

RHCE LOOPBACK VIRTUAL MEETUP
RHCE LOOPBACK VIRTUAL MEETUP

45,000

RHCE LOOPBACK VIRTUAL MEETUP

RHCE Loopback

Erich Morisse

Senior Solutions Architect
Team Lead, Verticals
emorisse@redhat.com

Performance and Scalability

RHEL5 --> RHEL6

D. John Shakshober (Shak)

Red Hat Performance Engineering
dshaks@redhat.com

Running IT at Red Hat

J Nick Otto

Sr. Director IT Business Systems
notto@redhat.com

Matt Hicks

Manager, Engineering Service Tower
mhicks@redhat.com

Unlocking the Value of the Cloud

Chad Tindel

Manager

Virtualization and Cloud Computing

Solution Architecture

ctindel@redhat.com

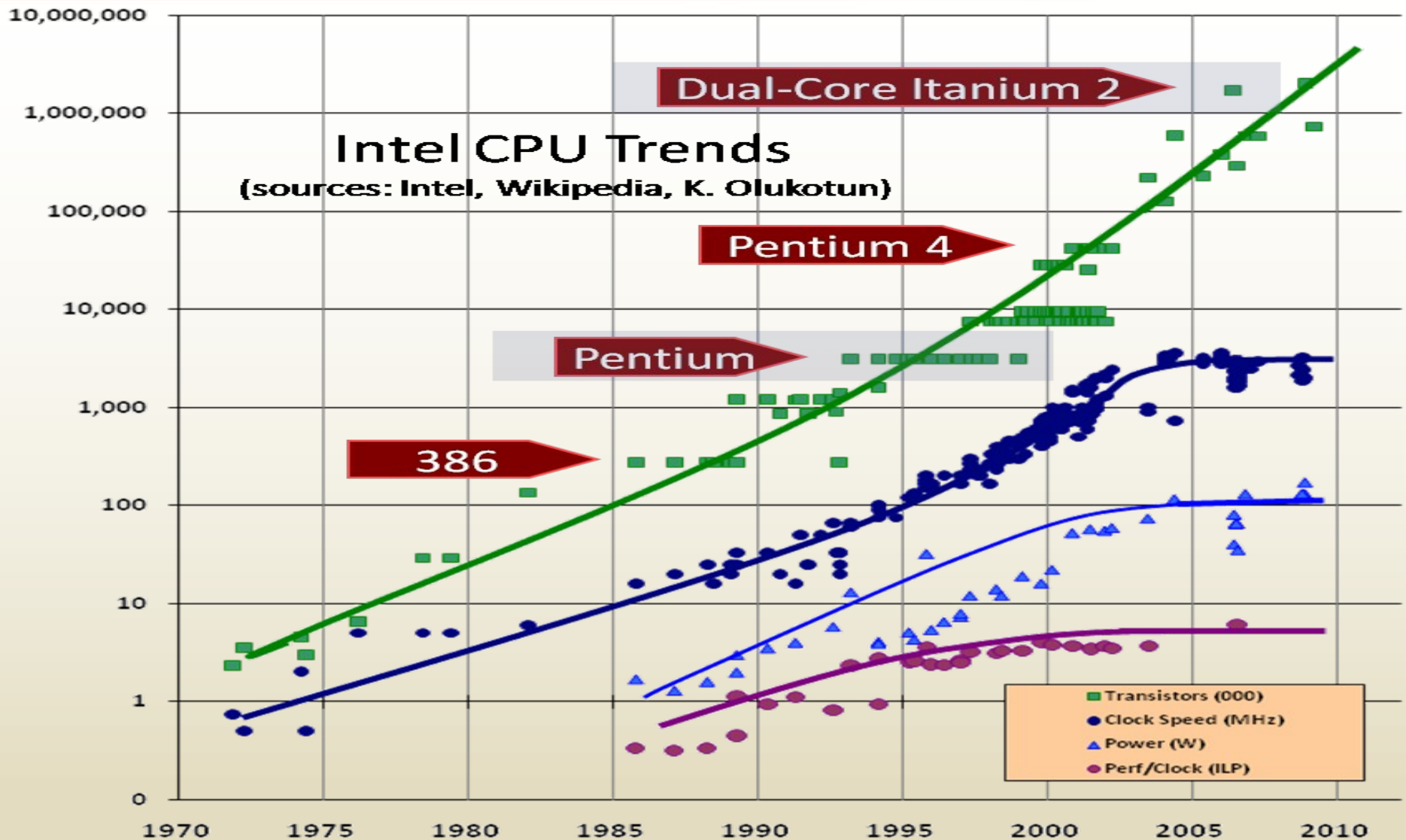


Red Hat Enterprise Linux Performance and Scalability RHEL5 --> RHEL6

September 2010

D. John Shakshober (Shak)
Red Hat Performance Engineering
dshaks@redhat.com

History of x86 Architecture (1985-2010)



RHEL5/6 Performance/Scalability Evolution

- Modular, on-the-fly switchable I/O schedulers (2.6.10)
 - Only provided as a boot option in RHEL4
 - Per-Queue selectable (previously system-wide)
- Conversion to 4-level page tables (2.6.11) (Architecture specific)
 - Allows x86-64 to increase from 512G to 128TB of virtual address space
- "Big Kernel Semaphore": Turns the Big Kernel Lock into a semaphore
 - Latency reduction, by breaking up long lock hold times and adds voluntary preemption
- X86 "SMP alternatives"
 - Optimizes a single kernel image at runtime for UP or SMP operation.
 - Ref: <http://lwn.net/Articles/164121/>

New Hardware

- Huge number of new hardware support features/platforms/drivers/etc
- General features: Multi-core support
 - x86-64 clustered APIC support (2.6.10)
 - Infiniband support (2.6.11) (mostly in RHEL4)
 - Hot plug
 - Generic memory add/remove & supporting functions (2.6.15)
 - (i386) hot plug CPU support of physical add of new processors (hotplug disable/enable of already existing CPUs was already supported)
- SATA/libata enhancements, additional hardware support (in RHEL4)
 - Increased SATA subsystem reliability
 - Increased performance with Native Command Queuing (NCQ) (2.6.18)
 - Hotplug support (2.6.18)
- EDAC (Error Detection & Correction) support (2.6.16) (in RHEL4)
 - Detects and reports errors that occur within the system (ECC memory, PCI bus parity, cache, etc)
- New ioatdma driver for the Intel(R) I/OAT DMA engine (2.6.18)

NUMA & Multi Core

- Cpusets (2.6.12)
 - Enable CPU & Memory assignment to sets of tasks
 - Allow dynamic job placement on large systems
- Numa-aware slab allocator (2.6.14)
 - Optimized locality & management of slab creation
- Swap migration. (2.6.16)
 - Swap migration relocates physical pages between nodes in a NUMA system while the process is running –improves performance
- Huge page support for NUMA (2.6.16)
- Netfilter ip_tables: NUMA-aware allocation (2.6.16)
- Multi-core
 - Scheduler improvements for shared-cache multi-core systems (2.6.17)
 - Scheduler power saving policy
 - Power consumption improvements through optimized task spreading



RHEL6 Scalability Enhancements

- Separate page-lists for anonymous & pagecache pages
- Ticketed spin-locks
- Transparent hugepages
- 1GB hugepage support
- One flush daemon per bdi/filesystem
- Finer grained tuning for very large memory systems
 - memory and cpu cgroups
 - disk and network cgroups

