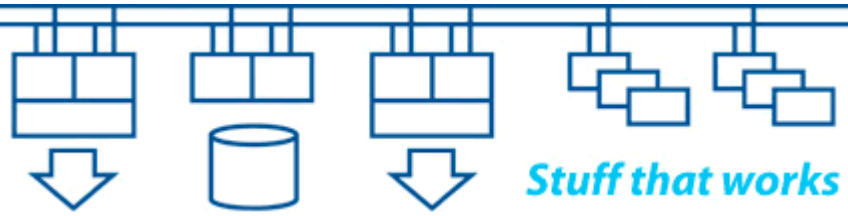
HP Dutchworld - OpenVMS sessions - November 2008

Clustering with OpenVMS:

1. Disaster-tolerant clusters – example
2. Cluster enhancements from OpenVMS Engineering (IP clusters, HBVS)

Colin Butcher, XDelta Limited

Copyright © Colin Butcher, XDelta Limited, Nov. 2008



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.

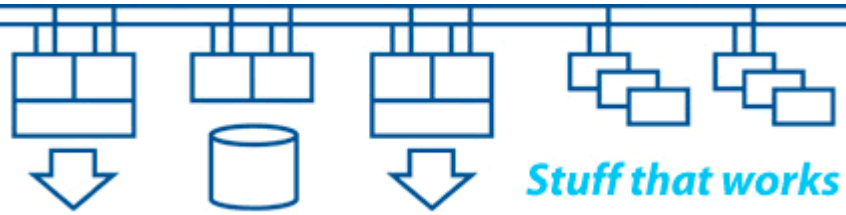
Copyright © Colin Butcher, XDelta Limited, Nov. 2008

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 (Mimer)
- 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

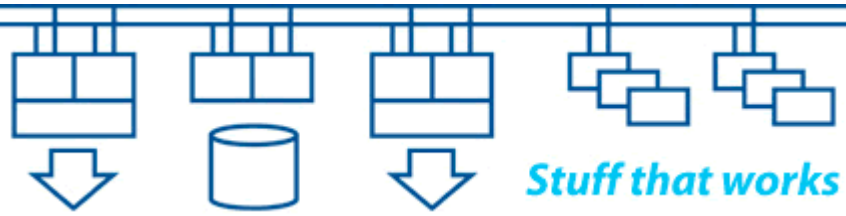
Copyright © Colin Butcher, XDelta Limited, Nov. 2008



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
- It's not a theoretical exercise!

Copyright © Colin Butcher, XDelta Limited, Nov. 2008



Do we really need 100% uptime?

- Safety-critical systems (especially 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.

Copyright © Colin Butcher, XDelta Limited, Nov. 2008

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?

Copyright © Colin Butcher, XDelta Limited, Nov. 2008

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

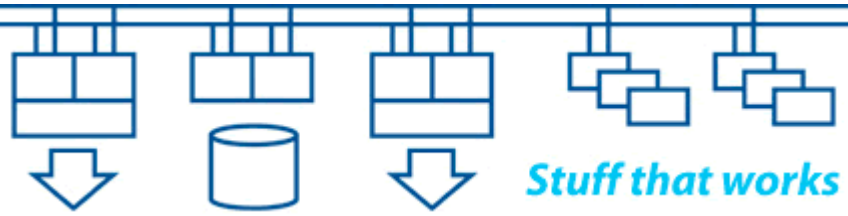
- What level of service outage can we tolerate?
- How quickly do we need to recover?
- How quickly do we need to be ready to deal with a subsequent failure?

Copyright © Colin Butcher, XDelta Limited, Nov. 2008

"Survivability test"

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

Copyright © Colin Butcher, XDelta Limited, Nov. 2008



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:

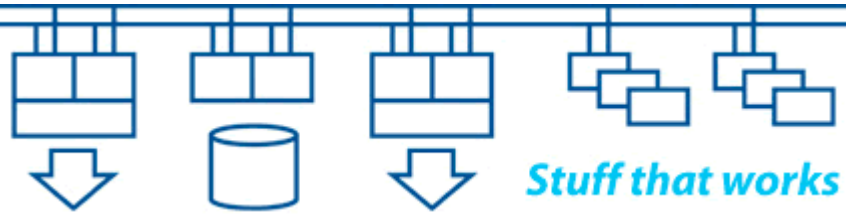
- Performance issues are often the cause of transient system failures and disruption
- The systems have to have sufficient capacity and performance to deal with the workload in an acceptable period of time

Availability is more important than performance

Copyright © Colin Butcher, XDelta Limited, Nov. 2008

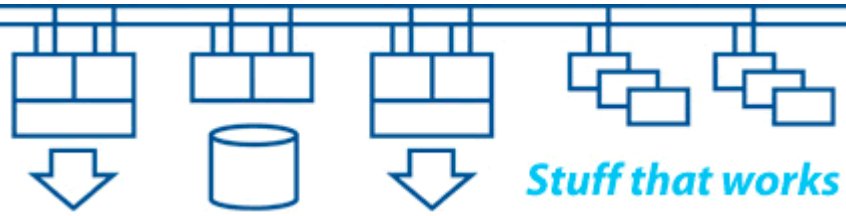
- Bandwidth – determines throughput
  - It's not just "speed", it's throughput in terms of "units of stuff per second"
- Latency – determines response time
  - Determines how much "stuff" is in transit through the system at any given instant
  - "Stuff in transit" is the data at risk if there is a failure
- 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

Copyright © Colin Butcher, XDelta Limited, Nov. 2008



- Size systems to meet realistic criteria, eg: throughput; response time; minimal loss of "data in transit"; etc.
- 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
- How will you generate a realistic load for testing?

