# Welcome

## Oracle Rdb Forum - 2016

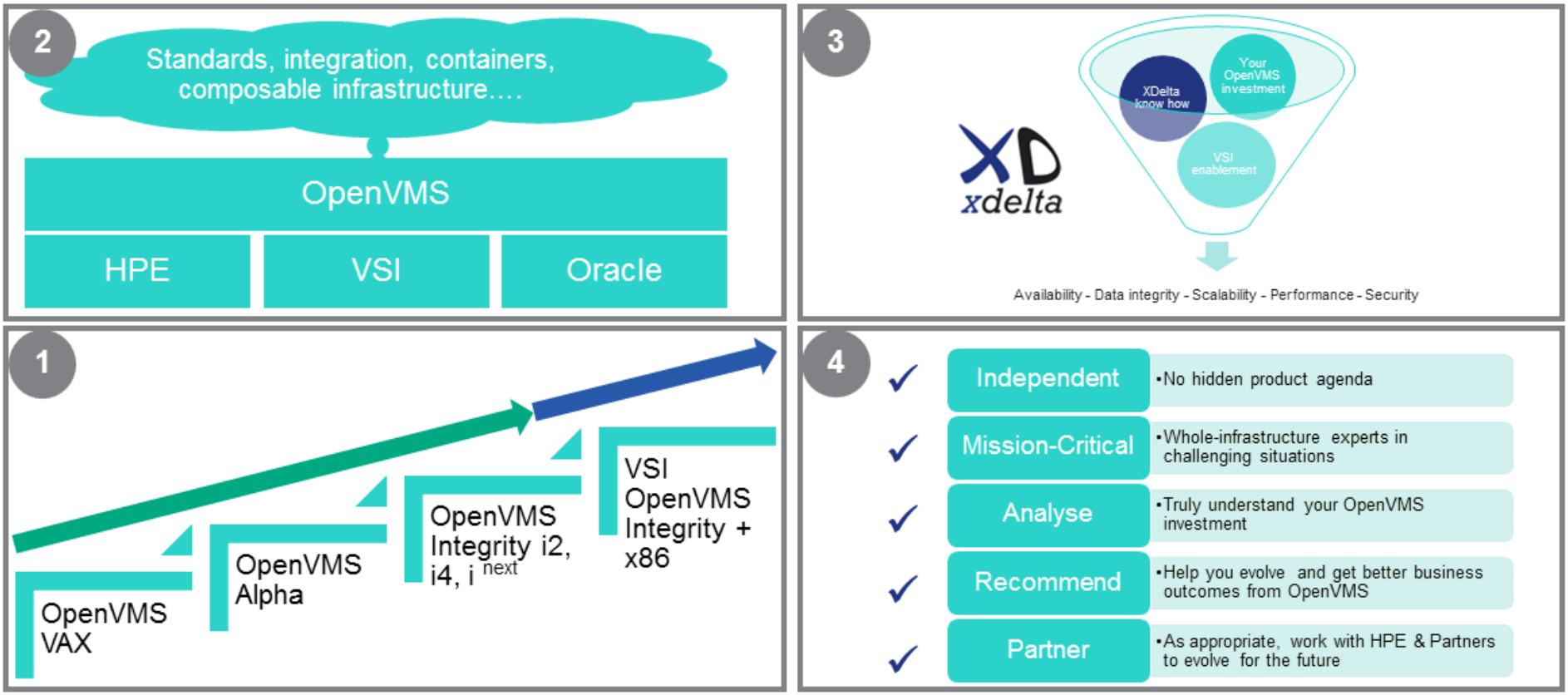
## Planning your move to VSI OpenVMS

Colin Butcher CEng FBCS CITP
Technical director, XDelta Limited
www.xdelta.co.uk

# XDelta – who we are

- VSI Professional Services Alliance member
- Independent consulting engineers since 1996:
  - UK based with international reach
  - Delivering OpenVMS based systems for 30+ years
- Technical leadership for business-critical systems
  - Design, planning and implementation
  - Mentoring and skills transfer
  - Systems engineering background
- Gartner (2009):
  - Identified XDelta as one of few companies world-wide capable of OpenVMS platform migration projects

