RED HAT SUMMIT

LEARN. NETWORK. EXPERIENCE OPEN SOURCE.

June 11-14, 2013
Boston, MA

redhat.

Performance Analysis and Tuning – Part 2

John Shakshober

Director of Performance Engineering, Red Hat

Jeremy Eder
Principal Performance Engineer, Red Hat

June 13, 2013

Agenda: Performance Analysis Tuning Part II

- Part I
  - RHEL “tuned” profiles, top benchmark results
  - Scalabilty – CFS Scheduler tunables / Cgroups
  - Hugepages – Transparent Hugepages, 2MB/1GB
  - NonUniform Memory Access (NUMA) and NUMAD
- Part II
  - Network Performance and Latency-performance
  - Disk and Filesystem IO - Throughput-performance
  - System Performance/Tools – perf, tuna, systemtap
- Q & A

RHEL 6 Networking performance – System setup

- Disable unnecessary services, runlevel 3
- Follow vendor guidelines for BIOS Tuning
  - Logical cores ? Power Management ? Turbo ?
- In the OS, consider
  - Disabling filesystem journal
  - Ensure mount using relatime
  - SSD/Memory Storage
  - Running swapless
  - Reducing writeback thresholds if your app does disk I/O

RHEL6 “tuned” package

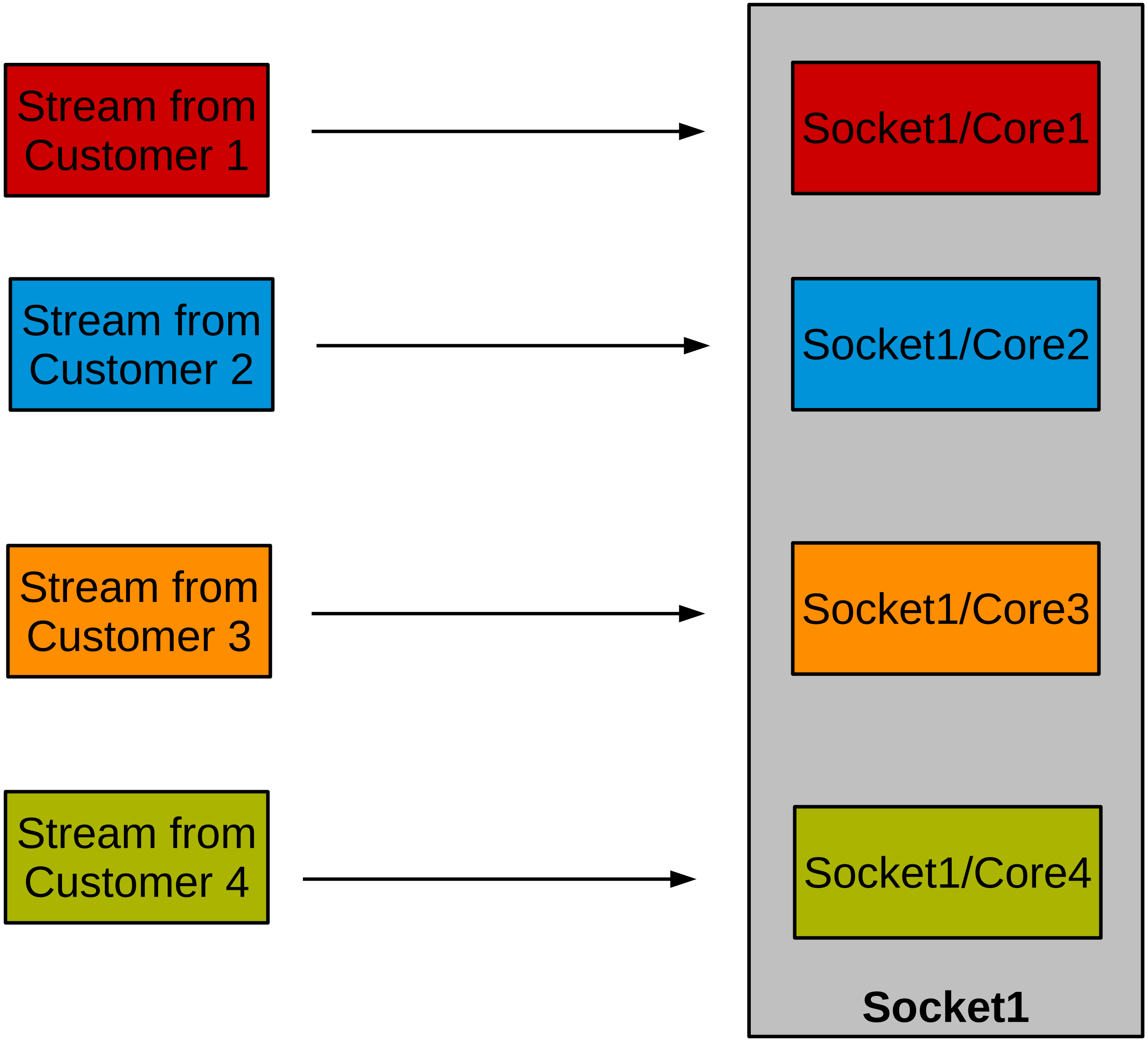
```
# yum install tune*
# tuned-adm profile latency-performance
# tuned-adm list
Available profiles:
- latency-performance
- default
- enterprise-storage
- virtual-guest
- throughput-performance
- virtual-host

Current active profile: latency-performance
# tuned-adm profile default   (to disable)
```

tuned Profile Comparison Matrix

Tunable	default	enterprise-storage	virtual-host	virtual-guest	**latency-performance**	throughput-performance
kernel.sched_min_granularity_ns	4ms	10ms	10ms	10ms		10ms
kernel.sched_wakeup_granularity_ns	4ms	15ms	15ms	15ms		15ms
vm.dirty_ratio	20% RAM	40%	10%	40%		40%
vm.dirty_background_ratio	10% RAM		5%			
vm.swappiness	60		10	30		
I/O Scheduler (Elevator)	CFQ	deadline	deadline	deadline	**deadline**	deadline
Filesystem Barriers	On	Off	Off	Off		
CPU Governor	ondemand	performance			**performance**	performance
Disk Read-ahead		4x				
Disable THP					**Yes**	
CPU C-States					**Locked @ 1**	

https://access.redhat.com/site/solutions/369093

Locality of Packets



Stream from Customer 1

Stream from Customer 2

Stream from Customer 3

Stream from Customer 4

Socket1/Core1

Socket1/Core2

Socket1/Core3

Socket1/Core4

Socket1

Network Tuning: IRQ affinity

- Use irqbalance for the common case
- New irqbalance automates NUMA affinity for IRQs
- Flow-Steering Technologies

- Move 'p1p1*' IRQs to Socket 1:

```
# tuna -q p1p1* -S1 -m -x
# tuna -Q | grep p1p1
```

- Manual IRQ pinning for the last X percent/determinism
  - Guide on Red Hat Customer Portal

