2300

HP-UX and Tru64 UNIX® : A Side-by-Side Comparison of Operating System Storage Configuration

Bradford Nichols
Technical Consultant
HP BCS
Transition Engineering and Consulting

HP WORLD 2003
Solutions and Technology Conference & Expo

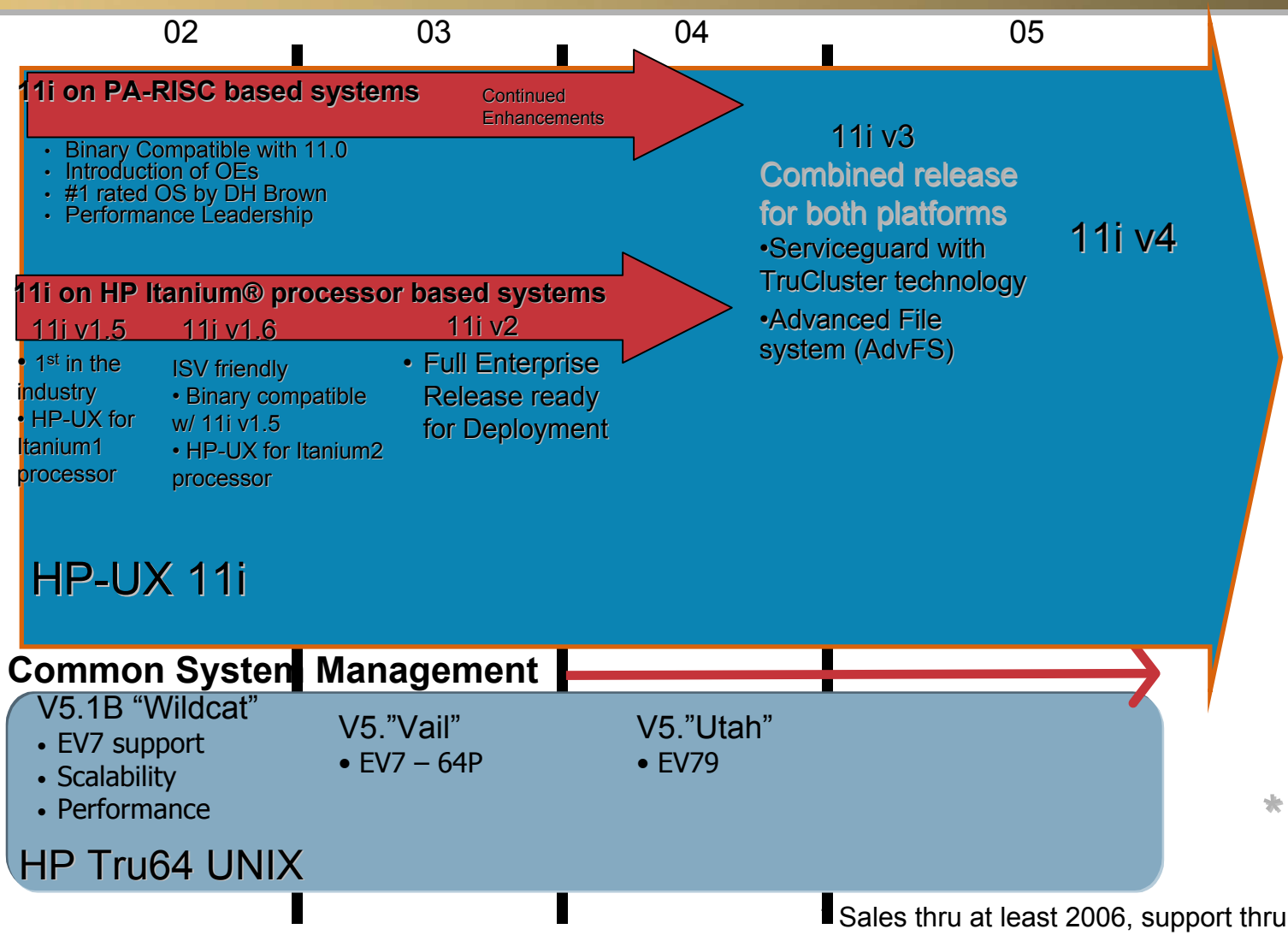
agenda

- overview and summary
- storage hw components
  - topologies
  - arrays
  - switches
  - hbas
- low level storage
  - device addressing/naming
  - driver
  - multi-pathing driver
  - disk class driver
  - device management
- high level storage
  - volume manager
  - file cache
  - file system

grounding

- both pre-merger UNIX operating systems and their associated storage hardware are capable providing cutting edge SAN-integrated, large scale fibre-based storage solutions
  - differences are largely in the “implementation details”, not in major feature discrepancies or “holes”
  - what feature differences exist will disappear as HP-UX continues to evolve and major storage related technology “goodies” from Tru64 are incorporated into HP-UX
    - primary: TCR, AdvFS
    - selected others: location independent device naming, hardware manager, unified file cache

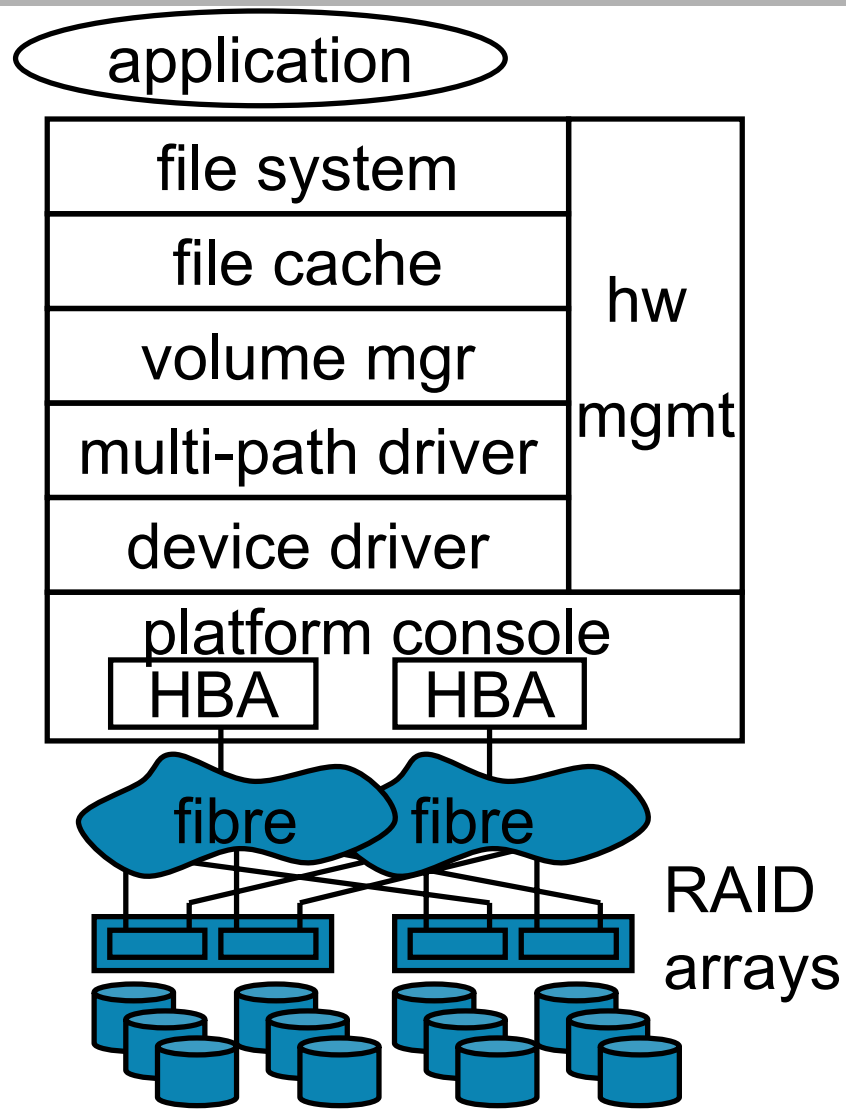
HP UNIX® operating system roadmap
HP-UX and HP Tru64 UNIX



customer value—investment protection and a better HP-UX

io stack components

HP-UX	Tru64
VxFS	AdvFS
dynamic buf cache	unified buf cache
VxVM or LVM	LSM(VxVM)
Auto Path, Secure Path,…	built-in
GIO/CDIO/WSIO	SCSI/CAM
Console	SRM
A5158A/A6795A	KGPSA-CA/DA
SW	SW
legacy AL	rarely used AL
VA7100	MSA1000
VA7410	MA/EMA
XP	EVA



HP-UX and Tru64 storage summary (1 of 3)

- storage hw components
  - both support and use fabric topologies, HP-UX has more of a AL legacy
  - array capabilities generally overlap but,…
    - (Hitachi) XP high-end from HP
    - EVA virtualization
  - switches all the same stuff underneath, but firmware requirements can be issue
  - hbas comparable (2gb,…) but different chipsets (HP-UX Tachyon, Tru64 Emulex)
- low level os layers
  - device identification and naming is different
    - ent naming: device name, id and special files are associated with a
    - iles are associated with a WWID of the device
    - nt naming: paths are ids and are the basis of special file
    - ths are ids and are the basis of special file
    - ns are ids and are the basis of special file

HP-UX and Tru64 storage summary (2 of 3)

- low level os layers, continued
  - multi-pathing
    - Tru64 implements it transparently/automatically in base os drivers
    - HP-UX relies on explicit configuration using features in a layered driver or volume manager
- console role in boot device configuration
  - Tru64 fibre boot devices must be scanned for, identified and mapped at console (wwidmgr) – console gives a fibre device a direct attached style console name “dga..” to pass to OS as what it has booted from
  - HP-UX does not require this additional configuration step – console and OS names are the same for devices (paths 0/8/2….)
- disk partition abstraction
  - eliminated from HP-UX some time ago, assumed use of volume manager

HP-UX and Tru64 storage summary (3 of 3)

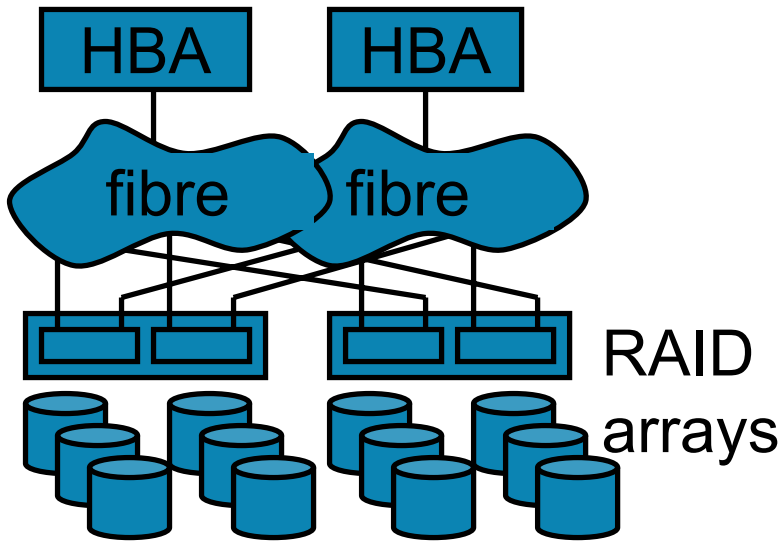
- high level os layers
  - Tru64's LSM is a derivative of Veritas's VxVM on HP-UX
  - the role and penetration rate of volume managers is different

    HP-UX solutions rely on volume managers for disk partitioning, multi-pathing and on-line resizing of file system, Tru64 platforms don't because this functionality is available elsewhere in the stack
- both have journal based file systems for fast recovery
  - storage model and resulting approach to on-line growth is different

