TRU64 UNIX

REBUILDING THE ROOT FILE SYSTEM

& RECOVERING ALL USER DATA

A guide to rebuilding/increasing the size of the root file system on systems with a mirrored boot disk

ITSD/TPSS, Vodafone Ltd.

The Courtyard, 2-4 London Road, Newbury, Berkshire, RG13 1JX

Amendment History

Date	Author	Issue	Remarks
23/8/99	Tony Miller	1.0	First Draft
25/8/99	Tony Miller	1.1	Modifications after initial testing
27/8/99	Tony Miller	1.2	Add use of mfs when doing restores
08/9/99	Tony Miller	1.3	Update for TRUcluster installations
09/9/99	Tony Miller	1.4	Minor updates on TRUcluster items
02/05/00	Tony Miller	1.5	Clarify fstab issues for BSD root f/s. Dosument AUTOACTION requirements
04/05/00	Tony Miller	1.6	volrootmir should have –a switch

Issue	1.6
Issue Date	4/5/00
Effective Date	
Author	Tony Miller
Approved by	
Approved by	

Purpose

To provide recommended and tested procedures for increasing the size of the root file system on a Digital UNIX system where the root is mirrored and is advfs based. This will include rebuilding of all file sets within the usr_domain since this is also part of rootdg disk group.

Scope

The pre requisite is that the individual doing the recovery has UNIX expertise. This can be generic and does not *have* to be Digital UNIX. Digital UNIX is an obvious advantage however.

Change Forecast

If any configuration change is made, either hardware or software then the recovery procedures (this document) will need to be updated.

Assumptions & Pre-Requisites

In order to rebuild the system, the following criteria must be met:-

- The following media must be available at the time of a recovery:

  1. Digital UNIX 4.0D CD-ROM
  2. Most recent system disk backup on tape
  3. DLT cleaning tape

- The output from the following commands:
  1. `volprint –hAt`
  2. `voldisk list`
  3. `df`
  4. `/etc/fstab`
  5. `volsave` (see the following section).
  6. `swapon –s`
  7. `ls –lR /etc/fdmns` after all services offlined/relocated.
  8. ASE configuration of all services to be moved during maintenance slot.

- All backups and other regularly scheduled maintenance cron jobs should be disabled.

Saving Current LSM Data

This section refers only to data under LSM control.

- **ASE Clusters only:** Fail over any services to the other node
- **ASE Clusters only:** Do a `ls –lR /etc/fdmns`
- Make sure that all oracle databases are stopped and that any raw lsm volumes are not being used.
- Make sure that all user data file systems are unmounted
- The `volsave` utility will save a configuration copy of the current LSM configuration. Make sure that the output from a `volsave` is part of the current backup (i.e., is this running as a root cron job). If it is not running as a root cron job, or is otherwise not part of the backup you will use to restore from, make a backup of this directory and manually restore it later. The backup location will be something like `/usr/var/lsm/db/LSM.19990823140003.montst2`.
- For each remaining LSM disk group (not rootdg), note the device special files belonging to this group, along with any domain and file set information then deport it:

  ```
        # voldg deport diskgroupname
  ```
- **ASE Clusters only:** Do the backup of all rootdg file systems now.

Recovering Digital UNIX

- At the console prompt (`>>>`) issue the `SHOW DEV` command. Note the hardware name and UNIX device name of:
  - Any swap devices
  - The disk device names of the rootdg disks
  - The CD device
  - The TZ89 tape device name (bus, target & LUN)

- Insert the Digital UNIX CD into the CD-ROM drive

- From the boot console issue the `>>>boot dklnnn` command where `dklnnn` is the CD device name.

- A menu screen will be displayed. Select the `UNIX Shell` option. Then:

```
# ksh                                !Select Korn shell
# export EDITOR=vi                   !Set default editor
# export TERM=vt320                  !Set default terminal type
# cd /dev                            !Need to create dev. files
# ./MAKEDEV rz?                      !root mirror 1
# ./MAKEDEV rz?                      !usr mirror 1
# disklabel -r rz?                   !Is there a default label?
# disklabel -z rz?                   !Erase current label
# disklabel -wr rz? hsz70            !Write a default label
# disklabel -r rz1 > /tmp/rzlabel    !Copy label to a file
# vi /tmp/rzlabel                    !Edit the file
```

