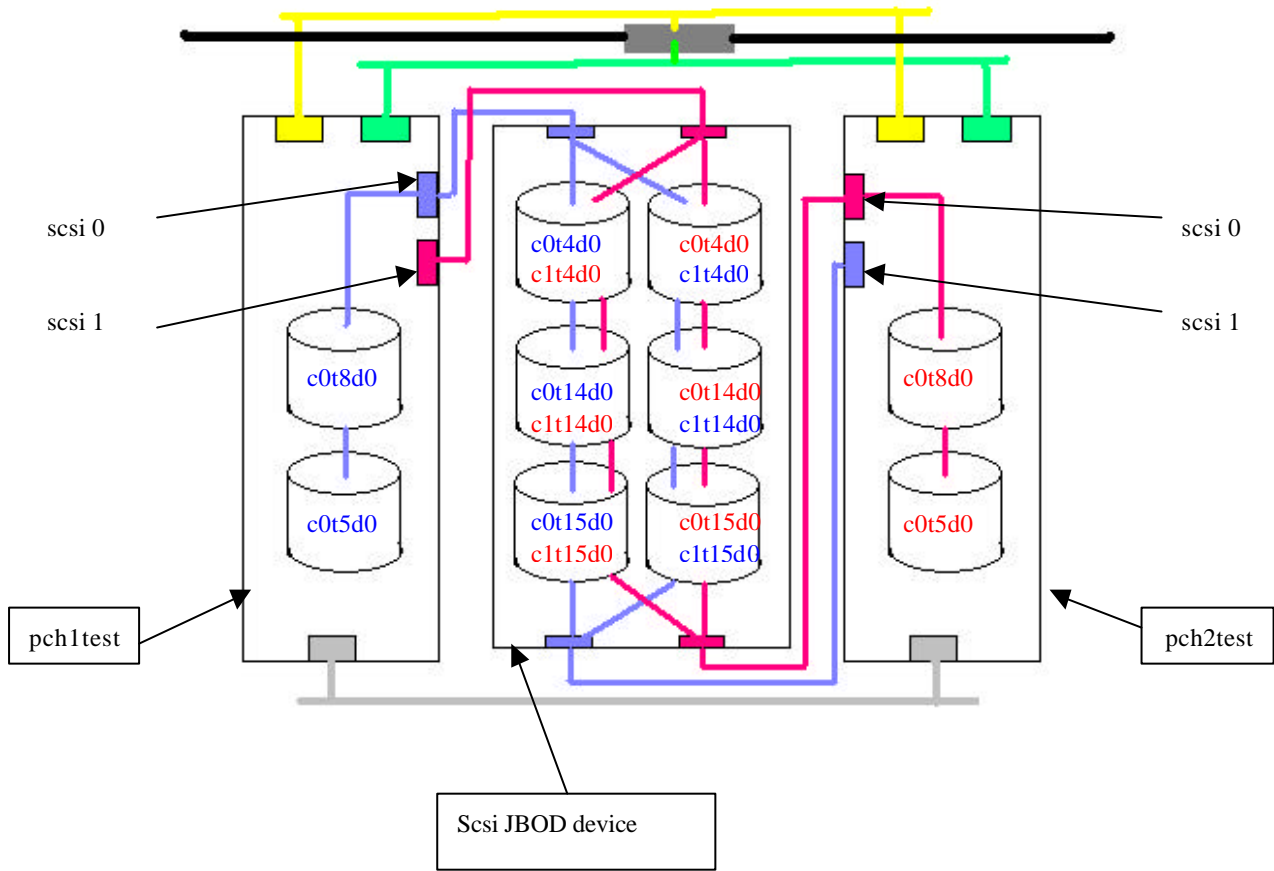


Fig 2.2

3) Changing instance numbers of controller ID's

To make the disk ID's the same on both machines we will need to change the scsi ID's for the controllers at the software level. To make this change, you only need change one machine, so we will use pch2test.

Be sure to have the ioinit(1M) patch installed:

10.20: PHCO_16407 (or greater)

11.00: PHCO_12555 (or greater)

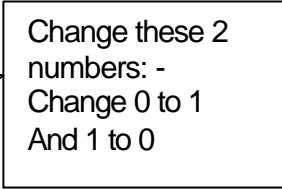
Usually Procedure I is sufficient and there is no need to try II or III. These are only needed if Procedure I fails. Procedure II usually works in all cases, but requires two reboots. Procedure III needs only one reboot and should work in all cases, but you need the ioconfig2infile tool (contact HP to get this unsupported tool).