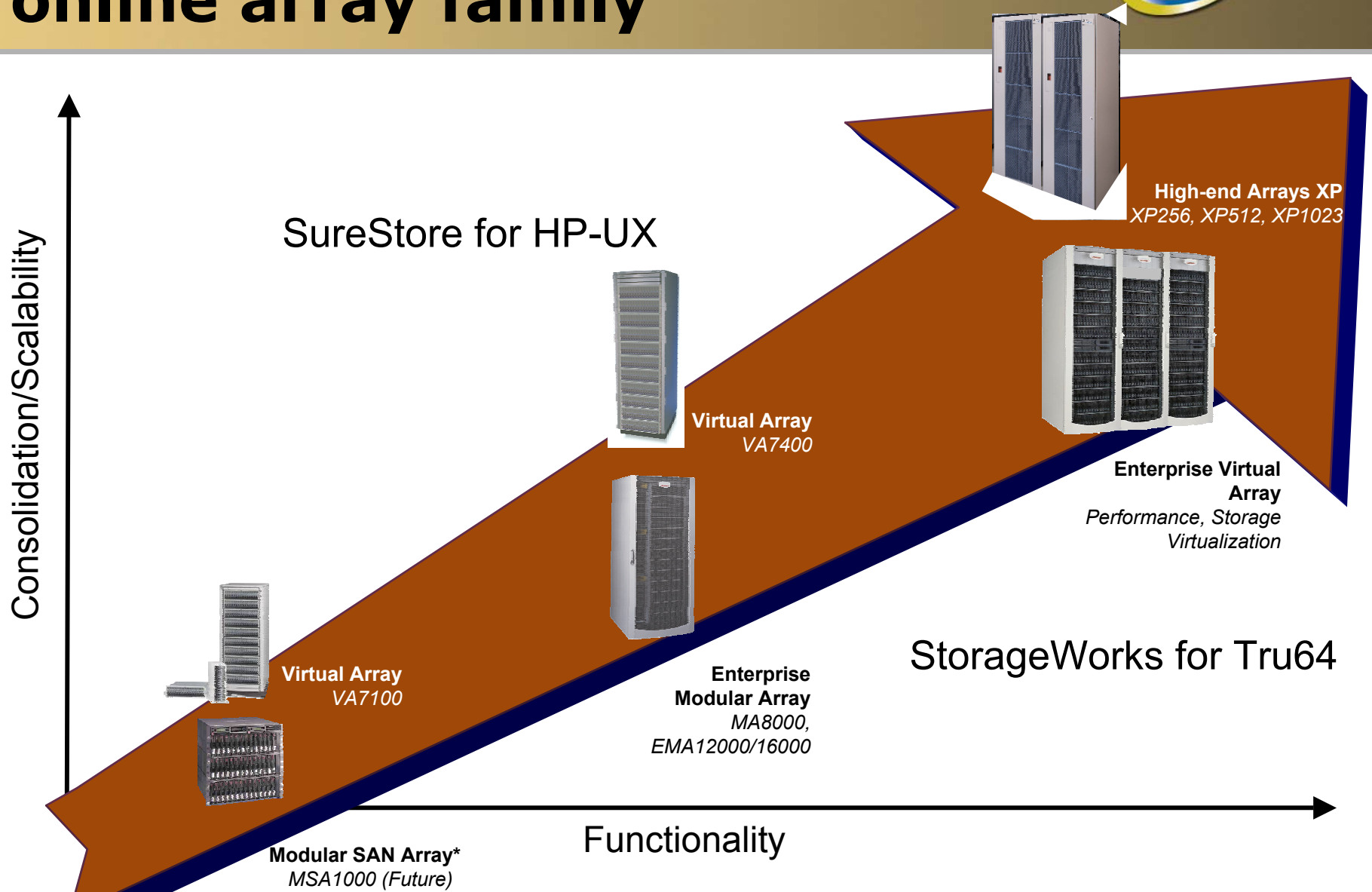
    multi-volume for AdvFS, dynamic resizing volumes for VxFS
  - large file support is transparent in Tru64, not transparent in HP-UX
- both have dynamically sized file caches
  - Tru64's cache is "unified" for a common view between memory mapped and buffered I/O access to the same file

storage components

- topics
- arrays
- switches
- HBAs



online array family



switches

Switch Supplier	SureStore (by HP-UX)	Storage Works (Tru64)
Brocade 2210	Not Supported	Not Supported
Brocade 2050	Not Supported	FC SAN Switch 8-EL
Brocade 2250	Not Supported	FC SAN Switch 16-EL
Brocade 2400	2400 Silkworm	FC SAN Switch/8
Brocade 2800	2800 Silkworm	FC SAN Switch/16
Brocade 6400 (64 port)	SureStore FC6164	SAN Switch Integrated 64
Brocade 6400 (32 port)	“depopulated” FC6164	SAN Switch Integrated 32
Brocade 3250	SureStore FC8B	FC SAN Switch 2/8
Brocade 3240	SureStore FC8B entry	FC SAN Switch 2/8-EL
Brocade 3600	Not Supported	FC SAN Switch 2/16-EL
Brocade 3800	SureStore FC16B	FC SAN Switch 2/16
McData ES-1600	F16	Not Supported
McData ES-1000	Not Supported	Supported; Not Resold
McData ES-3016	Supported; Sold by HP-CP; Serviced by McData	Supported; Not Resold
McData ES-3032	Supported; Sold by HP-CP; Serviced by McData	Supported; Not Resold
McData ES-5000	Supported; not resold	Supported; Not Resold
McData ED-6064	SureStore FC64	SAN Director 64

Tru64 UNIX
FC host bus adapters

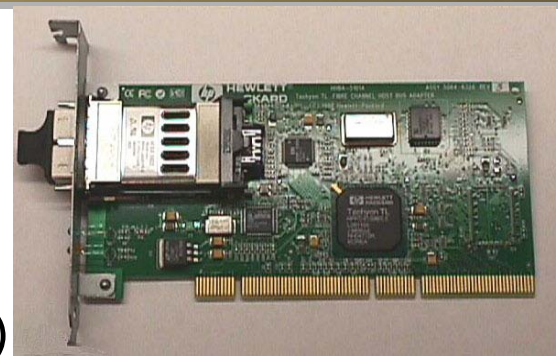
HP WORLD 2003
Solutions and Technology Conference & Expo

- DS-KGPSA-CA (aka Emulex LP8000)

  - fabric capable
    - 5V, 64-bit/33MHz PCI adapter
    - Supports 1Gb fabrics
    - Supports DS, ES and GS series of platforms
      DS10, DS20, DS25, ES40, ES45, GS80, GS160, GS320
      Full boot/dump support in Tru64 UNIX kernel
- FCA2354 (DS-KGPSA-DA – aka Emulex LP9002L)

  - fabric capable
    - 3.3V or 5V, 64-bit/66MHz PCI adapter
    - Supports 2Gb fabrics
    - Supports DS, ES and GS series of platforms
      DS10, DS20, DS25, ES40, ES45, GS80, GS160, GS320
      Full boot/dump support in Tru64 UNIX kernel

HP-UX
FC host bus adapters



- A5158A
  - fabric-capable
    - uses Agilent's Tachyon TL (Tach Lite) chip
    - supports 1Gb speeds
    - for PCI-bus servers (PA-RISC and IPF Servers)
      rp24xx, rp54xx, rp7400, v2200/2250/2500/2600, rp8400, rp7410, superdome
      rx4610, rx9610

- A6795A
  - fabric-capable
    - use Agilent's Tachyon XL2 chip
    - 2Gb support
    - for PCI-bus servers/workstations (PA-RISC)
      rp7400, rp54xx, rp24xx, rp8400, rp7410; Superdome
    - for PCI-bus servers/workstations (IPF)
      rx5670, rx2600, rx9610, rx6410, zx6000, zx2000

storage components: Tru64 migration

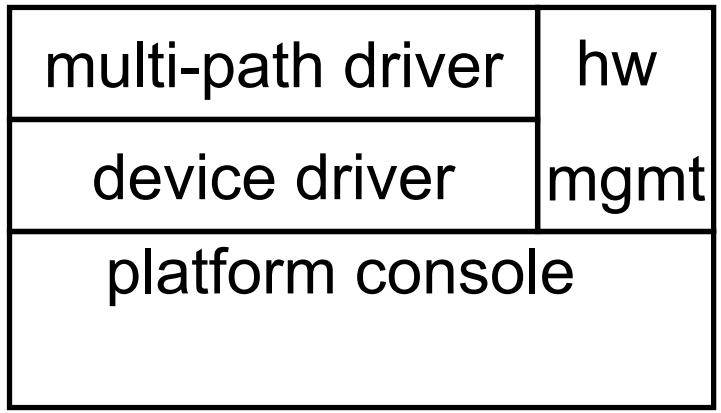
- re-use potential for StorageWorks arrays and switches
- StorageWorks HSG and EVA raid arrays
  - supported on Tru64
  - supported by Compaq on HP-UX prior to merger
- StorageWorks switches
  - many commonly supported, resold Brocade and MCData switches
  - firmware qual levels could be issue

new opportunities

- XP arrays
  - supported with HP-UX
  - Hitachi and HP claimed support on Tru64 prior to merger, new-HP plans formal (re)qual for Tru64

low level operating system layers

- topics
- role of console
- device naming
- device driver
- multi-pathing
- disk partitioning
- hardware management



platform console role in configuring fibre storage (1 of 2)

- some functions can be done in both consoles
  - register default boot device and boot flags
  - verify HBA installation in platform
- Tru64 AlphaServer has fibre specific functions (wwidmgr)
  - required to locate and register (map) fibre devices that can be used for booting or swap (wwidmgr)
    - side affect use to verify connectivity to storage devices in fabric
  - set HBA configuration parameters
- how do they do without this functionality on PA-RISC boxes?
  - platform and os have common device naming scheme
  - rely on probe/scan functionality is built into os including installation image

platform console role in configuring fibre storage (2 of 2)

- high level operational difference
  - HP-UX: rather then check for connectivity or configure settings on at console, boot OS and perform tasks from booted os
  - Tru64: must register/map boot and swap fibre devices at console before use when installing fibre storage can use console as a connectivity check before booting

platform console fibre storage related command comparison

- Tru64 console storage related commands
  - **`show config`**
    - list all components
      includes SCSI and mapped FC HBAs
  - **`show dev[ices]`**
    - list block devices
      disks, tapes, CDROM, Floppy, etc
      includes SCSI and mapped FC devices
  - **`wwidmgr`**
    - probe for fibre storage
    - map HBA ports, controller ports, FC disks
    - set fibre hba settings
    - a “must” to be able to boot from FC disks

- HP-UX storage related console commands
  - **`path`**
    - list or modify path to a device
  - **`sea[rch]`**
    - scan IO busses for direct connected or previously known devices and their paths
  - **`in[formation] -> io`**
    - list hw components and devices

Tru64 Console
listing direct attached and mapped scsi disks

```
P00>>>show dev
dka0.0.0.1.1             DKA0              RZ2CA-LA  N1H0
dka100.1.0.1.1           DKA100            RZ2CA-LA  N1H0
dkb100.1.0.2.1           DKB100            RZ26F  1Q0A
dkb400.4.0.2.1           DKB400            RZ28  X442
dqa0.0.0.15.0            DQA0              TOSHIBA CD-ROM XM-6302B  1012
dva0.0.0.1000.0          DVA0
ewa0.0.0.6.1             EWA0                00-00-F8-10-DF-3A
pga0.0.0.3.1             PGA0              WWN 2000-0000-c921-0d00
pgb0.0.0.5.1             PGB0              WWN 1000-0000-c920-cd9c
pgc0.0.0.1.0             PGC0              WWN 1000-0000-c920-a7ae
pgd0.0.0.2.0             PGD0              WWN 2000-0000-c921-07c4
pka0.7.0.1.1             PKA0              SCSI Bus ID 7
pkb0.7.0.2.1             PKB0              SCSI Bus ID 7  5.57
P00>>>
```

In this example, no fibre disks have been mapped yet and only direct attached scsi disks appear in the listing

