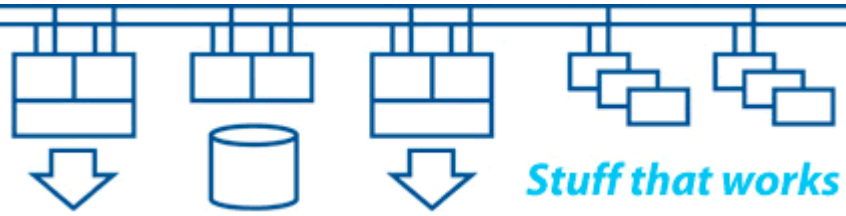
OpenVMS advanced technical symposium – May 2008

Recent experiences designing and implementing disaster-tolerant OpenVMS Integrity clusters

Colin Butcher, XDelta Limited



Context

- **Hardware upgrade from Alpha to Integrity**
- **Storage upgrade from HSG80 to EVA4100**
- **OpenVMS upgrade from V8.2 to V8.3-1H1**
- **Merge separate databases to single database**
- **Database major version upgrade**
- **Application updates**
- **Increased availability and performance demands**
- **Separate test & training environment**
- **Common configuration and setup across all clusters**

The biggest obstacle was determining if it was possible to migrate the data within an acceptable time window – so we built a proof of concept system to test it first

Designing for disaster tolerance

An overview of the issues to be considered when designing, implementing and running a disaster-tolerant, mission-critical split-site cluster.

Mission Critical systems

Mission critical systems need to be able to:

- **Survive failures (resilience and failover)**
- **Survive changes (adapt and evolve)**
- **Survive people (simplify and automate)**
- **Never corrupt or lose critical data (data integrity)**
- **Requirements never remain static over an extended period of time, so we need to be able to make changes during the operational lifetime of the system**
- **Circumstances change, so we often need to be able to extend the operational lifetime and scope of a system**

Do we really need 100% uptime?

- **Safety-critical systems (especially safety-critical real-time monitoring and control systems such as air traffic control) require exceedingly high levels of availability. They also have to be fail-safe in order not to endanger lives.**
- **True 24x365 mission-critical systems are fairly rare. With these there is no "downtime window" to take backups, fix faults or to make changes. So, whatever you do has to be done "live" – and very carefully!**
- **The closer you get to 100% uptime the more expensive a satisfactory solution will become.**

The systems “flight envelope”

- What does the business require the systems to do?
- What are the consequences if the systems fail?
- What happens if you push beyond the limits?
- How far from the edge are you?
- How do you know?
- What can we measure?
- What comparisons can we make?
- What evidence can we look at?

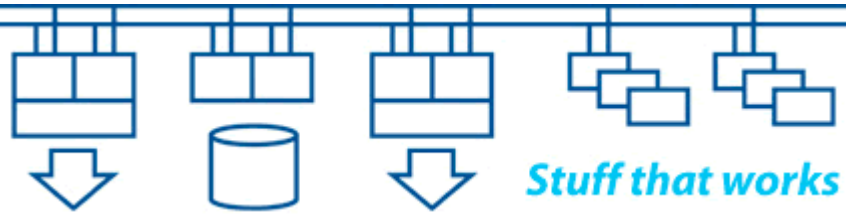
RPO and RTO

RPO = Recovery Point Objective

- **How much data can we tolerate losing?**
- **How quickly do we need to react to a failure?**

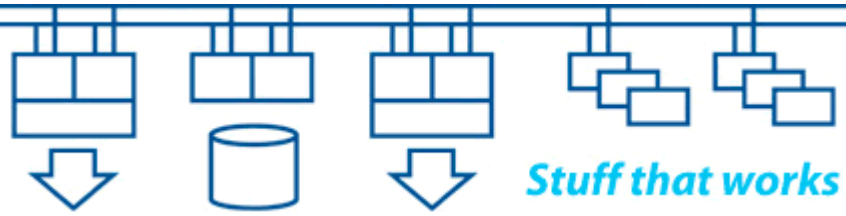
RTO = Recovery Time Objective

- **What level of service outage can we tolerate?**
- **How quickly do we need to recover?**
- **How quickly do we need to be ready to deal with a subsequent failure?**



"Survivability test"

Cause of Outage:	Planned (Maintenance)	Unplanned (Failure)
Hardware	?	?
Operating System	?	?
Network Layer	?	?
Layered Products	?	?
Application Software	?	?
Application Data	?	?
Environment	?	?
People	?	?