# XDelta - a trusted advisor to advance your critical OpenVMS application infrastructure




Hewlett Packard Enterprise

# Part 1

1. Why move to VSI OpenVMS ?
2. Benefits of moving to VSI OpenVMS
3. VSI OpenVMS on Alpha
4. VSI OpenVMS on HPE Integrity Servers
5. Moving to VSI OpenVMS – what's involved ?
6. Transition - design and planning
7. Moving to VSI OpenVMS – next steps

# VSI OpenVMS – evolution not revolution

- Need minimal risk of disruption to business:
  - Availability, data integrity, scalability, performance, security
- Need supported systems to run business
- Want to reduce long-term cost of ownership
- Cost of redesign or replacement can be prohibitive
- Complexity and difficulty of moving applications and data can make it almost impossible to change platform

# VSI OpenVMS – mission-critical platform

- Most OpenVMS systems are customer specific:
  - Tight integration with operating system and infrastructure
- Well engineered operating system:
  - Well structured and documented
  - Nothing better for multi-site mission-critical capabilities
  - Scales very well from small to large implementations
- Distinct culture:
  - People who like to understand things
  - People who like to do things properly

# Availability, performance, security

- Multi-site disaster tolerance
- Operating system based synchronous data replication
- Scale-up with big blade hardware
- Scale-out with “shared everything” OpenVMS clusters
- Low IO latency, high IO throughput
- Secure object protection, alerting and auditing

# Server hardware – design trends

- Standardisation and modularity
- Multi-core processors, high core counts, lower clock rates, lower power consumption
- More parallelism in memory and IO subsystems
- Solid-state memory and large memory configurations
- NUMA (non-uniform memory access)

# Infrastructure hardware

- HPE 3PAR and XP / XP7 fibrechannel arrays
- Multivendor fibrechannel storage:
  - Pure Storage FA400 and //m series fibrechannel arrays
  - More in progress
- SAS SSD devices for local storage
- SAN fabric switches at 8 GigFC / 16 GigFC
- Data network switches at 10 GigE

# Part 2

1. Why move to VSI OpenVMS ?
2. Benefits of moving to VSI OpenVMS
3. VSI OpenVMS on Alpha
4. VSI OpenVMS on HPE Integrity Servers
5. Moving to VSI OpenVMS – what's involved ?
6. Transition - design and planning
7. Moving to VSI OpenVMS – next steps

# VSI OpenVMS – development and support

- Future OpenVMS releases from VSI:
  - hardware platforms: x86-64, IA64, Alpha
  - new features and enhancements
  - up-to-date components (eg: SSL, Java)
  - prepares the way for transition to OpenVMS on x86-64
  - “rights to new versions” is part of VSI direct licensing strategy

- Support by VSI for VSI releases of OpenVMS:
  - Security patches, bugfix patches, patch rollup updates, new versions of OpenVMS and layered products
  - Support beyond end 2020

# VSI OpenVMS V8.4-2L1 – SSL update

- Updated SSL based on 1.0.2h
- APIs have changed in OpenSSL implementations
- Updated components that use SSL:
  - Encryption algorithms
  - Key exchange mechanisms
  - TLS 1.2
- Co-existence mechanism with earlier SSL 0.9.8

# VSI OpenVMS – new TCP/IP stack

Some of the major updates include:

- OpenSSL 1.0.2
- SSH (V1 & V2)
- DHCP v3
- IPv6 (complete application protocols supported)
- IPSEC (full support)
- Bind 9.9
- Kerberos 5
- advanced features such as IPS, paired network interface support, and improved performance monitoring capabilities

