P A R T I C I P A N T G U I D E

Sun Cluster 3.0 Lab Exercises

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303
U.S.A.

Part No.: TBD
Course No.: TBD
Revision 1.2, September 2000

LAB 1

Installing and Configuring Sun Cluster 3.0 with Solstice DiskSuite

Preinstallation

1. Install the cluster interconnect and storage cables. Verify that a supported topology is being used.

   Which topology does your cluster use? _____

2. Complete the worksheet on the following page.

Cluster Interconnect Worksheet

Adapters **Cabling** **Junctions**

Draw lines between cable endpoints

Node name _____

Adapter Name	Transport Type
	DLPI
	DLPI

Node name _____

Adapter Name	Transport Type
	DLPI
	DLPI

Node name _____

Adapter Name	Transport Type
	DLPI
	DLPI

Node name _____

Adapter Name	Transport Type
	DLPI
	DLPI

Junction name _____

Port Number	Description (optional)

Junction name _____

Port Number	Description (optional)

Install the Administrative Workstation

From the administrative workstation:

3. Locate the `Sun_Cluster_3.0/Packages` directory of the Sun Cluster CD-ROM image. Add the `SUNWccon` package:

```
# cd <Sun Cluster 3.0 CDROM Image>/Sun_Cluster_3.0/Packages
# pkgadd -d . SUNWccon
```

4. Make sure there is a valid entry for each cluster node as well as the terminal concentrator in the `/etc/hosts` file.
5. Create an `/etc/clusters` file with an entry for your cluster. The entry should have the following format:

```
<Name of Cluster> <Node1 Name> <Node2 Name> ...
```

6. Check the cabling of the Terminal Concentrator. Complete the following table:

Port on TC	Node Name
2	
3	
4	
5	

7. Use the table completed above to create an `/etc/serialports` file. The format of the file is (there should be one line per node):

```
<Node Name> <Terminal Concentrator Hostname> 5000 + <PortNumber>
```

8. Start up the Cluster Control Panel and double click on the `cconsole` icon. Make sure that you can communicate with each node of the cluster via the cluster console. The rest of the node installation should be done using the cluster console.

```
# /opt/SUNWcluster/bin/ccp <Name of cluster> &
```

Set up root's environment

On all nodes of the cluster:

9. Modify root's /.profile file for the Sun Cluster environment:
   a. Add /usr/cluster/bin to root's PATH
   b. Add /usr/cluster/man to root's MANPATH
10. Add the root user to group 14 (sysadmin)
11. Verify that the root disk has the proper slicing (100MB /globaldevices partition and a partition available for the local metadevice state databases)
12. **Install any required OS patches. See the instructor for a list of required patches for Sun Cluster 3.0.**

Install Sun Cluster 3.0 on each node

On only Node 1 of the cluster:

13. Locate the Sun_Cluster_3.0/Tools directory of the Sun Cluster 3.0 CD-ROM image. Run the scinstall program and perform an interactive installation. Refer to the configuration worksheets you completed earlier to answer the prompts presented by scinstall.

```
# cd <Sun Cluster 3.0 CDROM Image>/Sun_Cluster_3.0/Tools
# ./scinstall
```

14. Repeat step 11 for each remaining node in the cluster.

Complete the Cluster Initialization

On just one node of the cluster:

15. Based on your topology, determine which disks you want to use as your quorum device(s). Use `scdidadm -L` to determine the global device name (d#) for each disk you want to use as quorum disk. Note down the quorum device(s) in the following table:

Global Disk Name (d#)
(If Needed)

16. Run scstat -q. What are the vote counts currently assigned to each node. How many votes are required?

_____

17. Run `scsetup` to complete the cluster initialization and configure the quorum devices:

```
# scsetup
```

18. Use `scstat` to verify that each node has 1 quorum vote and the quorum devices have been properly configured:

```
# scstat -q
```

Install Solstice DiskSuite

On all nodes of the cluster:

19. Locate the CD-ROM image for Solstice DiskSuite 4.2.1 on the Solaris 8 CD. Change to the <*path of Solaris 8 CD*>/Solaris_8/EA/products/DiskSuite_4.2.1/sparc/packages directory of the CD-ROM image. Run the `pkgadd` program to install the Solstice DiskSuite packages. The required packages are SUNWmdr and SUNWmdu. If you are using a 64-bit kernel, SUNWmdx is also required.
20. Use `patchadd` to Install any required DiskSuite patches
21. For this lab, the `/kernel/drv/md.conf` will **not** need to be modified.
22. Shutdown and reboot all the nodes by executing the `scshutdown` command on *one* node, and then executing a normal boot on all nodes of the cluster.

```
On just one of the cluster nodes (all nodes will automatically shutdown):
# scshutdown -g 0 -y
...
On all cluster nodes:
ok boot
```

23. Once all nodes have rebooted, run the `scgdevs` command on **one** of the cluster nodes. This will ensure that the cluster can access the DiskSuite devices properly.