Tru64 UNIX Console
setting a WWID in NV memory

scan for fibre devices

```
P00>>>wwidmgr -show wwid
[0] UDID:5 WWID:01000010:6000-1fe1-0000-0cb0-0009-9130-8234-003a
[1] UDID:4 WWID:01000010:6000-1fe1-0000-0cb0-0009-9130-8234-0039
[2] UDID:3 WWID:01000010:6000-1fe1-0000-0cb0-0009-9130-8234-0038
[3] UDID:2 WWID:01000010:6000-1fe1-0000-0cb0-0009-9130-8234-0037
[4] UDID:1 WWID:01000010:6000-1fe1-0000-0cb0-0009-9130-8234-0036
[5] UDID:-1 WWID:01000010:6000-1fe1-0000-0cb0-0009-9130-8234-004
```

map disk with udid 5

```
P00>>>wwidmgr -quickset -udid 5
Disk assignment and reachability after next initialization:
```

wwid (wwn)

```
6000-1fe1-0000-0cb0-0009-9130-8234-003a
```

paths (4)

```
                                via adapter:          via fc nport:
dga5.1001.0.3.1                 pga0.0.0.3.1          5000-1fe1-0000-0cb
dga5.1002.0.3.1                 pga0.0.0.3.1          5000-1fe1-0000-0cb
dgd5.1001.0.2.0                 pgd0.0.0.2.0          5000-1fe1-0000-0cb
dgd5.1002.0.2.0                 pgd0.0.0.2.0          5000-1fe1-0000-0cb
```

Tru64 UNIX Console
accessing a fibre boot disk

mapped disk (udid 5) now appears

```
P00>>> init
…
P00>>> show dev
dga5.1001.0.3.1          $1$DGA5          HSG80  V85F
dga5.1002.0.3.1          $1$DGA5          HSG80  V85F
dgd5.1001.0.2.0          $1$DGA5          HSG80  V85F
dgd5.1002.0.2.0          $1$DGA5          HSG80  V85E
dka0.0.0.1.1             DKA0             RZ2CA-LA  N1H0
dka100.1.0.1.1           DKA100           RZ2CA-LA  N1H0
dkb100.1.0.2.1           DKB100           RZ26F  1Q0A
dkb400.4.0.2.1           DKB400           RZ28  X442
dqa0.0.0.15.0            DQA0             TOSHIBA CD-ROM XM-6302B
  1012
dva0.0.0.1000.0          DVA0
ewa0.0.0.6.1             EWA0                    00-00-F8-10-DF-
pga0.0.0.3.1             PGA0             WWN 2000-0000-c921-0d00
pgb0.0.0.5.1             PGB0             WWN 1000-0000-c920-cd9c
…
P00>>>
P00>>>set bootdef_dev dga5.1001.0.3.1, dga5.1002.0.3.1,
  dgd5.1001.0.2.1, dgd5.1002.0.2.1
```

4 paths

register disk as boot device

PA-RISC Console - sea[rch]

```
Main Menu: Enter command or menu > sea

Searching for potential boot device(s)
This may take several minutes.
To discontinue search, press any key (termination may not be immediate).

   Path#  Device Path (dec)  Device Path (mnem)  Device Type
   -----  -----------------  ------------------  -----------
   P0     0/0/1/1.2          intscsib.2          Random access media
   P1     0/0/1/1.0          intscsib.0          Random access media
   P2     0/0/2/0.2          intscsia.2          Random access media
   P3     0/0/2/0.0          intscsia.0          Random access media
   P4     0/0/2/1.2          media.2             Random access media
   P5     0/3/0/0.8                              Random access media
   P6     0/6/0/0.8                              Random access media


Main Menu: Enter command or menu >
```

PA-RISC Console in[formation]->io

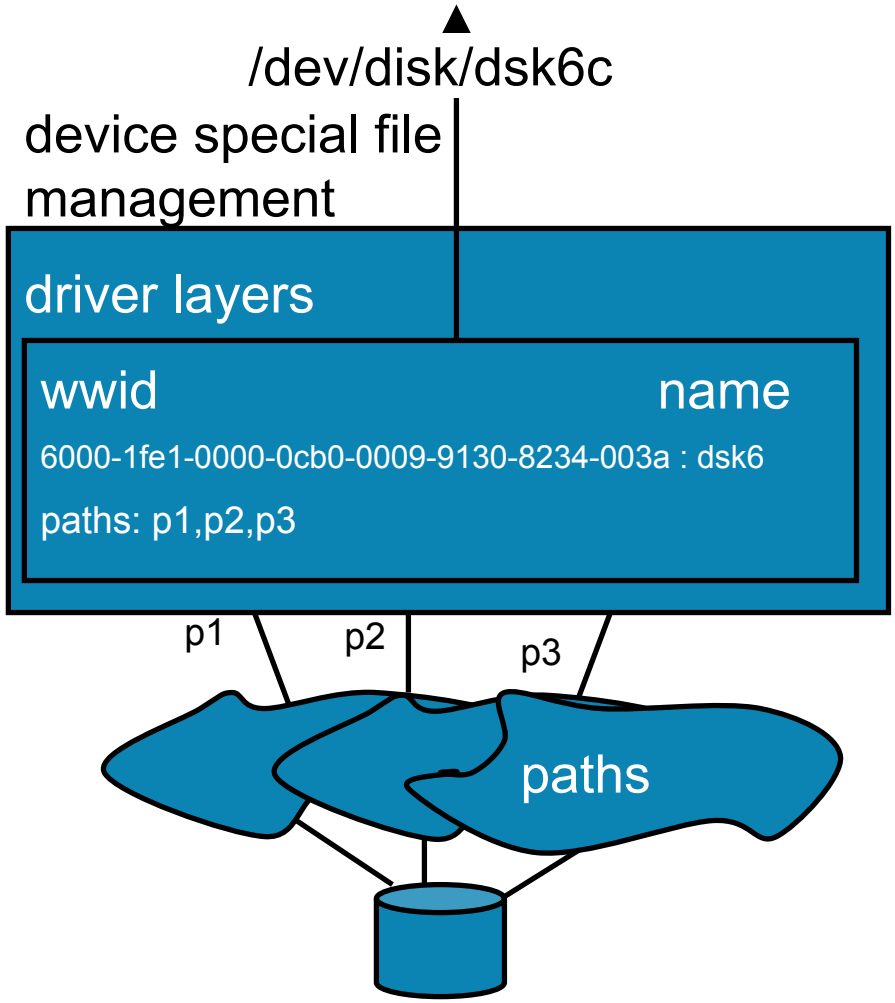
…

PCI DEVICE INFORMATION

Description	Path (dec)	Vendor Id	Device Id	Bus #	Slot #
Ethernet cntlr	0/0/0/0	0x1011	0x19	0	1
SCSI bus cntlr	0/0/1/0	0x1000	0xb	0	1
SCSI bus cntlr	0/0/1/1	0x1000	0xb	0	1
SCSI bus cntlr	0/0/2/0	0x1000	0xf	0	1
SCSI bus cntlr	0/0/2/1	0x1000	0xf	0	1
Comp. ser cntlr	0/0/4/0	0x103c	0x1048	0	2
Comp. ser cntlr	0/0/5/0	0x103c	0x1048	0	2
SCSI bus cntlr	0/2/0/0	0x1000	0xb	16	8
SCSI bus cntlr	0/2/0/1	0x1000	0xb	16	8
Fibre channel	**0/3/0/0**	**0x103c**	**0x1028**	**24**	**10**
Ethernet cntlr	0/4/0/0/4/0	0x1011	0x9	33	12
Ethernet cntlr	0/4/0/0/5/0	0x1011	0x9	33	12
Ethernet cntlr	0/4/0/0/6/0	0x1011	0x9	33	12
Ethernet cntlr	0/4/0/0/7/0	0x1011	0x9	33	12
PCI-to-PCI bridge	0/4/0/0	0x1011	0x24	32	12
Fibre channel	**0/6/0/0**	**0x103c**	**0x1028**	**48**	**9**
Ethernet cntlr	0/7/0/0/4/0	0x1011	0x9	57	11
Ethernet cntlr	0/7/0/0/5/0	0x1011	0x9	57	11
Ethernet cntlr	0/7/0/0/6/0	0x1011	0x9	57	11
Ethernet cntlr	0/7/0/0/7/0	0x1011	0x9	57	11
PCI-to-PCI bridge	0/7/0/0	0x1011	0x24	56	11

…

Tru64 UNIX device identification
WWIDs -> hw ids -> special file



identifying a storage device

- 1 device has 1 special file (name)
  - no matter how many paths
  - no matter how paths come and go over time
- Tru64 UNIX makes use of WWID's within driver layers (CAM)
  - records in a database for each disk: wwid, paths, hw id, device name
- how CAM collects WWID's is specified in the DDR database.
  - depends on each peripheral
  - Tru64 support WWID's for devices that do not "have them"
    - common for devices >5 years old
    - generate WWID's using the device serial no, TOD and/or device string

