OpenVMS HA and DT clusters

Designing and implementing high availability and disaster tolerant OpenVMS clusters

Colin Butcher
XDelta Limited

www.xdelta.co.uk
+44 117 904 8209

XDelta – who we are

- Independent consultants since 1996:
  - UK based with international reach
  - Over 30 years experience with OpenVMS
- We design and implement solutions:
  - Mission critical systems
- Cross-sector experience
- Engineering background
- Gartner (2009):
  - Identified XDelta as one of few companies world-wide capable of OpenVMS migration related projects

Business Partner

ITANIUM SOLUTIONS ALLIANCE
HUMANITARIAN IMPACT INNOVATION AWARD
HONORABLE MENTION

FINALIST
BCS IT INDUSTRY AWARDS
ComputerWeekly

OpenVMS HA and DT clusters

- Design principles
- Network and storage connectivity
- Storage layout, shadowing, booting
- Log file management
- Backup / restore
- Performance
- Monitoring and management

Design goals

- Design for change, not steady-state
- Operational safety – minimise risk of errors and disruption
- Understand the purpose and the target environment
- Build in logging and information gathering
- Adapt to changing requirements (performance, scalability)
- Think long-term (e.g.: company mergers)

Survivability matrix

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

Naming conventions

- Choose your naming conventions very carefully – they are the hardest thing to change later
- Don’t tie nodenames to physical locations
- Choose disc device IDs that identify meaningful things (e.g.: environment, site, array and purpose)
- Choose network addresses and hostnames that identify meaningful things and make sense in your context

Abstraction layers

“All problems in computing can be solved by introducing another layer of abstraction.”

“Most problems in computing are caused by too many layers of complexity.”

We need to strike a balance that is appropriate for the kinds of systems we’re building.

Example node naming convention

<n1><nn2>DC<n3>, where:

<n1> = “P” (Production), or
“T” (Test), or
“D” (Development)

<nn2> = 01 … 99 (node number within site)

DC = “data centre” (site)

<n3> = 1 … 9 (site number)

Example naming convention – FC discs

- $1$DGA<n1><n2><nn3>

  <n1> = site / environment (1 … 3 = Production A, B, C;
  4 … 6 = Test A, B, C;
  7 … 9 = Development A, B, C)

  <n2> = array within site (1 … 9)

