

Xen/ia64 Update

Progress Report

Gelato – April 2007, San Jose, CA

Joseph Szczypek

HP Open Source and Linux Organization



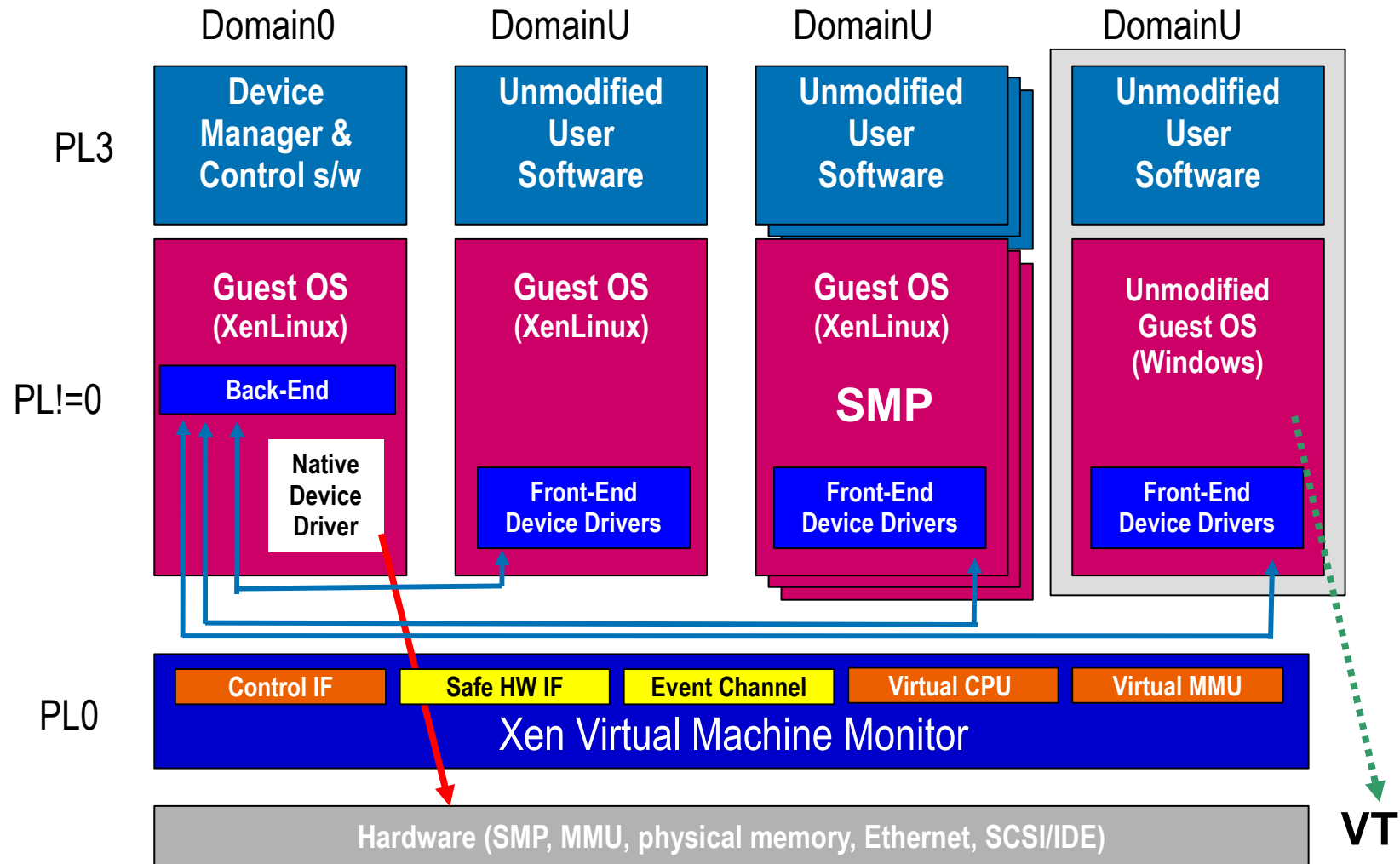
Agenda

- **Xen/ia64 Upstream Status and Progress**
- User and Performance Experience

Xen Intro

- What is it?
 - Virtualizes System
 - Runs entire instance of an OS
- Major components
 - Virtual Machine Monitor – VMM (Hypervisor)
 - runs at CPU privilege level 0
 - Manages Domain access to resources
 - Privileged host domain (Domain0)
 - Device manager and Control software
 - Guest domains (DomainU)
 - Run at lower privilege level

Xen Architecture



Xen Features

- Para-Virtualization (PV)
- Hardware Virtual Machine (HVM)
 - VT-i for ia64 available with Itanium 2 Dual Core
- Save/Restore
- Migration/Relocation
- Driver Domains
- PV-on-HVM
- Memory Ballooning

Xen/ia64 Background

- Xen/ia64 port started in late 2004 by Dan Magenheimer at HP Labs
- Xen/ia64 code base
 - Linux/ia64 code modified for Xen
 - New architecture-specific Xen code
 - Common Xen code
- Current maintainer is Alex Williamson, HP
 - Presented Xen/ia64 at last year's Spring Gelato

Status as of Gelato ICE April 2006

- No guest networking
- SMP guest support just introduced
- No migration
- Limited memory configurations

Status as of Gelato ICE April 2007

- Almost full feature parity with Xen/x86!
- Paravirtualized & HVM guests
- SMP support
- Save/Restore
- Migration
- Driver Domains
- PV-on-HVM drivers for Linux guests
- Memory Balloon
- Virtual I/O
- vCPU hotplug

More Upstream Status

- ia64 MCA error logging
- ia64 INIT handling
- FPSWA support
- EFI variable & time support
- Big Endian guest support (in progress)
- Open source VTi firmware image based on TianoCore as well as image from Intel

Runs on Systems from Multiple Vendors

- Bull - NovaScale
- Fujitsu - PRIMEQUEST
- HP – Integrity Series
- Intel - Tiger
- SGI – Altix (in progress)

Xen/ia64 Future Tasks

- Paravirt-ops
 - Paravirt-ops support for x86 underway
 - Xen/ia64 needs to be ported to paravirt-ops
- NUMA/Big system support
 - Locality & advanced memory maps
 - Jes Sorensen will be presenting later today
- IOMMU support
 - Some work done on this with Xen/x86
 - We'll need to look at this also
- HVM save/restore
 - New Xen/x86 feature in 3.0.5 which we will need to add

More Future Tasks

- Performance monitoring tools
- Debugger support
- Kexec/kdump
 - Integrated into Xen/x86 in 3.0.4
 - Booting Xen->Xen, Xen->Linux, Linux->Xen
 - Simon Horman & Magnus Damm

How to get involved

- Xen/ia64 mailing list: xen-ia64-devel@lists.xensource.com
- Xen/ia64 development branch:
<http://xenbits.xensource.com/ext/xen-ia64-unstable.hg>
– Maintainer: Alex Williamson, HP
- Xen development branch (usually working):
<http://xenbits.xensource.com/xen-unstable.hg>
- Grab a tarball (3.0.5 out soon):
http://www.xensource.com/download/index_3.0.4.html
- Xen/ia64 upstream bugzillas for problems found:
<http://bugzilla.xensource.com>

Agenda

- Xen/ia64 Upstream Status and Progress
- **User and Performance Experience**

Kicking the Tires with RHEL5

- RHEL5 ia64 Virtualization Technology Preview
 - Now available from Red Hat
 - Easier way to try the technology
 - Based on an earlier snapshot of Xen/ia64
- Read Red Hat's ia64 release notes
 - <https://www.redhat.com/docs/manuals/enterprise/>
 - This is a Technology Preview
 - Allows users to try out the feature
 - Generally not suitable for production use
- Usage hints at end of presentation
 - Additional installation and configuration instructions, etc.