NUMA Affinity CLI Reference

```
# numactl -N1 -m1 ./command
```

- Sets CPU affinity for 'command' to CPU node 1
- Allocates memory out of Memory node 1
- Chose node 1 because of PCI-bus wiring
  - Upstream kernel community working on automatic NUMA balancing.
  - Test numad in RHEL6

Network Tuning: NIC Offloads favor Throughput

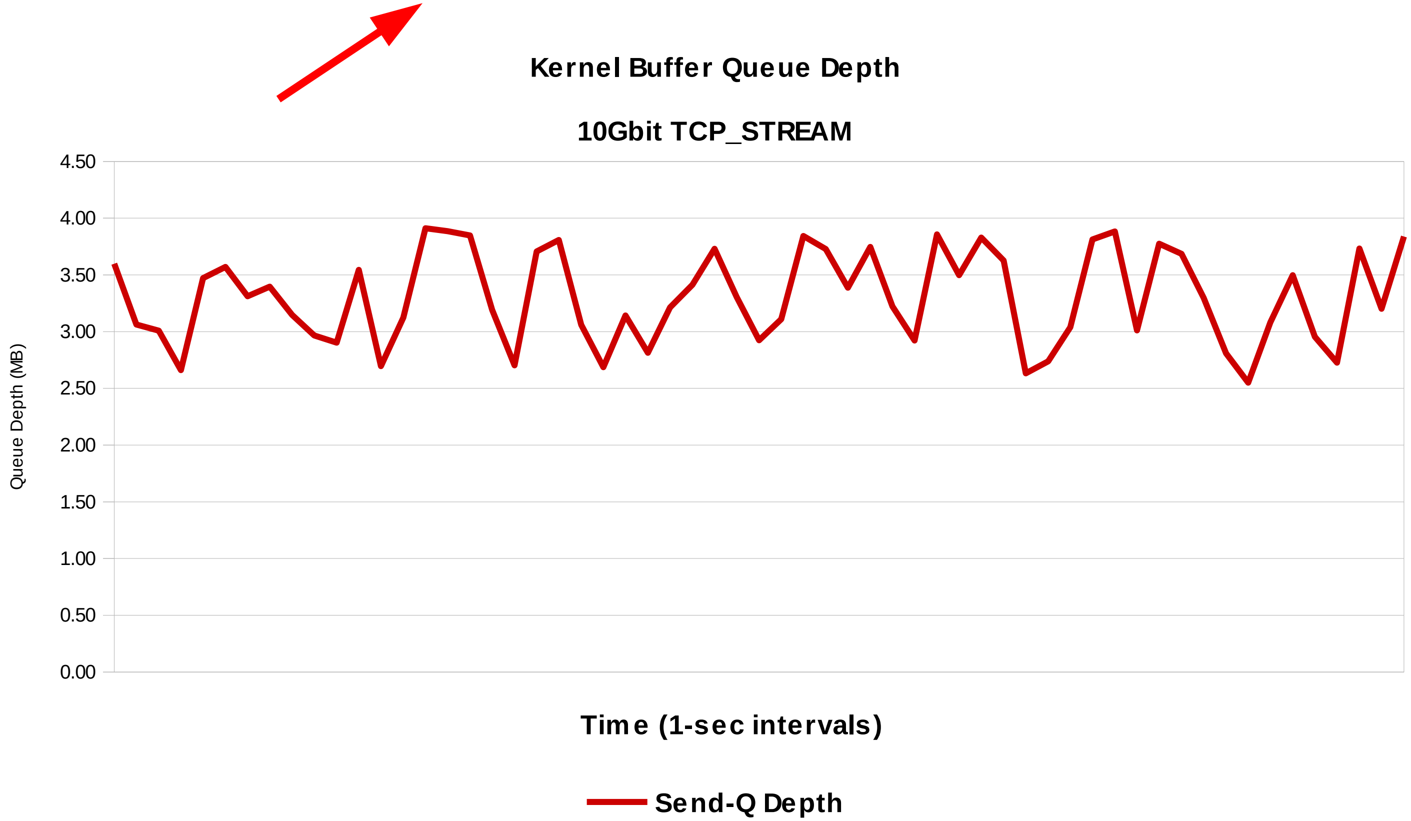
- Reduce the # of packets/IRQs the kernel processes
- Throughput vs Latency trade-off

Offload	Summary	Protocol	Direction
TSO tcp segment offload	MTU-chunking offloaded to NIC	TCP	TX
UFO udp fragment offload	MTU-chunking offloaded to NIC	UDP	TX
GSO generic segment offload	MTU-chunking done in-kernel	TCP, UDP	TX
GRO generic receive offload	NIC/driver batches certain RX packets	TCP, UDP	RX
LRO large receive offload	NIC/driver batches all RX packets	TCP	RX

Network Tuning: Buffer Bloat

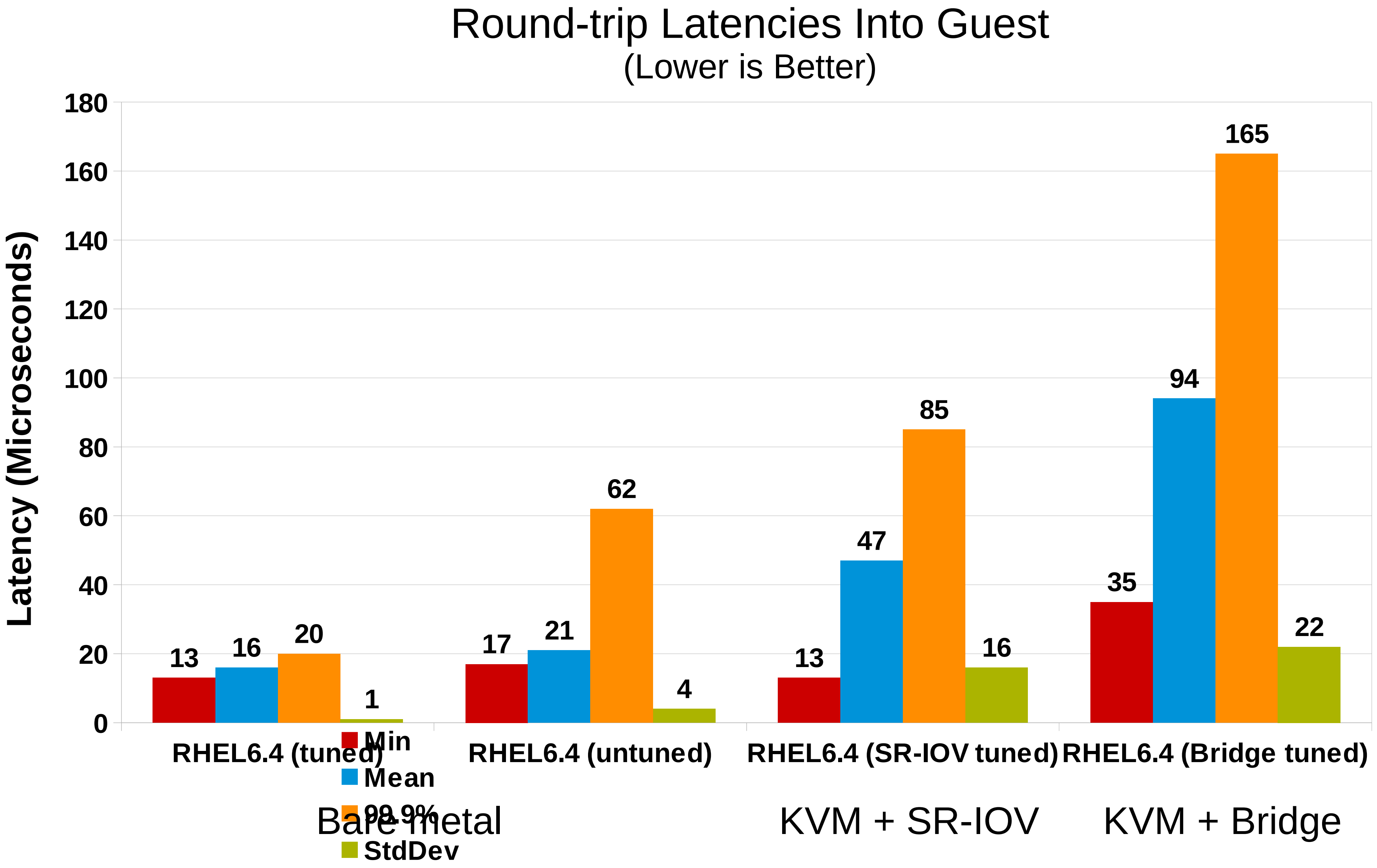
- Kernel buffers:

```
# ss |grep -v ssh
State    Recv-Q Send-Q   Local Address:Port         Peer Address:Port
ESTAB    0      0        172.17.1.36:38462          172.17.1.34:12865
ESTAB    0      3723128  172.17.1.36:58856          172.17.1.34:53491
```