Availability and performance

Availability:

- Business Continuity = ability to continue business operations
- Disaster Tolerance = ability to survive major failures (eg: site)
- High Availability = ability to survive equipment failures

Performance:

- Performance issues are often the cause of transient system failures and disruption
- The systems have to have sufficient capacity and performance to deal with the workload in an acceptable period of time

Availability is more important than performance

Throughput and response times

- **Bandwidth – determines throughput**
  - **It's not just "speed", it's throughput in terms of "units of stuff per second"**
- **Latency – determines response time**
  - **Determines how much "stuff" is in transit through the system at any given instant**
  - **"Stuff in transit" is the data at risk if there is a failure**
- **Jitter ("div latency" or variation of latency with time) – determines predictability of response**
  - **Understanding jitter is important for establishing timeout values**
  - **Latency fluctuations can cause system failures under peak load**

Designing for performance

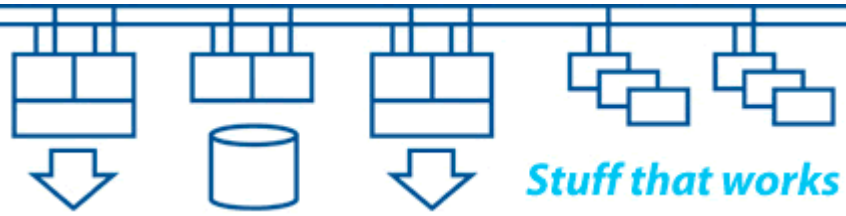
- **Size systems to cope with peaks in workload**
- **Understand how the applications could break down into parallel streams of execution**
- **Understand scalability – do as much as possible once only, do little as possible every time**
  - **The fastest IO is the IO you don't do**
  - **The fastest code is the code you don't execute**
- **Understand the need for synchronisation and serialisation of access to data structures**
- **Minimise "wait states" and contention**

Designing for availability

- **Which parts of the system are mission-critical?**
- **What kind of failure do we prefer?**
- **What happens to our data when things go wrong?**
- **What state transitions occur during failure and recovery?**
- **How can we recover from a failure without data loss or data corruption?**
- **Should we automate decision making?**
- **How can we get good information?**

Split-site issues

- Effects of distance on network and storage protocols
- Symmetric or asymmetric operation?
- Avoid booting across inter-site links
- Remote access for management and operation
- Centralised (and duplicated) monitoring and alerting
- Naming conventions
- Quorum and voting scheme
- Host-based volume shadowing scheme
- Full environmental monitoring for lights-out sites
- Avoid automation of decision making when a site fails