Evaluating Performance, Scalability & Stability

- Use AIM7 for workloads
 - Evaluate with compute and IO intensive workloads of different flavors
 - Good way to stress the system
- Specify load points
 - Run multiple guests
 - Run guests simultaneously
 - Run guests at the same load points
 - Why? Repeatability of results

The Hardware

- HP rx6600 Integrity Server
 - 4 socket, 8 cores
 - Virtualization-enabled (VT-i) for HVM guests
 - 96GB memory
 - 1 SmartArray RAID controller
 - 8 SAS drives
 - JBOD
 - 3 dual-channel FibreChannel PCI-X option cards
 - 6 MSA1000 arrays
 - 12 SCSI drives per MSA1000
 - JBOD

Domain0 Observations

- Domain0 boots up with 1 vCPU and 512MB memory
 - Increase memory to 1GB
 - `dom0_mem` in `elilo.conf`
 - Increase Domain0 vCPU count if desired
 - `dom0_max_vcpus` in `elilo.conf`.
 - `elilo.conf`
 - `append=dom0_mem=1G dom0_max_vcpus=2 -- root=/dev/sda2`
 - Entries to the left of – (dash dash) passed to hypervisor
- Shutdown of guests at Dom0 shutdown
 - Disable guest `save` feature in `/etc/sysconfig/xendomains`

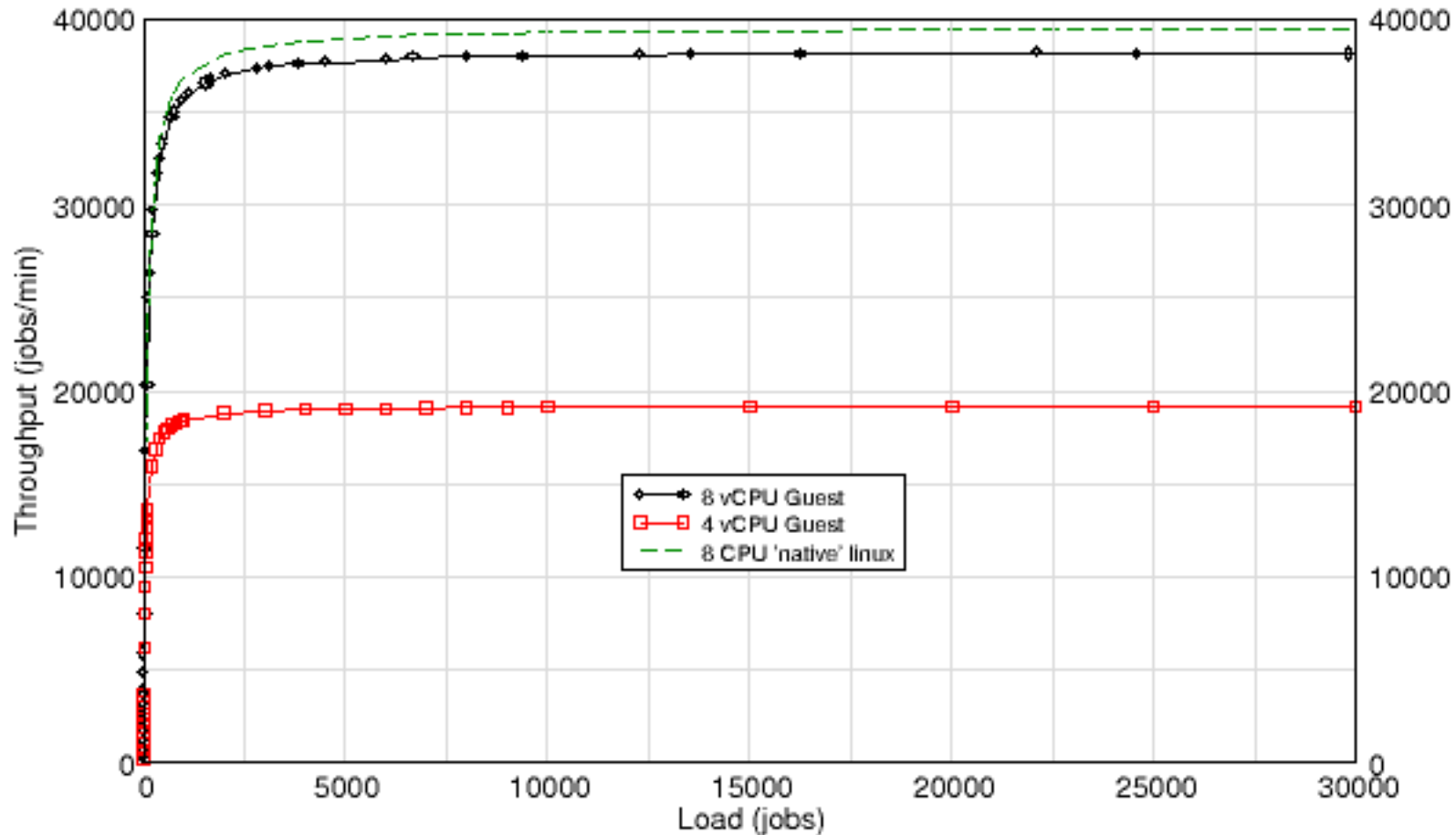
Domain Configs

- Guest (DomainU) configurations used
 - 4,8 vCPU, 32GB paravirtualized guests
 - 1 vCPU, 8GB paravirtualized guest
 - Up to 6 guests run simultaneously
 - 1 vCPU, 8GB fully-virtualized guest
- Domain0 configurations used
 - Domain0 memory: 1GB
 - Domain0 vCPU
 - 1 or 2 vCPUs
 - Unpinned – uses any core
 - Pinned – uses physical core it is pinned to
- All domains installed to their own phy. disk
- Some results...

