Working with Tru64 UNIX Hardware Manager

Dave Bechtold

Consultant, Tru64 UNIX Support
HP Services

Dave.Bechtold@hp.com

HP WORLD 2003
Solutions and Technology Conference & Expo

Working with Tru64 UNIX Hardware Manager

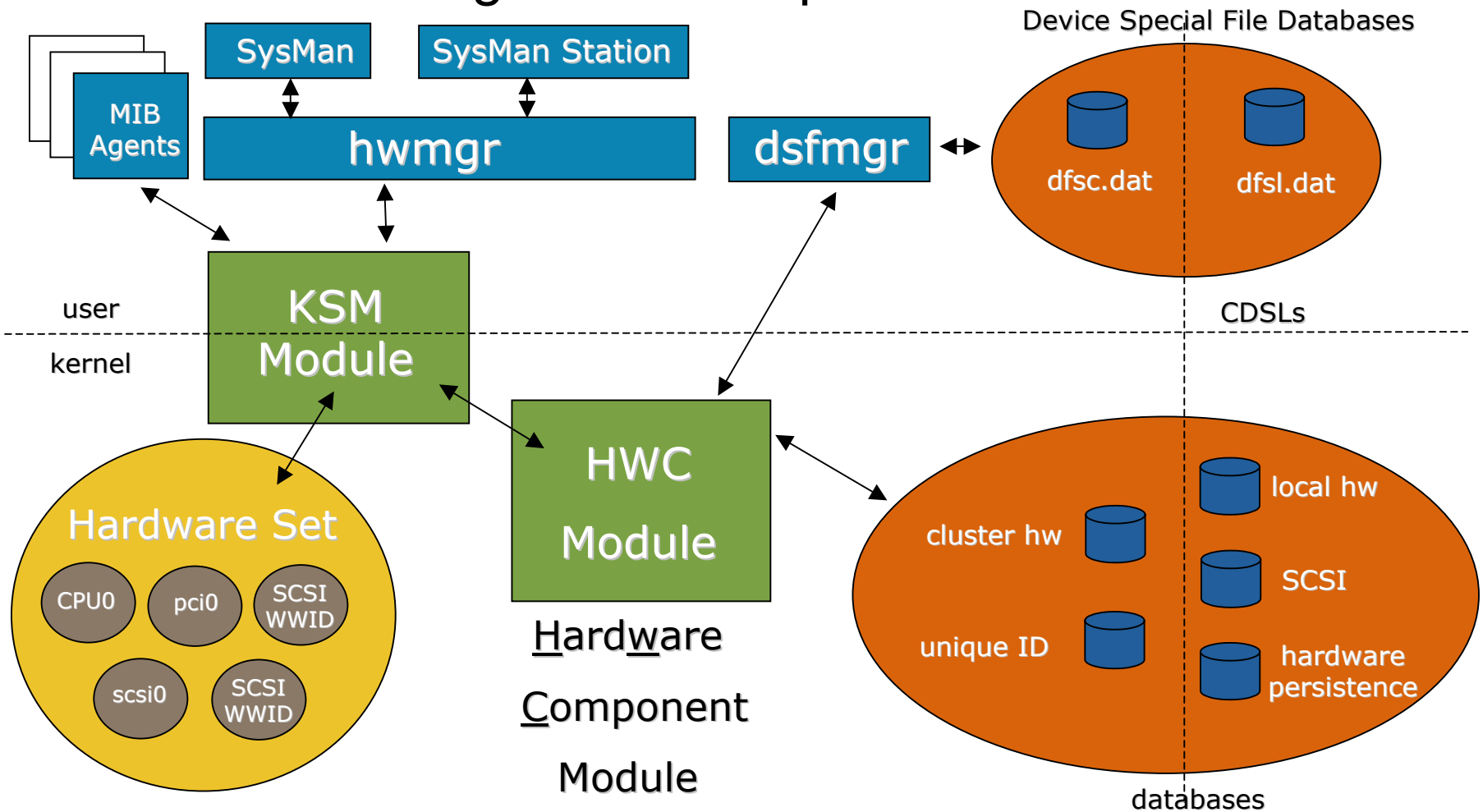
- Session Topics
  - **Hardware Management Overview**
  - Hardware Management Utilities & Commands
  - Identifying Hardware & Device Status Details
  - Recovering From Failures
  - Q & A