10 Year History of RHEL x86 Architecture (2001-2010)

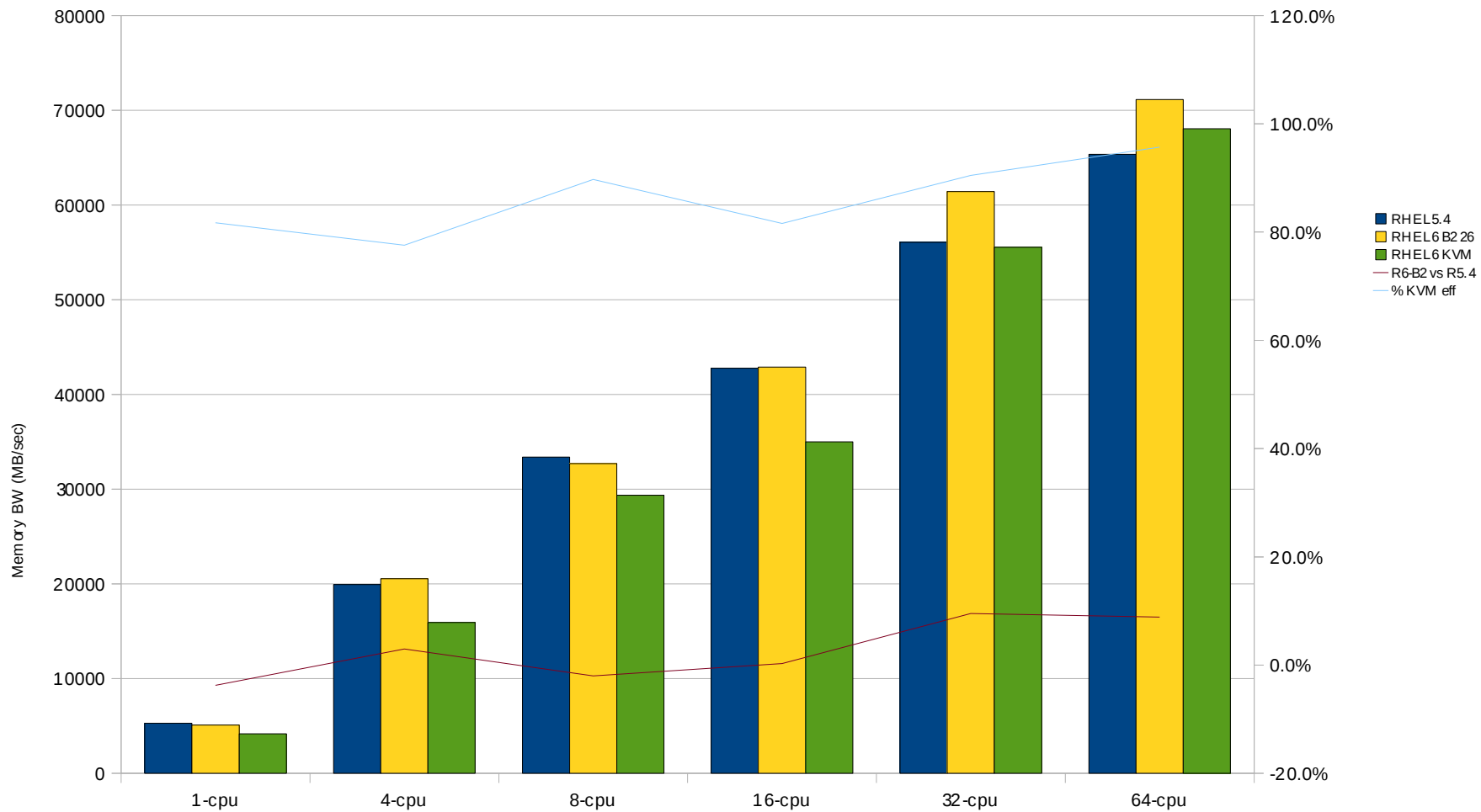
RHEL x86-64 ver	cpu	memory
2.1	4	64gb
3	16	128gb
4	32	256gb
5	255	1tb
6	4096	64tb

<http://www.redhat.com/rhel/compare/>

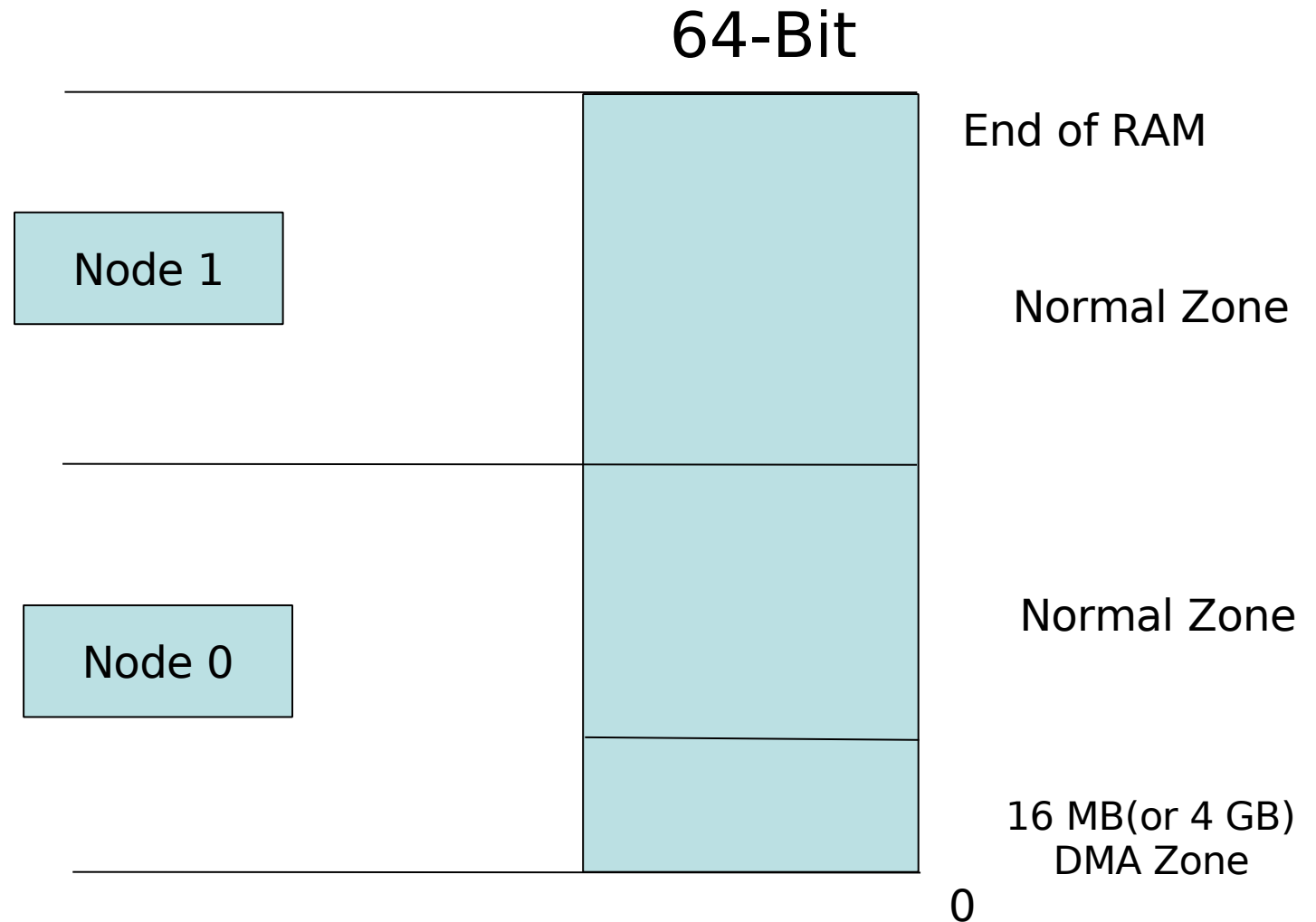
RHEL5/6 CPU Perf Scaling 64 cpu

RHEL6 B2 vs B1 and RHEL5.5 Streams

Intel EX 64-cpu, 128GB, FC



NUMA Nodes and Zones



NUMAstat and NUMActl

NUMAstat to display system NUMA characteristics on a numasystem

```
[root@perf5 ~]# numastat
```

	node3	node2	node1	node0
numa_hit	72684	82215	157244	325444
numa_miss	0	0	0	0
numa_foreign	0	0	0	0
interleave_hit	2668	2431	2763	2699
local_node	67306	77456	152115	324733
other_node	5378	4759	5129	711

NUMActl to control process and memory”

```
numactl [ --interleave nodes ] [ --preferred node ] [ --membind nodes ]  
[ --cpubind nodes ] [ --localalloc ] command {arguments ...}
```