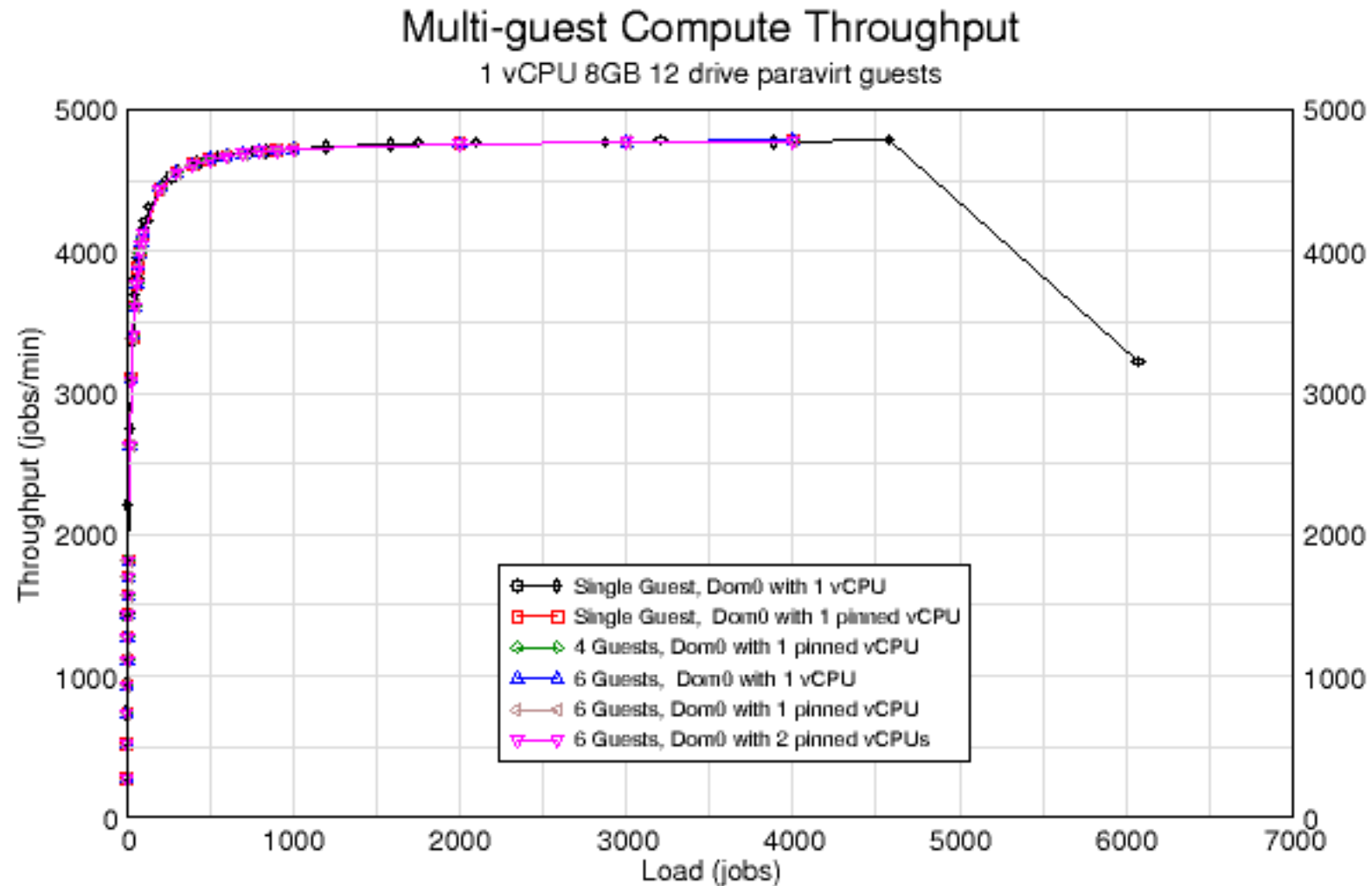
Excellent CPU Scaling for Compute Loads

4-way to 8-way Guest (DomU) Scaling

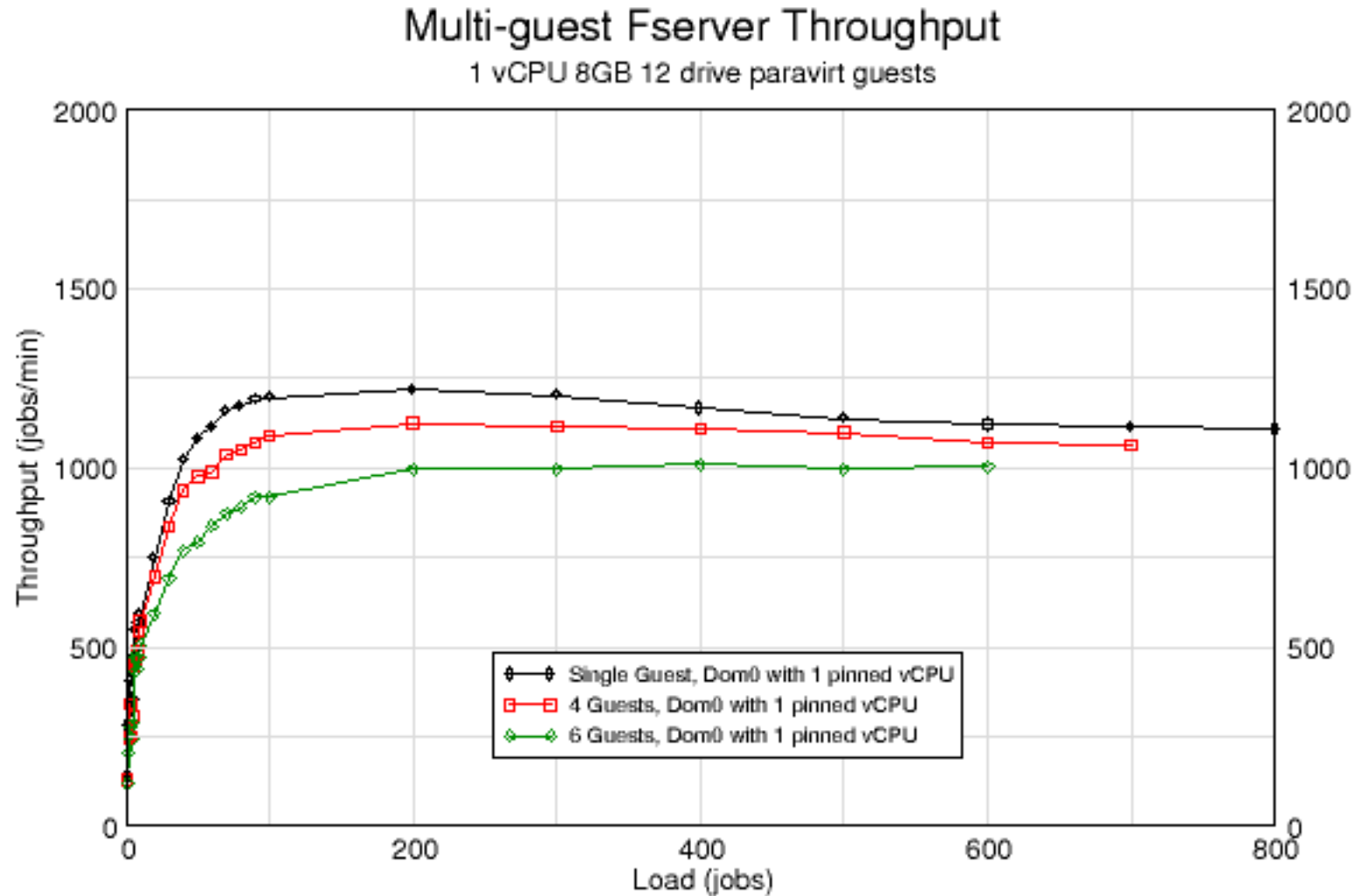
32GB 72 drive paravirt guests - Compute workload



