OpenVMS V8.2 “bare metal” install on Integrity Servers (RX2600)

Colin Butcher, XDelta Limited

Session goal

A “guided tour” of setting up a HP Integrity Server system (RX2600) and the installation of OpenVMS I64 V8.2.

This seminar is unlikely to answer all of your questions, so please be prepared to contribute and share your knowledge.

Topics

- Hardware platform differences (VAX, Alpha, Integrity)
- Introduction to the RX2600
- Console subsystems and the boot process
- Firmware updates
- OpenVMS installation on Alpha and Integrity
- Creating OpenVMS bootable optical media
- Licensing overview
- Mixed architecture clustering overview
- V8.2 new features overview
- Porting considerations

Useful aphorisms

- **Occam's Razor:**
  "Pluralitas non est ponenda sine neccesitate"
  "Entities should not be multiplied unnecessarily"
  "Keep it as simple as possible"

- **Hanlon's Razor:**
  "Never attribute to malice that which can be adequately explained by stupidity"

- **Colin's Caveat:**
  "Allow for failures, success is one of many possible outcomes"

OpenVMS and hardware naming

VAX and Alpha are used interchangeably to refer to system architectures, processors / microprocessors and platforms

- **VAX VMS was used up to V5.5-2H4**
- **OpenVMS VAX and OpenVMS Alpha are used from V6.0 onwards**

Intel implementations are different:

- **IA-64 is the Intel architecture**
- **Itanium is the Intel microprocessor family**
- **Integrity is the HP platform**
- **OpenVMS I64 is the HP operating system name**

VMS / OpenVMS Family Tree

- **V1.x … V3.x** **[included RSX11 compatibility mode]**
- **V4.x (V4.7A)** **[clustering started here]**
- **V5.x (V5.5-2H4)**
- **V6.x (V6.2-1H3)** **[Alpha started here]**
- **V7.x (V7.3-2)**
- **V8.x (V8.2)** **[Integrity started here]**

Current versions of OpenVMS:

- **V7.3 (VAX)**
- **V8.2 (Alpha & Integrity)**

OpenVMS is OpenVMS, regardless of the hardware platform

VAX - hardware summary

- **32bit CISC. Designed when memory was expensive, so instruction set tries to save space where possible (variable length instructions etc.)**
- **Includes specific instructions to assist when writing operating system software, such as providing guaranteed synchronisation of access to data structures**
- **Early VAXes (VAX-11) included PDP11 instructions. Later VAXes use PDP11 software emulation**
- **Support a wide range of IO devices and bus structures (Unibus, Q-bus, BI bus, XMI bus)**
- **Introduced SMP machines (up to 6 way)**

Alpha - hardware summary

- 64bit RISC, Load/Store architecture
- Best performance achieved with data aligned on 64bit boundaries (“natural alignment”)
- Needs synchronisation issues to be carefully considered and coded. What was a single “atomic instruction” on VAX will be multiple instructions on Alpha.
- “VAX like” console commands (SRM console)
- Predominantly PCI bus based
- Introduced partitioning (Galaxy) and NUMA
- Extended SMP machines to 64 way (Marvel GS1280), with consequent performance enhancements to OpenVMS

Integrity - hardware summary (1)

- 64bit EPIC. Designed when memory is plentiful and cheap
- Best performance achieved with data aligned on 64bit boundaries (“natural alignment”) –more so than on Alpha
- EPIC architecture relies on compiler to generate an efficient instruction and data flow through the CPU
- Instructions are packaged in “bundles” of up to three instructions per bundle – which are then processed entirely in parallel by the CPU
- Lots of register sets within the CPU
- O.S. neutral – not designed for a specific O.S.