# VSI OpenVMS on HPE Integrity -i4 servers

- Significant step up from HPE Integrity -i2 servers:
  - rx2800-i4 (16 cores) > 2x rx2800-i2 (8 cores) - same space
  - bl860c-i4 (16 cores) > bl870-i2 (16 cores) - half space
  - bl870c-i4 (32 cores) > bl890-i2 (32 cores) - half space
  - bl890c-i4 (64 cores) > 2x bl890-i2 (32 cores) - same space
- Socket based licences - per-core costs reduced
- Significant step up from AlphaServer GS1280, with modern storage and network infrastructure

# Migrating from Alpha to Integrity

- bl870c-i4 and bl890c-i4 are good for GS1280 migration
- Multi-core processors, NUMA, hyperthreading
- 10GigE network, 8GigFC SAN
- Smaller footprint, lower power, less cooling
- Blade chassis connectivity for bl8x0c-i4
- Fibrechannel storage migration

# Part 3

1. Why move to VSI OpenVMS ?
2. Benefits of moving to VSI OpenVMS
3. VSI OpenVMS on Alpha
   - evaluation kit, April 2016
   - “limited release”, September / October 2016
4. VSI OpenVMS on HPE Integrity Servers
5. Moving to VSI OpenVMS – what’s involved ?
6. Transition - design and planning
7. Moving to VSI OpenVMS – next steps

# Alpha – next steps

- If staying with Alpha is of interest, please get in touch
- Try the VSI OpenVMS for Alpha V8.4-2 evaluation kit in a safe non-production environment
- Understand the support issues and regulatory / legal implications if you choose to stay with old hardware
- Consider the surrounding infrastructure as well, especially storage subsystems

# Potential ways forward with Alpha

- Extended support past HPE deadline for organisations still reliant on Alpha with VSI OpenVMS “limited release”
- Physical hardware or emulated hardware
- Emulated systems can be used in a virtual environment in limited circumstances
- Buy time to complete migrations to Integrity
- Get current on Alpha before migrating to Integrity or x86-64
- Continue with Alpha until OpenVMS on x86-64 is available as a “production ready” system

# Some issues around staying with Alpha

- Becomes easy to put off moving to Integrity - creates a bigger challenge later when moving to x86-64
- Big system performance is hard to deliver from an emulator
- Limited hardware capabilities of Alpha platforms:
  - 1Gbps or 2Gbps fibrechannel (KGPSA)
  - 1Gbps ethernet (DEGPA)
  - Tied to older storage and infrastructure (EVA etc.)
- Greater risk of  hardware failures
- Hardware support becoming more difficult and expensive
- Space, power, cooling, etc.

# Part 4

1. Why move to VSI OpenVMS ?
2. Benefits of moving to VSI OpenVMS
3. VSI OpenVMS on Alpha
4. VSI OpenVMS on HPE Integrity Servers
5. Moving to VSI OpenVMS – what's involved ?
6. Transition - design and planning
7. Moving to VSI OpenVMS – next steps

