

Porting OpenVMS to the Intel® Itanium™ Processor Family

Helmut Ammer
CSSC München

1C02

25. DECUS München Symposium Bonn 2002

Outline

- Introduction
- Strategy and plans
- Porting the base operating system
- Moving your applications to Itanium™ processor family

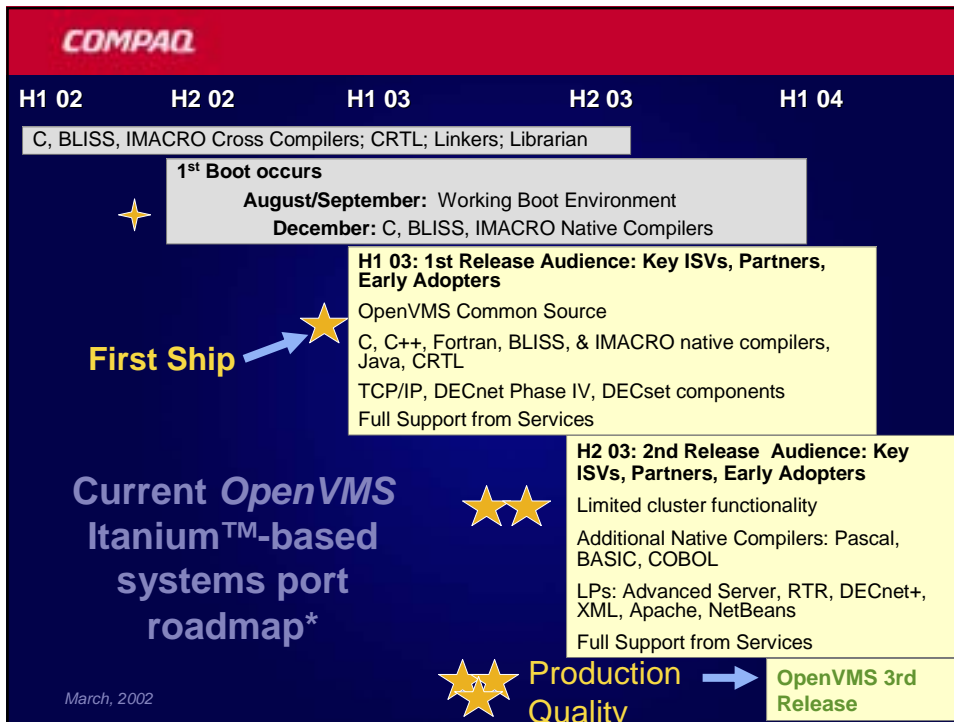
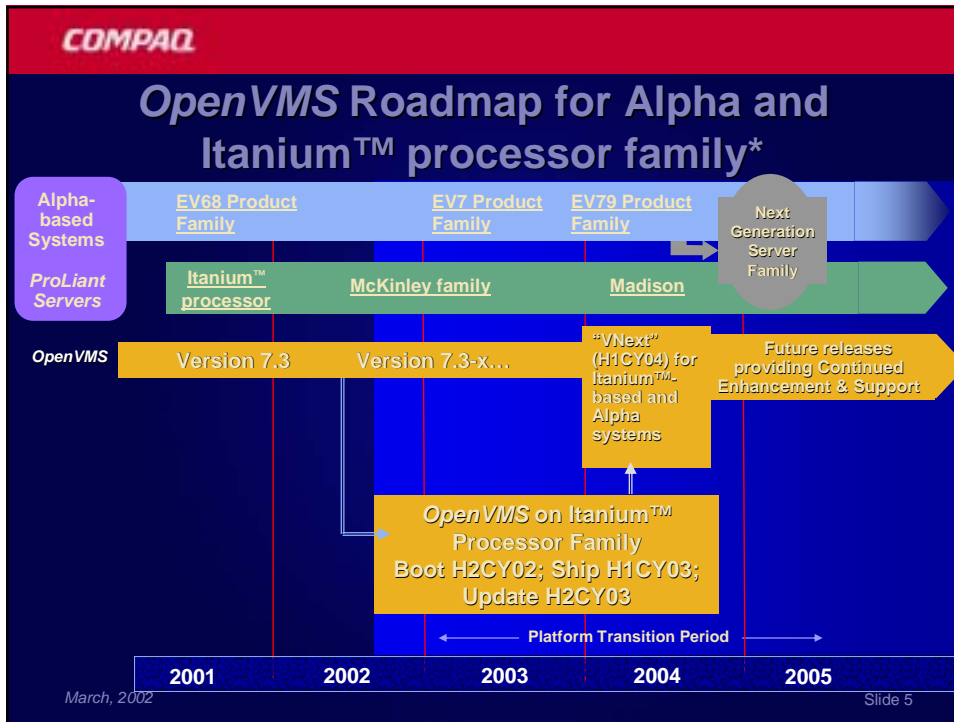
Terminology