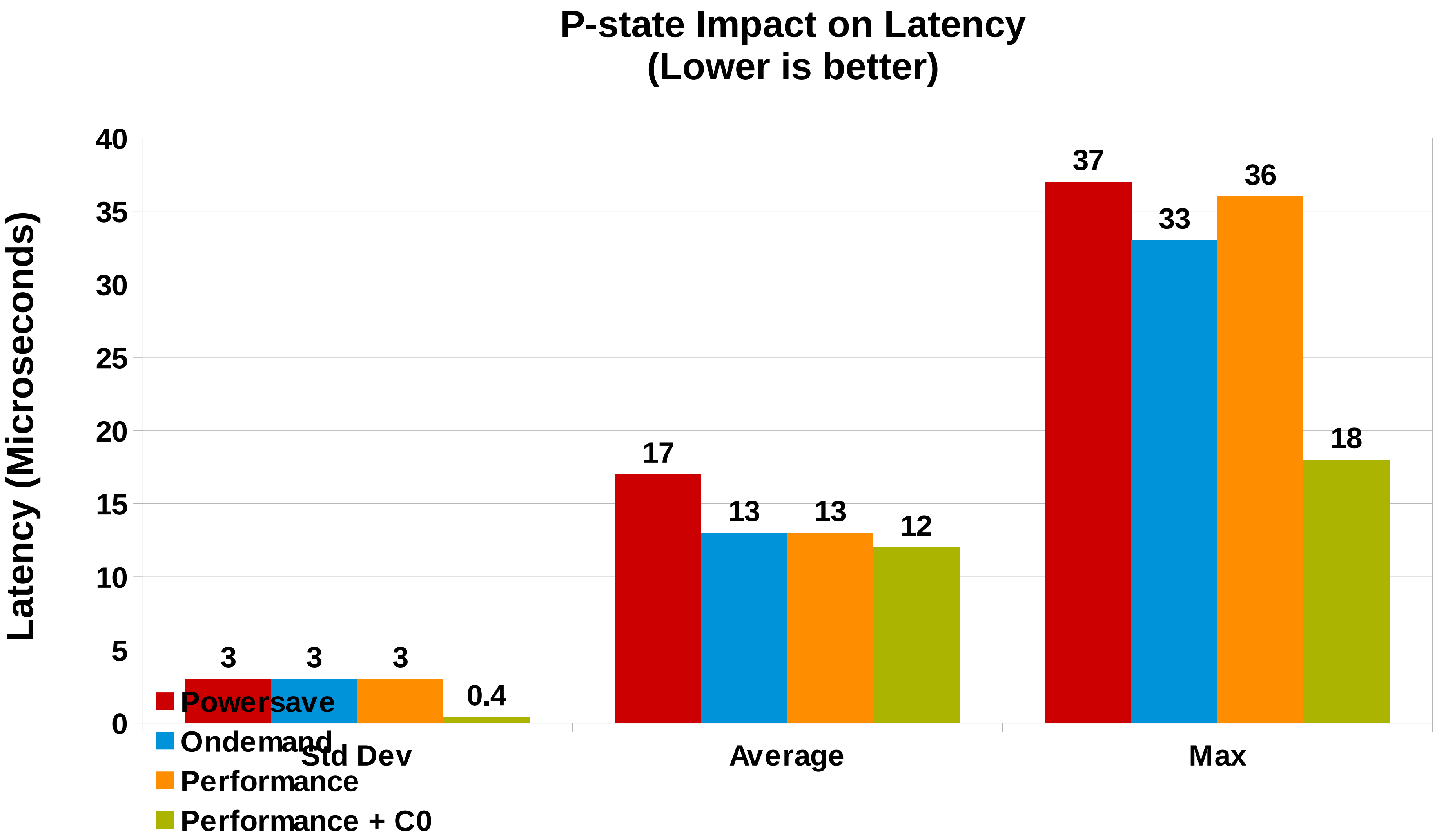
- 10G line-rate
- ~4MB queue depth
- Matching servers

