vms Software

News from VMS Software Inc. (VSI)

Rollout of V9.0 and Beyond
May 19, 2020

Clair Grant, CTO

Terry R. Holmes, CRO

Agenda

- Business Update
- Virtual Machines
- OpenVMS x86 Rollout Plan
  - Cross Tools Kit (already in place)
  - Initial V9.0 EAK Content
  - V9.0 EAK to V9.1 EAK Evolution
- V9.0 Details
- Getting to V9.2
- V9.2 – stability and performance

VSI Business Update

- Changes for OpenVMS Users
  - V8.4 Standard Support ends December 31, 2020
  - HPE i6 server sales ends December 31, 2020
  - Other Changes

- New Subscription Model
  - Ease into VSI (License / Support) bundles
  - OpEx vs. CapEx

- VSI keeps growing
  - Engineering / Developer Talent
  - Sales & Marketing Talent

- New Community License Program (Fall, 2020)

Emphasis on Virtual Machines

Since the very beginning of VMS Software Inc. we have placed heavy emphasis on running in multiple virtual machine environments.

- Many customers want this
- "We run our entire IT operations on VMs except OpenVMS, don't be different."
- Near-Term testing focus
  - Virtual Box
  - KVM
  - VMware

Virtual Machine Hypervisors

- We are not "virtualizing OpenVMS" - very misleading terminology

- We are "enhancing OpenVMS" to run in selected VM guests, just as we would for new hardware platforms

- A virtual machine is "just another platform" to OpenVMS

- Platform-specific code, if needed, is loaded during startup

- All virtual machine hypervisors are different, just as all hardware platforms are different (even in the same family)

- Example: Running on Virtual Box does not mean you run on VMware, just as running on i4 did not mean running on i6

Hardware Platform Support

- VSI is not abandoning direct HW platform support
- VSI will test a limited set of systems, for starters
  - HPE and Dell servers
  - Intel and AMD processors
- We will encourage customers to try other platforms and let us know the results
  - We assume there will be issues in most cases and will work with customers where appropriate
  - The problems will usually involve booting the system (UEFI and ACPI), once up most will run with few issues
- Bottom line: HW support will depend largely on the customers

Rollout Plan

Rollout Plan

- Cross Tools Kit
- V9.0 EAK – early rollout
  - 15-May: available for first customer - Software Concepts International (www.sciinc.com)
    - Announcement mail sent at 2:40PM
    - Customer booted and running at 4:20PM
  - 30-May: add 4 more customers
  - 26-June: 5 more
  - 31-July: 5 more
  - Etc.
  - Target: 40 users by the end of 2020
- Add native tools (Cross Tools Kit still available for some time period)
- V9.1 EAK – available to all customers, most system components present
- V9.2 – production release, complete operating environment
- Regular updates from now until V9.2

Cross Tools Kit

- Compile / link on IA64, run on x86
  - V1 - BLISS, C, XMACRO, Linker and associated tools
  - V2 - Updates plus FORTRAN
  - V3 - Updates plus PASCAL
  - COBOL and BASIC in debug / testing now
- Current users providing feedback
- No Cross Tools Kit on Alpha
- VSI has a list of customers and the compilers they need early
- Offering kit to more users when more compilers become available
- No C++
- No JAVA

Initial V9.0 EAK Content

- Early V9.0 is "very rough around the edges"
- Content - intersection between what people need to be productive and what VSI can have ready in a reasonable time
- Initial support for Virtual Box, adding KVM and VMware when fully tested
- All internal operating system functions are present
- Plus - DCL, pipes, RMS, file system, batch system, SDA, AUTOGEN, MOUNT/DISMOUNT, AUTHORIZE, SYSGEN, SYSMAN, BACKUP, DELTA, AUDIT_SERVER, ACCOUNTING, OPCOM, MONITOR, and many more.
- No Layered Products

V9.0 EAK to V9.1 EAK Evolution

- Series of updates to V9.0
  - Each kit will be a complete new installation and/or appliance, not a traditional "Update Kit"
  - New kits will be made available as layered products, compilers, and open source kits are ready

- Offered to more users

- Some customers want to get started even before all of the components needed by their applications are available

- When most system components are present, it will be called V9.1 and available for all

V9.0 Details

V9.0 Proof Points

- Real Boot
  - No special execlets
  - $ DIR returns to $ prompt
  - Boot transitions to system disk
- KVM & VirtualBox booting are equivalent
- Installation from webserver and DVD
- Crash Dumps, SDA
- Conversational Boot –SHOW and SET parameters
- Create User Accounts with AUTHORIZE
- MOUNT/DISMOUNT disks
- Run Batch Jobs
- BACKUP
- Run a threaded (POSIX) application
- TCPIP: FTP, Telnet
- DEBUG - user mode symbolic debugger

What is Not in Initial V9.0 Releases?