Integrity - hardware summary (2)

- Needs synchronisation issues to be carefully considered and coded, eg: spinlocks
- No equivalent of Alpha PALcode
- No “VAX like” or “Alpha like” console, multiple consoles:
  - Management Processor (MP) with LAN interface
  - Baseboard Management Controller (BMC)
  - Uses Extensible Firmware Interface (EFI) Uses ACPI for device detection
- PCI bus based (3.3volt only)
- Memory page size is variable (currently 8192 bytes)
- IEEE floating point only in hardware

Demonstration hardware

- **RX2600:**
  - **OpenVMS I64 V8.2**
  - **DECnet-Plus V8.2**
  - **TCP/IP V5.5**
  - **DECwindows/Motif V1.5**

- **AlphaServer DS10L:**
  - **OpenVMS Alpha V8.2**
  - **DECnet-Plus V8.2**
  - **TCP/IP V5.5**
  - **DECwindows/Motif V1.5**

RX2600 - hardware overview

- Beware disc lock inside cover
- Discs (top down) are:
  - Slot 2 SCSI ID = 2, Bus = 1 (DKB200:)
  - Slot 1 SCSI ID = 1, Bus = 0 (DKA100:)
  - Slot 0 SCSI ID = 0, Bus = 0 (DKA000:)
- SCSI bus 1 (second bus) also presented on back of machine
- Two ethernet adapters (Gigabit + 10/100)
- Can be single or dual processor
- Can be single or dual power supply
- Will power up with cover removed

VAX - Console firmware

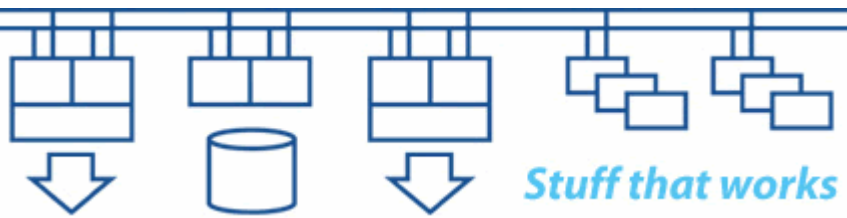
- Serial console port (pseudo-serial on VAXstations)
- Power-up self tests
- Device detection
- Device naming convention matches OpenVMS device naming convention
- Device configuration (SCSI ID, DSSI ID etc.)
- System configuration (language, auto boot, etc.)
- Data passed in CPU registers (R5 used for boot flags) to boot loader
- Console command set is machine specific

Alpha – SRM console firmware

- Serial and graphics consoles
- Power-up self tests
- Device detection
- Device naming convention matches OpenVMS device naming convention
- Environment variables
  - Device configuration (SCSI ID, DSSI ID etc.)
  - System configuration (language, auto boot, etc.)
- Data passed to boot loader, eg: boot –flags 0,0 <device>
- Console command set is machine specific

Integrity - Management Processor

- **MP console for "Management Processor"**
  - **Runs with base box level power, even with system off**
  - **Local (9pin serial), remote (9pin serial modem) and network connectivity (telnet and web console)**
  - **Console configuration (terminal type, etc.)**
  - **<ctrl-H> for delete, <ctrl-B> to get back to MP**
  - **Network configuration (hostname, IP address, etc.)**
  - **Multiple console sessions (one writer, many readers)**
  - **Provides ability to copy files over network (firmware updates via FTP etc.)**



Baseboard Management Controller

- **BMC console for "Baseboard Management Controller"**
  - **Runs when main board powered up**
  - **Local connectivity (9 pin serial) and logical connectivity (MP console sessions)**
  - **Power-up self tests**
  - **Device detection**
  - **Console configuration (terminal type, etc.)**
  - **<ctrl-H> for delete, <ctrl-B> to return to MP**
  - **No graphics console**

Extensible Firmware Interface

- **EFI (extensible firmware interface):**
  - **Mini operating system**
  - **FAT formatted file system (FAT12, FAT16, FAT32), OpenVMS currently presents a FAT16 partition to EFI**
  - **Boot menu and defaults**
  - **Environment variables (VMS_FLAGS etc.)**
  - **VMS_LOADER.EFI - finds and loads IPB.EXE**
  - **Data passed to boot loader, eg: efi> vms_loader –fl 0,0**
- **IPB.EXE (boot loader) understands the OpenVMS file structure, EFI does not**

VAX – OpenVMS boot process

- **Console firmware**
- **Loader finds VMB (method varies with platform), eg:**
  - **On system disc via boot block**
  - **Console ROM or NVRAM**
- **Boot flags passed in registers**
- **Reads executive into memory**
- **Reads system parameters (SYSBOOT> if flags set) and initialises fixed data structures**
- **Passes control to executive**