TIP

App < memory single NUMA zone

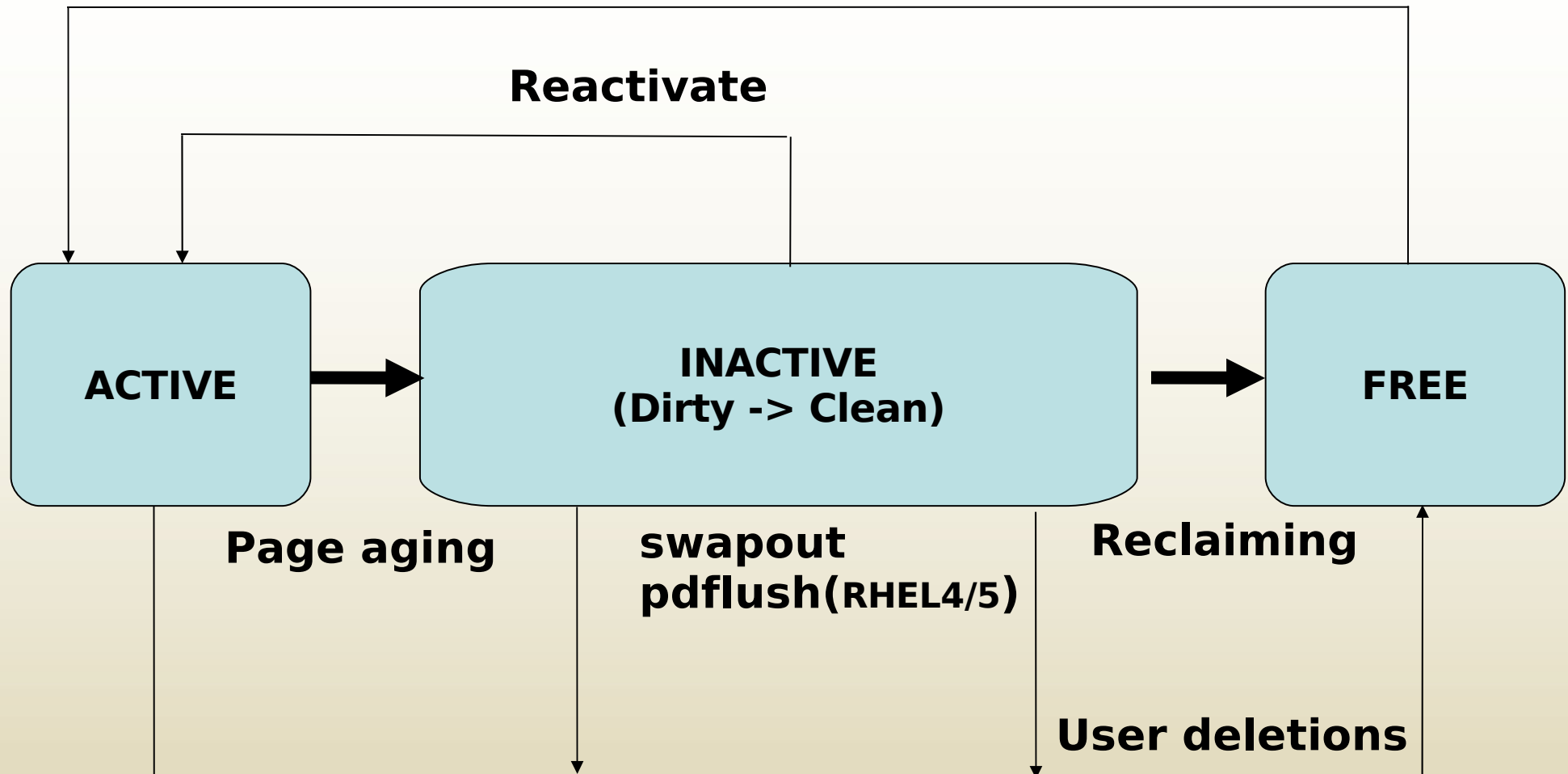
Numactl use -cpubind cpus within same socket

App > memory of a single NUMA zone

Numactl -interleave XY and -cpubind XY

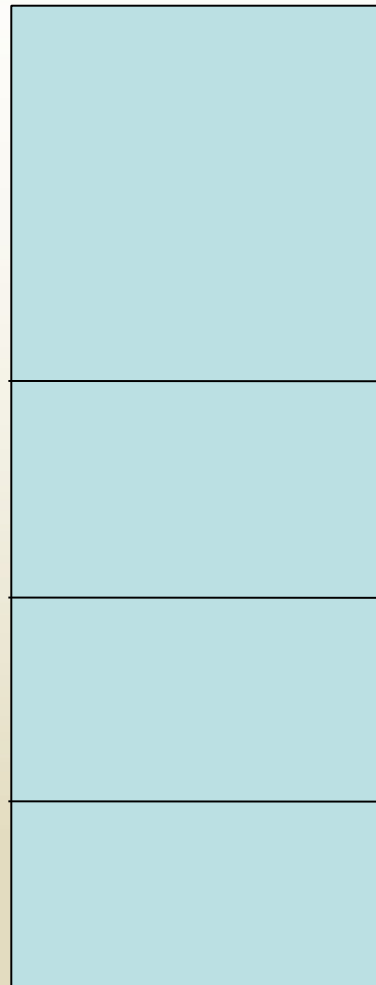
Per Node/Zone Paging Dynamics

User Allocations



Memory reclaim Watermarks

Free List



All of RAM

Do nothing

Pages High – kswapd sleeps above High
kswapd reclaims memory

Pages Low – kswapd wakesup at Low
kswapd reclaims memory

Pages Min – all memory allocators reclaim at Min
user processes/kswapd reclaim memory

0



redhat.

© Red Hat 2010

/proc/sys/vm/swappiness

- Controls how aggressively the system reclaims “mapped” memory:
 - Anonymous memory - swapping
 - Mapped file pages – writing if dirty and freeing
 - System V shared memory - swapping
 - Decreasing: more aggressive reclaiming of unmapped pagecache memory
 - Increasing: more aggressive swapping of mapped memory

/proc/sys/vm/swappiness

Sybase server with /proc/sys/vm/swappiness set to 60(default)

```
procs -----memory----- ---swap--  -----io----- --system--  -----cpu-----
r  b   swpd   free   buff  cache   si   so    bi    bo    in    cs   us  sy  id  wa
5  1  643644  26788   3544 32341788 880  120   4044 7496  1302 20846 25 34 25 16
```

Sybase server with /proc/sys/vm/swappiness set to 10

```
procs -----memory----- ---swap--  -----io----- --system--  -----cpu-----
r  b   swpd   free   buff  cache   si   so    bi    bo    in    cs   us  sy  id  wa
8  3     0   24228   6724 32280696  0    0   23888 63776  1286 20020 24 38 13
26
```