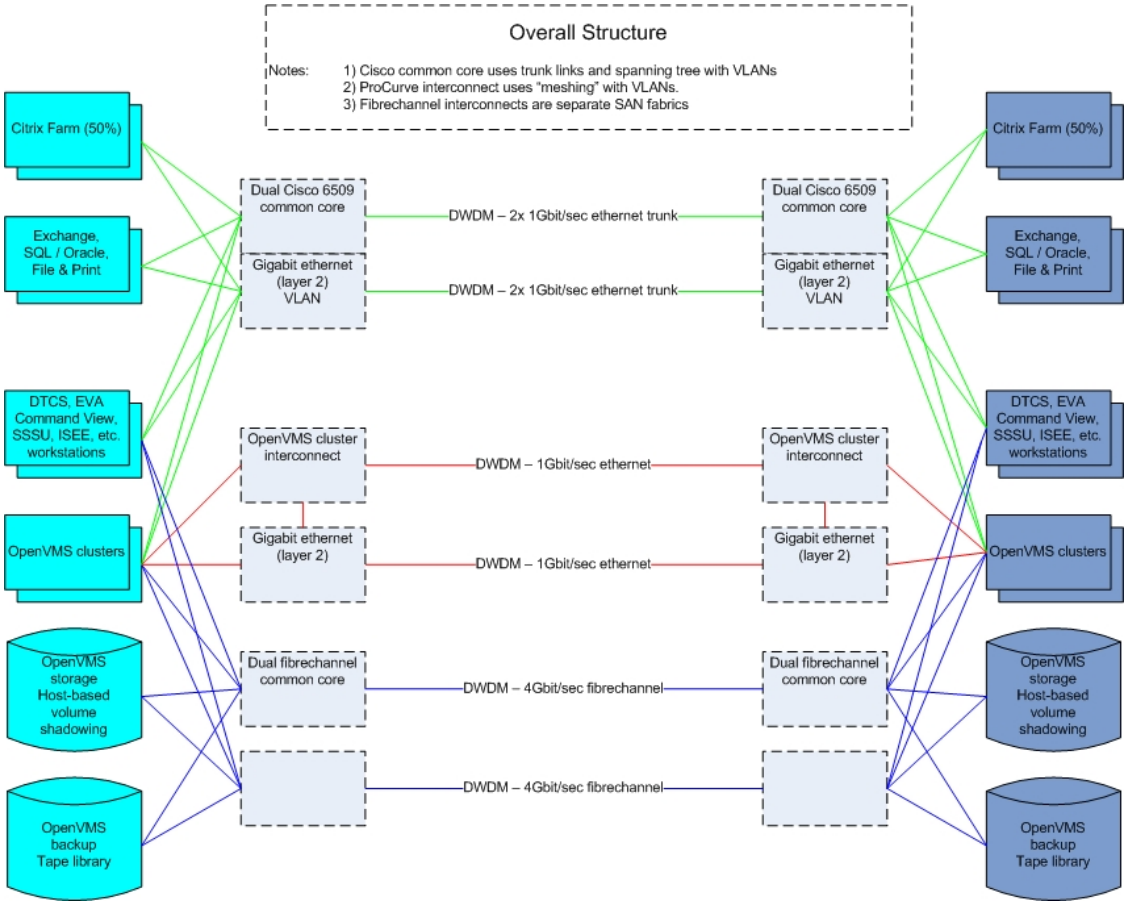
Testing - proving that it all works

- We need to test for scale as well as functionality
- We need to test every aspect of the system and surrounding infrastructure to satisfy ourselves that it behaves as we expect under normal, failure and recovery conditions
- We need to understand the underlying cause of problems so that we can fix them or avoid them
- We need to prove that service will continue with minimal disruption while both failure and recovery are in progress
- We need to know how to recover from failures without loss of service and without data corruption or data loss
- We need to regularly rehearse and test our procedures and plans to ensure that we stay current

2: The new systems

- **Integrity Servers (rx6600s and rx2660s)**
- **OpenVMS V8.3-1H1**
- **EVA 4100 storage arrays with 15k rpm 146GB drives**
- **MSL4048 tape libraries with Ultrium LTO4 FC drives**
- **SANswitch 4/32B fibrechannel switches (private SAN interconnects with dual 4GigFC inter-site links)**
- **Procurve 3500yl-24 network switches (private network interconnects with dual GigE inter-site links)**
- **Proliant DL380 G5 servers (DTCS monitoring, EVA command view, WEBES / ISEE reporting etc.)**
- **DTCS monitoring and alerting**

Where these systems fit in the new infrastructure



Functional layout

The systems are split up into:

