Best practices for OpenVMS clusters

The design, build, operation and support of OpenVMS systems and clusters

Connect Germany 2018 (Leipzig)

Colin Butcher (XDelta), with contributions and collaboration from Keith Parris (HPE Pointnext) and Nic Clews (DXC)

xdelta

Agenda

- OpenVMS boot, startup and shutdown (Alpha, Integrity)
- Firmware and operating system updates (Alpha, Integrity)
- Monitoring and house keeping
- Storage infrastructure (Fibrechannel, Local SAS / SCSI)
- Volume shadowing
- Network infrastructure
- Logical and physical structure of clusters
- Multi-site clusters
- Inter-site links and latency (IO write rate limits)
- Discussion

Examples from XDelta lab

- MSA1000 FC storage array
- ES40 Galaxy two node cluster:
  - Partitioned ES40 (two partitions)
    - 2x FC HBA, 1x GigE per partition
    - SMCI cluster interconnect
    - System disk as quorum disk
    - Local SCSI RAID as page / swap
- rx2660 single node cluster:
    - 4x FC HBA, 6x GigE
    - Local SAS RAID as page / swap

xdelta

OpenVMS boot and startup

OpenVMS boot and startup

- Power-up sequence
- Console view of hardware and devices
- Boot process
- Base system startup
- Device configuration
- LAN devices, pseudo-devices and IPCI (if needed)
- Cluster formation
- STARTUP.COM and site-specific SY*.COM files
- Network transports startup
- Layered products startup
- Application startup

Power-up sequence

- Hardware tests: CPUs, memory, devices, etc.
- Can disable tests for speed of power-up
- Auto-boot – disable in HA/DT environments
- Alpha console:
  - serial port (or MBM on last Alphas)
    - >>> prompt, some commands are machine specific
    - partitioning and galaxy
    - wwidmgr for FC devices
- Integrity console:
  - ILO
    - EFI / UEFI menu interface
    - Make EFI shell the first option (safe)

Finding devices to boot from

- Alpha console:
  - wwidmgr for FC devices
  - boot* environment variables (bootdef_dev etc.)
- Integrity console:
  - Decoding the view from EFI shell (FS<nn> devices)
  - FS<nn>: \efi\vms\ EFI scripts
- Boot flags to specify system root etc.

Boot loader

- Alpha:
  - APB on target system disk / root
- Integrity:
  - IPB on target system disk / root
  - EFI shell sees a FAT partition
  - FS<nn>:\efi\vms\vms_loader.efi is the initial loader
  - VMS sees the ODS2/5 file system
  - SYS$EFI.SYS is a container file within the VMS file system
  - EFI$CP is an undocumented tool to access SYS$EFI.SYS

OpenVMS early stages of boot

- Minimalist drivers for boot (and dump)
- Reads system parameter file
- If flag set, enters SYSBOOT> to make parameter changes
- Obtains paths to all system disks and dump device (all system disk shadow set members need to be in the list)
- Configures LAN devices (LL, VL, IPCI), before clustering
- Loads MSCP for disk serving, before clustering
- Forms / joins cluster
- Activates local page files (if present)

- Hands over to SYS$SYSTEM:STARTUP.COM

OpenVMS post-boot startup

- SYS$SYSTEM:STARTUP.COM
- Startup database and phases (SYSMAN STARTUP)
- Devices (SYS$DEVICES.DAT and SYSMAN IO EXCLUDE)
- SY*.COM files (site specific, system platform actions):
  - SYCONFIG
  - SYLOGICALS
  - SYPAGSWPFILES
  - SYSTARTUP_VMS

- Application startup actions separated from system platform actions (disks, queues, databases, service addresses, etc.)

Devices

- Fibrechannel tape:
  - SYSMAN IO FIND_WWID and IO CREATE_WWID
  - See SYS$DEVICES.DAT
- Exclude unused devices:
  - SYSMAN IO SHOW EXCLUDE
  - SYSMAN IO SET EXCLUDE = “<list>”
- LANCP DEFINE LL<nn> / VL<nn> etc.

SYS$SYSTEM:STARTUP.COM

- STARTUP_P2 sysgen parameter, or
- SYSMAN STARTUP SET OPTIONS

- Enable log output to SYS$SPECIFIC:STARTUP.LOG

- Enable information output re startup stages (checkpoints)

- SYSMAN STARTUP SHOW OPTIONS

STARTUP database

- SYSMAN STARTUP SHOW FILE /FULL
- Some layered products add themselves to the startup database
- Personal preference – call layered product and application startup from SYSTARTUP_VMS.COM

SYCONFIG.COM

- Called very early in startup, also called by AUTOGEN, so check if running in startup mode

- Typical uses:
  - Disable SCS on unused paths with SCACP STOP LAN <dev>
  - Set RMS defaults
  - Configure Audit Server (eg: disable)
  - Configure DECdtm transaction journaling (eg: disable)
  - Configure DECwindows (eg: disable workstation components)
  - Configure DECnet-Plus (eg: prevent DECdns and DTSS starting on unused interfaces)

SYLOGICALS.COM

- Called after SYCONFIG

- Typical uses:
  - Set mini-copy / mini-merge policies
  - Check / mount system disk shadow set members
  - Mount common disk shadow set members
  - Define logical names for files on common disk (UAF etc.)

- Keep this for system platform logical names etc.

SYPAGSWPFILES.COM

- Called after SYLOGICALS

- Typical uses:
  - Mount page / swap / dump disks (eg: local SAS RAID)
  - Create directory structures and page / swap / dump files if needed (;32767 prevents modification)
  - Delete page / swap / dump files from system disk if they still exist (rename, then delete after reboot)
  - Install page / swap files

SYSTARTUP_VMS.COM

- Called at end of startup sequence

- Typical uses:
  - Mount additional shadow sets
  - Start network protocols
  - Start queue manager
  - Start layered products
  - Start 3rd party products
  - Call application startup sequence

Application startup after system startup

- Separated from system startup sequence
- Called by SYSTARTUP_VMS.COM as final action
- Easy to control application startup on boot with SYSGEN user defined system parameters, so no need to edit SYSTARTUP_VMS etc.
- Calls <APPNAME>_STARTUP.COM

Application startup sequence

- Top level command file <APPNAME>_STARTUP.COM

- Calls a set of command files for actions, eg:
  - <APPNAME>_DISK_MOUNTS.COM
  - <APPNAME>_LOGICALS.COM
  - <APPNAME>_BATCH_QUEUES.COM
  - <APPNAME>_PRINT_QUEUES.COM
  - <APPNAME>_ENABLE.COM

Firmware and OS updates

Alpha firmware

- Alpha firmware CD, all on single load media
- Boot from CD
- Update firmware components
- Restart system

Integrity Server firmware

- Older hardware (prior to –i2) will require USB flash drive media for all firmware components
- Newer hardware (-i2 onwards) can use HPSUM firmware bundles for the base system, loaded over the network from Windows / Linux via the ILO
- Some firmware components will not load from the HPSUM bundle, but require USB flash drive media (SAS controller, LOM controller, HDDs, etc.)
- Power-cycle system and restart

Integrity Server firmware

- HPSUM bundles – execute with operating system shut down, but with server powered up
- Firmware components such as SAS, LOM, etc. – apply manually at EFI shell from components on USB media
- It is possible to load firmware into the FAT partition on an ODS 2/5 disk using EFI$CP, but this is unsupported
- Available HDD firmware load images for manual loading appear to be several versions behind those in the bundles

OpenVMS patching

- Patch all system disks
- Takes effect on reboot
- Use array controllers to make clones of system disks for rapid recovery
- Have alternate system disks for mixed version running, hardware replacement, etc.
- Consider the common disk with single UAF etc.

Planning the work

- Read all release notes carefully (especially firmware)
- Determine any pre-requisites for firmware versions etc.
- Determine current firmware versions of all components
- Capture current configurations
- What else needs updating (FC switches, storage arrays, network switches, tape libraries and drives, etc.)?
- How much downtime is permissible?
- When is downtime permitted?
- Plan access to data centres and equipment
- Record pre-update performance data
- Identify any existing issues, then decide whether to proceed

Overall sequence

- Check that everything is working properly
- Save current configurations of everything
- Update infrastructure components, should be possible with cluster nodes up and running:
  - FC switches (fabric by fabric, will trigger path switching)
  - Network switches (switch by switch, depends on topology)
  - Storage arrays (should be possible online, cluster by cluster)
  - Tape libraries and drives
- Update cluster nodes (cluster by cluster)
- Update Windows servers (monitoring, load hosts, etc.)

Sequence for minimal node downtime

- Make all nodes that boot from a given system disk quiescent (nothing running)
- Shut down all except one of those nodes if possible
- Apply OpenVMS patches
- Shut down node, leave powered up
- Use manual method via EFI shell / USB media for SAS, LOM, etc. firmware
- Use HPSUM to apply firmware bundle
- Reboot and test
- Boot other nodes and test
- Return to service

Planning for minimal downtime

- Reboot / power-cycle takes a long time, so minimise the number of such operations
- Do what you can in advance – preparation is key
- It may be shorter elapsed time and less complex to take the entire cluster and work on all nodes simultaneously
- Have checklists and backout plans

Hardware replacement and DC moves

Replacing all components on-the-fly (1)

- No free space, so remove old items first

- More ports in ethernet switches
- Firmware compatibility between ethernet switches
- Swap out one ethernet switch at a time

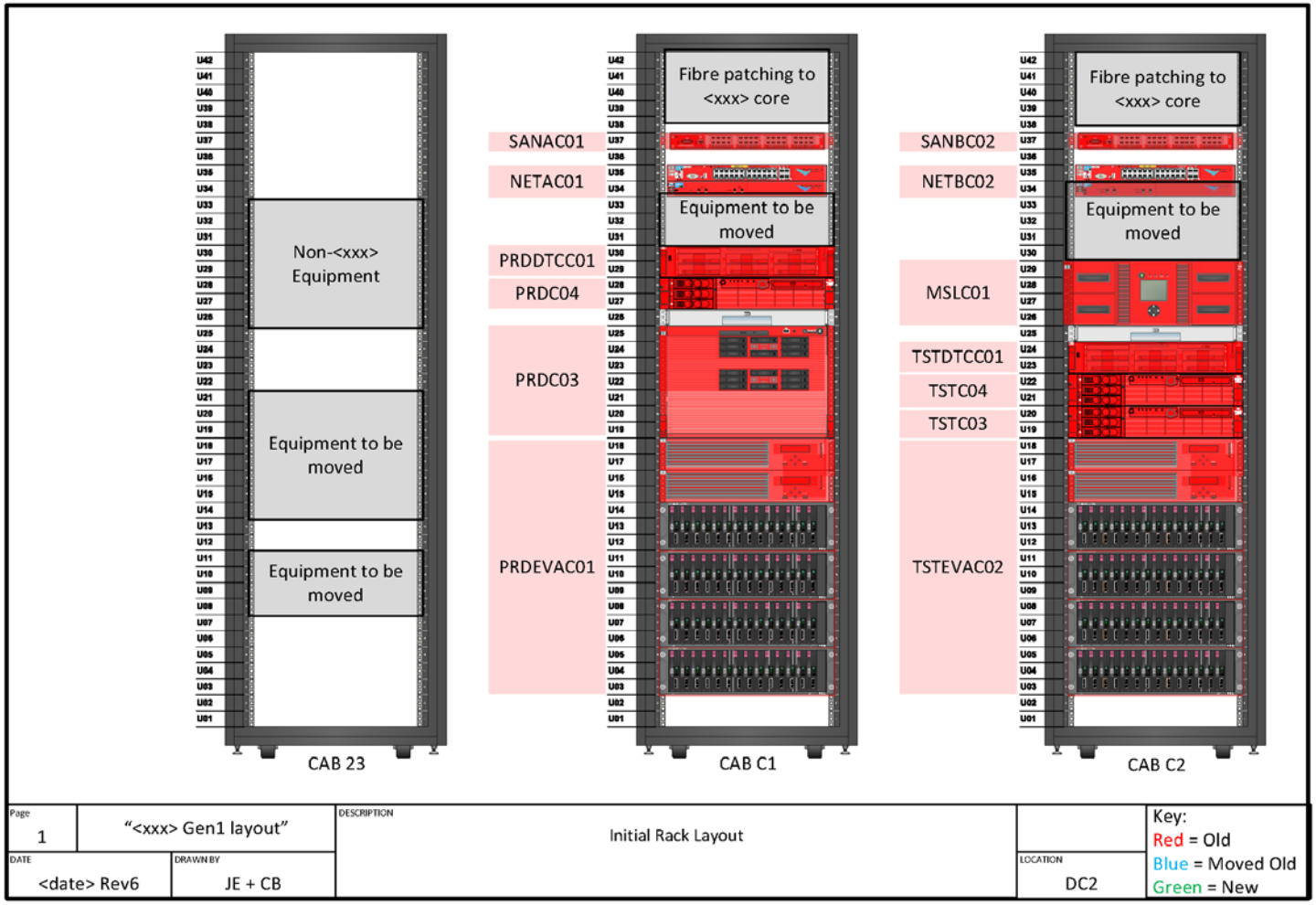
- FC switches with old and new zoning combined
- Swap out one fabric at a time
- EVAs – same UDID / LUN, but different WWID
- OpenVMS sees same devices, but path count changes
- Shadow copy to new storage takes a long time!

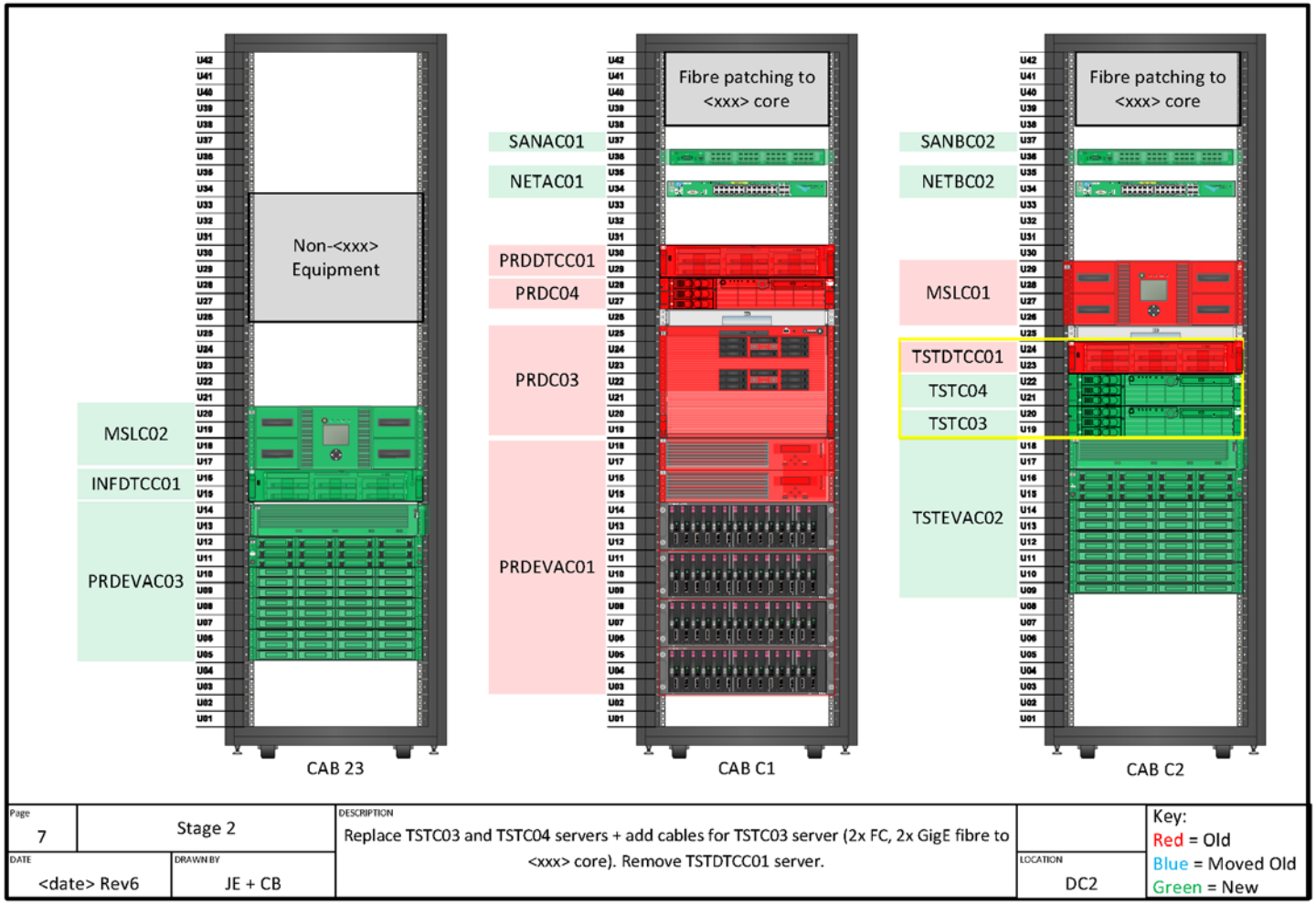
Replacing all components on-the-fly (2)