Excellent Guest Scalability for Compute Loads



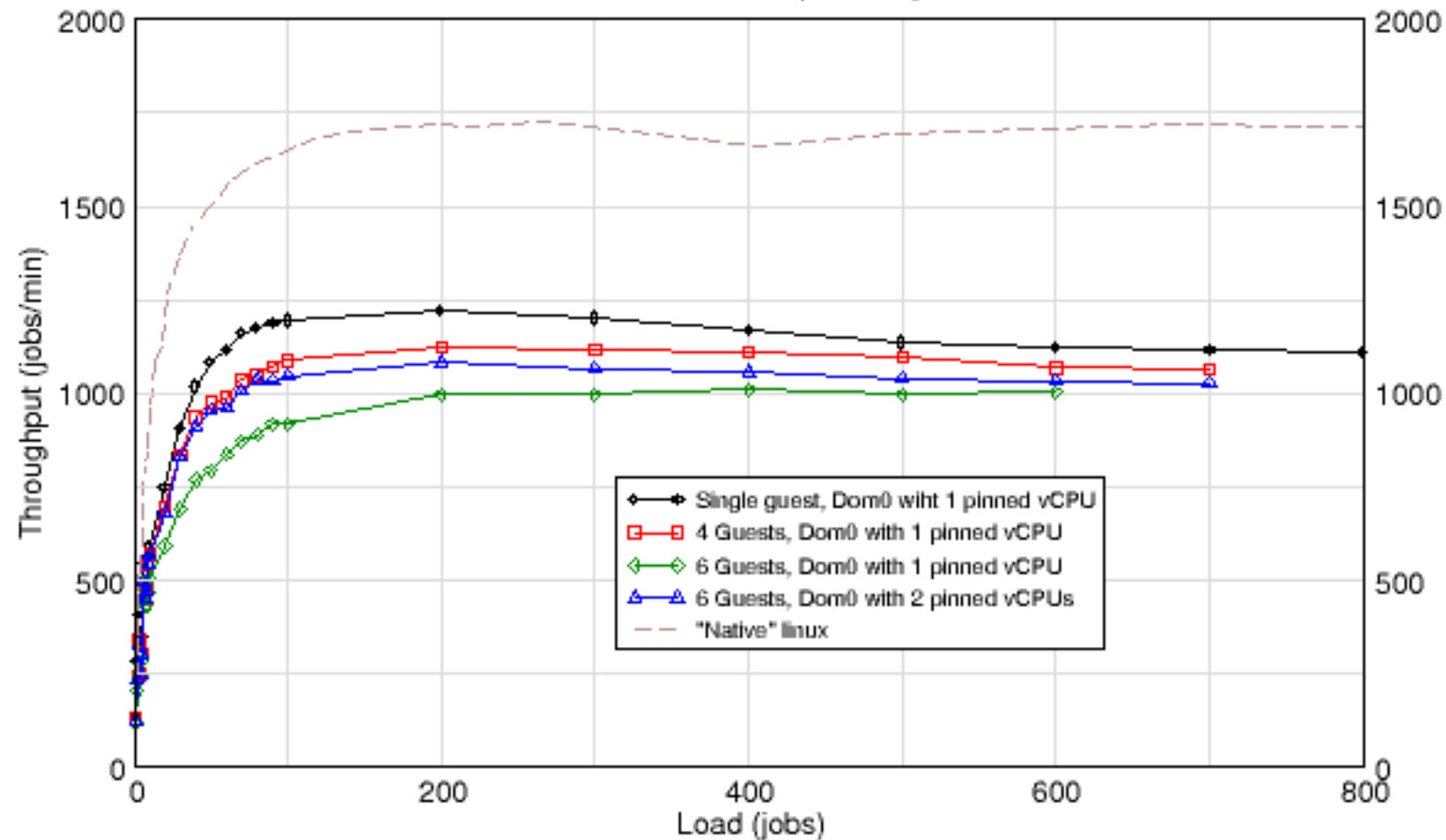
IO Intensive Loads Don't Scale as Well



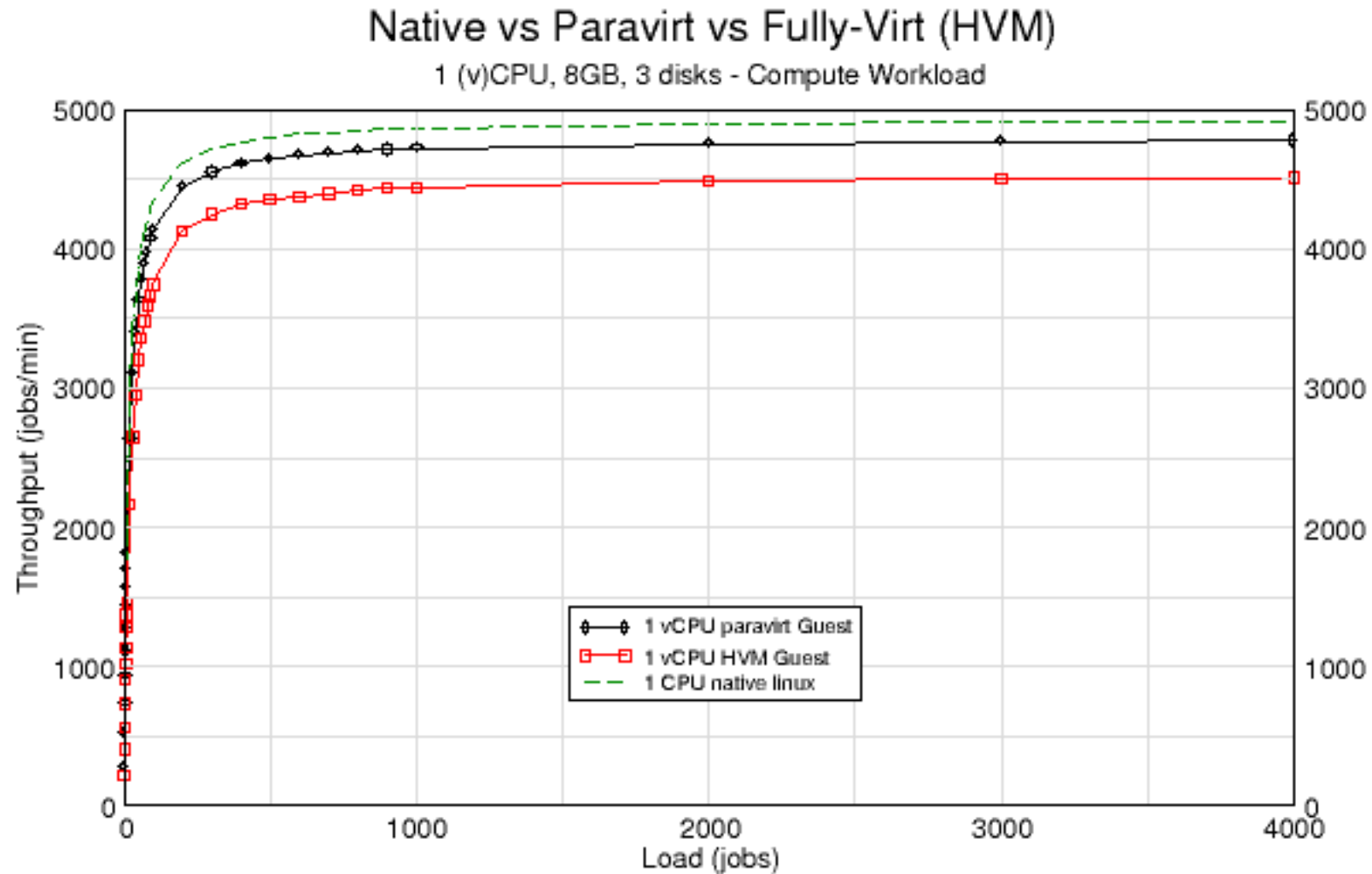
Improved IO Throughput with Extra vCPU

Multi-guest Fserver Throughput w/add'l Dom0 vCPU

1 vCPU 8GB 12 drive paravirt guests



Fully-Virtualized (HVM) Guest Looks Good



Issues Discovered

- Domain0 memory limitation
 - ≥ 2 GB memory gets 'fatal opcode'
- Large memory guest limitation
 - System crashes experienced with 88GB and 92GB memory
- Swapper oops!
 - Try not to use all your available memory
- Fully-virtualized (HVM) guest exhibits odd behavior
- IO intensive workloads can lead to crashes
 - Compute/Shared loads ran well
- Remember, this is a Technology Preview!
 - Report problems via Red Hat's bugzilla
<https://bugzilla.redhat.com>

Summary

- Xen/ia64 Upstream
 - Near feature parity with Xen/x86
 - Working on adding new features that are being/have been developed for Xen/x86
- User Experience
 - Look at ways to improve IO performance
 - Look at quality improvements
- Encourage you to get involved

Usage Hints

RHEL5 ia64 Virtualization Tech Preview

Install Notes

- First install without the virtualization packages
 - Do not specify the linux debug parameter at installation start
 - System will not boot after installation if Virtualization selected
- Second, install virtualization packages
- Modify elilo.conf
 - edit 'default' to boot the 'el5xen' kernel
 - A 'label' entry in elilo.conf contains the full string to use
 - Example: `default=2.6.18-8.1.1.el5xen`

RPMs to install for Virtualization

- VT/xen-3.0.3-25.el5.ia64.rpm
- VT/virt-manager-0.2.6-7.el5.ia64.rpm
- VT/virt-manager-0.2.6-7.el5.ia64.rpm
- VT/libvirt-python-0.1.8-15.el5.ia64.rpm
- VT/python-virtinst-0.99.0-2.el5.noarch.rpm
- Server/bridge-utils-1.1-2.ia64.rpm
- Server/kernel-xen-2.6.18-8.el5.ia64.rpm
- Server/xen-libs-3.0.3-25.el5.ia64.rpm
- Server/gnome-python2-gnomekeyring-2.16.0-1.fc6.ia64.rpm

Add'l RPMs for Fully-Virtualized Domains

- Obtain via Red Hat Network (RHN)
 - xen-3.0.3-25.0.3.el5.ia64.rpm (or newer)
 - xen-libs-3.0.3-25.0.3.el5.ia64.rpm (or newer)
 - Address problem starting fully virtualized domains
- Install Guest Firmware Package for Fully-Virtualized Domains
 - Found on the Supplementary CD-ROM
 - Supplementary/xen-ia64-guest-firmware-1.0.0-8.ia64.rpm

Serial Console Notes

- */etc/inittab* & */etc/securetty* may need to be modified
 - Serial console always ttyS0 in Dom0
 - Modify */etc/inittab* if required.
 - If co: set to use ttyS0 you're ok
 - If not, add a line like
 - s0:2345:respawn:/sbin/agetty ttyS0 vt100-nav
 - Modify */etc/securetty* to allow login on the serial console
 - add ttyS0
- If Dom0 already running
 - Do `'/sbin/init q'` to get prompt

RHEL5 Paravirtualized Domain Installation

- Do not use virt-manager
 - Issue with paravirtualized frame buffer support
- Use virt-install
 - Can use interactive mode or specify all parameters
 - Recommended Values
 - `--ram=2048`
 - HP engineers have tried 1GB-32GB. Larger values may cause issues
 - `--file="/dev/sda"` (for example)
 - HP engineers have used disks, partitions, LVM volumes, file images.
 - file images not recommended –
 - blktap not available at this time
 - An older access method will be used
 - `--nographics` directs DomU console output to the terminal
 - `--location` specifies installation image
 - See RHEL5 Installation Guide

More Paravirtualized Domain Install Notes

- DomU (guest) won't boot after install
 - pygrub doesn't understand gpt nor FAT filesystems
- Modify DomU config file in /etc/xen on Dom0
 - First, copy DomU ramdisk, kernel, and elilo.conf
 - use lomount command to mount DomU disk image
 - copy to Dom0 /var/lib/xen/boot to avoid SELinux issues
 - Comment out pygrub bootloader line
 - Specify location of DomU kernel and ramdisk images
 - kernel="/var/lib/xen/boot/vmlinuz..."
 - ramdisk="/var/lib/xen/boot/initrd..."
 - Specify root location
 - root="value as set in elilo.conf"
 - Use 'extra' line to specify additional kernel parameters
 - ie: extra="3" which will boot you to runlevel 3.