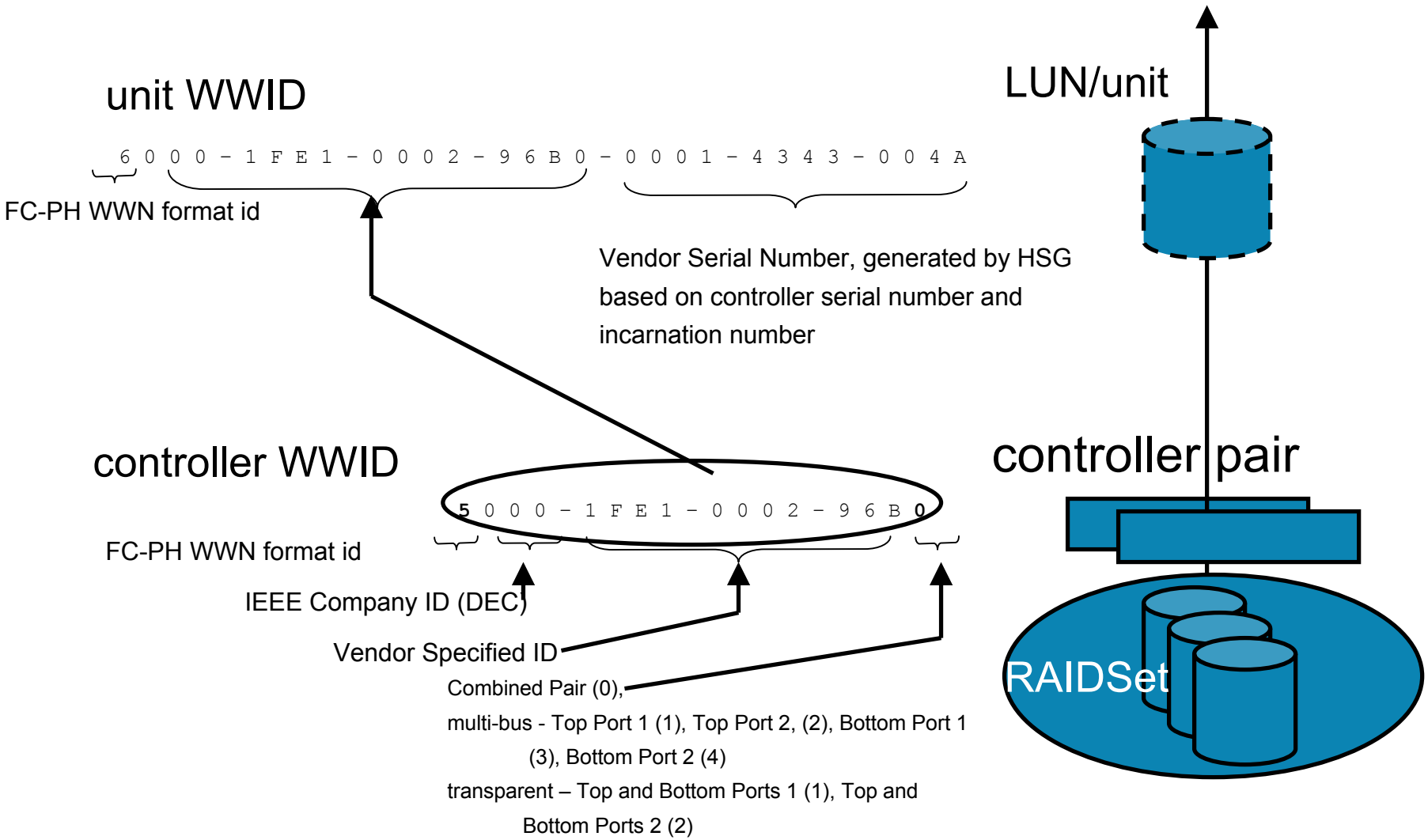
Tru64 Unix device identification WWID

- example of an new wwid capable device

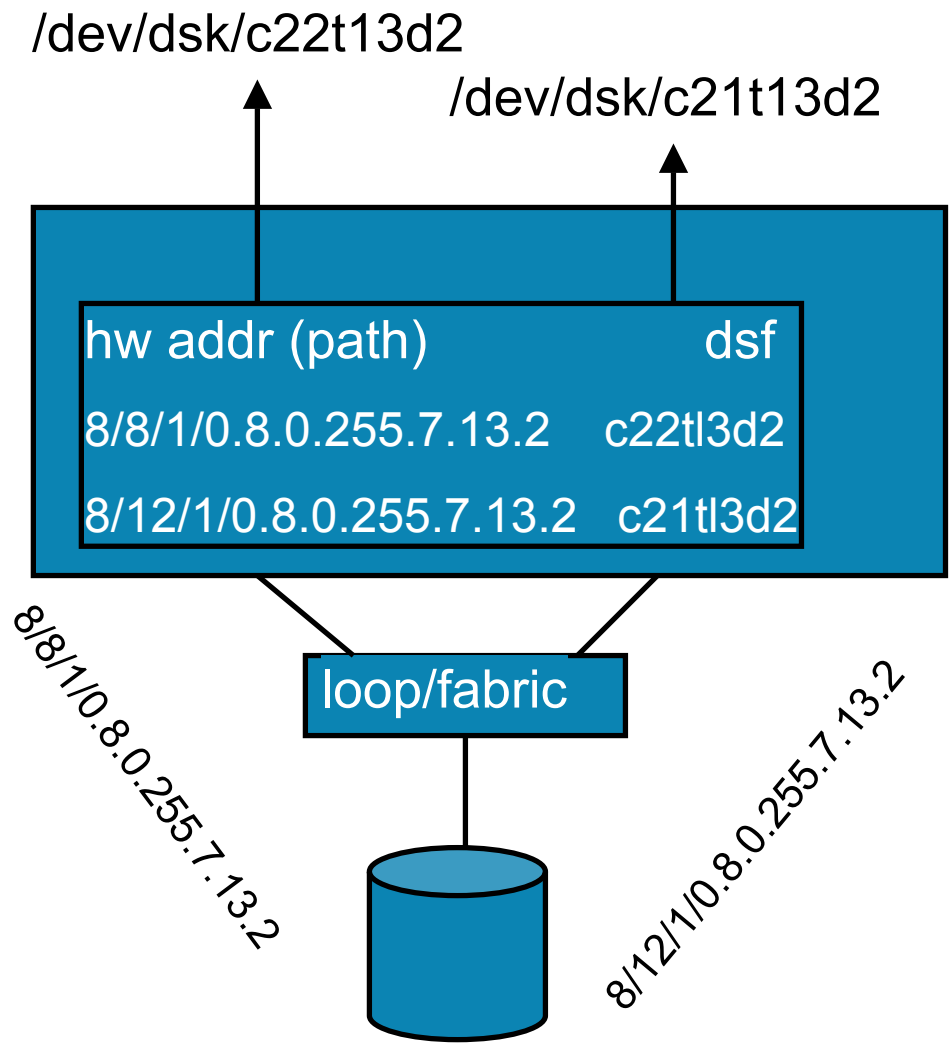
```
host1 > hwmgr -get attr -id 133
133:
  name = SCSI-WWID:01000010: 6000-1fe1-0000-0cb0-0009-9130-8234-
003a
  category = disk
  sub_category = generic
  architecture = SCSI
  phys_location = IDENTIFIER=5
  dev_base_name = dsk17
  capacity = 17768677
  block_size = 512
  model = HSG80
  boot_capable = 1
```

- this disk will always be "dsk17" no matter how many paths exist to it or how they come and go overtime
- but,… admin can change the name at anytime if needed/desired

Tru64 UNIX device identification
Raidset WWIDs



HP-UX device identification
path identification



- devices are identified by their hardware address (path) – of which they may have more then one
- each path will have its own special files
- hardware address (path) format will depend on device topology (3 types) and connection type (3 types)
- 3 topologies times 3 connection types equals nine ways a device address may be represented

HP-UX: a disk has multiple hw addresses and special files

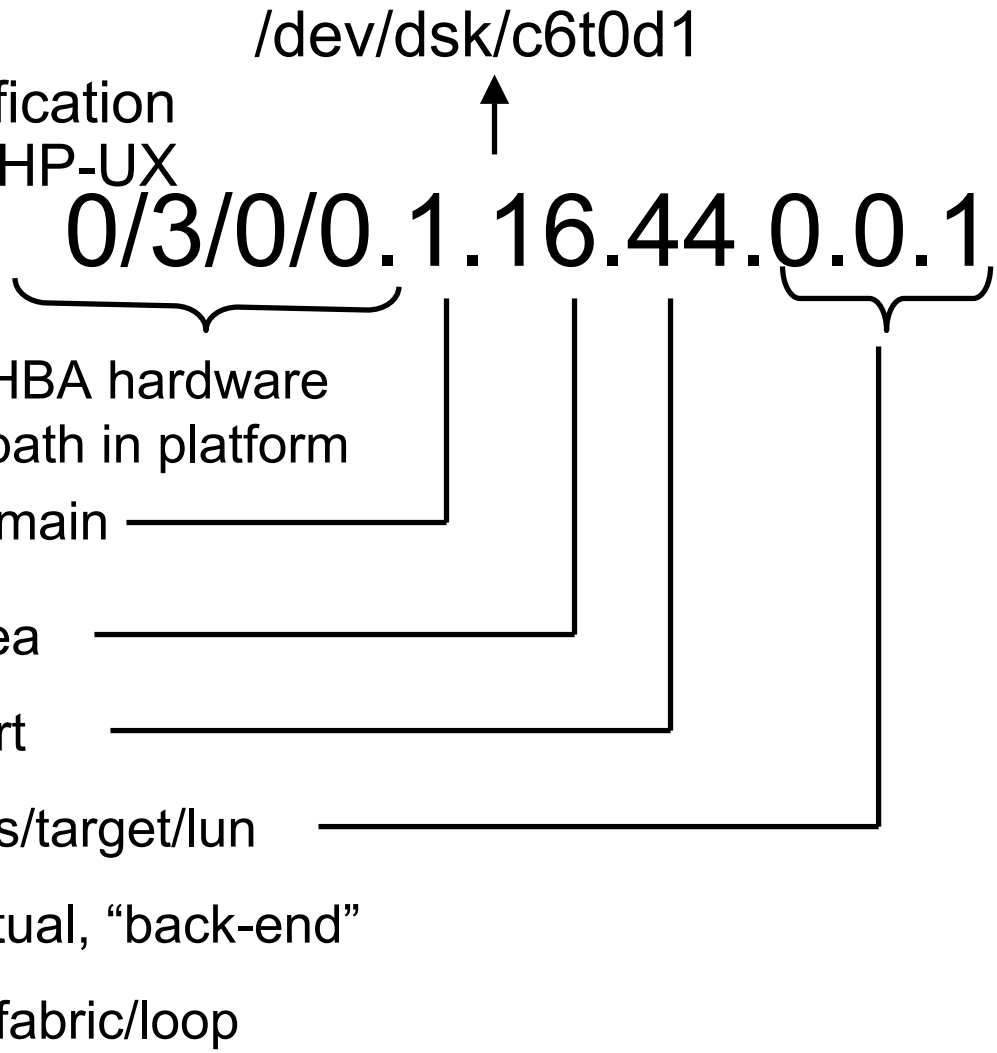

```
# ioscan -fn -C disk
Class      I  H/W Path          Driver      S/W State   H/W Type     Description
=============================================================================

…
disk       5  0/3/0/0.1.16.44.0.0.0     sdisk    CLAIMED     DEVICE       HP
                       /dev/dsk/c6t0d0   /dev/rdsk/c6t0d0
disk       6  0/3/0/0.1.16.44.0.0.1     sdisk    CLAIMED     DEVICE       HP
                       /dev/dsk/c6t0d1   /dev/rdsk/c6t0d1
disk       7  0/3/0/0.1.16.44.0.0.2     sdisk    CLAIMED     DEVICE       HP
                       /dev/dsk/c6t0d2   /dev/rdsk/c6t0d2
…
disk      13  0/6/0/0.1.30.43.0.0.0     sdisk    CLAIMED     DEVICE       HP
                       /dev/dsk/c8t0d0   /dev/rdsk/c8t0d0
disk      14  0/6/0/0.1.30.43.0.0.1     sdisk    CLAIMED     DEVICE       HP
                       /dev/dsk/c8t0d1   /dev/rdsk/c8t0d1
disk      15  0/6/0/0.1.30.43.0.0.2     sdisk    CLAIMED     DEVICE       HP
                       /dev/dsk/c8t0d2   /dev/rdsk/c8t0d2
…
```

these could be the same disk

HP-UX disk hardware path formats

- The unique identification of a fibre device in HP-UX is a path



HP-UX FC addressing matrix

	adapter	domain	area	port	bus	target	lun
Private Loop							
LUA	HBA	8	0	Loop_id	Bus	Target	LUN
VSA	HBA	8	0	Loop_id	Q of LUN 7bit	Q of LUN 4bit	Q of LUN 3bit
PDA	HBA	8	0	255	Q of Loop_id	Q of Loop_id	0
Public Loop							
LUA	HBA	SW #	SW port #	AL-PA	Bus	Target	LUN
VSA	HBA	SW #	SW port #	AL-PA	Q of LUN 7bit	Q of LUN 4bit	Q of LUN 3bit
PDA	HBA	SW #	SW port #	255	Q of AL-PA	Q of AL-PA	0
Fabric							
LUA	HBA	SW #	SW port #	AL-PA	Bus	Target	LUN
VSA	HBA	SW #	SW port #	0	Q of LUN 7bit	Q of LUN 4bit	Q of LUN 3bit
PDA	HBA	SW #	SW port #	255	Q of Fport	Q of Fport	0

examples

```
0/ 3/0/0.1.16. 44.0. 0.1

8/12/1/0.8. 0.255.7.13.3
```

addressing matrix decoder

- Q of Loop_id: take the quartets of the Loop_id

if loop id = decimal 43

1. convert to 8-bit binary: 00101011
2. separate into two, 4bit quartets: 0010 1011
3. convert each quartet to its decimal value: 0010 = 2, 1011 = 7

the first quartet 2 is the Bus, the second quartet 7 is the Target

- Q of AL-PA: convert the AL_PA to binary, separate into two, 4bit quartets, convert each quartet into its decimal value and insert in the Bus and Target fields
- Q of LUN: convert the LUN number into a 14bit binary number:

decimal 179 becomes binary 00000010110011

separate into three fields of 3bits, 4bits and 7bits: 0000001 0110 011

the seven bit field is the Bus, the 4bit field the Target and the 3bit field is the LUN: 1.6.3

fibre HBA device drivers

	Tru64	HP-UX
name	emulex (emx)	fibre channel mass storage (fcms)
hw architecture	emulex - LP7000, LP8000,	agilent tachyon, tachlite
management as software	shipped in with base os distribution	shipped in with base os distribution
	standard kernel build tool will statically link into kernel if HBA is detected	statically linked into kernel when kernel build files modified
	updated through patch kits	updated through patch kits
tools	sysconfig	ioscan
	emxmgr	fcmsutil