Install the Sun Cluster data service software

On each node of the cluster:

24. Install the software required for supporting NFS, DNS, LDAP, Apache, Oracle8i and Netscape Enterprise Server on the cluster. Run `scinstall` (located in `/usr/cluster/bin`) and choose option 4 from the main menu:

```
# cd /usr/cluster/bin
# ./scinstall
```

Configure Solstice DiskSuite

On each node of the cluster:

25. Create the local metadevice state databases. There should be a 10MB slice on your root disk set aside for the metadevice state databases. If not, reduce the size of swap by 2-10MB and create a new partition on the root disk for the metadevice state databases (make sure to disable swap first by using the "swap -d <Swap Slice>" command -- don't forget to re-enable it!)

```
# metadb -a -f -c 3 <MetadbSlice>

Where
<MetadbSlice> is the slice to use for the Metadevice state databases
(e.g. c0t0d0s7)
```

26. We will be creating 2 disksets, one which we can use later for NFS data and one for web server data. For the purposes of this lab, each diskset will require just 2 disks each. Make sure that the disks used for each diskset are in separate arrays. Determine which disks will be used in each diskset and complete the following table (use the `scdidadm -l` command to determine the DID -> Physical disk device path mappings.

Diskset 1 Name: `nfs-ds1`:

Disk	DID Disk Name (d#)	Physical Disk (c#t#d#)
1		
2		

Diskset 2 Name: `webserver-ds1`:

Disk	DID Disk Name (d#)	Physical Disk (c#t#d#)
1		
2		

On just one of the nodes of the cluster[1]:

27. Create the disksets. Based on the disks assigned to each diskset (in step 26), determine which nodes in the cluster are eligible to master the disksets (based on the actual physical connectivity to the disks in the diskset). Use the `metaset` command to create each diskset:

```
# metaset -s nfs-ds1 -a -h <List of nodes (space separated)>
# metaset -s webserver-ds1 -a -h <List of nodes (space separated)>
```

Verify the creation of the disksets by running the `metaset` command without any arguments. Both disksets should be listed.

28. Add the appropriate disks to each diskset. Refer to step 26 for the list of disks to add to each diskset:

```
# metaset -s nfs-ds1 -a /dev/did/rdsk/<diddisk1>          \
                        /dev/did/rdsk/<diddisk2>
# metaset -s webserver-ds1 -a /dev/did/rdsk/<diddisk1>    \
                             /dev/did/rdsk/<diddisk2>
Where
<diddisk1> - First DID disk device for the diskset (from step 26)
<diddisk2> - Second DID disk device for the diskset (from step 26)
```

Use the `metaset -s <DiskSetName>` command to verify the contents of each diskset.

29. Repartition each disk in the diskset. Make sure to leave slice 7 alone (slice 7 holds the state database replicas for the diskset).

    a. Use `format` to partition the first disk in the `nfs-ds1` diskset as follows (you will need to know the actual physical disk name (i.e., c#t#d#) for this disk, refer to step 26):

Slice	Size
7	Do not change
6	1% of disk or 64MB, whichever is smaller
0	Rest of the disk
2	Entire Disk (except slice 7)

1. This step should be done on one of the nodes that is physically connected to the disks you intend to use in the disksets

b. Assuming that the other 3 disks used in the disksets (the other disk in the nfs-ds1 diskset and the 2 disks in the webserver-ds1 diskset) are of the same geometry, the fmthard command can be used to apply the identical formatting set up in step 25a. Run the following command for the remaining disks to be partitioned:

```
# prtvtoc /dev/did/rdsk/<SlicedDIDdisk>s2 | fmthard -s - \
          /dev/did/rdsk/<DIDdisktoSlice>s2
Where
  <SlicedDIDdisk> - is the DID disk sliced in step 29a
  <DIDdisktoSlice> - is the disk you want to apply the
                     formatting to

Example:
To apply the formatting for DID disk d5 to DID disk d7:
# prtvtoc /dev/did/rdsk/d5s2 | fmthard -s - /dev/did/rdsk/d7s2
```

c. If the remaining disks in the disksets are not of the same geometry, the format command must be used to slice the disks accordingly.