Procedure 1

```
# ioscan -f | grep -e INTERFACE -e DEVICE | \
  grep -v target | \
  awk '{print $3, $1, $2}' > /infile
```

```
# vi /infile
```

```
8/0/0 tty 0
8/4 ext_bus 0
8/4.4.0 disk 0
8/4.5.0 disk 1
8/4.6.0 ctl 0
8/4.8.0 disk 2
8/4.14.0 disk 3
8/4.15.0 disk 4
8/8 ext_bus 1
8/8.4.0 disk 5
8/8.5.0 disk 6
8/8.6.0 ctl 1
8/8.8.0 disk 7
8/8.14.0 disk 8
8/8.15.0 disk 9
8/16/0 ext_bus 3
8/16/5 ext_bus 2
8/16/5.1.0 tape 0
8/16/5.2.0 disk 10
8/16/5.7.0 ctl 2
8/16/6 lan 0
8/16/7 ps2 0
8/20/2 tty 1
8/20/5/1 lan 1
20/5/8 lan 2
```



Change these 2
numbers: -
Change 0 to 1
And 1 to 0

```
# init 1
# /sbin/iocinit -f /infile -r

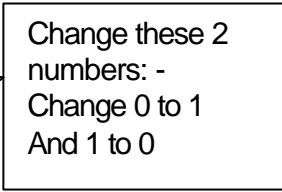
# mv /etc/lvmtab /etc/lvmtab.orig

# vgscan
```

Procedure 2

```
# ioscand -f | grep -e INTERFACE -e DEVICE | \
  grep -v target | \
  awk '{print $3, $1, $2}' > /infile
```

```
# vi /infile
8/0/0 tty 0
8/4 ext_bus 0
8/4.4.0 disk 0
8/4.5.0 disk 1
8/4.6.0 ctl 0
8/4.8.0 disk 2
8/4.14.0 disk 3
8/4.15.0 disk 4
8/8 ext_bus 1
```



Change these 2
numbers: -
Change 0 to 1
And 1 to 0

```
# mv /stand/ioconfig /stand/ioconfig.sav
```

```
# mv /etc/ioconfig /etc/ioconfig.sav
```

```
# shutdown -ry 0
```

```
# /sbin/iointit -c
```

```
# /sbin/iointit -f /infile -r
```

```
# mv /etc/lvmtab /etc/lvmtab.orig
```

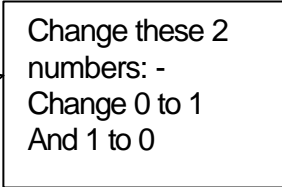
```
# vgscan
```


Procedure 3

```
# ioconfig2infile /etc/ioconfig >/infile
```

```
# vi /infile
```

```
8/0/0 tty 0  
8/4 ext_bus 0  
8/4.4.0 disk 0  
8/4.5.0 disk 1  
8/4.6.0 ctl 0  
8/4.8.0 disk 2  
8/4.14.0 disk 3  
8/4.15.0 disk 4  
8/8 ext_bus 1
```