Tru64 emxmgr example

```
# emxmgr -d

The available adapter instances are:
    emx0       emx2      emx3      emx4
foo> emxmgr -t emx0

emx0 state information:
  Link :  connection is UP
          Point to Point
          Fabric attached
          FC DID 0x210513
  Link is SCSI bus 2 (e.g. scsi2)
          SCSI target id 255
          portname is 1000-0000-C921-0D00
          nodename is 2000-0000-C921-0D00
```

Tru64 emxmgr example (cont)


```
N_Port at FC DID 0x210113 - SCSI tgt id 0 :
  portname 5000-1FE1-0000-0CB4
  nodename 5000-1FE1-0000-0CB0
  Present, Logged in, FCP Target, FCP Logged in,
 N_Port at FC DID 0x210213 - SCSI tgt id 1 :
  portname 5000-1FE1-0000-0CB2
  nodename 5000-1FE1-0000-0CB0
  Present, Logged in, FCP Target, FCP Logged in,
 N_Port at FC DID 0x210613 - SCSI tgt id 254 :
  portname 1000-0000-C921-07C4
  nodename 2000-0000-C921-07C4
  Present, Logged in, FCP Initiator, FCP Target, FCP Logged in,
 N_Port at FC DID 0xfffffc - SCSI tgt id -1 :
  portname 20FC-0060-6900-0428
  nodename 1000-0060-6900-0428
  Present, Logged in, Directory Server,
 N_Port at FC DID 0xfffffe - SCSI tgt id -1 :
  portname 2005-0060-6900-0428
  nodename 1000-0060-6900-0428
  Present, Logged in, F PORT,
```

HP-UX checking installed HBA state

- fcmsutil

```
# ls /dev/td*
/dev/td0  /dev/td1
# fcmsutil /dev/td0

                         Vendor ID is = 0x00103c
                         Device ID is = 0x001028
                TL Chip Revision No is = 3.0
           PCI Sub-system Vendor ID is = 0x00103c
                  PCI Sub-system ID is = 0x000006
                             Topology = PRIVATE_LOOP
                    Local N_Port_id is = 0x000002
                      Local Loop_id is = 124
           N_Port Node World Wide Name = 0x50060b00000900d1
           N_Port Port World Wide Name = 0x50060b00000900d0
                         Driver state = ONLINE
                    Hardware Path is = 8/8/1/0
               Number of Assisted IOs = 46
      Number of Active Login Sessions = 0
```

Tru64 loading emx driver

- Tru64 kernel build utility is `doconfig`
- doconfig by default runs a subscript called `sizer` which provides input to create a template kernel build definition file
  - `/sys/conf/HOSTNAME`
- `sizer` probes for attached storage (via hwmgr) and with `doconfig` places necessary drivers in kernel built file
- so, when doing initial install of a system with a fibre HVA or after adding first HBA:
- `doconfig` will detect hardware, build kernel including driver
- to confirm the os “sees” HBA
  - from console : `show config`
  - from booted os : `hwmgr –view hier`

HP-UX configuring fcms into a kernel: cli

- GUI – automated in SAM
- command line

verify that device is seen by kernel using iostat –f

- "s/w state" of the device is "claimed"
- means the device driver is loaded and in use

create a kernel build configuration file containing the attributes of the currently running kernel

- `/usr/lbin/sysadm/system_prem –s system`

add the driver token to the file

- `/usr/sbin/kmsystem -S /stand/build/system -c Y driver-name`

verify that the driver is listed in /stand/build/system

build kernel

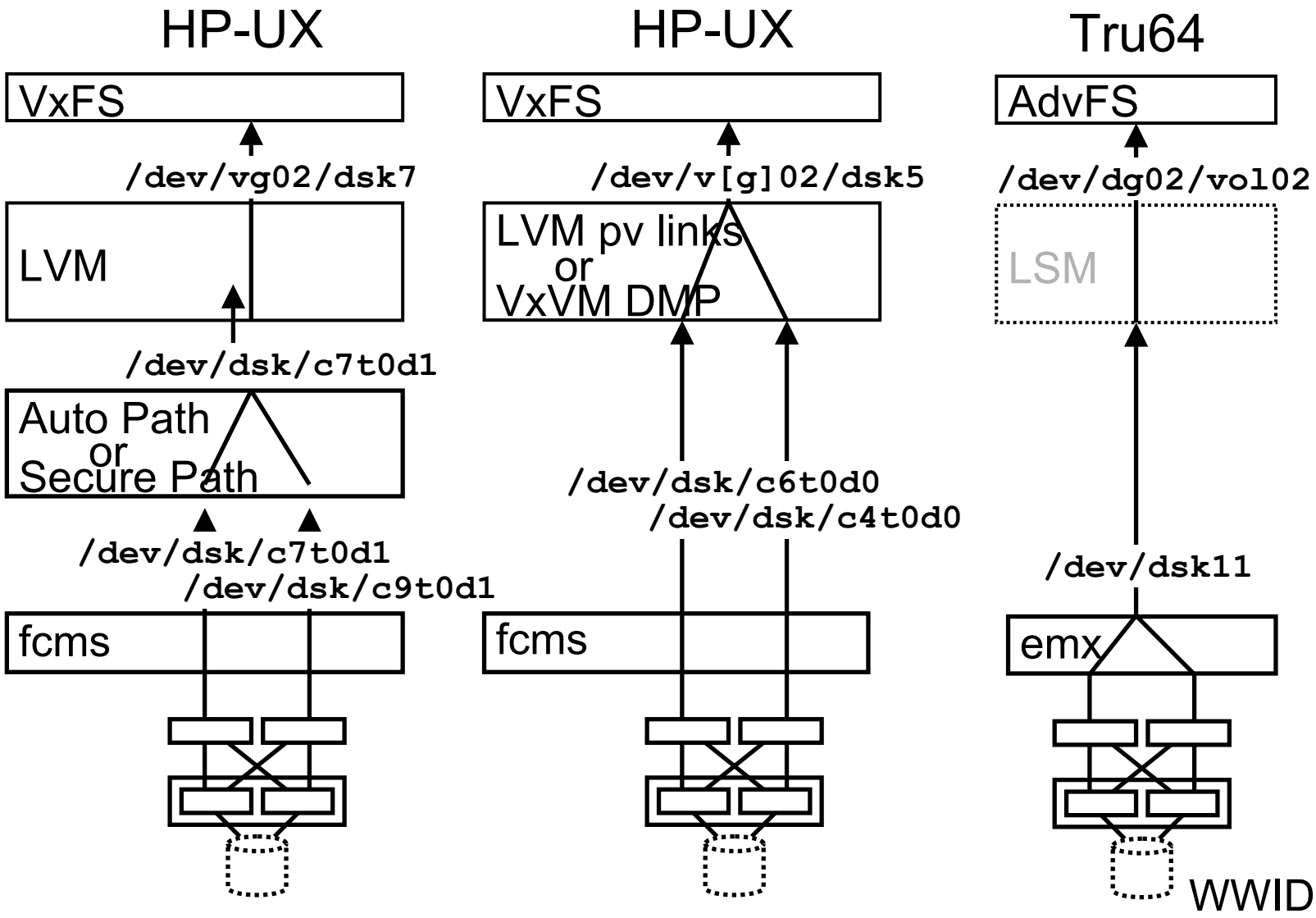
- `/usr/sbin/mk_kernel –s \ /stand/build/system`

move new kernel and config file

- `mv /stand/build/system\ /stand/system`
- `mv /stand/build/vmunix \ /stand/vmunix`

reboot

io stack: multi-pathing options



multi-path capabilities

Description	HP-UX SureStore AutoPath	HP-UX LVM PVLinks	HP-UX VxVM DMP	Tru64 UNIX CAM
fail Over: automatically fails over to an alternate path when the primary path is no longer available	√	√	√	√
fail Back: automatically recognizes the newly available path when a failed path comes back up alive.	√		√	√
static load-balancing: Balances I/O load to a LUN among all available paths with user selectable load-balancing policies.	√		√[2]	
dynamic load-balancing System automatically balances I/O load among all available paths to a LUN based on run-time statistics such as device queue depths etc.	√			√

[1]Full version of VxVM , [2]Does not provide user selectable load-balancing policies.

disk partitions/sections

- Tru64 UNIX disk abstraction allows dividing a disk into multiple partitions for separate use
  - independent of a volume manager or file system
  - partition table stored in first physical block
  - 8 partitions identified a thru h
  - default “label” or assignment
  - administered through `disklabel` command
- overlaps functionality in volume managers and hw arrays
- HP-UX has no equivalent today
- now assumed use of volume manager to provide roll
- early versions allowed disk “sections” to be used
  - still traces in default special files created for disks
    - c*X*t*X*l*X*[s*X*]

hardware management

- both Tru64 and HP-UX have the ability to track and name hardware devices including end storage disks
- Tru64 “hardware management”
- more formalized
  - “registry” style
  - cluster SSI aware: single name space, io forwarding
  - scalable
  - multi-pathing aware, paths are an attribute of devices
  - rich APIs for other OS components and user-mode tools
- being considered as “selected other technologies” to be ported into HP-UX

Tru64 hw management role and tools

device discovery, configuration and attribute repository

- automated device discovery

  during boot process

  or by automated polling of FC busses
- automated loading of drivers based on connected devices
- device special files created automatically

`hwmgr`

- used to view capabilities, connectivity and properties of devices
- used to modify properties of a disk device

`dsfmgr`

- used to view, verify and manage HW to DSF mappings

Tru64 hwmgr - hw management tool

```
# hwmgr -view hier
HWID:   hardware hierarchy
-------------------------------------------------------
   1:   platform AlphaServer 1000A 5/400
   2:     cpu CPU0
   6:     bus pci0
   9:       connection pci0slot8
  28:         bus pci1
  29:           connection pci1slot0
  39:             scsi_adapter isp0
  40:               scsi_bus scsi0
  48:                 disk bus-0-targ-1-lun-0 dsk0
  49:                 disk bus-0-targ-4-lun-0 cdrom0
  50:                 disk bus-0-targ-14-lun-0 dsk1
  31:           connection pci1slot1
  41:             scsi_adapter pza0
  42:               scsi_bus scsi1
  51:                 disk bus-1-targ-1-lun-0 dsk2
```

Tru64 hw mgmt: the files behind the scene

- common /etc
  - dec_devsw_db[.bak]
  - dec_hw_db[.bak]
  - dec_hwc_ldb[.bak]
  - dec_scsi_db[.bak]
  - dec_hwc_cdb[.bak]
  - dec_unid_db[.bak]
  - dfsc.dat | bak | h00
  - dccd.dat | bak | h00
  - dcdd.dat | bak | h00
- (cluster root) Member Specific /cluster/…/etc
  - dfsl.dat | bak | h00
  - cfginfo

HP-UX hardware management

- all hardware components identified by a hardware path
- logical hardware topology tree starting with top level of the hierarchy a

```
# ioscan
H/W Path       Class                                      Description
=============================================================
               bc
8              bc                                         I/O Adapter
8/0                ext_bus                                GSC add-on Fast/Wide
8/0.4                 target
8/0.4.0                  ctl                              Initiator
…
8/8                ba                                     GSCtoPCI Bridge
8/8/1/0                  fc                               HP Tachyon TL/TS Fib
…
10/12/6                lan                                Built-in LAN
10/12/7                ps2                                Built-in Keyboard/Mo
…
4/6            processor                                  Processor
```

HP-UX
auto-configuring a device

- device detection and device specific file creation at system boot
  - runs /sbin/ioinitrc from /etc/inittab at sysinit
- a few details on /sbin/ioinitrc
  - at start, verifies whether this is a diskless boot or not

    if diskless, only scans for new network/console/terminal/etc devices

    i.e. *not* disks… ☺

    if we're booting from a local disk device

    issues /sbin/ioinit –i –r

    options:

    -i = invoke /sbin/insf if there's a discrepancy between kernel loaded drivers and info stored in /etc/ioconfig

    -r = reboot to correct any inconsistencies between kernel device structures and contents of /etc/ioconfig