SR-IOV: RHEL 6.4



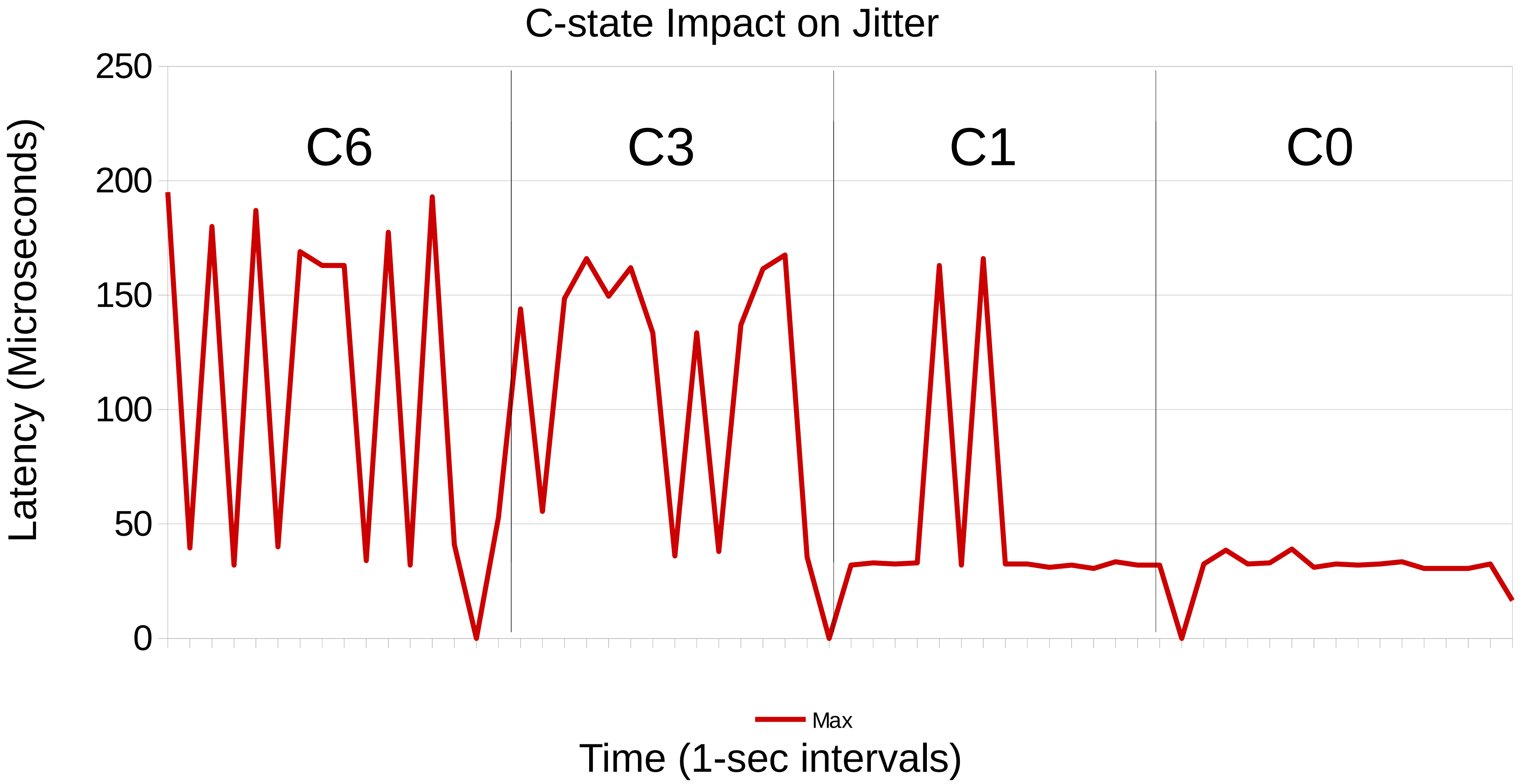
CPU Tuning: P-states (frequency)

- Variable frequencies for each core



CPU Tuning: C-states (idle states)

- Variable idle states for each core
- Cstate lock disables turbo, but increases determinism



Turbostat shows P/C-states on Intel CPUs

turbostat begins shipping in RHEL6.4, cpupowerutils package

```
Default
pk cor CPU    %c0  GHz  TSC    %c1    %c3    %c6    %c7
 0   0   0   0.24 2.93 2.88   5.72   1.32   0.00  92.72
 0   1   1   2.54 3.03 2.88   3.13   0.15   0.00  94.18
 0   2   2   2.29 3.08 2.88   1.47   0.00   0.00  96.25
 0   3   3   1.75 1.75 2.88   1.21   0.47   0.12  96.44
latency-performance
pk cor CPU    %c0  GHz  TSC    %c1    %c3    %c6    %c7
 0   0   0 0.00   3.30 2.90 100.00   0.00   0.00   0.00
 0   1   1 0.00   3.30 2.90 100.00   0.00   0.00   0.00
 0   2   2 0.00   3.30 2.90 100.00   0.00   0.00   0.00
 0   3   3 0.00   3.30 2.90 100.00   0.00   0.00   0.00
```

Power Consumption RHEL6.4 vs RHEL6.4@C0

- C-state lock increases power draw over “out of the box”

Test	Efficiency [Wh] % Diff
Kernel Build	+12.5%
Disk Read	+32.2%
Disk Write	+25.6%
Unpack tar.gz	+23.3%
Active Idle	+41%

- Use cron to set latency-performance tuned profile when necessary.
- Set tuned profile in application init script

