
Welcome

Oracle Rdb Forum - 2016

Planning your move to VSI OpenVMS

Colin Butcher CEng FBCS CITP

Technical director, XDelta Limited

www.xdelta.co.uk

Welcome

Oracle Rdb Forum - 2016年

VSI OpenVMSへの移行の計画

Colin Butcher CEng FBCS CITP

Technical director、XDelta Limited

www.xdelta.co.uk

XDelta – who we are



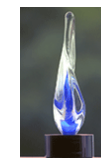
- VSI Professional Services Alliance member
- Independent consulting engineers since 1996:
 - UK based with international reach
 - Delivering OpenVMS based systems for 30+ years
- Technical leadership for business-critical systems
 - Design, planning and implementation
 - Mentoring and skills transfer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



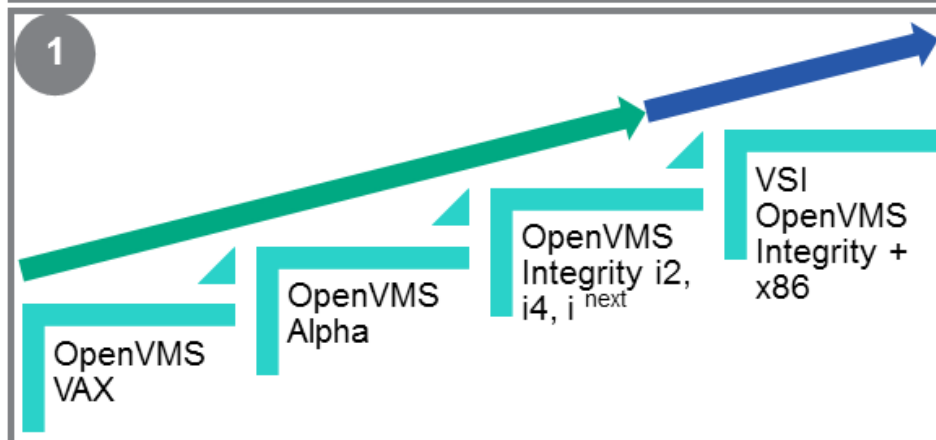
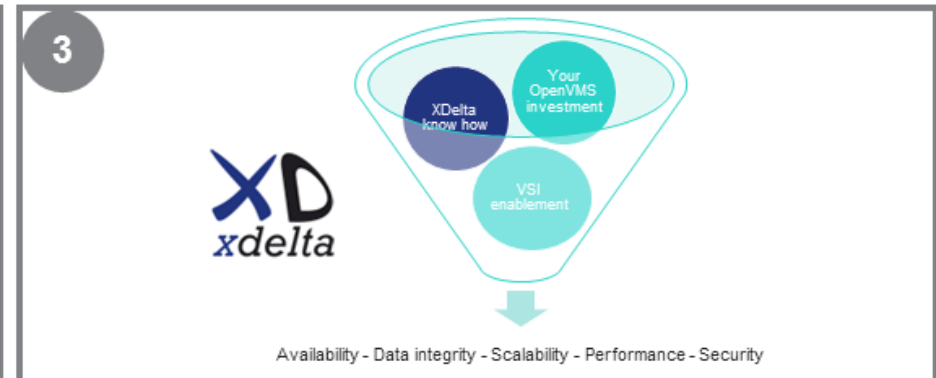
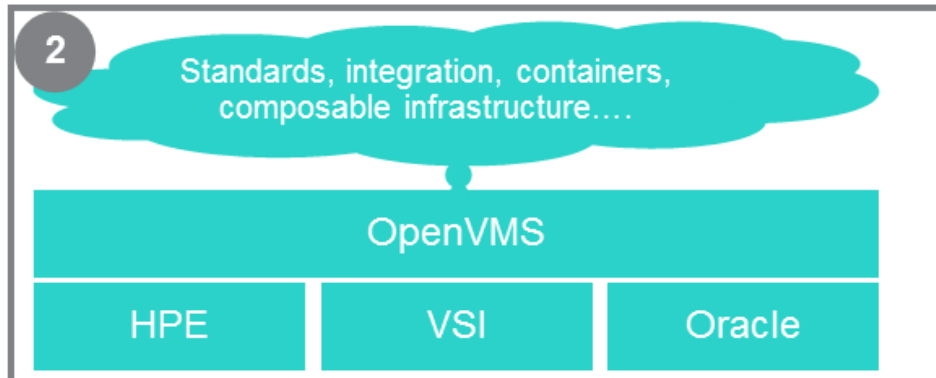
XDeltaについて



