Red Hat Summit, Boston, 2010

KVM@IBM: Virtualization, Consolidation and Maximizing Server Utilization

Copyright Steven Ginn
Used with permission

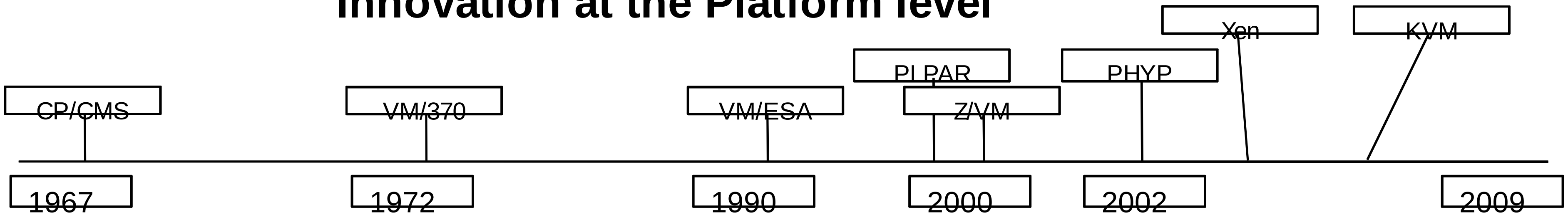
Gerrit Huizenga

Agenda

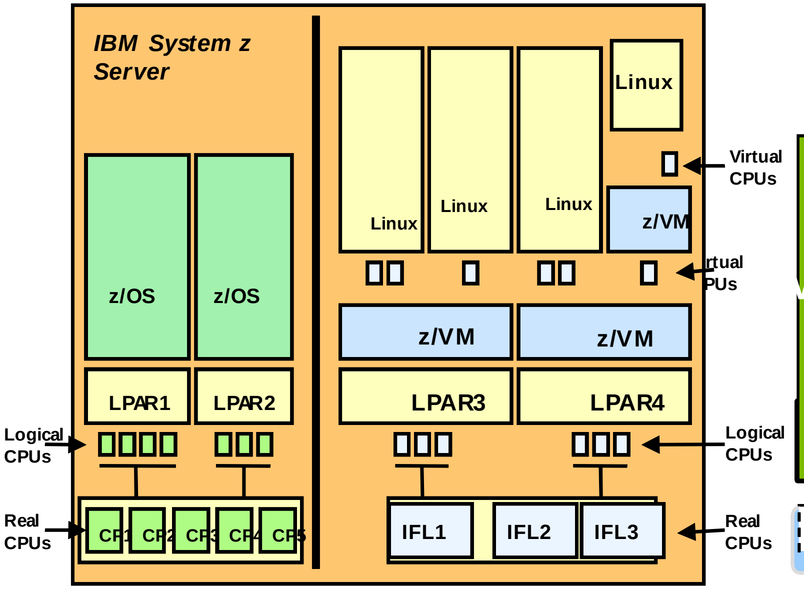
- **Background / History and Red Hat Partnership**
- KVM and Cloud Requirements
- IO in Virtualized Environment
- Memory Resources
- EX5 Systems : Designed with Virtualization in mind

The Evolution of Virtualization

Innovation at the Platform level

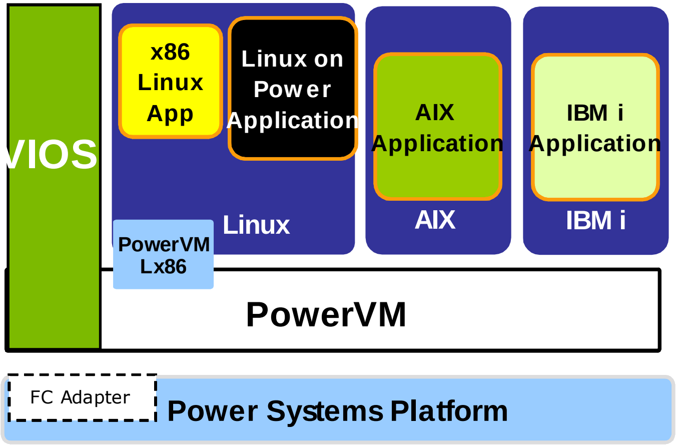


CP/CMS
1967
VM/370
1972
VM/ESA
1990
PLPAR
Z/VM
2000
PHYP
2002
Xen
KVM
2009



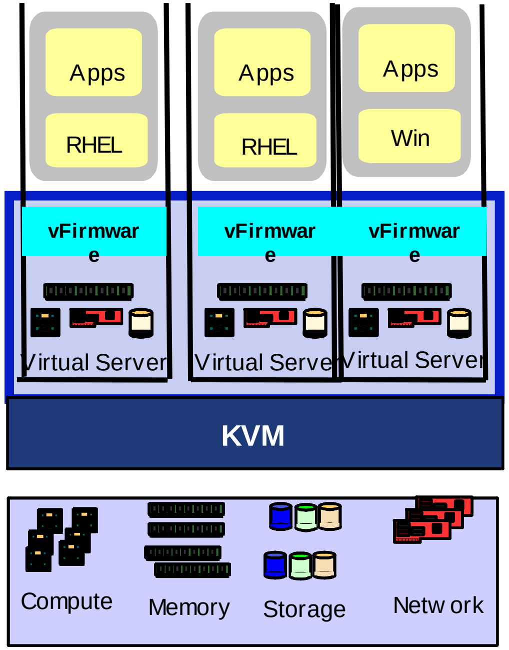
IBM System z Server
z/OS
z/OS
LPAR1
LPAR2
Logical CPUs
Real CPUs
CP1
CP2
CP3
CP4
CP5
Linux
Linux
Linux
Linux
z/VM
Virtual CPUs
rtual PUs
z/VM
z/VM
LPAR3
LPAR4
Logical CPUs
IFL1
IFL2
IFL3
Real CPUs

System z



VIOS
x86 Linux App
Linux on Power Application
AIX Application
IBM i Application
PowerVM Lx86
Linux
AIX
IBM i
PowerVM
FC Adapter
Power Systems Platform