Change these 2
numbers: -
Change 0 to 1
And 1 to 0

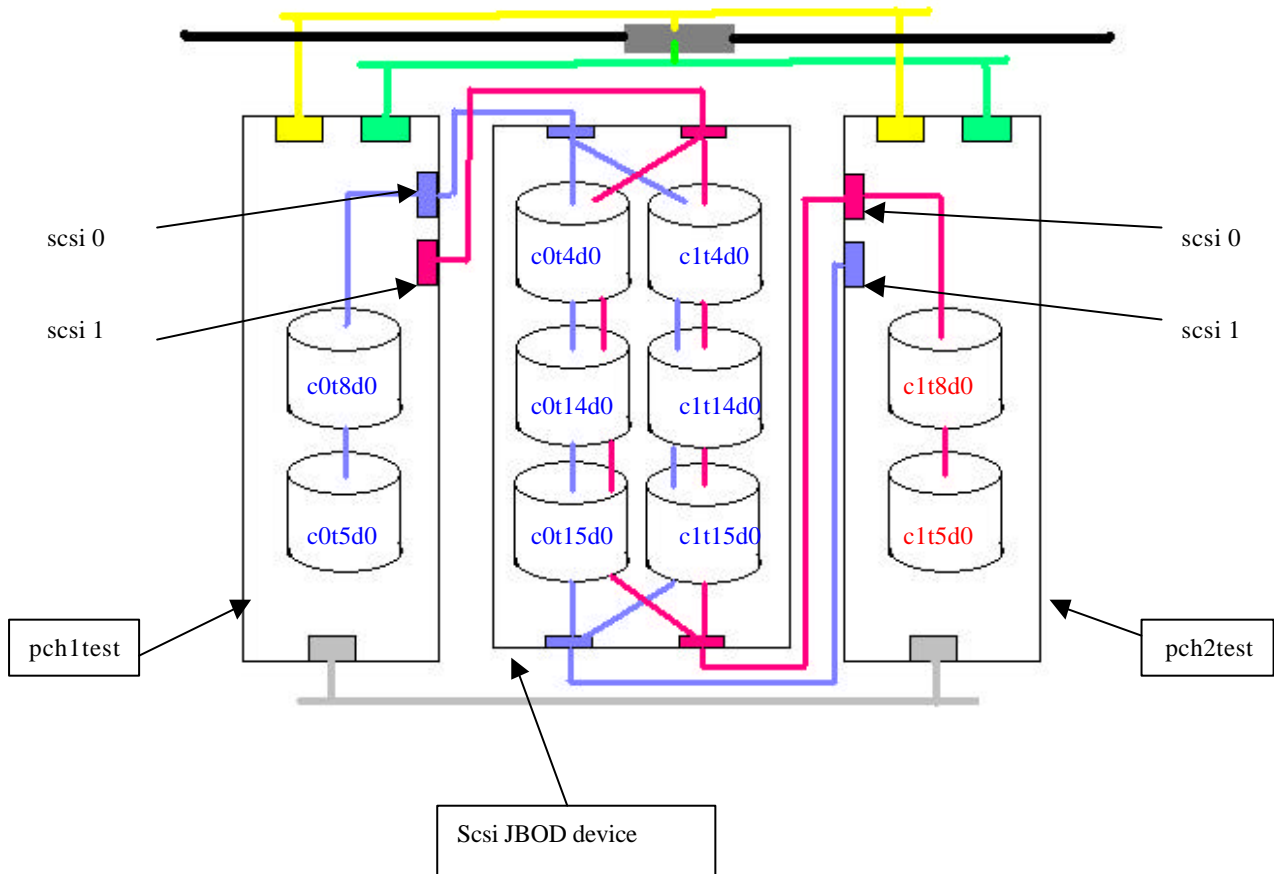
```
# init 1
```

```
# /sbin/iocinit -f /infile -r
```

```
# mv /etc/lvmtab /etc/lvmtab.orig
```

```
# vgscan
```

Assuming that one of the previous operations worked, you see the following layout: -



4) Some software requirements before you start.

Before installing Service Guard you need to set up a few things in order for the system to work effectively. You will need to mirror the boot disks on both machines, set up Network Time Protocol (NTP), and setup Domain Name Service (DNS).

Configure NTP

To configure NTP follow these steps on both machines: -

```
# vi /etc/ntp.conf
    server 127.127.1.1
    peer 192.168.1.2      (IP address of other node)
# vi /etc/rc.config.d/netdaemons
    export NTPDATE_SERVER="192.168.1.2"
    export XNTPD=1
# /sbin/init.d/xntpd start
```

To check that NTP is running, use the following command: -

```
# ntpq -p
```

(NOTE: - you will need to wait for about 5 minutes for the servers to sync)

The output should be something like this: -

```
remote      refid      st t when poll reach  delay  offset  disp
-----
LOCAL(1)   LOCAL(1)   3 l  54   64   377   0.00   0.000  10.01
*pch2test  LOCAL(1)   4 u  545 1024  377   1.17   -2.045  4.58
```

Configure DNS

To configure DNS follow these steps on both machines: -

```
# vi /etc/resolv.conf
    domain gb.tntpost.com
    nameserver 164.39.9.19
    nameserver 164.39.176.10

# vi /etc/nsswitch.conf
    hosts:          files [NOTFOUND=continue] dns
```

To check that DNS is running, use the following command: -

```
# nslookup gb898_m
```

Mirroring Boot disks on both machines

The following method needs to be applied to both machines. (Watch out for the controllers ID's, as they may be different for machine pch2test).

```
# pvcreate -f -B /dev/rdisk/c0t5d0

# vgextend vg00 /dev/dsk/c0t5d0

# mkboot -c -v /dev/dsk/c0t5d0

# mkboot -l /dev/dsk/c0t5d0

# lifls /dev/dsk/c0t5d0

# lvinboot -v vg00

# for I in `vgdisplay -v vg00|grep -I "lv name"| awk '{ print $NF }'`
> do
> lvextend -m 1 $I
> done

# lvinboot -v vg00

# lifcp /dev/dsk/c0t5d0:AUTO -

# lifcp /dev/dsk/c0c8d0:AUTO -

# for I in `lifls /dev/dsk/c0t8d0`
> do
> lifcp /dev/dsk/c0t8d0:$I /dev/dsk/c0t5d0:$I
> done

# mkboot -a "hpux -lq" /dev/rdisk/c0t5d0

# mkboot -a "hpux -lq" /dev/rdisk/c0t8d0

# setboot -a 0/0/2/1.2.0
```

5) Ready to install Service Guard.