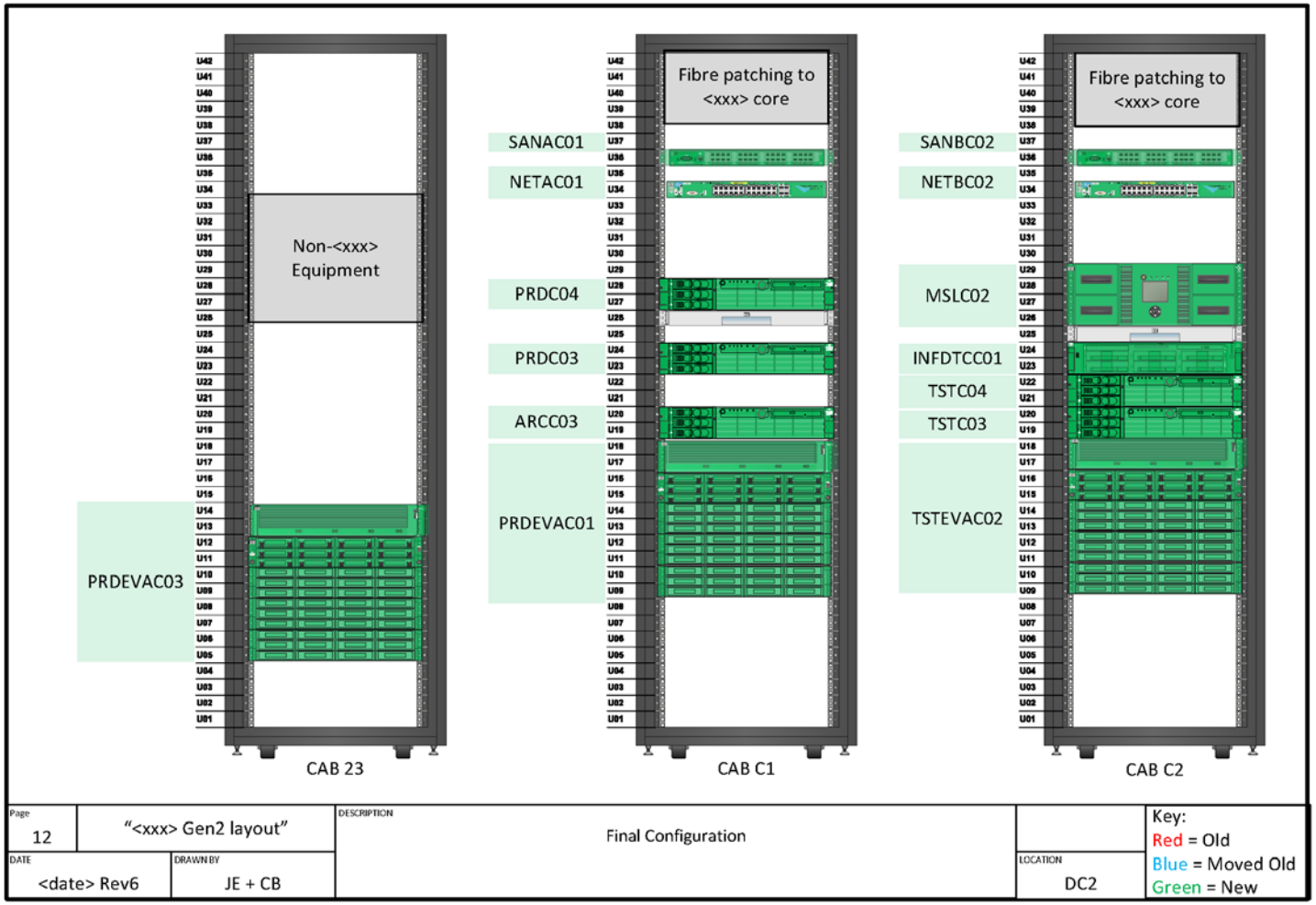
- Rack-mount servers (rx2660 / rx6600 to rx2800-i2)
- New cabling to switches for parallel running
- Local SAS “maintenance boot” for testing connectivity
- Alternate FC system disks per site (for h/w support)
- Kept split-site cluster up during entire replacement
- Detailed planning and check lists
- No single person working

Example of planning detail

- Everything removed and replaced in DC1 and DC2
- Drawings show start, mid-point and end for DC2
- Accompanied by:
  - Detailed check lists
  - Timeline
  - Assignment of responsibilities
  - Contingency plans
  - Back-out plans
- Everything worked as anticipated
- Months of planning for a few days work!







Moving data centres without loss of service

- Pre-cable at new site (every cable needs testing!)
- Moved ethernet switches, FC switches, EVAs and servers
- Local SAS “maintenance boot” for testing
- Wrote DCL to test every fibre path to storage overnight
- Different inter-site distances, so latency changes
- Moved from DWDM over dark fibre to MPLS pseudo-wires
- Temporary configurations for “half-cluster” single-site running during moves
- Kept cluster up during entire replacement
- Detailed planning and check lists
- No single person working

Monitoring and “house keeping”

Monitoring and alerting – Why it matters

- Monitoring is essential in a high availability system with no single points of failure.
- HA and DT systems do their best to survive failures.
- The second failure kills you if you don't detect the first!
- You need to know what's going on, even if it's nothing
- You need good information to understand what happened when something fails

Monitoring and alerting

- Application availability, depends on:
  - All related physical equipment
  - Cluster interconnects
  - Storage volumes
  - Network connections
  - Performance behaviour
  - Queues, databases, 3rd party products, etc. …

- It's the combination of data from different sources that enables you to understand what's happening
- Implement periodic tests and checks

House keeping

- Disk space usage / free space
- File fragmentation
- Disk fragmentation
- Shadow set consistency
- Log files
- Audit files
- Backup / archive records
- Etc. …

- Don’t let log and audit files become too big and unwieldy
- Clean up files on a cycle time that makes sense in the context of your operational regime

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

Performance monitoring

- Free T4 for long-term monitoring and trend analysis
- Implement your own modules to instrument your application

- Availability Manager (AMDS)

- More comprehensive performance monitors, eg:
  - TDC
  - Perfdat

- Find a set of tools that works for you.

Performance engineering

- Avoid guesswork - run T4 all the time
- Useful tools: 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
- The idle loop is anything but idle

Console logging and monitoring

- Console logging can alert you to a developing situation or a transient problem.
- Consider how much OPCOM output you want – if logging all consoles, maybe restrict OPCOM output to be per-node
- Find a set of tools that works for you, for example:
  - DTCS includes IAM Consoles
  - Cockpit Manager
  - TDI Consoleworks
  - Networking Dynamics

DTCS for OpenVMS (now with DXC)

- 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.)

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

Cluster design

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	?	?

xdelta

Naming conventions

- Choose your naming conventions very carefully – they are the hardest thing to change later
- Don’t tie nodenames to physical locations. Physical server names different to nodenames.
- Choose disk 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 server (ILO) naming convention

S<nn2>DC<n3>, where:

“S” = Server (the physical machine and ILO)

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

“DC” = “data centre” (site)

<n3> = 1 … 9 (site number)

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 disks

- 1DGA<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> = disk, matches DSA<nn> shadow set name

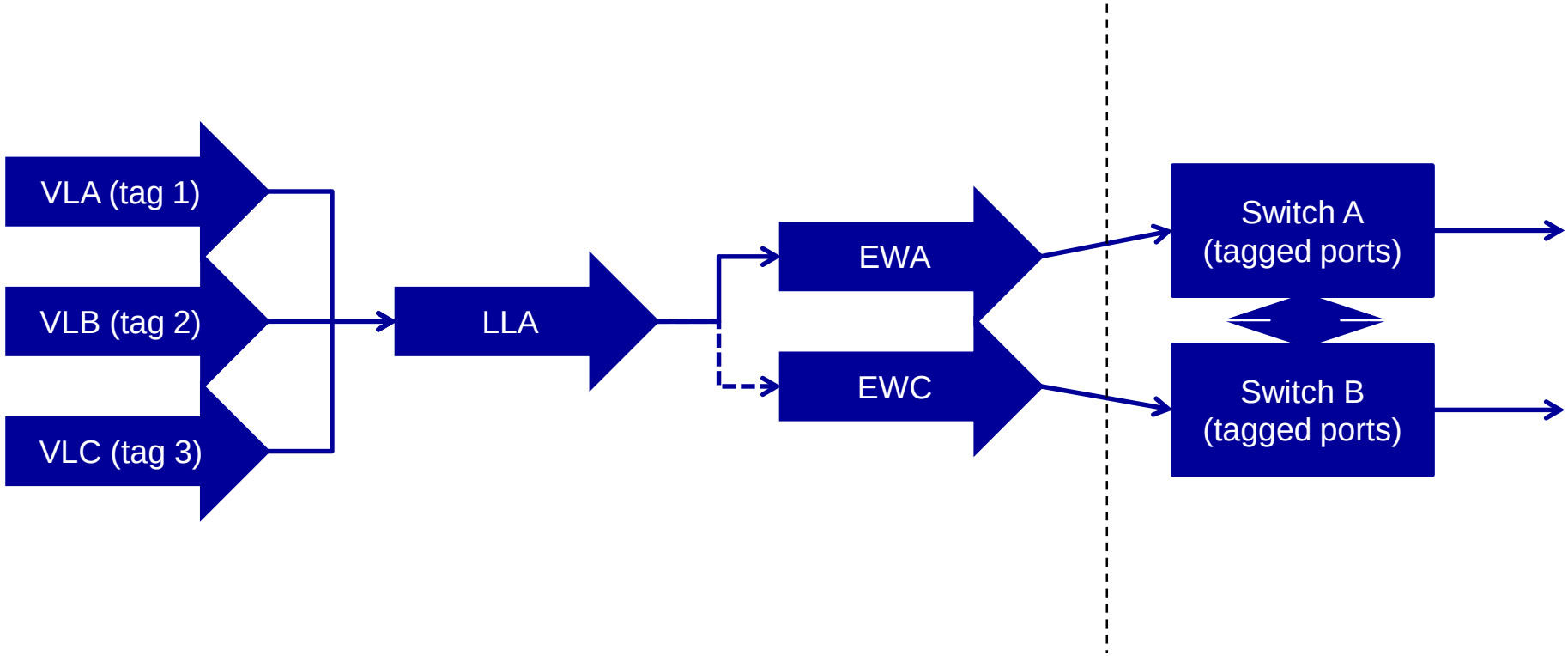
Note: UUID must be unique across the fabric, ALLOCLASS = 1

OpenVMS network connectivity

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



xdelta

OpenVMS storage 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

Inter-site links (storage)

Inter-site links - storage

- MSCP over PEdriver:
  - Poor for performance, useful fallback
  - Requires cluster member nodes at remote site
- FC extension (DWDM over dark fibre):
  - Best for performance
  - Set inter-switch link port characteristics
- FC over IP:
  - Need low latency and low jitter IP network
  - Use “fast write” in FCIP devices

Multi-site cluster – storage

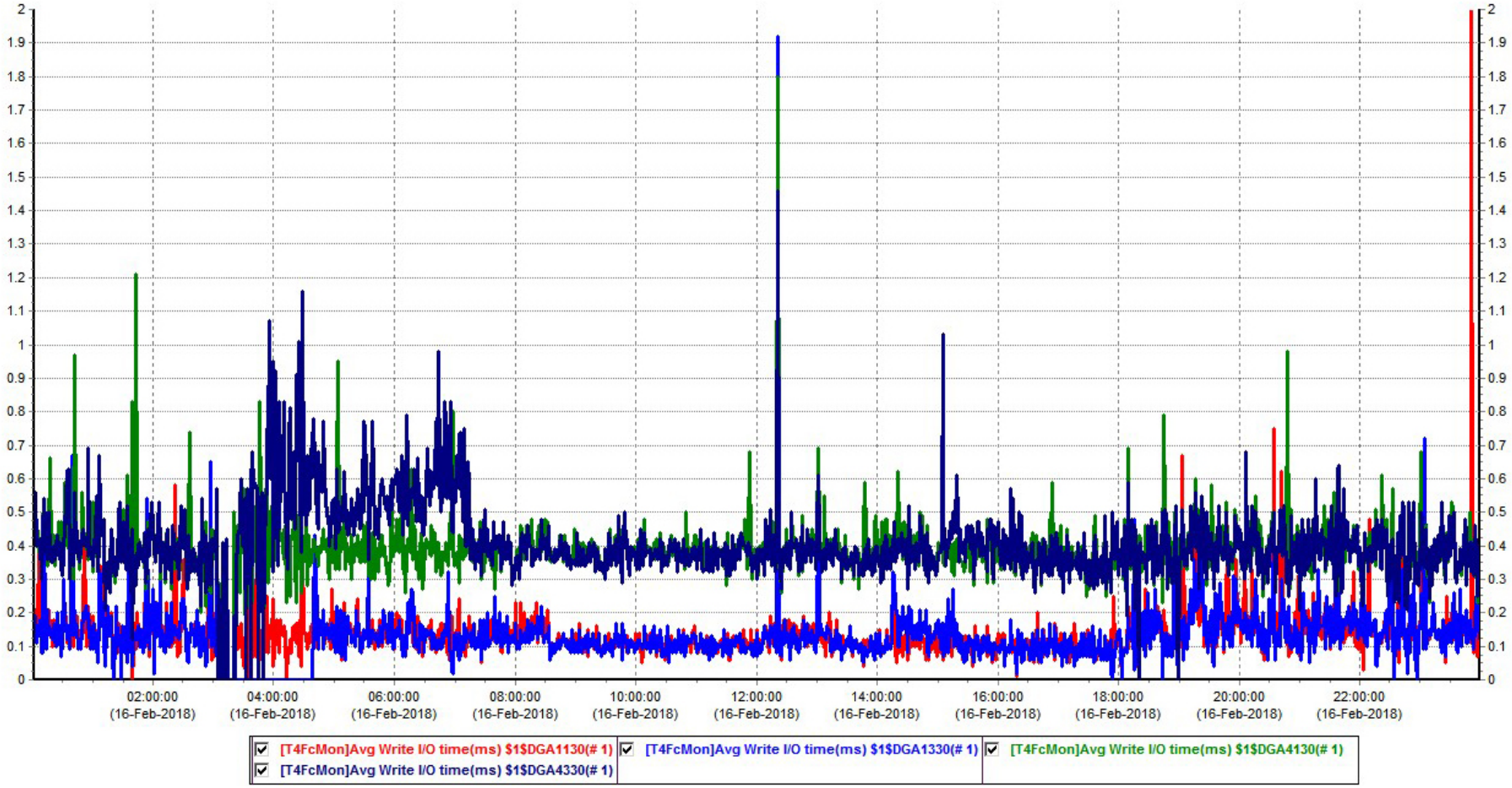
- MSCP over PEdriver:
  - No direct path to MSCP served devices at remote sites
  - Greater probability of triggering shadow merge

- FC extension and FC over IP:
  - Multi-path fibrechannel devices
  - Direct path to devices from all nodes
  - Set the SITE value to bias reads from local site

Multi-site cluster – storage performance

- Reads should be biased from local storage
- Shadowing writes are synchronous, so remote shadow set members are the limiting factor.
- Write latency is a combination of:
  - Local IO write latency (system to local storage array)
  - ISL endpoint latency (FC switches etc.)
  - Distance latency
- IO write rate: 1ms = 1000 IOs per second, 2ms = 500, …

Multi-site shadow set (4 members)