Hardware Management Overview

- Introduction
  - Single user interface for managing hardware
  - Support for new device naming convention needed to support SCSI-3, and SSI Clustering – unique names
  - Provides the following Service:
    - Automatic Device Recognition
    - Device Special File (Cluster-wide unique)
    - Per-device properties/attributes
    - Topology Information (Hierarchy / Relationship)
    - Category relationship (type-based)
    - Online replacement (OLAR)
    - MIB relationships
    - Hardware Database management

Hardware Management Overview

- Hardware Management Components



Hardware Management Overview

- **Kernel Set Manager (KSM Module)**
  - Generic API for all kernel Sets
  - Used to initiate hardware component tasks and retrieve hardware component information via user level application requests, local and remote
  - Manage hardware data organized as the **Hardware Set**

Hardware Management Overview

- **Hardware Set**
  - In-memory container describing known / registered hardware components on a system
  - A Hardware component joins the Hardware Set when registered by its driver and then assigned a unique HWID
  - Hardware Set is managed by the **hwmgr** utility via KSM

Hardware Management Overview

- **Hardware Component Manager (HWC Module)**
  - Contains registration routines to create and maintain the Hardware Set and device nodes for device special file management
  - Maintains the Hardware Database on disk

Hardware Management Overview

- Hardware Management Utilities & Commands
  - **hwmgr** – Hardware Manager CLI for managing hardware components and devices, cluster-aware
  - **dsfmgr** – Device Special File Manager used to maintain device special files, directories and associated databases
  - **dn_setup** – jacket script used to execute properly sequenced dsfmgr commands
  - **ddr_config** – Dynamic Device Recognition utility used by SCSI CAM subsystem to dynamically configure devices
  - **scu** – SCSI CAM Utility provides a tool to interrogate and test peripherals attached to the SCSI buses

Hardware Management Overview

- ***sysman -menu***
  - Interface provided for terminals
  - Used to view hierarchy & devices, CPU status, manage CPUs on SMP system and OLAR Policy configuration (olar.config)
- ***sysman –station***
  - Interface provides a GUI for monitoring of Hardware, File Systems, Network, and Storage
  - Provides Hardware Topology and ability to view component attributes
- **Insight Manager**
  - Web based interface for Insight Management SNMP Agents
  - Used to monitor status of system hardware and subsystems, launch Sysman Station or Menu.

Hardware Management Overview

- Hardware Management Database
  - 10 Primary database files compose the system's Hardware Database
  - **Should not be edited** by hand; hwmgr via HWC Kernel Module and dsfmgr programs manage these databases
  - Each of the 10 files have .bak – **backup files**; device special file databases also have historical files h00-07
  - Some database files are text, others binary
  - All databases exist in /etc or a CDSL referenced from the /etc directory (standalone vs. cluster)
  - All generated at installation time, not on OS distribution
  - **Heavily inter-dependent**

Hardware Management Overview

- Hardware Database Files

File	Utility	CDSL	Description
dec_devsw_db	devswmgr	Yes[1]	Kernel dev switch database, tracks driver major numbers, instances and if in use
dec_scsi_db	hwmgr & dsfmgr (scsimgr)	Yes[1]	SCSI CAM database, tracks WWIDs for all SCSI devices using SCSI ID (-did), tracks path state information
dec_hw_db	hwmgr	Yes[1]	Maintains hardware persistence information – device relationships / topology, the “name” database
dec_unid_db	hwmgr	No	Unique ID database used to assign hwid to devices (hwids never re-used), device special file group ids, cluster minor ids

Hardware Management Overview

- Hardware Database Files

File	Utility	CDSL	Description
dec_hwc_ldb	hwmgr	Yes[1]	Local hardware component database file, maintains device components and saved device attributes
dec_hwc_cdb	hwmgr	No	Cluster hardware component database that contains information for devices shared by all members of a cluster
dccd.dat	dsfmgr	No	Device Category to Class Directory attributes database used to describe the device special files
dcdd.dat	dsfmgr	No	Device Class Directory Default Database used to describe the device special file directory structure

Hardware Management Overview

- Hardware Database Files

File	Utility	CDSL	Description
dfsl.dat	dsfmgr	Yes[2]	Systems local device special file database, describes device special file associated with a hwid
dfsc.dat	dsfmgr	No	Cluster-wide device special file database, describes device special file associated with a hwid shared across cluster