Alpha – OpenVMS boot process

- **Console firmware**
- **Loader finds APB on system disc via boot block**
- **Boot flags passed in registers**
- **Reads executive into memory**
- **Reads system parameters (SYSBOOT> if flags set) and initialises fixed data structures**
- **Passes control to executive**

Integrity – OpenVMS boot process

- **EFI boot loader (vms_loader.efi) from FAT partition (hidden as a container file in the system disc file structure)**
- **Boot flags passed through environment variables**
- **Reads executive into memory**
- **Reads system parameters (SYSBOOT> if flags set) and initialises fixed data structures**
- **Passes control to executive**
- **Structures created with SET BOOTBLOCK command**
- **2048 byte boot block for IDE/ATAPI optical media**
- **512 byte boot block for SCSI optical media**

VAX - device detection

- **Console firmware**
- **Devices appear as a set of CSRs (Control and Status Registers) in physical memory - the IO space**
- **Devices have Interrupt Vectors which connect a device interrupt request to the device driver Interrupt Service Routine**
- **CSR addresses and contents indicate device type**
- **SYSGEN AUTO ALL will scan IO space to find devices and set up OpenVMS device drivers to communicate with the hardware**

Alpha - device detection

- **Console firmware**
- **Devices appear as a set of CSRs (Control and Status Registers) in physical memory - the IO space**
- **Devices have Interrupt Vectors which connect a device interrupt request to the device driver Interrupt Service Routine**
- **CSR addresses and contents indicate device type**
- **SYSMAN IO AUTO will scan IO space to find devices and set up OpenVMS device drivers to communicate with the hardware**

Integrity - device detection

- **Itanium Processor Family architecture uses ACPI (Advanced Configuration and Power Interface) for device detection by firmware**
- **Devices appear as a set of CSRs (Control and Status Registers) in physical memory - the IO space**
- **Devices have Interrupt Vectors which connect a device interrupt request to the device driver Interrupt Service Routine. Device data obtained from ACPI data.**
- **ACPI data indicate device type**
- **SYSMAN IO AUTO will query ACPI data to find devices and set up OpenVMS device drivers to communicate with the hardware**

This should be familiar territory…

- **OpenVMS VAX V7.3**
- **OpenVMS Alpha V7.3-2**

- **OpenVMS Alpha V8.2 and OpenVMS I64 V8.2 share a common code base**

OpenVMS Alpha V8.2 prerequisites

- **CDs:**
  - **Firmware Update (PALcode etc.) – V6.9 firmware CD is current**
  - **Operating System (O.S.)**
  - **Layered Products (L.P.)**
- **Licence PAKs**
- **Installation requirements documentation (O.S. and L.P.)**
- **Hardware configuration information**
- **Console setup (SET BOOTDEF_DEV DKA0 etc.)**

OpenVMS Alpha V8.2 components

- **OpenVMS Alpha V8.2 CD kit**
  - **DECnet-Plus V8.2**
  - **TCP/IP Services V5.5**
  - **Motif V1.5**
- **Licence data (PAKs)**
- **Base L.P. kits:**
  - **ZIP, DFU, TSM, etc. (from Freeware CDs)**
  - **Compilers etc. (from Layered Product CDs)**
- **Patches**
- **Post install procedures**

OpenVMS Alpha V8.2 installation

- **Boot firmware update CD**
- **Install firmware updates**
- **Boot OpenVMS installation CD**
- **Select installation menu (option 1)**
- **Answer installation questions**
- **Install O.S. patches**
- **Install 'base' L.P. patches (e.g. TCP/IP)**
- **Set up system parameters ready for layered product installations (read product requirements)**

"And now for something completely different…"

OpenVMS I64 V8.2 prerequisites

- **DVDs / CDs / FTP:**
  - **Firmware Updates (MP console, BMC console etc.)**
  - **Operating System (O.S.) - DVD not CD**
  - **Layered Products (L.P.) - CD**
  - **Freeware (V7 is set of 3 CDs)**
- **Installation requirements documentation (O.S. and L.P.)**
- **Hardware configuration information**
- **Terminal emulator setup**
- **Console setup (MP console, BMC console, EFI boot options)**
- **EFI configuration (VMS_LOADER etc.)**