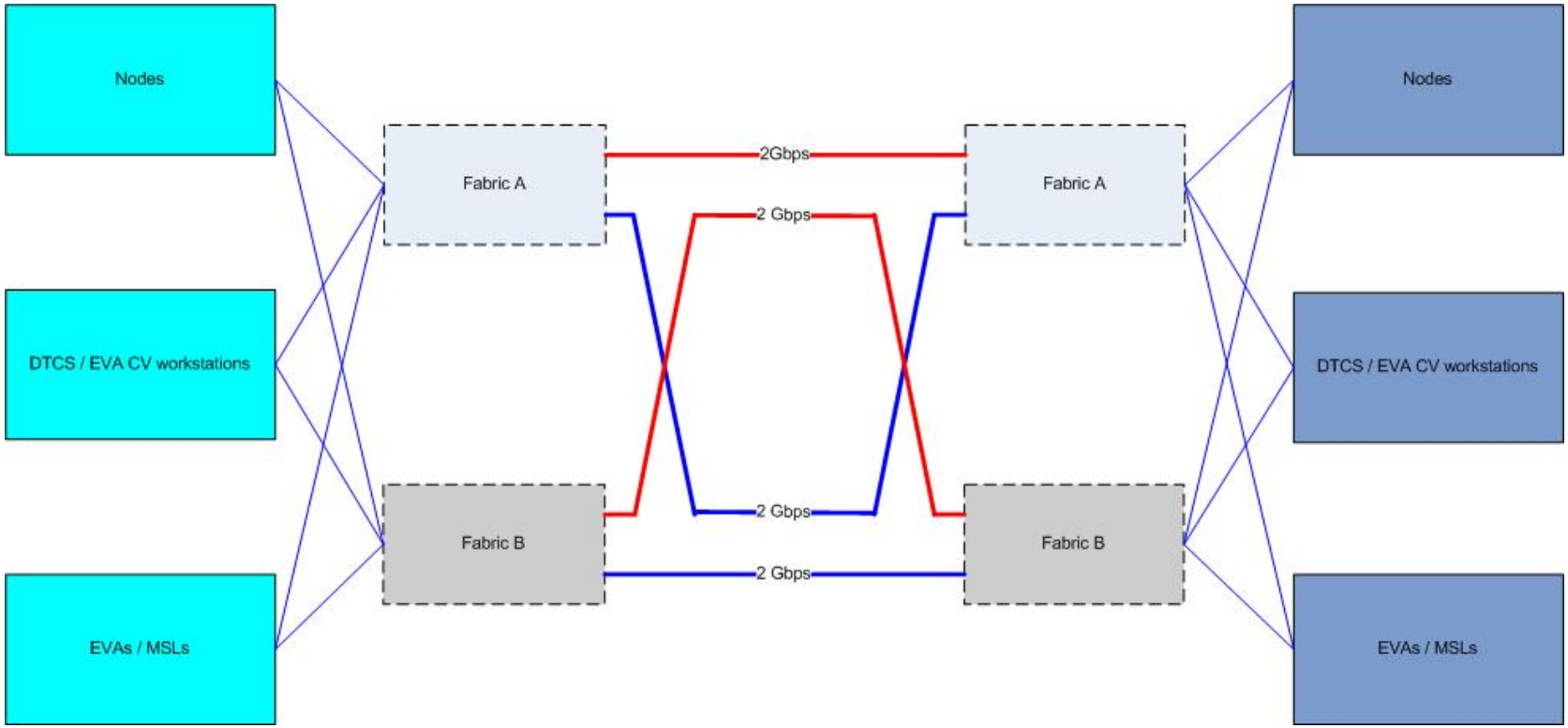


xdelta

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 disks at their site, even if shadowed to all sites (eg: per-site shadowed system disks)

Example SAN connectivity



xdelta

Inter-site links (data networking)

Inter-site links – data networking (1)

- 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

Inter-site links – data networking (2)

- LAN extension – layer 2 (DWDM over dark fibre):
  - Best for performance, lowest latency
  - Passes all layer 2 protocols, so use SCS, AMDS, etc.
- LAN extension – layer 2 (MPLS pseudo-wire):
  - Variable latency, depends on topology and contention
  - Passes all layer 2 protocols, so use SCS, AMDS, etc.
- TCP/IP – layer 3:
  - Variable latency, depends on topology and contention
  - IP only, use IPCI, no AMDS or other layer 2 protocols

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”

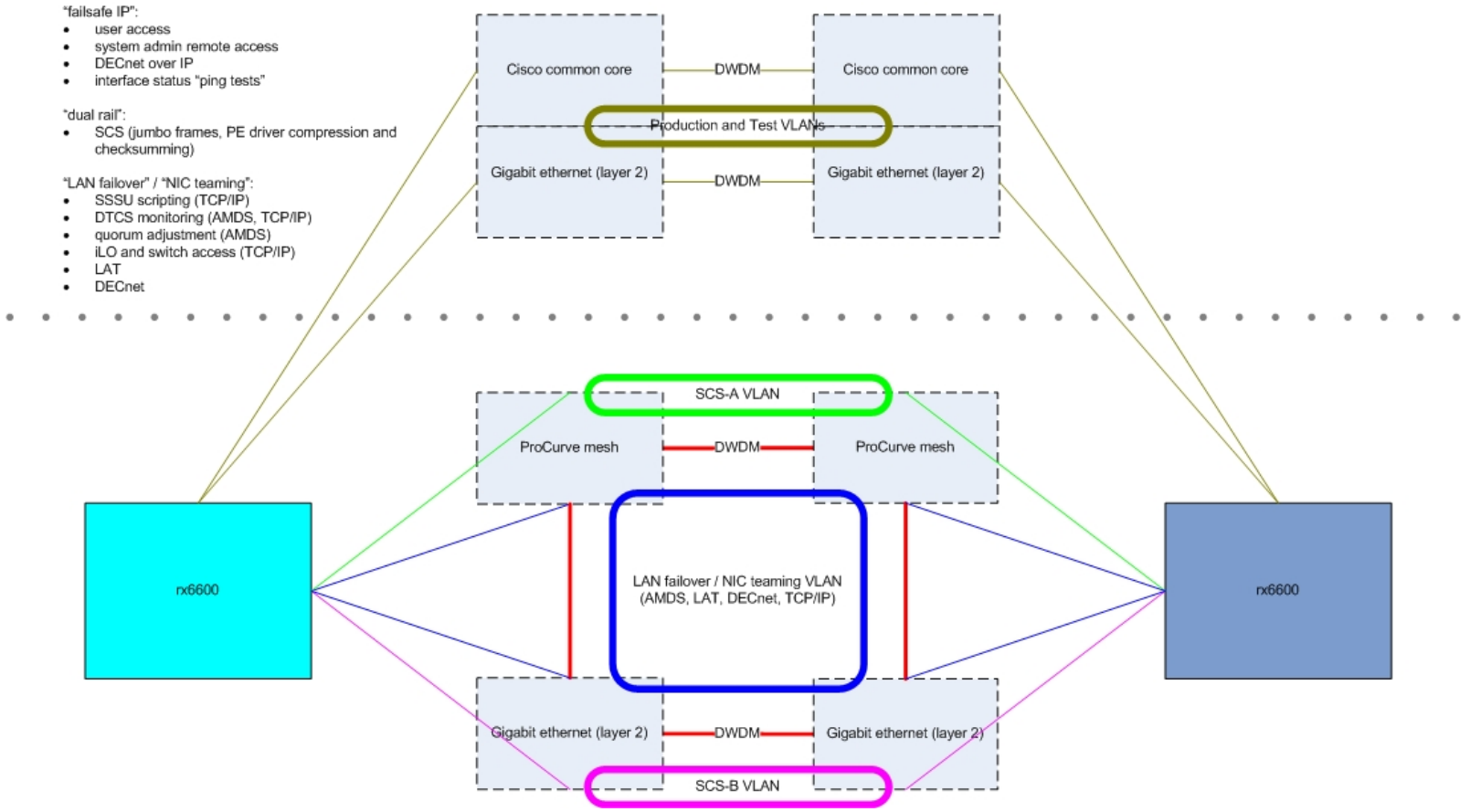
Multi-site cluster – data networking

- LANCP to configure and manage LAN devices
- LLDRIVER for LAN failover
- VLDRIVER for VLANs

- PEDRIVER implements channels and virtual circuits
- SCACP to manage PEDRIVER
- Disable SCS on unused LAN devices

- Enable jumbo frames if network infrastructure permits

Example data network connectivity



Multi-site cluster – cluster interconnects

- How many channels should we have?
- Depends on network infrastructure

- Single LAN failover device – simple, single channel, multiple NICs, performance and failover behaviour depends on LLDRIVER

- Multiple LAN devices – more NICs (or VL devices), multiple channels, better performance and failover behaviour

- VLANs – tagged (VLDRIVER) or untagged?

HPE POINTNEXT

Maximizing the Performance of Your OpenVMS Cluster Interconnect

OpenVMS Boot Camp 2017, SID 301

Keith Parris, Engineer

Hewlett Packard Enterprise

SCS and cluster performance

- OpenVMS System Communication Services (SCS) uses credit-based flow control
- Each connection between SYSAPs is assigned a number of credits, which is the number of sequenced messages OpenVMS can send over the connection before having to wait for an acknowledgement
- If we want to send a message but have no credits, this is counted as an SCS Credit Wait event
  - Credit waits can be detected via:
    - $ SHOW CLUSTER/CONTINUOUS (CR_WAITS field)
      - ADD CONNECTIONS,REM_PROC_NAME,CR_WAITS and then SET CR_WAITS/WIDTH=10
    - T4 (in the T4_*_SCS.CSV file)
- Two of the SCS credit counts can be user-controlled via SYSGEN parameters:
  - MSCP Serving requests: MSCP_CREDITS (remote disk operations)
    - Default of 32; maximum of 1024
  - VMS$VAXcluster connection: CLUSTER_CREDITS (lock requests, OPCOM messages, CWPS, etc.)
    - Default of 32; maximum of 128
- Underneath SCS is another communications layer with its own independent flow control mechanism

SCS and cluster performance

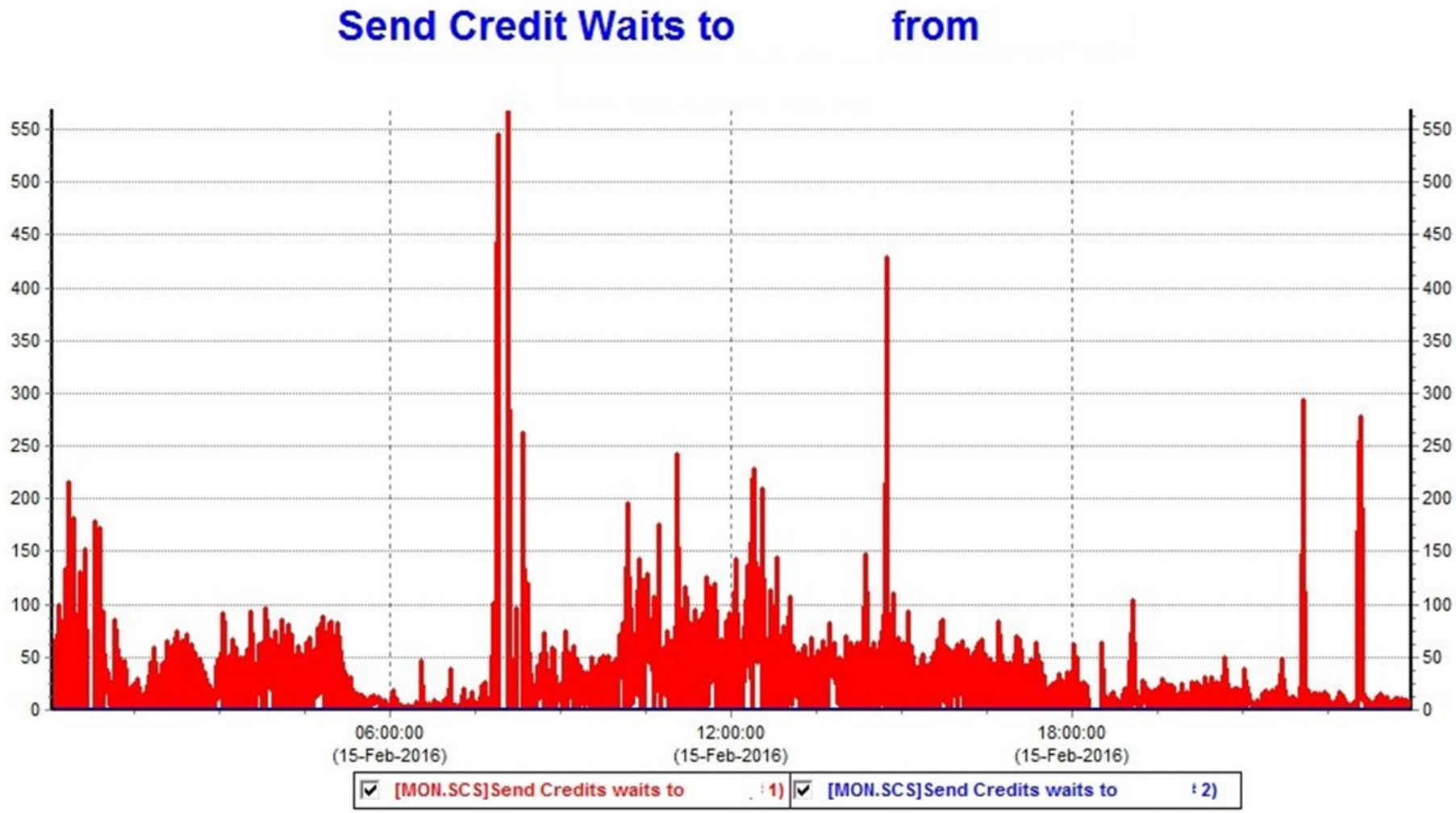
Detecting SCS Credit Waits with $SHOW CLUSTER/CONTINUOUS

```
View of Cluster from system ID 1133  node: NODE9           29-NOV-2012 09:24:55
+--------+--------------+---------+------------------+---------+-----------+
|        SYSTEMS         | MEMBERS |         CONNECTIONS         | COUNTERS  |
+--------+--------------+---------+------------------+---------+-----------+
|  NODE  |   SOFTWARE   |  STATUS |   LOC_PROC_NAME  | CON_STA | CR_WAITS  |
+--------+--------------+---------+------------------+---------+-----------+
|        |              |         | VMS$DISK_CL_DRVR | OPEN    |         0 |
| NODE7  | VMS V8.4     | MEMBER  | SCA$TRANSPORT    | OPEN    |       139 |
|        |              |         | VMS$DISK_CL_DRVR | OPEN    |         0 |
|        |              |         | VMS$VAXcluster   | OPEN    |     36760 |
|        |              |         | MSCP$DISK        | OPEN    |         0 |
| NOD12  | VMS V8.4     | MEMBER  | VMS$DISK_CL_DRVR | OPEN    |         0 |
|        |              |         | MSCP$DISK        | OPEN    |         0 |
|        |              |         | VMS$VAXcluster   | OPEN    |     61575 |
| NODE8  | VMS V8.4     | MEMBER  | VMS$DISK_CL_DRVR | OPEN    |         0 |
|        |              |         | MSCP$DISK        | OPEN    |         0 |
|        |              |         | VMS$VAXcluster   | OPEN    |     36345 |
| NODE6  | VMS V8.4     | MEMBER  | VMS$DISK_CL_DRVR | OPEN    |         0 |
|        |              |         | MSCP$DISK        | OPEN    |         0 |
|        |              |         | VMS$VAXcluster   | OPEN    |     32668 |
└────────┴──────────────┴─────────┴──────────────────┴─────────┴───────────┘
```

HPE POINTNEXT

SCS and cluster performance

Detecting SCS Credit Waits with T4 *_SCS.CSV data, viewed with TLviz



HPE POINTNEXT

SCS and cluster performance

Looking at SCS Credit Waits and SCS credit counts using SDA

Some SCS credit counts are not adjustable at all (e.g. SCA$TRANSPORT, used for queue manager and DECdtm):

```
SDA> SHOW CONNECTIONS
...
VMScluster data structures
--------------------------
          --- Connection Descriptor Table (CDT) 886CC480 ---
State:           0002 open              Local Process:         SCA$TRANSPORT
Blocked State:   0000                   Remote Node::Process: VC5::SCA$TRANSPORT

Local Con. ID    2A910014    Datagrams sent          0    Message queue    886CC4BC
Remote Con. ID   81680012    Datagrams rcvd          0    Send Credit Q.   886CC4C4
Receive Credit          6    Datagram discard        0    PB address       885EF1C0
Send Credit             5    Message Sends           4    PDT address      882AE7B8
Min. Rec. Credit        0    Message Recvs           4    Error Notify     D038C070
Pend Rec. Credit        0    Mess Sends NoFP         4    Receive Buffer   887277E0
Initial Rec. Credit     6    Mess Recvs NoFP         4    Connect Data     8872AAB0
Rem. Sta.    0000000000F8    Send Data Init.         0    Aux. Structure   8872AA00
Rej/Disconn Reason      0    Req Data Init.          0    Fast Recvmsg Rq  00000000
Queued for BDLT         0    Bytes Sent              0    Fast Recvmsg PM  00000000
Queued Send Credit      0    Bytes rcvd              0    Change Affinity  00000000
                             Tot bytes map           0
```