----- Set disk label up as required. ---
----- Note: ESCAPE is ctrl/3 (‘3’ on main keyboard) -----
----- Check terminal is *not* set to VT420. Editor may not work -----
----- properly if set greater than VT320. Use VT100 if in doubt. ------
----- Set the terminal to 7-bit characters & 7-bit controls ------

```
# disklabel –R –r –t advfs rz? /tmp/rzlabel hsz70    !Recover label & write boot blocks
```

----- Repeat the above sequence of commands in order ---
----- to write a disklabel to the first of the two usr_domain disks. ---
----- use `disklabel -R rznn file.name` to write the label to the disk. i.e., no boot blocks needed. ---
----- You may need to disklabel –z and disklabel –wr first -----

- The next section of this guide describes how to restore data stored in the rootdg disk group from a backup tape.

Recovering the System Disk

- Place the backup tape into the drive.

- Make the device special files for the tape drive. Use the following command to identify the tape device (bus, target, LUN):

  ```
  # scu show edt | grep –i sequential
  ```

  Create the tz device name as follows (where nn is (bus*8)+target):

  ```
  #cd /dev
  #./MAKEDEV tznn
  ```

- Use the `file /dev/rmth*` command and note the device special files. This command will produce a display of the form:

  ```
      /dev/rmt0a: SCSI #7 TZ89     tape #74 (SCSI ID 2)
  ```

  An alternative is to use the command: `mt –f /dev/rmtnh status` which will give more detail about the drive including whether a tape is loaded etc.

- **Note:** If you are uncertain as to which file set on the backup tape is `/` and `/usr`, use appropriate combinations of the following commands. Note use of the *rewind* and *no rewind* devices:

  ```
  > mt -f /dev/rmt0a rewind
  > vrestore -tvf /dev/nrmt0a     !Lists 1st file set
  > mt -f /dev/rmt0a rewind
  > mt -f /dev/nrmt0a fsf 1       !Skip to 2nd file set
  > vrestore -tvf /dev/nrmt0a     !Lists 2nd file set
  > mt -f /dev/rmt0a rewind
  > mt -f /dev/nrmt0a fsf 2       !Skip to 3rd file set
  > vrestore -tvf /dev/nrmt0a     !Lists 3rd file set
  ```

- When booted from the CD, UNIX shell option, `vrestore`'s may fail with a message to the effect that `/var` is out of space. This is due to the small default size of the memory file system (mfs) created for `/var` and `/tmp`. The solution is to create a larger mfs mounted on `/tmp` and to use `/tmp` as the mount point for all further restores.

  Restore the root (`/`) file system. Create the /etc/fdmns structure and make the new root file system:

  ```
  # cd /etc/fdmns
  # mkfdmn –l 2048 /dev/rznna newroot_domain      !use the a partition on device rznn
  # mkfset newroot_domain root
  # mfs -s 204800 /tmp
  # mkdir /tmp/newroot
  # mount newroot_domain#root /tmp/newroot
  # cd /tmp/newroot
  ---position the tape so as it is at the start of the appropriate archive ---
  # vrestore -xf /dev/nrmt0h
  ```

- Restore the /usr file system:

```
# mkfdmn –l 2048 /dev/rznna   newusr_domain       !use the a partition on device rznn
# mkfset newusr_domain usr
# mkdir /tmp/newusr
# mount newusr_domain#usr /tmp/newusr
# cd /tmp/newusr
---Position the tape ready to read the /usr vdump archive---
# vrestore -xf /dev/nrmt0h
```

- The recovery is now partially complete. The system cannot be rebooted for several reasons. These include:

  1. The restored system thinks it has a mirrored system disk under LSM control - which it does not.
  2. The system thinks it has valid advfs file domains and file sets for data volumes available - which it does not.

These problems will now be overcome.

- Prevent LSM from starting. Undo modifications in the `inittab` file:

```
# cd /tmp/newroot/etc
# cp inittab inittab.vodafone
# vi inittab    !Remove the lsm, lsmr & vol lines
```

- Prevent non root disk file systems from mounting. Edit the `fstab` file:

```
# cd /tmp/newroot/etc
# cp fstab fstab.vodafone
# vi fstab
```

Note 1: Modify swap to use `/dev/rz?b` instead of `/dev/vol/rootdg/swapvol`.

Note 2: Remove any secondary swap areas.

Note 3: Delete all lines not relating to root, primary swap and `/usr`.

Note 4: Ensure that the root and usr *based* file systems are of the form `domainname#filesetname` as you will be restoring to an advfs file structure. If either of these were formerly BSD file systems under LSM control (/dev/vol/…/…) or BSD volumes within a partition (/dev/rz?a) then they should also be changed to use the domain format.