RX2600 console setup

- **MP console - LAN attributes (IP address etc.)**
- **BMC console – get there via “CO” at MP console level**
- **If system origin is unknown, use “reset to factory defaults”**
- **Set up standard input, output and error devices to be VT100+ (lower choice via PCI bus device):**
  - *** Acpi(HWP0002,700)/Pci(1|1)/Uart(9600 N81)/VenMsg(Vt100+)**
- **Make “EFI shell [built-in]” the default boot option during console setup, firmware updates and operating system installation**
- **Use “cold reset” after clearing boot options etc.**

RX2600 firmware updates (boot CD)

- **Download updates from Internet**
- **Read release notes and installation instructions carefully**
- **Burn bootable CD from ISO image**

- **Boot CD via EFI shell:**
  - **Shell> “fsN:” to select ATAPI DVD-ROM drive**
  - **Shell> “cd \efi\boot” to select correct directory**
  - **Shell> “bootia64” to execute efi script and boot from CD**

- **Note: Need to update MP console via E0.2.26 (FTP method only) if starting from an earlier version**

RX2600 firmware updates (FTP)

- **Download updates from Internet**
- **Read release notes and installation instructions carefully**
- **Move files to directory on FTP server (eg: /E0226) containing all files within the compressed kit**

- **Load from FTP server (eg: PC, HPUX, OpenVMS etc.) at MP console command level:**
  - **MP> “co” to get to command level**
  - **CM> “lc” to check LAN connectivity**
  - **CM> “xu” to start firmware update process**

FTP firmware update example (1)

CM>xu
Firmware Revision E.02.23 Jul 1 2003,11:10:23

This command upgrades MP firmware. All connections will be closed,
the session will be aborted and the modem connection will be dropped
immediately, web and telnet connections will be dropped upon completion.

Current Firmware Upgrade Parameters:
I - Source IP : 10.255.255.241
P - File Path : /tftp/E0226
L - Login : anonymous
W - Password : anonymous

Enter Parameter(s) to revise, Y to confirm, or [Q] to Quit: y

Cont.

FTP firmware update example (2)

-> MP firmware upgrade in progress....

-> Retrieving upgrade file using FTP.

-> Retrieved an upgrade file successfully.
Programming ROM. Percent Complete: 100.

-> Retrieving upgrade file using FTP.

-> Retrieved an upgrade file successfully.
Programming ROM. Percent Complete: 100.

-> Retrieving upgrade file using FTP.

-> Retrieved an upgrade file successfully.
Programming ROM. Percent Complete: 100.

Cont.

FTP firmware update example (3)

```
-> MP firmware upgrade complete - Web and telnet connections will
   be dropped.  MP will now reset....

HP Management Processor
Firmware Revision E.02.26 Oct 26 2003,23:12:41
(c) Copyright Hewlett-Packard Company 1999-2003.  All Rights Reserved.

*******************************************************************************
 Only default users are configured.
 Use one of the following user/password pairs to login:

 Admin/Admin
 Oper/Oper
*******************************************************************************

MP login:
```

OpenVMS I64 V8.2 components

- **OpenVMS Integrity V8.2 DVD kit**
  - **DECnet-Plus V8.2**
  - **TCP/IP Services V5.5**
  - **DECwindows/Motif V1.5**
- **OS licences are different – "Operating Environment" based.**
- **Base L.P. kits:**
  - **ZIP, DFU etc. (from Freeware CDs)**
  - **Compilers etc. (from Layered Product CDs)**
- **Patches**
- **Post install procedures**

OpenVMS I64 V8.2 install overview

- **Install firmware updates**
- **Boot OpenVMS installation DVD**
- **Select installation menu (option 1 – no upgrades from EFT)**
- **Answer installation questions**
- **Use BOOT_OPTIONS.COM to set up bootable devices**
- **Install O.S. patches**
- **Install 'base' L.P. patches (e.g. TCP/IP)**
- **Set up system parameters ready for layered product installations (read product requirements)**