- DEBUG (user-mode symbolic debugger)
- DECwindows
- DECnet Phase IV
- DECnet Phase V (OSI)
- Clusters
- Process dumps
- Volume Shadowing
- Reserved Memory
- SMP
- XFC
- INSTALL /RESIDENT
- Support for privileged applications, for example 1) user written device drivers or 2) code that directly calls internal system routines such as those that manage page tables
- No VAX floating point support in the V9.0 cross compilers; all fp is IEEE. For V9.1 native compilers there will be VAXfp except for C++. (NOTE: It is TBD if VAXfp will ever be in C++ for x86.)

User-Visible Differences in V9.0

User-Visible Differences

- Boot Manager

- Dump Kernel

- Microarchitectural Data Sampling (MDS) action notification during startup, e.g.
  MDS Mitigation active, variant verw(MD_CLEAR)

- Loader warning (-W-) if the image has a 'Write + Execute' psect
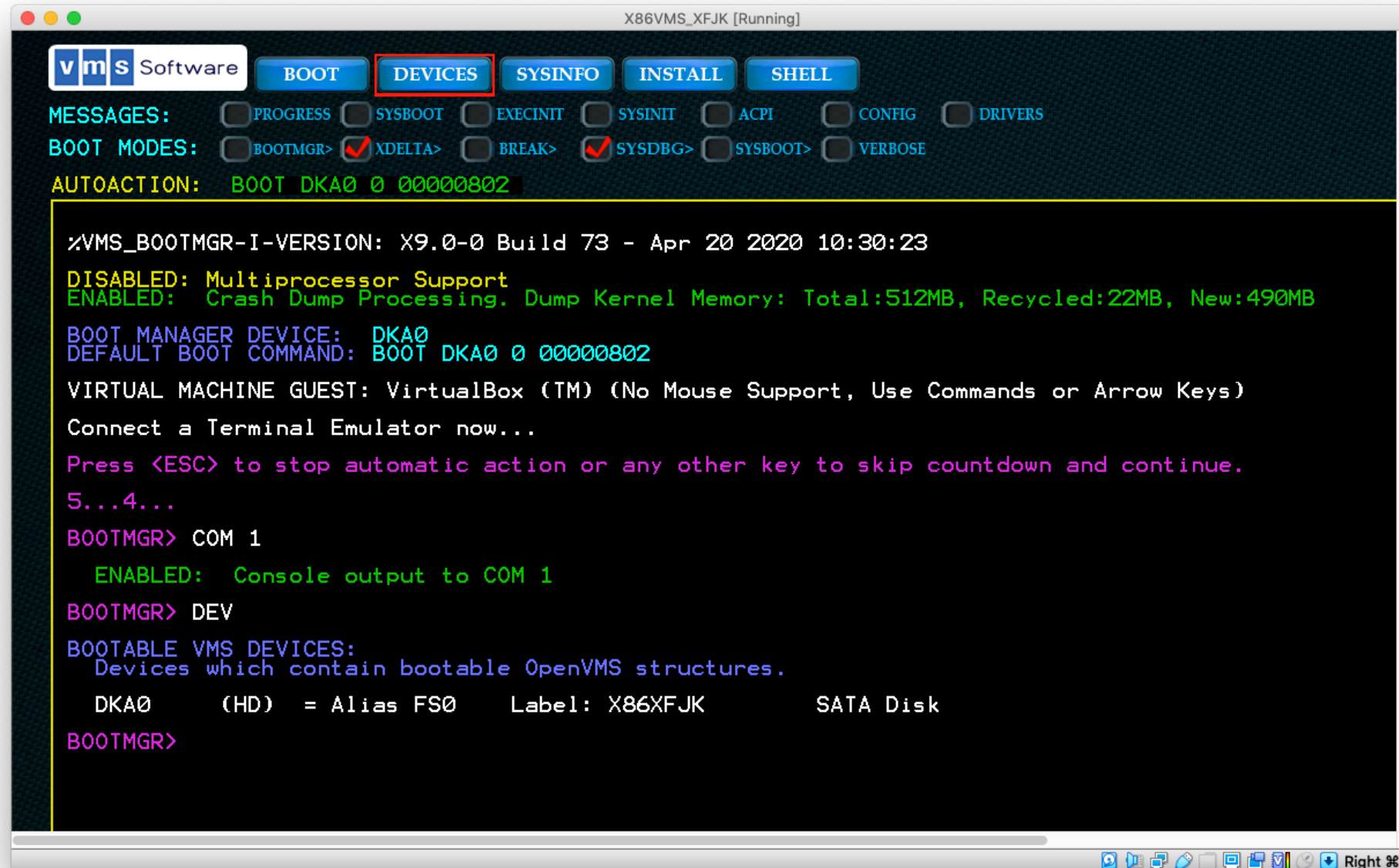
- 64-bit code addresses in image maps and output displays