- VSIプロフェッショナル・サービス・アライアンス・メンバー
- 1996年以來の独立コンサルティングエンジニア：
 - イギリスを拠点に国際的に展開
 - OpenVMSベースシステムを30年以上の間提供
- ビジネス・クリティカル・システムの技術リーダーシップ
 - デザイン、計画、およびインプリメンテーション
 - メンタリングとスキル移転
 - システムエンジニアリングバックグラウンド
- Gartner(2009年)：
 - OpenVMSプラットフォーム移行プロジェクトが可能な国際的企業としてXdeltaを認定

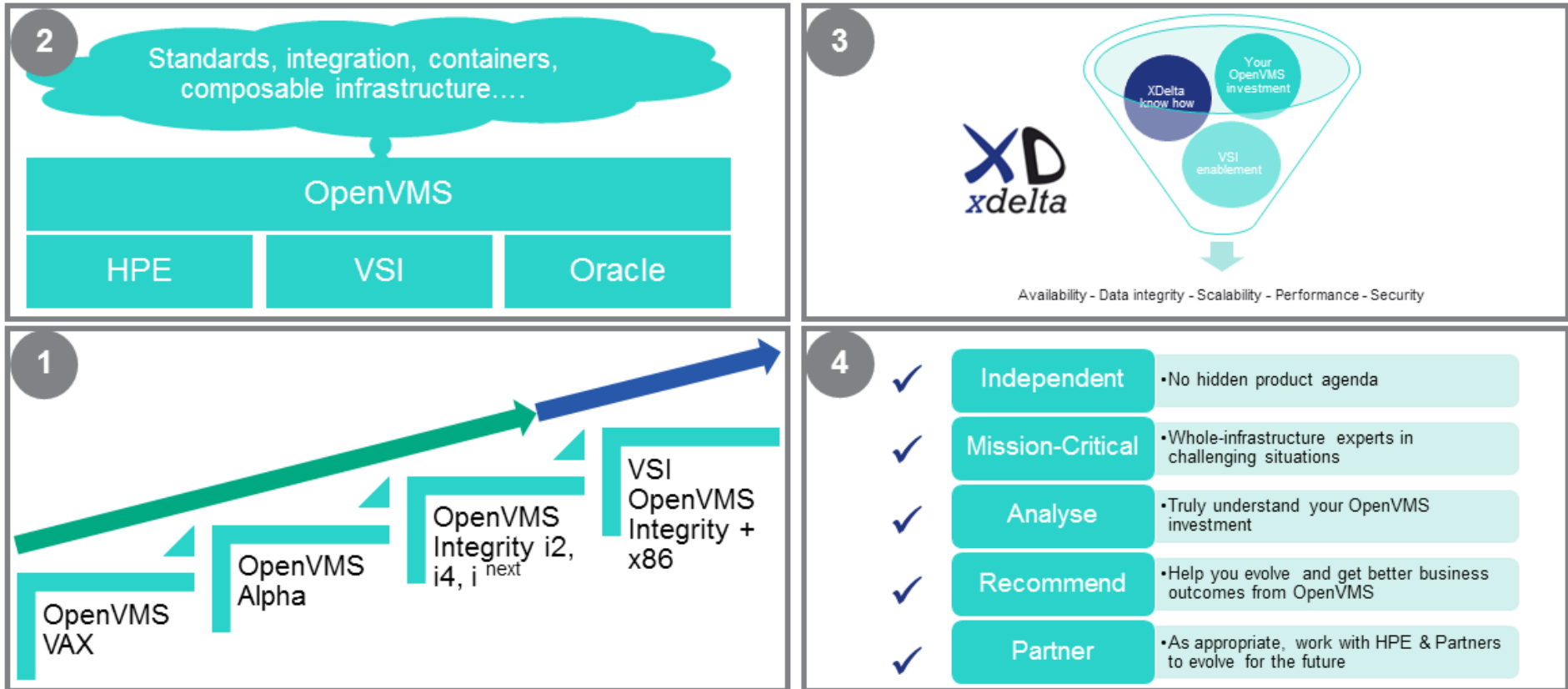


XDelta - a trusted advisor to advance your critical OpenVMS application infrastructure



- 4
- ✓ Independent •No hidden product agenda
 - ✓ Mission-Critical •Whole-infrastructure experts in challenging situations
 - ✓ Analyse •Truly understand your OpenVMS investment
 - ✓ Recommend •Help you evolve and get better business outcomes from OpenVMS
 - ✓ Partner •As appropriate, work with HPE & Partners to evolve for the future