30. Create the metadevices to be used in each diskset. We will need a single metatrans device in each diskset as the top-level metadevice.

a. Create an md.tab file in /etc/lvm/md.tab. Use the following as a template, (substitute the appropriate disk device names for <FirstDIDDisk> and <SecondDIDDisk> in each diskset):

```
# Sample md.tab file for training LAB 1
# nfs-ds1 diskset
nfs-ds1/d10  -t   nfs-ds1/d11 nfs-ds1/d12
  nfs-ds1/d11 -m nfs-ds1/d13 nfs-ds1/d14
    nfs-ds1/d13 1 1 /dev/did/rdsk/<FirstDIDDisk>s0
    nfs-ds1/d14 1 1 /dev/did/rdsk/<SecondDIDDisk>s0
  nfs-ds1/d12 -m nfs-ds1/d15 nfs-ds1/d16
    nfs-ds1/d15 1 1 /dev/did/rdsk/<FirstDIDDisk>s6
    nfs-ds1/d16 1 1 /dev/did/rdsk/<SecondDIDDisk>s6

# webserver-ds1 diskset
webserver-ds1/d10 -t webserver-ds1/d11 webserver-ds1/d12
  webserver-ds1/d11 -m webserver-ds1/d13 webserver-ds1/d14
    webserver-ds1/d13 1 1 /dev/did/rdsk/<FirstDIDDisk>s0
    webserver-ds1/d14 1 1 /dev/did/rdsk/<SecondDIDDisk>s0
  webserver-ds1/d12 -m webserver-ds1/d15 webserver-ds1/d16
    webserver-ds1/d15 1 1 /dev/did/rdsk/<FirstDIDDisk>s6
    webserver-ds1/d16 1 1 /dev/did/rdsk/<SecondDIDDisk>s6
```

b. Use the `md.tab` file to create the disksets using the `metainit` command (There will be numerous warning messages concerning the creation of the metamirrors, these can be safely ignored):

```
# metainit -s nfs-ds1 -a
... <Numerous messages/warnings from metainit>
# metainit -s webserver-ds1 -a
... <Numerous messages/warnings from metainit>
```

c. Verify that all metadevices have been created properly by running the `metastat` command for each diskset:

```
# metastat -s nfs-ds1
...<Status of each metadevice for the nfs-ds1 diskset>
# metastat -s webserver-ds1
...<Status of each metadevice for the webserver-ds1 diskset>
```

31. Create file systems on the metatrans devices for each diskset:

```
# newfs /dev/md/nfs-ds1/rdsk/d10
... <newfs output>
# newfs /dev/md/webserver-ds1/rdsk/d10
... <newfs output>
```

On each node of the cluster:

32. Create a mount point for the file systems created in step 27:

```
# mkdir /global/nfs
# mkdir /global/webdata
```

On each node of the cluster that is able to master the diskset:

33. Add the new file systems to the `/etc/vfstab` file. Specify that the files be mounted globally. The new entries should look like this:

```
/dev/md/nfs-ds1/dsk/d10   /dev/md/nfs-ds1/rdsk/d10   /global/nfs  ufs 2 yes global
/dev/md/webserver-ds1/dsk/d10   /dev/md/webserver-ds1/rdsk/d10   /global/webdata  ufs 2 yes global
```

On just one of the nodes that is able to master the diskset:

34. Mount the file systems:

```
# mount /global/nfs
# mount /global/webdata
```

On each node of the cluster:

35. Verify that the `/global/nfs` and `/global/webdata` file systems are available:

```
# mount
...<Mount output>
/global/nfs on /dev/md/nfs-ds1/d10 ...
...
/global/webdata on /dev/md/webserver-ds1/d10 ...
...
```

36. Configure dual string mediators. On one node of the cluster, use the `metaset` command to configure the mediators for each diskset:

```
# metaset -s nfs-ds1 -a -m <Space Separated List of Nodes>
# metaset -s webserver-ds1 -a -m <Space Separated List of Nodes>
```

37. Check the status of the mediators using the `medstat` command:

```
# medstat -s nfs-ds1
# medstat -s webserver-ds1
```

Configure PNM

On each node of the cluster:

38. Use the `pnmset` command to configure the public network interfaces for each node.
39. Verify the state of the NAFO groups on each node by using the `pnmstat -l` command

Configure `ntp.conf`

On each node of the cluster:

40. Edit the `/etc/inet/ntp.conf` file. Delete the following for nodes that do not exist. For example, in a four node cluster, the following entries would need to be deleted:

```
peer clusternode5-priv
peer clusternode6-priv
peer clusternode7-priv
peer clusternode8-priv
```

Verify the cluster installation

41. Run the scconf -p and scstat -p commands to display the current configuration and state of the cluster:

```
# scconf -p
...<Output from scconf command> ...
# scstat -p
...<Output from scstat command> ...
```

Things to try

42. Now that we have a working cluster, let's try a few things to determine how the cluster reacts to various events:

    a. Try "crashing" a node, how do the other node's in the cluster react?

    b. Pull out both cluster interconnect paths from a node. Did the failure fencing work properly?