- **Itanium™ Architecture** - Intel® processor architecture
- **Itanium™ processor family** - Family of Intel's 64 bit microprocessors
- **Itanium™ processor** - Specific member of the family

Strategic Compaq OpenVMS actions

- Announced the alignment of all 64-bit enterprise servers on Intel® Itanium™ processor family architecture microprocessor
- Long-term collaborative technology/marketing agreement with Intel®
- Full port of OpenVMS to new Intel architecture
 - If you're on Alpha, you're a recompile and requal "away" from being on Itanium™ processor family
- Enhanced server roadmaps for OpenVMS, NonStop™ Himalaya, Linux, UNIX, and Windows customers
 - Alpha EV6 speedups and EV7 Marvel program
 - New NonStop™ Himalaya speedups





OpenVMS Itanium™ Product Porting Rollout

2003

OpenVMS Core: Clusters - Limited configurations, Volume Shadowing, DECwindows Motif, Monitor Utility

Networks: DECnet Phase IV, DECnet-Plus, TCP/IP, Advanced Server, DFS

Development Tools: LSE, CMS, MMS, DTM, Enterprise Toolkit

e-business: XML, CSWS (Apache), CSWB (Mozilla), NetBeans, RTR, COM

Compilers: C, C++, COBOL, BLISS, ADA, Java, Fortran, IMACRO, CRTL, Pascal, BASIC

2004 additions

OpenVMS Core: Expanding Clusters (multiple phases), DECram, RMS Journaling, GKS, Phigs, Media Mgmt Svcs, Galaxy

Networks: X.25

Development Tools: PCA

e-business: BridgeWorks, Enterprise Directory

Mail & Messaging: MAILbus 400, IMAP4 Server

Service Tools: WEBES

Middleware: DCE, ACMS, DECforms, FMS, DECforms and TP Web Connectors, TP Desktop Connector, Datatrieve

System Management: Availability Mgr, Web Agents, TDC, OMS, ECP Tools, GCU/GCM

Storage Products: ABS, SLS, SW RAID, DFO

Many others...

Porting the base operating system

Porting OpenVMS to Itanium™ Processor Family

Q: What version of OpenVMS are you porting?

A: We are adding Itanium™ processor family support to the Alpha code base to create a single set of sources. We will build both Alpha and Itanium™ processor family releases from the same sources. A future OpenVMS release will support both platforms. However, its OpenVMS version number cannot be predicted at this time. The OpenVMS release that supports the Itanium™ processor family platforms will reflect on-going OpenVMS development work as it is available.

Alpha-to-Itanium™ vs. VAX-to-Alpha

- VAX-to-Alpha: huge volume of coding work
 - AMACRO and 1100+ VAX MACRO-32 modules
 - 32b to 64b
 - data alignment
 - atomicity
 - multiple, out-of-order execution streams
- Alpha-to-Itanium™ base
 - much less coding but more complex

Four Processor Modes

- High 0 - kernel
- 1 - exec
- 2 - supervisor
- Low 3 - user

Identical to VAX

Big Challenges for the Base OS...

...but not insurmountable

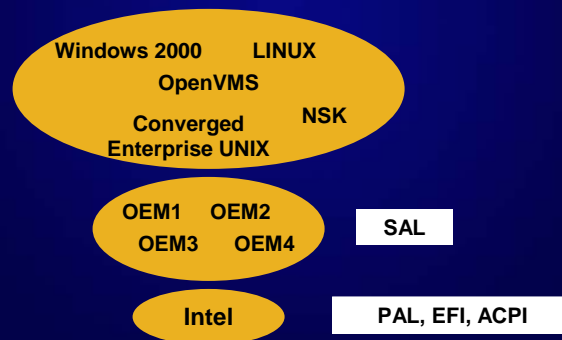
- No Alpha Console
- No Alpha PALcode
- Different primitives in the CPU
 - Register Conventions
 - Exception Handling
 - Interrupts