- **Common infrastructure (SAN fabrics, private network interconnects etc.)**
- **Production environment (a split-site cluster with host-based volume shadowed storage)**
- **Test environment (a split-site cluster on a smaller scale)**
- **Archive environment (a single node at Site A)**
- **Duplicated monitoring and reporting facilities**
- **External connectivity for users**

Split-site layout

The systems are split across two sites:

Site A:

- Half of Production cluster + 2x Production EVAs
- Half of Test cluster (including rx6600) + 1x Test EVA
- Archive server (shares one of the Production EVAs)
- DTCS workstations and MSL tape library

Site B:

- Half of Production cluster + 1x Production EVAs
- Half of Test cluster (including rx6600) + 1x Test EVA
- EVA storage for Archive server (shares Production EVA)
- DTCS workstations and MSL tape library

Integrity Servers (rx6600 and rx2660)

- Dual-core Itanium 2 CPUs
  - 3 socket / 6 core 1.6GHz 12MB cache in rx6600 (4 socket max.)
  - 1 socket / 2 core 1.6GHz 9MB cache in rx2660 (2 socket max.)
- 64GB in rx6600 (192GB max.)
- 16GB in rx2660 (32GB max.)
- 8 port built-in SAS array controller (2x IM arrays with hot spare)
- 4Gbps fibrechannel
  - 2x dual port HBAs in rx6600
  - 1x dual port HBA in rx2660
- 1Gbps ethernet
  - 4x fibre (user network), 4x copper (private interconnect) in rx6600
  - 2x fibre (user network), 4x copper (private interconnect) in rx2660
- iLO and serial console

EVA 4100 storage arrays

- Single “disk group” using “double sparing”
- All spindles are 146GB 15k rpm
- All presented “Vdisks” are RAID 0 + 1
- SAN zoning ensures that
  - all EVAs are available to all Integrity Server systems
  - only Production EVAs are available to Production DL380s
  - only Test EVAs are available to Test DL380s
- Vdisk presentations control which Integrity Server systems can see which Vdisk devices within each EVA
- Mirrorclones, snapclones and snapshots within the EVA are used to take copies of data for backups and other purposes

MSL4048 tape libraries

- Two Ultrium LTO4 1840 FC tape drives per library
- 48 slot magazines and a robot to load the drives
- Web interface for direct management and setup
- Use 6 character tape labels for OpenVMS (ANSI standard)
- Auto-cleaning (label the cleaning cartridges!)
- Uses MDMS software to perform tape library management
  - 2x MDMS server nodes, 1 per site
  - MDMS GUI on Proliant DL380s
  - MDMS clients use DECnet to communicate with MDMS servers
- SAN zoning ensures that all MSL4048s are available to all Integrity Server and Proliant systems

Proliant DL380-G5 servers

- 8 port battery-backed SAS array controller
- 1x dual port 4Gbps fibrechannel HBA (to manage EVAs)
- 1Gbps ethernet using NIC teaming in failover mode (equivalent to OpenVMS LAN failover)
  - 2x fibre (Cisco user network) for external connectivity
  - 2x copper (HP ProCurve interconnect)
- Windows Server 2003 32bit with Proliant support pack
- DTCS management station
- ISEE, WEBES, mail forwarding
- EVA Command View, SSSU server
- FTP / TFTP server for firmware updates etc.
- iLO2 console

SANswitch 4/32B “powerpack” FC switches

- **Dual-fabric SAN – two entirely separate extended fabrics**
- **Each switch is 32ports 4Gbps fibrechannel**
- **Ports grouped up for EVAs, tapes, systems, management stations and inter-site links**
- **4Gbps inter-site link for each fabric using a trunked pair of 2Gbps DWDM links**
- **SAN Zoning follows the “single initiator, multiple targets” model and controls which devices (EVAs, tapes) are visible to which systems (rx6600s, rx2660s, DL380s)**

How are the SAN switches and EVAs configured?