Copyright © Colin Butcher, XDelta Limited, Nov. 2008



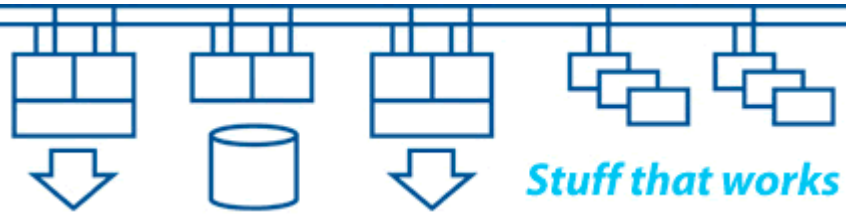
- Which parts of the system are mission-critical?
- What kind of failure do we prefer?
- What happens to our data when things go wrong?
- What state transitions occur during failure and recovery?
- How can we recover from a failure without data loss or data corruption?
- Should we automate decision making?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

Copyright © Colin Butcher, XDelta Limited, Nov. 2008

Split-site issues

- Effects of distance on network and storage protocols
- Symmetric or asymmetric operation – how good is your "crystal ball"?
- 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

Copyright © Colin Butcher, XDelta Limited, Nov. 2008



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 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 during failure and recovery
- 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

Copyright © Colin Butcher, XDelta Limited, Nov. 2008

How the OpenVMS clusters are configured

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

Copyright © Colin Butcher, XDelta Limited, Nov. 2008

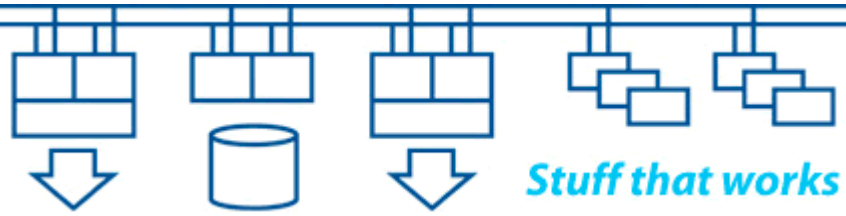
What external IP addressing do we use?

- All NICs in all systems are dual path to two separate Cisco switches (fibre, not copper)
- Uses "failsafe IP" for bandwidth and flexibility - failsafe IP addresses move from NIC to NIC on the same machine or 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.

Copyright © Colin Butcher, XDelta Limited, Nov. 2008

- 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

Copyright © Colin Butcher, XDelta Limited, Nov. 2008



- 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 by maintaining write bitmaps
- Lots of small shadow sets give good granularity and control over HBVS behaviour

Copyright © Colin Butcher, XDelta Limited, Nov. 2008

- The Integrity Server MP / EFI boot menus should be configured to disable auto-boot and to disable auto power-up following power loss
- The nodes boot from EVA presented Vdisk devices. Boot paths to SAN devices are configured using the BOOT_OPTIONS.COM mechanism by setting the boot device and the correct system root [SYSn.]
  Hint: adding a new node to an existing cluster means booting from another disk (eg: copy of the installation DVD) then mounting the target disc read-only
- Remove root [SYS0] to prevent anything booting accidentally into the cluster

Copyright © Colin Butcher, XDelta Limited, Nov. 2008

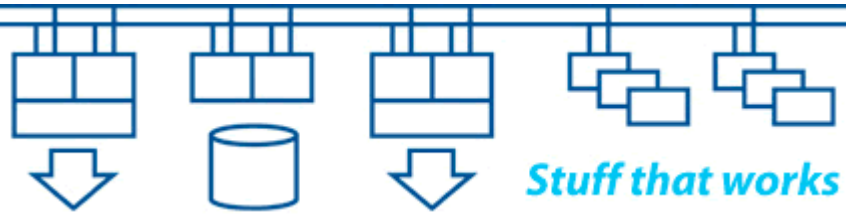
- OpenVMS reads the load image from the specified device and proceeds to boot from the specified root
- The cluster is formed
- LAN failover virtual LAN interfaces are created
- DTC_MOUNT_DISKS runs to mount the cluster-common disk – it prompts if needed
- The networking layers are started (DECnet, then 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

Copyright © Colin Butcher, XDelta Limited, Nov. 2008

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

Copyright © Colin Butcher, XDelta Limited, Nov. 2008



Thank you for your participation

Q & A

Copyright © Colin Butcher, XDelta Limited, Nov. 2008