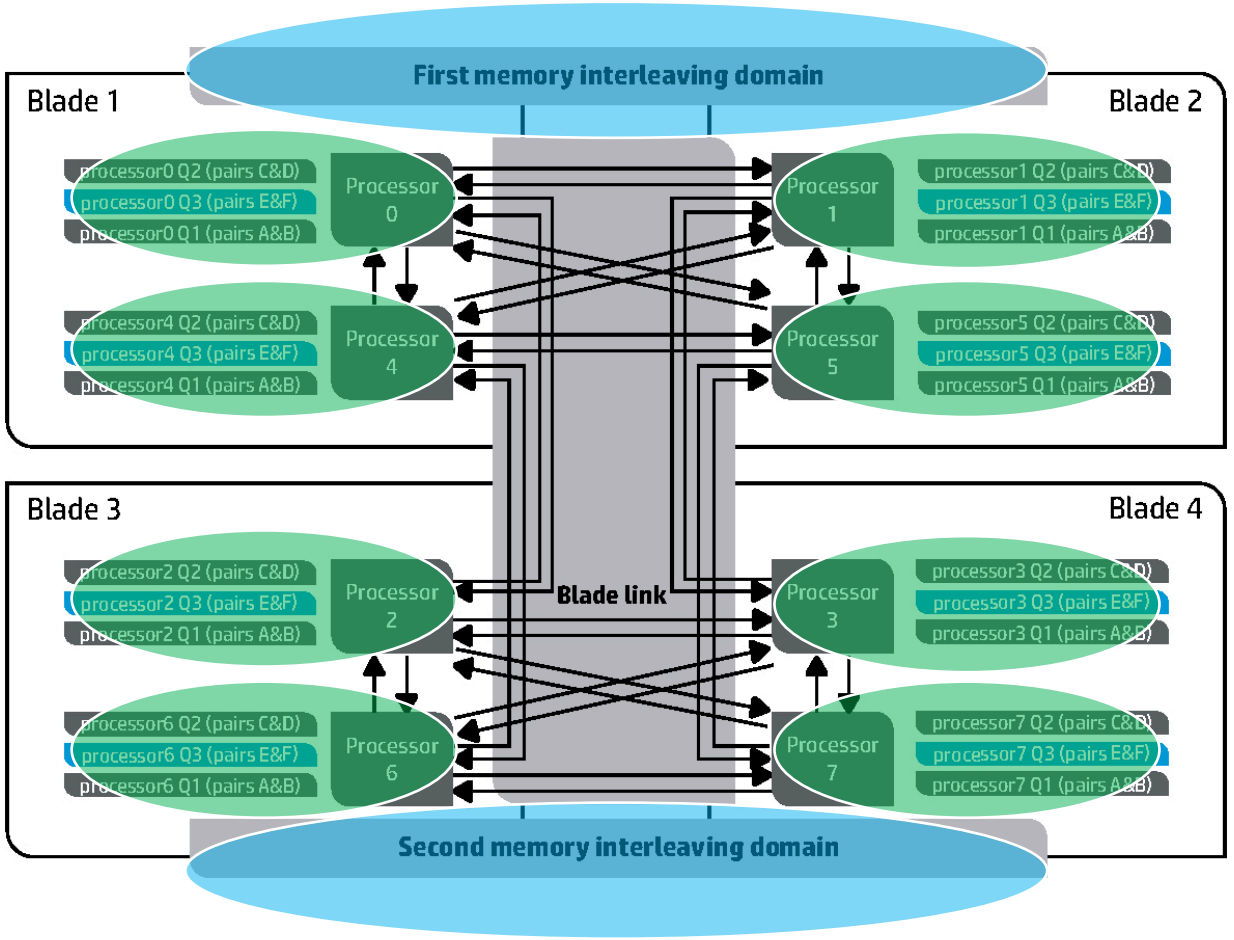
# OpenVMS - High core count

- CPU 00 is the primary CPU - try to reduce its workload
- Fastpath CPU selection - be aware of physical layout
- CPU choice for dedicated lock manager
- CPU choice for TCPIP (BG device)
- CPU choice for PEdriver
- Consider physical layout - RADs and NUMA

# OpenVMS - NUMA

- OpenVMS uses large shared memory regions:
  - XFC (50 % available memory by default)
  - RMS global buffers
  - Global sections (especially database caches)
  - Memory disc driver (MD devices)
- Set memory interleave behaviour with “memconfig” at EFI shell (requires reboot)
- Useful starting point for OpenVMS is “mostly UMA”