    Without replacing the interconnect path cables, reboot the node and see if it can rejoin the cluster. What happens?

    Replace the interconnect cables and have the node rejoin the cluster.

    c. Simulate an "amnesia" condition. Shut down all nodes of the cluster, one at a time. Try restarting the first node you shut down. Does it boot completely? What needs to happen before it can boot completely?

    d. Verify that all nodes can read from and write to the global file systems (`/global/nfs and /global/webdata`). Determine which node is currently mastering the disksets (`nfs-ds1 and webserver-ds1`) using the `metaset` command (`metaset -s <DiskSetName>` on **all** nodes) or the `scstat -D` command. What happens when we shut down or crash the node mastering the diskset(s)? Does the global filesystem remain available?

LAB 2

Installing and Configuring Sun Cluster 3.0 with Veritas Volume Manager

Preinstallation

1. Install the cluster interconnect and storage cables. Verify that a supported topology is being used.

   Which topology does your cluster use? _____

2. Complete the configuration worksheets on the following pages before installing the cluster.

Cluster Interconnect Worksheet

Adapters **Cabling** **Junctions**

Draw lines between cable endpoints

Node name _____

Adapter Name	Transport Type
	DLPI
	DLPI

Node name _____

Adapter Name	Transport Type
	DLPI
	DLPI

Node name _____

Adapter Name	Transport Type
	DLPI
	DLPI

Node name _____

Adapter Name	Transport Type
	DLPI
	DLPI

Junction name _____

Port Number	Description (optional)

Junction name _____

Port Number	Description (optional)

Install the Administrative Workstation

From the administrative workstation:

3. Locate the Sun_Cluster_3.0/Packages directory of the Sun Cluster CD-ROM image. Add the SUNWsccon package:

```
# cd <Sun Cluster 3.0 CDROM Image>/Sun_Cluster_3.0/Packages
# pkgadd -d . SUNWsccon
```

4. Make sure there is a valid entry for each cluster node as well as the terminal concentrator in the `/etc/hosts` file.
5. Create an `/etc/clusters` file with an entry for your cluster. The entry should have the following format:

```
<Name of Cluster> <Node1 Name> <Node2 Name> ...
```

6. Check the cabling of the Terminal Concentrator. Complete the following table:

Port on TC	Node Name
2	
3	
4	
5	

7. Use the table completed above to create an `/etc/serialports` file. The format of the file is (there should be one line per node):

```
<Node Name> <Terminal Concentrator Hostname> 500<PortNumber>
```

8. Start up the Cluster Control Panel and double click on the cconsole icon. Make sure that you can communicate with each node of the cluster via the cluster console. The rest of the node installation should be done using the cluster console.

```
# ccp <Name of cluster> &
```

Set up root's environment

On all nodes of the cluster:

9. Modify root's `/.profile` file for the Sun Cluster environment:
   a. Add `/usr/cluster/bin` to root's PATH
   b. Add `/usr/cluster/man` and `/opt/VRTSvxvm/man` to root's MANPATH
   c. Set your terminal type to `vt220`
10. Verify that the root disk has the proper slicing, with a 100MB partition available for the global device namespace
11. Install any required OS patches. See the instructor for a list of required patches for Sun Cluster 3.0.

Install Sun Cluster 3.0 on each node

On Node 1 of the cluster:

12. Locate the `Sun_Cluster_3.0/Tools` directory of the Sun Cluster 3.0 CD-ROM image. Run the `scinstall` program and perform an interactive installation. Refer to the configuration worksheets you completed earlier to answer the prompts presented by scinstall.

```
# cd <Sun Cluster 3.0 CDROM Image>/Sun_Cluster_3.0/Tools
# ./scinstall
```

13. Wait for the node to complete its reboot before continuing.
14. Repeat steps 12 and 13 on the remaining nodes of the cluster.