OpenVMS I64 V8.2 install notes

- **Read the Release Notes and Installation Guide carefully**
- **Become familiar with the MP, BMC and EFI "consoles"**
- **Use BOOT_OPTIONS.COM (on the installation media as well as the installed system disc) to configure boot devices – this is the only way to configure fibrechannel storage as boot / dump devices. There is no WWIDMGR**
- **Set up EFI boot menu for automatic boot of target system disc – these can sometimes change when you use SET BOOTBLOCK or perform OpenVMS upgrades**
- **Manual boot via VMS_LOADER –FL <root,flag_bits>**
- **The boot flags are "sticky" until reset**

OpenVMS post-installation

Configure base system:

- **Edit / create startup files**
- **Configure network software**
- **Page / swap / dump files**
- **Set up other system components (e.g.: LAT, MOP)**
- **Set up queues and queue manager database**
- **System parameters**
- **Layered products (compilers etc.)**
- **BACK IT UP (and test restore)**

Creating bootable optical media (1)

- **Use LD driver to create container files (LD is built in to V7.3-2 and later, latent in V7.3-1, Freeware CD before that)**
- **Disable file caching for LD container files (SET FILE <container>/CACHING_ATTRIBUTE=NOCACHING)**
- **Integrity - EFI uses the native sector size of the device, so:**
  - **IDE/ATAPI optical media needs a 2048 byte boot block**
  - **SCSI optical media needs a 512 byte boot block**
- **Container size as multiple of 4 blocks (allows for 2048 byte IDE/ATAPI native sector size). Note: 2048blocks = 1Mbyte.**
- **INIT <container> /CLUSTER=4 /ERASE /NOHIGH /INDEX=BEGIN /MAXIMUM=<max_files> etc.**

Creating bootable optical media (2)

- **Create minimum bootable environment:**
  - **Use STABACKIT <LD_dev> <SYSE> on VAX**
  - **Use AXPVMS$PCSI_INSTALL_MIN.COM on Alpha**
  - **Use I64VMS$PCSI_INSTALL_MIN.COM on Integrity**
- **Use the SET BOOTBLOCK/IA64/BLOCK=2048 <LD_dev> command for Integrity target ATAPI optical media**
- **Set the SYSGEN WLKSYSDSK parameter appropriately**
- **Use CDRECORD on VAX / Alpha (see Freeware CDs), or copy container to PC (FTP in binary mode) and burn CD / DVD as an image file with 2048 byte boot sector**

OpenVMS image backup and restore

- **Backup and restore:**
  - **Image backups (/image/noalias)**
  - **Image restores:**
    - **May need to use SET BOOTBLOCK command**
    - **May need to delete / re-add boot option in EFI**
- **Can use OpenVMS Alpha V7.3-2 and later to backup and restore OpenVMS I64 system discs**
- **Can use OpenVMS Alpha V7.3-1 and later to prepare OpenVMS I64 system discs (see [ALPHA_TOOLS] directory)**

OpenVMS I64 licensing

- **Licenses use the HPUX type licensing model. They are based around the concept of “Operating Environments”.**
- **Multi-processor systems use a “soft compliance” license model – it will complain, but it won’t stop your systems**

- **MC.O.E.- Mission Critical O.E.**
- **E.O.E. - Enterprise O.E.**
- **F.O.E. - Foundation O.E.**
- **Can add “extras” as required to enhance a given O.E. level (eg: E.O.E. + HBVS)**

OpenVMS V8.2 changes overview (1)

- **Most changes are related to the port of OpenVMS to Integrity Servers**
- **Use standard mechanisms – don't "roll your own"!**

- **Image layout for Integrity Servers (ELF / DWARF)**
- **Calling standard for Integrity Servers (register usage etc.)**
- **Linker / debugger changes for Integrity Servers**
- **System services**
- **DCL lexical functions**

OpenVMS V8.2 changes overview (2)

- **Host-based mini-merge (HBMM)**
- **Internals and data structures changes (eg: SWIS)**
- **System service logging**
- **SDA extensions**
- **Performance features such as finer granularity of locks**
- **Extended lock value blocks**
- **Revised system parameter defaults**
- **DCL command line length and parse token length**
- **DCL mailboxes**