You should now be in a position to install Service Guard from a depot or CD. After installing, you will need to add on some patches. Please check for the current patches at www.itrc.hp.com.

After installing Service Guard, check that all the software and patches are installed correctly using the following command: -

```
# swlist -l fileset -a state |grep -i service
```

You will now need to make a couple of file edits to make sure that Service Guard call see all the nodes in the cluster. These edits should be done on both nodes.

```
# vi /etc/hosts
    164.39.8.92 pch1test
    164.39.8.93 pch2test

# vi /etc/cmcluster/cmclnodelist
    pch1test root
    pch2test root

# vi /.rhosts
    pch1test root
    pch2test root
```

NOTE:- The entry in /.rhosts should be removed after the configuration of the cluster. It's just there to make copying files around the cluster easy.

6) Creating a cluster

Before you configure the cluster, you will need to create a volume group on the shared disks, so that Service Guard can create a “Cluster Lock Disk”.

On pch1test

```
# pvcreate /dev/rdisk/c0t4d0
# pvcreate /dev/rdisk/c1t4d0
# mkdir /dev/vg01
# mknod /dev/vg01/group c 64 0x010000
# vgcreate vg01 /dev/dsk/c0t4d0 /dev/dsk/c1t4d0
# mkdir /etc/cmcluster/maps
# vexport -p -s -m vg01 /etc/cmcluster/maps/vg01
```

On pch2test

```
# mkdir /dev/vg01
# mknod /dev/vg01/group c 64 0x010000
# mkdir /etc/cmcluster/maps
# rcp pch1test:/etc/cmcluster/maps/vg01 /etc/cmcluster/maps/vg01
# vgimport -s -m /etc/cmcluster/maps/vg01 vg01
# vgchange -a y vg01
# vgchange -a n vg01
```

You can now go ahead and create the cluster.

```
# cd /etc/cmcluster
# cmquerycl -v -C cmclconfig.ascii -n pch1test -n pch2test
```

After creating this file, you will need to edit it to suit your cluster. You will need to alter the following lines: -

```
# vi cmclconfig.ascii
    CLUSTER_NAME          simeon1
    NODE_NAME             pchltest
    NETWORK_INTERFACE     lan0
    HEARTBEAT_IP          192.168.100.101
    NETWORK_INTERFACE     lan1
    HEARTBEAT_IP          164.39.8.92
    NETWORK_INTERFACE     lan2

    NODE_NAME             pch2test
    NETWORK_INTERFACE     lan0
    HEARTBEAT_IP          192.168.100.102
    NETWORK_INTERFACE     lan1
    HEARTBEAT_IP          164.39.8.93

    NETWORK_INTERFACE     lan2
    HEARTBEAT_INTERVAL   2000000
    NODE_TIMEOUT          5000000
    MAX_CONFIGURED_PACKAGES 10
```

To check that there are no mistakes in the file, use the check command.

```
# cmcheckconf -C /etc/cmcluster/cmclconfig.ascii
```

To apply the configuration file, use the apply command.

```
# cmapplyconf -C /etc/cmcluster/cmclconfig.ascii
```

You are now ready to startup the cluster.

On pch1test: -

```
# vgchange -a n vg01
# cmruncl -v
# cmviewcl -v
# vgchange -c y vg01
```

On Both nodes: -

```
# netstat -in
# vi /etc/rc.config.d/cmcluster
    AUTOSTART_CMCLD=1
```

On pch1test: -

```
# cmhaltcl -v
# vgchange -a n vg01
```


7) Creating a basic package.

Once the cluster is up and running, we can create a package to run in it. This package will run an "xclock" on a client PC, and mount a file system, using the IP Address 164.39.8.101

The first step is to make a logical volume.