  <nn3> = disc, matches DSA<nn> shadow set name

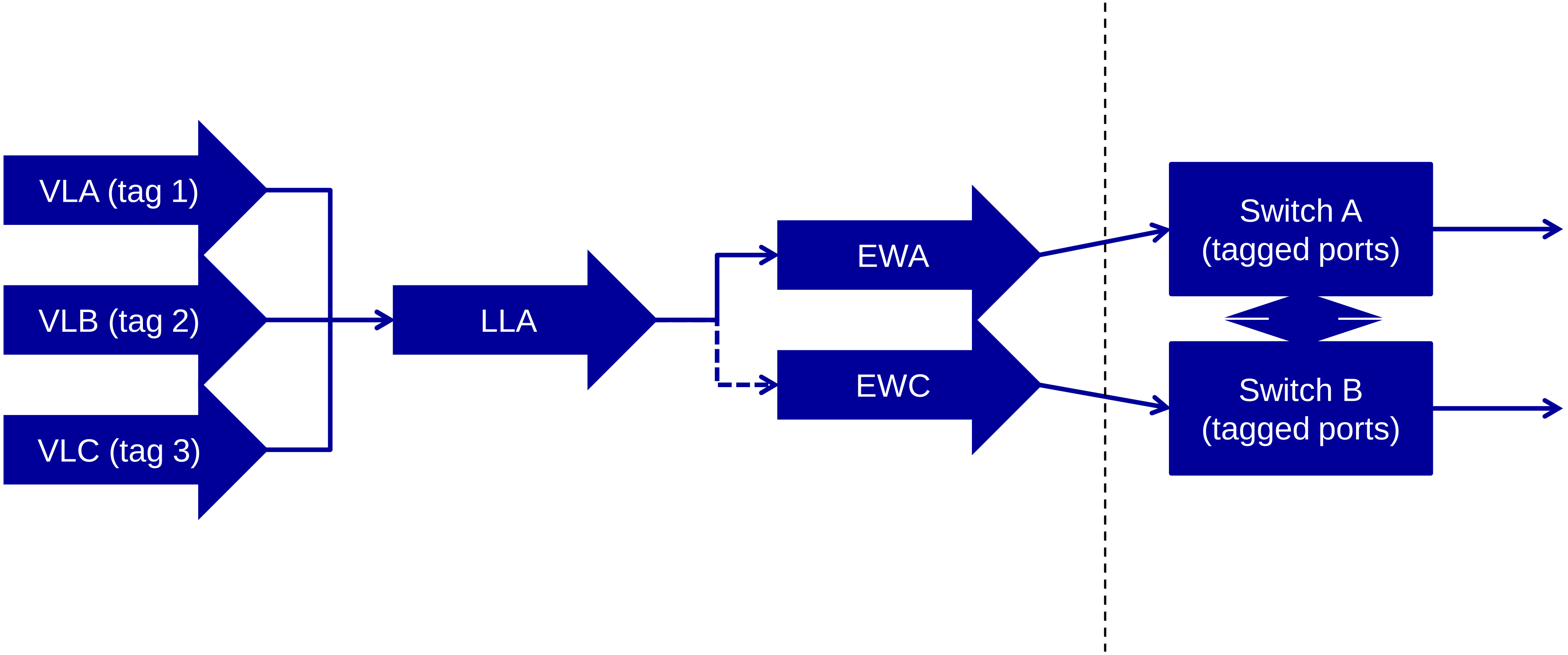
- $1$DGA<nnnnn> = common (RO) discs

Note: UUID must be unique across the fabric, ALLOCLASS = 1

Network connectivity

- Multiple protocols: SCS, TCPIP, DECnet, AMDS
- Use LAN failover with multiple NICs for hardware resilience
- Use VLAN tagging and/or LAN failover sets to separate traffic flows
- VL / LL devices map to physical NICs, do not configure protocols on physical NICs.
- Use “service addresses” to separate data flows
- Use QoS in data network for different data flow types
- Use SCACP to control which port(s) SCS runs on
- Use LATCP to control which port(s) LAT runs on
- Disable unused protocols (eg: DECdns, DTSS)

OpenVMS networking: connectivity



Inter-site data network links

- Extended layer 2 or routed layer 3 ?
- SCS at layer 2 or “clusters over IP” ?
- Preference is to use extended layer 2 with QoS on specific VLANs to control latency and bandwidth
- LAT is a useful protocol to test connectivity paths at layer 2
- AMDS (Availability Manager) is a layer 2 protocol
- Avoid MSCP serving, especially with shadow sets

Extended layer 2 LANs

- DWDM over dark fibre
- MPLS

- Traffic separation with VLAN 802.1Q tags

- Use QoS to control traffic flows

- Switches have manufacturer specific features:
  - HP Procurve has “meshing”
  - Cisco has “etherchannel”
  - Extreme has “EAPS ring”

Example data network connectivity

"failsafe IP":