PEDRIVER and cluster performance

- PEDRIVER keeps track of how much time it normally takes to acknowledge a packet on the average, and how much average deviation in this round-trip time occurs. If an acknowledgement hasn't been received within what PEDRIVER considers to be a "reasonable" amount of time, PEDRIVER concludes that the packet must have been lost. This is called a Sequenced Packet Timeout (TMO). PEDRIVER retransmits the packet.
- Potential reasons for a sequenced packet timeout:
  - Packet has been lost by the network. Maybe a CRC error, packet buffers get filled up, or such.
  - The packet got through and was received just fine, but the acknowledgement gets lost going back in the reverse direction.
  - Either the packet or its acknowledgement gets delayed an unusual amount of time within the network

PEDRIVER and cluster performance

- Philosophy of PEDRIVER – or, the PEDRIVER world view:
  - Packets which time out waiting for an acknowledgement, and must be retransmitted, are a sign of network congestion (network overload)
  - Network congestion is solely the fault of PEDRIVER, and only PEDRIVER can solve it
  - PEDRIVER will voluntarily throttle back its utilization of the network in an attempt to prevent network congestion
- The PEDRIVER transmit window size is PEDRIVER's "gas pedal"
  - The Current transmit window size is the number of packets PEDRIVER will transmit before having to wait for an acknowledgement
  - The Current transmit window size starts at 1 at boot time, and climbs slowly upward until it reaches the Maximum
    - The rate the Current transmit window size grows is a function of the rate of successfully-acknowledged packets
  - If PEDRIVER retransmits one packet, it will cut the Current transmit window size to one-half of the Maximum
  - If PEDRIVER retransmits multiple packets within a short period of time, it will cut the Current transmit window size all the way back down to 1
  - When PEDRIVER doesn't have a transmit window slot open, it must wait. This is counted as a Window Full (WinFull) event. Getting lots of these hurts performance.

Fast Path assignments

- $ SHOW FASTPATH displays how OpenVMS assigns device interrupt-state work to CPUs
- Default assignments by OpenVMS tend to spread things pretty randomly across CPUs and this has proven to be sub-optimal in practice
- Intelligent assignment can reduce interrupt-state time and multiprocessor synchronization time
- Rules:
  - Whenever any single CPU is of concern as a potential bottleneck for the system, such as the Dedicated-CPU Lock Manager CPU or the CPU handling locking via PEDRIVER, ***turn hyperthreading off***
  - Certain functions (like TQEs firing) occur on the Primary CPU (normally CPU 0). Try to always avoid saturation of the Primary CPU in interrupt state.
  - Try to place devices with the same driver on the same CPU
  - Try to place drivers that pass packets back and forth on the same CPU. This avoids an inter-processor interrupt, and the associated MP_Synch time.
  - If saturation of any CPU occurs in interrupt state, assign some of its work to other CPUs. This will be sub-optimal in terms of efficiency, but may be necessary.
- T4 *_COMP.CSV data is excellent for looking for interrupt-state saturation of any CPU.

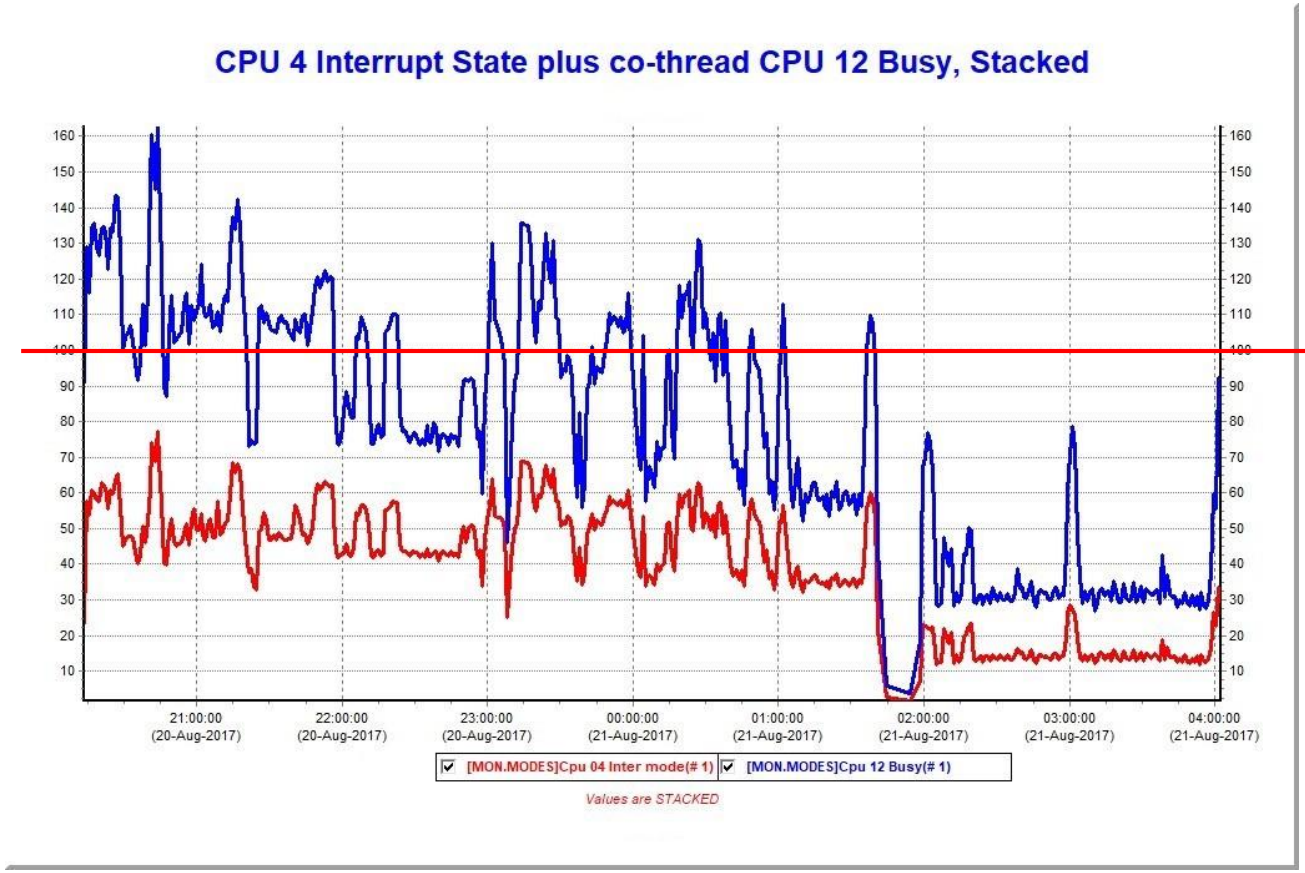
Fast Path assignments

Case study

– Two-site, two-node cluster of rx2800 i2 Servers, two sockets (8 cores) each, hyperthreads enabled

– “Connection Lost” messages periodically, coupled with poor performance. Reboot clears problem.

– PEDRIVER interrupts directed to CPU 4. Co-thread CPU is CPU 12.



Assessing Cluster Interconnect Health

Typical set of data-gathering commands I use:

```
$ SET TERMINAL/WIDTH=132
$ DATE = F$CVTIME(,"ABSOLUTE","DATE")
$ DEFINE/USER SYS$OUTPUT COLLECTED_DATA_'DATE'.TXT
$ MCR SYSMAN SET ENVIRONMENT/CLUSTER
DO MCR SCACP SHOW PORT
DO MCR SCACP SHOW CIRCUIT
DO MCR SCACP SHOW VC /ALL
DO MCR SCACP SHOW CHANNEL /ALL
DO MCR SCACP SHOW IP_INTERFACE /ALL
DO MCR SCACP SHOW LAN_DEVICE /ALL
DO MCR LANCP SHOW CONFIG
DO MCR LANCP SHOW CONFIG /USER
DO MCR LANCP SHOW DEVICE
DO MCR LANCP SHOW DEVICE /REVISION
DO MCR LANCP SHOW DEVICE /CHARACTERISTICS
DO MCR LANCP SHOW DEVICE /COUNTERS
DO MCR LANCP SHOW DEVICE /INTERNAL /DEBUG
DO IF F$SEARCH("SYS$SYSTEM:TCPIP$CLUSTER.DAT") .NES. "" THEN TYPE/HEADER SYS$SYSTEM:TCPIP$CLUSTER.DAT
DO IF F$SEARCH("SYS$SYSTEM:PE$IP_CONFIG.DAT") .NES. "" THEN TYPE/HEADER SYS$SYSTEM:PE$IP_CONFIG.DAT
DO SHOW FASTPATH
DO SHOW CPU/FULL
DO IF F$SEARCH("SYS$TEST:HTHREADS.EXE") .NES. "" THEN RUN SYS$TEST:HTHREADS.EXE
PARAM USE ACTIVE
PARAM SHOW /ALL
PARAM SHOW /SPECIAL
PARAM USE CURRENT
PARAM SHOW /ALL
PARAM SHOW /SPECIAL
EXIT
$! then zip up and send file COLLECTED_DATA_'DATE'.TXT as an e-mail attachment
```

Assessing Cluster Interconnect Health using LANCP

- In LANCP> SHOW DEVICE /CHARACTERISTICS output, look for:
  - `No  Full duplex operational`
  - `10  Line speed (mbps)`
- In LANCP> SHOW DEVICE /COUNTERS output, look for:
  - <Non-zero> `Unavailable station buffers`
  - <Non-zero> `Unavailable user buffers`
  - <Non-zero> `Frame check errors`
- In LANCP> SHOW DEVICE /INTERNAL_COUNTERS output, look for:
  - <Non-Zero> `Duplex mode mismatches`
  - Under `--- Driver Messages ---` you might see:

    `Possible duplex mode mismatch condition detected`
    - Caused by multiple Frame Check errors within 2 seconds
    - Also generates OPCOM message:

      %%%%%%%%%%% OPCOM 8-AUG-2017 16:10:04.69 %%%%%%%%%%% (from node NODE5 at 8-AUG-2017 16:10:12.89)
      Message from user SYSTEM on NODE5
      LANACP LAN Services
      %EIF0, Possible duplex mode mismatch condition detected

Assessing Cluster Interconnect Health using SCACP

– In SCACP> SHOW VC /ALL output, check:
  – Low **`Xmt:TMO`** values, particularly less than 1000, which is technically unsupported according to the OpenVMS Cluster Software SPD, which says: “The average packet-retransmit timeout ratio for OpenVMS Cluster traffic on the LAN from any system to another must be less than 1 timeout in 1000 transmissions.”
  – **`Cur XmtWindow`** values less than **`Max`**
  – Compare the **`Receive Duplicates`** counts with the **`Transmit Retransmits`** counts *in the opposite direction* to see if most retransmitted packets are due to unusual packet delays rather than actual packet loss
  – Look at the ratio of **`Transmit Messages`** to **`Transmit WinFull`** events to see how often a Window Full event occurs
– In SCACP> SHOW CHANNEL /ALL output, check:
  – Which paths are in the ECS – all the ones you expect? Any surprises?
    – If not, look at reason(s) why an expected path is not in the ECS – Slow latency? Lossy (high packet loss rate)? Inferior maximum packet payload size or priority?
  – Low **`Xmt:Rexmit`** ratios, particularly less than 1,000, again technically unsupported
  – Look at **`Delay (uSec)`** to see if the round-trip time seems reasonable
    – Minimum PEDRIVER reports is 250 uSec, so for greater accuracy / range, use LOCKTIME.COM tool
– See OpenVMS Technical Journal V17 article “A Guide to understanding the SCACP counters” for details

Assessing Cluster Interconnect Health using SCACP

Case study 1

- Customer problem: Entire 4-node OpenVMS Cluster suddenly became very poor in performance. Shut down 2 nodes; still bad.
- SCACP symptoms:
  - Very low Xmt:Rexmit ratios on some paths:

Xmt:Rexmit	NODE1_EIA	NODE1_EID	NODE1_EIE	NODE2_EIA	NODE2_EID	NODE2_EIE	NODE3_EIA	NODE3_EID	NODE3_EIE	NODE4_EID	NODE4_EIE
NODE1_EIA			70688	51905		75719	88913		65774		114350
NODE1_EID					426			544		626	
NODE1_EIE	70689			91210		69549	110066		132829		76041
NODE2_EIA	89827		82179			70800	78091		83758		61570
NODE2_EID		420						587		808	
NODE2_EIE	49912		48468	70801			126279		122007		69871

  - Very low Current Transmit window size:

```
SCACP> show vc/all

NODE1 PEA0 VC Summary 15-SEP-2017 09:57:00.74:

  Remote   VC     Total            Channels  ECS MaxPkt    ReXmt    --XmtWindow--
  Node     State Errors  Xmt:TMO  Open ECS  Pri  Size   TMO(uSec) Cur  Max  Mgt
  ------   ----- ------ ---------  ---- ---  ---  ----  --------- ---- ---- ----
   NODE4   Clsd  116236      2142    0   0 -126  1426  3000000.0    1  128    0
   NODE3   Clsd  103943      2626    0   0 -126  1426  3000000.0    1  128    0
   NODE2   Open  145364      2055    5   5    0  1426   220152.2    2  128    0
```

  - Workaround: **SCACP> SET LAN_DEVICE EID /PRIORITY=-10**

Assessing Cluster Interconnect Health using SCACP

Case study 2

– Customer problem: 5-node cluster including quorum node. Performance is sometimes slow. Processes often show up in RWSCS state.
  – LOCKTIME.COM showed reasonable round-trip times considering inter-site distance. Checking results from previous years showed round-trip time was no worse than last year; maybe slightly better.
  – In SCACP> SHOW CHANNEL output, Xmt:Rexmit ratios were sometimes less than ideal, and PEDRIVER Current transmit window sizes are less than Maximum.
  – In SCACP> SHOW VC /COUNTERS output, counts of Retransmits in one direction came close to matching the Duplicates Received in the opposite direction:

%Retransmits=Duplicates	NODE01	NODE02	NODE03	NODE04	NODEQ
NODE01		99.7%	99.5%	97.7%	100.0%
NODE02	99.3%		100.0%	99.4%	
NODE03	99.7%	99.7%		100.0%	
NODE04	99.9%	95.1%	99.6%		100.0%
NODEQ	100.0%	100.0%	100.0%	100.0%	

OpenVMS Cluster Performance in Less-than-ideal Networks

- PEDRIVER keeps track of how much time it should take to acknowledge a packet
- If an acknowledgement hasn't been received within what PEDRIVER considers a "reasonable" amount of time, PEDRIVER will assume it has been lost, and will proactively retransmit it, but:
  - When a retransmission occurs, PEDRIVER cuts the Current transmit window size to ½ of the Maximum value, and
  - If multiple retransmissions occur within a short period of time, PEDRIVER cuts the Current transmit window size all the way back to 1
  - Transmit window size grows (slowly) back up at a rate proportional to the rate of successfully-acknowledged packets
  - If we need to transmit without a free transmit window slot, we must wait -- counted as a Window Full (WinFull) event
- Some networks introduce a lot of jitter in packet delay, causing PEDRIVER to retransmit (and throttle back)
- Symptom: If this is happening a lot, in SCACP> SHOW VC /COUNTERS output you'll find the Retransmits counts and the Duplicates counts in the opposite direction are roughly equal numbers
- SYSGEN parameter PE2 allows you to tell PEDRIVER to be more patient before retransmitting
  - Units are number of 10-millisecond clock ticks, so value of 1 is 10 milliseconds extra; 2 is 20 milliseconds extra, etc.
- If most of the retransmitted packets are really getting lost (not just delayed), then raising PE2 would tend to hurt rather than help performance, because it would delay the retransmitting of lost packets