Fully-Virtualized (HVM) Domain Installation

- Use Red Hat's Virtualization Guide
 - <http://www.redhat.com/docs/manuals/enterprise/RHEL-5-manual/Virtualization-en-US/vir>
- Use virt-manager or virt-install
 - Graphics can be used - no frame buffer issue with fully virtualized domains
- Note the following:
 - ISO installation image
 - If SELinux enabled, place in /var/lib/xen/iso for use during installation
 - allows xend to have access to image
 - Virtual memory
 - HP engineers suggest using 1GB-8GB
 - Set Startup Memory = Max Memory
 - Set Virtual CPU count to 1 for installation
 - 2 or more vCPUs can create issues during installation
 - vCPUs can be increased after installation
 - Modify "vcpus=" in DomU configuration file

More Fully-Virtualized Domain Install Notes

- EFI shell entered at start of guest installation
 - Choose filesystem being used by CD-ROM/ISO image
 - Shell> fs0:
 - fs0:\>
 - Type 'elilo' to start guest OS installation
- When installation completes
 - Fully Virtualized Domain will attempt to reboot
 - It will shutdown, but not reboot
 - Use 'xm create' to boot your Fully Virtualized Domain

Additional Information

- Difficult mouse tracking with HVM console
 - HP recommends using alternates
 - ie: text, vnc, ssh
 - At ELILO boot prompt, choose an alternate method
 - ie: ELILO boot: linux vnc
 - See Red Hat's Installation Guide for more on these options
 - http://www.redhat.com/docs/manuals/enterprise/RHEL-5-manual/Installation_Guide-en-US/
- Slow installation from virtual CD-ROM
 - use alternate source, ie: network
 - Example
 - ELILO boot: linux vnc askmethod
 - Can then specify NFS

Xen/ia64 Update

Progress Report

Gelato – April 2007, San Jose, CA

Joseph Szczypek

HP Open Source and Linux Organization



© 2005 Hewlett-Packard Development Company, L.P.
The information contained herein is subject to change without notice.



Agenda

- **Xen/ia64 Upstream Status and Progress**
- User and Performance Experience

2 April 18, 2007



Xen Intro

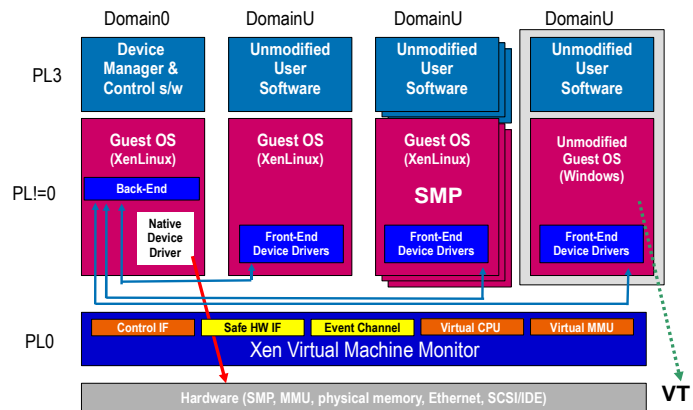
- What is it?
 - Virtualizes System
 - Runs entire instance of an OS
- Major components
 - Virtual Machine Monitor – VMM (Hypervisor)
 - runs at CPU privilege level 0
 - Manages Domain access to resources
 - Privileged host domain (Domain0)
 - Device manager and Control software
 - Guest domains (DomainU)
 - Run at lower privilege level

3 April 18, 2007



- Xen virtualizes an entire system using the same instruction set as the platform. It runs entire instances of an OS.
- Xen is “hostless”, meaning the only code running at privilege level 0 is the hypervisor itself. “Hosted” VMMs share PLO with the host OS
- Xen supports a single privileged host domain, “Domain 0” or “Dom0”. This domain provides device drivers for I/O as well as hosts Xen control software such as Xend.
- Guest domains in Xen are often called “Domain U” or “DomU” for Un- or Under-privileged.
- Domains or guest VMs are “ring compressed” - “demoted” to lower privilege levels, allowing privileged operations to trap into the VMM.

Xen Architecture



4 April 18, 2007



- This diagram shows the basic Xen architecture.
- At the lowest level (in grey) is the underlying platform hardware.
- The Xen VMM (in dark blue) runs in PL0 virtualizing CPUs and memory for the guests above. Interfaces are provided to allow a safe hardware channel for domain0. Event channels and the control interface allow domains to receive virtual interrupts and access the VMM via hypercalls.
- Domain 0 contains device drivers for I/O. Backend drivers in domain 0 export virtualized I/O interfaces to guest domains. Domain 0 also hosts the control software such as Xend.
- Guest domains may be either modified to support running on Xen (here referred to as "XenLinux") or unmodified if the underlying hardware supports full virtualization extensions. Guest domains have frontend drivers which use the I/O exported from the domain 0 backend drivers.
- In all cases, the application level code is untouched.

Xen Features

- Para-Virtualization (PV)
- Hardware Virtual Machine (HVM)
 - VT-i for ia64 available with Itanium 2 Dual Core
- Save/Restore
- Migration/Relocation
- Driver Domains
- PV-on-HVM
- Memory Ballooning

5 April 18, 2007



- Xen supports two types of guests; “para-virtualized” guest and “fully virtualized” guest.
- Para-virtualized guests make modifications to the OS both to support the virtualization and to improve performance. On both x86 and ia64, not all of the operations that need to trap into the VMM are privilege sensitive. Converting these to use hypercalls allows the VMM to perform operations on behalf of the guest. Operations may also be batched to provide more efficient hypercalls.
- Full virtualization is handled by the “Hardware Virtual Machine” or HVM layer of Xen. This layer takes advantage of virtualization extensions (ie: VT-i) found in newer processors. Itanium-2 processors are now available with the VT-i extensions
- Xen supports “Save” and “Restore” of domains allowing a guest to be stopped and restarted at some time in the future.
- “Migration” or “Relocation” is actually an extension of Save/Restore, allowing a domain to be transferred from one physical system to another in real time, with only milliseconds of downtime.
- “Driver Domains” allow guest domains the ability to directly access one or more pieces of underlying hardware. The guest domain “owns” the hardware (usually a NIC or HBA) providing faster access and eliminating bottlenecks through domain 0.
- In a typical fully-virtualized guest, QEMU is used to provide virtual I/O devices to the guest. PV-on-HVM drivers are virtualization-aware drivers that provide the same high performance frontend drivers to the guest as used on PV domains.
- Memory ballooning allow PV guests to dynamically size their memory usage based on requirements of the system.

Xen/ia64 Background

- Xen/ia64 port started in late 2004 by Dan Magenheimer at HP Labs
- Xen/ia64 code base
 - Linux/ia64 code modified for Xen
 - New architecture-specific Xen code
 - Common Xen code
- Current maintainer is Alex Williamson, HP
 - Presented Xen/ia64 at last year's Spring Gelato

6 April 18, 2007



- The Xen code base is actually quite similar in layout to Linux. There's a clear division between architecture specific and generic code. Wherever possible, Xen/ia64 tries to make use of this common code base. When and where necessary we add ia64 specific code. This code is often derivatives of or exact copies of Linux/ia64 code.
- Current maintainer for Xen/ia64 is Alex Williamson at HP. I'd like to thank Alex for his work as the Xen/ia64 development branch maintainer, and for his Xen/ia64 upstream status input for this presentation.

Status as of Gelato ICE April 2006

- No guest networking
- SMP guest support just introduced
- No migration
- Limited memory configurations

7 April 18, 2007



- Last year, at Gelato Spring 2006, Alex Williamson, the xen/ia64 tree maintainer, presented xen/ia64 status.
- Xen/ia64 was functional, but missing several key features
- No guest networking - patches available to make guest domain networking functional.
- SMP guest support was new in the tree.
- Migration wasn't implemented.
- Domain0, DomainU (guests), and overall system memory configurations were significantly limited.

Status as of Gelato ICE April 2007

- Almost full feature parity with Xen/x86!
- Paravirtualized & HVM guests
- SMP support
- Save/Restore
- Migration
- Driver Domains
- PV-on-HVM drivers for Linux guests
- Memory Balloon
- Virtual I/O
- vCPU hotplug

8 April 18, 2007



- The Xen/ia64 port is now to a point where we almost have full feature parity with Xen/x86. Mostly lags in the areas of new feature development.
- Linux is the primary paravirtualized guest. There is also the mini-os port for Xen/ia64 based on FreeBSD. “mini-os” is a stripped down OS intended to sit below a fully-virtualized guest for better I/O access (reducing the domain 0 bottleneck and allowing better process time accounting).
- Xen/ia64 can run a variety of fully virtualized guests: ie: many flavors for Linux, Windows Server 2003 (for ia64)
- SMP – Full support for host and guest SMP. Number of virtual CPUs can be much greater than the number of physical CPUs
- Save/Restore/Migration – works
- Driver Domains – works
- PV-on-HVM drivers – available for HVM Linux guests
- Memory Ballooning - works
- Virtual I/O – Network & block frontend/backend work
- Virtual CPU hotplug – works on both PV & HVM guests

More Upstream Status

- ia64 MCA error logging
- ia64 INIT handling
- FPSWA support
- EFI variable & time support
- Big Endian guest support (in progress)
- Open source VTi firmware image based on TianoCore as well as image from Intel

9 April 18, 2007



- Moving on to more ia64 specific features...
- MCA error logging is handled transparently through Xen. Salinfod running in dom0 is able to retrieve error logs from the system and across all physical CPUs.
- INIT handling allows individual domains to be sent a virtual INIT for generating crash dumps. Xen also intercepts physical INITs for dumps.
- FPSWA – The “Floating Point SoftWare Assist” module is transparently multiplexed among domains.
- Basic EFI services like variable and time services are available to guest domains. (mainly read-only for guests other than domain 0).
- Big Endian guest support is in progress. The mini-os port of FreeBSD has a build time switch allowing either Endian-ness to be compiled. The virtual I/O devices still need work to swap bytes at the appropriate points.
- For fully virtualized HVM guests, Xen requires a guest firmware image for QEMU. On x86 this is based on Bochs. ia64, of course, needs EFI/SAL/PAL, etc... For this we have a choice of either the Intel Guest Firmware Image or more recently, an open source image based on TianoCore.