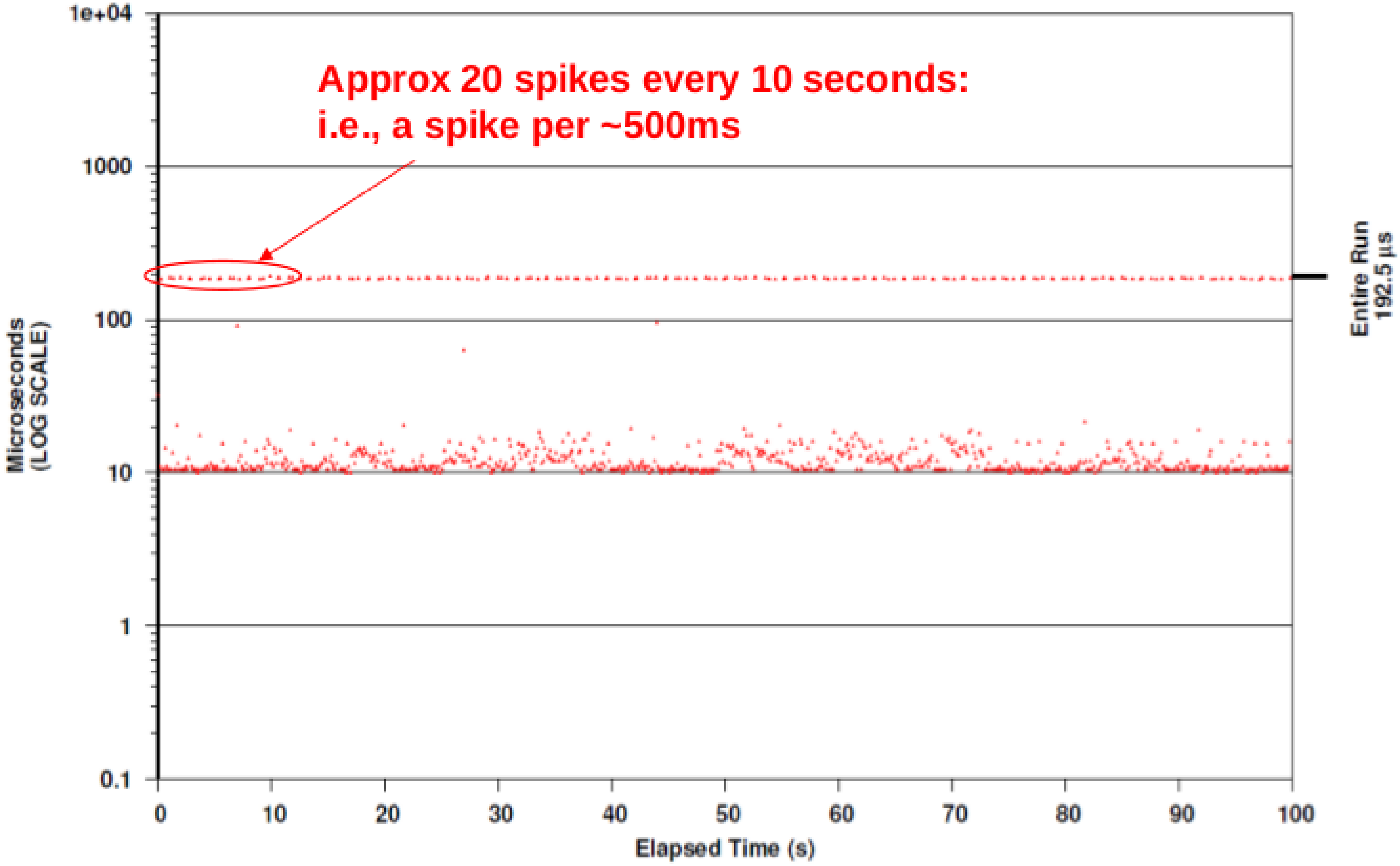
Memory Tuning: Transparent Hugepages

Introduced in RHEL 6

- Added counters in RHEL 6.2
- Enhanced again to reduce overhead in 6.4

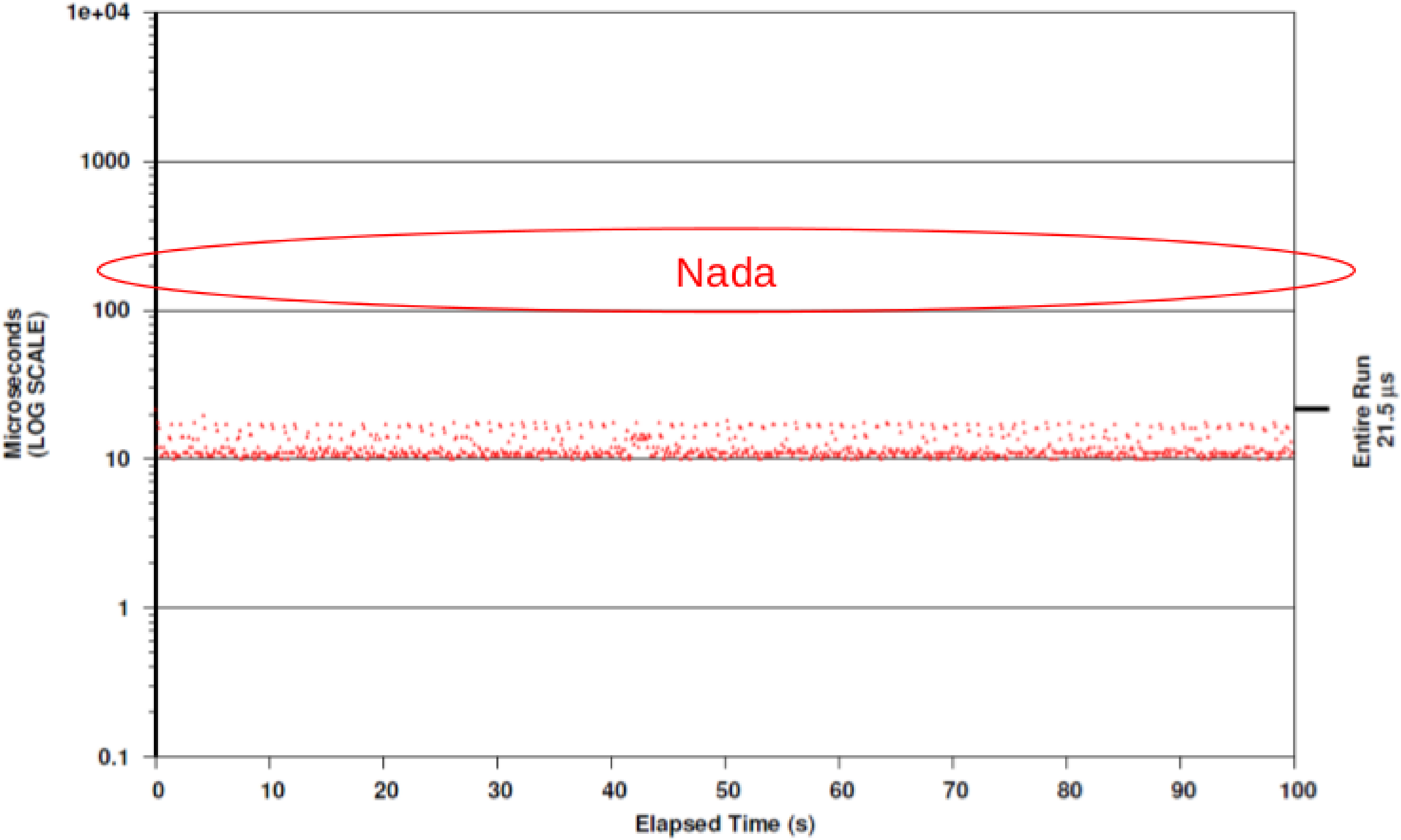
```
 # egrep 'trans|thp' /proc/vmstat
nr_anon_transparent_hugepages 2018
thp_fault_alloc 7302
thp_fault_fallback 0
thp_collapse_alloc 401
thp_collapse_alloc_failed 0
thp_split 21
```