/etc/ioconfig, /stand/ioconfig

- used to retain information on system's IO configuration across reboots
- contains two types of information:
  - mappings of dynamically allocated major numbers to drivers
  - mappings of instance numbers to hardware paths
- at boot time this file is read and the information is stored in the io_tree kernel data structure
- file is created by insf at install time
  - modified by insf, rmsf, and ioscan when devices are added or removed
- the only purpose of the ioconfig file to maintain configuration information when the system is not running
  - while the system is running, all accesses are made directly to the kernel io_tree structure, although any tools that change the kernel structures must also keep ioconfig consistent

low level os layers: Tru64 migration

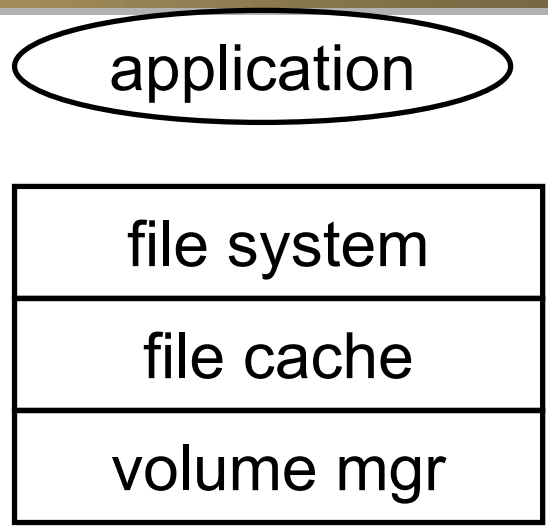
- short term
- solution architects
  - HP-UX device management and multi-pathing options meet solution needs in availability and performance
  - note: storage array hardware choice may dictate multi-pathing. example EVA requires Secure Path on HP-UX
- admins
  - many differences in implementation and ease of configuration

    but, path naming and multi-pathing in special driver are familiar concepts from Tru64 4.X
- future: HP-UX with TCR/AdvFS
  - will meet solution needs with Tru64-style ease of management

high level operating system layers

topics

- volume managers
  - options
  - VxVM and LSM
  - VxVM(LSM) and LVM
- file system caches
  - dbc and ubc
- file systems
  - VxFS and AdvFS



available volume managers

- Tru64 UNIX
  - **logical storage manager (LSM)**

    licensed port of Veritas VxVM

    based on Veritas VxVM 2.3 code base with code enhancements and extensions by Tru64 lab engineers

    minor feature differences from VxVM 3.2 from Veritas

    cluster capable (CLSM)

    Compaq added capabilities to support TruCluster

    full multi-instance single system view of disks and logical volumes

    file system or raw devices

- HP-UX
  - **logical volume manager (LVM)**

    internally HP lab developed and supported evolution of LVM from IBM/OSF

    cluster capable (SLVM)

    file system or raw failover of volume groups between cluster members

    raw access from multiple nodes
  - **Veritas volume manager (VxVM)**

    current version 3.2

    third party relationship

    cluster capable (CVM)

    file system or raw failover of volume groups between cluster members

    or raw access from multiple nodes

usage and level of deployment

Tru64 UNIX

- LSM usage in Tru64 server customer base is **a minority**
- why? a volume manager is viewed as optional for the following reasons;
  - disklabling [a-g] of physical volumes allows splitting large physical volumes into smaller units (up to 8)
  - multi-pathing is done transparently at the driver level (never implemented as a possible option in the volume manager)
  - the file system AdvFS provides a mechanism for on-line growth independent of a volume manager

HP-UX

- volume manager usage in HP-UX is **near universal**
- why? a volume manager is viewed as a necessity for the following reasons;
  - only mechanism to slice (subdivide) physical volumes into smaller units (no disklabeling paradigm in HP-UX as in Tru64 UNIX)
  - multi-pathing is not implemented in lower level drivers leaving two other means:
    - A feature of the volume manager (LVM pvlinks or VxVM dmp)
    - A special driver from the storage vendor such as StorageWorks SecurePath or SureStore Auto Path
  - the file system VxFS on-line growth model is based on volume manager's ability to grow underlying volume

LSM and VxVM

- features
- abstractions
- commands

VxVM and LSM features (1 of 2)

	HP-UX VxVM 3.2	Tru64 LSM (VxVM 2.3 Code Base)
spanning, concat, RAID 0, 0+1, 5	yes	yes
vmsa GUI	yes	lsmsa
dirty region logging	yes	yes
disassociate a plex for backup	yes	yes
snapshot volumes	yes	yes
persistent fast resync	yes	5.1B (SRL/FPA)
hot sparing/relocation	yes	yes
autoconfig	yes	yes
config load balancing	yes	yes
dynamic io sizes	yes	yes
configuration save and restore	yes	yes

VxVM and LSM features (2 of 2)

	HP-UX VxVM 3.2	Tru64 LSM (VxVM 2.3 Code Base)
resize coordinated with file system	yes	NA - AdvFS has own multi-volume model, 5.1B resize of an existing volume
dynamic multi-pathing	yes	NA – handled in base os drivers
layered volumes	yes	no
online relayout	yes	no
non-persistent fast resync	yes	no
SmartSync recovery accelerator for databases	yes	no
configurable power fail timeout	yes	no
task management (vtask)	yes	no

VxVM and LSM abstractions

- the same - derived from the same volume manager
  - diskgroup -> diskgroup
  - disk media -> disk media
  - subdisk -> subdisk
  - plex -> plex
  - volume -> volume

VxVM and LSM commands

- most are the same with substitution of “vol” for “vx”
  - `vxassist,vxctl,vxdg,vxdisk,vxdiskadd, vxdiskadm,vxdisksetup,vxedit,vxevac,vxinfo, vxinstall,vxliod,vxlmake,vxlmend,vxlmirror, vxlnotify,vxlplex,vxlprint,vxlreattach,vxlrecover,vxlroo tmir,vxlsd,vxlstat,vxltrace`
- a couple need a little more name munging
  - `vxconfigd->vold,vxvol->volume,vxsparecheck- >volwatch,dgcfgrestore->volrestore,dgcfgbackup->volsave`
- some are VxVM specific
  - `vxpfto,vxr5check,vxrelayout,vxrelocd,vxresize,vxtask,vxv mconvert,vx_emerg_start,vxbootsetup,vximportdg,vxvmboot`
- some are LSM specific
  - `volencap,volreconfig,vollogcnvt,volsetup,lsmsetup, volunroot,volmigrate,volunmigrate`

VxVM(LSM) and LVM

- features
- abstractions
- commands

LVM and VxVM features (1 of 2)

	VxVM	LVM
spanning, concat, RAID 0	yes	yes
import/export	yes	yes
config save and restore	yes	yes
integrated with OS installation	yes	yes
root boot support	3.5	yes
active/standby multi-pathing	DMP	pv links
RAID 1	yes	MirrorDisk/UX
logging mirrors	DRL	Mirror Write Cache
fast resync of mirrors	fast resync	split/fast remerge
hot sparing	yes	yes - vgextend –z y
on-line resize with file system	with VxFS	with VxFS
RAID 0+1	yes	yes, but fixed stripe width based on volume size
RAID 5	yes	
active/active multi-pathing	DMP	

LVM and VxVM features (2 of 2)

	VxVM	LVM
admin bells and whistles	many (vxma, vtask,.)	few
MC/SG integration	less - cannot be used for lock disk or root/boot, package startup delay (due to lengthy vxdg import), no shared read only activation. – CVM	more - SLVM
PRM/WRM capability		yes*

LSM (VxVM) and LVM
abstraction comparison

abstraction	VxVM/LSM	LVM
physical disk	disk media (dm)	physical volume (pv)
logical volume	volume(v)	logical volume (lv)
administrative unit of physical disks and logical volumes	disk group (dg)	volume group (vg)
group of physical volume with different HBAs	NA	physical volume group (pvg)
contiguous extent on a physical volume	subdisk (sd)	physical extent (pe
contiguous logical extent of storage for use in a volume	plex (p)	logical extent (le)

LSM (VxVM) and LVM
command equivalence
physical volumes/disk media

VxVM		LVM	
vxdisksetup	brings a disk under VxVM control 1 in the vxdiskadm menu adds or initializes one or more disks	pvcreate	makes a disk an LVM disk
vxdisk list	lists information about VxVM disks	pvdisplay	displays information about physical volumes in a volume group

LSM (VxVM) and LVM
command equivalence
volume groups/disk groups

VxVM		LVM	
vxdiskaddvxdg init	creates a new disk group and/or adds disks to a disk group	vgcreate	creates a volume group
vxdglist	displays the contents of a disk group	vgdisplay	displays information on all volume groups.
vxprint	displays information about all objects or a subset of objects.		
vxdiskadd	adds a disk to the disk group	vgextend	extends a volume group by adding one or more disks to it
vxdg deport	deports a disk group from the system.	vgexport	removes a volume group from the system
vxdg import	imports a disk group.	vgimport	adds a volume group to the system by scanning physical volumes which have been exported using vgexport

LSM (VxVM) and LVM
command equivalence
logical volume / volume

VxVM		LVM	
vxassist	creates volumes with the make parameter	lvcreate	create a logical volume
vxassist	increases a volume in size with the growto or growby parameter	lvextend	grow the size of a logical volume
vxassist snapshot	the snapshot operation takes one of the attached temporary mirrors and creates a new volume with the temporary mirror as its one plex	lvsplit	splits a mirrored logical volume into two logical volumes
vxrecover rvxvol start	the vxrecover command performs resynchronize operations for the volumes, or for volumes residing on the named disks (medianame or the VxVM name for the disk)	lvsync	synchronizes mirrors that are stale in one or more logical volumes

volume manager migration from Tru64

- today
  - Tru64 site may be unfamiliar with heavy use of a volume managers for disk partitioning, multi-pathing, or on-line file system growth
  - using VxVM will be an easier transition because LSM and VxVM are essentially the same volume manager:
    - VxVM features are a superset of LSM features
    - superficial command and feature name changes
  - carefully evaluate volume managers with MC/SG clustering
    - support only raw devices as active/active shared among members
    - different level of features
- future - HP-UX with TCR/AdvFS
  - most fully integrated volume manager for TCR/AdvFS on HP-UX is planned to be LVM

file system caches

Tru64 UNIX

unified buffer cache (UBC)

- dynamic with settable limits via tunable parameters
- based on CMU Mach work in a extensible memory management system
- similar in concept to Solaris (SysV R4?) dynamic cache
- cache of file pages
- common abstractions/structures (memory objects etc.) as memory of processes (stack, heaps, etc.)
- memory mapped files are unified (synchronized) between buffered io (read()/write()) and memory mapped (mmap()) access

HP-UX

dynamic buffer cache

- configurable as fixed or dynamic based on tunable parameters
- extension to classic UNIX “buf cache”
  - buf structures, etc
- cache of disk blocks, variable sized
- not integrated with process memory cache so memory mapped files are not unified (synchronized) between buffered io (read()/write()) and memory mapped mmap() access

future: unified file cache (UFC)

- will integrate buffered and memory mapped file access

tuning buffer caches

Tru64 5.1A

- `/etc/sysconfigtab`, `vm` subsystem
- `sysconfig`
- maximum and minimum percent of physical memory cache can consume

```
ubc_minpercent = 10
ubc_maxpercent = 100
```

- immediate “giveback” to free list if start paging threashold is hit

```
ubc_borrowpercent = 20
```

- percent of cached pages that can be dirty

```
vm_ubcdirtypercent = 40
```

- limit percent of cached pages that can belong to a single file

```
vm_ubcseqstartpercent = 50
vm_ubcseqpercent = 10
```

HP-UX 11i v1.6

- `/stand/system`
- `kmtune`
- to **enable static** buffer cache

  `nbuf != 0` or `bufpages != 0`
- If static, number of 4 KB pages in file system static buffer cache.

```
bufpages = 0
```

- If static, number of cache buffer headers. can be zero in which case number of headers will be bufpages/2

```
nbuf = 0
```