- **Dual-fabric SAN with no single point of failure**
- **All dual-port HBAs in all systems are dual path to two separate SANswitches**
- **SAN zoning and EVA presentations control access to devices**
  - **Only Production EVAs can be managed by Production EVA Command View workstations**
  - **Only Test EVAs can be managed by Test EVA Command View workstations**
  - **All initial EVA configurations were created using EVA scripts from the OpenVMS systems**

HP ProCurve 3500yl-24 network switches

- **Ports grouped up for iLO / console devices, systems, cluster interconnects and inter-site links**
- **ProCurve meshing uses “shortest path” mechanism**
  - **Ports 23 & 24 use fibre for mesh links between switches**
  - **Ports 24 are the inter-site links using DWDM 1GigE links**
- **VLANs separate out the traffic types:**
  - **VLAN for TCP/IP to iLOs and SSU scripting, AMDS for DTCS monitoring / quorum adjustment, DECnet for MDMS, LAT**
  - **VLAN for SCS path A**
  - **VLAN for SCS path B**

What protocols and VLANs do we use?

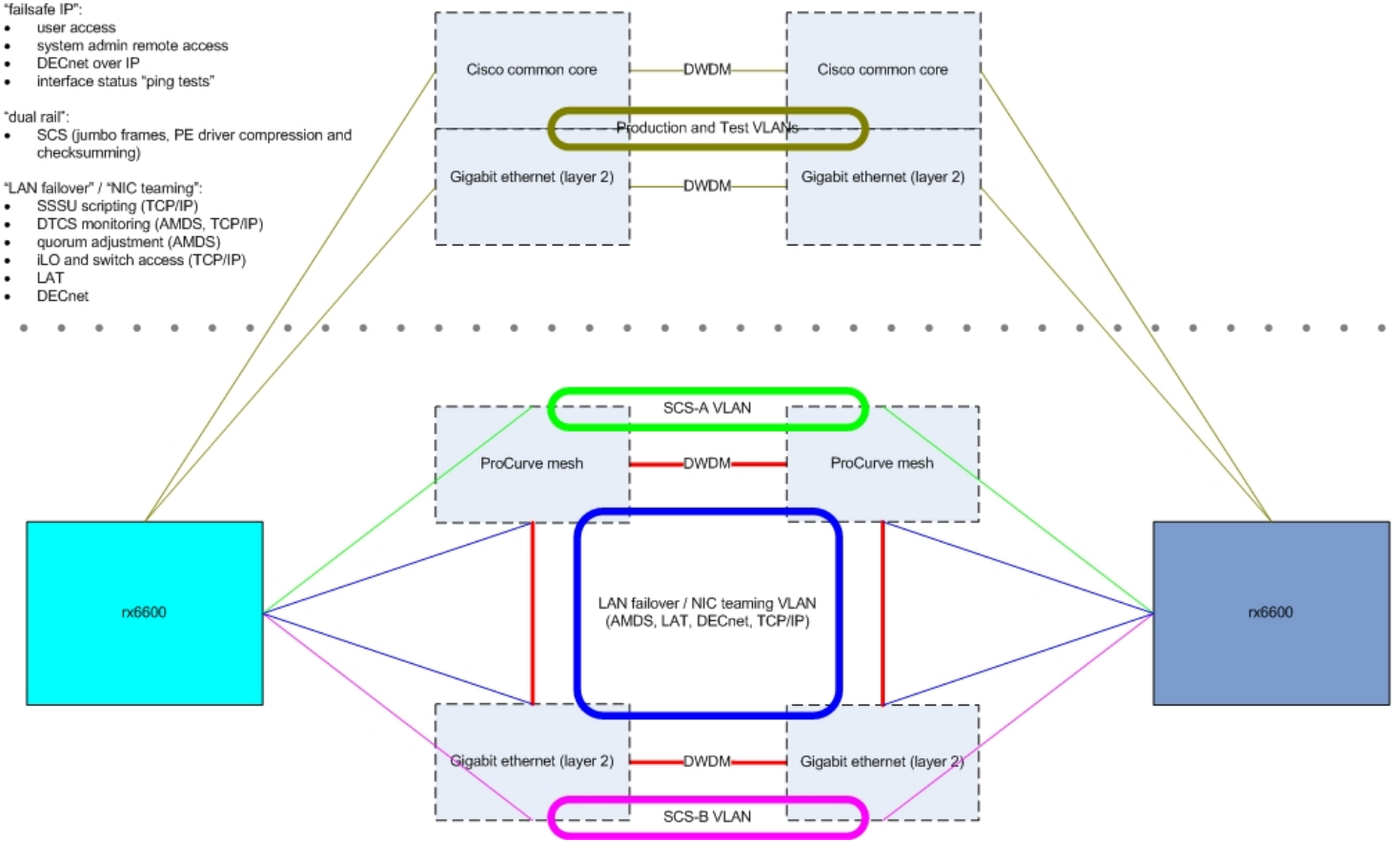
- **Dual-rail VLANs over ProCurve mesh**
  - **SCS (locking, HBVS bitmap copying, etc.)**
- **LAN failover VLAN over ProCurve mesh**
  - **AMDS (DTCS monitoring and Quorum adjustment)**
  - **TCP/IP (SSSU access to EVA CV workstations, iLO access to all systems, iLO monitoring by DTCS, access to MSL4048s, access to SAN and network switches, TFTP / FTP firmware updates, etc.)**
  - **DECnet-Plus (MDMS, data copying etc.)**
  - **LAT (last-ditch terminal access)**
- **All NICs in all systems (except devices with a single port) are dual path to two separate ProCurve switches**