- Modify kernel options file to disable LSM:

```
# cd /tmp/newroot/etc
# cp sysconfigtab sysconfigtab.vodafone
# vi sysconfigtab          !Delete the following lines:
  lsm:
  lsm_rootdev_is_volume = 1
  lsm_swapdev_is_volume = 1
```

- **ASEcluster only:** Disable starting of cluster software:

```
# cd /tmp/newroot/etc
# vi sysconfigtab          !Add  the following lines:
  clubase:
  cluster_disable=1
```

- Remove LSM swap information:

```
# cd /tmp/newroot/sbin
# rm swapdefault                !Link to /dev/vol/rootdg/swapvol
# rm swapdefault.encap          !Link to /dev/rzxxy
```

- Remove LSM and plex information for system disk:

```
# rm -rf /tmp/newroot/dev/vol* !Ditch volconfig, volevent, volinfo, voliod
# rm /tmp/newroot/etc/vol/volboot          !The volboot file
# rm -r /tmp/newroot/dev/plex              !Plex information
```

- Remove the `/usr` file domain links and replace with a link to `rz?a`:

```
# cd /tmp/newroot/etc/fdmns/usr_domain
# rm rootdg.usrvol                     !Link to /dev/vol/usrvol
# ln –s /dev/rz?a rz?a
```

- Remove the / file domain links and replace with a link to rz?a. **Note:** If the system had a UFS based root file system, create the link and the domain:

```
# cd /tmp/newroot/etc/fdmns/root_domain
# rm rootvol                        !Link to /dev/vol/rootvol
# ln –s /dev/rz?a rz?a
```

- Temporarily stop any 3rd party applications from starting (e.g., oracle).

```
# cd /tmp/newroot/sbin/rc3.d
```

---Comment out any S script as appropriate, rename them to 'dontstartSnnxxx'---

- Check the `/etc/fdmns` directory structure. It should only have entries for items in the `fstab` file (i.e., no cluster service domains). Delete as appropriate.

- Shut down after dismounting `/tmp/newroot and /tmp/newusr`:

```
# cd /
# umount /tmp/newroot
# umount /tmp/newusr
# halt                    !exit from the shell
```

---Note, you cannot shutdown –h now or init 0 from the shell. This will hang the system requiring a manual halt of the system ---

- Verify the setting of the console parameter `autoaction`. If this is set to anything other than HALT, then change it to HALT). This will prevent unexpected boot ups should the INIT command be given or a system power cycle be required. This is especially useful when the system is on a remote unmanned site. **Note:** This parameter will be reset to its expected value when the restore is complete and verified as good:

```
>>> show autoaction                     !Note the setting
>>> set autoaction halt
>>> show autoaction                     !Verify it has been changed
```

- Shut the system down and ***boot single user*** from the root disk you have just restored after setting the console device `bootdef_dev` to point *only* to the single device just partitioned etc. Verify that both / and /usr file systems are mounted properly and from the expected disks. Do not proceed further than this point until single user boot problems have been fixed.

```
>>> set bootdef_dev dkln
>>> boot –fl s
# mount –u /
# mount -a
```

- Move to multi-user mode:

```
#init 3
```

- Restore the `/tmp` file system - the 3rd file set. Note the use the *rewind* and *no rewind* tape devices. This restore will take approximately 5 minutes to complete:

  ```
  # mkfset usr_domain tmp                         !Make the tmp file set
  # cat /etc/fstab.vodafone | grep -i \           !Update fstab so that
         tmp >> /etc/fstab                        ! /tmp is mounted in future
  # mount –a                                      !Mount it
  ---position the tape accordingly---
  # vrestore -xf /dev/nrmt0h -D /tmp              !Restore it
  ```

- Restore any other usr_domain file sets.

  Restore any other usr_domain based file sets (e.g., /u00). Remember to create the file sets, mount them and update fstab after restoring them

  Similarly check that any other domains located on the system disk (or `usr_domain` disk) have been recovered:

- LSM now needs to be configured. Encapsulate root, swap and the usr_domain into lsm. **Note:** that the separate encapsulation of `usr_domain` is only required if this is *not* on the system boot disk:

  ```
  # volencap rznn    (nn is disk / & swap are on)
  # volencap usr_domain
  # shutdown –r now
  ```

- Check that /, primary swap, and all file systems within usr_domain are properly mounted and have been encapsulated into LSM (`volprint –hAt)`.