On node1: -

```
# vgchange -a e vg01
# lvcreate -L 20 vg01
# lvextend -m 1 /dev/vg01/lvol1
# newfs -F vxfs /dev/vg01/rlvol1
# mkdir /pkg_1
# vi /etc/fstab
    #/dev/vg01/lvol1 /pkg_1 vxfs delaylog 0 2 # MC/SG
# vgchange -a n vg01
# vgexport -p -s -m vg01 /etc/cmcluster/maps/vg01
```

On Node 2: -

```
# vgexport vg01
# mkdir /dev/vg01
# mknod /dev/vg01/group c 64 0x010000
# vgimport -s -m /etc/cmlcluster/maps/vg01 vg01
# mkdir /pkg_1
# vi /etc/fstab
    #/dev/vg01/lvol1 /pkg_1 vxfs delaylog 0 2 # MC/SG
```

Now we can make the package.

On Node1: -

```
# cd /etc/cmcluster
# mkdir pkg_1
# cd pkg_1
# cmmakepkg -p pkg_1.conf
```

The file pkg_1.conf will need to be edited to suit.

```
# vi pkg_1.conf
PACKAGE_NAME      pkg_1
NODE_NAME         pchltest
NODE_NAME         pch2test
RUN_SCRIPT        /etc/cmcluster/pkg_1/pkg_1.cntl
RUN_SCRIPT_TIMEOUT NO_TIMEOUT
HALT_SCRIPT       /etc/cmcluster/pkg_1/pkg_1.cntl
HALT_SCRIPT_TIMEOUT NO_TIMEOUT
SERVICE_NAME     pkg_1_service
SUBNET            164.39.8.0
```

Add the IP address that will be used for the package in the hosts file.

```
# vi /etc/hosts
164.39.8.101    pkg_1    xclock
```

```
# cmmakepkg -s pkg_1.cntl
```

The file pkg_1.cntl will need to be edited to suit

```
# vi pkg_1.cntl
    VG[0]="vg01"
    LV[0]="/dev/vg01/lvol1"; FS[0]="/pkg_1"; FS_MOUNT_OPT[0]=" "
    IP[0]="164.39.8.101"
    SUBNET[0]="164.39.8.0"
    SERVICE_NAME[0]="pkg_1_service"
    SERVICE_CMD[0]/etc/cmcluster/pkg_1/service_pkg_1"
    #SERVICE_RESTART[0]=" "

function customer_defined_run_cmds
{
# ADD customer defined run commands.
if [ $(hostname) = "pch1test" ]
then COLOUR="blue"
else COLOUR="red"
fi
/usr/bin/X11/xclock -bg "$COLOUR" -update 1 -display \
164.39.11.232:0.0 &
test_return 51
}

function customer_defined_halt_cmds
{
# ADD customer defined halt commands.
if
ps -ef |grep -v grep |grep -q "xclock"
then
kill -9 $(ps -ef |grep -v grep |grep "xclock"|cut -c10-14)
fi
test_return 52
}

# vi pkg_1_service

#!/usr/bin/sh
#This is a script to monitor that xclock is running
while true
do
if
ps -ef |grep -v grep |grep -q "xclock"
then
sleep 5
else
echo "Package xclock failed at $(date) from node \
$(hostname) > /etc/cmcluster/pkg_1/pkg_1.cntl.log"
exit 99
fi
done

# chmod u+x pkg_1_service
```

On node2: -

```
# cd /etc/cmcluster
# rcp -r pch1test:/etc/cmcluster/pkg_1 .
```

On node 1: -

```
# cmcheckconf -P pkg_1.conf
# cmapplyconf -P pkg_1.conf
# cmmodpkg -e -n pch1test -n pch2test pkg_1
# cmviewcl -v -p pkg_1
# cmhaltpkg pkg_1
# cmviewcl -v|more
# tail /etc/cmcluster/pkg_1/pkg_1.log
# tail /var/adm/syslog/syslog.log
```

8) Creating an NFS package

Before you create the NFS package, you should load the following software: -

```
B5140BA          A.11.00          MC/ServiceGuard NFS Toolkit
```

This provides you with pre-written scripts to help you create an NFS package, using the IP Address 164.39.8.104

The first step is to create a filesystem on the shared disks that can be exported.