- to **enable dynamic** buffer cache

  `nbuf = 0` and `bufpages = 0`
- To control maximum and minimum percentage of physical memory cache can consume

```
dbc_min_pct = 5
dbc_max_pct = 50
```

monitoring buffer caches

Tru64 5.1A

collect and collgui report 8k ubc pages

```
 # collect -s m
Initializing (10.0 seconds) ... done.

#### RECORD    1 (1033136450:0) (Fri Sep 27 10:20:50 2002)
####

# MEMORY STATISTICS
#(<------------ MegaBytes -------------> <---------
Pages/sec -
#  Free    Swap     Act    InAc  Wire    UBC PI PO Zer Re COW
SW
      6       2      68      16    22     42  4  0   2  0   1

#### RECORD    2 (1033136460:0) (Fri Sep 27 10:21:00 2002)
####

# MEMORY STATISTICS
#(<------------ MegaBytes -------------> <---------
Pages/sec -
#  Free    Swap     Act    InAc  Wire    UBC PI PO Zer Re COW
SW HI
      6       2      68      16    22     42  0  0   0  0   0
0

#### RECORD    3 (1033136470:0) (Fri Sep 27 10:21:10 2002)
####

# MEMORY STATISTICS
#(<------------ MegaBytes -------------> <---------
Pages/sec -
```

HP-UX 11i v1.6

glance reports buf cache size

```
# glance -m
B3692A GlancePlus C.03.58.00    10:17:55 elizabet 9000/800    Curr
-------------------------------------------------------------------
CPU  Util                                                        |
Disk Util   F                                                    |
Mem  Util   S  SU UB B                                           | 2
Swap Util   UR                                                   |
-------------------------------------------------------------------

                                MEMORY REPORT
Event          Current    Cumulative    Current Rate    Cum Rate    Hig
-------------------------------------------------------------------
Page Faults          0           107             0.0         6.5       12
Page In              0            47             0.0         2.8        5
Page Out             0             0             0.0         0.0
KB Paged In        0kb           0kb             0.0         0.0
KB Paged Out       0kb           0kb             0.0         0.0
Reactivations        0             0             0.0         0.0
Deactivations        0             0             0.0         0.0
KB Deactivated     0kb           0kb             0.0         0.0
VM Reads             0             0             0.0         0.0
VM Writes            0             0             0.0         0.0

Total VM : 305.5mb   Sys Mem   : 171.8mb   User Mem: 100.7mb   Phys
Active VM: 255.9mb   Buf Cache: 127.8mb    Free Mem:  1.61gb
ProcList CPU Rpt  Mem Rpt  Disk Rpt       NextKeys SlctProc   Help
```

buffer cache Tru64 migration implications

- Tru64 applications:
  - today: apps that combine buffered io and memory mapped io to the same file (and assume a synchronized view of the file's data) will need to be modified
  - future: a future version of HP-UX will include unified file cache (ufc) eliminating this
- system management:
- no significant issues

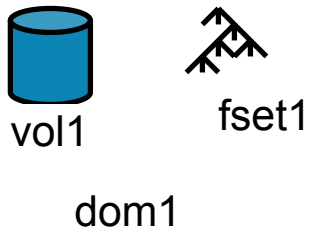
journal file systems

	Tru64 UNIX AdvFS	HP-UX JFS (VxFS)
storage model	multi-volume	single volume
journals	meta-data optional – user file data, Atomic Data Logging	meta-data
allocation abstraction	extents	extents
recovery	automatic on mount	external tool `fsck`, run in `bcheckrc`
on-line resize	`addvol, rmvol` or `mount -o (expand)`	(*volume mgr cmds*) + `fsadm, expandfs`
read-only file system copies	clones (`clonefset, mount`)	snapshots (`mount -F vxfs -o snapof=...`)
on-line defragmentation	`defragment,vfast`	`fsadm`

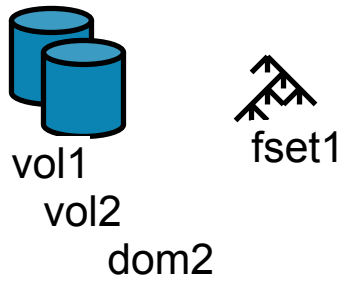
storage models: examples

AdvFS

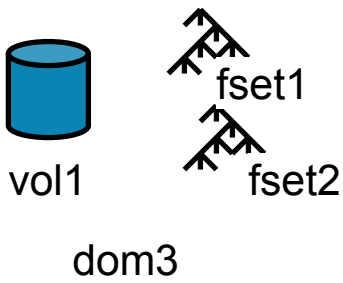
single volume, single fileset domain



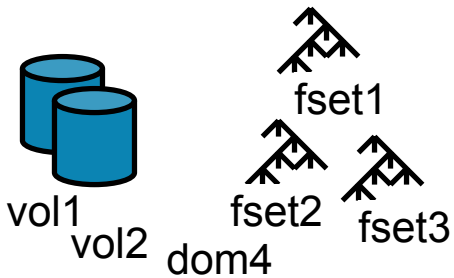
multi volume, single fileset domain



single volume, multi fileset domain



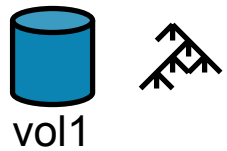
multi volume, multi fileset domain



cmds: `mkfdmn,mkfset,addvol,rmvol,balance`

VxFS

single volume file system



cmds: `mkfs`

storage model comparison

	AdvFS Tru64 UNIX	VxFS HP-UX
multi-volume model in file system	yes "domain" abstraction representing a pool of volumes to be used for a file system (`mkfdmn`, `addvol`, `rmvol`)	no for VxFS every file system is associated with a single volume (`mkfs`)
multiple mountable rooted trees per file system	yes AdvFS “fileset” abstraction representing an individual mountable tree within a domain (`mkfset`)	no every file system has one mountable tree with exception of the special case of a snap shot

creation and mounting commands

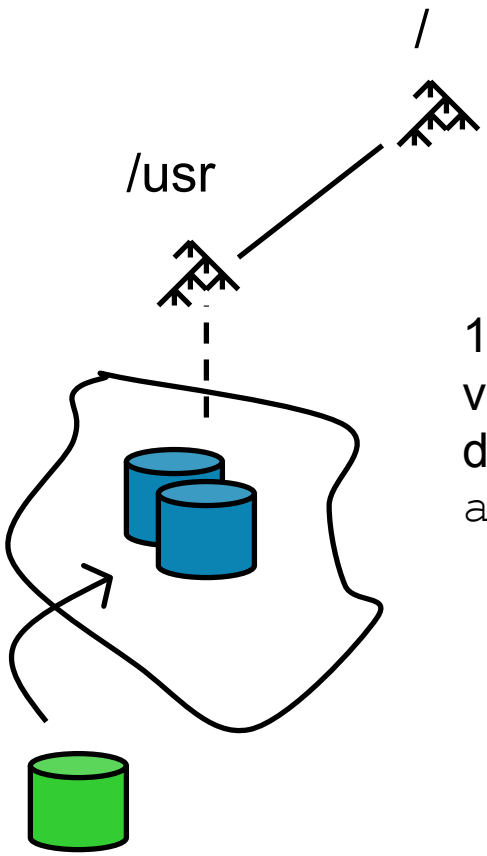
	AdvFS	VxFS

create a file system	`# mkfdmn vol dom`
	`# mkfset dom fset`
	`# mount dom#fset dir`
		`# mkfs vol`
		`# mount vol dir`

increase file system size	`# addvol vol dom`	*increase volume size* `# fsadm`
list the storage of a file system	`# showfdmn dom`	`# df`
list mounted file systems	`# mount`	`# mount`
determine unmounted file systems	`# more /etc/fstab` `# ls -R /etc/fdmns`	`# more /etc/fstab` `# fsck /…vol…`

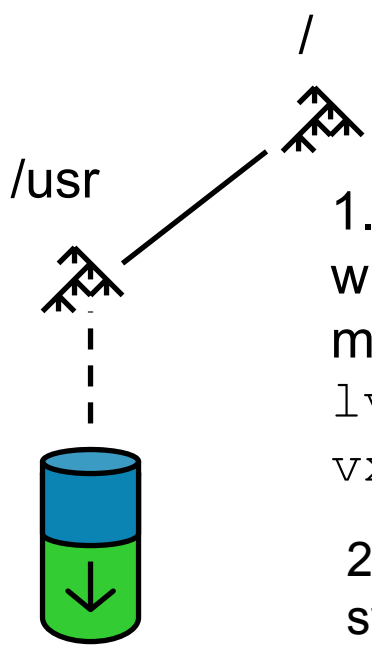
on-line resizing

AdvFS



1. add a volume to domain:
`addvol`

VxFS

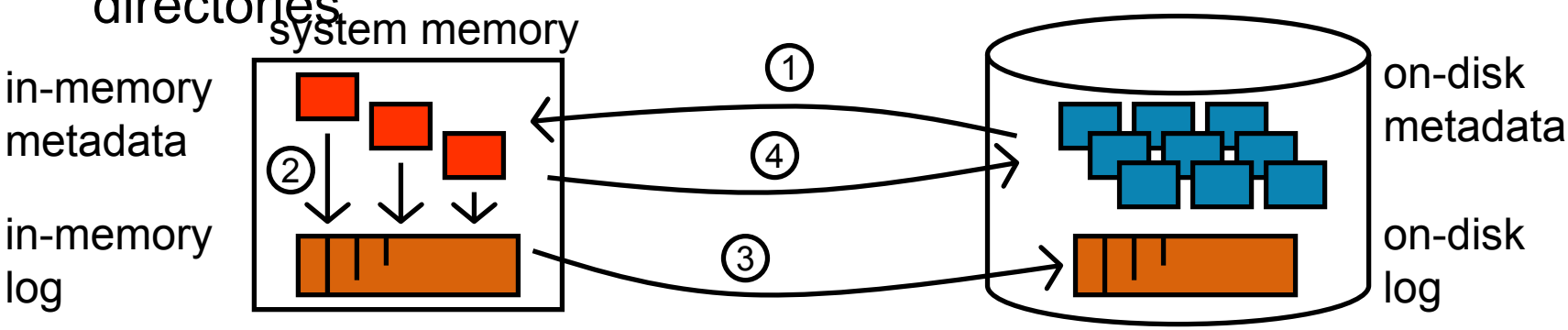


1. grow volume with volume manager:
`lvextend / vxassist`

2. notify file system of growth:
`fsadm/extendfs`

logging

- both file systems offer fast recovery based on transactional logging of metadata
- why?
  - faster recovery and mounting of file systems that did not unmount cleanly - system crash, storage path failure, etc.
- how?
  - file system maintains an in-memory and on-disk log of metadata changes made as a result of operations on files and directories



logging comparison (1 of 4)

	AdvFS Tru64 UNIX	VxFS HP-UX
metadata logging (journaling) for fast file system recovery	yes	yes

tunable log size	yes
	▪ creation time with
	`mkfdmn -l num_pages`
	▪ dynamically with
	`switchlog`
		yes
		▪ creation time with
		`mkfs logsize=n`

specify device to contain on-disk log	yes, `switchlog` utility or indirectly through `addvol`, `rmvol` commands	na, storage model is based on single volume file systems

logging comparison (2 of 4)

	AdvFS Tru64 UNIX	VxFS HP-UX
tunable log behavior	no	yes, `mount` options
▪ adjusting when a system call returns success from an operation in relation to when the associated log records recording the metadata changes are on disk		▪ "`log`" is default comparable to AdvFS ▪ "`delaylog`" potentially increases performance but lowers consistency guarantees to that of traditional non-logging UNIX file systems ▪ "`tmplog`" | "`nolog`" even lower consistency guarantees

logging comparison (3 of 4)

	AdvFS Tru64 UNIX	VxFS HP-UX
tunable log behavior	no	yes, `mount` option ▪ "`datainlog`" is the optimization and is the default for "HP OnLineJFS product" ▪ "`nodatainlog`" do not enable the optimization
▪ optimize synchronous write performance by combining time and data changes in logging		

logging comparison (4 of 4)