Xdelta – お客様のクリティカルOpenVMSアプリケーションインフラを
進化させる信頼できるアドバイザー



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

パート1

1. なぜVSI OpenVMSに移行するのか
2. VSI OpenVMSに移行することの利点
3. VSI OpenVMS on Alpha
4. VSI OpenVMS on HPE Integrity Servers
5. VSI OpenVMSへの移行 – その内容は？
6. 移行 - デザインと計画
7. VSI OpenVMSへの移行 - 次のステップ

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－革命ではなく革新

- ビジネス中断のリスクを最小化する必要性：
 - 可用性、データの完全性、スケーラビリティ、性能、セキュリティ
- 事業継続にはサポートされたシステムが必要。
- 長期的に所有コストを下げたい。
- 再設計や置き換えは予算的に不可。
- アプリケーションとデータを移行する複雑さと難しさのため、プラットフォーム変更はほぼ不可能。

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

VSI OpenVMS - ミッションクリティカルプラットフォーム

- ほとんどのOpenVMSシステムは顧客固有：
 - オペレーティングシステムとインフラストラクチャーの緊密な統合
- よく設計されたオペレーティングシステム：
 - よく構造化されて、文書化されている
 - マルチサイトのミッションクリティカルの構築に最適
 - 小さな構成から大規模にスケールアップ可能
- 独特の文化：
 - 物を理解することが好きな人々
 - 物事を正しく行うことが好きな人々

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

可用性、性能、セキュリティ

- マルチサイトの災害耐性
- オペレーティングシステムベースのデータの同期レプリケーション
- 大きなブレードハードウェアによるスケールアップ
- “Shared everything” OpenVMSクラスタによるスケールアウト
- 低いIO待ち時間、高いIOスループット
- 安全なオブジェクト保護、警報、および監査

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)

サーバーハードウェア – デザイン動向

- 標準化とモジュール性
- マルチコアプロセッサ、多くのコア数、低いクロック、低い電力消費量
- メモリーとIOサブシステムの並列処理
- ソリッド・ステート・メモリーと大量のメモリーコンフィギュレーション
- 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

インフラストラクチャーハードウェア

- HPE 3 PARとXP / XP7 fibrechannelアレイ
- マルチベンダーfibrechannelストレージ：
 - Pure Storage の fibrechannel アレイ: FA400、//mシリーズ
 - さらに追加予定
- ローカルなストレージとしてSAS SSDデバイス
- SANファブリックスイッチ 8 GigFC / 16 GigFC
- データのネットワークスイッチ 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

パート2

1. なぜVSI OpenVMSに移行するのか
2. VSI OpenVMSに移行することの利点
3. VSI OpenVMS on Alpha
4. VSI OpenVMS on HPE Integrity Servers
5. VSI OpenVMSへの移行 – その内容は？
6. 移行 - デザインと計画
7. VSI OpenVMSへの移行 - 次のステップ

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 - 開発とサポート

- 今後のOpenVMSはVSIからリリース：
 - ハードウェアプラットフォーム: x86-64、IA64、Alpha
 - 新しい機能と性能向上
 - 最新のコンポーネント(eg: SSL、Java)
 - x86-64のOpenVMSへの移行方法の準備
 - “rights to new versions” などのVSIの直接ライセンス計画
- VSI OpenVMSリリースのVSIによるサポート：
 - セキュリティパッチ、bugfixパッチ、パッチロールアップアップデート、OpenVMSの新バージョン、およびレイヤード製品
 - 2020年を越えるサポート

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 V8.4-2L1 - SSLアップデート

- SSL 1.0.2hに基づいたアップデート
- OpenSSLのインプリメンテーションでのAPI変更
- SSLを使うコンポーネントのアップデート：
 - 暗号化アルゴリズム
 - キー交換メカニズム
 - TLS 1.2
- 以前のSSL0.9.8との共存メカニズム

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 - 新しいTCP/IPスタック

主要なアップデート :

- OpenSSL 1.0.2
- SSH (V1 & V2)
- DHCP v3
- IPv6(サポートされた完全なアプリケーションプロトコル)
- IPSEC(完全なサポート)
- Bind9.9
- Kerberos5
- IPS、ネットワークインタフェースのペアのサポート、改善された性能
監視などの高度な機能