On node1: -

```
# pvcreate /dev/rdisk/c0t14d0
# pvcreate /dev/rdisk/c1t14d0
# mkdir /dev/vg02
# mknod /dev/vg02/group c 64 0x020000
# vgcreate vg02 /dev/dsk/c0t14d0 /dev/dsk/c1t14d0
# lvcreate -L 20 vg02
# lvextend -m 1 /dev/vg02/lvol1
# newfs -F vxfs /dev/vg02/rlvol1
# mkdir /nfs_test
# vi /etc/fstab
    #/dev/vg02/lvol1 /nfs_test vxfs delaylog 0 2 # MC/SG
# vgchange -a n vg02
# vgchange -c y vg02
# vexport -p -s -m /etc/cmcluster/maps/vg02 vg02
```

On node 2: -

```
# mkdir /dev/vg02
# mknod /dev/vg02/group c 64 0x020000
# vgimport -s -m /etc/cmcluster/maps/vg02 vg02
# mkdir /nfs_test
# vi /etc/fstab
    #/dev/vg02/lvol1 /nfs_test vxfs delaylog 0 2 # MC/SG
```

On both nodes, switch on the NFS daemons.

```
# vi /etc/rc.config.d/nfsconf
    NFS_CLIENT=1
    NFS_SERVER=1

# /sbin/init.d/nfs.server start

# /sbin/init.d/nfs.client start

# vi /etc/exports
    #/nfs_test

# vi /etc/hosts
    164.39.8.104    nfs_server
```

On node 1, copy and edit the pre-written scripts into place.

```
# cp /opt/cmcluster/nfs /etc/cmcluster/nfs

# cd /etc/cmcluster/nfs

# vi nfs.conf
    PACKAGE_NAME    nfs_server
    NODE_NAME       pchltest
    NODE_NAME       pch2test
    SUBNET          164.39.8.0

# vi nfs.cnt1
    VG[0]="vg02"
    LV[0]="/dev/vg02/lvol1"; FS[0]="/nfs_test"; FS_MOUNT_OPT[0]=" " \
XFS[0]=/nfs_test
    IP[0]="164.39.8.104"
    SUBNET[0]="164.39.8.0"
    SERVICE_NAME[0]="nfs.monitor"
    SERVICE_CMD[0]="/etc/cmcluster/nfs/nfs.mon"
```

On node 2: -

```
# cd /etc/cmcluster
# rcp -r pchltest:/etc/cmcluster/nfs .
```

On node 1, you will now need to update the cluster with the new volume group.

```
# vi /etc/cmcluster/cmclonfig.ascii
      VOLUME_GROUP          /dev/vg02
# cd /etc/cmcluster/nfs
# cmcheckconf -C cmclconfig.ascii
# cmapplyconf -C cmclconfig.ascii
# cmcheckconf -P nfs.conf
# cmapplyconf -P nfs.conf
# cmmmodpkg -e nfs_server
```

On node 1, edit the crossmount script. (This will mount the exported file system on the local node where the package is running.)

```
# vi nfs_xmnt
      SNFS[0]="nfs_server:/nfs_test"; CNFS[0]="/dog"
```

When the NFS package is up and running then you can run the xmnt script

```
# /etc/cmcluster/nfs/nfs_xmnt start
# /etc/cmcluster/nfs/nfs_xmnt stop
```

9) Creating an Oracle package

Before you create the Oracle package, you should load the following software: -

```
B5139BA          A.11.04      Enterprise Cluster Master Toolkit
```

This provides you with some pre-written scripts to help you create an Oracle package, using the IP Address 164.39.8.107

The first step is to create a few filesystems on the shared disks that can be mounted by both systems.

On node1: -

```
# pvcreate /dev/rdsk/c0t15d0
# pvcreate /dev/rdsk/clt15d0
# mkdir /dev/vg03
# mknod /dev/vg03/group c 64 0x030000
# vgcreate vg03 /dev/dsk/c0t15d0 /dev/dsk/clt15d0
# lvcreate -L 20 vg03 (create 4 logical volumes)
# lvextend -m 1 /dev/vg03/lvol1
# vextend -m 1 /dev/vg03/lvol2
# lvextend -m 1 /dev/vg03/lvol3
# newfs -F vxfs /dev/vg03/rlvol1
# newfs -F vxfs /dev/vg03/rlvol2
# newfs -F vxfs /dev/vg03/rlvol3
# mkdir /u01
# mkdir /database1
# mkdir /database2
# mkdir /database3
```



```
# vi /etc/fstab
/dev/vg03/lvol1 /u01 vxfs delaylog 0 2 # MC/SG
/dev/vg03/lvol2 /database1 vxfs delaylog 0 2 # MC/SG
/dev/vg03/lvol3 /database2 vxfs delaylog 0 2 # MC/SG
/dev/vg03/lvol4 /database3 vxfs delaylog 0 2 # MC/SG
```