	AdvFS Tru64 UNIX	VxFS HP-UX

optional to log user file data changes for consistency	yes, atomic data logging (adl)
	▪ can be set for all files by admin with
	`mount -o adl`
	▪ can be set for individual files by admin with:
	`chfile -L`
	▪ programming api via `fcntl()`
		no

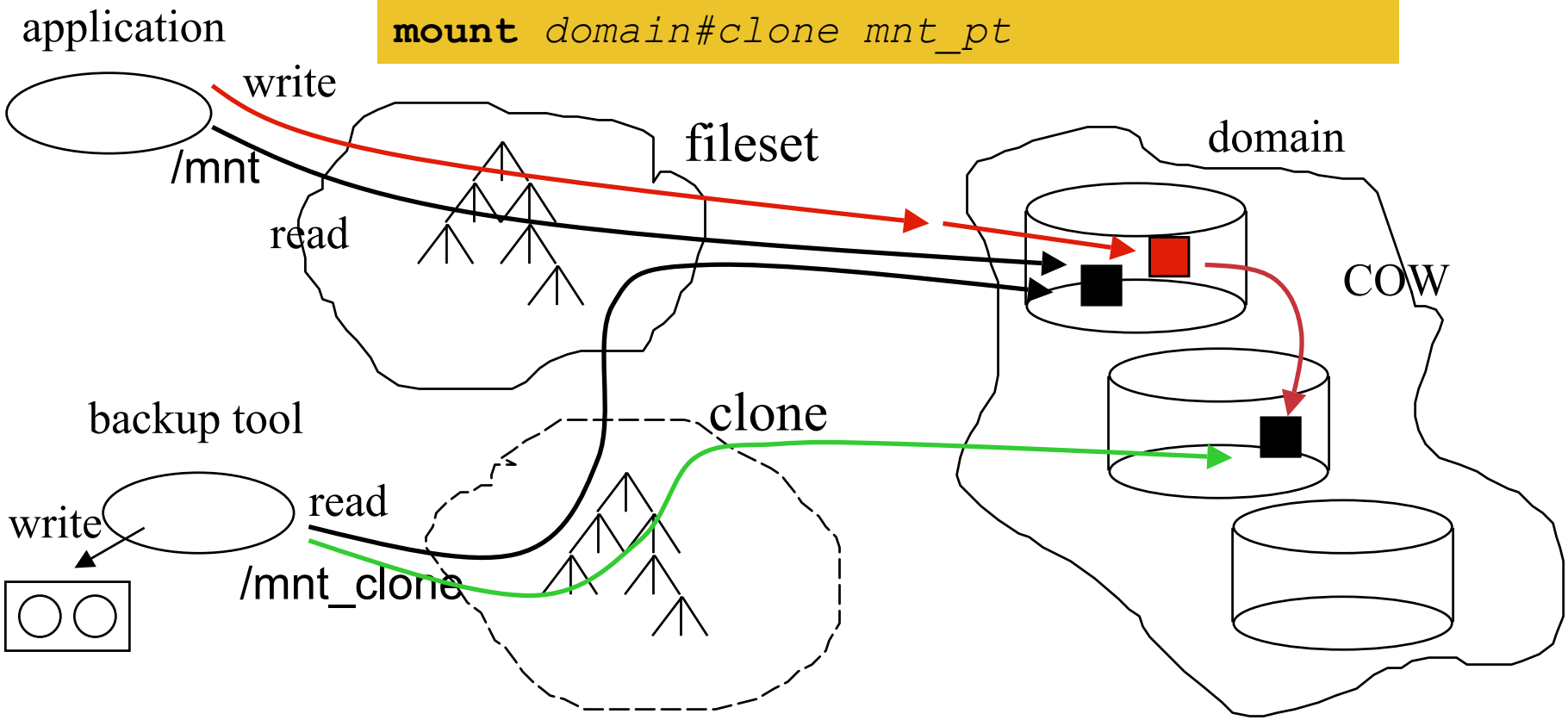
read-only point-in-time file system copies: AdvFS clones/VxFS snapshots

VxFS snapshot file system	AdvFS clone fileset
VxFS snapped file system	AdvFS *master* fileset

- common concepts
  - a clone/snapshot is not a separate copy of the data
  - writes to original file system trigger copy-on-write(cow) transfer of blocks to clone/snapshot
- AdvFS clones
  - clone exists even master is not mounted
  - master file set can not be deleted as long as clone exists
  - storage for clone comes out of domain of master
- VxFS snapshots
  - a snapped file system cannot be unmounted until any corresponding snapshots are first unmounted
  - a snapshot file system ceases to exist when unmounted
  - a snapshot file system has its own defined storage pool (volume) separate from that of the snapped file system
  - with VxVM also have (snapshot volumes of file systems) – persistent

AdvFS: Fileset Clones

```
clonefset   domain   fileset   clonename
mount domain#clone mnt_pt
```



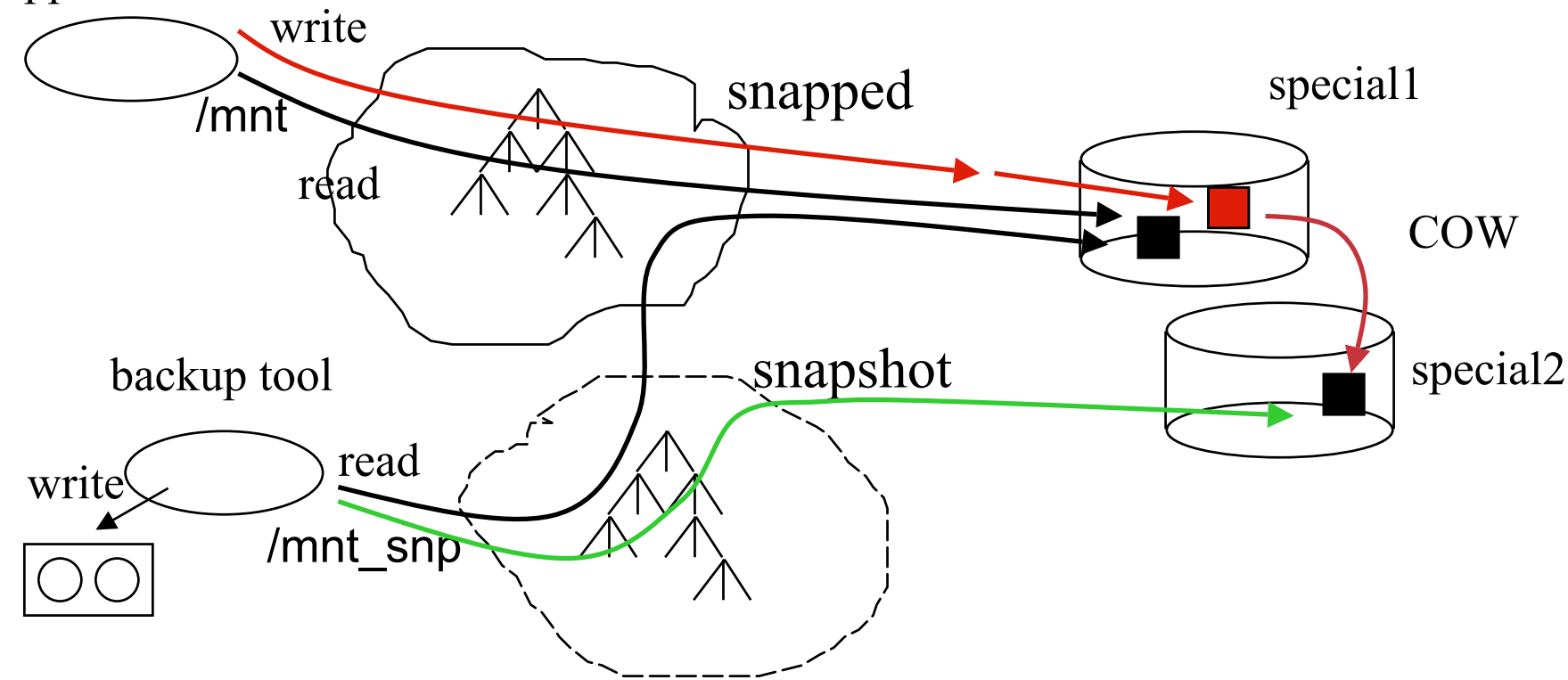
after clone is created, before any writes

first write to a block in the original (master) fileset

access to COW write blocks in the cloned fileset

VxFS: shapshots


```
mount -o snapof=special1,snapsize=size special2 mnt_pt
```



after snapshot is created, before any writes

first write to a block in the original snapped file system

access to COW write blocks in the snapshot fileset

large file support

- both provide support for files larger then 2^32
  - HP-UX/VxFS 2TB
  - Tru64 AdvFS 16TB
- AdvFS interface transparent
  - Tru64 has a single file API which supports a size_t (64 bit) as a file offset
  - no special flags, format or mount steps are required for large files
- VxFS interface explicit
  - to create, store and reference files larger the 2GB, all of these must be done
    - the file system must be formatted with "mkfs -o largefiles“
    - mounted with "mount -o largefiles“
    - opened by the application with "open(.. O_LARGE_FILE..)“

miscellaneous

- more tunables / knobs for VxFS
  - modify(overrides) for application I/O style
  - modify extend policy, allocation size,..
- both have modes to ensure no uninitialized blocks appear in a file if the system crashes
- both provide access control lists (acl)s
- both DMAPI API support
- both UNIX traditional user file system quotas
- both provide directIO/QuickIO IO modes
- both support freeze/thaw to support hw clones

command equivalence (1 of 3)

	AdvFS	VxFS
create a file system	`mkfdmn, mkfset`	`mkfs, newfs`
remove a file system	`rmfset, rmfdmn`	NA
make a file system available	`mount,umount`	`mount,umount`
defragment a file system	`defragment, vfast`	`fsadm`
check/repair a file system	`mount, fxfdmn, verify, salvage`	`fsck,fsdb,ff`
increase the storage of a file system	`addvol,rmvol or mount -u`	`fsadm`
administrate quotas on a file system	`edquota,ncheck,quot ,quota,quotacheck,r epquota`	`edquota,ncheck,qout ,quota,quotacheck`
report per user disk usage	NA	`vxdiskusg`

command equivalence (2 of 3)

	AdvFS	VxFS
backup a file system	vdump, vrestore	vxdump,vxrestore
monitoring performance	advfsstat	NA
tuning a file system	sysconfigtab, chvol,chfset,switchlog	fsadm,vxtunefs
create a read-only copy	clonefs,mount, umount,rmfset	mount -o snapof,fscat,umount
suspend file system activity for HW snapshot or clone	thawfs, freezefs	API
manage multi-volume file system	addvol,rmvol,showfdmn,showfsets,balance,migrate	NA

command equivalence (3 of 3)

	AdvFS	VxFS
manage ACLs	`setacl,getacl`	`setacl,getacl`
show a file's extents	`showfile`	`getext`
set a file's extent policy	NA	`setext`
tuning configuration file	`/etc/sysconfigtab`	`/etc/vx/tunefstab`
make manage physical copies of file systems	`dd,disklabel`	`volcopy,labelit`
in-place conversion from Berkley UNIX file system	NA	`vxfsconvert`
in-place conversion to latest on-disk format	NA	`vxupgrade`

file system migration issues from Tru64

- data: endian issue
  - automatic file data migration tool or support from HP currently not planned and extremely unlikely due to nature of problem

    only application knows what is the granularity of swapping bits/bytes.. byte/word/long….
  - will not be able read data from Tru64 AdvFS file system disks on HP-UX PA-RISC or IPF systems
  - some discussion of making the file system mountable… but you couldn't read the data in your files
- administration/solution architecture
  - today: VxFS offers comparable features to AdvFS

    different command interface
  - future (11.31): AdvFS will be incorporated in HP-UX offering all significant features currently in Tru64/AdvFS

    some command interface changes may exist in HP-UX to offer familiarity to HP-UX classic VxFS users.

    "clone" concept probably will be renamed "snapshot"

    mkfs probably extended to offer mkfdmn/mkfset functionality

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