Power

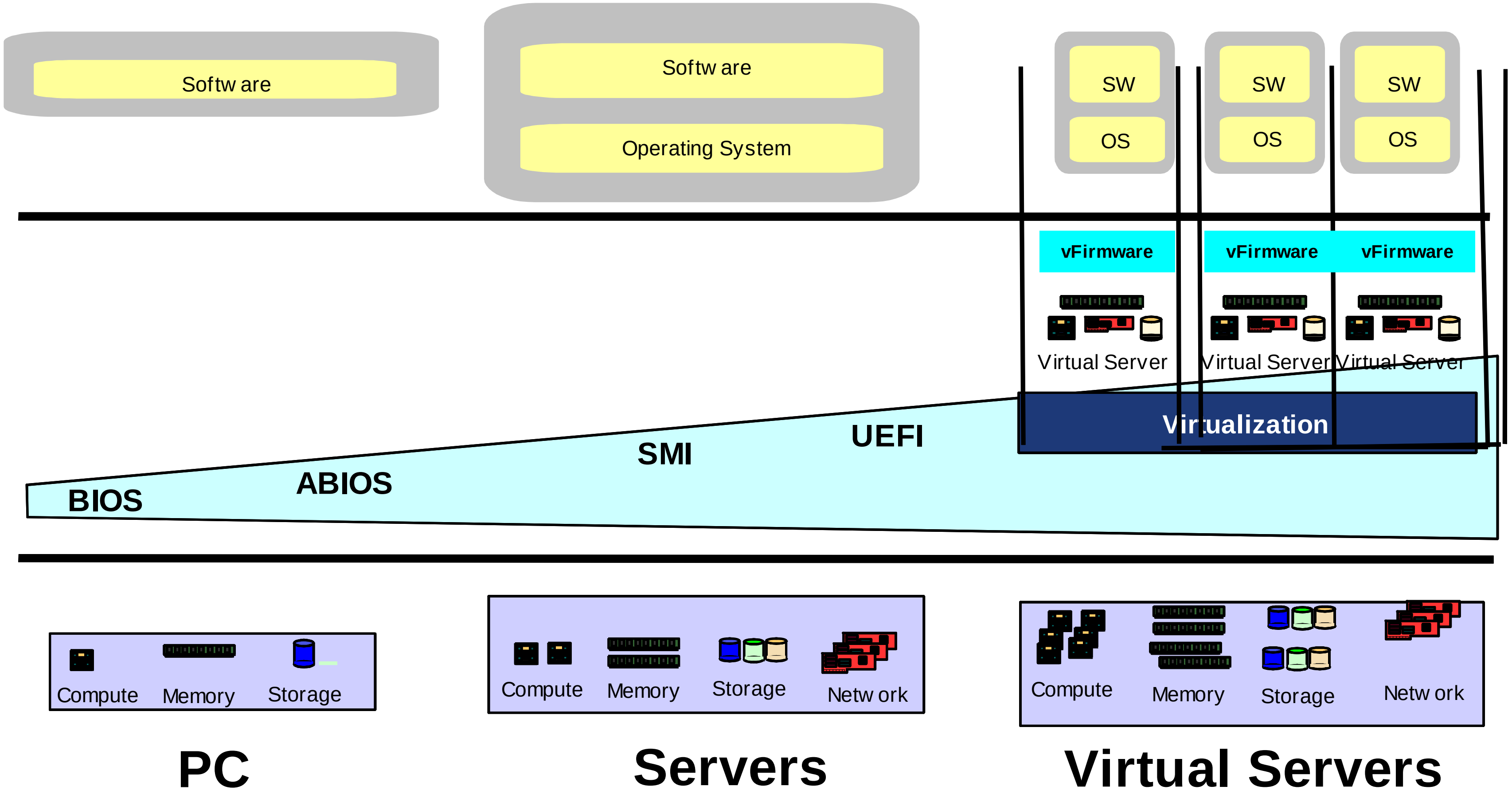


Apps
RHEL
Apps
RHEL
Apps
Win
vFirmwar e
vFirmwar e
vFirmwar e
Virtual Server
Virtual Serve
Virtual Server
KVM
Compute
Memory
Storage
Netw ork

System x

The Evolution of the x86 “Platform”

Processors , firmware, operating systems, and applications all continue to evolve



KVM (Kernel-base Virtual Machine): Overview

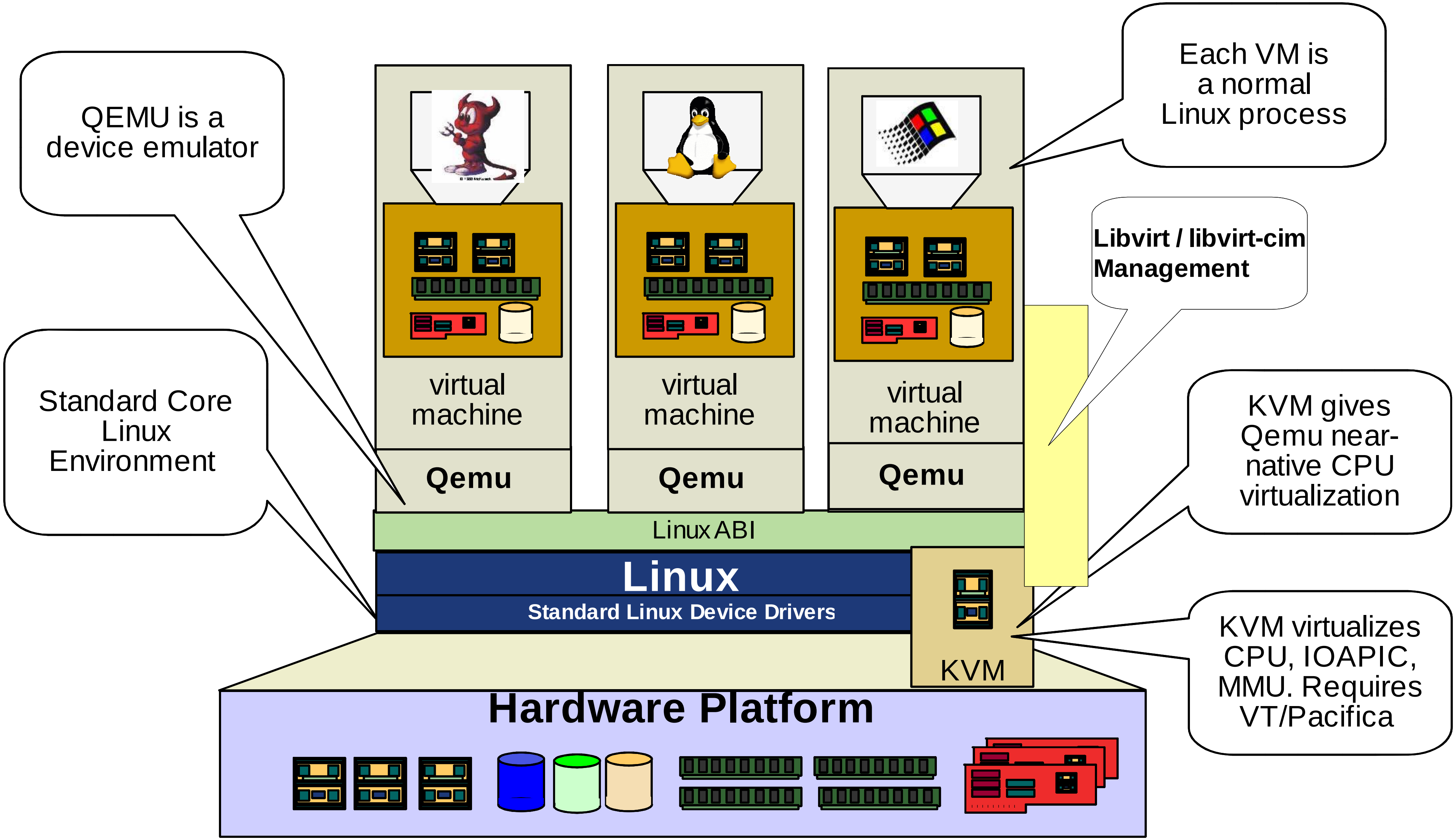
http://www.linux-kvm.org

- Integrated Hypervisor for Linux
  - **Converts Linux into a Type-1 Hypervisor**
- Runs Windows, Linux and other guests
- Allows for Hybrid-mode operation
  - **Run regular Linux applications along side VM guests**
- Upstream since Linux 2.6.20 (2007)
- Control over future evolution is held by linux development community
- Supported in RHEL since v5.4 (Sept. 2009)
- Elegant, simple design reuses Linux and builds upon CPU virtualization assistance

... So KVM Developers can focus on Virtualization

- **While Linux ecosystem continues to provide essential core services**
  - Hardware support
  - Bootstrap
  - Memory Management
  - Process Management and Scheduling
  - Access control
  - IPC and Sharing infrastructure
  - Scaling
  - RAS
  - Power Management