[1] /cluster/members/{memb}/boot_partition/etc/ - TruCluster Only otherwise in /etc
[2] /cluster/members/{memb}/etc/ - Always a CDSL – standalone or TruCluster

Hardware Management Overview

- Hardware Database Support Files

File	Utility	CDSL	Description
cfginfo	N/A	Yes[2]	Text file which lists files and directories that comprise the Hardware Database This file comes from OS CD
gen_databases	N/A	Yes[1]	Text file to convert database name to database file and database handler This file comes from OS CD

[1] /cluster/members/{memb}/boot_partition/etc/ - TruCluster Only otherwise in /etc
[2] /cluster/members/{memb}/etc/ - Always a CDSL – standalone or TruCluster

Hardware Management Overview

- Dynamic Device Recognition (DDR)
  - SCSI CAM I/O subsystem component used to define new or non-standard SCSI devices (Disk, Tape, Changer, etc..)
  - Provides means to modify a devices default attributes (TagQueueDepth), and define defaults for new devices
  - Two database files used to support DDR
    - **/etc/ddr.dbase** – ACSII source file for edits
    - **/etc/ddr.db** – Binary compiled file read by SCSI CAM
  - The ***ddr_config*** utility is used to view the in-memory DDR database, compile ASCII file (ddr.dbase -> ddr.db), and sync on-disk ddr.db with in-memory database
  - Help – ddr_config(8), ddr.dbase(4), scsi(7), tz(7)

Hardware Management Overview

Hardware Discovery at Boot Time

1) SRM BOOT command executed from console
2) Primary, Secondary and Tertiary (*osf_boot*) loaders execute to locate and load kernel (/vmunix)
3) The kernel initializes (sizing memory, creates data structures and kernel threads)
4) During kernel initialization hardware discovery routines are executed before **hardware devices announce on console**

   HW_TOPOLOGY database (/etc/dec_hw_db) is read using **BOOTDEF_DEV** console variable and reconciled with temporary HW_DB in-memory

```
>>> boot
…
.. <Loads kernel and initializes memory, etc…> …
**************************************************
Alpha boot: available memory from 0x20ac000 to 0x1fffc000
Compaq Tru64 UNIX V5.1B (Rev. 2650); Thu Oct 17 05:51:11
EDT 2002
physical memory = 512.00 megabytes.
available memory = 476.68 megabytes.
using 1884 buffers containing 14.71 megabytes of memory
Master cpu at slot 0
Starting secondary cpu 1
Firmware revision: 6.0
PALcode: UNIX version 1.23
AlphaServer 4000 5/400 4MB
pci1 (primary bus:1) at mcbus0 slot 5
Loading SIOP: script c0000000, reg 7f6ed00, data c000a000
scsi0 at psiop0 slot 0 rad 0
isp0 at pci1 slot 2
isp0: QLOGIC ISP1040B/V2
isp0: Firmware revision 5.57 (loaded by console)
isp0: Fast RAM timing enabled.
scsi1 at isp0 slot 0 rad 0
tu0: DECchip 21140: Revision: 2.0
tu0: auto negotiation capable device
tu0 at pci1 slot 4
tu0: DEC TULIP (10/100) Ethernet Interface, hardware
address: 00-00-F8-1F-59-BA
tu0: auto negotiation on: will advertise 100BaseTX (UTP)
port: full duplex
```

Hardware Management Overview

- Hardware Discovery at Boot Time (Continued…)

5) The **init** process is created and initialized, the system switches to single-user mode. Kernel "**Hardware Set**" - no device special files in single-user mode, exist in memory

6) The init process continues by processing the /etc/inittab file and performs the following hardware related tasks:

   A. *hsd:Ss:sysinit:**/sbin/hotswapd***

   B. *fs:23:wait:**/sbin/bcheckrc***

      a. ***/sbin/dn_setup –sanity_check*** executes *dsfmgr –zx*

      b. ***/sbin/mountroot*** mount root read/write; update io sysconfig attr

         - ***/sbin/dn_setup -boot***

         i. *dsfmgr –C* Create any device directories if needed

         ii. *dsfmgr –N* "updating kernel basenames for system at /"

         iii. *dsfmgr –K* Creates all device special files

         iv. *dsfmgr -Sv* verifies /dev directories and devnode files, silently

      c. ***/sbin/mount –a*** mount all other file systems from /etc/fstab

Working with Tru64 UNIX Hardware Manager

- Session Topics
  - Hardware Management Overview
  - **Hardware Management Utilities & Commands**
  - Identifying Hardware & Obtaining Device Status Details
  - Recovering From Failures