OpenVMS Cluster Performance in Less-than-ideal Networks

PE2 Parameter

– Units for PE2 are in 10-millisecond clock ticks:
  – Value of 1 means 10 milliseconds more patience
  – Value of 2 means 20 milliseconds more patience, etc.
  – In selecting an appropriate value, it may be helpful to look at SCACP> SHOW VC /ALL output and view the ReXmt TMO (uSec) and the VC Round Trip Time and VC Round Trip Deviation numbers:

```
NODE4 PEA0 VC Summary 19-SEP-2017 16:19:10.85:

  Remote  VC    Total           Channels  ECS MaxPkt   ReXmt    --XmtWindow--    Xmt    Total    ----------- Most Recent -----------
  Node    State Errors  Xmt:TMO  Open ECS  Pri  Size  TMO(uSec) Cur  Max  Mgt  Options Pkts(S+R)   VC Opened Time     VC Closed Time
  ------  ----- ------ ---------  ---- ---  ---  ----  --------- ---- ---- ---- ------  --------- ----------------- -----------------
     VC4  Open       1   1289403    2   2    0  1426   545135.7  128  128    0            2581688 17-AUG 14:49:50.91  (No time)
     VC5  Open       1   1503975    2   2    0  1426   616598.0  128  128    0            3073025 17-AUG 14:49:50.91  (No time)
...
NODE4 PEA0 VC Equivalent Channel Set (ECS) Membership Criteria 19-SEP-2017 16:19:10.85:

  Remote Number Number  Buffer Size  ECS   ECS   Load   Current  - ECS Speed Thresholds(uS) -   #LAN Devices VC Round Trip
  Node   Epochs NewECS   VC   ECS    Pri.  Hops  Class  MinDly  Faster  Promote Demote    Mgt   Loc   Rem    Time    Deviation
  ------ ------ ------  ----- -----  ----  ----  -----  ------- ------- ------- -------- ------- ----- -----  -------- --------
   NODE4      2      2   1426  1426     0     2   2000    250.0     0.0  1093.6   1531.2     0.0     2     2   55504.5  61203.9
   NODE5      2      2   1426  1426     0     2   2000    250.0     0.0  1093.6   1531.2     0.0     2     2   32972.4  72953.2
```

– PE2 parameter is dynamic, but you may have to stop and restart the LAN_DEVICE for the new value to take effect (or you could reboot)

Jumbo Frames

– Jumbo Frames allow larger packets than the standard Ethernet packets of roughly 1500 bytes

– Only available with 1- or 10-Gigabit Ethernet (or FDDI), not with 100-megabit Fast Ethernet

– Network equipment must support Jumbo Frames for the cluster to be able to use them

– Must enable Jumbo Frames via:

  – `LANCP> SET DEVICE/JUMBO` *`device`*

  – or by setting LAN_FLAGS bit 6 (hex 40 bit value)

– PEDRIVER probes for and will use Jumbo Frames automatically when available

– Transmitting the same amount of data in fewer packets lowers interrupt-state overhead

– Jumbo Frames primarily help performance of:

  – Lock Tree Remastering

  – MSCP Serving

– Lock requests are small packets and can’t benefit from Jumbo Frames

IPCI: TCP/IP Software vs. PEDRIVER: Who is smarter?

- With IPCI, each NIC has a different unique IP address
- PEDRIVER tracks paths on a per-endpoint (i.e. per-IP address) basis
- When PEDRIVER passed a UDP packet to the TCP/IP software for transmit, TCP/IP software sometimes thought it knew better and decided to send it out a different NIC than PEDRIVER intended, making it difficult for PEDRIVER to keep accurate path statistics.
  - TCP/IP software might even send a packet out through a NIC which has not been configured for IPCI or has been disabled for SCS traffic via:

    `SCACP> STOP IP_INTERFACE xxn`
- Fixed in VMS84I_DRIVER-V0400 plus TCP/IP HPE-I64VMS-NET_PAT-V0507-13ECO5-4 patch kit
- Must set SYSGEN parameter PE3 bit 5 (value of 32) to enable this fix

Assessing Cluster Interconnect Health using SCACP

Case study

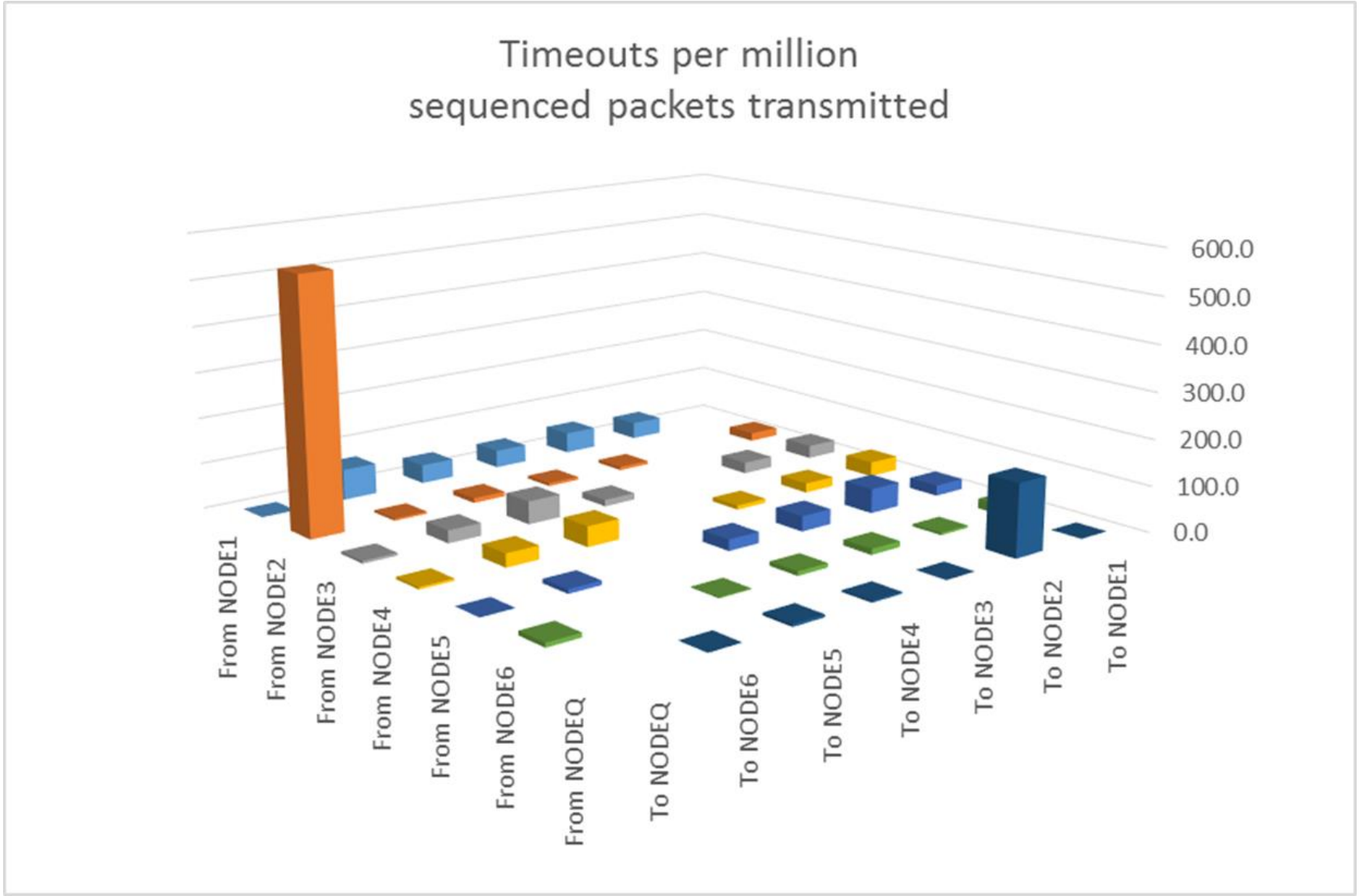
- Customer problem: Poor cluster performance in 3-site OpenVMS cluster. New i4 servers, and all-new network equipment.
- Symptoms:
  - High retransmit rates due to high packet loss rates.
  - ECS data showed both LANCI and IPCI paths were available between all nodes at all sites.
- Talked with the network administrator on a conference call. I noted that some of the LANCI paths between sites had unusually high packet loss rates. He said there was no bridging between sites, so there could not possibly be any such paths.
- Turned out there was an old bridged link between sites which was somehow being used. It was 10 megabits, half duplex.
- We used SCACP> SET CHANNEL /PRIORITY to -

Tools to aid in Cluster Interconnect Health Analysis

- These DCL command procedures take output from SCACP> SHOW VC /ALL and SHOW CHANNEL /ALL in a cluster and create .CSV files containing cluster-wide pictures of crucial metrics.
- Tools have names of SHOW_*_CSV.COM and are located at http://encompasserve.org/~parris/
- SCACP metrics available so far:
  - Xmt:TMO ratio
  - ECS membership
  - Xmt:Rexmit ratio
  - Delay (round-trip latency time)
  - Duplicates, WinFulls
  - Transmit Window (Current, Maximum, Current as a percentage of Maximum)

Tools to aid in Cluster Interconnect Health Analysis

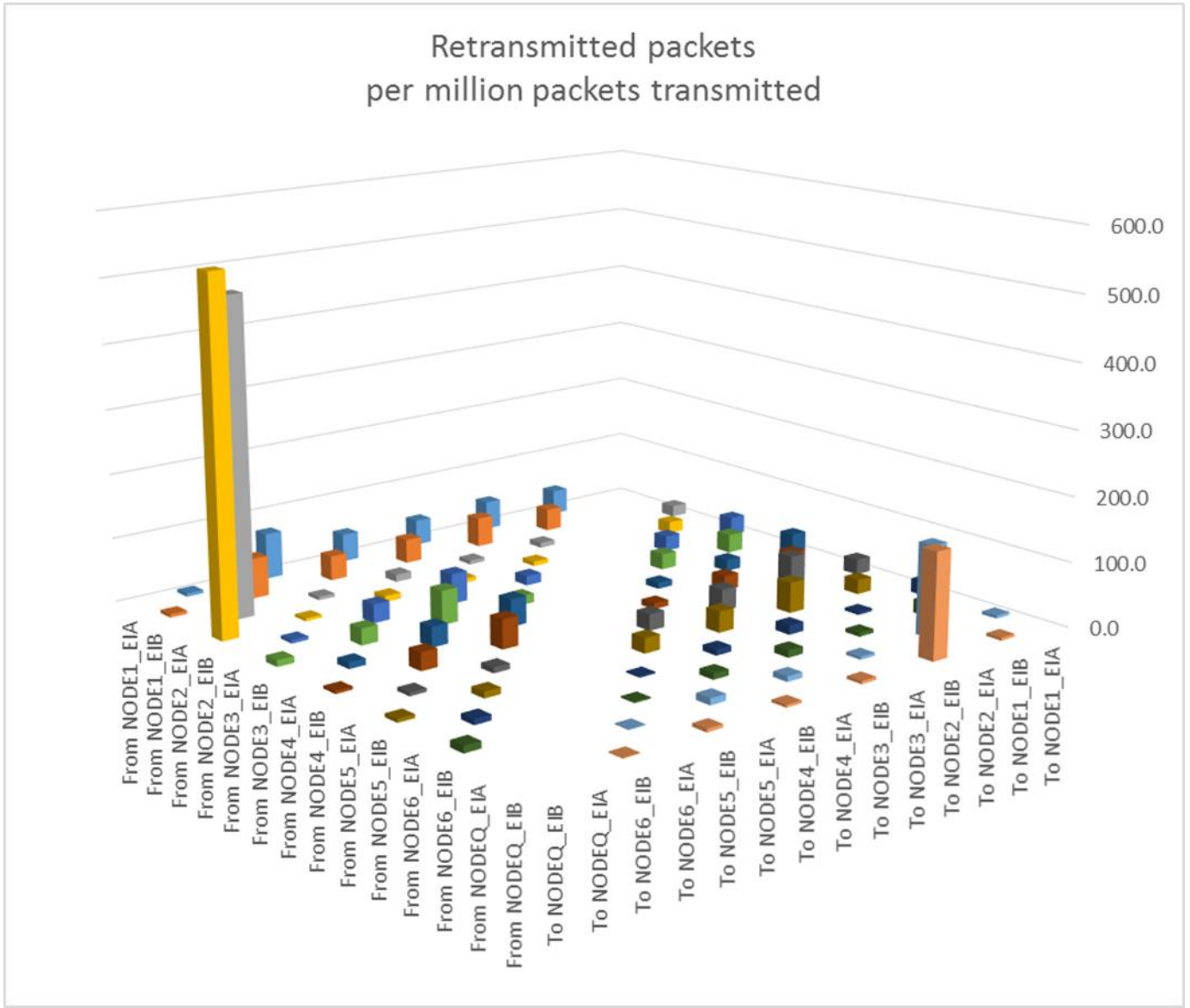
`Xmt:TMO` metric



HPE POINTNEXT

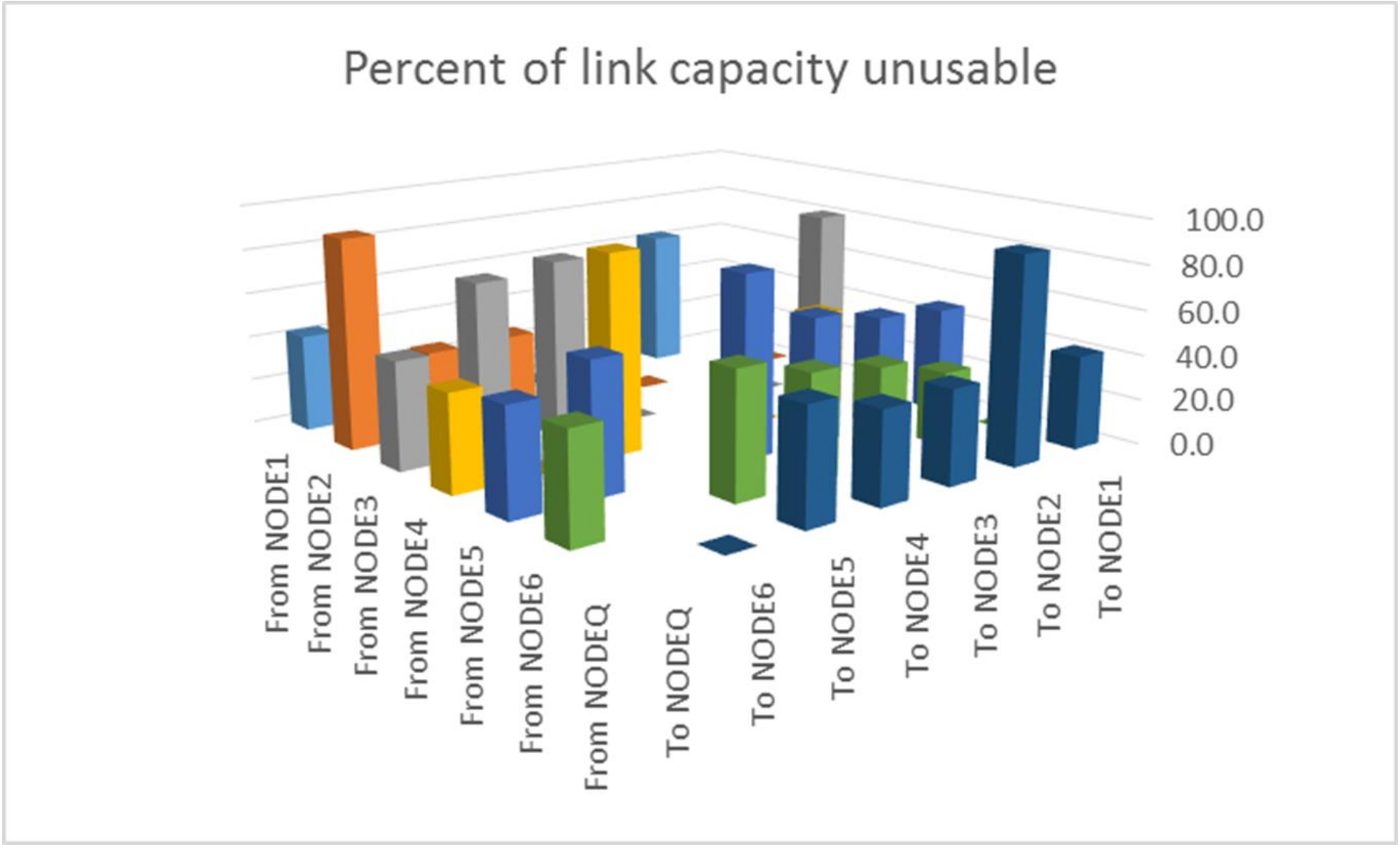
Tools to aid in Cluster Interconnect Health Analysis