Complete the Cluster Initialization

On just one node of the cluster:

15. Based on your topology, determine which disks you want to use as your quorum device(s). Use `scdidadm -L` to determine the global device name (d#) for each disk you want to use as quorum disk. Note down the quorum device(s) in the following table:

Global Disk Name (d#)
(If needed)

16. Run `scstat -q` on one node of the cluster. What are the vote counts assigned to each node? _____

17. Run `scsetup` to complete the cluster initialization and configure the quorum devices:

```
# scsetup
```

18. Use `scstat` to verify that each node has 1 quorum vote and the quorum devices have been properly configured:

```
# scstat -q
```

Install Veritas Volume Manager

On all nodes of the cluster:

19. Disable Dynamic Multipathing (DMP):

```
# mkdir /dev/vx
# ln -s /dev/dsk /dev/vx/dmp
# ln -s /dev/rdsk /dev/vx/rdmp
```

20. Use the `format(1M)` command to check that each disk which will be used by VxVM has a slice 2 which encompasses the **entire** disk. Adjust any disks as necessary.
21. Locate the CD-ROM image for Veritas Volume Manager. Change to the `pkgs` directory of the CD-ROM image. Use `pkgadd` to install the Veritas Volume Manager packages:

```
# cd <Location of VxVM CD-ROM Image>/pkgs
# pkgadd -d . VRTSvxvm VRTSvmdev VRTSvmman
```

22. Add any required VxVM patches using patchadd.

Install the Sun Cluster data service software

On each node of the cluster:

23. Install the software required for supporting NFS, DNS, LDAP, Apache, Oracle8 and Netscape Enterprise Server on the cluster. Run `scinstall` and choose option 4 from the main menu. You be asked to provide the location of the Sun Cluster Data Services CD-ROM image:

```
# cd /usr/cluster/bin/
# ./scinstall
```

Configure Veritas Volume Manager

On each node of the cluster:

24. Verify that the major number used for the vxio device is identical on all nodes of the cluster. Check the `/etc/name_to_major` file on each node of the cluster to verify that the entry for `vxio` is identical on each node. If they are not identical, choose a

number which can be used on each node (make sure not to choose a number that is already in use on any of the nodes) and edit the `/etc/name_to_major` files accordingly.

25. If your cluster does not have an SSA or A5x00 storage array, license VxVM using the `vxlicense` utility. Your instructor will provide you with current evaluation license keys:.

```
# vxlicense -c
Please enter your key:
```

26. Initialize the rootdg disk group. Choose whether you want to encapsulate the root disk, encapsulate a non-root disk.

If encapsulating the root disk:

a. On each node of the cluster, run `vxinstall`. At the main menu choose "Custom Install". When given the option, choose to encapsulate the boot disk. Choose unique root disk names on each node. When `vxinstall` presents with the list of disks on each controller, choose the "Leave these disks alone" option. Do NOT have `vxinstall` automatically reboot the nodes.

b. Modify the entry on `/etc/vfstab` for the `/global/.devices/node@X` file system. Change the entry to use the actual c#t#d#s# device instead of the DID device (find the commented out entry for `/globaldevices`, this entry will have the proper device to used for the `/global/.devices/node@X` entry)

c. Use the `scshutdown` command to shut down the entire cluster. After the nodes have been halted, boot each node into non-clustered mode using the -x argument to the boot command. The nodes will reboot themselves again into the cluster as part of the encapsulation process.

```
# scshutdown -g 0 -y
...<Shutdown messages>...
ok boot -x
```

d. The boot process will be interurrupted by an `fsck` error (on the `/global/.devices/node@X` filesystem) on some of the cluster nodes. This is normal, just hit `Control-D` to continue the boot.

e. One of the nodes will have successfully mounted its `/global/.devices/node@X` filesystem. Use the `mount` command to determine which `/global/.devices/node@X` file system was mounted and then unmount that single file system.

f. Reminor the devices in rootdg to be unique on each node. Set each node to have a base minor number which is 100*NodeID (e.g. 100 on node 1, 200 on node 2, etc.)

```
# vxdg reminor 100   (On Node 1)

# vxdg reminor 200   (On Node 2)
...
```