Hardware Management Utilities & Commands

- Using **hwmgr(8)** – Hardware Manager
  - Requires ‘root’ privilege
  - Cluster-aware (-cluster & -member for scope of command)
  - Supports environment variables *hwmgr –view env* (HWMGR_HEXINTS, HWMGR_NOWRAP)
  - Uses data files for internal command formatting

    /etc/hwmgr/*.dat files
  - Commands asynchronously executed – cluster!!

    # evmwatch -f '[name sys.unix.hw.]' | evmshow*
  - Commands targeted at
    - a hardware subsystem (component, name, scsi, fibre)
    - a category (CPU, network, disk, tape)
    - a driver (unload & reload - dynamic driver)

Hardware Management Utilities & Commands

- Using **hwmgr(8)** – Hardware Manager

  Extensive help in man pages

  – **hwmgr(8)** – general information

  hwmgr –full | -verbose | -cluster | -member

  - **hwmgr_get(8)** - Commands that display and set component attributes and identify component categories

    hwmgr -get attr | -get category | -set attr –id NN –a xxxx=y

  - **hwmgr_view(8)** - Displays information about the status of the system and its hardware components

    hwmgr –view hierarchy | device | cluster

  - **hwmgr_show(8)** - Displays information from subsystems

    *hwmgr –show scsi | comp | name | **fibre***

  - **hwmgr_ops(8)** - Hardware management commands for performing operations on components

    *hwmgr locate | **delete scsi -did** | delete –id | scan scsi | refresh comp*

Hardware Management Utilities & Commands

- Using **dsfmgr(8)** – Device Special File Manager
  - Manages /dev directory and associated databases
    dccd.dat – Device Category to Class-Directory, Prefix
    dcdd.dat – Device Class Directory Default Database
    dfsl.dat - Local dev_t, hwid, type (char/block), name
    dfsc.dat – Cluster dev_t, hwid, type, name
  - Requires 'root' privilege
  - Cluster-aware (-a Add entry – scope dcdd.dat, -R remove)
  - ***dsfmgr –hx*** – Extended feature help, environment variables, current databases in use, remove session lock

Hardware Management Utilities & Commands

- Using **dsfmgr(8)** – Device Special File Manager
  - Extensive suite of options
    - **-v** Verify integrity of device special file DBs and directories
    - **-s** Display all data from databases, next instance numbers
    - **-I** Reset device instance number, clone-copy-delete (-vI)
    - **-F** Automatically fixes problems in DBs and /dev (-vFV)
    - **-x** Extended function, format, and information
    - **-V** Verbose mode
  - Touch file dsfmgr.log to capture command execution status in /tmp, /var/tmp, or /var/adm/smlogs

Hardware Management Utilities & Commands

- **dsfmgr** Exchange, Move and **hwmgr –redirect scsi**
  - **dsfmgr -e** Exchange device special files for named nodes <==>
    - Used when the new device special file name currently exists on the system, *hwmgr –view device*
    - Devices can be active – DRIVER OWNER not 0
  - **dsfmgr –m** Move device special files from one devnode to another =>
    - Used when the new device special file name does not exist on system, any system - cluster
  - **hwmgr –redirect scsi –src {did} –dest {did}** Redirects all attributes of a SCSI device to another device including the hwid, did, dev_t values (major / minor), device special file names, both devices must be active
    - Used to replace SCSI disks and tapes that have a valid path

Hardware Management Utilities & Commands

- Using **dn_setup(8)** Utility – Device Special Files
  - Script utility for executing dsfmgr commands
  - Documented in v5.1B man pages – **dn_setup(8)**
  - Run at boot time by bcheckrc script and mountroot
  - Advised to **ONLY use *–sanity_check*** to check for database inconsistencies and dsfmgr to correct problems
    - Executes *dsfmgr –zx* to check all databases
  - Be careful with ***dn_setup –init*** and ***–clean***
    - Run /sbin/mountroot in single-user prior to executing
    - Assumes /dev/MAKEDEV can be copied to / before deleting
    - If using LSM, run *volinstall update* to recreate device files
    - If using /dev/fd device need to remake by hand