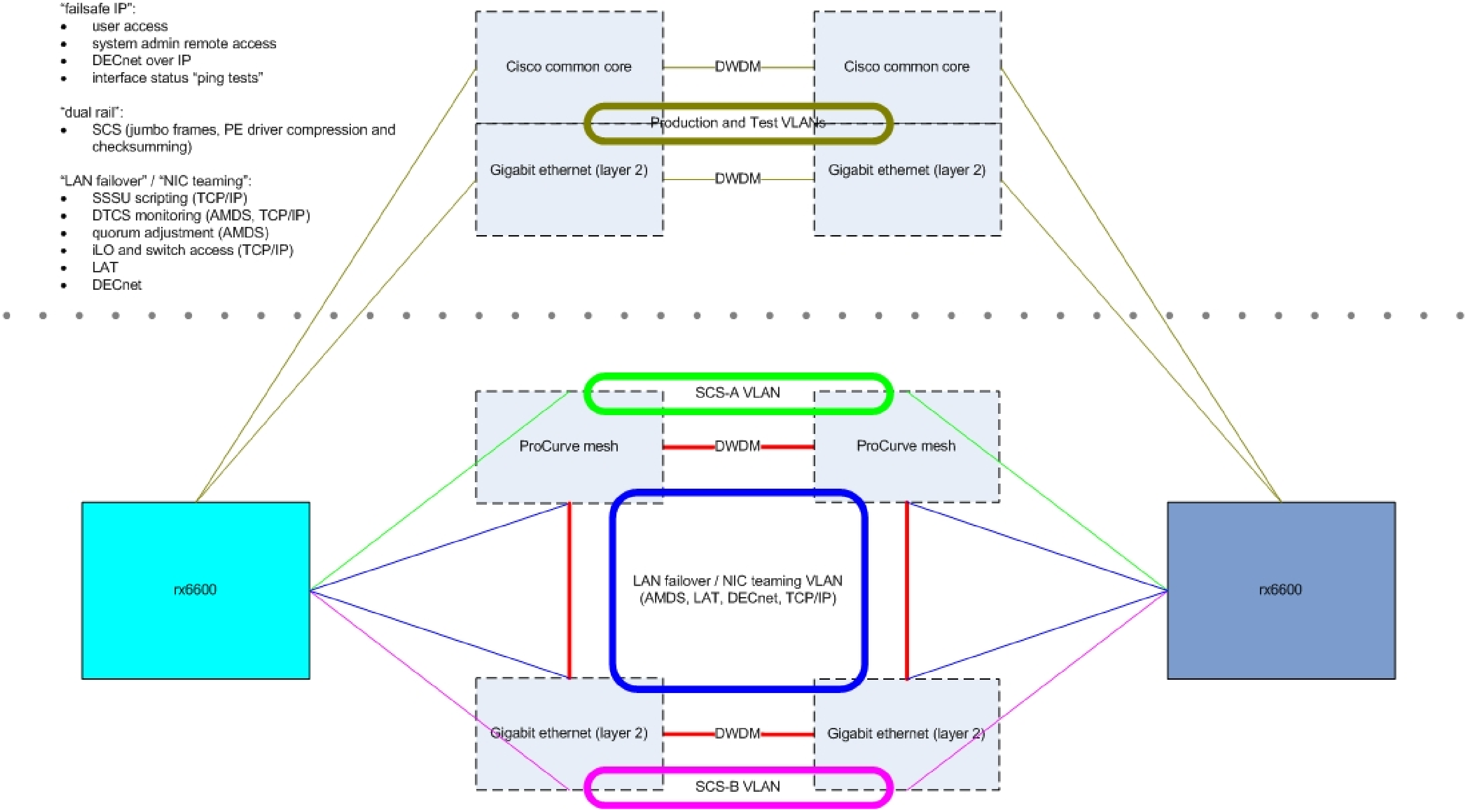
- user access
- system admin remote access
- DECnet over IP
- interface status "ping tests"