dirty_ratio and dirty_background_ratio

pagecache



100% of pagecache RAM dirty

pdflushd and write()'ng processes write dirty buffers

dirty_ratio(40% of RAM dirty) – processes start synchronous writes

pdflushd writes dirty buffers in background

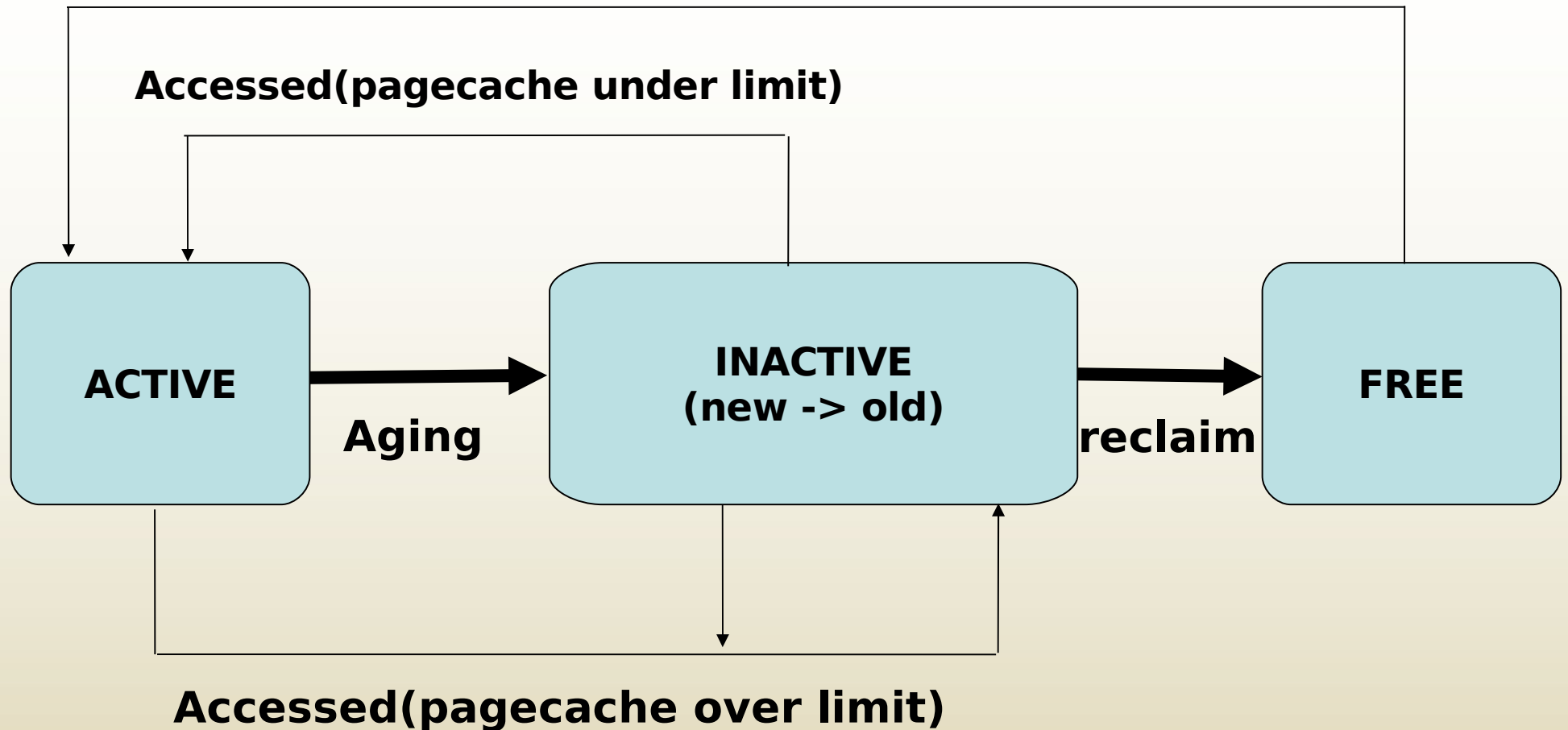
dirty_background_ratio(10% of RAM dirty) – wakeup pdflushd

do_nothing

0% of pagecache RAM dirty

Pagecache Tuning

Filesystem/pagecache Allocation



RHEL6 Control Groups (cgroups)

■ 1GB/2CPU subset of a 16GB/8CPU system

- `#mount -t cgroup xxx /cgroups`
- `#mkdir -p /cgroups/test`
- `#cd /cgroups/test`
- `#echo 1 > cpuset.mems`
- `#echo 2-3 > cpuset.cpus`
- `#echo 1000000000 > memory.limit_in_bytes`
- `#echo $$ > tasks`



cgroups

- `[root@dhcp-100-19-50 ~]# forkoff 10GB 100procs &`
- `[root@dhcp-100-19-50 ~]# top -d 5`
- `top - 12:24:13 up 1:36, 4 users, load average: 22.70, 5.32, 1.79`
- `Tasks: 315 total, 93 running, 222 sleeping, 0 stopped, 0 zombie`
- `Cpu0 : 0.0%us, 0.2%sy, 0.0%ni, 99.8%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st`
- `Cpu1 : 0.0%us, 0.2%sy, 0.0%ni, 99.8%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st`
- `Cpu2 :100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st`
- `Cpu3 : 89.6%us, 10.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.2%hi, 0.2%si, 0.0%st`
- `Cpu4 : 0.4%us, 0.6%sy, 0.0%ni, 98.8%id, 0.0%wa, 0.0%hi, 0.2%si, 0.0%st`
- `Cpu5 : 0.4%us, 0.0%sy, 0.0%ni, 99.2%id, 0.0%wa, 0.0%hi, 0.4%si, 0.0%st`
- `Cpu6 : 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st`
- `Cpu7 : 0.0%us, 0.0%sy, 0.0%ni, 99.8%id, 0.0%wa, 0.0%hi, 0.2%si, 0.0%st`
- `Mem: 16469476k total, 1993064k used, 14476412k free, 33740k buffers`
- `Swap: 2031608k total, 185404k used, 1846204k free, 459644k cached`

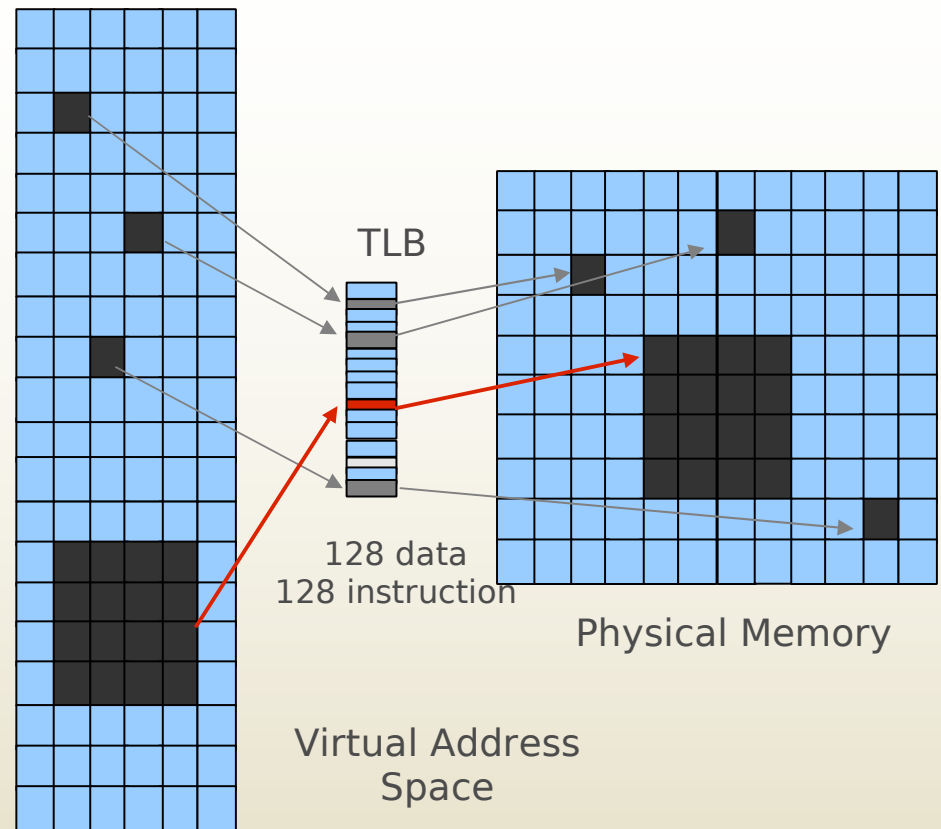


cgroups

```
• [root@dhcp-100-19-50 ~]# memory 2GB &
• [root@dhcp-100-19-50 ~]# vmstat 1
• procs -----memory----- ---swap-- -----io----- --system-- -----cpu-----
•  r  b   swpd   free   buff  cache   si   so   bi   bo   in   cs  us  sy  id  wa  st
•  0  0     0 15465636  33636 459612    5   67   16   68   46   27   1   0  99   0   0
•  0  0     0 15465504  33636 459612    0    0    0    0  246  160   0   0 100   0   0
•  1  0     0 14598736  33636 459612    0    0    0    0 1648  299   1   5  94   0   0
•  1  0 114092 14484980  33636 459528    0 114176    0 114176 2974 1031   0   6  82  12   0
•  0  1 264672 14479896  33636 459508    0 150496    0 150496 2630  568   0   2  90   7   0
•  0  1 375612 14479524  33636 459612    0 110940    0 110940 2301  322   0   4  76  19   0
•  0  1 500064 14477788  33636 459692    0 124452    0 124452 1869  273   0   2  91   7   0
•  1  0 609908 14477540  33636 459628    0 109888    0 109888 1960  198   0   8  76  15   0
•  0  1 709996 14478476  33636 459400    0 100044    0 100044 2243  260   0   3  91   6   0
•  0  1 818924 14478352  33636 459600    0 108928    0 108928 2210  342   0   4  77  18   0
•  0  1 932920 14478476  33636 459548    0 113996    0 113996 1951  303   0   2  91   7   0
•  1  0 1055352 14476864  33636 459516    0 122560    0 122560 1885  197   0   6  76  17   0
```