API Differences in V9.0

API Differences

- The LINKER default will result in code being loaded into 64-bit space. If you find this to be a problem, use the LINK qualifier /SEGMENT=CODE=P0.
- The $LKWSET system service is problematic on x86. Its uses in an image should be replaced by a single call to LIB$LOCK_IMAGE.

Boot Manager


```
                                          X86VMS_XFJK [Running]

  VMS Software      BOOT    DEVICES    SYSINFO    INSTALL    SHELL

 MESSAGES:      ◯ PROGRESS   ◯ SYSBOOT   ◯ EXECINIT   ◯ SYSINIT   ◯ ACPI   ◯ CONFIG   ◯ DRIVERS
 BOOT MODES:    ◯ BOOTMGR>   ✓ XDELTA>   ◯ BREAK>   ✓ SYSDBG>   ◯ SYSBOOT>   ◯ VERBOSE
 AUTOACTION:    BOOT DKA0 0 00000802

  %VMS_BOOTMGR-I-VERSION: X9.0-0 Build 73 - Apr 20 2020 10:30:23

  DISABLED: Multiprocessor Support
  ENABLED:  Crash Dump Processing. Dump Kernel Memory: Total:512MB, Recycled:22MB, New:490MB

  BOOT MANAGER DEVICE:   DKA0
  DEFAULT BOOT COMMAND:  BOOT DKA0 0 00000802

  VIRTUAL MACHINE GUEST: VirtualBox (TM) (No Mouse Support, Use Commands or Arrow Keys)

  Connect a Terminal Emulator now...

  Press <ESC> to stop automatic action or any other key to skip countdown and continue.

  5...4...

  BOOTMGR> COM 1

     ENABLED:   Console output to COM 1

  BOOTMGR> DEV

  BOOTABLE VMS DEVICES:
     Devices which contain bootable OpenVMS structures.

     DKA0     (HD)  = Alias FS0    Label: X86XFJK        SATA Disk

  BOOTMGR>
```

Booting from the Memory Disk File

- x86 always boots from the Memory Disk, a file which contains the images needed to get OpenVMS through the early boot path

- Regardless of boot device – physical disk, virtual disk, or network download, a memory disk file is loaded and booting starts

- A switch to accessing the system disk occurs during the boot sequence

- There are no longer boot drivers for all possible boot devices

- The only boot driver is the memory disk driver

- Since there are no longer boot drivers……….

Dump Kernel

- When the system first starts, the Boot Manager
  - Loads and **starts** the Primary Kernel
  - Loads but **does not start** the Dump Kernel (512 MB of memory allocated)
- When a system crash occurs
  - Primary Kernel goes down
  - Boot Manager starts the Dump Kernel
  - Dump Kernel boots, gathers the needed data, writes the crash dump, and halts
  - Boot Manager loads and starts the Primary Kernel and loads the dump kernel
- Dump Kernel is a small subset of the Primary Kernel
  - Runs as a user process
  - Uses standard file system and I/O drivers

From Here to V9.2

Work to Do

- Next Step: End of May update
    - TCPIP
    - Improved quality
    - KVM
    - Standard installation procedure
- Include the "Not in Initial V9.0" components
- Compilers
    - Complete BASIC and COBOL ports and add to Cross Tools Kit within a couple months
    - LLVM (backend code generator)
        - Update to a more recent release
        - Port to OpenVMS
        - integrate into native x86 compilers
    - Port clang (C++) to OpenVMS
- Build and test layered products and open source kits
- Continuous overall system stress and regression testing

V9.2 – Stability and Performance

- Respond to V9.0 and V9.1 User Feedback
- Stability
  - Regression Tests
  - Stress Tests
  - Continuously add more users
- Performance
  - Trace Buffer Analysis
  - Performance Counters
  - Compiler optimization
  - Storage measurements
  - Network measurements

Short Demo

- Virtual Box Guest Configuration
- Boot Manager
- System startup
- OPCOM, Audit Server
- Batch Queues
- A few interesting commands
- Paging Stress Test & MONITOR
- UETP and SHOW PROCESS
- Force crash, write dump file, and reboot

Thank You

To learn more please contact us:

vmssoftware.com
info@vmssoftware.com
+1.978.451.0110