Console Functions

- Booting
- Special Run Time Services
- Power Management

Console Architecture

- Processor Abstraction Layer (PAL)
- System Abstraction Layer (SAL)
- Extensible Firmware Interface (EFI)
- Advanced Configuration and Power Interface (ACPI)



Console: Booting on EFI

- Requires “OS loader” to be in a FAT32 file partition
- *OpenVMS* implementation
 - A PC-style Master Boot Record overlays the ODS-2 “boot block”
 - The MBR contains a pointer to an ODS-2 container file which acts as the FAT32 partition
 - Our “OS loader” loads IPB
 - VMS_LOADER and IPB do on the Itanium™ processor family what the Alpha console and APB do on Alpha in preparing the system for SYSBOOT

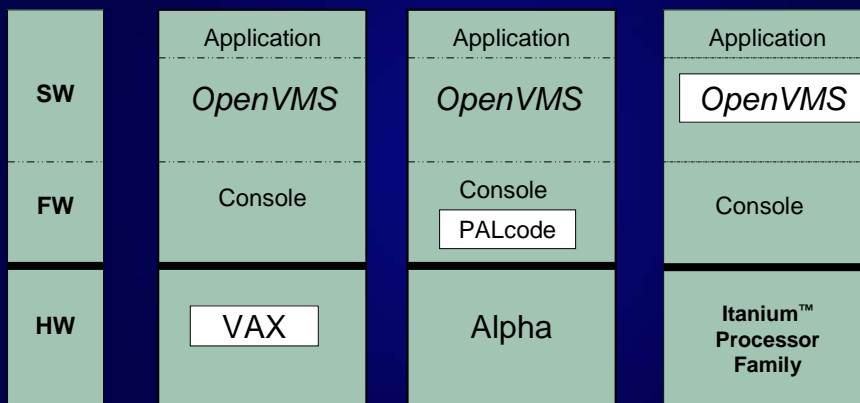
Console: Special Run Time Services

- Operator Terminal I/O - *OpenVMS* will do it
- CPU Management, e.g. \$ STOP CPU
- Partitioning
 - Need “ownership of devices” info
 - Need “switch of ownership” capabilities
 - Need “notification of configuration changes”
 - Soft partitions (Galaxy) will be very challenging
- Crash Dump
 - Driver
 - Secondary CPUs halted to ‘console mode’

Console: Power Management through ACPI

- ACPI provides initial configuration data
- Defined interface to control power
- Current *OpenVMS* RAS support
 - ACPI provides a lot, but not all we need
 - Additions to *OpenVMS*
 - Create configuration tree from ACPI data
 - Probable run-time functions in EFI applications
- OS can provide its own Boot Manager interface

It's All in the Software



Privileged Architecture Library (PALcode)

- Alpha PALcode execution environment
 - Complete control of machine state
 - Interrupts disabled
 - I-stream mapping disabled
- A CALL_PAL is very expensive
- Not all functions need such complete control

PALcode Functions

- Instructions
 - Complex sequencing and atomic operation
 - VAX interlocked instructions
 - Privileged instructions
- Translation buffer management
- Interrupt and exception setup and dispatching
- Synchronization primitives

CALL_PAL Examples

- CHMK *
- REI *
- SWPCTX
- CFLUSH
- MFPR / MTPR *
- Interlocked Queue Instructions *

* = VAX instruction

Remove from head of queue, interlocked

- **VAX:** microcoded instruction REMQHI
- **Alpha:** CALL_PAL REMQHIL
- **Itanium™ processor family:**
OpenVMS system service SYS\$PAL_REMQHIL

IPL / ASTs / Software Interrupts

- 0 - 31 IPLs (but we only define 14 of them...
(2,3,4,5,6,7,8,9,10,11,15,21,22,31)
 - Map 16-31 directly onto a 16-bit interrupt register
 - Use another register for software levels 0-15
- *OpenVMS* controls IPL and mode changes and delivers ASTs and software interrupts
- Alpha “registers” (e.g. ASTSR, IPL, ...) become
 - Itanium™ registers, or
 - CPU database cells or HWPCB cells

Memory Management

- Page Size
- Page Protection
- Virtual Address Space
- PTE Format

Page Size

- Page size will be 8KB, initially
- 3 levels of page tables
- Granularity hints (GH)
 - concept is the same as Alpha
 - size options are slightly different
 - GH regions (a.k.a. huge pages) will be handled by the *OpenVMS* TLB miss handler
- Address translation
 - Page tables
 - Virtual Hash Page Table

