

Easy HP-UX MC/ServiceGuard Active/Standby cluster configuration (revision 0)

Andrea Leofreddi

January 16, 2003

Contents

1 Prerequisites	2
2 Cluster configuration	3
3 Packages configuration and management	4

1 Prerequisites

Configure the network (in class example I've used *node1* (10.0.0.1) and *node2* (10.0.0.2) for cluster environment), and edit */etc/cmcluster/cmclnodelist* using fields in that format:

```
<hostname or ip>                username
```

That will ensure rights to distribute over all the cluster configuration files. Then install *MC/ServiceGuard* and *MirrorDisk/UX* packages using the HP-UX package manager graphical frontend:

```
swinstall
```

Ensure that it has been installed correctly typing

```
swlist | grep -i service
```

Now we're ready to configure cluster LVMs. Each node in the cluster will have its own root device, but both will share a I/O bus to allow package eventually migrate, so let's assume that we have other unused pair of disk shared between nodes, let's call them */dev/dsk/c2t0d0* and */dev/dsk/c3t0d0*. We'll configure software mirroring using */dev/dsk/c2t0d0* as main package data disk and */dev/dsk/c3t0d1* as mirror. Let's create a *virtual group* to realize that:

```
pvcreate -f /dev/rdisk/c2t0d0
pvcreate -f /dev/rdisk/c3t0d0
```

Then let's create a directory to handle the virtual group, and link there our disks:

```
mkdir /dev/vgdata
mknod /dev/vgdata c 64 0xaa0000
vgcreate -g bus0 /dev/vgdata /dev/dsk/c2t0d0
vgcreate -g bus1 /dev/vgdata /dev/dsk/c3t0d0
lvcreate -L 500 -m 1 -s g /dev/vgdata
```

This should have created a logical volume in *vgdata*. Now we need to build a filesystem there:

```
newfs -F vxfs /dev/vgdata/rlvol1
mkdir /mnt/data
mount /dev/vgdata/lvol1 /mnt/data
```

Now we have correctly configured and mounted a mirrored logical volume. Let's export its map (to do that we need to unmount it).

```
umount /mnt/data
vgchange -a n /dev/vgdata
vgexport -p -s -m /tmp/vgdata.map /dev/vgdata
```

Now we can just send that map to other node, using the *rcp* command or its secure SSL version, *scp*, and rebuilding there the same vg configuration:

```
mkdir /dev/vgdata
mknod /dev/vgdata c 64 0xaa0000
```

Then we can import its configuration from vgmap, using the *vgimport* command.

```
vgimport -s -m /tmp/vgdata.map /dev/vgdata
```

Similarly, configure a lock logical volume/virtual group.

2 Cluster configuration

So let's start cluster configuration. We can use *cmquerycl* command to build a template configuration file for node1 and node2 typing:

```
cmquerycl -v -C /etc/cmcluster/cluster.config -n node1 -n node2
```

Configure the template to suits your needs. Note that *FIRST_CLUSTER_LOCK_VG* and *FIRST_CLUSTER_LOCK_PV* are the volume group and the physical volume chosen as cluster lock, *cmquerycl* will try to autoprobe the best choice for your configuration. Define a *NETWORK_INTERFACE* and *HEARTBEAT_IP* for each node, then define heartbeat timings, then define which vg are shared in the cluster with the *VOLUME_GROUP* directive (in our example, */dev/vgdata*).

Done configuration stuff, we can check if our configuration seems to be correct using *cmcheckconf* command.

```
cmcheckconf -k -v -C /etc/cmcluster/cluster.config
```

If checkconf didn't report errors, we can spread our configuration to the other node:

```
vgchange -a y /dev/vglock
cmapplyconf -k -v -C /etc/cmcluster/cluster.config
vgchange -a n /dev/vglock
```

Now cluster should be up and running, we can check it out using MC/ServiceGuard Manager (using windows client or HP-UX version, *sgmgr*).

3 Packages configuration and management

Having correctly configured and running cluster is useless without packages, so we need to spread across some to get them runned in the High Availability environment that we had just setup. Let's create a directory for pkg, and let's generate there pkg configuration a script:

```
mkdir /etc/cmcluster/pkg
cmmakepkg -p /etc/cmcluster/pkg/pkg.config
cmmakepkg -s /etc/cmcluster/pkg/pkg.sh
```

Now we can configure the package template, and the script (set VG array to be used and mount parameters). Carefully follow those important steps:

1. Correctly setup VGs and their mountpoints
2. Apply the configuration
3. Distribute control script to the other node
4. Run the package, and check that it can be moved from other node
5. Halt the package
6. Configure package IP
7. Apply the configuration and distribute control script to the other node
8. Run the package, and control that it works as expected

We can use *cmcheckconf* command to verify the package script:

```
cmcheckconf -k -v -P /etc/cmcluster/pkg/pkg.config
```

To distribute package script (as for cluster configuration) we can use *rcp*.

To distribute configuration file, let's use *cmcheckconf* command:

```
cmcheckconf -C /etc/cmcluster/pkg/pkg.config -P /etc/cmcluster/cluster.config
vgchange -a y /dev/vglock
cmapplyconf -v -C /etc/cmcluster/pkg/pkg.config -P /etc/cmcluster/cluster.config
vgchange -a n /dev/vglock
```

Now our package is correctly configured and distributed. We can check its status using the *cmviewcl* command:

```
cmviewcl -v
```

Of course we can start or move it, using these commands:

```
cmrunpkg -n <node> pkg
```

We can deactivate a node using the *cmhaltnode* command:

```
cmhaltnode <node>
```