KVM + QEMU Architecture


QEMU is a device emulator
Standard Core Linux Environment
virtual machine
virtual machine
virtual machine
Qemu
Qemu
Qemu
Linux ABI
Linux
Standard Linux Device Drivers
KVM
Hardware Platform
Each VM is a normal Linux process
Libvirt / libvirt-cim Management
KVM gives Qemu near-native CPU virtualization
KVM virtualizes CPU, IOAPIC, MMU. Requires VT/Pacifica

KVM is a Virtualization Driver

- KVM is a small kernel driver that adds virtualization support on multiple architectures
  - AMD, Intel (included in 2.6.20)
  - KVM-lite: PV Linux guest on non-VTx / non-SVM host
  - IA64 (included in 2.6.26)
  - S390 (included in 2.6.26)
  - Embedded PowerPC (power.org, included in 2.6.26)
- About 30k LOCS
- Compared to ~250k LOCS for Xen
- Uses QEMU in userspace as a device model
- Safe to use by unprivileged userspace processes
- Can leverage almost all Linux features

KVM Development Communities - 2009

- KVM-devel
  - 18,303 messages
  - 884 unique participants
  - 382 unique address domains

9471	redhat.com
1382	ibm.com
929	intel.com
949	novell.com

- Qemu
  - 23,562 messages
  - 757 unique participants
  - 349 unique address domains

8751	redhat.com
2643	ibm.com
819	aurel32.net
712	codesourcery.com

- Libvirt
  - 8,835 messages
  - 370 unique participants
  - 194 unique address domains

5791	redhat.com
415	meyering.net
260	ibm.com
230	sun.com

2010 LTC KVM Focus Areas

Core KVM

- Cooperative Memory Management
- Balloon driver
- Qemu maintainership
- KVM function/feature
- VirtFS
- Energy management - CPU folding

Networking, I/O

- Virtio, vhost-net enhancements
- PCI device assignments to Vms
- SRIOV support
- Efficient interrupt handling/routing
- Vswitch
- Advanced ACLs, SNMP MIBs
- Automatic profile migrat

Performance

- Cooperative Memory Management
- Memory overcommit study
- SPECvirt
- Micro-benchmarks
- Network I/O
- Storage & FileSystem

Systems Management

- Libvirt-cim function/feature
- Libvirt storage & network pools
- libvirt-cim maintainership
- Director integration
- Cloud management integration

Security

- Flexible policy support in sVirt
- Common criteria certification
- Blueprints: Cloud security

Hygiene

- RAS – tracepoints, dump, serviceability
- ID
- Support
- Test

Early Deployment Team

- Compute Cloud
- Private Clouds
- Systems Management Integration
- PoC, Partner Engagements

Agenda

- Background / History and Red Hat Partnership
- **KVM and Cloud Requirements**
- IO in Virtualized Environment
- Memory Resources
- EX5 Systems : Designed with Virtualization in mind

Cloud Computing and Hypervisors

- **Cloud Computing is primarily about Economics**
  - Driving down the cost of all aspects of Data Center Operations
  - Sharing Data Center Resources for increased Flexibility
- **For KVM, this translates to:**
  - Upward pressure on VM Density
  - KVM must get more out of less hardware
  - Downward pressure on Energy Consumption
  - Increased Security and Auditing needs
  - Creative use of storage resources

KVM Performance Activities

- **Six separate focus areas of performance analysis**
- Memory Usage and Over-commitment
- Storage (local, SAN, and NAS)
- Network (10G, SR-IOV, paravirtual)
- Windows VM performance
- SPECVirt and complex workload analysis
- Micro benchmarks and regression analysis

Agenda

- Background / History and Red Hat Partnership
- KVM and Cloud Requirements
- **IO in Virtualized Environment**
- Memory Resources
- EX5 Systems : Designed with Virtualization in mind

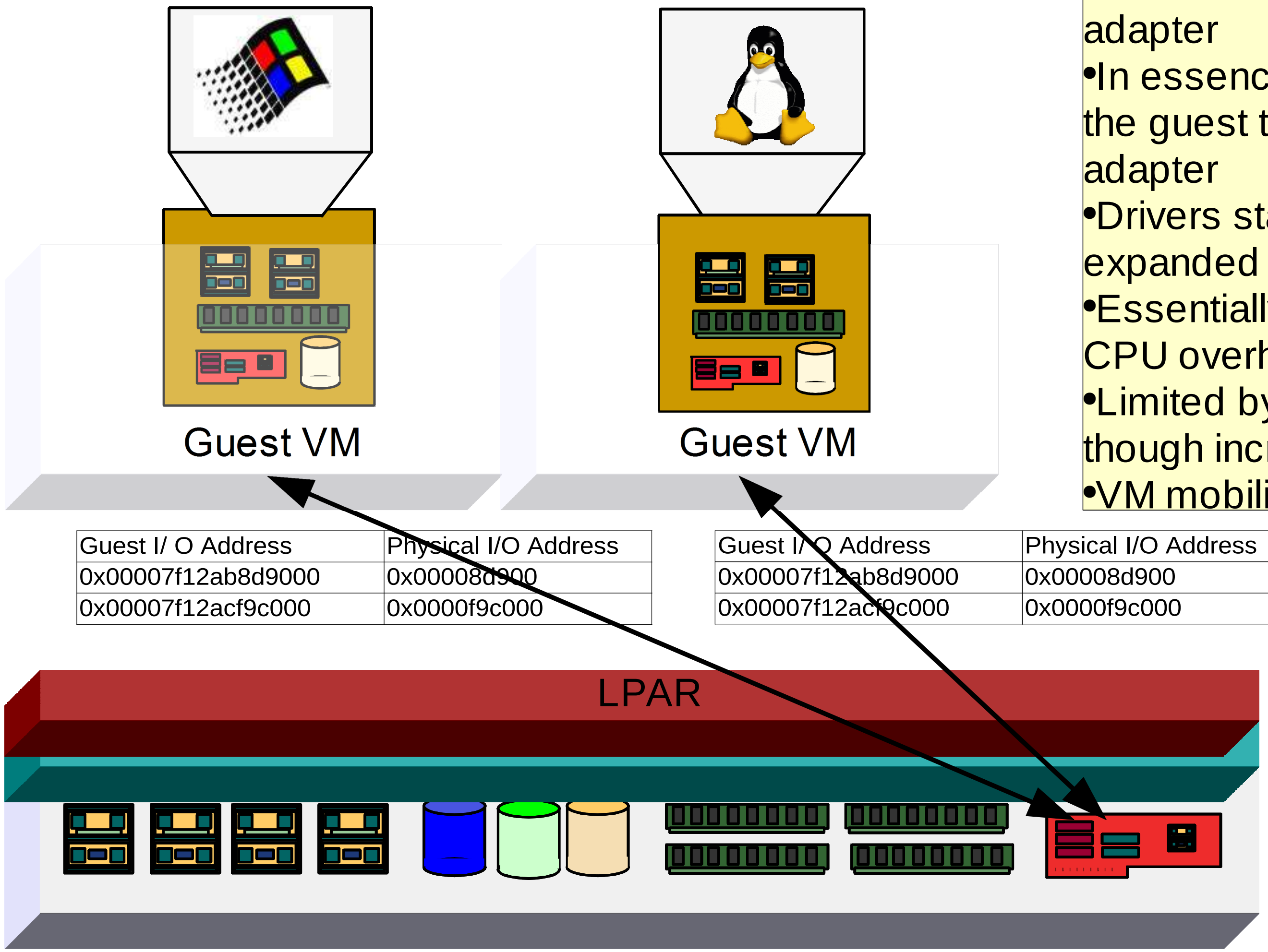
I/O Virtualization – The Current Bottleneck