"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

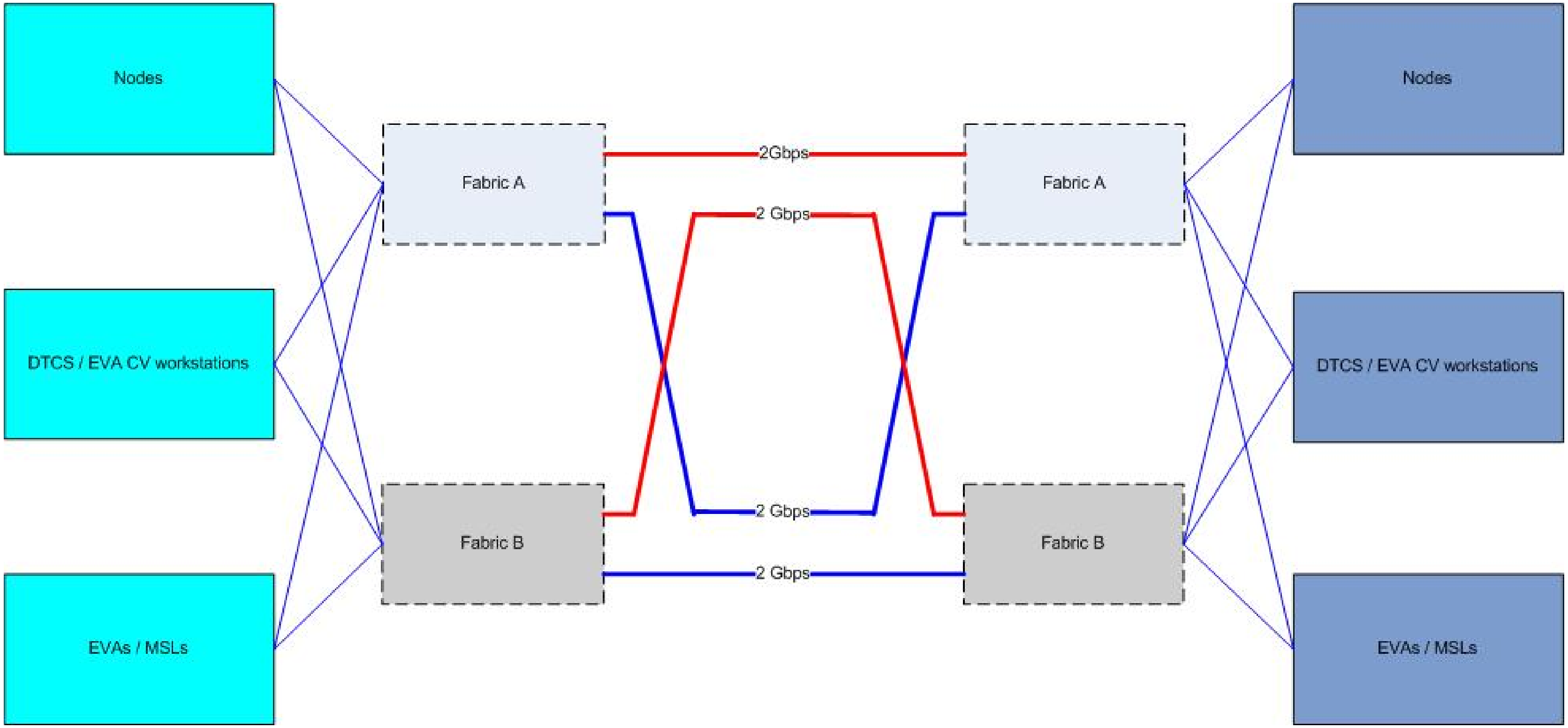

Cisco common core
DWDM
Cisco common core
Production and Test VLANs
Gigabit ethernet (layer 2)
DWDM
Gigabit ethernet (layer 2)
SCS-A VLAN
ProCurve mesh
DWDM
ProCurve mesh
rx6600
LAN failover / NIC teaming VLAN
(AMDS, LAT, DECnet, TCP/IP)
rx6600
Gigabit ethernet (layer 2)
DWDM
Gigabit ethernet (layer 2)
SCS-B VLAN

Inter-site storage (SAN) links

- Use direct path fibrechannel with SAN extension
- Avoid path switching by dual-path connection per fabric
- Enable MSCP as an alternate path mechanism
- Use mini-copy and mini-merge
- Avoid cross-site booting
- Only mount site-specific discs at their site, even if shadowed to all sites (eg: per-site shadowed system discs)