HugeTLBFS

- The Translation Lookaside Buffer (TLB) is a small CPU cache of recently used virtual to physical address mappings
- TLB misses are extremely expensive on today's very fast, pipelined CPUs
- Large memory applications can incur high TLB miss rates
- HugeTLBs permit memory to be managed in very large segments
- Example: x86_64
 - Standard page: 4KB
 - Huge page: 2MB
 - 512:1 difference
 - File system mapping interface
- Ideal for databases
- Example: 128 entry TLB can fully map 256MB
- * RHEL6 – 1GB hugepage support



Hugepages - using

```
#echo 2000 > /proc/sys/vm/nr_hugepages
```

```
# mount -t hugetlbfs hugetlbfs /huge
```

```
#cp 1GB-file /huge/junk
```

```
# vmstat
```

```
procs -----memory----- ---swap-- -----io----- --system-- -----cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo    in   cs  us  sy  id  wa  st
 0  0       0 10526632 31168 1401780    0    0    129    10  156   63   1   0  98   1   0
```

```
$cat /proc/meminfo
```

```
LowTotal:      16301368 kB
```

```
LowFree:       11524756 kB
```

```
...
```

```
HugePages_Total: 2000
```

```
HugePages_Free: 1488
```

```
HugePages_Rsvd: 0
```

```
Hugepagesize: 2048 kB
```

Hugepages - releasing

```
$rm /huge/junk
$cat /proc/meminfo
MemTotal:      16301368 kB
MemFree:       11524776 kB
...
HugePages_Total: 2000
HugePages_Free: 2000
HugePages_Rsvd: 0
Hugepagesize: 2048 kB
```

```
$echo 0 > /proc/sys/vm/nr_hugepages
```

```
$vmstat
```

```
procs -----memory----- --swap-- -----io----- --system-- -----cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo    in   cs us sy id wa st
 0  0     0 15620488 31512 401944    0    0    71    6   149   59  1  0 98  1  0
```

```
$cat /proc/meminfo
MemTotal:      16301368 kB
MemFree:       15620500 kB
...
HugePages_Total: 0
HugePages_Free: 0
HugePages_Rsvd: 0
Hugepagesize: 2048 kB
```



RHEL 6 Transparent Hugepages

- Boot argument: `transparent_hugepages=always`
- Dynamic:
- `# echo always > /sys/kernel/mm/redhat_transparent_hugepage/enabled`

```
[root@dhcp-100-19-50 code]# time ./memory 15GB
```

```
real 0m7.024s
```

```
user 0m0.073s
```

```
sys 0m6.847s
```

```
[root@dhcp-100-19-50 ~]# cat /proc/meminfo
```

```
...
```

```
AnonHugePages: 15572992 kB
```

```
...
```



Transparent Hugepages

- `echo never > /sys/kernel/mm/transparent_hugepages=never`

```
[root@dhcp-100-19-50 code]# time ./memory 15 0
```

```
real 0m12.434s
```

```
user 0m0.936s
```

```
sys 0m11.416s
```

```
[root@dhcp-100-19-50 ~]# cat /proc/meminfo
```

```
AnonHugePages: 0 kB
```

SPEEDUP $12.4/7.0 = 1.77x$, 56%



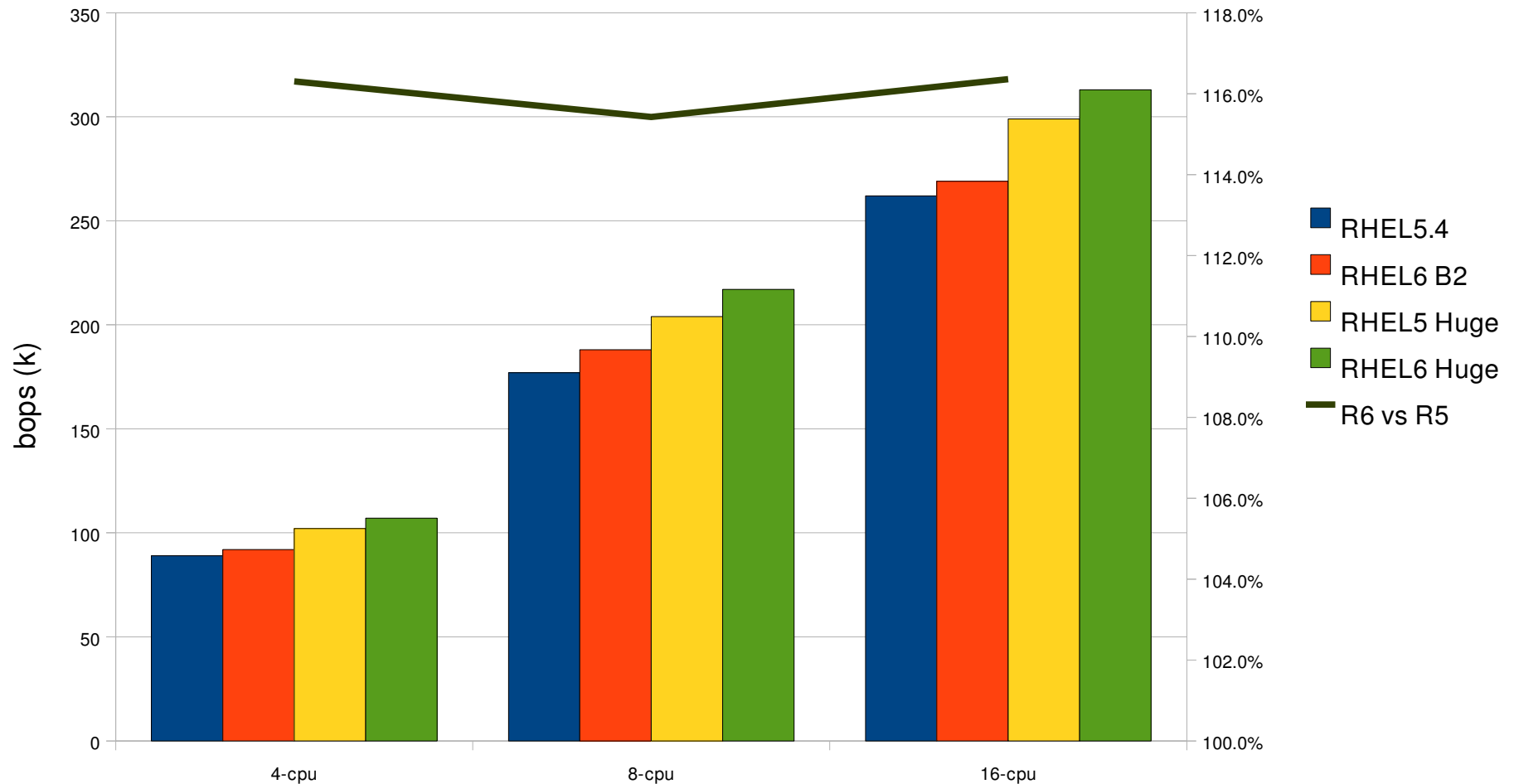
Finer grained tuning / scheduler

- **`/proc/sys/kernel/sched_*` to increase quantum on par with RHEL5**
 - `echo 10000000 > /proc/sys/kernel/sched_min_granularity_ns`
 - `echo 15000000 > /proc/sys/kernel/sched_wakeup_granularity_ns`
 - `echo 80000000 > /proc/sys/kernel/sched_latency_ns`
 - `echo 15834234 > /proc/sys/kernel/sched_features`