I/O and Virtualization

- Hardware assisted Virtualization
  - Support for advanced hardware features for both KVM and Xen
    - **VT-d** for secure PCI Pass-thru on Intel platforms
    - **IOMMU** for secure PCI Pass-thru on AMD platforms
    - PCI Single-Root I/O Virtualization (**SR-IOV**)
  - Delivers native I/O performance for network and block devices
- Emulated I/O
- Paravirtualized Drivers for KVM/Linux
  - virtio was chosen to be the main platform for IO virtualization in KVM
  - The idea behind it is to have a common framework for hypervisors for IO virtualization (same in XEN)
  - network/block/balloon/PCI passthrough devices are supported for KVM
  - The host implementation is in userspace - qemu, so no driver is needed in the host (but still has some performance issues)
- Support for Microsoft Windows Servers guests
  - Paravirtualized drivers for network and disk (WHQL certified -> Enterprise Distros)
  - Microsoft SVVP Certification (-> Enterprise Distros)

Single-root I/O Virtualization (SRIOV)

“Root” in “Single-root” refers to PCI bus and device tree



Guest I/ O Address	Physical I/O Address
0x00007f12ab8d9000	0x00008d900
0x00007f12acf9c000	0x0000f9c000

Guest I/ O Address	Physical I/O Address
0x00007f12ab8d9000	0x00008d900
0x00007f12acf9c000	0x0000f9c000

- An SRIOV PCI Device has multiple PCI functions
- Each function behaves like a distinct physical adapter
- In essence, the PCI device virtualizes itself, but the guest thinks it is controlling a dedicated I/O adapter
- Drivers started to appear with RHEL 5.4, expanded with 5.5, and growing...
- Essentially native performance with minimal CPU overhead
- Limited by number of Virtual Functions (Vfs) – though increasing in latest gen adapters
- VM mobility still needs to be addressed

I/O Paravirtualization

- KVM Community in general prefers paravirtualized I/O
  - Performance can be comparable to direct pass-through
  - More flexible
    - Live Guest Migration
    - Integrated virtual switching
  - Hypervisor can optimize I/O scheduling to meet different performance or resource goals
  - SR and MR -IOV hardware can be paravirtualized in creative ways

Virtio

- First proposed by Rusty Russell
  - Based on our experiences with Xen frontend/backend architecture

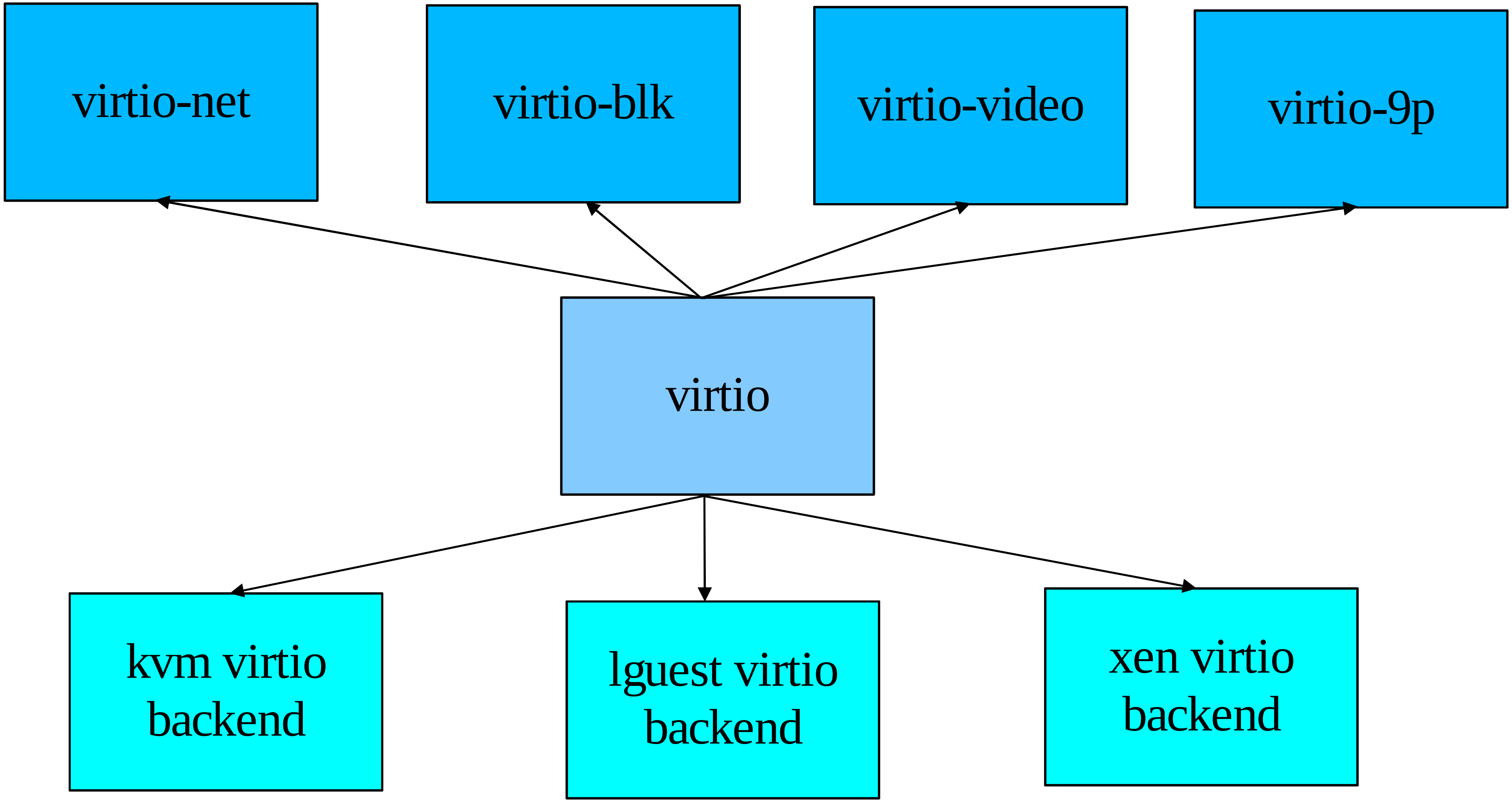
- virtio is an abstraction of the common mechanism of VMMs
  - A single driver could, with little modification, run on many different VMMs

- Addressed a number of concerns:
  - Clear separation between protocol and transport to allow multiple hypervisors to utilize
  - Each component uses well defined interface and is replaceable
  - Minimum driver implementation required
  - Fits on top of existing hardware abstraction well (PCI)