# Memory architecture – bl890c-i4 (10 RADs)








xdelta

# OpenVMS - Hyperthreading

- Hyperthreading is extremely workload dependent
- In general the OpenVMS scheduler does a better job
- Enable / disable hyperthreading with “cpuconfig” at EFI shell (requires reboot)
- “CPU” count will appear to double when enabled
  - V8.4-2 supports 64 “scheduling units”
  - V8.4-1H1 supports 32 “scheduling units”
  - Can “STOP/CPU <nn>” to selectively shut down co-threads

# Performance engineering

- Without good data you cannot do good performance work

- Avoid guesswork - run T4 all the time
- If needed, use T4 “expert mode” and SDA extensions
- New tools and services coming

- 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

# Tips – VSI OpenVMS on -i(n) servers

- Disable devices you don’t use (SYSMAN)

- Experiment with memory interleave setting (memconfig)
- Use memory reservations to place XFC etc.
- Experiment with RAD memory layout if you have a lot of shared memory regions in your applications

- Set fastpath CPUs for fastpath capable devices
- Experiment with dedicated CPU for TCPIP + LCKMGR

- Experiment with hyperthreading (cpuconfig) if you have a lot of small concurrently active processes

# Part 5

1. Why move to VSI OpenVMS ?
2. Benefits of moving to VSI OpenVMS
3. VSI OpenVMS on Alpha
4. VSI OpenVMS on HPE Integrity Servers
5. Moving to VSI OpenVMS – what's involved ?
6. Transition - design and planning
7. Moving to VSI OpenVMS – next steps

# What kind of move do you intend ?

- Stay with same server & infrastructure hardware
  - Update operating system and products
- Stay with same architecture and update hardware
  - Change hardware (servers, infrastructure)
  - Update operating system and products
- Migrate to newer architecture
  - Change hardware (servers, infrastructure)
  - Migrate and update operating system and products
  - Migrate applications

# Same server & infrastructure hardware

- Update operating system and products

- New licences, new kits
- Upgrade system disc(s) or build new system disc(s) ?

- Take advantage of new features ?
- Configuration changes, eg: new TCP/IP stack ?
- What else might be affected ?

- Similar process to any other operating system update

# Same architecture, update hardware

- Update hardware, plus operating system and products

- New hardware, new licences, new kits
- Upgrade system disc(s) or build new system disc(s) ?

- Take advantage of new features ?
- Configuration changes, eg: new TCP/IP stack ?
- What else might be affected ?

- Similar process to any other hardware replacement

# New architecture, change hardware

- Change architecture, update operating system and products

- Migrate applications, new hardware, new licences, new kits
- Build new system disc(s), migrate data

- Understand differences between architectures
- Get current on old architecture first
- Take advantage of new features ?
- What else might be affected ?

- Similar process to any other architecture migration project

# Disc layout – system, common, data

- Separate the system from the data:
  - Per-site / per-node system discs
  - Cluster-wide common disc(s) – UAF, queues, etc.
  - Cluster-wide application and data discs
- Have alternate system discs for rapid switch-over:
  - Use array to clone discs
  - Can build offline copy of whole system for testing
- Write configuration and architecture independent DCL

# Making use of CPU

- Introduce parallelism in workflows where possible
- Hyperthreading (IA64, x86-64) – very workload dependent
- Fastpath IO devices and distributed interrupt handling
- Dedicated CPU for lock manager (local locking)
- Compression and encryption, eg: SCS compression, BACKUP compression and encryption (hardware based)
- QUANTUM, workload dependent – many SYSGEN parameter defaults changed in V8.2
- Power management

# Making use of memory

- Use all that memory (XFC, resident images, DECram etc.)
- Revisit working set sizes - WSMAX and process quotas (WSDEF / WSQUO / WSEXT / PGFLQUO)
- Use RMS global buffers
- Revisit RMS system defaults
- XFC - beware “double caching”
- GH regions map lots of memory with a small number of page table entries
- INSTALL /RESIDENT and GH region size
- 64bit P2 space and memory reservations
- DECram and HBVS to disc