Performance – RHEL6 B2 Linux Intel EX Specjbb Java – Huge/Transparent Huge Pages

RHEL5.5 /6 SPECjbb Scaling Intel EX



RHEL6 Technology Innovation

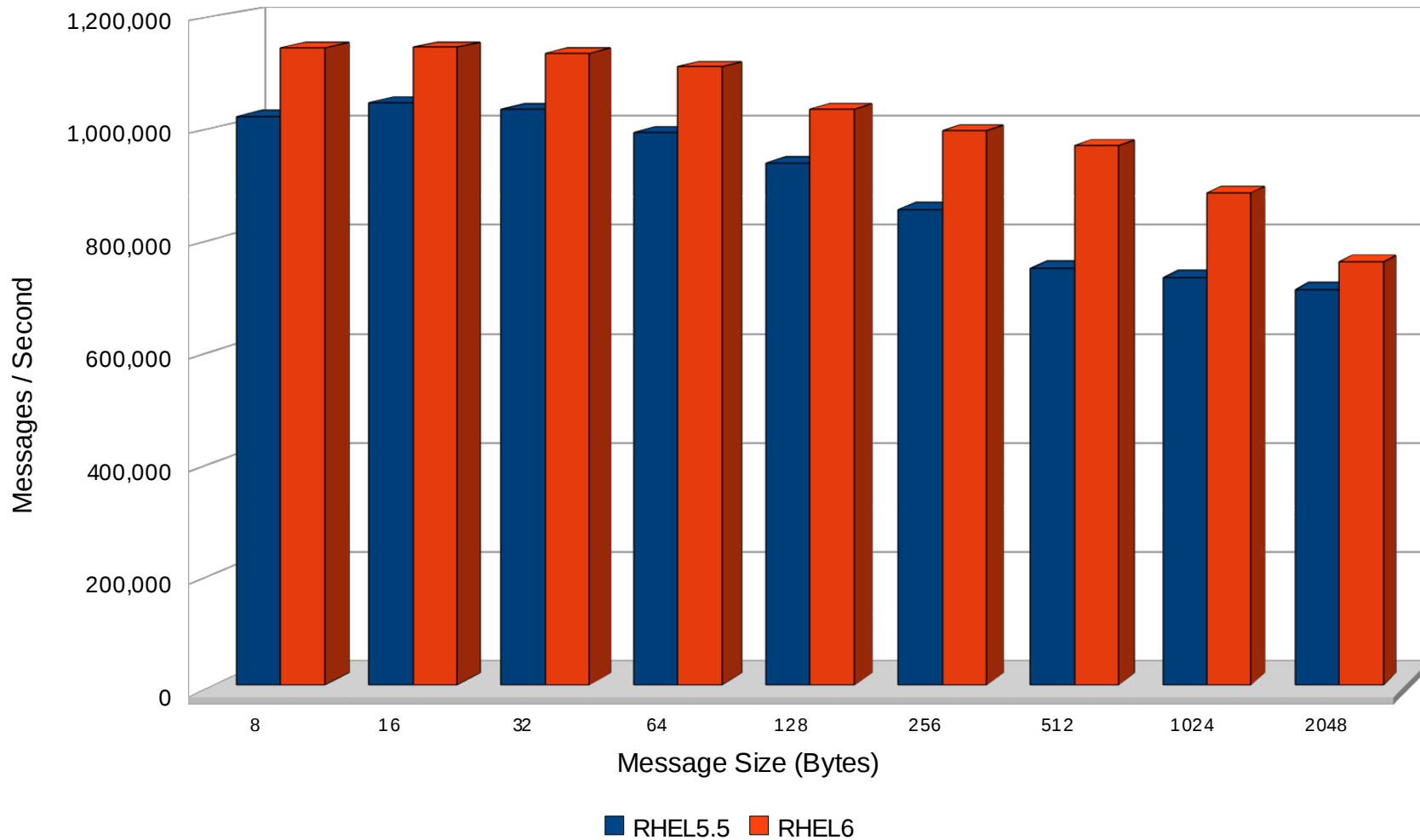
■ Networking

- Multi-queue
- Tools to monitor dropped packets –tc, dropwatch.
- RCU adoption in stack
- Multi-CPU receive to pull in from the wire faster.
- 10GbE driver improvements.
- Data center bridging in ixgbe driver.
- FcoE performance improvements throughout the stack.

RHEL5.5 to RHEL6 AMQP TCP mess/sec (Bigger=Better)

RHEL5 vs RHEL6 (preliminary)

Message Rates

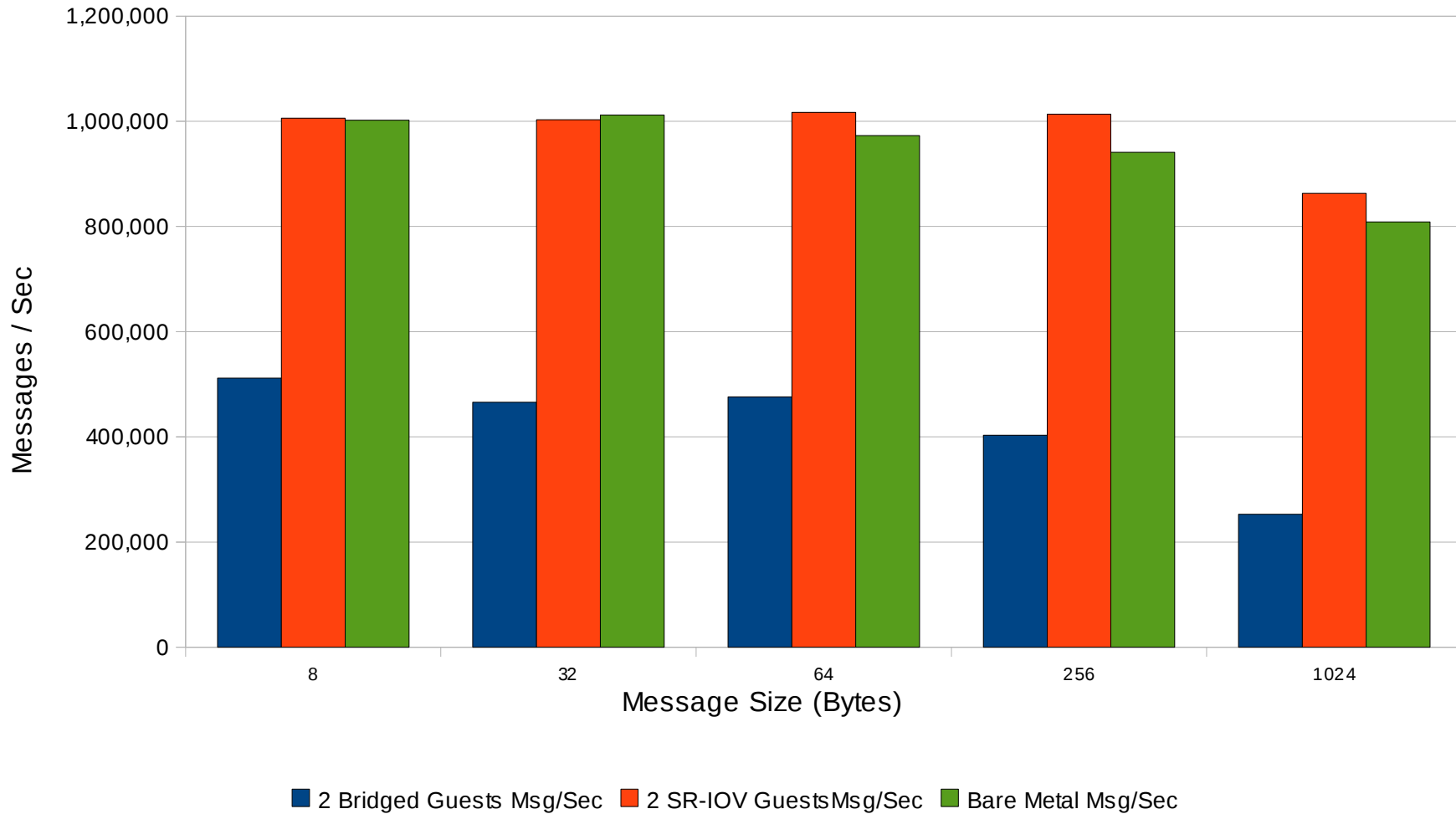


10 Gbit Ethernet (Mellanox)

MRG 1.3 / AMQP RHEL6 KVM w/ SRIOV virtualized (Throughput KVM ~5% bare metal - Bigger=better)