Transparent Hugepages

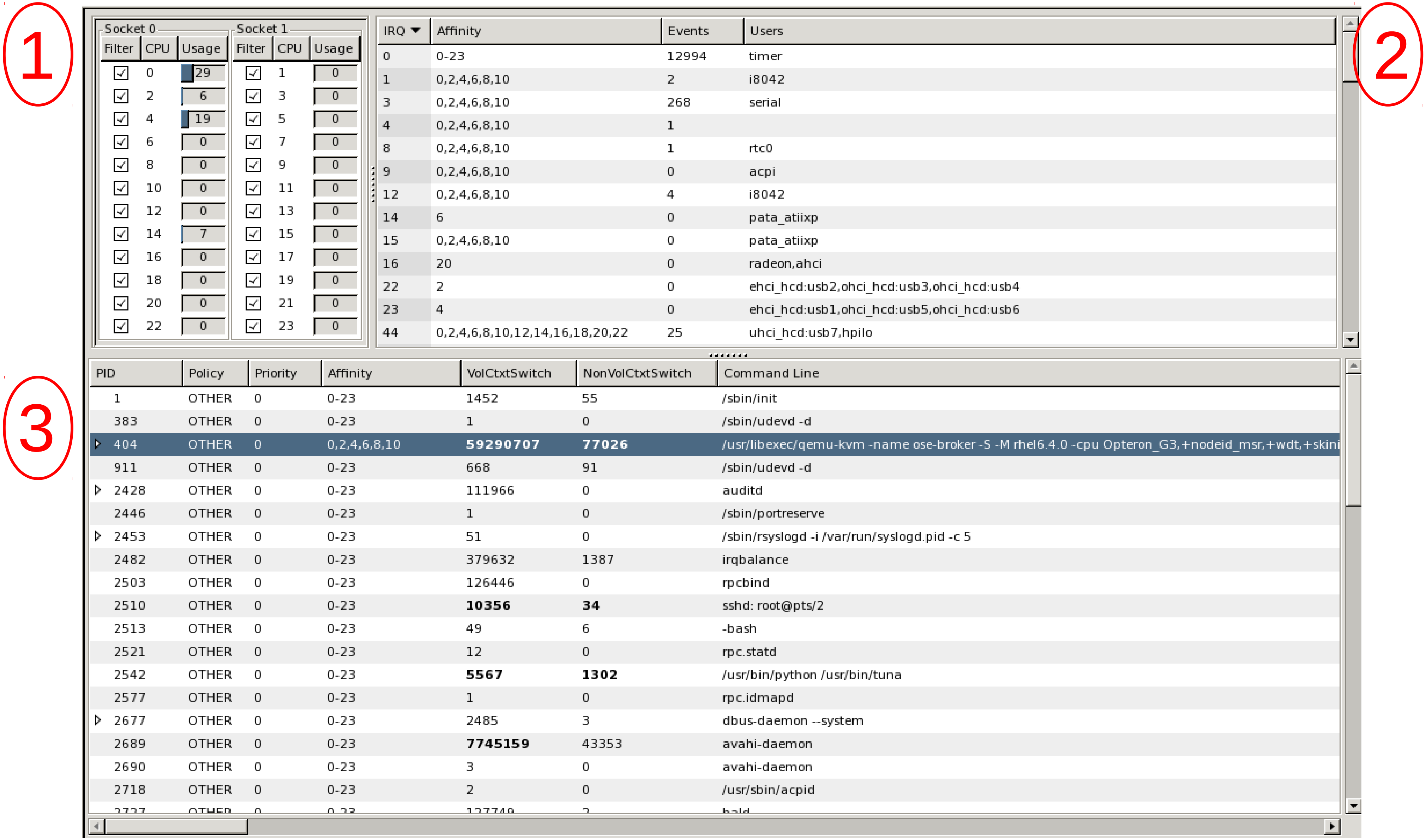


Approx 20 spikes every 10 seconds:
i.e., a spike per ~500ms

Transparent Hugepages Disabled



Tuna (new in RHEL6.4)



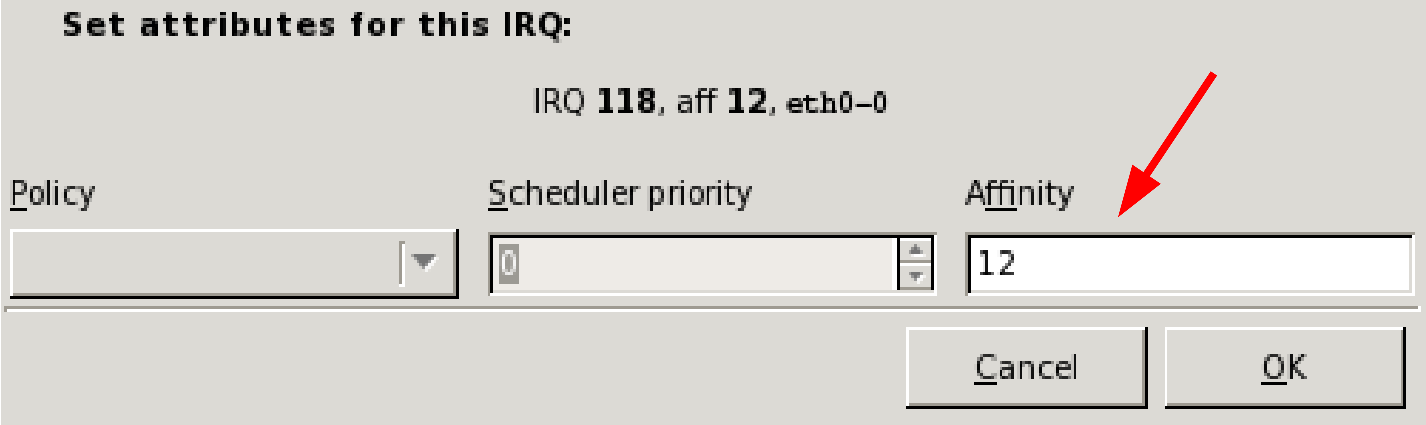
Socket 0			Socket 1		
Filter	CPU	Usage	Filter	CPU	Usage
☑	0	29	☑	1	0
☑	2	6	☑	3	0
☑	4	19	☑	5	0
☑	6	0	☑	7	0
☑	8	0	☑	9	0
☑	10	0	☑	11	0
☑	12	0	☑	13	0
☑	14	7	☑	15	0
☑	16	0	☑	17	0
☑	18	0	☑	19	0
☑	20	0	☑	21	0
☑	22	0	☑	23	0

IRQ ▾	Affinity	Events	Users
0	0-23	12994	timer
1	0,2,4,6,8,10	2	i8042
3	0,2,4,6,8,10	268	serial
4	0,2,4,6,8,10	1	
8	0,2,4,6,8,10	1	rtc0
9	0,2,4,6,8,10	0	acpi
12	0,2,4,6,8,10	4	i8042
14	6	0	pata_atiixp
15	0,2,4,6,8,10	0	pata_atiixp
16	20	0	radeon,ahci
22	2	0	ehci_hcd:usb2,ohci_hcd:usb3,ohci_hcd:usb4
23	4	0	ehci_hcd:usb1,ohci_hcd:usb5,ohci_hcd:usb6
44	0,2,4,6,8,10,12,14,16,18,20,22	25	uhci_hcd:usb7,hpilo

PID	Policy	Priority	Affinity	VolCtxtSwitch	NonVolCtxtSwitch	Command Line
1	OTHER	0	0-23	1452	55	/sbin/init
383	OTHER	0	0-23	1	0	/sbin/udevd -d
▷ 404	OTHER	0	0,2,4,6,8,10	**59290707**	**77026**	/usr/libexec/qemu-kvm -name ose-broker -S -M rhel6.4.0 -cpu Opteron_G3,+nodeid_msr,+wdt,+skini
911	OTHER	0	0-23	668	91	/sbin/udevd -d
▷ 2428	OTHER	0	0-23	111966	0	auditd
2446	OTHER	0	0-23	1	0	/sbin/portreserve
▷ 2453	OTHER	0	0-23	51	0	/sbin/rsyslogd -i /var/run/syslogd.pid -c 5
2482	OTHER	0	0-23	379632	1387	irqbalance
2503	OTHER	0	0-23	126446	0	rpcbind
2510	OTHER	0	0-23	**10356**	**34**	sshd: root@pts/2
2513	OTHER	0	0-23	49	6	-bash
2521	OTHER	0	0-23	12	0	rpc.statd
2542	OTHER	0	0-23	**5567**	**1302**	/usr/bin/python /usr/bin/tuna
2577	OTHER	0	0-23	1	0	rpc.idmapd
▷ 2677	OTHER	0	0-23	2485	3	dbus-daemon --system
2689	OTHER	0	0-23	**7745159**	43353	avahi-daemon
2690	OTHER	0	0-23	3	0	avahi-daemon
2718	OTHER	0	0-23	2	0	/usr/sbin/acpid
2727	OTHER	0	0-23	127749	2	hald