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

VSI OpenVMSでのHPE Integrity-i4サーバーの利用

- HPE Integrity-i2サーバーからの重要なステップアップ：
 - rx2800-i4 (16コア) > 2x rx2800-i2 (8コア) - 同じスペース
 - bl860c-i4 (16コア) > bl870-i2 (16コア) - 半分のスペース
 - bl870c-i4 (32コア) > bl890-i2 (32コア) - 半分のスペース
 - bl890c-i4 (64コア) > 2x bl890-i2 (32コア) - 同じスペース
- ソケットベースのライセンス - コアごとのコストの低減
- 最新のストレージとネットワークインフラストラクチャーによるAlphaServer GS1280からの大きなステップアップ

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

AlphaからIntegrityに移行

- bl870c-i4とbl890c-i4はGS1280移行に最適
- マルチコアプロセッサ、NUMA、ハイパースレッド
- 10GigEネットワーク、8GigFC SAN
- より小さなフットプリント、より少ないパワー、より少ない冷却
- bl8x0c-i4のブレードシャシ接続
- Fibrechannelストレージへの移行

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

パート3

1. なぜVSI OpenVMSに移行するのか
2. VSI OpenVMSに移行することの利点
3. VSI OpenVMS on Alpha
 - 2016年4月の評価キット
 - 2016年9月または10月の限定リリース
4. VSI OpenVMS on HPE Integrity Servers
5. VSI OpenVMSへの移行 – その内容は？
6. 移行 - デザインと計画
7. VSI OpenVMSへの移行 - 次のステップ

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

Alpha - 次のステップ

- Alphaの継続利用をご希望でしたら、VSIに連絡してください。
- 安全なテスト環境でVSI OpenVMS Alpha V8.4-2評価キットを試してください。
- 古いハードウェアに留まる場合、サポートの問題と規制/法的責任を考慮してください。
- 周辺機器のインフラストラクチャー、特にストレージサブシステムに注意してください。

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

Alpha継続について考えられる方法

- VSI OpenVMS限定リリースを利用した、Alphaに依存している組織への、HPEサポート終了以降のサポート提供
- 物理ハードウェアまたはエミュレートされたハードウェア
- エミュレートされたシステムは、一定の制限において、バーチャル環境で用いることができます。
- Integrityへの移行完了を遅らせる。
- Integrityまたはx86-64に移行する前に、Alpha上で最新バージョンに更新しておく。
- x86-64のOpenVMSが実用システムとしてリリースされるまで、Alphaを継続利用する。

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.

Alphaの継続に関するいくつかの問題

- Integrity移行を先延ばしできる - x86-64への移行時により大きな挑戦となります。
- エミュレータでは高いシステム・パフォーマンスは期待できない。
- Alphaプラットフォームの制限されたハードウェア機能：
 - 1Gbpsまたは2Gbpsのfibrechannel(KGPSA)
 - 1Gbpsのイーサネット(DEGPA)
 - 古いストレージとインフラストラクチャー(EVAなど)への限定
- ハードウェア障害のリスクの増大
- より困難で高価なハードウェアサポート
- スペース、パワー、冷却など

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

パート4

1. なぜVSI OpenVMSに移行するのか
2. VSI OpenVMSに移行することの利点
3. VSI OpenVMS on Alpha
4. **VSI OpenVMS on HPE Integrity Servers**
5. VSI OpenVMSへの移行 – その内容は？
6. 移行 - デザインと計画
7. VSI OpenVMSへの移行 - 次のステップ

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 – 多数のCPUコア

- CPU 00はプライマリCPU - ワークロードを減らしてください。
- Fastpath CPU の選択 - 物理レイアウトに注意してください。
- 専用ロックマネジャーのCPUの選択
- TCPIP(BGデバイス)用のCPUの選択
- PEdriver用のCPUの選択
- 物理レイアウトの考慮 - RADとNUMA