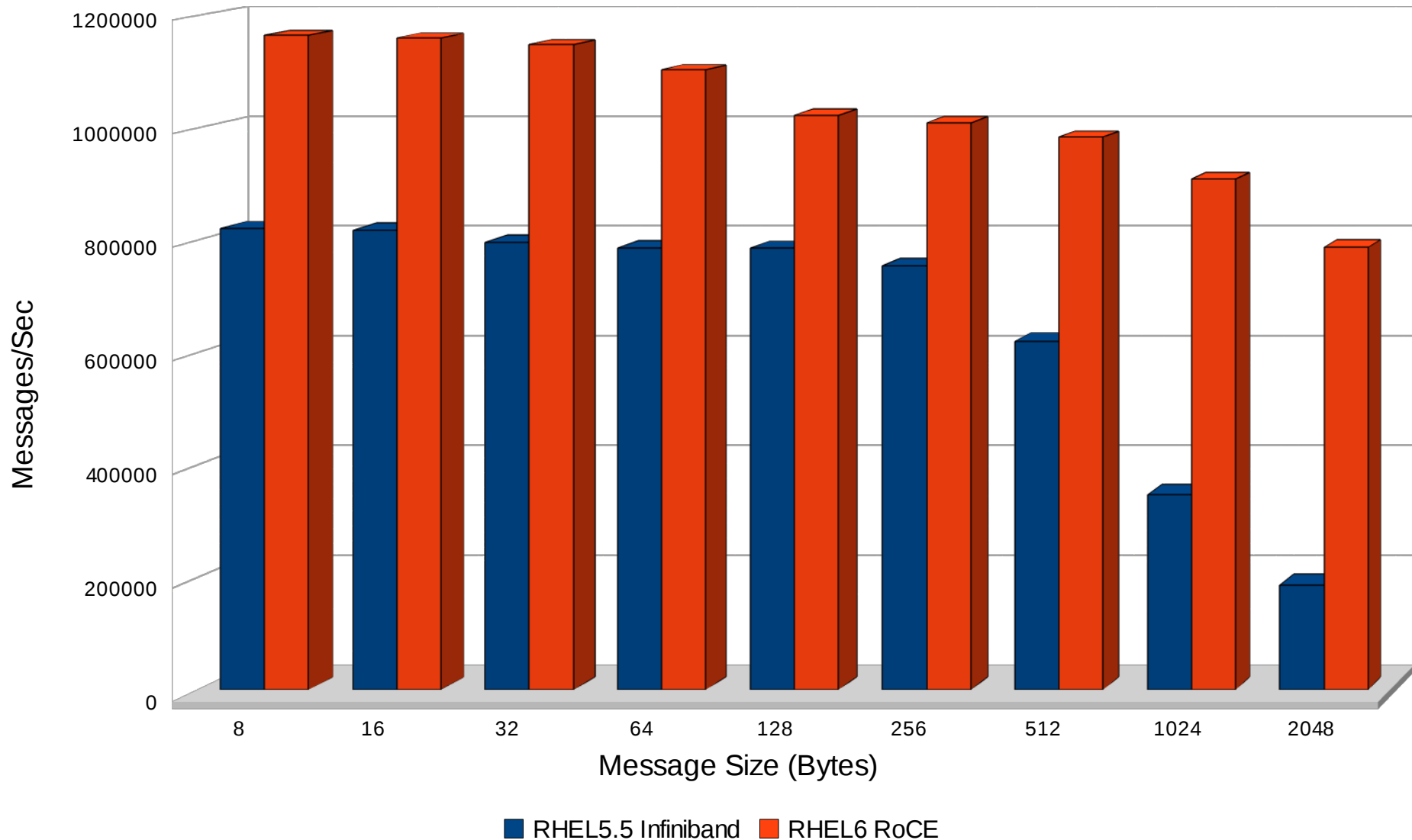
Perftest - Bare Metal and KVM

Message Rates with Different Technologies



RHEL6 IB vs 10Gb RDMA over Converged Ethernet (RoCE) (Messages / Sec - Bigger = Better)

Comparing RHEL55 Mellanox Infiniband and RHEL6 Mellanox 10Gb with RoCE

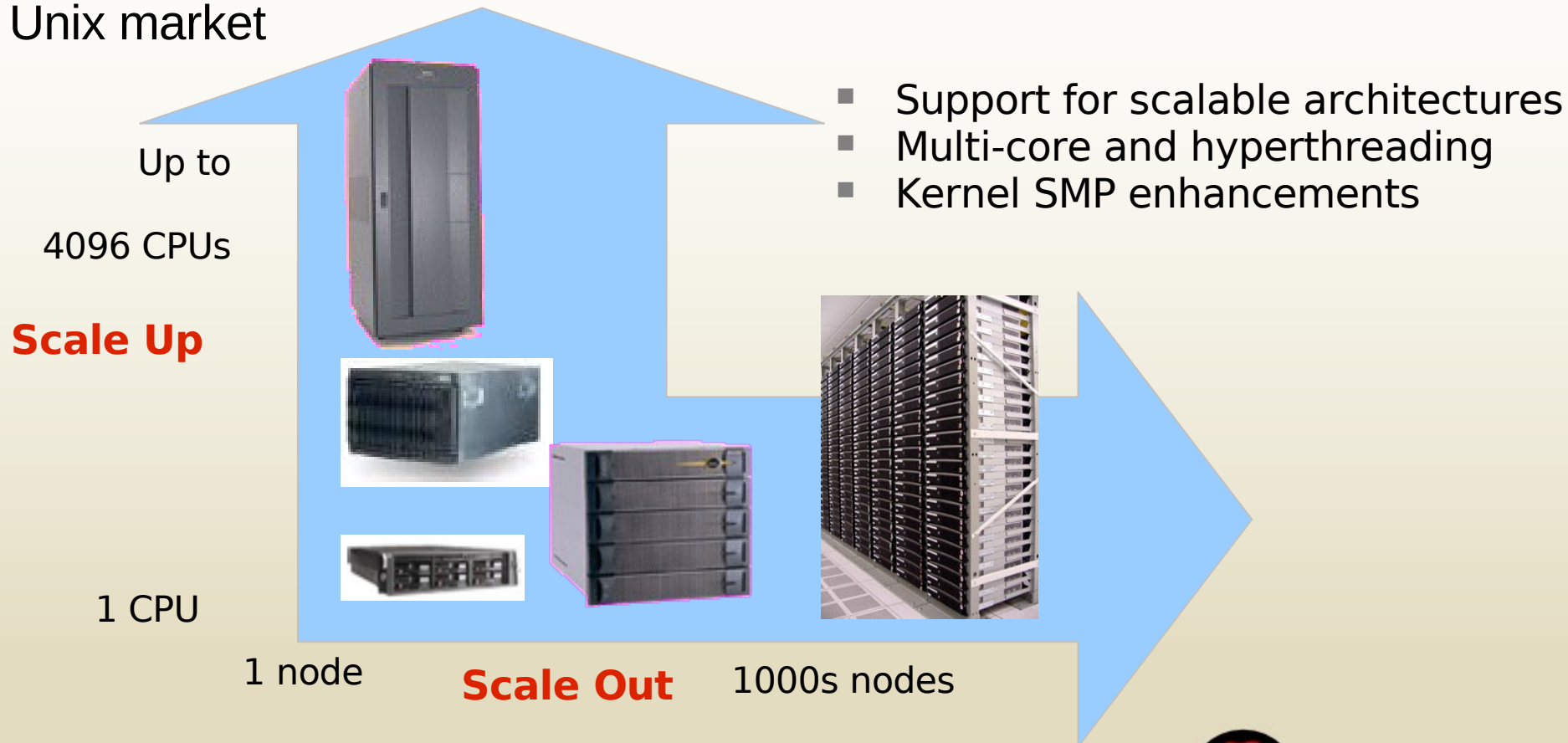


Tuning

- Tuning can provide excellent improvements
- Steps are different for throughput vs latency, goal is the same.
 - Try to maximize CPU cache hits and localize memory
 - Use NUMA if possible
 - *numactl -c1 -m1 /root/qpid/cpp/src/qpidd --auth no -m no --pid-dir /var/run/qpidd --data-dir /var/lib/qpidd --load-module /root/qpid/cpp/src/.libs/rdma.so -P rdma*
 - Move IRQ handlers as needed
 - Understand the NIC parameters, tune as necessary