# Making use of storage I/O

- FC bandwidth is important – what else are you sharing your storage bandwidth with? Why?
- Rotational latency no longer matters
- SAN zoning, preferred paths and inter-site links
- Array cache size and volume characteristics
- HBVS (shadowing):
  - only shadow what you really need to
  - many shadow sets let you control how rapidly shadow copying proceeds during recovery
  - mini-copy and mini-merge policies

# Making use of storage infrastructure

- Use both shadowing and array features
- Array based replication – synchronous and asynchronous:
  - Long distance off site copies
  - Virtual tape off site copies
- Snaps, clones:
  - Don’t leave snaps running for a long time
- Minimise array controller overheads:
  - Have enough spindles to only use mirroring and striping
- Spread the load over array ports and controllers with explicit path selection

# Making use of network I/O

- Split protocols across adapters
- 802.1Q VLAN tagging
- QoS (Quality of Service) controls
- Jumbo frames (LAN_FLAGS and LANCP)
- SCS window size
- TCPIP multiple NICs and IP addressing scheme design
- TCPIP - keepalive, delack, MTU sizes
- SCS and DECnet-Plus single rail with LAN failover *v* multiple rails with load balancing
- Understand latency effects of long-distance networks
- RMS block sizes for network IO

# Part 6

1. Why move to VSI OpenVMS ?
2. Benefits of moving to VSI OpenVMS
3. VSI OpenVMS on Alpha
4. VSI OpenVMS on HPE Integrity Servers
5. Moving to VSI OpenVMS – what's involved ?
6. Transition - design and planning
7. Moving to VSI OpenVMS – next steps

# Design goals

- Transition with minimal disruption
- Operational safety – minimise risk of errors
- Incorporate logging and information gathering
- Adapt to changing requirements (performance, scalability)
- Minimise complexity and risk
- Preserve data integrity and access to old data

# Design for transition

- Transition is often the main issue for planning and design
- How big a downtime window can we have ?
- When is a downtime window possible ?
- Who do we need to be involved ?
- How do we know it’s been successful ?

# Planning for transition

- How can we minimise the impact of transition ?
- Much cheaper to make mistakes on paper
- Need to involve everyone who might be affected
- How do we minimise what we have to do under pressure ?
- What can be done in advance ?
- What can be done afterwards ?

# Transition – minimising the risk of mistakes

- Have everything written down in advance:
  - start and end states
  - activities
  - timeline
  - responsibilities
  - approximate timings
  - expected results
  - decision points
  - back-out actions

- Don’t make it up on-the-fly !

# Testing - systems are never perfect

- If there's an issue after transition, did it exist before, or is it a real problem attributable to transition ?
- What issues (if any) do we need to resolve first ?
- Do we have reference data sets and reference systems ?
- Can we "parallel run" and compare results ?
- What's the best we can reasonably do ?

# Part 7

1. Why move to VSI OpenVMS ?
2. Benefits of moving to VSI OpenVMS
3. VSI OpenVMS on Alpha
4. VSI OpenVMS on HPE Integrity Servers
5. Moving to VSI OpenVMS – what's involved ?
6. Transition - design and planning
7. Moving to VSI OpenVMS – next steps

# Upgrade and integrate

- Evolution is lower risk than revolution
- Migrating to VSI OpenVMS now will help you migrate to x86-64 when it becomes “production ready”
- Retain best of existing investment
- Code compatibility and data portability
- Phased approach over time, not “rip and replace”

# It’s easy to proceed

- Licences, support and services direct from VSI, or partners
- VSI operating system and layered products – active development and release streams
- OpenVMS patches and improvements, especially security
- ISV layered products
- HPE servers and multivendor infrastructure

# Thank you

Oracle Rdb Forum - 2016

Planning your move to VSI OpenVMS

Colin Butcher CEng FBCS CITP
Technical director, XDelta Limited
www.xdelta.co.uk