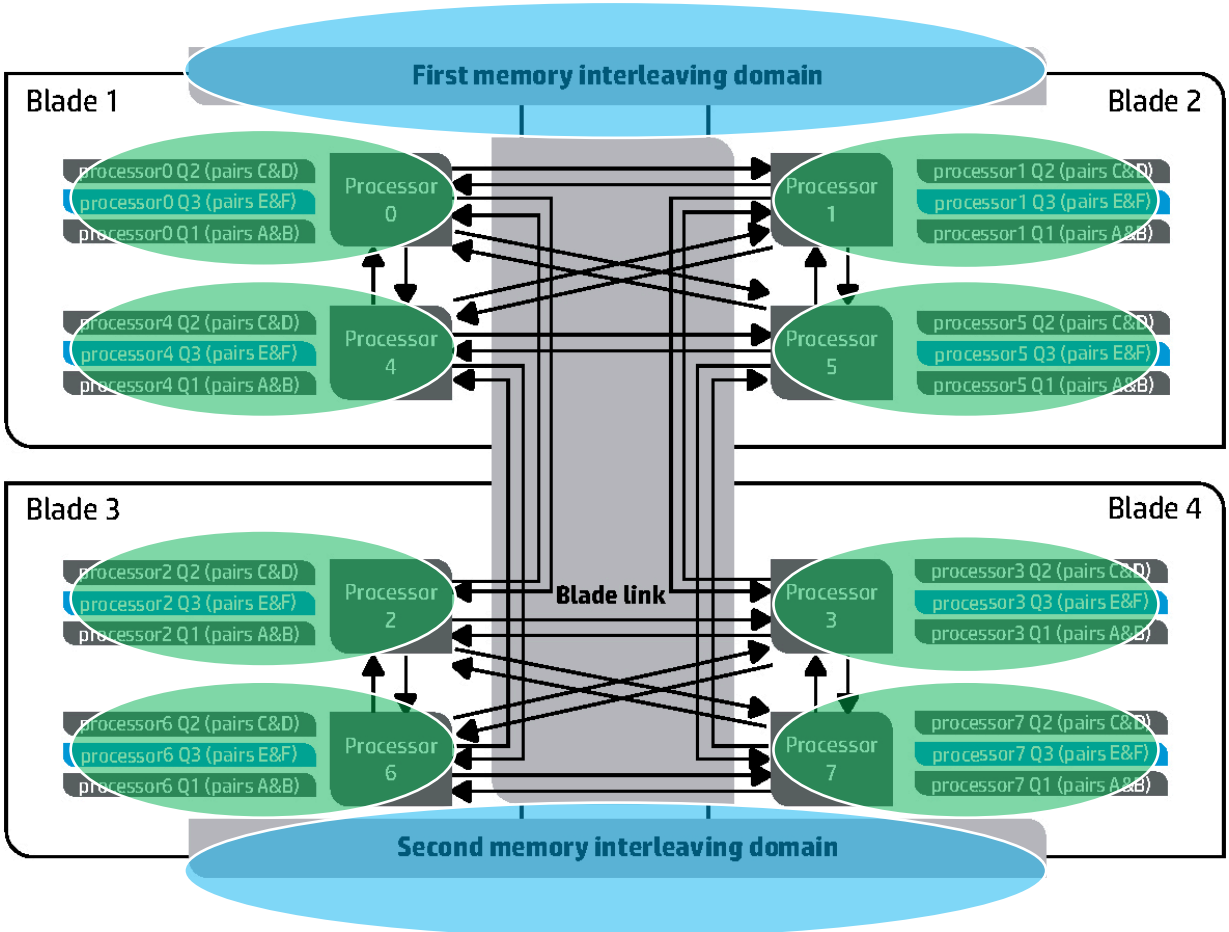
OpenVMS - NUMA

- 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”