Note: beware recent bug with minicopy “policy=dismount”

Example SAN connectivity



Storage connectivity

- Fibrechannel uses WWIDs:
- WWN = World Wide Name
- WWNN = World Wide Node Name (points to entire array or tape drive or multi-port HBA)
- WWPN = World Wide Port Name (point to specific port in array controller or tape drive or HBA)

- Zoning – single initiator, multiple target, use WWPN

- Storage element presentation to HBA
- OpenVMS uses UUID to set device name

Example disc layout

- Maximum of six arrays and three sites
- Three environments (Production, Test, Development)
- Systems boot from fibrechannel
- All fibrechannel discs shadowed:
  - System and common discs
  - Data discs
- Array based copies for backup (snaps, clones)

- All local discs (partitioned RAID) used by local node only
  - Page/swap/dump/T4/”DVD” discs
  - Local “full” boot for system maintenance

Example disc naming – FC discs

- DSA10 ($1$DGA1010, 2010, 1110, 2110) – common
- DSA11 - system disc, site A
- DSA12 - alternate system disc, site A
- DSA13 - system disc, site B
- DSA14 - alternate system disc, site B
- DSA15 - system disc, site C
- DSA16 - alternate system disc, site C
- DSA21 … DSA39 – data (small shadow sets)
- etc.

Example disc naming – local discs

- 8 slot SAS array, RAID 6, 2x hot spares, BBWBC
- DKA0 - page/swap/dump files
- DKA1 - T4 data
- DKA2 - local boot (non-clustered, full system)
- DKA3 - local <SYSE> boot (STABACKIT)
- DKA4 - copy of OpenVMS DVD media + kits etc.

Shadowing

- Many shadow sets for performance with multi-path discs
- Small shadow sets to minimise copy/merge time (especially common disc)
- Enough arrays per site to always have local source
- Only mount system discs on nodes booted from that disc
- System disc at a site is shadowed to other sites
- Use minicopy and minimerge for performance

Note: Beware bug with mincopy and dismount keyword in policy – see customer advisory on HP OpenVMS web site

Array configuration

- Use RAID 0+1 (EVA vRAID1) for best performance
- Use double sparing, single disc group (EVA)
- Snaps are only a short-term point in time temporary entity – they can hurt array controller performance
- Clones have better performance, but require more space
- Consider explicit path specification and explicit controller preference for preferred path configuration

Note: 3Par cannot change UUID after VV is created

Booting

- Requires firmware support for HBA and array
- Boot drivers are lightweight
- View from EFI shell is extremely hard to interpret
- Use BOOT_OPTIONS.COM to configure boot paths, or use efi$bcfg.exe directly (see command line help)
- When adding a node to an existing cluster, ALWAYS mount the target system disc READ ONLY
- Delete root <SYS0> to avoid unexpected booting with unconfigured hardware

Identifying discs from EFI shell (1)

- Create a “flag file” in the EFI partition:

```
SYSTEM on RX2660 $ create XD_RX2660_DKB1.TXT
EFI flag file:

Author:        Colin Butcher, XDelta Limited, +44 117 904 8209, www.xdelta.co.uk
System:         rx2660

dkb1 - copy of 8.4 system disc for vsi eft installation test

SYSTEM on RX2660 $ convert XD_RX2660_DKB1.TXT /fdl="record; format stream_lf;"
_Output: XD_RX2660_DKB1.TXT
SYSTEM on RX2660 $ mc efi$cp
EFI$CP> mount DKB1:[VMS$COMMON.SYS$LDR]SYS$EFI.SYS/dev=dkb1efi:/ove=id
EFI$CP> copy SYS$SYSROOT:[SYSMGR]XD_RX2660_DKB1.TXT/text dkb1efi:
EFI$CP> dismount dkb1efi:
EFI$CP> exit
```