Runs on Systems from Multiple Vendors

- Bull - NovaScale
- Fujitsu - PRIMEQUEST
- HP – Integrity Series
- Intel - Tiger
- SGI – Altix (in progress)

10 April 18, 2007



- Xen/ia64 now runs on quite a large spectrum of systems.
- Altix support isn't fully enabled yet, but significant progress has been made.

Xen/ia64 Future Tasks

- Paravirt-ops
 - Paravirt-ops support for x86 underway
 - Xen/ia64 needs to be ported to paravirt-ops
- NUMA/Big system support
 - Locality & advanced memory maps
 - Jes Sorensen will be presenting later today
- IOMMU support
 - Some work done on this with Xen/x86
 - We'll need to look at this also
- HVM save/restore
 - New Xen/x86 feature in 3.0.5 which we will need to add

11 April 18, 2007



- Paravirt-ops provides a common set of virtualization hooks for all types of virtualization. Likely path to get there is - Xen/ia64 will need to port to use the paravirt-ops interfaces in the xen ia64 tree, then merge with the main xen tree. It will then need to merge with the linux tree (kernel.org)
- Xen was originally intended to be a very tiny hypervisor layer that didn't know much about the platform. That works fairly well for simple uniform systems, but has problems when system layouts get big. How do we handle memory and processor locality and other big system topologies without pulling all of Linux into Xen?
- We're currently not taking advantage of the IOMMU hardware available on some ia64 platforms. This may also tie in with the big system/locality issues. Some work has been done for IOMMU support on Xen/x86, but it's not merged yet.
- HVM save/restore – This is a new feature for Xen/x86 in 3.0.5. Xen/ia64 needs to catch up and try to implement this too.

More Future Tasks

- Performance monitoring tools
- Debugger support
- Kexec/kdump
 - Integrated into Xen/x86 in 3.0.4
 - Booting Xen->Xen, Xen->Linux, Linux->Xen
 - Simon Horman & Magnus Damm

12 April 18, 2007



- As Xen/ia64 moves into a production phase, performance monitoring and debugging support will become more necessary.
- Along those lines, Kexec and Kdump support was integrated into Xen/x86 in 3.0.4. Simon Horman and Magnus Damm have been working to add the same for ia64, but would certainly appreciate help.

How to get involved

- Xen/ia64 mailing list: xen-ia64-devel@lists.xensource.com
- Xen/ia64 development branch:
<http://xenbits.xensource.com/ext/xen-ia64-unstable.hg>
– Maintainer: Alex Williamson, HP
- Xen development branch (usually working):
<http://xenbits.xensource.com/xen-unstable.hg>
- Grab a tarball (3.0.5 out soon):
http://www.xensource.com/download/index_3.0.4.html
- Xen/ia64 upstream bugzillas for problems found:
<http://bugzilla.xensource.com>

13 April 18, 2007



- How to try Xen/ia64 or get involved with development?
- xen-ia64-devel mailing list - the main gathering place of the Xen/ia64 community. Over 200 members subscribed to the list.
- Thank you to Alex Williamson at HP for his work as the Xen/ia64 development branch maintainer, and for his Xen/ia64 upstream status input for this presentation.
- Much like Linux, the bulk of the architecture specific changes happen in a sub branch which gets pulled into the main tree on a fairly regular basis. Here, the architecture branch is the xen-ia64-unstable.hg tree, which gets pulled into xen-unstable.hg.
- A tarball is also available as an alternative to mercurial
- Report problems found with upstream Xen/ia64 via the listed website.

Agenda

- Xen/ia64 Upstream Status and Progress
- **User and Performance Experience**

14 April 18, 2007



Kicking the Tires with RHEL5

- RHEL5 ia64 Virtualization Technology Preview
 - Now available from Red Hat
 - Easier way to try the technology
 - Based on an earlier snapshot of Xen/ia64
- Read Red Hat's ia64 release notes
<https://www.redhat.com/docs/manuals/enterprise/>
 - This is a Technology Preview
 - Allows users to try out the feature
 - Generally not suitable for production use
- Usage hints at end of presentation
 - Additional installation and configuration instructions, etc.

15 April 18, 2007



- Red Hat Enterprise Linux 5 Server for ia64 contains the Virtualization Technology Preview. The goal of my work was to determine how well Red Hat's Technology Preview ran on HP's ia64 hardware.
- The ia64 Virtualization Technology Preview is based on an earlier snapshot of Xen/ia64. Improvements added to upstream (ie: xen-ia64-unstable) since the snapshot was taken may not be present in the Technology Preview.
- Read Red Hat's release notes for ia64. They explain what a 'Technology Preview' is and contain additional notes for the ia64 Virtualization Technology Preview.
- There are important notes at the end of this presentation which cover how to install the ia64 Virtualization Technology Preview and guests domains.

Evaluating Performance, Scalability & Stability

- Use AIM7 for workloads
 - Evaluate with compute and IO intensive workloads of different flavors
 - Good way to stress the system
- Specify load points
 - Run multiple guests
 - Run guests simultaneously
 - Run guests at the same load points
 - Why? Repeatability of results

16 April 18, 2007



invent

The Hardware

- HP rx6600 Integrity Server
 - 4 socket, 8 cores
 - Virtualization-enabled (VT-i) for HVM guests
 - 96GB memory
 - 1 SmartArray RAID controller
 - 8 SAS drives
 - JBOD
 - 3 dual-channel FibreChannel PCI-X option cards
 - 6 MSA1000 arrays
 - 12 SCSI drives per MSA1000
 - JBOD

17 April 18, 2007



- This configuration is believed to be representative of a configuration a customer would use to do virtualization.
- SAN storage is used since it is likely a customer would already have virtualized their storage.

Domain0 Observations

- Domain0 boots up with 1 vCPU and 512MB memory
 - Increase memory to 1GB
 - `dom0_mem` in `elilo.conf`
 - Increase Domain0 vCPU count if desired
 - `dom0_max_vcpus` in `elilo.conf`.
 - `elilo.conf`
 - `append=dom0_mem=1G dom0_max_vcpus=2 – root=/dev/sda2`
 - Entries to the left of `–` (dash dash) passed to hypervisor
- Shutdown of guests at Dom0 shutdown
 - Disable guest save feature in `/etc/sysconfig/xendomains`

18 April 18, 2007



- The ia64 Tech Preview will boot up with 1 vCPU and 512MB by default. This differs from xen/x86 where you see all vCPUs and memory.
- Increase vCPU count for Domain0 by adding `dom0_max_vcpus = 'x'` to `elilo.conf`. Add to the 'append' line, before the '--'.
- Memory can be increased by using `dom0_mem='y'` in `elilo.conf`. Add to the 'append' line, before the '--'.
- By default save/restore is enabled for guests. If you wish to shutdown guests (vs save guests) at Dom0 shutdown time, disable the guest save feature by editing `xendomains` in `/etc/sysconfig` on Dom0. Set the `XENDOMAINS_SAVE="directory"` entry equal to nothing/blank.

Domain Configs

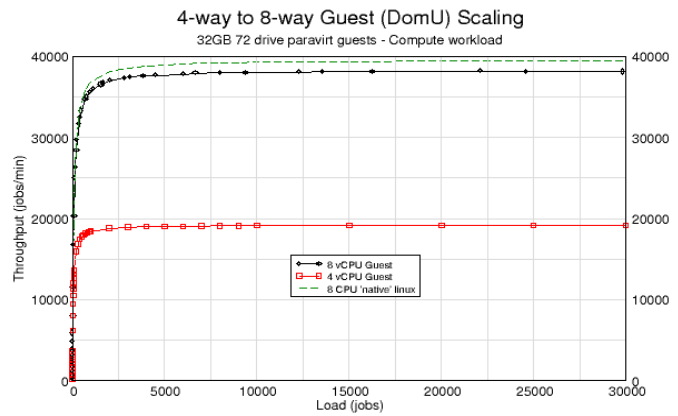
- Guest (DomainU) configurations used
 - 4,8 vCPU, 32GB paravirtualized guests
 - 1 vCPU, 8GB paravirtualized guest
 - Up to 6 guests run simultaneously
 - 1 vCPU, 8GB fully-virtualized guest
- Domain0 configurations used
 - Domain0 memory: 1GB
 - Domain0 vCPU
 - 1 or 2 vCPUs
 - Unpinned – uses any core
 - Pinned – uses physical core it is pinned to
- All domains installed to their own phy. disk
- Some results...