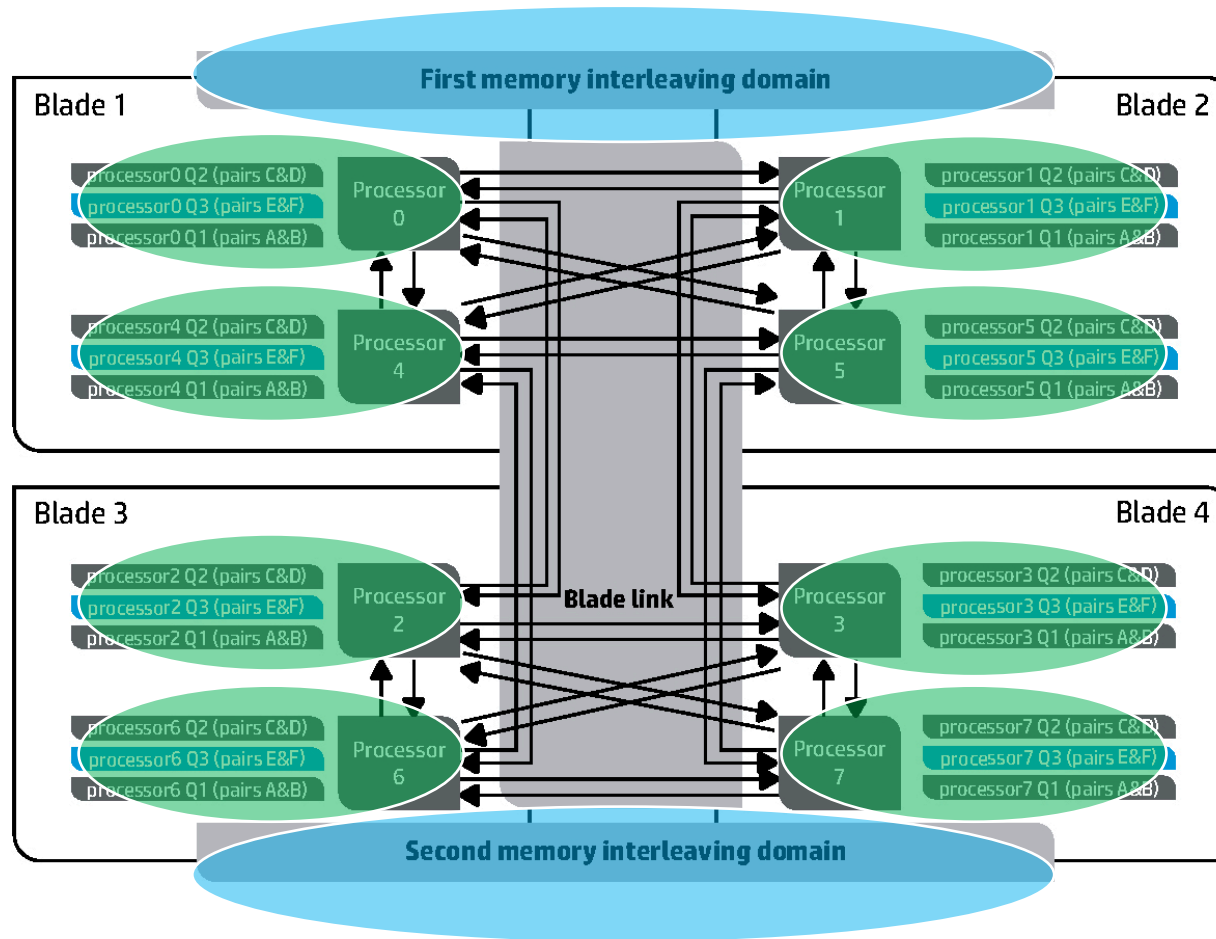
OpenVMS - NUMA

- OpenVMSは大きな共用メモリアリジョンを使います：
 - XFC(デフォルトで利用可能なメモリの50%)
 - RMSグローバルバッファ
 - グローバル・セクション(特にデータベースキャッシュ)
 - メモリーディスクドライバー(MDデバイス)
- EFIシェルで、メモリアリジョンを memconfig で設定してください (再起動が必要)。
- 最初のOpenVMS設定として "mostly UMA" が適当です。

Memory architecture – bl890c-i4 (10 RADs)



メモリーアーキテクチャ--bl890c-i4 (10 RAD)



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

OpenVMS - ハイパースレッド

- ハイパースレッドの効果はワークロードに強く依存します。
- 一般にOpenVMSスケジューラの方がよい仕事をします。
- EFIシェルで、「cpuconfig」によってハイパースレッドを有効/無効にできます。(再起動が必要)
- 有効時に「CPU」カウントは二倍になります。
 - V8.4-2 は64個の「スケジューリングユニット」をサポートします。
 - V8.4-1H1 は32個の「スケジューリングユニット」をサポートします。
 - 「STOP/CPU <nn>」により片方のスレッドを止められます。

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

性能改善のエンジニアリング

- よいデータなしでは、性能改善はうまくできません。
- あて推量を避けてください – 常にT4を動かしてください。
- 必要ならば、T4「expert モード」とSDA拡張を使ってください。
- 今後も新しいツールとサービスが期待されます。

- より速いマシンはより迅速に待ちます！
- 速くしようとせず、遅くなるのを防いでください。
- 最も速いIOは、実行しないIOです。
- 最も速いコードは、実行しないコードです。
- アイドル・ループは単なるアイドルではありません。

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

i(n)サーバーでVSI OpenVMS利用時のヒント

- 利用していないデバイスを無効にしてください (SYSMAN)。
- メモリーインターリーブ設定(memconfig)を試してください。
- XFC等を置くために、メモリー予約を使ってください。
- アプリケーションの中に多くの共用メモリーリージョンがある場合、RADメモリーレイアウトを試してください。
- fastpath CPUをfastpath有能なデバイスに設定してください。
- TCPIP + LCKMGRの専用のCPU設定を試してください。
- 小さな同時にアクティブなプロセスがたくさんある場合、ハイパースレッド(cpuconfig)を試してください。