Tuna IRQ/CPU affinity context menus

- CPU affinity for IRQs

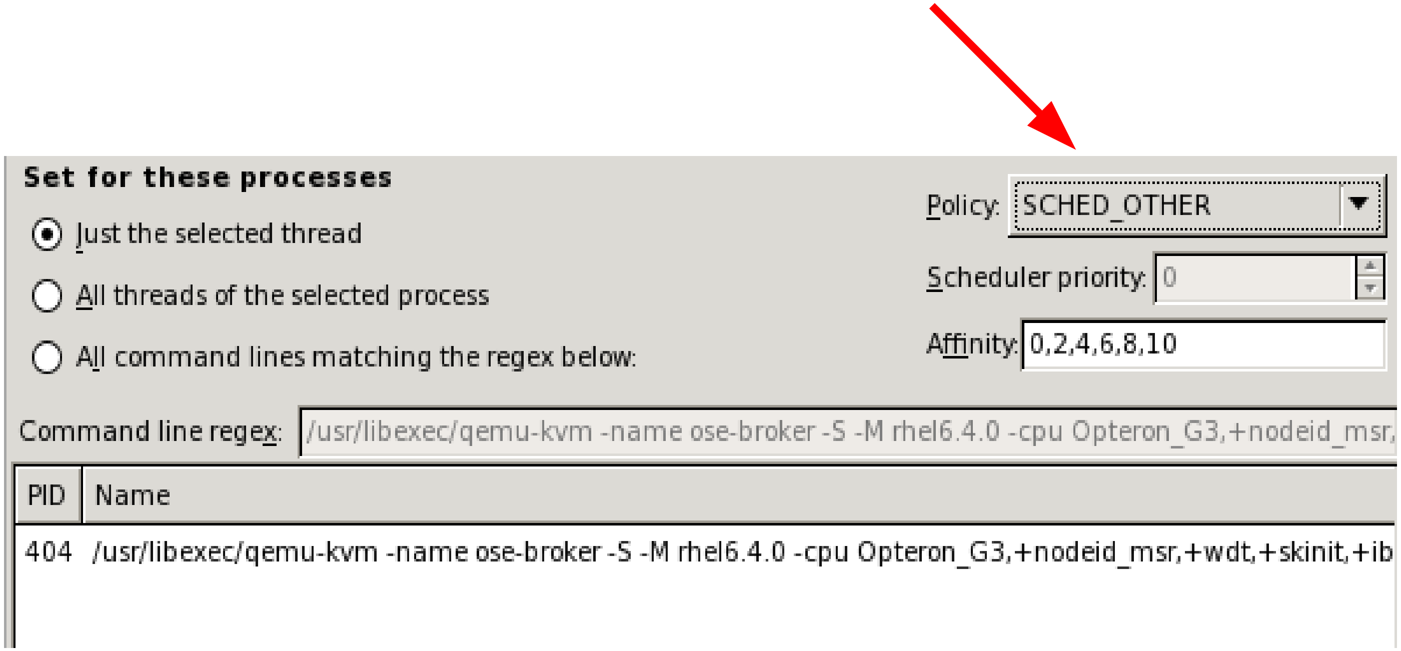


Set attributes for this IRQ:

IRQ **118**, aff **12**, eth0-0

Policy	Scheduler priority	Affinity
	0	12

- CPU affinity for PIDs
- Scheduler Policy
- Scheduler Priority



Set for these processes

● Just the selected thread

○ All threads of the selected process

○ All command lines matching the regex below:

Policy: SCHED_OTHER

Scheduler priority: 0

Affinity: 0,2,4,6,8,10

Command line regex: /usr/libexec/qemu-kvm -name ose-broker -S -M rhel6.4.0 -cpu Opteron_G3,+nodeid_msr,

PID	Name
404	/usr/libexec/qemu-kvm -name ose-broker -S -M rhel6.4.0 -cpu Opteron_G3,+nodeid_msr,+wdt,+skinit,+ib

Tuna – for processes

```
 # tuna -t netserver -P
                      thread       ctxt_switches
  pid SCHED_ rtpri affinity voluntary nonvoluntary             cmd
13488   OTHER     0    0xfff         1            0       netserver

 # tuna -c2 -t netserver -m
 # tuna -t netserver -P
                      thread       ctxt_switches
  pid SCHED_ rtpri affinity voluntary nonvoluntary             cmd
13488   OTHER     0        2         1            0       netserver
```

Tuna – for IRQs

- Move 'p1p1*' IRQs to Socket 1:

```
# tuna -q p1p1* -S0 -m -x
  # tuna -Q | grep p1p1                     Core
    78 p1p1-0                                   0   sfc
    79 p1p1-1                                   1   sfc
    80 p1p1-2                                   2   sfc
    81 p1p1-3                                   3   sfc
    82 p1p1-4                                   4   sfc
...
```

Tuna – for core/socket isolation

```
# tuna -S1 -i

# grep Cpus_allowed_list /proc/`pgrep rsyslogd`/status
```

```
Cpus_allowed_list:   0-15
```

```
# tuna -S1 -i (tuna sets affinity of 'init' task as well)
```

```
# grep Cpus_allowed_list /proc/`pgrep rsyslogd`/status
```