Red Hat Enterprise Linux: Scale Up & Out

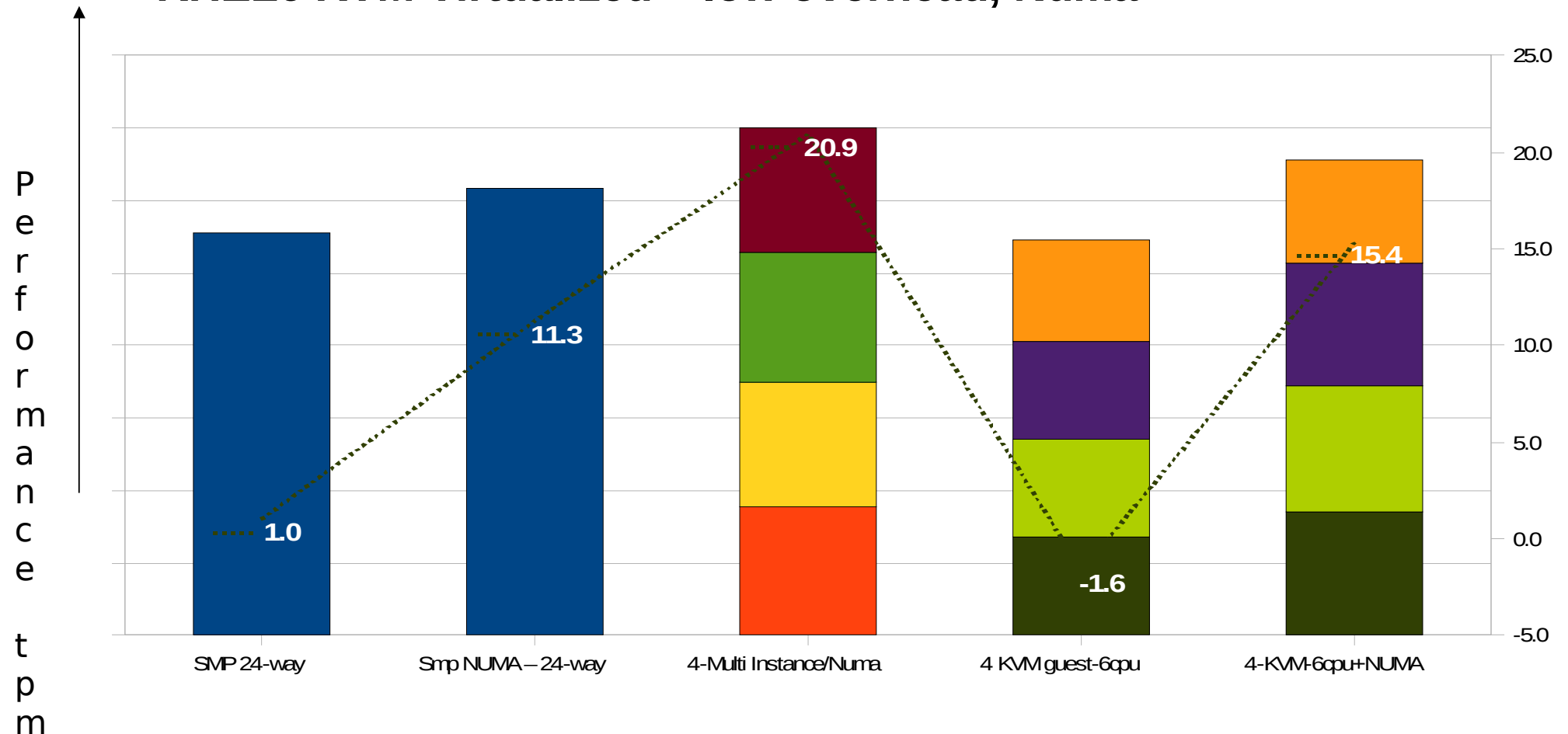
- Traditional scale-out capabilities have been complemented over the past two years with scale-up capabilities
 - Brings open source value and flexibility to the traditional large Unix market





RHEL6 extends traditional SMP scaling AMD 24

- RHEL6 Database Performance in (tpm) w/ Numa
- RHEL6 Multi-instance DB > SMP, uses Numa
- RHEL6 KVM Virtualized – low overhead, Numa



Summary RHEL6 extends x86_64 performance improved

- CPU
 - Ticketed Locks – order locks for numa hierachy
 - Completely Fair Scheduler – tickless timers
- Memory
 - Non-Uniform memory (NUMA) improvement, split LRU
 - Transparent HugePages – dynamically choose hugepages
- Control Groups (Cgroups) – cpu/memory/disk/network – control sharing
- Disk / Network
 - Flushd per lun,
 - Multi-queue
- Virtualization
 - KVM large smp – upto 64 vcpu
 - Block - Aio-host – new implementaton of Linux kernel async-io w/ KVM
 - Net – vhost-net – KVM virtio network support into kernel – tech preview



redhat.

© Red Hat 2010

Questions?

Running IT at Red Hat

9/27/10

J Nick Otto / notto@redhat.com
Matt Hicks / mhicks@redhat.com



redhat.com

Red Hat IT Overview

Presenters:

J Nick Otto, Sr. Director IT Business Systems

notto@redhat.com

Matt Hicks, Manager, Engineering Service Tower

mhicks@redhat.com

-
- 65+ offices, 80+ labs, 3 DCs
 - 1,200 – 1,500 managed servers
 - Over 500 TB of storage
 - Over 500 core network devices, 15,000 network endpoints
 - 12 call centers
 - Over 2 million emails per day
 - Over 3,500 Accounts
 - 150+ Enterprise Applications
-



Red Hat Linux *is* Enterprise

The *value* of an integrated platform...



Infrastructure Implementation

Segmentation

Red Hat Enterprise Linux

Integration

JBoss Enterprise Application Platform

JBoss Enterprise SOA Platform

Scale

Red Hat Enterprise Virtualization



Value of RHCE Certified Associates

- Understanding of fundamentals
- Hands on experience
- Value the platform and the brand
- RHCE - just the beginning

Practical IT: Making Good IT Decisions*

1. Is it proven and sustainable?
 2. Does it adhere to Open Standards?
 3. Will it scale to meet your needs? Is it predictable?
 4. Is it easy to manage?
1. Will it work with what you are currently running?
 2. What type of support is offered?
 3. Is it extensible?
 4. Can you have a long-term relationship with the vendor?

* <https://www.redhat.com/training/catalog/> [1]

Managers, protect yourself against...



Questions



Attributions

- Photo #1 (pedal board) - <http://tinyurl.com/3abgue2>
 - Photo #2 (boss board) - <http://tinyurl.com/2qgles>
 - Photo #3 (Crowd) - <http://tinyurl.com/l52wxm>
 - Photo #4 (Bull) - <http://tinyurl.com/35rqnyh>
1. <https://www.redhat.com/training/catalog/> Pages 4 and 5, written by J Nick Otto and Randy Russel.

RHCE LOOPBACK VIRTUAL MEETUP

RHCE Loopback

Erich Morisse

Senior Solutions Architect
Team Lead, Verticals
emorisse@redhat.com

Performance and Scalability

RHEL5 --> RHEL6

D. John Shakshober (Shak)
Red Hat Performance Engineering
dshaks@redhat.com

Running IT at Red Hat

J Nick Otto

Sr. Director IT Business Systems
notto@redhat.com

Matt Hicks

Manager, Engineering Service Tower
mhicks@redhat.com

Unlocking the Value of the Cloud

Chad Tindel

Manager
Virtualization and Cloud Computing
Solution Architecture
ctindel@redhat.com

RHCE LOOPBACK VIRTUAL MEETUP
RHCE LOOPBACK VIRTUAL MEETUP

Get Involved

Be a presenter at a RHCE Loopback event:
rhceloopback@redhat.com

Talk with other certified experts:
<https://www.redhat.com/training/certification/>

Join a mailing list or three:
<http://www.redhat.com/mailman/listinfo>

Join us on Facebook

You tell us: rhceloopback@redhat.com