Common network infrastructure

"failsafe IP":
- user access
- system admin remote access
- DECnet over IP
- interface status "ping tests"

"dual rail":
- SCS (jumbo frames, PE driver compression and checksumming)

"LAN failover" / "NIC teaming":
- SSSU scripting (TCP/IP)
- DTCS monitoring (AMDS, TCP/IP)
- quorum adjustment (AMDS)
- iLO and switch access (TCP/IP)
- LAT
- DECnet

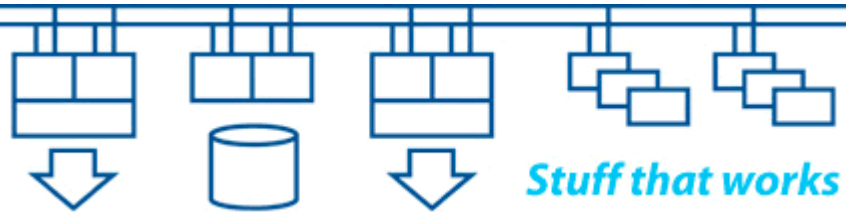


What external IP addressing do we use?

- All NICs in all systems are dual path to two separate Cisco switches
- All Cisco connections use fibre, not copper
- Uses “failsafe IP” for maximum bandwidth and flexibility
- Failsafe IP addresses move from NIC to NIC on the same machine or even within a cluster if a NIC or switch fails
- We use three kinds of IP address on the Cisco interfaces:
  - “Hidden” dedicated IP addresses for each NIC used for local reachability testing
  - Per-machine failsafe IP addresses used for systems management access
  - Application service failsafe IP addresses used for access to the applications and Databases. Disabled when not available. Only made available when the systems are ready for use.

How the OpenVMS clusters are configured

An overview of how the OpenVMS systems are configured and booted.