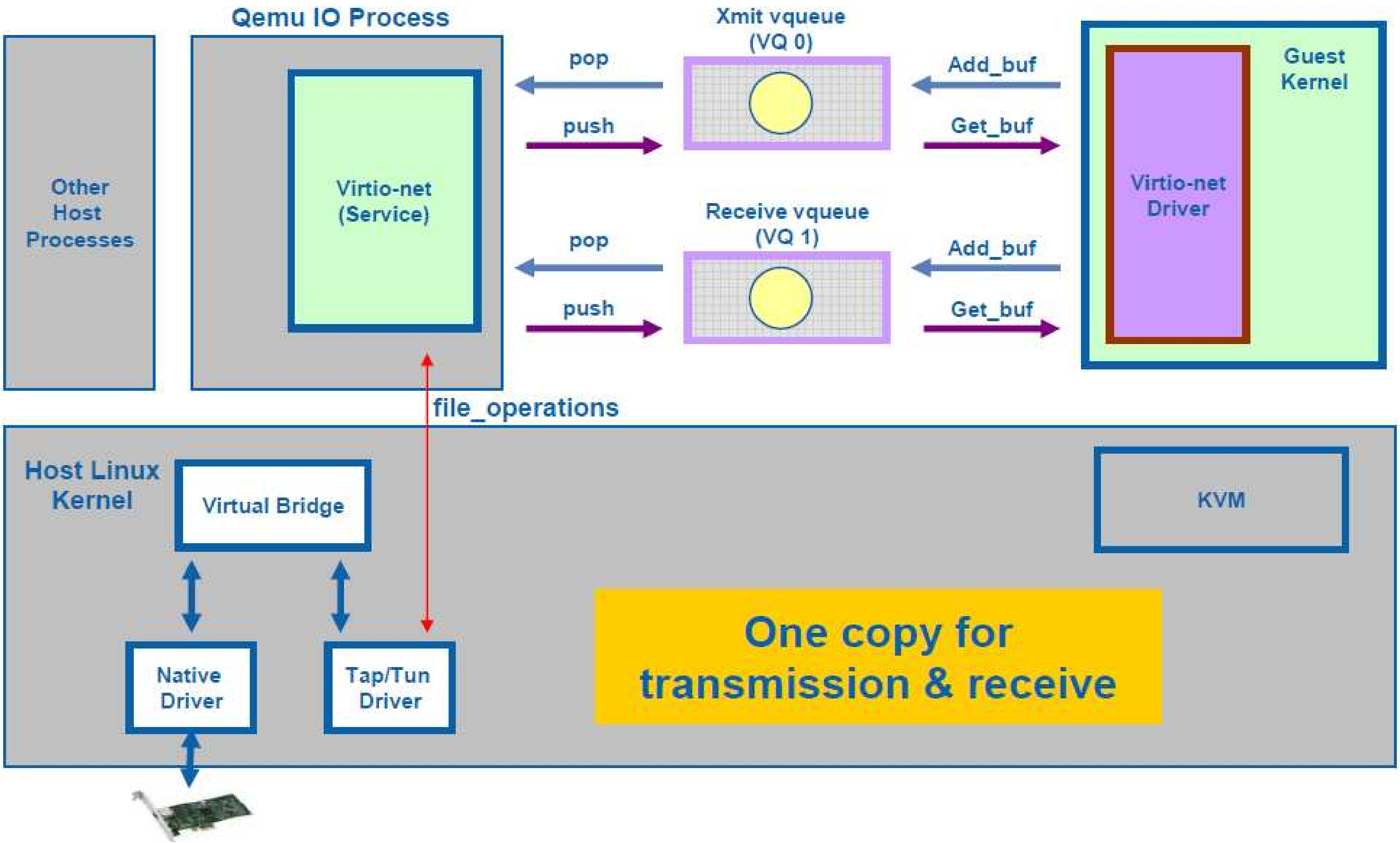
- Linux will support lguest, KVM, Xen, KVM-lite, PHYP, VMware, Viridian, and possibly more
  - If each has 4-5 PV drivers, that's 35 new drivers!
  - All drivers would be doing the same thing

- Especially important for “small” drivers (entropy driver, CPU hotplug, ballooning, etc.)

Virtio Architecture

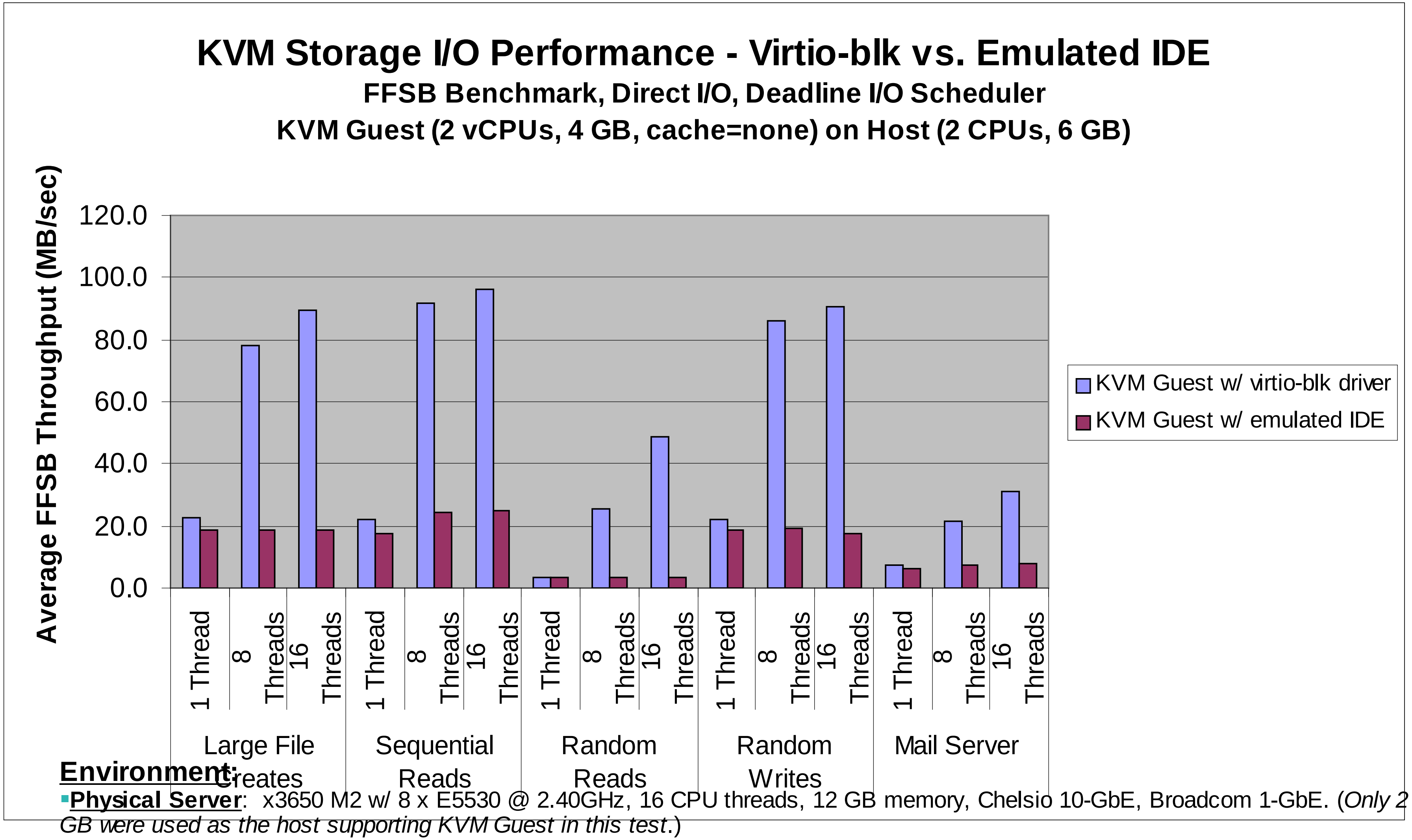


Virtio-net



One copy for transmission & receive

Virtio-block vs Emulated IDE



Environment:

- **Physical Server**: x3650 M2 w/ 8 x E5530 @ 2.40GHz, 16 CPU threads, 12 GB memory, Chelsio 10-GbE, Broadcom 1-GbE. (*Only 2 cores and 6 GB were used as the host supporting KVM Guest in this test*.)
- **Storage**: 8 x 24-disk RAID10 arrays; 4 x DS3400 controllers w/ 4-gbps host fiber links; *a single LVM volume was created across all disk arrays, then formatted with ext3 filesystem, and passed to KVM guest as a block device (/dev/vda).*
- **Host OS**: RHEL 5.5 GA
- **KVM Guest OS**: RHEL5.5 GA

virtio-9p

- A lot of work has focused on block devices, virtio-9p provides a paravirtual file system interface for guests
- Use 9p over virtio and v9fs within the guest
- Able to boot a RHEL5 guest from a v9fs root file system
- virtio-9p transport is in mainline Linux since 2.6.27
- Without any optimization, already able to beat NFS over virtio-net
- A great deal of additional optimizations are possible

Agenda

- Background / History and Red Hat Partnership
- KVM and Cloud Requirements
- IO in Virtualized Environment
- **Memory Resources**
- EX5 Systems : Designed with Virtualization in mind

Cloud is Driving KVM Development...