- Set up root mirroring. **Note: This could be left until after all upgrades etc are complete. This will permit a recovery via the original system should this be required.**

  Write a new disk label to the o/s mirror disk as per the o/s disk (a and b partitions only) then set up root mirroring. The disk label should be as per the primary disk *before* the encapsulation took place.

  ```
  # volrootmir –a rznnn &
  ```

- Set up /usr mirroring. **Note: This could be left until after all upgrades etc are complete. This will permit a recovery via the original system should this be required.**

  Write an appropriate disk label to the usr_domain mirror disk (as you did for the current usr_domain disk before encapsulation). Set this disk up as a noprivate disk (nopriv) before adding it to LSM:

  ```
  # voldisk define rz..a type=nopriv
  # voldiskadd rz..a
  ```

  Note: Answer NO to the ‘reinitialise’ prompt.

  Now start the mirroring process:

  ```
  # volassist -g rootdg mirror usrvol rz..a &
  ```

Recovering Secondary Swap Areas

Secondary swap (swap space not on the system disk) should now be recovered.

- Write an appropriate disk label to the two disks (for mirrored swap). This will probably have all partitions as 0 length 0 offset except for partition 'C' which is the default of 'whole disk'.

- Add both disks to LSM into the rootdg:
```
# voldiskadd rz?1?
# voldiskadd rz?1?
```

- Note the length of the public region of both disks, create a subdisk to use all of the disk space available (both plexes), create the plexes and create the volume. Note that you should create the volume with only one plex:
```
# voldisk list rz?1?                !Note the size
# volmake sd rz?1?-01  rz?1?,0,nnn
# volmake sd rz?2?-01  rz?2?,0,nnn
# volmake plex swapvol1-01 sd=rz?1?-01
# volmake plex swapvol1-02 sd=rz?2?-01
# volmake vol swapvol1 plex=swapvol1-01 usetype=gen \
   start_opts=norecov
# volume start swapvol1
```

- Confirm the volume exists and is started:
```
# volprint –g rootdg -ht
```

- Update the /etc/fstab file with an entry of the following format. This should be added immediately below the primary swap line:
```
/dev/vol/rootdg/swapvol1 swap2    ufs sw 0 2
```

- Add the swap volume so it becomes live and in use:
```
# swapon /dev/vol/rootdg/swapvol1
# swapon –s
```

- **Note:** The volume exists and is now active. It will become active on future system reboots automatically. However, only one plex exists at this time. This is clearly an insecure situation and the second plex should be added as soon as possible. **Note: This could be left until after all upgrades etc are complete to save time.**

  If the plex is added now, the mirror synchronisation will take some time to complete. Obviously, the system cannot be rebooted until the synchronisation has completed.

  To add the second plex, use the commands listed below:
```
# volplex att swapvol1 swapvol1-02 &
# volprint –g rootdg –ht &
```

  Use the `volprint` command to verify that the plex is mirroring. The new plex will be in a `WO` state whilst the mirror is in progress.

- Add any further secondary swap volumes in a similar fashion.

Recovering Previously saved LSM Data

This section refers only to data formerly under LSM control.

- Add the disk media records into LSM for each disk within each disk group you deported.
  ```
  # voldisk define rznn
  ```

- For each LSM disk group, import it:
  ```
  # voldg import diskgroupname
  ```

- Verify the disk group exists:
  ```
  # volprint –g diskgroupname -ht
  ```

- Start all volumes in all disk groups
  ```
  # volume –g diskgroupname startall
  ```

- Restore the `fstab` file such that it includes details of all user lsm volumes as before.

- Mount all lsm volumes:
  ```
  # mount -a
  ```

- **Note: Before proceeding, make sure that any background mirror tasks have completed.** Use the `wait` command.

- Check that the system is set to mirror boot. Shut system down and check the console parameters `bootdef_def (`set to `dkl???,dkm???)`.

- Reset the console parameters `AUTOACTION` which was set to `HALT` earlier in the recovery procedure. Reboot to multi-user mode.

- Restore any S start scripts within /sbin/rc3.d such that oracle etc will start on future system reboots. **Note: This could be left until after all upgrades etc are complete. This will prevent layered products from starting during upgrades.**

- Do a full system reboot to ensure the system starts up exactly as before the start of this procedure.

Post Restore Tasks

- Enable any backups and other regular maintenance cron jobs disabled at the start of this procedure.