`Xmt:Rexmit` metric



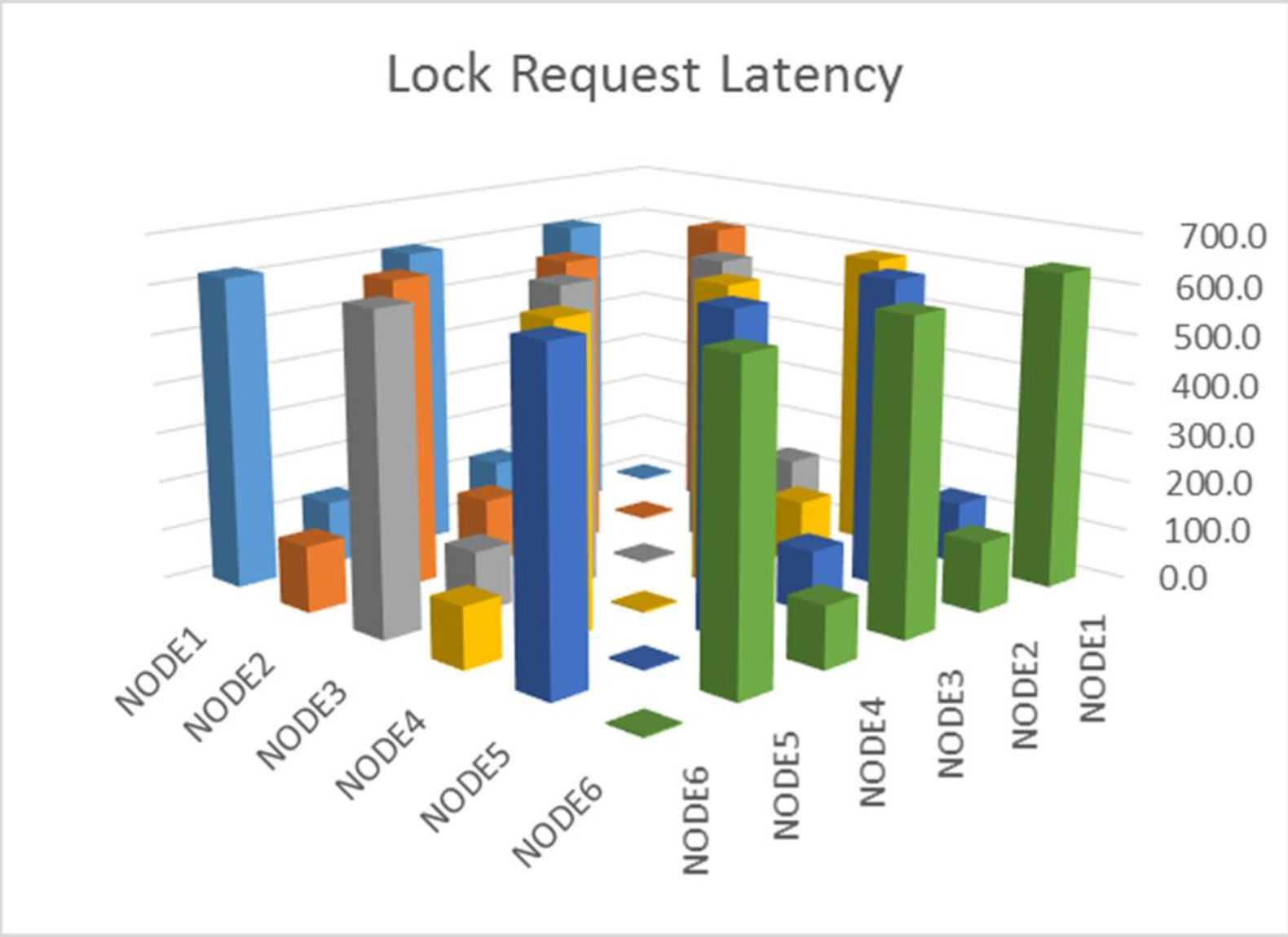
Tools to aid in Cluster Interconnect Health Analysis

`XmtWindow Cur` as a percentage of `XmtWindow Max`



Tools to aid in Cluster Interconnect Health Analysis

Lock Request Latency as measured by LOCKTIME.COM tool



HPE POINTNEXT

Questions?

HPE
POINTNEXT

Thank you

keith.parris@hpe.com

Hewlett Packard
Enterprise

OpenVMS storage layout

Example disk layout

- Maximum of six arrays and three sites
- Three environments (Production, Test, Development)
- Systems boot from fibrechannel
- All fibrechannel disks shadowed:
  - System disks
  - Common disk
  - Data disks
- Array based copies for backup (snaps, clones)
- All local disks (partitioned RAID) used by local node only
  - Page/swap/dump/T4/”DVD” disks
  - Local “full” boot for system maintenance

Example disk naming – DSA and FC disks

- DSA10 (1DGA1010, 2010, 1110, 2110) – common disk

- DSA11 - system disk, site A
- DSA12 - alternate system disk, site A
- DSA13 - system disk, site B
- DSA14 - alternate system disk, site B
- DSA15 - system disk, site C
- DSA16 - alternate system disk, site C

- DSA21 … DSA39 – data (small shadow sets)
- etc.

Example disk naming – local disks

- 8 slot SAS array, RAID 6, 2x hot spares, BBWBC:
  - DKA0 - page/swap/dump files (non-shadowed)
  - DKA1 - T4 data
  - DKA2 - House keeping data
  - DKA3 - Staging area for backups
  - DKA4 - local boot (non-clustered, full system)
  - DKA5 - copy of OpenVMS DVD media + kits etc.

Shadowing

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

Array configuration

- Use RAID 0+1 (EVA vRAID1) for best performance
- Use double sparing, single disk 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

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 disk READ ONLY
- Delete root <SYS0> to avoid unexpected booting with unconfigured hardware

HPE POINTNEXT

Volume Shadowing Best Practices

OpenVMS Boot Camp 2017, SID 304

Keith Parris, Engineer

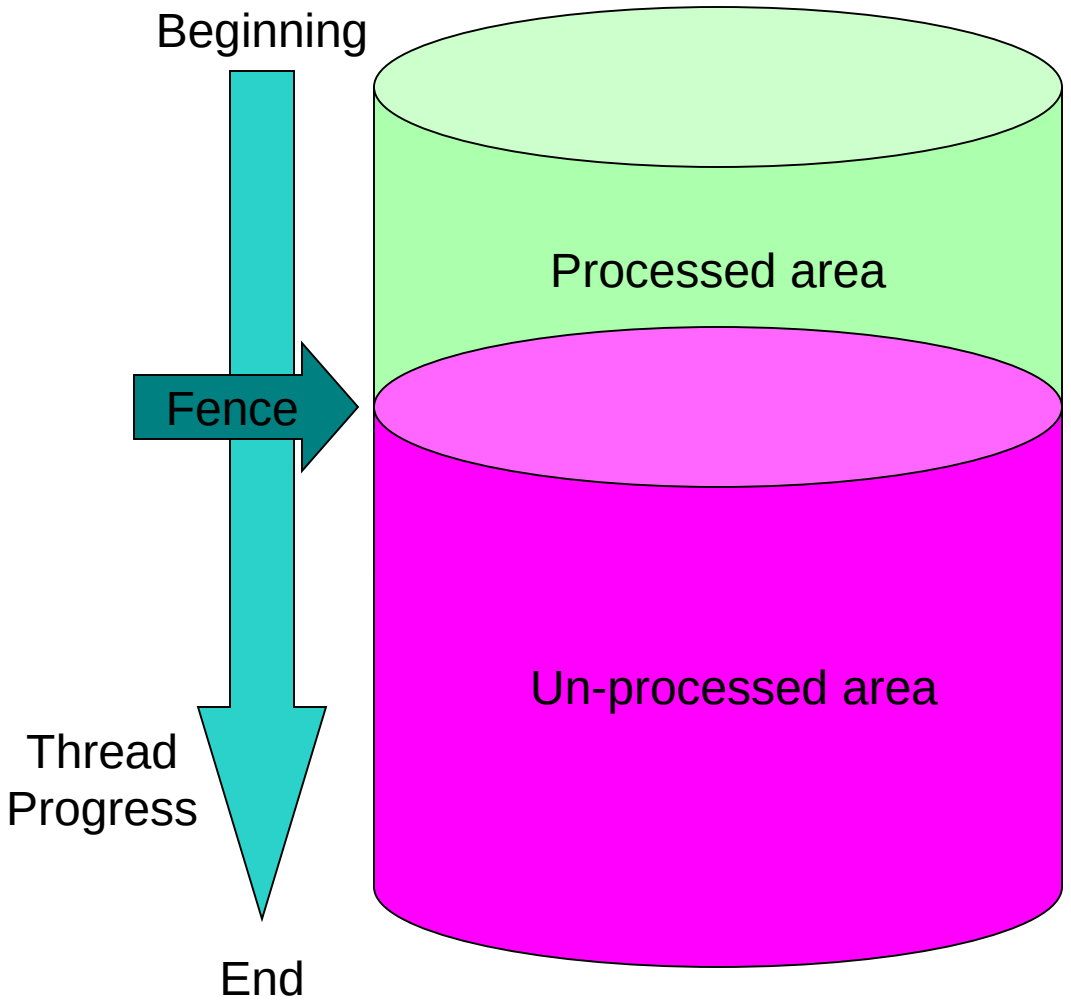
Hewlett Packard Enterprise

Shadowing Full-Copy and Full-Merge Algorithms

– Start at first Logical Block on disk (LBN zero)

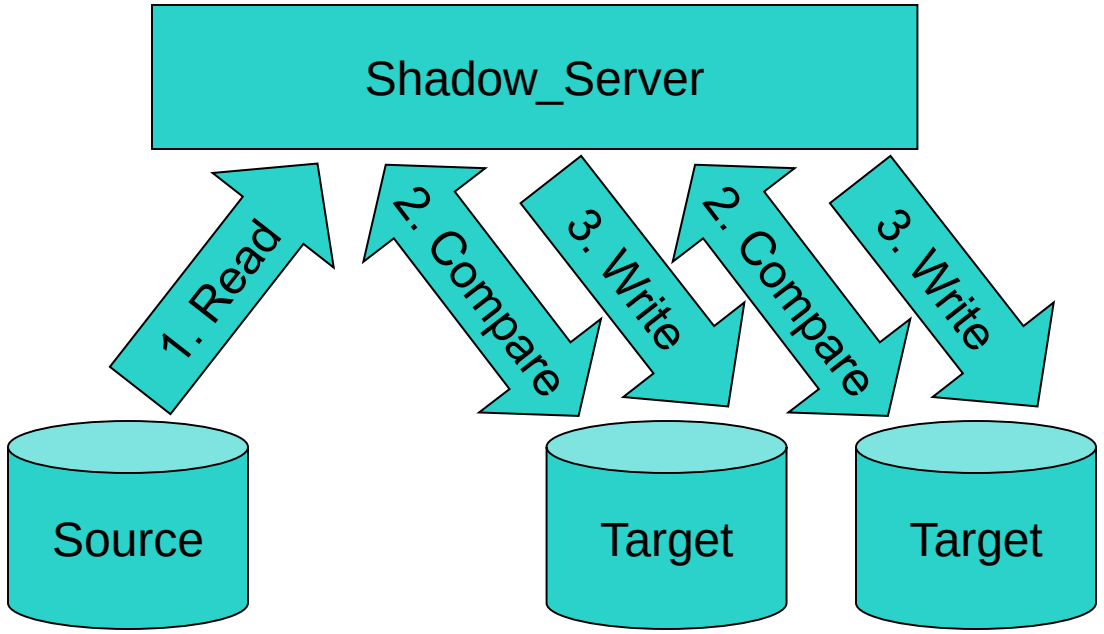
– Process 127 blocks at a time from beginning to end

– Symbolic “Fence” separates processed area from un-processed area



Shadowing Full-Copy Algorithm

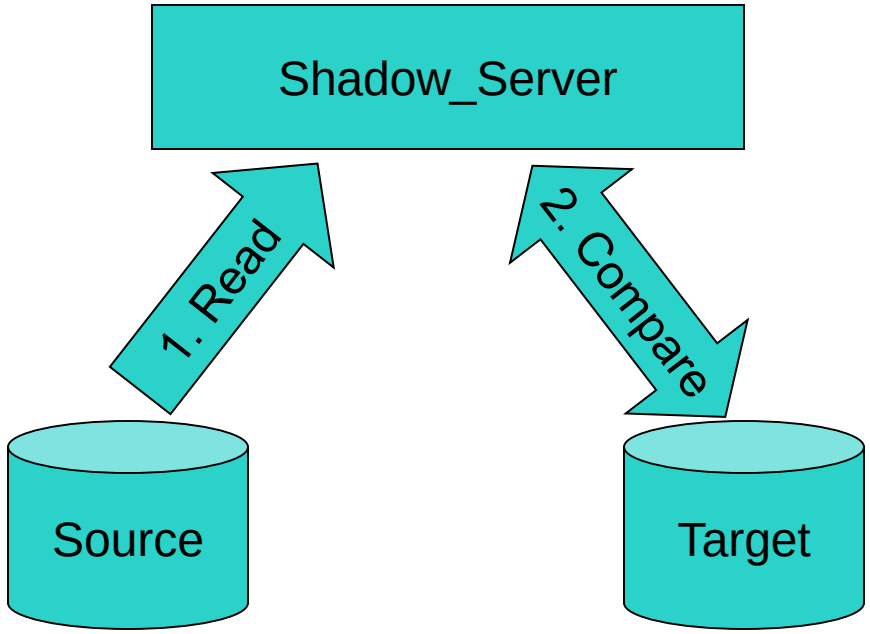
1. Read from (a) source member
2. Compare with target member(s)
3. If different, write data to target and start over at Step 1.