g. If there was a separate /usr partition, you must manually reminor the usr volume. Manually remove the device entries in /dev/vx for the usr volume. Obtain the minor number to be used for usr by running vxprint. Manually create the device nodes for the usr volume by using mknod.

```
# rm /dev/vx/dsk/usr /dev/vx/dsk/rootdg/usr
# rm /dev/vx/rdsk/usr /dev/vx/rdsk/rootdg/usr
# vxprint -l -v usr
Disk group: rootdg

Volume:   usr
info:     len=1024480
type:     usetype=fsgen
state:    state=ACTIVE kernel=ENABLED cdsrecovery=0/0 (clean)
assoc:    plexes=usr-01
policies: read=ROUND exceptions=GEN_DET_SPARSE
flags:    open writeback
logging:  type=NONE
apprecov: seqno=0
recov_id=0
device:   minor=103 bdev=65/103 cdev=65/103 path=/dev/vx/dsk/rootdg/usr
perms:    user=root group=root mode=0600
# grep vxio | /etc/name_to_major
vxio 65
# mknod /dev/vx/dsk/usr b 65 103
# mknod /dev/vx/dsk/rootdg/usr b 65 103
# mknod /dev/vx/rdsk/usr c 65 103
# mknod /dev/vx/rdsk/rootdg/usr c 65 103
```

h. Check the /etc/vfstab files on each node to see if the volume name for the /global/.devices/node@X entry is unique on each cluster node. (**If you specified unique volume names in step a, you will not need to do this**) If they are not (for example, each node has the volume name rootdisk4vol), each volume needs to be renamed on the nodes of the cluster. Rename the

volume on the first node by appending a ‘1” to the current volume name, on the second node, rename the volume by appending a “2” to the volume name and so on, for all the nodes of the cluster.

```
# vxedit rename rootdisk4vol rootdisk4vol1    (On Node 1)

# vxedit rename rootdisk4vol rootdisk4vol2    (On Node 2)
...
```

i. If you used vxedit to reame the `/global/.devices/node@X` volume in step h, make sure to edit entries in the `/etc/vfstab` file on each node of the cluster to reflect the new volume names.

j. Reboot the cluster using the `scshutdown` command:

```
# scshutdown -g 0 -y
...
ok boot
```

If encapsulating a non-root disk:

a. Run `vxinstall` on each node of the cluster, choose the “Custom Install” option at the main menu of `vxinstall`. When given the option, choose **not** to encapsulate the boot disk.

b. When `vxinstall` presents you with the disk you want to encapsulate, choose the “Install one disk at a time” option. If the disk presented (`vxinstall` presents lists of disks connected to each controller) is not listed, choose the “Leave these disks alone” option.

c. When `vxinstall` asks if you want to automatically reboot the node, choose no. Use `scshutdown` to shut down and reboot the nodes:

```
# scshutdown -g 0 -y
...
ok boot
```

27. We will be creating two disk groups, one which we can use later for NFS data and one for web server data. For the purposes of this lab, each disk group will require just 2 disks each. Make sure that the disks used for each disk group are in separate arrays. Determine which disks will be used in each disk group and complete the following table:

Disk Group 1 Name: `nfs-dg1`:

Disk	Physical Disk (c#t#d#)
1	
2	

Disk Group 2 Name: `webserver-dg1`:

Disk	Physical Disk (c#t#d#)
1	
2	

On just one of the nodes of the cluster[1]:

28. Create the disk groups. Use the `vxdiskadd or vxdiskadm` command to create and populate the disk groups:

```
# vxdiskadd c#t#d#
...
Which disk group [<group>,none,list,q,?] (default: rootdg) nfs-dg1
...
<Repeat for remaining disk in nfs-dg1 and for the disks in
webserver-dg1>
```

Verify the creation of the disk group by running the `vxdg list` command, both disk groups should be listed.

29. Create a 500 MB volume in each disk group. Use the `vxassist` command.:

```
# vxassist -g nfs-dg1 make vol-01 500m layout=mirror,log
# vxassist -g webserver-dg1 vol-01 500m layout=mirror,log
```

1. This step should be done on one of the nodes that is physically connected to the disks you intend to use in the disk group

30. Use `scsetup` to register the disk group with the cluster framework. This should be done on the node which currently "owns" the disk group. Run `vxdg list` on each node to determine which node currently owns the disk group.
31. Create file systems on the volumes in the disk groups:

```
# newfs /dev/vx/rdsk/nfs-dg1/vol-01
... <newfs output>
# newfs /dev/vx/rdsk/webserver-dg1/vol-01
... <newfs output>
```

On each node of the cluster:

32. Create a mount point for the file systems created in step 31:

```
# mkdir /global/nfs
# mkdir /global/webdata
```

On each node of the cluster that is able to import the disk group:

33. Add the new file systems to the `/etc/vfstab` file. Specify that the files be mounted globally, with the `syncdir` and `logging` option. The new entries should look like this:

```
/dev/vx/dsk/nfs-dg1/vol-01   /dev/vx/rdsk/nfs-dg1/vol-01   /global/nfs  ufs 2 yes global,logging
/dev/vx/dsk/webserver-dg1/vol-01   /dev/vx/rdsk/webserver-dg1/vol-01   /global/webdata  ufs 2 yes
global,logging
```