Identifying discs from EFI shell (2)

```
…
  fs0          :   Acpi(HPQ0002,PNP0A08,400)/Pci(0|0)/Pci(0|0)/Scsi(Pun0,Lun0)/HD(Part1,Sig11726511-375E-11E3-A219-
AA000400FFFF)
  fs1          :   Acpi(HPQ0002,PNP0A08,400)/Pci(0|0)/Pci(0|0)/Scsi(Pun0,Lun0)/HD(Part3,Sig11726510-375E-11E3-A21A-
AA000400FFFF)
  fs2         :   Acpi(HPQ0002,PNP0A08,400)/Pci(0|0)/Pci(0|0)/Scsi(Pun0,Lun1)/HD(Part1,SigB7852DB1-E5B4-11E4-BC94-
AA000400FFFF)
…

…
fs2:\> type XD_RX2660_DKB1.TXT
File: fs2:\XD_RX2660_DKB1.TXT, Size 169
EFI flag file:

Author:       Colin Butcher, XDelta Limited, +44 117 904 8209, www.xdelta.co.uk
System:        rx2660

dkb1 - copy of 8.4 system disc for vsi eft installation test
fs2:\>
```

Quorum and voting

- Is application “cluster aware” or rapid failover ?
- What do you want to happen when a site fails ?
- Avoid quorum disc if possible
- HP VM based quorum node can be useful
- Availability manager / DTCS quorum adjustment
- <Ctrl-C> quorum adjustment on Integrity

Log file management

- Fragmentation is a problem worth avoiding
- Use LD containers: write log files to the LD device, then simply move containers to archive.
- Block net$server.log (and others) by creating an empty ;32767 version

Hardware maintenance and replacement

- Keep firmware up to date – plan sequence to avoid disruption
- FC devices with same UUID but different WWPNs will show up as the SAME device but with extra paths
- Keep systems modular with minimal configuration per node
- Save / restore ILO configurations with USB flash drive

Performance engineering

- Avoid guesswork - run T4 all the time

- Other good tools: Perfdat, SDA extensions

- Without good data you cannot do good performance work

- A faster machine just waits more quickly
- Don’t make it go faster, stop it going slower
- The fastest IO is the IO you don’t do
- The fastest code is the code you don’t execute

Availability manager

- Windows management stations (typically one per site)
- Gives “real time” view of nodes in management group(s)
- Uses AMDS protocol (layer 2 – use LL or VL device)
- Interacts with OpenVMS driver at high IPL
- Permits modification of running system:
    - Quotas
    - Dynamic parameters
    - Quorum

Cockpit Manager for OpenVMS

- Agent software per-node with:
  - Monitoring of cluster member nodes

- OpenVMS management stations (typically one per site) with:
  - Monitoring:
    - Storage arrays (SNMP)
    - Storage infrastructure (SNMP)
    - Network infrastructure (SNMP)
    - Reachability (PING etc.)
  - Console access and logging via ILO
  - Remote DCL command execution
  - Notification (SMS etc.)

DTCS for OpenVMS

- Per-node software with:
  - Control of multi-site shadow set formation on boot, supporting up to 6-way shadowing
  - Rule based monitoring of cluster member nodes

- Windows management station (typically one per site) with:
  - Rule based monitoring:
    - storage arrays (WEBES, SNMP etc.)
    - Storage infrastructure (SNMP)
    - Network infrastructure (SNMP)
    - reachability (PING etc.)
  - Console access and logging via ILO
  - Alerts and notifications (email etc.)

Wish list!

- Single node cluster licence as part of base OS
- Single member shadow sets as part of base OS
- ALLOCLASS per storage array / tape library (WWNN based?)
- Do not start SCS / DECdns etc. by default on all NICs
- What else ? Let us know!

OpenVMS HA and DT clusters

Designing and implementing high availability and disaster tolerant OpenVMS clusters

Colin Butcher
XDelta Limited

www.xdelta.co.uk
+44 117 904 8209