- Physical Resource Over-provisioning
  - As long as guests don't experience peak load concurrently, we can “borrow” compute, I/O, and memory resources from one guest and “loan” them to another guest
  - Transparent memory sharing
  - Memory “Ballooning” (memory borrowing)
  - Host memory swapping
  - VCPU over-provisioning
    - Virtual CPUs > physical CPUs
- In best cases, resources can be highly leveraged

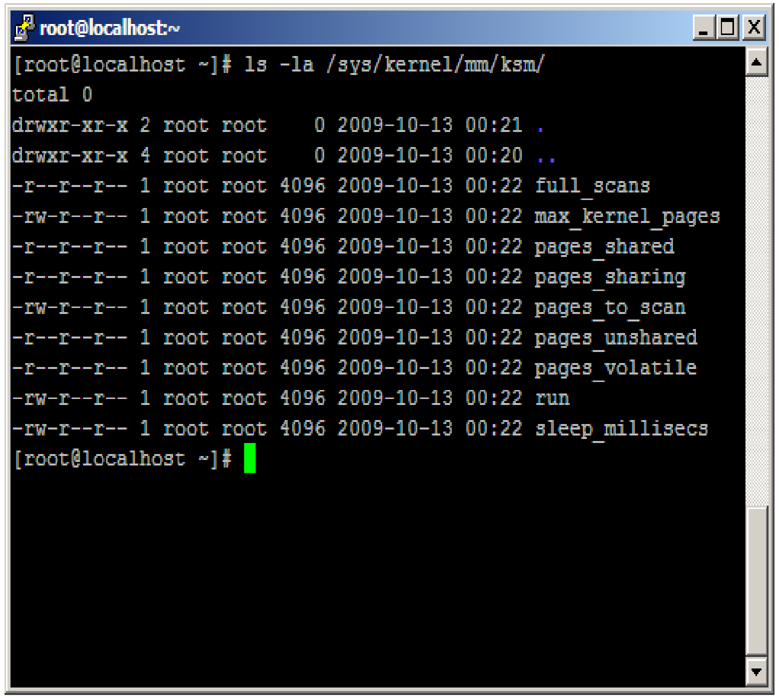
KSM - Memory Page Sharing

- Implemented as loadable kernel module
  - Kernel SamePage Merging (KSM) included in Linux Kernel 2.6.32 (Izik Eidus)

- Kernel scans memory of virtual machines
  - Looks for identical pages
  - Merges identical pages
  - Only stores one copy (read only) of shared memory
  - If a guest changes the page it gets it's own private copy

- qemu-kvm KSM-patch added to kvm development tree after kvm-88 release

- Significant hardware savings
  - Better consolidation ratio
  - Allows more virtual machines to run per host
- Memory Overcommit (avoiding Linux Swapping)



```
[root@localhost ~]# ls -la /sys/kernel/mm/ksm/
total 0
drwxr-xr-x 2 root root    0 2009-10-13 00:21 .
drwxr-xr-x 4 root root    0 2009-10-13 00:20 ..
-r--r--r-- 1 root root 4096 2009-10-13 00:22 full_scans
-rw-r--r-- 1 root root 4096 2009-10-13 00:22 max_kernel_pages
-r--r--r-- 1 root root 4096 2009-10-13 00:22 pages_shared
-r--r--r-- 1 root root 4096 2009-10-13 00:22 pages_sharing
-rw-r--r-- 1 root root 4096 2009-10-13 00:22 pages_to_scan
-r--r--r-- 1 root root 4096 2009-10-13 00:22 pages_unshared
-r--r--r-- 1 root root 4096 2009-10-13 00:22 pages_volatile
-rw-r--r-- 1 root root 4096 2009-10-13 00:22 run
-rw-r--r-- 1 root root 4096 2009-10-13 00:22 sleep_millisecs
[root@localhost ~]#
```

http://www.linux-kvm.com/content/using-ksm-kernel-samepage-merging-kvm

Other Memory overprovisioning...

- Memory “Ballooning”
  - allows the hypervisor to borrow memory pages from one guest and lend those pages to another guest.
  - guest kernel decides which pages it should release for use by another guests
  - implemented in many hypervisors including VMware ESX, z/VM, Xen, and KVM
  - device driver acts like a "balloon" which can be inflated or deflated.
  - guest responds to the "inflation" by freeing memory and giving that memory to the balloon device driver
  - balloon driver hands those memory pages over to KVM, which allows another guest to borrow the memory.
- Host memory swapping
  - Evicting any type of page to a block device extracts a huge performance penalty, to be paid both when the page is evicted, and again when it must be faulted back into memory.
  - Compcache
    - virtual memory manager first evicts a page by compressing it and writing the compressed contents to the compcache device (which is a RAM disk)
    - When the compcache device is full, it de-compresses the oldest pages and writes them to the swap file on secondary storage.

Some simple handwave calculations...

- 2-4 GB / VM
- 2 socket * 8 core * 2 HW threads = 32 ʟCPUs
- Observed average System utilization 10-20%
  - So let's say 5x CPU overprovisioning possible
- 5 Guests/ʟCPU * 32 ʟCPUs * 2-4GB/Guest = 320-640GB
- 320-640GB / 8 GB/DIMM = 40-80 DIMMs
- And many Server workloads utilizing even more memory...

We've covered some software approaches for addressing capacity, but of course one can also use a platform with greater Memory / CPU ratio...

Agenda

- Background / History and Red Hat Partnership
- KVM and Cloud Requirements
- IO in Virtualized Environment
- Memory Resources
- **EX5 Systems : Designed with Virtualization in mind**

IBM System x3850 X5 and Red Hat

Flagship System x platform for leadership scalable performance and capacity

Versatile 4-socket, 4U rack-optimized scalable enterprise server provides a flexible platform for maximum utilization, reliability and performance of compute- and memory-intensive workloads.

Maximize Memory

- 64 threads and 1TB capacity for 3.3x database and 3.6x the virtualization performance over industry 2-socket x86 (Intel Xeon 5500 Series) systems
- MAX5 memory expansion for 50% more virtual machines and leadership database performance
- Run more VMs and larger VMs with RHEV-H

Minimize Cost

- Lower cost, high performance configurations reaching desired memory capacity using less expensive DIMMs
- eXFlash 480k internal IOPs for 40x local database performance and $1.3M savings in equal IOPs storage
- Red Hat Enterprise Virtualization for Servers offers industry-leading performance, scalability, and lower total cost of ownership compared to other virtualization solutions.

Simplify Deployment

- FlexNode Partitioning and Automatic Node failover for maximum flexibility and application uptime
- Pre-defined database and virtualization workload models for faster deployment and faster time to value

System Specifications

- ✓ 4x next-generation Intel Xeon (Nehalem EX) CPUs
- ✓ 64 to 96 DDR3 DIMMs
- ✓ 6 open PCIe slots (+ 2 additional)
- ✓ Up to 8x 2.5” HDDs or 16x 1.8” SSDs
- ✓ RAID 0/1 Std, Optional RAID 5/6
- ✓ 2x 1GB Ethernet LOM
- ✓ 2x 10GB Ethernet SFP+ Virtual Fabric / FCoEE
- ✓ Scalable to 8S, 192 DIMM
- ✓ Internal USB for embedded hypervisor
- ✓ IMM, uEFI & IBM Systems Director

IBM System x3690 X5 and Red Hat

Industry's first high end scalable 2-socket for maximum memory and performance

High-end 2-socket, 2U scalable server offers up to four times the memory capacity of today's 2-socket servers with double the processing cores for unmatched performance and memory capacity.