Hardware Management Utilities & Commands

- Using **scu(8)** – SCSI CAM Utility
  - Used to maintain and diagnose problems with SCSI peripherals
  - Man pages – **scu(8)**, has extensive online help *scu> help*
  - Show "active" device paths (*show edt*), show reservation keys (*show reservations* – csid), diagnose hardware problems by testing for problems (*tur, verify media, show sense*)
  - Switch active paths to backup for multi-path devices
    - scu> *sbtl {inactivepath}* (Ex. Scu> sbtl 0 1 11)
    - scu> *start*
    - Initiate I/O to device and review path_xfers & path_state

      # hwmgr –get attr –id {hwid}

Working with Tru64 UNIX Hardware Manager

- Session Topics
  - Hardware Management Overview
  - Hardware Management Utilities & Commands
  - **Identifying Hardware & Device Status Details**
  - Recovering From Failures

Identifying Hardware & Device Status Details

- Display system hardware Topology / Relationships
  Name database **dec_hw_db**

 # hwmgr view hierarchy

```
HWID:   hardware hierarchy
-------------------------------------------------
  99:   platform AlphaServer ES45 Model 2
 100:     cpu CPU0
 105:     bus iop0
 106:       bus hose0
 107:         bus pci0
 108:           connection pci0slot7
 118:             bus isa0
 119:               connection isa0slot0
 120:                 keyboard keyboard0
 121:                 pointer mouse0
```

… continue…

Identifying Hardware & Device Status Details

 # hwmgr view hierarchy

… continued…

```
 110:            connection pci0slot8
 131:              fibrechannel_adapter emx0
 132:                scsi_bus scsi0
  80:                  disk IDENTIFIER=1 dsk0
  81:                  disk IDENTIFIER=2 dsk1
…<snip>…
  92:                  disk IDENTIFIER=25 dsk12
  98:                  control_port bus-0-targ-0-lun-0 scp0
 114:            connection pci0slot12
 115:              unconfigured_hardware Unconfigured-device-
                         (<NULL>)-at-pci0slot12
 116:            connection pci0slot16
 135:              ide_adapter ata0
 136:                scsi_bus scsi2
 207:                  disk bus-2-targ-0-lun-0 cdrom10
 137:                scsi_bus scsi3
```

Identifying Hardware & Device Status Details

- Display component status

  Component databases **dec_hwc_ldb** & **dec_hwc_cdb**

  #hwmgr –show comp

```
 HWID:  HOSTNAME   FLAGS SERVICE COMPONENT NAME
-------------------------------------------
    5:  oscar      rcd-- none    dmapi
   33:  oscar      --d-- iomap   FDI-fdi0-unit-0
   80:  oscar      rcd-- iomap   SCSI-WWID:01000010:6005-08b4-0001-48cd-0001-1000-0093-0000
   81:  oscar      rcd-- iomap   SCSI-WWID:01000010:6005-08b4-0001-48cd-0001-1000-0099-0000
   99:  oscar      r---- none    AlphaServer ES45 Model 2
  100:  oscar      r---- none    CPU0
  101:  oscar      r---- none    CPU1
  103:  oscar      r-d-- none    scp
  104:  oscar      r-d-- none    kevm
  107:  oscar      r---- none    pci0
  108:  oscar      r---- none    pci0slot7
  110:  oscar      r---- none    pci0slot8
  115:  oscar      r---- none    Unconfigured-device-(<NULL>)-at-pci0slot12
```

Identifying Hardware & Device Status Details

- Display device status & details of SCSI disk

  # hwmgr –get attr –id {hwid}

  - Device characteristics

```
211:
  name = SCSI-WWID:01000010:6005-08b4-0001-48cd-0001-1000-00de-0000
  category = disk
  sub_category = generic
  architecture = SCSI
  phys_location = IDENTIFIER=11
  dev_base_name = dsk10
...
  model = HSV110 ©COMPAQ
...
  power_mgmt_capable = 0
  boot_capable = 0
  registration_time = Mon Jul  7 20:37:58 2003
  user_name = (null)  (settable)
  location = (null)  (settable)
```

… continued….