Mixed architecture clustering

- **VAX and Alpha are supported in the same cluster**
- **Alpha and Integrity are supported in the same cluster**
- **VAX and Integrity are not supported in the same cluster**

- **Be careful with mixed version clusters. Older versions of OpenVMS can restrict overall cluster performance and functionality.**
- **OpenVMS VAX V7.3 does not include full support for HBMM bitmaps, ODS5 extended file naming etc.**

OpenVMS porting considerations (1)

- **Binary translation (VAX to Alpha, Alpha to Integrity)**
- **Implicit assumptions, eg: (if .not. vax)**
- **Synchronisation and serialisation of access to data structures**
- **Performance features**
- **Platform independent interfaces (system services, lexical functions etc.)**
- **IEEE floating point format for best performance**
- **Privileged code may require additional work**

OpenVMS porting considerations (2)

- **Make a thorough audit and analysis of what you're doing at the moment**
- **Consider taking advantage of new features and new ways to do things**
- **There may be no direct equivalents**
- **Direct 'bug for bug' port?**
- **Opportunity to re-implement application?**
- **Plan for expansion & growth**
- **Are you going to have to continue to support non-migrated systems as well?**

Starting on VAX

- **Why move?**
- **Hardware constraints, eg: IO devices**
- **Software constraints, eg: tied to VAX hardware**
- **Operating system version constraints**
- **Layered product version constraints**
- **Network constraints, eg: X.25**
- **Performance constraints (interrupt latency)**

VAX to Integrity

- **On VAX – move to current version of Operating System and Layered Products**
- **Tidy up application code (eg: synchronisation & serialisation [lock manager calls], hard-coded values [512 bytes], run-time library calls, system service calls, implicit assumptions, etc. …)**
- **Move from VAX current to Alpha equivalent (eg: OpenVMS VAX V7.3 to OpenVMS Alpha V7.3-2)**
- **Use IEEE floating point format for best performance**
- **Move from Alpha to Integrity (eg: OpenVMS Alpha V7.3-2 to OpenVMS I64 V8.2)**

Starting on Alpha

- **Why move?**
- **Hardware constraints, eg: IO devices**
- **Software constraints, eg: tied to Alpha hardware**
- **Operating system version constraints**
- **Layered product version constraints**
- **Performance constraints (big multiprocessor systems at the moment until new Itanium based systems appear after OpenVMS I64 V8.2 and later)**

Alpha to Integrity

- **On Alpha – move to current version of Operating System and Layered Products**
- **Tidy up application code (eg: synchronisation & serialisation [lock manager calls], hard-coded values [512 bytes, 8192 bytes], run-time library calls, system service calls, implicit assumptions, etc. …)**
- **Use IEEE floating point format for best performance**
- **Move from Alpha to Integrity (eg: OpenVMS Alpha V7.3-2 or V8.2 to OpenVMS I64 V8.2)**

Further OpenVMS information

- **OpenVMS main site** (http://www.hp.com/go/openvms)
- **Technical Journal** (http://www.hp.com/go/openvms/journal)
- **FAQ** (http://www.hp.com/go/openvms/wizard)
- **Ask The Wizard** (http://www.hp.com/go/openvms/wizard)
- **Integrity Servers** (http://www.hp.com/go/integrity)
- **Alpha Servers** (http://h18002.www1.hp.com/alphaserver/)
- **HP DSPP** (http://www.hp.com/dspp)
- **Intel Itanium 2 systems** (http://www.intel.com/design/itanium/documentation.htm)

OpenVMS Symposium ("bootcamp")

Week of June 6th 2005 at Sheraton Tara hotel, Nashua (next door to OpenVMS Engineering building ZKO)

- **An "Ambassadors" style week for non-HP staff.**
- **An intense week of learning, information gathering and meeting people. Well worth going to.**
- **Contact Sue Skonetski (susan.skonetski@hp.com).**
- **See http://www.hp.com/go/openvms/bootcamp**

Contacts

Colin Butcher, XDelta Limited

Office: +44 117 904 8209

E-mail: colin.butcher@xdelta.co.uk

Web: http://www.xdelta.co.uk