You can now install Oracle as described in appendix 1.

```
# umount /u01; umount /database1; umount /database2; umount /database3

# vi /etc/fstab
#/dev/vg03/lvol1 /u01 vxfs delaylog 0 2 # MC/SG
#/dev/vg03/lvol2 /database1 vxfs delaylog 0 2 # MC/SG
#/dev/vg03/lvol3 /database2 vxfs delaylog 0 2 # MC/SG
#/dev/vg03/lvol4 /database3 vxfs delaylog 0 2 # MC/SG

# vgchange -a n vg03

# vgchange -c y vg03

# vexport -p -s -m /etc/cmcluster/maps/vg03 vg03
```

On node 2: -

```
# mkdir /dev/vg03

# mknod /dev/vg03/group c 64 0x030000

# vgimport -s -m /etc/cmcluster/maps/vg03 vg03

# mkdir /u01

# mkdir /database1

# mkdir /database2

# mkdir /database3

# vi /etc/fstab
#/dev/vg03/lvol1 /u01 vxfs delaylog 0 2 # MC/SG
#/dev/vg03/lvol2 /database1 vxfs delaylog 0 2 # MC/SG
#/dev/vg03/lvol3 /database2 vxfs delaylog 0 2 # MC/SG
#/dev/vg03/lvol4 /database3 vxfs delaylog 0 2 # MC/SG
```

On both nodes, add the IP address of the package into the hosts file.

```
# vi /etc/hosts
164.39.8.107 oracle_test
```

On node 1, you can now create the oracle package.

```
# mkdir /etc/cmcluster/oracle

# cd /etc/cmcluster/oracle

# cmmakepkg -p oracle.conf

# vi oracle.conf
PACKAGE_NAME nfs_server
NODE_NAME pch1test
NODE_NAME pch2test
RUN_SCRIPT /etc/cmcluster/oracle/oracle.cnt1
RUN_SCRIPT_TIMEOUT NO_TIMEOUT
HALT_SCRIPT /etc/cmcluster/oracle/oracle.cnt1
HALT_SCRIPT_TIMEOUT NO_TIMEOUT
SERVICE_NAME oracle_service
SUBNET 164.39.8.0

# cmmakepkg -s oracle.cnt1

# vi oracle.cnt1
VG[0]="vg03"
LV[0]="/dev/vg03/lvol1"; FS[0]="/u01"; FS_MOUNT_OPT[0]=" "
LV[1]="/dev/vg03/lvol2"; FS[1]="/database1"; FS_MOUNT_OPT[1]=" "
LV[2]="/dev/vg03/lvol3"; FS[2]="/database2"; FS_MOUNT_OPT[2]=" "
LV[3]="/dev/vg03/lvol4"; FS[3]="/database3"; FS_MOUNT_OPT[3]=" "
IP[0]="164.39.8.107"
SUBNET[0]="164.39.8.0"
SERVICE_NAME[0]="oracle_service"
SERVICE_CMD[0]="/etc/cmcluster/oracle/oracle.mon"
#SERVICE_RESTART[0]=" "

function customer_defined_run_cmds
{
# ADD customer defined run commands.
/etc/cmcluster/oracle/ORACLE.sh start
test_return 51
}

function customer_defined_halt_cmds
{
# ADD customer defined halt commands.
/etc/cmcluster/oracle/ORACLE.sh shutdown
test_return 52
}
```

```

# cp /opt/cmcluster/toolkit/oracle/ORACLE.sh \
    /etc/cmcluster/oracle/ORACLE.sh

# vi /etc/cmcluster/oracle/ORACLE.sh
ORA_7_3_X=no
ORA_8_0_X=yes
ORA_8_1_X=no
SID_NAME=test
ORACLE_HOME=/u01/app/oracle/product/8.0.5
SQLNET=no
NET8=yes
LISTENER_NAME=
LISTENER_PASS=
MONITOR_INTERVAL=60
PACKAGE_NAME=
TIME_OUT=30
set -A MONITOR_PROCESSES

HOST=`hostname`
DATE=`date`
PATH=${ORACLE_HOME}/bin:/sbin:/usr/bin:/usr/sbin:/etc:/bin
export ORACLE_SID=${SID_NAME}
export ORACLE_HOME

```

On node 2.

```

# cd /etc/cmcluster

# rcp -r pchltest:/etc/cmcluster/oracle .