Maximize Memory

- 33% more cores and 5x more memory capacity for 1.7x more transactions per minute and 2x more RHEV-H virtual machines than 2-socket x86 (Intel Xeon 5500 Series) systems
- MAX5 memory expansion for additional 46% more virtual machines and leadership database performance
- Run more VMs and larger VMs with RHEV-H

Minimize Cost

- Achieve 4-socket memory capacity with 2-socket software license costs and cheaper "2-socket only" processors
- eXFlash 720k internal IOPs for 40x local database performance and $2M savings in equal IOPs storage
- Red Hat Enterprise Virtualization for Servers offers industry-leading performance, scalability, and lower total cost of ownership compared to other virtualization solutions.

Simplify Deployment

- FlexNode Partitioning and Automatic Node failover for maximum flexibility and application uptime
- Pre-defined database and virtualization workload models for faster deployment and faster time to value

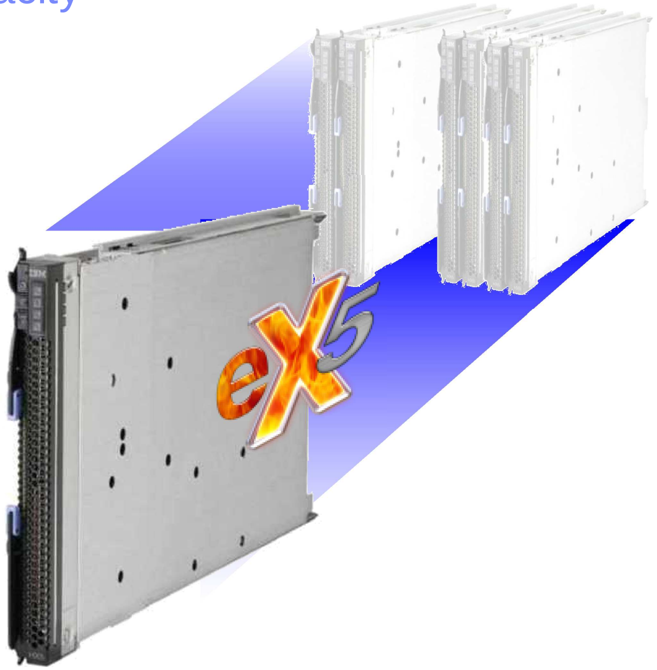
System Specifications

- ✓ 2x next-generation Intel Xeon (Nehalem EX) CPUs
- ✓ 32 to 64 DDR3 DIMMs
- ✓ 2 x8 PCIe slots, 2 x8 Low Profile slots
- ✓ Up to 16x 2.5" HDDs or 32x 1.8" SSDs
- ✓ RAID 0/1 Std, Opt RAID 5
- ✓ 2x 1GB Ethernet
- ✓ Optional 2x 10GB SFP+ Virtual Fabric / FCoEE
- ✓ Scalable to 4S, 64 DIMM or 128 DIMM
- ✓ Internal USB for embedded hypervisor
- ✓ IMM, uEFI, and IBM Systems Director

IBM BladeCenter HX5 and Red Hat

Scalable high end blade for high density compute and memory capacity

Scalable blade server enables standardization on same platform for 2- and 4-socket server needs for faster time to value, while delivering peak performance and productivity in high-density environments.



Maximize Memory

- 1.7x greater performance over 2-socket x86 (Intel Xeon 5500 Series) systems while using same two processor SW license
- MAX5 memory expansion to 320GB in 60mm for over 25% more VMs per processor compared to competition
- Run more VMs and larger VMs with RHEV-H

Minimize Cost

- Upgrade to 80 DIMM for max memory performance or to save over $4K by using smaller, less expensive DIMMs
- Memory bound RHEV-H customers can consolidate more workloads on each blade with memory rich 2-socket configurations

Simplify Deployment

- FlexNode Get up and running up to 2x faster by qualifying a single platform for 2- and 4-socket server needs
- Partitioning of 4-socket to two 2-sockets without any physical system reconfiguration, and automatically fail over for maximum uptime

System Specifications

- ✓ 2x next-generation Intel Xeon (Nehalem EX) CPUs
- ✓ 16x DDR3 VLP DIMMs
- ✓ MAX5 memory expansion to 2S, 40 DIMM
- ✓ Scalable to 4S, 32 DIMM or 4S, 80 DIMM
- ✓ UP to 8 I/O ports and to 2x SSDs per node
- ✓ Optional RAID 5 with battery backed cache
- ✓ Optional 10GB Virtual Fabric Adapter / FCoEE
- ✓ Internal USB for embedded hypervisor
- ✓ IMM, uEFI, and IBM Systems Director

MAX5: Memory Access for eX5

Greater productivity and utilization through memory expansion and flexibility

Take your system to the MAX with ***MAX5***

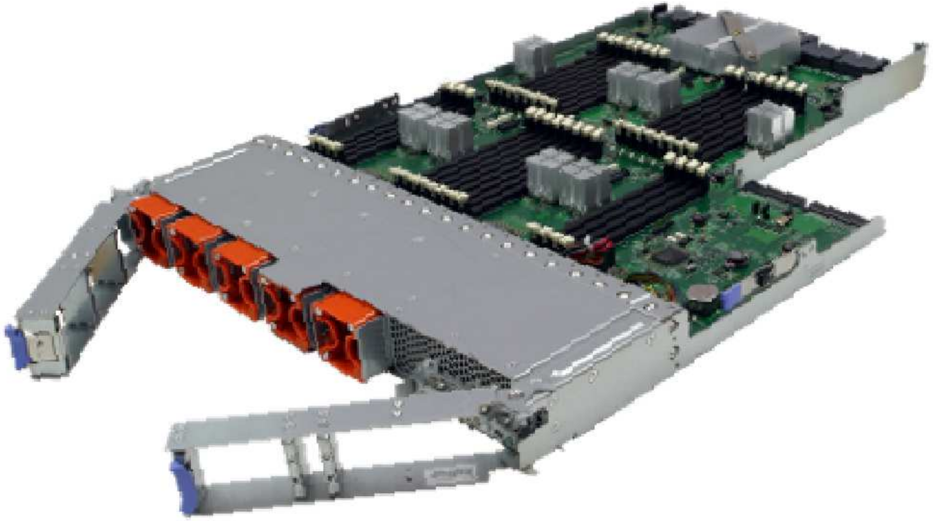
MAX memory capacity

-An additional 32 DIMM slots for x3850 X5 and x3690 X5

-An additional 24 DIMM slots for HX5

MAX virtual density

- Increase the size and number of VMs



MAX flexibility

- Expand memory capacity, scale servers, or both

MAX productivity

- Increase server utilization and performance

MAX license optimization

- Get more done with fewer systems

eX5 Rack System Configurations

Leadership high end x86 performance and flexibility

x3690 X5
(4S 64 DIMM)

Memory Enhanced

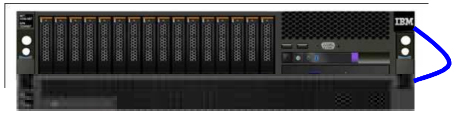
x3690 X5 w/ MAX5
(4S 128 DIMM)

x3850 X5
(8S 128 DIMM)

