State of the Port to x86
March 2016

State of the Port to x86
March 2016

This information contains forward looking statements and is provided solely for your convenience. While the information herein is based on our current best estimates, such information is subject to change without notice.

Update Agenda

- Early Boot Path
- Code Generation
- General Topics

vms

Early Boot Path – VMS_LOADER.EFI

- Updated to UEFI V2.3 toolkit – a 12 yr. advancement
- Eliminated VMS boot drivers (now using UEFI-provided drivers)
- Debugging (on Itanium) new crash dump strategy and implementation
- Eliminated code specific to HPE firmware
- Always boot from memory disk
- Physical memory
  - Processed memory descriptors presented by firmware
  - Allocated memory for operating system use
- Making better use of ACPI interface and tables
- Creating a graphical Boot Manager
- Prototyping (on x86) machine check handling
- Created x86 virtual machine environments
  - kvm, xen, VirtualBox, VMware
  - Early testing of VMS_LOADER.EFI
  - Later use for operating system work

vms

Code Generation

- LLVM built on VMS Itanium

- GEM-to-LLVM internal representation converter
  - Rapidly increasing the number of C language test suite successes
  - Compile on VMS Itanium (DEC C frontend combined with LLVM)
  - Link and run on linux for verification

- XMACRO
  - Completed register mapping design
  - Attaching XMACRO to LLVM "assembler level" interface
  - Developing VAX to x86 instruction mapping
  - Developing VAX to x86 addressing modes mapping

- Outlined LINKER tasks

- Tracking Calling Standard additions/modifications needed

General Topics

- Completed x86 instruction decoder for use by SDA / DEBUG / XDELTA
- Completed memory management investigation
- PROBE instruction emulation rewritten to do PTE lookup
- Translation-buffer-purge rewritten to accommodate four access modes
- Prototyping (on Itanium) running in two HW access modes and two modes handled in SW

- Reviewing designs for
  - Paravirtualized Storage Driver (increased virtual machine performance)
  - Software Interrupt Services (SWIS) changes

- Evaluating
  - VAX / Alpha / IA64 conditionalized code
  - Non-standard calling sequences

vms

For more information, please contact us at:

RnD@vmssoftware.com

VMS Software, Inc. • 580 Main Street • Bolton MA 01740 • +1 978 451 0110