```

On node 1, you will now need to edit the cluster to include the new volume group, and add in the oracle package.

```

# cd /etc/cmcluster

# vi /etc/cmcluster/cmclonfig.ascii
    VOLUME_GROUP          /dev/vg03

# cmcheckconf -C cmclconfig.ascii

# cmapplyconf -C cmclconfig.ascii

# vgchange -c y vg03

# cd /etc/cmcluster/oracle

# cmcheckconf -P oracle.conf

# cmapplyconf -P oracle.conf

# cmmodpkg -e oracle

```

10) Other administration stuff.

Starting and stopping a package.

If you want to stop a package, use the following commands: -

```
# cmviewcl -v -p <package name>
# cmahalt <package name>
# cmviewcl -v -p <package name>
```

If you want to start a package, then use the following commands: -

```
# cmviewcl -v -p <package name>
# cmmodpkg -e <package name>
OR
# cmrunpkg -n <host name> <package name>
# cmmodpkg -e <package name>
```

Deleting a package.

To delete a package, use the following commands: -

```
# cmdeleteconf -p <package name>
```

Stopping and starting the cluster.

To stop the cluster, use the following commands: -

```
# cmiewcl -v
# cmhaltcl
OR
# cmhaltcl -f
```

To start the cluster: -

```
# cmruncl -v
# cmviewcl -v
```

Stopping and starting a node in the cluster.

To stop a node in the cluster.

```
# cmhaltnode -n <node name>
# cmviewcl -v
# vi /etc/rc.config.d/cmcluster
    AUTOSTART_CMCLD=0
```

To start a node in the cluster.

```
# cmrunnode -n <name name>
# cmviewcl -v
# vi /etc/rc.config.d/cmcluster
    AUTOSTART_CMCLD=1
```

Deleting a cluster.

When there are no packages left in the cluster, and you want to delete the cluster, use the following commands: -

```
# vgchange -c n vg01
# cmhaltcl -f
# cmdeleteconf -c <cluster_name>
# vgchange -a y vg01
# lvremove /dev/vg01/lvol1
# vgrename vg01
# rm -r /dev/vg01
```

Other useful commands.

```
# cmgetconf -c <cluster_name> <cluster.ascii output_file>
# cmviewconf -o <output_file>
```

Appendix 1

Here is a simple method of how to install oracle 8.0.5 onto HPUX 11.00

Tune kernel or Oracle install.

```
shmmmax=          0x40000000      1073741824
dbc_max_pct=      10
dbc_min_pct=      5
fs_async=         0
maxfiles=         512
maxfiles_lim=    1024
maxurpc=          75
shmmni=           200
shmseg=           120
semms=            200
semni=            64
bufpages=         0
```

Create "dba" group

```
# groupadd dba
```

Create "oracle" user

```
# useradd -m oracle
```

Edit "oracle's" .profile

```
# vi /home/oracle/.profile
export ORACLE_HOME=/u01/app/oracle/product/8.0.5
export ORACLE_SID=test
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
export ORACLE_TERM=xterm
export PATH=$PATH:$ORACLE_HOME/bin
export ORACLE_BASE=/u01/app/oracle
```

As root, do the following: -

```
# chown -R oracle:dba /u01
# /usr/sbin/pfs_mountd &
# /usr/sbin/pfsd &
```

```
# vi /etc/pfs_fstab
    /dev/dsk/c2t2d0 /cdrom pfs-rrip xlat=unix 0 0

# pfs_mount /cdrom

# export ORACLE_OWNER=oracle

# /cdrom/orainst/oratab.sh
```

As oracle, do the following: -

```
# cd /cdrom/orainst

# ./orainst /m
    Select "RDBMS 8.0.5"
```

After install as oracle: -

```
# mkdir -p /project/805/src/rdbms
# ln -s /u01/app/oracle/product/8.0.5/lib \
/project/805/src/rdbms/lib

# /u01/app/oracle/product/8.0.5/network/admin/ listener.ora
    (ADDRESS= (PROTOCOL= IPC)(KEY= EXTPROC))
    (ADDRESS= (PROTOCOL= TCP)(Host= <ip address>)(Port= 1521)
    (GLOBAL_DBNAME= test.tnteww.com)
    (ORACLE_HOME= /u01/app/oracle/product/8.0.5)
    (SID_NAME = test)
```

Administration of databases: -

```
# $ORACLE_HOME/bin/lsnrctl
> start
> stop
> status
> quit

# $ORACLE_HOME/bin/svrmgrl
> connect internal ;
> startup ;
> shutdown immediate ;
> quit

# ps -ef |grep -i pmon

# ps -ef |grep -i list
```