Identifying Hardware & Device Status Details

- Display device status & details of a SCSI disk

  # hwmgr –get attr –id {hwid} (continued…)

  - Performance Statistics

```
active_i/o_cnt = 0
pending_i/o_cnt = 0
queue_depth = 25
read_ops = 138
write_ops = 246
total_ops = 425
active_que_usec = 2226594
total_pending_ops = 0
pending_que_usec = 0
read_bytes = 8735744
write_bytes = 17072128
active_paths = 2
standby_paths = 2
failed_paths = 0
donot_use_paths = 0
path_fail_limit = 12
device_starvation_time = 25 (settable)
dev_busy_usec = 453130
```

Identifying Hardware & Device Status Details

- Display device status & details of a SCSI disk

 # hwmgr –get attr –id {hwid}

- Path Status

```
path_port_id_0 = 0
path_target_id_0 = 0
path_lun_id_0 = 11
path_dev_i/o_cnt_0 = 0
path_state_0 = 1
path_new_state_0 = 0
path_xfer_0 = 123
path_wds_0 = 71936
path_avserv_0 = 287112
path_pxfer_0 = 0
path_avwait_0 = 0
```

 # hwmgr –show scsi –full –id {hwid}

- shows valid paths

```
BUS    TARGET   LUN    PATH STATE
-------------------------------------
  0      0      11      valid
  0      1      11      valid
  1      0      11      valid
  1      1      11      valid
```

path_states: **1 = Active**
2 = Backup/Failover
8 = Stale

Working with Tru64 UNIX Hardware Manager

- Session Topics
  - Hardware Management Overview
  - Hardware Management Utilities & Commands
  - Identifying Hardware & Obtaining Device Status Details
  - **Recovering From Failures**

Recovering from Failures

- Replacing Failed Hardware Devices
  - Considerations for replacing failed hardware devices
    - It's advised to perform hardware replacement procedures during a **maintenance period**. The potential to cause a system panic or hang condition is greater when a device has failed
    - The **DRIVER OWNER** field of the *hwmgr show scsi* output is the Peripheral Driver Owner exclusive lock of the device – if not 0 (zero) it's open by a driver, can not be deleted – reboot to clear, attempts can be made using *lsof* to find and kill owning process
    - When replacing Adapters that have failed, generally no operating system maintenance is required (NIC, SCSI HBA) – assumes same model and bus slot

Recovering from Failures

- Replacing Failed Hardware Devices
  - Replacing failed SCSI Disk drive

1. Locate the failed disk under ***hwmgr show scsi*** – no valid path
2. Check DRIVER OWNER field, if non zero (2), the drive is still mounted / open – unmount the file system(s)
3. Remove the failed drive and install replacement drive
   - Must be the same model disk drive
4. Execute ***hwmgr –scan scsi*** to register new device
5. Exchange the device special files from new device to original/failed device to retain name

   dsfmgr –e dsk7 dsk3

Recovering from Failures

- Replacing Failed Hardware Devices
  - Replacing failed SCSI Tape drive

1. Locate the failed tape under ***hwmgr show scsi*** – no valid path
2. Check the DRIVER OWNER field, if non zero (4), the drive is still open by the driver.
3. Remove the failed drive and install replacement drive
   - Must be the same model tape drive
4. Execute ***hwmgr –scan scsi*** to register new device
5. Exchange the device special files from new device to original/failed device to retain name

   dsfmgr –e tape2 tape0

Recovering from Failures

- Managing Stale Paths to SCSI devices – Multi-Path
  - Hardware failures and reconfiguration of SCSI Disk and SCSI Tape subsystems cause Paths to go Stale
  - Use ***hwmgr –show scsi –full*** to review path status
  - A stale path will persist after reboot
    - Consider temporarily unavailable devices expected to return
  - Clearing Stale Paths
    - Perform during **maintenance period**
    - **First Valid Path** is Stale – must **delete scsi** device (did) , scan and use dsfmgr –m (move)
    - Not first valid path use ***hwmgr –refresh scsi*** to clear
    - If device is under control of LSM, must be removed first