Split-site clustering

- **Split-site OpenVMS clusters give us “shared everything” access to data with protection from loss or corruption, even in the event of site failure**
- **Host-based volume shadowing (HBVS) ensures that data is consistent across all members of the shadow sets. It does not ensure that data is correct – that’s up to you!**
- **The quorum scheme lets Site A continue if Site B fails and protects us from data corruption due to a partitioned cluster**
- **The DTCS software monitors the systems for us and (most important of all) controls the formation of shadow sets when the systems boot and when systems rejoin the cluster**

Production cluster

- 1x rx6600 + 1x rx2660 at each site
- 2x EVA at Site A
- 1x EVA at Site B
- 1x DL380 at each site
- Shared access to tape libraries via MDMS
- HBVS – 2 copies at Site A, 1 copy at Site B
- Quorum – biased to Site A (expected votes = 5)
- Nodes PRDA01, PRDA02 boot from Site A EVAs
- Nodes PRDB03, PRDB04 boot from Site B EVA
- Node PRDA01 is the “primary” node for Database and application software (votes = 2, rest have votes = 1)

Test cluster

- 1x rx6600 + 1x rx2660 at Site A
- 2x rx2660 at Site B
- 1x EVA at each site
- 1x DL380 at each site
- Shared access to tape libraries via MDMS
- HBVS – 1 copy at each site
- Quorum – biased to Site A (expected votes = 5)
- Nodes TSTA01, TSTA02 boot from Site A EVAs
- Nodes TSTB03, TSTB04 boot from Site B EVA
- Node TSTA01 is the “primary” node for performance testing (votes = 2, rest have votes = 1)

Archive “cluster” (single node)

- 1x rx2660 at Site A
- 1x EVA at each site (shares Production EVAs)
- Monitored by the Production DL380s
- Shared access to tape libraries via MDMS
- HBVS – 1 copy at each site
- Node ARCA01 boots from Site A EVA

Host-based volume shadowing (HBVS)

- The Production cluster uses 3 member shadow sets across 3x EVAs (2 EVAs at Site A, 1 EVA at Site B)
- The bootable system disk shadow sets at a site are only mounted by the nodes physically located at that site
- Local storage for page / swap / dump files
- The cluster-common disk is mounted by all nodes in the cluster and holds those files that must be unique and consistent across the entire cluster
- We make use of mini-copy and mini-merge by setting HBVS policies. These significantly speed up the catch-up process under most circumstances by maintaining write bitmaps on all members of the cluster that mount the same shadow set
- Lots of small shadow sets to give good granularity and control over HBVS behaviour

Booting OpenVMS (1)

- **The Integrity Server systems don't boot on power-up (like the Alphas) – or on reboot (unlike the Alphas). EFI behaves differently and looks extremely different from an Alpha console**
- **The MP / EFI boot menus need to be configured to disable auto-boot and to disable auto power-up following power loss**
- **The nodes boot from EVA disk devices. This is configured using the BOOT_OPTIONS.COM mechanism by setting the boot device and the correct system root [SYSn.]**
  Hint: adding a node to a running cluster means booting from another disk then mounting the target disc read-only
- **One slot in the SAS enclosure contains a bootable copy of the OpenVMS V8.3-1H1 installation media**

Booting OpenVMS (2)

- OpenVMS reads the load image from the specified device and proceeds to boot from the specified system root
- The cluster is formed
- LAN failover virtual LAN interfaces are created
- DTC_MOUNT_DISKS runs to mount the cluster-common disk – it prompts if needed
- The networking layers are started (DECnet, then TCP/IP)
- DTC_MOUNT_DISKS runs again to mount all the shadow sets – it prompts if needed
- Layered products are started
- The Database and applications are started

DTCS – some of its capabilities

- DTCS is a set of HP and 3rd party products with installation, configuration and support services
- Remote console access, management and console output logging
- Rule based monitoring of individual systems / nodes (eg: required OpenVMS processes, cluster members etc.)
- DTC_MOUNT_DISKS.COM to control shadow set mounts on boot
- Integrated AMDS monitoring and quorum adjustment
- Rule based SNMP polling of equipment for expected device state, port state etc.
- Rule based TCP/IP “ping reachability” polling of addresses
- GUI and e-mail based alerting
- Scripting of failover and recovery actions across all nodes being monitored and controlled

Thank you for your participation

Q & A