Shadowing Full-Copy Algorithm

Data identical

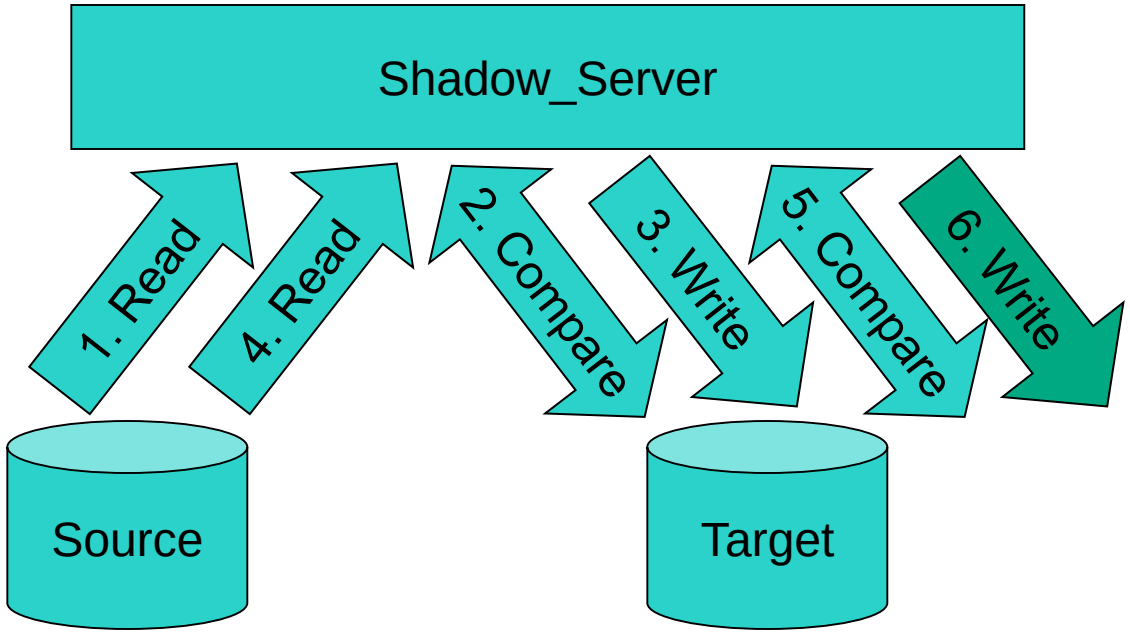
1. Read from source
2. Compare with target (matches, so done)



Shadowing Full-Copy Algorithm

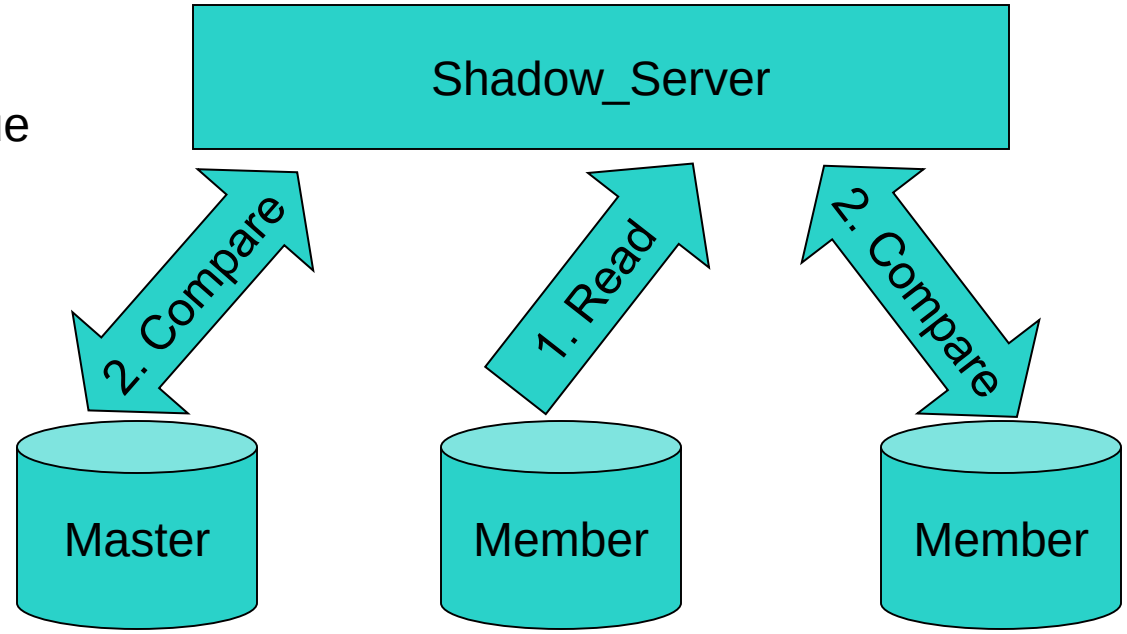
Data different

1. Read from source
2. Compare with target (difference found)
3. Write to target
4. Read from source
5. Compare with target (difference seldom found). If different, write to target and start over at Step 4.



Shadowing Full-Merge Algorithm

1. Read from any member
2. Compare with other member(s)
3. If different, do a Fix-Up: halt all I/Os to the shadowset, fix up differences using data from the Master member, then allow I/Os to continue



General Volume Shadowing Best Practices for Performance

- Because shadow copy/merge threads operate from the start to the end of the disk in 127-block increments:
  - If you want to protect a given amount of data using Shadowing, the shadow copies will tend to complete faster if you have more of smaller volumes than if you have fewer, larger volumes.
  - Data is not redundant until all shadow copies complete, so avoid a single or a few huge volumes and the test smaller
- Because copies and merges are faster when the data is identical:
  - If you have a new disk to add to a shadowset, it's actually faster in elapsed time to do a $ BACKUP /PHYSICAL from the DSAnnn: device to the target disk (mounted /FOREIGN) and then add it to the shadowset afterward, than to simply add it to the shadowset while it still contains mostly-different data

Write Bitmaps

– Mini-Merges and Mini-Copies take advantage of Write Bitmaps – data structures residing in memory
  – These bitmaps record writes to each 127-block increment of blocks on a disk
  – When a node leaves the cluster, it takes away with it any write bitmaps it had in its memory
  – If you need to use a Write Bitmap either for a Mini-Copy or a Mini-Merge, one of these needs to survive on a node which remains up in the cluster after a failure
  – In a multi-site disaster-tolerant cluster, you can lose an entire site, so you need Write Bitmaps to always survive on nodes at the surviving site

Write Bitmaps

- When Write Bitmaps are in use, before an actual write occurs to a shadowset, any associated Write Bitmap(s) are updated
  - This implies messages need to pass between nodes, with potential adverse performance impact
    - As a performance optimization, we keep a Local Bitmap for each remote Master Write Bitmap, and if we find a bit already sent, indicating we've already sent notification for a change to that 127-block segment, we don't need to send another notice
  - In 8.3-1H1 and earlier, these write bitmap updates occurred sequentially, one write bitmap at a time
  - In 8.4 these updates were done in parallel for increased performance, but obscure cluster hang problems caused this code to revert to the pre-8.4 code in post-8.4 patch kits VMS84A_SYS-V0600 and VMS84I_SYS-V0600. The release notes said:

    "An OpenVMS V8.4 cluster hang had been seen occasionally during system shutdown or boot while using host-based mini-merge (HBMM) for the system disk shadow set, shared by multiple systems. This was occurring during a bitmap update operation when a write bitmap message got lost and the device remained in a write locked state.

    In OpenVMS V8.3-1H1 and in previous versions, master bitmap update messages were sent one at a time, updating each remote master bitmap sequentially. In OpenVMS V8.4, these remote master bitmap update operations were made parallel. We have reverted back the code that performs parallel bitmap updates to sequential.

    This problem has been fixed."

Full Merges and Mini-Merges

Why do Full Merges hurt so much?

- When Full-Merges were first designed, developers thought the merge I/Os would be the primary bottleneck. In the real world, the bottleneck is reads ahead of the merge fence, where we must merge the area of the read before returning the data to the user, and do this for each and every I/O until the merge fence passes over the hot area. This is where the pain comes from.
- Conclusions:
  - Get Full Merges over as quickly as possible. Don't throttle back the merge thread – that just prolongs the pain.
  - Consider putting hot files at the beginning of the disk if possible, so the merge fence can get past them quicker.
- Set up a policy and use Mini-Merges instead of Full Merges whenever possible.

How many Mini-Merge Write Bitmaps should I have?

– Single cluster: Probably two. Probably not six.

– Two-site Cluster: Probably four (2 per site)

– Three-site Cluster with storage at all 3 sites: Probably six (2 per site)

Recently-added Volume Shadowing features

– Automatic Mini-Copy on Volume Processing (AMCVP) Feature:
  – MULTIUSE keyword in Mini-Merge policy: Automatically converts a Mini-Merge Write Bitmap to Multi-Use (both Mini-Merge and Mini-Copy) Write Bitmaps so the removed shadowset members can be re-added later with a Mini-Copy operation
– DISMOUNT keyword in Mini-Merge: Automatically convert the specified number of Mini-Merge Write Bitmaps to Multi-Use (both Mini-Merge and Mini-Copy) Write Bitmaps when a shadowset member is $DISMOUNTed, so it can be re-added later with a Mini-Copy operation

Site IDs and Read Costs

- In multi-site clusters, you can set different Site IDs for each site if you wish
  - Volume Shadowing will automatically select default Read Costs
    - MSCP Served disks: difference of 499 in Read Cost
    - Fibre Channel disk at different site: difference of 40 in Read Cost
  - Advantage: Biases read operations toward using disk at local site rather than remote disk, or both disks
  - Disadvantage: If you get a burst of reads from one local site, you won't start sending any reads to the opposite site until the queue length gets very big (40, or 499)

Checking integrity of shadowset data

- $ ANALYZE /DISK /SHADOW compares contents of member disks, and reports any differences
- If differences are found, next question is: Which member has the most valid data?
  - Tip: You can vary the Read Cost values between the members to select first one and then (each of) the other(s) for reads, and identify which is the best copy of the data, then remove the other(s) and start a Full Copy to fix the bad member(s).

Monitoring Shadowset Membership

– You don’t want a shadowset member to drop out unnoticed

– Console Management systems can scan for dismount messages for shadowset members

– DCL procedure can be used to track shadowset membership and send alert on any changes; see example SHADOW$TRACK.COM from http://encompasserve.org/~parris/

Questions?

HPE
POINTNEXT

Thank you

keith.parris@hpe.com

Hewlett Packard
Enterprise

OpenVMS cluster layout

OpenVMS Clusters
Design and Support

Nic Clews
Warrington

November 10th 2011

What is a cluster?

- Two or more computers cooperatively processing data
  - OpenVMS can be a cluster of one
- Different types
  - Shared nothing
  - Shared everything
  - Failover / high availability
  - Distributed processing / high performance
- Performance is not in the speed of the CPU
  - Data integrity is paramount to OpenVMS
- OpenVMS is still the best general purpose high availability – high performance clustering operating system supporting a useful number of cooperative systems
- Disaster proof, in 13 seconds...

The basics of a system in a cluster

- A connection to all participant systems
  - Rules governing the use of those connections
  - Direct connection, not “via” another node
  - Criteria for membership validity
- Agreed protocol
  - Avoid ambiguity
  - Provide flexibility
  - SCA System Communication Architecture, SCS System Communication Services (used interchangeably)
- Common cluster software environment (Single System Image)
  - Same expectations of all members of their behaviour
  - OpenVMS allows mixed version mixed architecture clustering

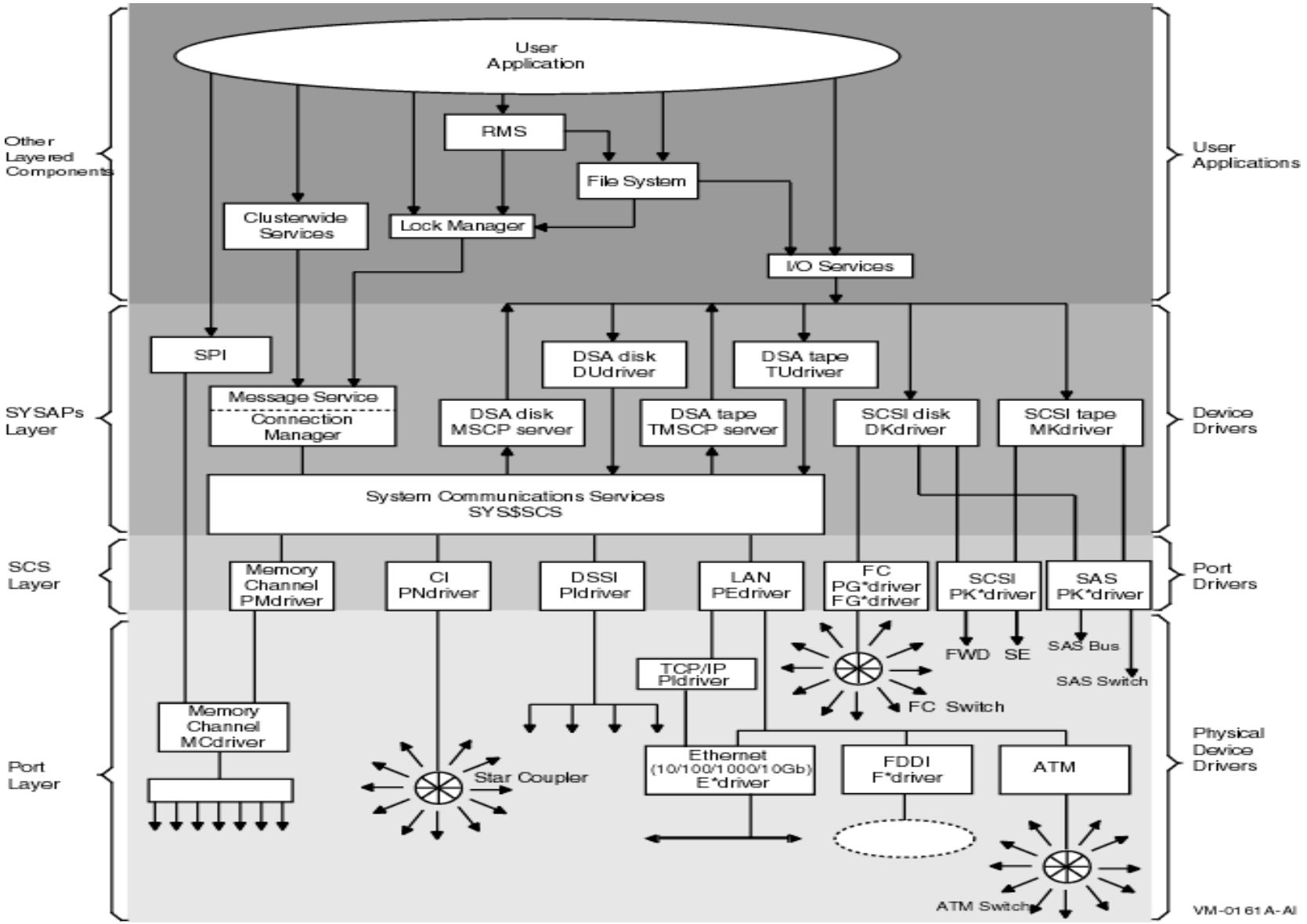
OpenVMS Clusters

- Often work when badly or incorrectly configured
  - "Oh its working. It must be right!" (WRONG)
- Withstand a lot of abuse
  - Even from very ingenious idiots
- Highly tolerant of stupidity and ignorance
  - The unwilling led by the unknowing into the wolf's lair
- If correctly designed (and tested)
  - Disaster proof!
  - Highly available
  - Flexible service
  - Happy users
  - Low stress factor

Inside the system

- Our view of the cluster needs to be from the system's point of view
- To understand system behaviour we need to ignore the cluster diagram, and think how the system would 'think' with a given set of conditions
- The outside world reality could be very different from the view this system has (creeping doom)
- Ambiguous naming, panic, poor documentation, late-night-early-morning call and a parallel set of events could be a recipe for catastrophic failure
- "Murphy's Law"
  - "If anything can go wrong, it will"
- If you pull a plug is MUST be the right one !
- Avoid Single Points of Failure

Cluster logical diagram (OpenVMS 8.4)



Requirements to make things easier

- Configuration planning
- Proper device naming
- Logical and sensible disk volume naming
- Proper SAN/Fibre configuration and naming conventions
- Proper network naming and configuration
- Proper network / cabling setup
- Correct installation (joint / twin sourced power supplies)
- Failure planning / scenario
- Testing pre production
- Testing post production – testing schedule and ongoing
- Management tools (Availability Manager etc.)
- Common sense !

Common characteristics

- Point to point between all participating (clustered) systems
  - A direct path not via another system
  - System “must see all” to make decisions
- Periodic heartbeat
  - Detect path loss (TIMVCFAIL)
- A break in the path can look like the loss of a system
  - This is one of the main issues of membership to a cluster, did we lose the path or the system?
- Storage only, messaging only, storage and messaging
  - SCSI, FibreChannel: storage only (no messaging)
  - Network: Messaging only
  - DSSI (Digital specific): Storage and messaging (based on SCSI)

Multiple Interconnects

- Irrespective of the use of multiple media interconnects, one must be common between all (usually NI)
- The node will manage the “best” path, and choose based on approximate latency and speed
- NI with 10g and jumbo frames for transfer is just about the fastest
  - But a frame needs to be filled so using the standard Ethernet packet size is acceptable too
- Failure of one path will ‘fail over’ to another working path
- In all cases they are known as a VC – Virtual Circuit via a “port”
- Parameters
  - TIMVCFAIL
  - RECNXINTERVAL

CLUSTER_AUTHORIZE.DAT

- Contains the cluster number
  - Used to separate clusters on the same LAN
  - Used to determine the broadcast (LAN and IPCI)
  - One way password encryption
  - Copy the file or use SYSMAN to reset information
- Errors against PEA0 can indicate a matching cluster number with different password
- Delete this file on permanently removed members
- Potential security risk
  - Allows shared access to resources even if AUTHORIZATION file not matched!