On just one of the nodes that is able to import the disk group:

34. Mount the file systems:

```
# mount /global/nfs
# mount /global/webdata
```

On each node of the cluster:

35. Verify that the `/global/nfs` and `/global/webdata` file systems are available on all the cluster nodes:

```
# mount
...<Mount output>
/global/nfs on /dev/vx/dsk/nfs-dg1/vol-01 ...
...
/global/webdata on /dev/vx/dsk/webserver-dg1/vol-01 ...
...
```

Configure PNM

On each node of the cluster:

36. Use the `pnmset` command to configure the public network interfaces for each node.
37. Verify the state of the NAFO groups on each node by using the `pnmstat -l` command

Configure `ntp.conf`

On each node of the cluster:

38. Edit the `/etc/inet/ntp.conf` file. Delete the following for nodes that do not exist. For example, in a four node cluster, the following entries would need to be deleted:

```
peer clusternode5-priv
peer clusternode6-priv
peer clusternode7-priv
peer clusternode8-priv
```

Verify the cluster installation

39. Run the `scconf -p` and `scstat -p` commands to display the current configuration and state of the cluster:

```
# scconf -p
...<Output from scconf command> ...
# scstat
...<Output from scstat command> ...
```

Things to try

40. Now that we have a working cluster, let's try a few things to determine how the cluster reacts to various events:
    a. Try "crashing" a node, how do the other node's in the cluster react?
    b. Pull out both cluster interconnect paths from a node. Did the failure fencing work properly?

Without replacing the interconnect path cables, reboot the node and see if it can rejoin the cluster. What happens?

Replace the interconnect cables and have the node rejoin the cluster.

c. Simulate an “amnesia” condition. Shut down all nodes of the cluster, one at a time. Try restarting the first node you shutdown. Does it boot completely? What needs to happen before it can boot completely?

LAB 3

Configuring Resource Groups

In this lab, we will be configuring the NFS and Apache data services. We will use the disksets / disk groups which have been configured in the installation labs for our applications. The NFS data service will be configured as a failover service while we will configure the Apache data service as a scalable service

Install the Sun Cluster Data Service Agents

On each node of the cluster:

1. If you haven't done so previously (in Lab 2), install the software required for supporting NFS and Apache on the cluster. Run the following `scinstall` command:

```
# scinstall -ik -s nfs,apache -d <Path to SC 3.0 dataservices CD>
```

Installing NFS as a failover data service

On each node of the cluster and the administrative workstation:

2. Create an entry in `/etc/hosts` for the logical host address we will be configuring with the NFS data service. Your instructor will provide an IP address that can be used for this purpose (Don't forget to do this on the admin workstation, so it may be used as a client for the data service).

```
nfs-server    aaa.bbb.ccc.ddd
```

On Just One node of the cluster:

3. Before we can configure the resource group, we must create an administrative directory which will contain the dfstab file for the NFS resource. This directory must be placed on a cluster file system.

```
# cd /global/nfs
# mkdir nfs-admin
# cd nfs-admin
# mkdir SUNW.nfs
```

4. In the /global/nfs/nfs-admin/SUNW.nfs directory, create a dfstab file with an entry to share /global/nfs. Call the file dfstab.nfs-rs. Note: The trailer used in this filename must match the name of the NFS resource in the resource group.

```
# cd /global/nfs/nfs-admin/SUNW.nfs
# vi dfstab.nfs-rs
...<Add an entry to share /global/nfs>
```

5. Register the NFS resource type with the cluster framework.

```
# scrgadm -a -t SUNW.nfs
```

6. Create a resource group for our failover NFS server. Make sure to set the PATHPREFIX property to point at the directory containing the SUNW.nfs directory created in the step 4..

```
# scrgadm -a -g nfs-server-rg -h <node1>,<node2> \
          -y PATHPREFIX=/global/nfs/nfs-admin
```

7. Add a logical hostname resource to the nfs-server-rg resource group. Use the appropriate NAFO groups and nodes for your cluster:

```
# scrgadm -a -L -g nfs-server-rg -l nfs-server \
                -n nafo0@<node1>,nafo0@<node2>...
```

8. Now, add the NFS resource to the resource group

```
# scrgadm -a -g nfs-server-rg -j nfs-rs -t SUNW.nfs
```

9. Enable and bring the resource group online:

```
# scswitch -Z -g nfs-server-rg
```

10. Check the status of the resource group using the scstat command:

```
# scstat -g
```

11. Use the scswitch command to switch the resource group from its current primary to another node in the cluster

```
# scswitch -z -h <Node to switch to> -g nfs-server-rg
```