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

パート5

1. なぜVSI OpenVMSに移行するのか
2. VSI OpenVMSに移行することの利点
3. VSI OpenVMS on Alpha
4. VSI OpenVMS on HPE Integrity Servers
- 5. VSI OpenVMSへの移行 – その内容は？**
6. 移行 - デザインと計画
7. VSI OpenVMSへの移行 - 次のステップ

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

どのような移行を意図していますか？

- 同じサーバーとインフラストラクチャーのハードウェアに留まる
 - オペレーティングシステムと製品のアップデート
- 同じアーキテクチャで、ハードウェアをアップデート
 - ハードウェアの変更(サーバー、インフラストラクチャー)
 - オペレーティングシステムと製品のアップデート
- 新しいアーキテクチャに移行
 - ハードウェアの変更(サーバー、インフラストラクチャー)
 - オペレーティングシステムと製品を移行、アップデート
 - アプリケーションを移行

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

同じサーバーとインフラストラクチャーのハードウェア

- オペレーティングシステムと製品のアップデート
- 新しいライセンス、新しいキット
- システムディスクをアップグレードするか、新しくシステムディスクを作成しますか？
- 新しい機能を利用しますか？
- 構成に変更がありますか？ 例:新しいTCP/IP
- 他に影響があるものは？
- 他のオペレーティングシステムアップデートでも同様のプロセス

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

同じアーキテクチャ、ハードウェアのアップデート

- ハードウェアのアップデート、オペレーティングシステム、および製品のアップデート
- 新しいハードウェア、新しいライセンス、新しいキット
- システムディスクをアップグレードするか、新しくシステムディスクを作成しますか？
- 新しい機能を利用しますか？
- 構成に変更がありますか？ 例:新しいTCP/IP
- 他に影響があるものは？
- 他のハードウェア置換でも同様のプロセス

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

新しいアーキテクチャ、ハードウェア変更

- アーキテクチャの変更、オペレーティングシステム、および製品のアップデート
- アプリケーション移行、新しいハードウェア、新しいライセンス、新しいキット
- 新システムディスクを構築し、データを移動してください。
- アーキテクチャの違いを理解してください。
- 最初に古いアーキテクチャ上で最新バージョンに更新してください。
- 新しい機能を利用しますか？
- 他に影響があるものは？
- 他のアーキテクチャ移行プロジェクトでも同様のプロセス

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

ディスクレイアウト - システム、共有、データ

- データからシステムを分離してください：
 - サイトごと、ノードごとのシステムディスク
 - クラスタ共通のディスク - UAF、キューなど
 - クラスタ共通のアプリケーションとデータのディスク
- 短期間の切り替えのための代替のシステムディスク：
 - ディスクアレイでのディスクの複製
 - テストのためにシステム全体のオフラインコピーが可能
- コンフィギュレーションとアーキテクチャに依存しないDCLを書いてください。

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

CPUの利用

- 可能なかぎり、ワークフローに並列処理を導入してください。
- ハイパースレッド(IA64、 x86-64) - ワークロードに依存
- Fastpath IOデバイスと分散の割込みの扱い
- ロックマネジャー(ローカルロック)のための専用のCPU
- 圧縮と暗号化、例:SCS圧縮、BACKUP圧縮、および暗号化(ハードウェア・ベース)
- QUANTUM、ワークロードに依存 - 多くのSYSGENパラメータデフォルトはV8.2で変わりました。
- パワー・マネジメント

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

メモリーの利用

- すべてのメモリーの利用(XFC、レジデント・イメージ、DECramなど)
- ワーキング・セット・サイズを再検討してください - WSMAXとプロセスのクォータ(WSDEF / WSQUO / WSEXT / PGFLQUO)
- RMSグローバルバッファを使ってください。
- RMSシステムのデフォルトを再検討してください。
- XFC - 「二重キャッシュ」に用心してください。
- GHリージョンは少数のページ・テーブル・エントリでたくさんのメモリーをマップします。
- INSTALL/RESIDENTとGHリージョンサイズ
- 64ビットのP2スペースとメモリー予約
- DECramとディスクへのHBVS

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

ストレージI/Oの利用

- FC帯域幅は重要です - ストレージ帯域幅を他と共有していますか？なぜ？
- 回転遅延は重要ではありません。
- SANゾーニング、優先パス、およびサイト間リンク
- アレイキャッシュサイズとボリューム特性
- HBVS(シャドウイング)：
 - 本当に必要な物のみをシャドウしてください。
 - 多くのシャドウセットを持つと、リカバリー時の急速なコピーがコントロールできます。
 - ミニ・コピーとミニ・マージのポリシー