19 April 18, 2007



- All guests (DomainUs) are RHEL5.
- The 1 vCPU paravirtualized guests had 12 target drives each for workloads to use. The 4 and 8 vCPU DomainUs had 72 target drives for the workloads to use. The 4 and 8-way guests were run to look at CPU scaling. The fully-virtualized (HVM) guest had 3 target drives for workloads to use.
- Pinning of vCPUs to physical CPUs was done to investigate how such a configuration ran.
- Pin Domain 0 physical CPUs to vCPUs by doing the following: add 'dom0_vcpus_pin' to 'append' line in elilo.conf, to the left of the double dash ("--").

Excellent CPU Scaling for Compute Loads

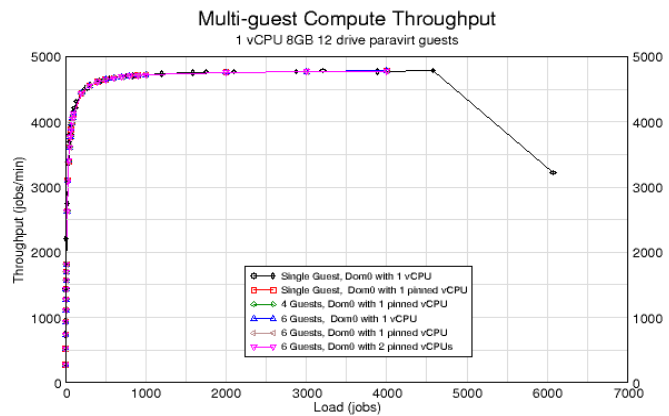


20 April 18, 2007



- CPU Scaling: 4 vCPU paravirtualized DomU and an 8 vCPU paravirtualized DomU are shown on this graph. There is nearly a 100% improvement (doubling of throughput) when vCPUs are doubled from 4 to 8.
- For these graphs, with respect to terminology, DomU = DomainU = Guest.
- The 'native' linux trace is included for reference.

Excellent Guest Scalability for Compute Loads

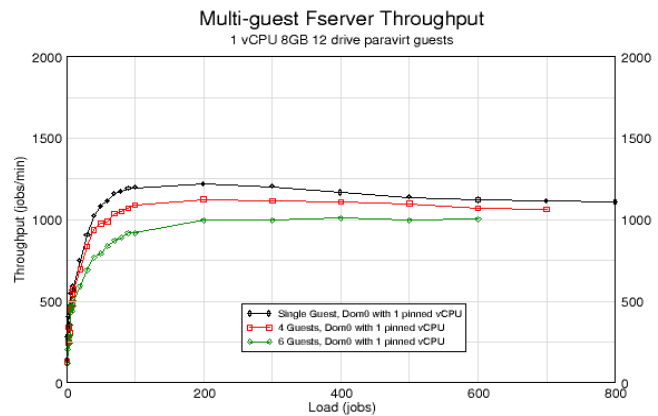


21 April 18, 2007



- This set of data shows guest throughput with a compute-intensive workload. The data for a single guest run by itself, and the data for multiple guests run simultaneously (4 guests and 6 guests) all plot on top of each other.
- There is excellent scalability for guest counts up to 6 guests. 4 guests running this workload simultaneously get an aggregate of 4x that of a single guest running by itself. 6 guests running this workload simultaneously get an aggregate of 6x that of a single guest running by itself.
- Analysis of larger guest counts is planned for the future.

IO Intensive Loads Don't Scale as Well

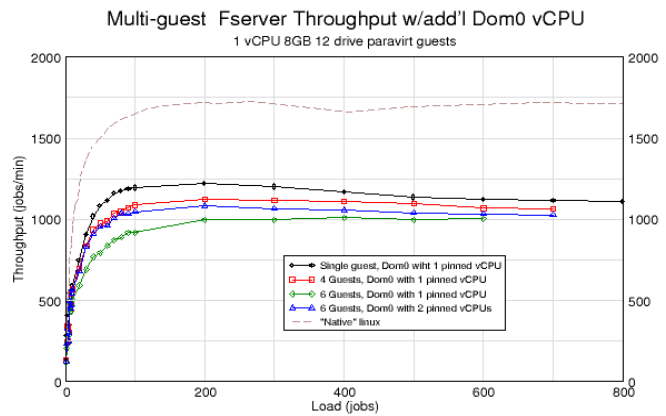


22 April 18, 2007



- This graph shows IO intensive loads don't scale as well, but also show that a substantial amount of work is still being done when multiple guests (DomUs) are running the more IO intensive workload.
- As an increasing number of guests is run, performance of any single instance of 'multiple guests' is impacted when compared to the case where a single guest is running by itself.
- With 4 guests, you get ~96% when compared to a single guest running by itself at a load of 600 jobs.
- With 6 guests, you get ~88% when compared to a single guest running by itself at a load of 600 jobs.
- When you add in the fact that multiple guests are doing work concurrently, you find the aggregate throughput of these guests is much higher than that of a single guest running by itself.

Improved IO Throughput with Extra vCPU

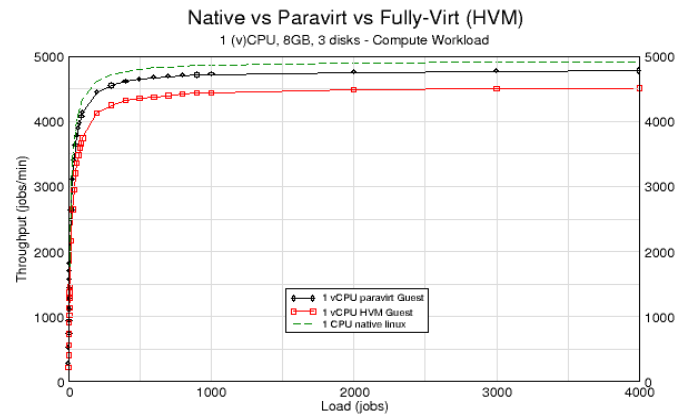


23 April 18, 2007



- This graph builds on the previous one. A 2nd vCPU has been added to Dom0, increasing its ability to handle the IO load.
- 6 guests were again run with the fserver workload. Note that there is an improvement when comparing the instances of these 6 guests to the single guest, run by itself, with a 1 vCPU Dom0. (At the 600 job load point, you improve from ~88% of the performance of the single guest running by itself on the system, to ~92%
- Overall aggregate throughput for the multiple guests will be increased.
- The 'native' linux trace is included for reference.

Fully-Virtualized (HVM) Guest Looks Good



24 April 18, 2007



- This slide shows how a guest running on 'bare metal' (HVM using VT-i) compares to a paravirtualized guest, and to the system running a 1 CPU/8GB "native" linux
- Paravirtualized guest is ~97% of 'native' linux
- Fully-virtualized guest is ~92% of 'native' linux
- With compute-intensive loads, a fully-virtualized guest looks reasonable when compared to a paravirtualized guest and 'native' linux. How it would do with more IO-intensive workloads is of course a interesting question...

Issues Discovered

- Domain0 memory limitation
 - >= 2GB memory gets 'fatal opcode'
- Large memory guest limitation
 - System crashes experienced with 88GB and 92GB memory
- Swapper oops!
 - Try not to use all your available memory
- Fully-virtualized (HVM) guest exhibits odd behavior
- IO intensive workloads can lead to crashes
 - Compute/Shared loads ran well
- Remember, this is a Technology Preview!
 - Report problems via Red Hat's bugzilla
<https://bugzilla.redhat.com>

25 April 18, 2007



• This is a Technology Preview. Please file bugzillas for problems encountered in Red Hat's Bugzilla system.

• Have experienced 'fatal opcode' errors when trying to run Dom0 with dom0_mem set to 2G or greater.

• Large memory paravirt guest limitation: guests with 88GB, 92GB memory had issues. In this case you will see many "Cannot handle page request order 0!" messages followed by a couple of couple traces, then, just before the panic, by "unwind.desc_label_state(): out of memory"

• Instability can be encountered with IO intensive workloads. Limiting number of guests and/or controlling load may help...

• Fully-virtualized guests have exhibited odd behavior such as getting an "illegal instruction" message when trying to run a setup tool or unmounting a filesystem. Also have seen a case where a file was not available/accessible.