Recovering from Failures

- Checking for Inconsistencies in Hardware Database
  - *hwmgr –show comp –incon -full*
    - Displays inconsistencies in the Local & Cluster databases dec_hwc_ldb, dec_hwc_cdb
    - The **–full** option provides further specifics on inconsistency
  - *dsfmgr –vV*
    - Validates the device special file databases dcdd.dat, dccd.dat, dfsl.dat, dfsc.dat
  - *dsfmgr –zx*
    - Checks for inconsistencies in all binary databases and device special file databases – executed by bcheckrc at boot, very extensive

Recovering from Failures

- Recovering / Repairing individual database files
  - Depending on extent of inconsistencies, individual databases may not be repairable, recover from a backup

    See “Hardware Management Database Recovery”
  - If file corruption use .bak file(s) to recover databases

    ls –l and *sum* to compare primary to .bak
  - Device special file inconsistencies can be repaired using dsfmgr –F command

    # *dsfmgr –vV* – shows inconsistencies

    # *dsfmgr –vFV* – Fix inconsistencies

    # *dsfmgr –vV* – Confirm inconsistencies fixed

Recovering from Failures

- Hardware Management Database Recovery
  - Best Practice
    - Recover configured databases from known good backup
  - **Proactive Disaster Recovery**

    Data saved:
    - Current backups of root /, /usr, /var
    - *sys_check –escalate* tar file
    - disklabels for all disks saved to file
    - LSM - *volsave* & *volprint –Aht*
    - Set SAN Logical Identifiers (LIDs) to same number as device name (dsk1 – ID=1) for ease of recovery
    - Use hwmgr *set* commands to further define disk use

      hwmgr set attr –id saved –a user_name="user_domain"

      hwmgr set attr –id saved –a location="volume1"

Recovering from Failures

- Recover configured databases from backup

  Standalone system
  - Standard recovery procedure for individual files
  - Fails to boot single-user - clear BOOTDEF_DEV prior to booting Operating System CD
  - Restore the following 10 hardware database files and associated .bak files
    1) /etc/dec_hw_db*
    2) /etc/dec_hwc_ldb*
    3) /etc/dec_hwc_cdb*
    4) /etc/dec_scsi_db*
    5) /etc/dec_unid_db*
    6) /etc/dec_devsw_db*
    7) /etc/dccd.*
    8) /etc/dcdd.*
    9) /etc/dfsl.* (CDSL)
    10) /etc/dfsc.*
  - Boot to single-user mode and check databases & CDSLs

    # hwmgr –show comp –incon -full

    # dsfmgr –vV

    # /sbin/bcheckrc

    # /usr/sbin/cdslinvchk

Recovering from Failures

- Recover configured databases from backup

  TruCluster system

  - Considerations when dealing with Hardware Database inconsistencies in TruCluster system
    - Maintain the **Emergency Repair Disk (ERD)** – consider storing current copies of hardware databases there along with other Disaster Recovery files
    - Keep current backups of member boot disks file systems along with cluster_root, cluster_usr, cluster_var
    - If only one member is experiencing problem, fix the one member if possible – focus on member specific files – /cluster/members/{memb}/boot_partition/etc/ and /cluster/members/{memb}/etc/dfsl.*

Recovering from Failures

- Recover configured databases from backup TruCluster system
  - Location of Hardware Database files for TruCluster
  - Member Boot Disk (CDSLs)

    /cluster/members/{memb}/boot_partition/etc/
    - dec_devsw_db*
    - dec_hw_db*
    - dec_hwc_cdb*
    - dec_hwc_ldb*
    - dec_scsi_db*
  - Cluster Root Disk

    /cluster/members/{memb}/etc/ (CDSL)
    - dfsl.*

    /etc/
    - dfsc.*
    - dccd.*
    - dcdd.*
    - dec_unid_db*

Recovering from Failures

- Recover configured databases from backup TruCluster system
  - Standard recovery procedure for individual files
  - Fails to boot single-user - clear BOOTDEF_DEV prior to booting Operating System CD
  - Restore the 10 hardware database files from backup
  - Restore files to proper member specific ./boot_partition and /etc directory
  - Reboot off boot member disk to single-user mode and check databases & CDSLs

    # hwmgr –show comp –incon -full

    # dsfmgr –vV

    # /sbin/bcheckrc

    # /usr/sbin/cdslinvchk

Questions

HP WORLD 2003
Solutions and Technology Conference & Expo

Questions ?

Interex, Encompass and HP bring you a powerful new HP World.