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

ストレージインフラストラクチャーの利用

- シャドウとアレイの両方の機能を使ってください。
- アレイベース複製 – 同期および非同期：
 - 長距離のオフサイトコピー
 - 仮想テープのオフサイトコピー
- スナップ、クローン：
 - スナップを長い間おかないでください。
- アレイコントローラオーバーヘッドを最小化：
 - ミラーリングとストライピングのために十分なスピンドルを利用
- 明示的なパス選択でロードをアレイポートとコントローラに分散

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

ネットワークI/Oの利用

- アダプターごとにプロトコルを分散
- 802.1Q VLANタグ
- QoS(サービスの質)コントロール
- ジャンボ・フレーム(LAN_FLAGSとLANCP)
- SCSウィンドウサイズ
- TCPIP複数NICとIPアドレス・スキーム・デザイン
- TCPIP - keepalive、delack、MTUサイズ
- SCSとDECnet Plusシングルレール(LAN フェイルオーバ) v 複数レール
(ロードバランス)。
- 長距離ネットワークによる遅延を理解。
- ネットワークIOのためのRMSブロックサイズ

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

パート6

1. なぜVSI OpenVMSに移行するのか
2. VSI OpenVMSに移行することの利点
3. VSI OpenVMS on Alpha
4. VSI OpenVMS on HPE Integrity Servers
5. VSI OpenVMSへの移行 – その内容は？
- 6. 移行 - デザインと計画**
7. VSI OpenVMSへの移行 - 次のステップ

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

デザインのゴール

- 最小の中断での移行
- 操作上の安全 - エラーのリスクを最小化
- ロギングと情報収集の取り入れ
- 変化する要件(性能、スケーラビリティ)への適合
- 複雑さとリスクの最小化
- データの完全性の保存と古いデータへのアクセス

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 ?

移行のデザイン

- 移行はしばしば、計画とデザインの主要な問題です。
- 休止期間はどれほどの間持てるか？
- 休止期間はいつが可能か？
- 誰に協力してもらう必要があるか？
- 順調かどうかはどのように分かるか？

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 ?

移行の計画

- どのように移行の影響を最小化できるか？
- 机上での誤りははるかに安い
- 影響のある人すべてが参加する必要
- プレッシャーの下で行う作業をどう最小化するか
- 事前にできる事？
- 後でできる事？

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 !

移行 - 誤りのリスクの最小化

- 事前にすべてを文書化：
 - 開始、終了の状態
 - 活動
 - タイムライン
 - 責任
 - 大まかなタイミング
 - 予期される結果
 - 決定を下すポイント
 - 撤回行動
- その場その場で行わない！

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 ?

テスト - システムは決して完全でない。

- 移行後に問題が出た場合、それは以前に存在していたか？それとも、移行に起因する問題か？
- どの問題(もしあれば)を先に解決する必要があるか？
- 参照データセットと参照システムがあるか？
- 「並列実行」して、結果を比較できるか？
- 合理的にできるベストの対応は何か？

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**

パート7

1. なぜVSI OpenVMSに移行するのか
2. VSI OpenVMSに移行することの利点
3. VSI OpenVMS on Alpha
4. VSI OpenVMS on HPE Integrity Servers
5. VSI OpenVMSへの移行 – その内容は？
6. 移行 - デザインと計画
7. VSI OpenVMSへの移行 - 次のステップ

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”

アップグレードと統合

- 発展は革命より低リスク
- 今VSI OpenVMSに移行することは、x86-64が実用製品になった時に移行するのに役立ちます。
- 現存する投資を有効に維持してください。
- コードの互換性とデータのポータビリティ
- 完全入れ替えではなく、時間をかけた段階的なアプローチを

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

開始は容易

- ライセンス、サポート、およびサービスはVSIまたはパートナーから
- VSIオペレーティングシステムとレイヤード製品 - アクティブな開発とリリースのストリーム
- OpenVMSパッチと改良、特にセキュリティ
- ISVレイヤード製品
- HPEサーバーとマルチベンダーのインフラストラクチャー

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

Thank you

Oracle Rdb Forum - 2016年

VSI OpenVMSへの移行の計画

Colin Butcher CEng FBCS CITP

Technical director、XDelta Limited

www.xdelta.co.uk