Summary

- Xen/ia64 Upstream
 - Near feature parity with Xen/x86
 - Working on adding new features that are being/have been developed for Xen/x86
- User Experience
 - Look at ways to improve IO performance
 - Look at quality improvements
- Encourage you to get involved

26 April 18, 2007



- User experience is for the technology as delivered with RHEL5.
- Need to repeat experience with upstream bits to determine current status.



Usage Hints

27 April 18, 2007



RHEL5 ia64 Virtualization Tech Preview Install Notes

- First install without the virtualization packages
 - Do not specify the linux debug parameter at installation start
 - System will not boot after installation if Virtualization selected
- Second, install virtualization packages
- Modify elilo.conf
 - edit 'default' to boot the 'el5xen' kernel
 - A 'label' entry in elilo.conf contains the full string to use
 - Example: default=2.6.18-8.1.1.el5xen

28 April 18, 2007



- RHEL5 release notes state that you should use 'linux debug' to install the Virtualization Technology Preview. If you do so and select the Virtualization feature, you will encounter an issue – The 'vmm' entry in elilo.conf will be missing and the system will not boot after installation.
- Modifications to elilo.conf are required in order to boot the technology preview. The 'default' entry in elilo.conf will need to be modified to boot Dom0. The string to enter into the 'default=' field can be found by looking at the 'label' fields in elilo.conf. One will be of the form 2.6.18-8...el5xen

RPMs to install for Virtualization

- VT/xen-3.0.3-25.el5.ia64.rpm
- VT/virt-manager-0.2.6-7.el5.ia64.rpm
- VT/virt-manager-0.2.6-7.el5.ia64.rpm
- VT/libvirt-python-0.1.8-15.el5.ia64.rpm
- VT/python-virtinst-0.99.0-2.el5.noarch.rpm
- Server/bridge-utils-1.1-2.ia64.rpm
- Server/kernel-xen-2.6.18-8.el5.ia64.rpm
- Server/xen-libs-3.0.3-25.el5.ia64.rpm
- Server/gnome-python2-gnomekeyring-2.16.0-1.fc6.ia64.rpm

29 April 18, 2007



• Packages listed are those needed to run the ia64 Virtualization Technology Preview

Add'l RPMs for Fully-Virtualized Domains

- Obtain via Red Hat Network (RHN)
 - xen-3.0.3-25.0.3.el5.ia64.rpm (or newer)
 - xen-libs-3.0.3-25.0.3.el5.ia64.rpm (or newer)
 - Address problem starting fully virtualized domains
- Install Guest Firmware Package for Fully-Virtualized Domains
 - Found on the Supplementary CD-ROM
 - Supplementary/xen-ia64-guest-firmware-1.0.0-8.ia64.rpm

30 April 18, 2007



- These packages are the additional packages that need to be installed to run fully-virtualized domains.
- Upgrading via Red Hat Network will get you packages that address an issue with the startup of fully-virtualized domains.
- The Guest Firmware Image must also be installed in order to use fully-virtualized domains.

Serial Console Notes

- */etc/inittab* & */etc/securetty* may need to be modified
 - Serial console always ttyS0 in Dom0
 - Modify */etc/inittab* if required.
 - If *co:* set to use ttyS0 you're ok
 - If not, add a line like
 - s0:2345:respawn:/sbin/agetty ttyS0 vt100-nav
 - Modify */etc/securetty* to allow login on the serial console
 - add ttyS0
 - If Dom0 already running
 - Do *'/sbin/init q'* to get prompt

31 April 18, 2007



- Modify */etc/inittab* & */etc/securetty* to use ttyS0. This the serial console for Dom0.

- If Dom0 is already running, there would have been no login prompt, so you're probably logged in via the network. After making the changes, do a *'telinit q'* or *'/sbin/init q'* and the serial console login prompt should appear.

RHEL5 Paravirtualized Domain Installation

- Do not use virt-manager
 - Issue with paravirtualized frame buffer support
- Use virt-install
 - Can use interactive mode or specify all parameters
 - Recommended Values
 - `--ram=2048`
 - HP engineers have tried 1GB-32GB. Larger values may cause issues
 - `--file="/dev/sda"` (for example)
 - HP engineers have used disks, partitions, LVM volumes, file images.
 - file images not recommended –
 - blktp not available at this time
 - An older access method will be used
 - `--nographics` directs DomU console output to the terminal
 - `--location` specifies installation image
 - See RHEL5 Installation Guide

32 April 18, 2007



- Paravirtualized guests are RHEL5
- If file images are to be used for the installation, be aware that ia64 blktp will not be used as it was not available in time for this release. Performance will not be as good as that which is expected with blktp.

More Paravirtualized Domain Install Notes

- DomU (guest) won't boot after install
 - pygrub doesn't understand gpt nor FAT filesystems
- Modify DomU config file in /etc/xen on Dom0
 - First, copy DomU ramdisk, kernel, and elilo.conf
 - use lomount command to mount DomU disk image
 - copy to Dom0 /var/lib/xen/boot to avoid SELinux issues
 - Comment out pygrub bootloader line
 - Specify location of DomU kernel and ramdisk images
 - kernel="/var/lib/xen/boot/vmlinuz..."
 - ramdisk="/var/lib/xen/boot/initrd..."
 - Specify root location
 - root="value as set in elilo.conf"
 - Use 'extra' line to specify additional kernel parameters
 - ie: extra="3" which will boot you to runlevel 3.

33 April 18, 2007



• In order to boot DomU, set up entries in the DomU config file, found in /etc/xen, to specify where the kernel image, ramdisk image, and root are located.

- DomU images and elilo.conf can be copied to Dom0 after DomU's disk image is mounted using lomount
- ie: lomount -diskimage /dev/... -partition 1 /mnt
- Create the /var/lib/xen/boot directory on Dom0 and copy the files to this directory.
- Edit DomU's config file in /etc/xen and add entries for 'kernel=', 'ramdisk=', and 'root='. 'kernel' and 'ramdisk' should be set to point to the images in /var/lib/xen/boot. 'root' should be set to specify where the DomU root is located – this can be obtained by examining the elilo.conf you copied from DomU's disk. Be sure to enclose the kernel, ramdisk, and root entries in double quotes.
- Don't forget to unmount after copying. You can now boot the DomU using the 'xm create' command.

Fully-Virtualized (HVM) Domain Installation

- Use Red Hat's Virtualization Guide
 - <http://www.redhat.com/docs/manuals/enterprise/RHEL-5-manual/Virtualization-en-US/vir>
- Use virt-manager or virt-install
 - Graphics can be used - no frame buffer issue with fully virtualized domains
- Note the following:
 - ISO installation image
 - If SELinux enabled, place in /var/lib/xen/iso for use during installation
 - allows xend to have access to image
 - Virtual memory
 - HP engineers suggest using 1GB-8GB
 - Set Startup Memory = Max Memory
 - Set Virtual CPU count to 1 for installation
 - 2 or more vCPUs can create issues during installation
 - vCPUs can be increased after installation
 - Modify "vcpus=" in DomU configuration file

34 April 18, 2007



- When installing from an ISO image, place the image under the /var/lib/xen directory. HP suggests creating a /var/lib/xen/iso directory and placing your ISO installation image here. This will allow xend to have access to the installation image should you have SELinux enabled.
- If you wish to increase the number of vCPUs you are using with DomU, edit the DomU configuration file. Change the 'vcpus=' entry to your desired value.
- The DomU configuration file can be found on Dom0, in /etc/xen.

More Fully-Virtualized Domain Install Notes

- EFI shell entered at start of guest installation
 - Choose filesystem being used by CD-ROM/ISO image
 - Shell> fs0:
 - fs0:\>
 - Type 'elilo' to start guest OS installation
- When installation completes
 - Fully Virtualized Domain will attempt to reboot
 - It will shutdown, but not reboot
 - Use 'xm create' to boot your Fully Virtualized Domain

35 April 18, 2007



- When you boot to start the installation, you will enter the EFI shell. Select the appropriate filesystem (where your installation image is), and then type 'elilo' to continue the installation.

Additional Information

- Difficult mouse tracking with HVM console
 - HP recommends using alternates
 - ie: text, vnc, ssh
 - At ELILO boot prompt, choose an alternate method
 - ie: ELILO boot: linux vnc
 - See Red Hat's Installation Guide for more on these options
 - http://www.redhat.com/docs/manuals/enterprise/RHEL-5-manual/Installation_Guide-en-US/
- Slow installation from virtual CD-ROM
 - use alternate source, ie: network
 - Example
 - ELILO boot: linux vnc askmethod
 - Can then specify NFS

36 April 18, 2007