Cluster Reconfiguration

- Timers (OpenVMS)
  - TIMVCFAIL (1600 = 16 seconds)
    - The time in milliseconds that determines a connection is broken
  - RECNXINTERVAL (20 = 20 seconds)
    - Time (seconds) allowed to “repair” the connection before the member is lost
    - May not be the same connection if an alternative VC path becomes active
    - Processes stall if QUORUM lost, disks go MOUNT VERIFY
    - Dependent transactions will wait until decision taken.
- If an interconnect changes state, reconfiguration is triggered
  - A system may be coming (back) online
  - A system may have failed
  - Interconnect repaired
  - Interconnect breaks

Cluster reconfiguration (quorum lost and regained)

- All processes stall (QUORUM capability bit on CPU)
- Disks go mount verification
- Continue if QUORUM for the cluster not lost
  - Rebuild lock database
  - Resume processing
- If QUORUM is lost, await further events...
  - But nothing is corrupted
- Until QUORUM regained
  - Rebuild lock database
  - Resume processing
- Excluded member(s) must rejoin as new members (reboot) so will voluntarily crash (CLUEXIT) if the VC to the cluster is repaired
  - A satellite will not know if the whole cluster has survived or not

Every system works out a new subcluster

- Connection Manager
- Each system works out a number of sub clusters
  - Best to worst
- Surviving, communicating systems exchange their best scenario
- When agreement is reached, the reconfiguration completes
  - excluded members may not rejoin, they voluntarily CLUEXIT or may have gone anyway
  - The new configuration may or may not have QUORUM!
    - Theoretically has more votes than other subformations
- Process is synchronized by a transaction co-ordinator
  - Highest of software version and SCSSYSTEMID of all present members
- A satellite will not know if the whole cluster has survived or not

Some warnings

- REXNCINTERVAL, TIMVCFAIL, QDSKINTERVAL must all be the same throughout the cluster!
- Why?
- Lack of synchronization
- Proposing reconfiguration...
- Proposing reconfiguration...
- Proposing reconfiguration...
- ...
- (crash and boot)

Quorum and voting

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

HPE
POINTNEXT

Best Practices for Multi-site and Disaster-Tolerant OpenVMS Clusters

OpenVMS Boot Camp 2017, SID 303

Keith Parris, Engineer

Hewlett Packard
Enterprise

What things can I do to maximize performance in a multi-site cluster?

What things can I do to maximize performance in a multi-site cluster?

- The distance between sites adversely affects performance primarily in two areas:
  - Remote disk writes to keep cross-site shadowsets in synchronization
  - Remote lock operations
    - Including remote directory lookups
- Estimating inter-site latency:
  - See Excel spreadsheet to calculate latency due to speed of light through fiber optics at https://sites.google.com/site/keithparris/Latency_due_to_Speed_of_Light.xls
  - Rule of thumb: 1 millisecond round-trip for each 100 kilometers (roughly 62 miles)
  - Actual circuit path length may be (significantly) longer than physical inter-site distance
  - LOCKTIME.COM tool from http://encompasserve.org/~parris/ may be used to measure actual round-trip times in an existing cluster

What things can I do to maximize performance in a multi-site cluster?

- Speeding Remote Shadowset Writes:
  - If access to the disks at the remote site is via MSCP Serving or most Fibre Channel SAN Extension methods, remote writes will take **two round trips** between sites to complete
  - Remote Writes can be done in only **one round trip** between sites with either:
    - Cisco I/O Acceleration, or
    - Brocade Fast Write
  - Reads typically take **one round trip**

What things can I do to maximize performance in a multi-site cluster?

- Ensure inter-site network is:
  - Clean (no packet loss)
  - Lowest latency possible, given the inter-site distance
    - This may mean avoiding LAN-within-IP encapsulation methods or MPLS which tend to add latency
  - Low as possible in jitter of packet latency
  - Sufficient in bandwidth to perform a Volume Shadowing Full-Copy operation to restore redundancy of all the cross-site shadowsets in a “reasonable” amount of time (say, overnight)
- With longer inter-site distances, the SYSGEN parameter CLUSTER_CREDITS may need to be raised to avoid SCS credit waits on the VMS$VAXcluster SYSAP connection
- If MSCP Serving is in place, the MSCP_CREDITS and/or MSCP_BUFFER parameters may need to be raised on the VMS$DISK_CL_DRVR → MSCP$DISK SYSAP connection
- Performance of MSCP Serving and Lock Tree Remastering can be better if the network supports Jumbo Packets

What things can I do to maximize performance in a multi-site cluster?

– With a high-quality inter-site network, PEDRIVER transmit and receive window sizes may need to be increased for greater throughput with longer inter-site distances
  – Use SCACP> CALCULATE WINDOW_SIZE /SPEED=*megabits* /DISTANCE=*units=distance*

```
SCACP> help calculate example

CALCULATE

  Example


     SCACP> CALCULATE WINDOW_SIZE /SPEED=1000/DISTANCE=KILOMETERS=500

         The command in this example calculates the window size to be
         used between two nodes that are 500 kilometers apart, connected
         by a 1 Gigabit/Second line speed. The command produces output
         similar to the following:

            Calculate Window Size  2-JUN-2006 17:49:18.41:
                   Inter-node link DISTANCE:          500 KILOMETERS
                   Inter-node link SPEED:            1000 Mb/s
                   ------------------------        -----------------
                   Recommended WINDOW SIZE:          1024 frames
```

How many sites can I have in a single cluster?

How many sites can I have in a single cluster?

- The OpenVMS Cluster design limit for number of nodes in a cluster is 256
  - So with one node per site, that implies an absolute limit of 256 sites
- The OpenVMS Cluster Software SPD indicates support for a maximum of 96 nodes
- Largest known OpenVMS cluster was 151 nodes
- Host-Based Volume Shadowing supports a maximum of 6 shadowset members, so if Volume Shadowing is used to keep data replicated between sites, that implies a maximum of 6 sites with storage, and maybe a quorum node at a 7th site

How far apart can my sites be?

“The maximum system separation is 150 miles. With proper consulting support via HP Services Disaster Tolerant Consulting Services, the maximum system separation is 500 miles.”

-- OpenVMS Cluster Software Product Description
http://h41379.www4.hpe.com/doc/spdclusters.pdf

How far apart can my sites be?

- OpenVMS Clusters have no inherent design limit on distance
- Host-Based Volume Shadowing has been tested out to a simulated distance of over 60,000 miles successfully
- During development of IP as a Cluster Interconnect (IPCI), OpenVMS Engineering successfully formed and tested a cluster with nodes on several continents simultaneously
- So why the limit in the SPD?
  - Concerns about application performance, because of the latency due to the speed of light between sites
- Increased distance primarily affects performance in two areas:
  - Remote disk writes to keep shadowsets in synchronization
  - Remote lock operations (including directory lookups)

How far apart can my sites be?

– We advise that before the datacenter sites are chosen, you first test application performance at the proposed inter-site distance using a network emulator to introduce network latency to simulate the proposed inter-site circuit path length (distance)

– We know of a number of customer sites running OpenVMS Clusters with circuit path lengths in the 600-900 mile range and one with 3,000 miles between sites, but “your mileage may vary” depending on your applications. If you need an official support statement, send e-mail to OpenVMS.Programs@hpe.com

How far apart can my sites be?

Case Study

- Customer question: Can I spread my OpenVMS cluster across a 1,000 mile distance for disaster tolerance?
- Investigation: 1,000 miles between datacenter sites implies a 16 millisecond round-trip time. With MSCP Serving for access to remote disks, each disk write would take 2 x 16 = 32 milliseconds.
- We asked the customer questions about their application performance and expectations:
  - Questions: How many disk writes are needed for your customer transactions, and what response time expectations do your users have for these transactions?
  - Answers: Transaction requires 4 disk writes, and the response time requirement is 1 second.
  - Analysis: 4 x 32 milliseconds = 128 milliseconds, or about 1/8 second, so this would appear to easily meet the 1 second maximum response time requirement.
- This gave us a rough indication that the application would likely still work acceptably in a 1,000-mile cluster. Testing of the actual application performance with latency simulated via a network emulator box would still be advisable, to confirm this initial prediction.

How should I choose nodenames?

How should I choose nodenames?

- Avoid picking names descriptive of the hardware, platform, architecture or model
  - Nodenames tend to stay forever, but may be applied to different hardware boxes over time. Many customers still have nodenames like VAX01 yet are Integrity Server boxes now.
- Avoid choosing nodenames based on the company name, because of buyouts, mergers, name changes, etc.
- Choose a scheme that allows easily identifying the datacenter site based on the nodename
- Choose a scheme that is scalable, in terms of:
  - Number of nodes
  - Number of sites
- Examples:
  - Fixed prefix, and range of node numbers indicates site
    - e.g. XYZ1nn for nodes at Site 1; XYZ2nn for nodes at Site 2; XY3nn for nodes at Site 3
  - Prefix indicates site, suffix differentiates nodes within that site
    - e.g. DEN001, DEN002 in Denver, ATL001, ATL002 in Atlanta

How should I choose nodenames?

– Case studies:
  – Poor foresight: First two nodes at the first site were Node 1 and Node 2. Second site had Node 3 and Node 4. Then a node was added to each site, becoming Node 5 at the first site and Node 6 at the second site. Very confusing.
  – Not scalable: Even-numbered nodes are at one site; odd-numbered nodes are at opposite site: Problem: How do you name a quorum node at a 3rd site? How do you expand from two sites to 3 sites if you need to?

How should I number my disk devices?

How should I number my disk devices?

- Choose disk unit numbers so the site at which the disk is located can be easily identified from the unit number
  - e.g. 1DGA1xxx disks are at Site 1; 1DGA2xxx are at Site 2, etc.
- Leave room for expansion
- Number shadowsets logically as well
  - e.g. DSA0 through DSA999 for cross-site shadowsets with members located at multiple sites
  - e.g. DSA1xxx for shadowsets located only at Site 1, such as a local system disk shadowset for the site
- Choose disk unit numbers and shadowset device names so it is also easy to identify which shadowset the disk is a member of
  - e.g. Shadowset DSA14 has members 1DGA1141 and 1DGA1142 at Site 1 and 1DGA2141 and 1DGA2142 at Site 2 and 1DGA3141 and 1DGA3142 at Site 3

How should I choose allocation classes?

How should I choose allocation classes?

– It is no longer necessary for the node doing MSCP Serving to have the same allocation class as the served disk(s).

– If you have server nodes with similar hardware, simpily pick a unique allocation class per node. This prevents device name conflicts for devices like DVDs as DQA0 or such.

When should I use IPCI (IP as a Cluster Interconnect)?

When should I use IPCI (IP as a Cluster Interconnect)?

- Primary reason:
  - When the network cannot provide at least the illusion of a bridged LAN between sites
- Secondary reasons:
  - When the methods available to encapsulate LAN packets within IP to simulate bridging introduce so much latency, jitter and/or packet loss that using the IP network directly using IPCI will provide better performance and availability
  - As a backup path to provide higher availability in the event of a problem on the LAN, e.g. ride through Spanning Tree reconfigurations
- Why do you prefer LANCI over IPCI in general?
  - IPCI has slightly lower maximum packet payload size than LANCI
  - IPCI has slightly higher host CPU overhead than LANCI, since the code path includes TCP/IP Software as well as PEDRIVER
  - Because IP Multicast is seldom set up for IPCI at most customer sites, the Hello packets used for path discovery and path status must use Unicast rather than Multicast packets, adding a small amount of overhead to the hosts, which is greater as the number of nodes in the cluster goes up
- Market shift: When IPCI was invented, the threat was that all extended or bridged LANs would go away. But because Virtual Machines can now migrate between servers, transparent to the VM, and do so even between sites, for purposes of DR, the market has demanded the ability to support for the same IP segment at different sites. Network vendors have developed technology to meet this demand, one that OpenVMS Engineering had no way to predict. This same technology can support OpenVMS Clusters.

Should I use different Site Numbers in my multi-site OpenVMS cluster?

Should I use different Site Numbers in my multi-site OpenVMS cluster?

- Site Numbers are a shorthand way of setting Read Costs for shadowset members
- SHADOW_SITE_ID may be set to a positive, non-zero value to designate a node as being located at a given site
  - Alternatively, you could use $ SET DEVICE/SITE=n DSAnnn: for each and every shadowset to set the server Site ID, but this is an older method entailing a lot more work
- With different Site ID values, the read costs for member disks are set to default values:
  - Local DECram disk: 1
  - Local SAS or Fibre Channel magnetic disk: 2
  - Remote Fibre Channel Disk: 42 (difference of 40 above local disk)
  - MSCP-Served disk: 501 (difference of 499 above local disk)
- The decision of whether or not to use different Site numbers will then depend on whether these default Read Cost settings are acceptable, or whether it is considered worthwhile to override them all manually to still retain the Site number setting and yet have customized Read Cost values, set at system startup time via a set of $ SET DEVICE /READ_COST commands.

How should I set Read Costs for Host-Based Volume Shadowing?

How should I set Read Costs for Host-Based Volume Shadowing?

- Because Shadowing keeps members disks identical, a read operation could be satisfied from any one of the member disks
- Idea of differing Read Costs is to send reads from servers within a site to disks at the same site, to avoid the inter-site latency
- When a shadowset is read, Shadowing adds the local queue length for each member disk to the Read Cost and directs the read to the member with the lowest total. For disks with equal Read Costs, Shadowing sends reads to the member disks in “round-robin” order.
- Sometimes a node experiences a burst of reads.
  - With equal Read Cost values, the reads can be equally distributed across all the member disks for lower average response times.
  - With different Read Cost values, the reads will be distributed first to member(s) with lower Read Cost value(s), and only after the local queue length to those disks rises, to members with higher Read Cost values.
  - With small differences read costs, when the local queue depth reaches the difference in read costs, Volume Shadowing will start sending some of the reads to the remote disk, allowing them to be satisfied faster.
  - With large differences in read costs, the local queue depth has to get very high before any reads start to be directed to the remote disk. The local disk must handle more of the burst of reads.

How should I set Read Costs for Host-Based Volume Shadowing?

Example:

– Local random read disk response time is 6 milliseconds

– Circuit path length between sites is 200 kilometers (about 124 miles), so round-trip time between sites is 2 milliseconds.

– Remote disks are MSCP-served, so remote reads take 2 round trips, or 4 milliseconds, plus the regular 6 millisecond disk response time, for a total of 6 + (2 x 2) = 10 milliseconds (assuming the disk is idle).

– So when the local disk queue length rises to 1, a local disk read would have to be placed in the queue behind the first request, so it would take 6 x 2 = 12 milliseconds for the 2nd read request. At this point, if the local disk is likely to be idle, it becomes better to send a read to the remote disk (10 millisecond response time) than to add it to the queue for the local member disk (12 millisecond response time).

– So relatively small differences in Read Cost values allow reads to be biased toward the local disk, but retain the option to start sending some of the reads across to the disk at the opposite site if the local queue length gets too long.

How should servers in my disaster-tolerant cluster start up?

How should servers in my disaster-tolerant cluster start up?

– Divide node startup into 3 separate and distinct phases:

1. Booting
2. Mounting cross-site shadowsets
3. Starting applications and then allowing access to users

– Set all systems up to boot “Conversational” by default, either at the EFI (Integrity) or SRM (Alpha) level, so that instead of rebooting automatically after a crash or temporary power failure, they will always stop at the SYSBOOT> prompt, allowing manual intervention and control.

– System startup should be either strictly manually controlled or at least human-directed (i.e. use a spare SYSGEN parameter like USERD1 or something to indicate whether or not, and if so, how cross-site shadowsets should be mounted, and if applications should then be started up or not)

– Depending on the circumstances, the cross-site shadowsets might need to be:
  – Left unmounted for further manual troubleshooting, followed by appropriate manual action by the system manager
  – Mounted using member disks from both (all) sites at once
  – Mounted using only member disks from Site 1 (perhaps that site has the only valid copy of the data)
  – Mounted using only member disks from Site 2
  – Similarly, in a 3-site cluster, options to mount shadowsets using only member disks from Site 3, or a pair of sites: 1&2, 2&3, or 1&3

– Because preservation of data on disk is so crucial in a disaster-tolerant cluster, and it is important to avoid a “wrong-way” shadow copy, use the qualifier /POLICY=(REQUIRE_MEMBERS,VERIFY_LABEL) on all the MOUNT commands for cross-site shadowsets.

Questions?

HPE
POINTNEXT

Thank you

keith.parris@hpe.com

Hewlett Packard
Enterprise

Overall structure

- Keep the overall structure as simple as you can
- Balanced node performance is best
- Active – active (or 3 way active) delivers least interruption to service
- Understand the whole thing, then you can make better decisions when something unexpected (bad) happens

Discussion

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 ?

Best practices for OpenVMS clusters

Thank you for your participation

Connect Germany 2018 (Leipzig)

Colin Butcher (XDelta), with contributions and collaboration from Keith Parris (HPE Pointnext) and Nic Clews (DXC)

xdelta