Memory Enhanced

x3850 X5 w/ MAX5
(8S 192 DIMM)

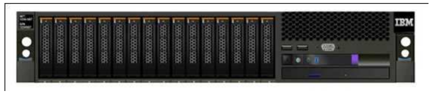
Memory Enhanced



x3690 X5 w/ MAX5
(2S 64 DIMM)

x3850 X5 w/ MAX5
(4S 96 DIMM)

Base Systems



x3690 X5
(2S 32 DIMM)

x3850 X5
(4S 64 DIMM)

IBM BladeCenter Scalable Blades

Maximum performance and flexibility for database and virtualization in a a

HX5 Blade

Never before seen levels of scaling…

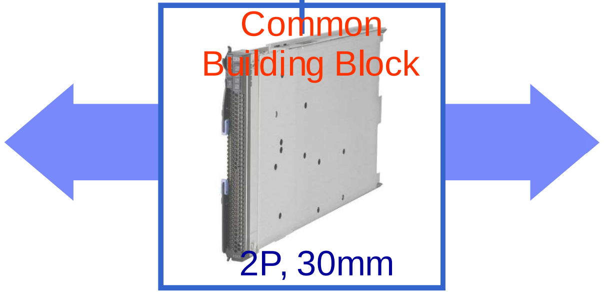
- 2-socket, 30mm building block
- 2-socket → 4-socket w/ logical partitioning

2-socket,
16DIMM
8 I/O ports
30mm

4-socket,
32DIMM
16 I/O
60mm

Max compute density!

- Up to 32 cores in a 1¼ U equivalent space
- Modular scalability in 2-socket increments to get to 4-socket
- *Targeted for database*, and compute intensive simulations



HX5 Blade with MAX5

Bringing the goodness of eX5 to blades…

- Snaps onto base blade (sold as a bundle w/ base HX5)
- Enables more memory than any other blades

2-socket,
40DIMM
8 I/O
60mm

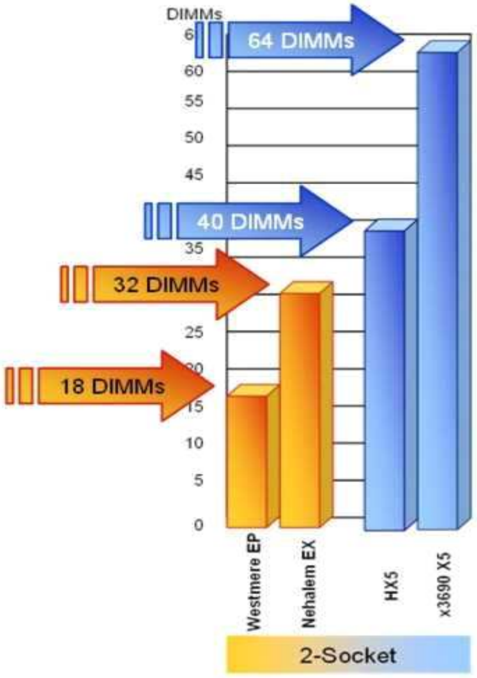
4-socket,
80DIMM
16 I/O
120mm

Blade leadership!

- Up to 30% more VMs than max competition blade
- Flexible configurations & unmatched memory capacity, scaling from 1-socket, 32D → 4-socket, 80D
- Uses processors that cost up to 30% less than the competition for scaling
- *Targeted for Virtualization* & DB for customers that need a blade form factor

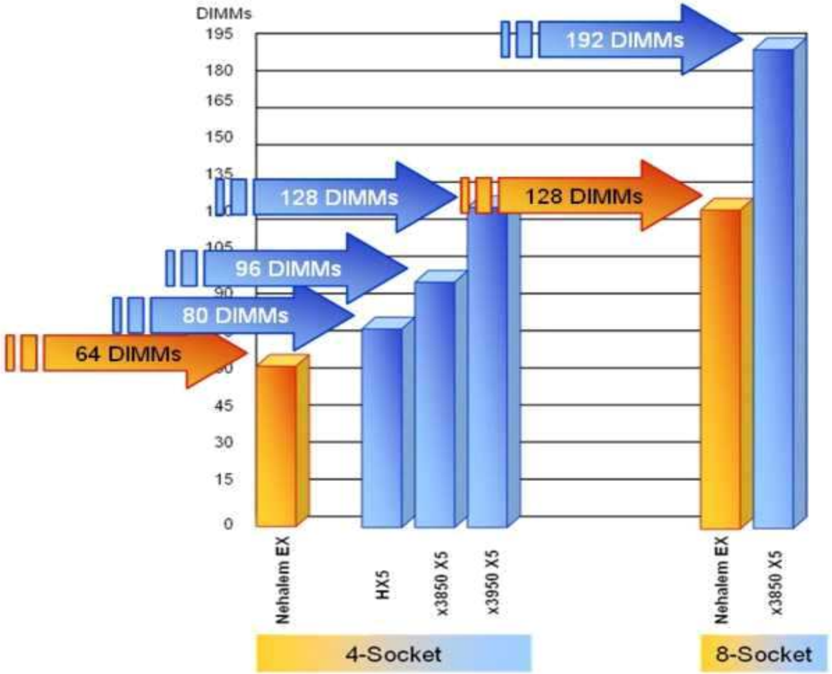
MAX5 for eX5 racks and blades enables systems to support more memory than x86 limits

MAX5 enables up to 2x DIMMs of memory per system



x3690 X5 w/MAX5
(2-socket / 64 DIMMs)

HX5 w/ MAX5
(2-socket / 40 DIMMs)



x3850 X5 w/ MAX5
(8-socket / 192 DIMMs)

x3850 X5 w/ MAX5
(4-socket / 96 DIMMs)

HX5 w/ MAX5
(4-socket / 80 DIMMs)

Thanks for material, input, and lots of work to:

- Frank Novak
- Mike Day
- Thomas Schwaller
- Anthony Liguori
- Ryan Harper
- Andrew Theurer
- Khoa Huynh
- Tom Lendacky
- Badari Pulavarty
- And the larger Virtualization Teams at IBM and Red Hat

Questions ?

Legal

Trademarks and Disclaimers

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries:

IBM, the IBM logo, ibm.com, Smarter Planet and the planet icon, BladeCenter, Power, System Storage, System x, System z, WebSphere, DB2 and Tivoli are trademarks of IBM Corporation in the United States and/or other countries. For a complete list of IBM trademarks, please see www.ibm.com/legal/copytrade.shtml

The following are trademarks or registered trademarks of other companies:

Java and all Java based trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries or both

Microsoft, Windows,Windows NT and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Cell Broadband Engine is a trademark of Sony Computer Entertainment Inc.

InfiniBand is a trademark of the InfiniBand Trade Association.

Other company, product, or service names may be trademarks or service marks of others.

NOTES:
Linux penguin image courtesy of Larry Ewing (lewing@isc.tamu.edu) and The GIMP

Any performance data contained in this document was determined in a controlled environment. Actual results may vary significantly and are dependent on many factors including system hardware configuration and software design and configuration. Some measurements quoted in this document may have been made on development-level systems. There is no guarantee these measurements will be the same on generally-available systems. Users of this document should verify the applicable data for their specific environment.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

Information is provided “AS IS” without warranty of any kind.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices are suggested US list prices and are subject to change without notice. Starting price may not include a hard drive, operating system or other features. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Any proposed use of claims in this presentation outside of the United States must be reviewed by local IBM country counsel prior to such use.

The information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any.