March, 2002

Slide 25

Page Protection

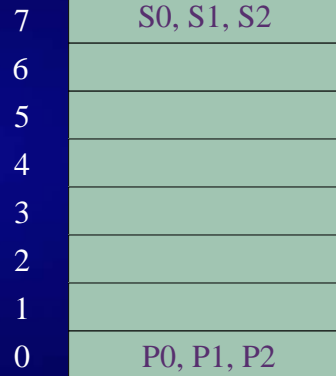
- Itanium™ processor family access rights and privilege levels can do all common VAX and Alpha page protections except
 - User Read Exec Write (UREW) - considering using User Read Super Write (URSW)
 - User Read Super Write (URSW) and Super Read Exec Write (SREW) won't allow execute access. (Privileged code must change: write code, then set page read only.)
- New possibilities
 - 'no execute' pages
 - protection key registers for additional protection

March, 2002

Slide 26

Virtual Address Space

- address space will be 8TB in size initially
- 32-bit System Page Table (SPT) window will still be created in S1 space for 32-bit device driver code
- Each Itanium™ architecture region will have its own page table space
- P0, P1, S0, S1 will be 32-bit; P2 and S2 will be 64-bit



Synchronization Techniques

- Requirement: to read/write a shared location in a single atomic operation
- Example *OpenVMS* Uses:
 - Spinlock
 - MUTEX
 - Semaphore
 - Queue instructions
- Alpha: CALL_PAL, LDxL / STxC and MB
- Itanium™ processor family: FETCHADDx, CMPXCHGx, XCHGx, MF, and acquire/release semantics on loads and stores

Process Context Switching

- More registers - 128 general, 128 floating... but
 - 2 FEN bits distinguish 32 registers vs. up to 128 registers in use
 - register stack engine knows the general registers to save
- 2 Stacks
 - Memory stack - move a pointer
 - RSE backing store
 - synchronous register spill
 - fill happens in the background

March, 2002

Slide 29

Binary Translator

- Will translate Alpha OpenVMS binary images and libraries linked under all VMS versions from 6.2 to current version
- Will translate a VESTed image that was translated by DECmigrate from a VAX binary image
- Restrictions: Alpha binary code
 - Only user- mode apps
 - Non privileged instruction
 - No self-modifying code
 - No sys. Memory space reference
 - No user-written system services
 - No applications written in PL/1

March, 2002

Slide 30

So, what's different? (1 of 3)

- Calling Standard
 - Intel calling standard with OpenVMS modifications
 - Still being designed
 - All Compaq provided tools will “know” about these changes
 - Your code that “knows” about the standard may have to change
 - Most code not affected

So, what's different? (2 of 3)

- Object file format
 - ELF/DWARF industry standard
 - ELF - Executable and Linkable Format, Itanium™ Architecture object code, images, etc.
 - DWARF - Debugging and traceback information (embedded in ELF).
 - All Compaq provided tools will “know” about these changes
 - User written code that “knows” the object file format may have to change
 - Image header “tricks” may no longer work (Flip a bit to turn on/off debugging)

So, what's different? (3 of 3)

- Floating point data types
 - Itanium™ processor family supports IEEE float only
 - All compilers that currently support F, D, G, S, T, and X (S and T are native IEEE formats) should continue to do so on Itanium™ processor family
 - Document forthcoming with details
 - Tell us what you need

What code will I need to change?

- Architecture Specific code
 - Assembler code
- Build command files
 - `-$ if .not. Alpha ! Assumes VAX`
- C code
 - `#ifndef (alpha) // Assumes VAX`
- Alpha specific compiler built-ins may have to be recoded

What can I do today?

- Examine your code for known differences and architecture dependencies
- Review the OpenVMS layered product plans
http://www.compaq.com/hps/ipf-enterprise/ovms_plans.html
- Review the OpenVMS Partner plans
http://www.compaq.com/hps/ipf-enterprise/partner_quotes.html
- Tell us now what concerns you about moving to the Itanium™ processor family
 - Are there tools I haven't mentioned?

For more information

- November WEB cast
<http://www.presentationselect.com/cpq-alpha-itanium/library.asp>
- Compaq and Intel Initiative
<http://www.compaq.com/hps/ipf-enterprise/>
- Itanium™ family architecture:
<http://developer.intel.com/design/itanium/archSysSoftware/>
- Itanium™ family processor hardware:
<http://developer.intel.com/design/itanium/manuals.htm>
- Software manuals:
http://developer.intel.com/design/itanium/arch_spec.htm