12. Create a shell script to access the NFS share via the logical host from the admin workstation. Crash the node which is the current primary for the resource group. How long does it take to for the NFS resource group to failover? _____

Configure the RGM for Scalable Apache

On each node of the cluster:

13. Create an entry in `/etc/hosts` for the logical host address we will be configuring with the Apache web server. Your instructor will provide an IP address that can be used for this purpose (Don't forget to do this on the admin workstation, so it may be used as a client for the data service).

```
web-server    aaa.bbb.ccc.ddd
```

14. Apache is included with Solaris8, so no explicit installation is needed. The Apache binaries are located in `/usr/apache/bin`. The configuration files are located in `/etc/apache`.
15. Create a `conf` directory in `/etc/apache`

```
# mkdir /etc/apache/conf
```

16. Copy the sample `/etc/apache/httpd.conf-example` to `/etc/apache/conf/httpd.conf`

```
# cp /etc/apache/httpd.conf-example /etc/apache/conf/httpd.conf
```

17. Edit the `httpd.conf` file, change the following entries:

```
ServerName web-server  (Uncomment this line)

DocumentRoot /global/webdata/htdocs
<Directory "/global/webdata/htdocs>" (from "/var/apache/htdocs")

ScriptAlias /cgi-bin/ "/global/webdata/cgi-bin"
<Directory "/global/apache/cgi-bin"> (from "/var/apache/cgi-bin")
```

On one node of the cluster:

18. Create directories for the html and cgi files

```
# mkdir /global/webdata/htdocs
# mkdir /global/webdata/cgi-bin
```

19. Copy the sample html documents to the htdocs directory

```
# cd /var/apache/htdocs
# cp -r ./* /global/webdata/htdocs
```

20. Copy the file called "test-cluster.cgi" from the classroom server to /global/webdata/cgi-bin. We will use this file to test the scalable service. Make sure that test-cluster.cgi is executable by all users.

```
# cp /net/<trng-svr>/export/class/test-cluster.cgi \
              /global/webdata/cgi-bin
# chmod ugo+rw /global/webdata/cgi-bin/test-cluster.cgi
```

21. Register the resource types required for the Apache data service:

```
# scrgadm -a -t SUNW.apache
```

22. Create a resource group for the shared address resource. Use the appropriate node names for the -h argument:

```
# scrgadm -a -g web-server-rg -h <Node1>,<Node2>...
```

23. Assign an address and NAFO groups to the resource group. This will add a resource named web-server to the web-server-rg resource group:

```
# scrgadm -a -S -g web-server-rg -l web-server \
          -n nafo0@mars,nafo0@venus
```

24. Create another resource group (dependent on the shared address resource group configured in step 22) for the Apache resource (you may need to use a different number of Maximum_primaries and Desired_primaries depending on the size of your cluster).

```
# scrgadm -a -g apache-rg -y Maximum_primaries=2
          -y Desired_primaries=2 -y RG_dependencies=web-server-rg
```

25. Add a resource of the SUNW.apache type to the resource group:

```
# scrgadm -a -j apache-res -g apache-rg -t SUNW.apache        \
    -x ConfDir_list=/etc/apache/conf                          \
    -x Bin_Dir=/usr/apache/bin
    -y Scalable=TRUE                                           \
    -y Network_resources_used=web-server                      \
```

26. Bring the shared address and apache web server resource groups online:

```
# scswitch -Z -g web-server-rg
# scswitch -Z -g apache-rg
```

27. Run `scstat -g` to determine display the state of the resource groups

Test the scalable service

Try connecting the web server using the browser on the admin workstation. Connect using the URL: `http://web-server/cgi-bin/test-cluster.cgi`.

Repeatedly hit the refresh button on the browser. The test-cluster.cgi script will display the actual nodename which is servicing the request (It may take several iterations before the packet is distributed to a new node)

Try crashing the node to which the browser is currently connected. Hit the refresh button -- does the server still respond? How long did it take to respond?

Simulate Failures ...

Now that we have a few applications installed, go ahead and simulate failures in the cluster --- observe what happens to the data services. Do they remain available on nodes of the cluster?

If time permits, go ahead an try to install other data services. Follow the guidelines outlined in Chapter 6 of your training manual.

Sun Microsystems, Inc.
2550 Garcia Avenue
Mountain View, CA 94043
U.S.A.

Part No.: TBD
Course No.: TBD
Revision 1.1, April 2000