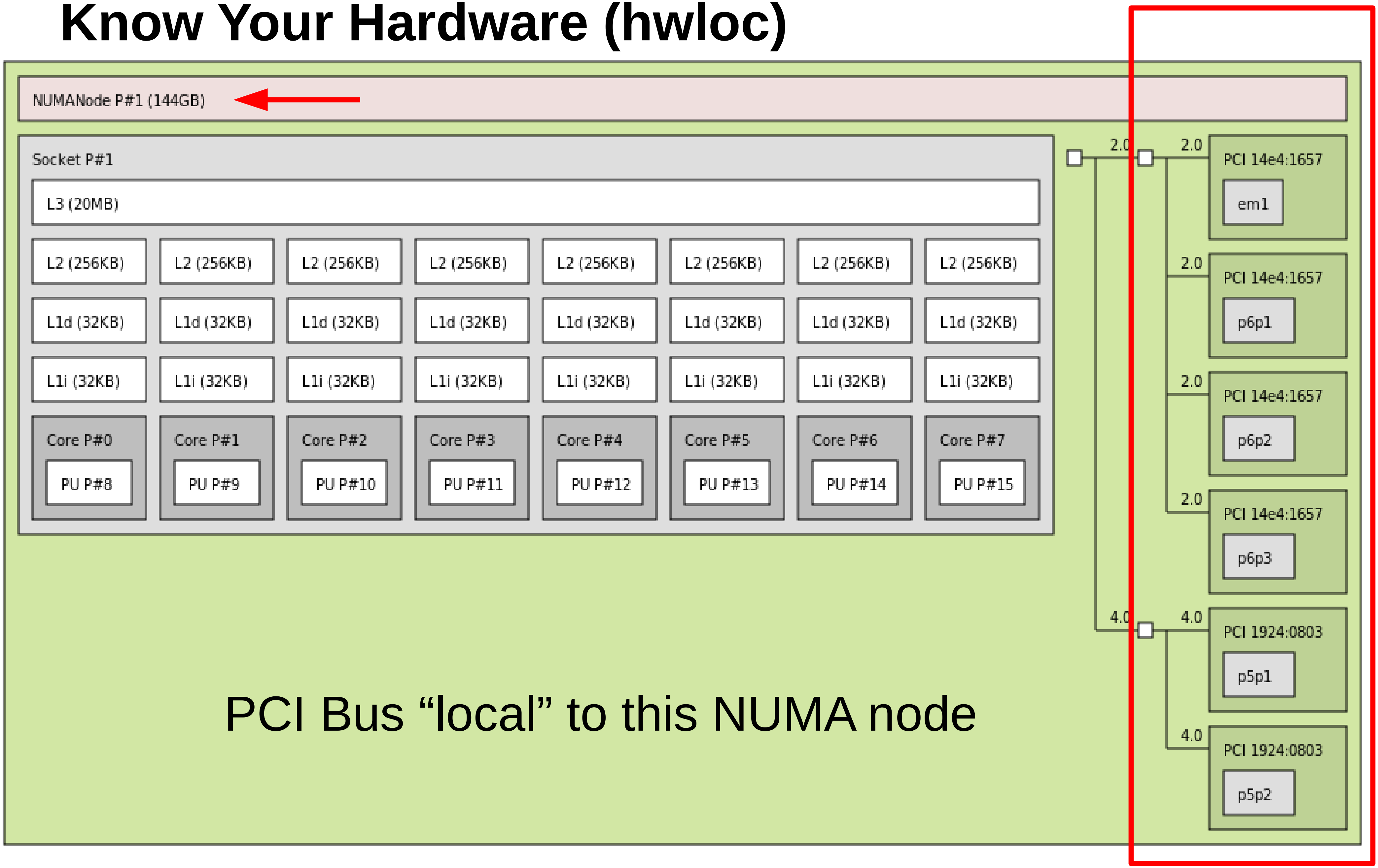
```
Cpus_allowed_list:   0,1,2,3,4,5,6,7
```

NUMA Topology and PCI Bus

- Servers may have more than 1 PCI bus.
- Install adapters “close” to the CPU that will run the performance critical application.
- When BIOS reports locality, irqbalance handles NUMA/IRQ affinity automatically.

```
42:00.0 Network controller: Mellanox Technologies MT27500 Family [ConnectX-3]

# cat /sys/devices/pci0000\:40/0000\:40\:03.0/0000\:42\:00.0/local_cpulist

  1,3,5,7,9,11,13,15


# dmesg | grep "NUMA node"

pci_bus 0000:00: on NUMA node 0 (pxm 1)

pci_bus 0000:40: on NUMA node 1 (pxm 2)

pci_bus 0000:3f: on NUMA node 0 (pxm 1)

pci_bus 0000:7f: on NUMA node 1 (pxm 2)
```

Know Your Hardware (hwloc)



PCI Bus “local” to this NUMA node

PCI Device Affinity

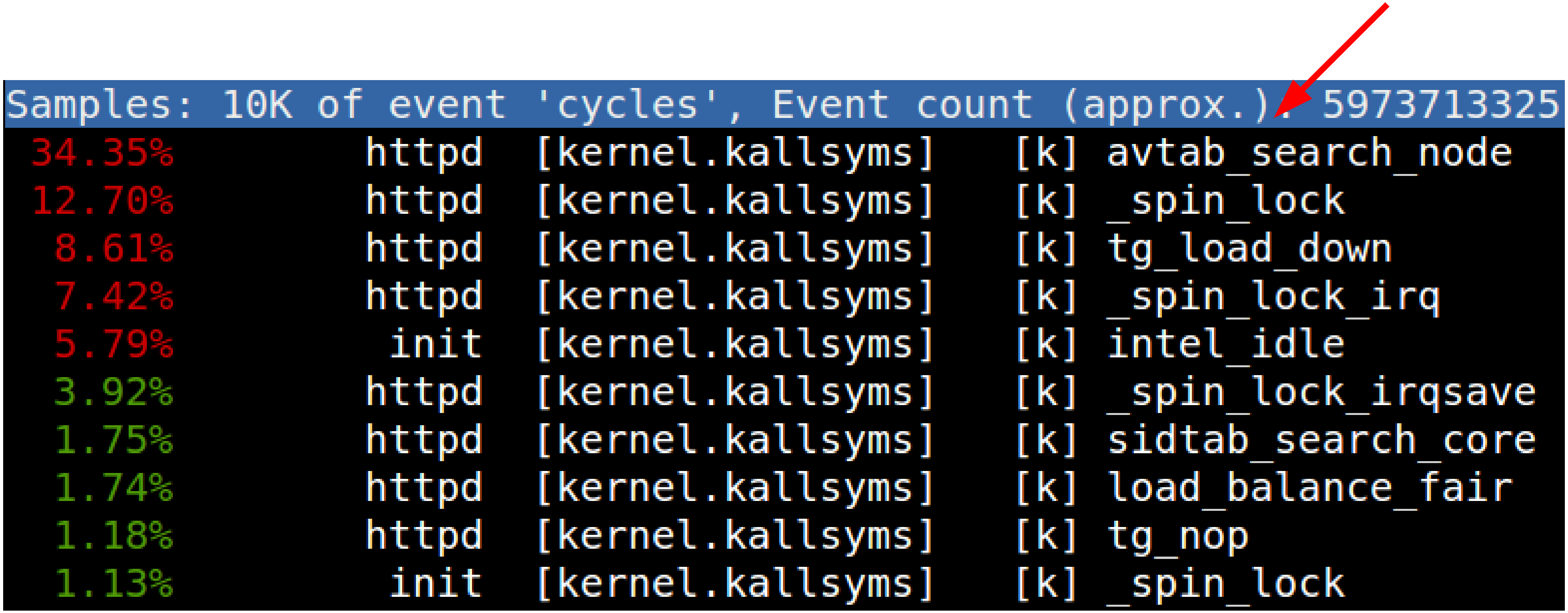
```
# lstopo-no-graphics |egrep 'NUMA|eth4'
  NUMANode L#0 (P#0 144GB)
  NUMANode L#1 (P#1 144GB)
          Net L#10 "eth4"
```

Performance Monitoring Tool - perf

- Userspace tool to read CPU counters and kernel tracepoints
  - RHEL 6.4 includes perf from upstream kernel 3.6
  - https://perf.wiki.kernel.org
  - perf top (dynamic)
  - perf record / report (save and replay)
  - perf stat <command> (analyze a particular workload)

Tracing with 'perf': perf top

- System-wide 'top' view of active processes

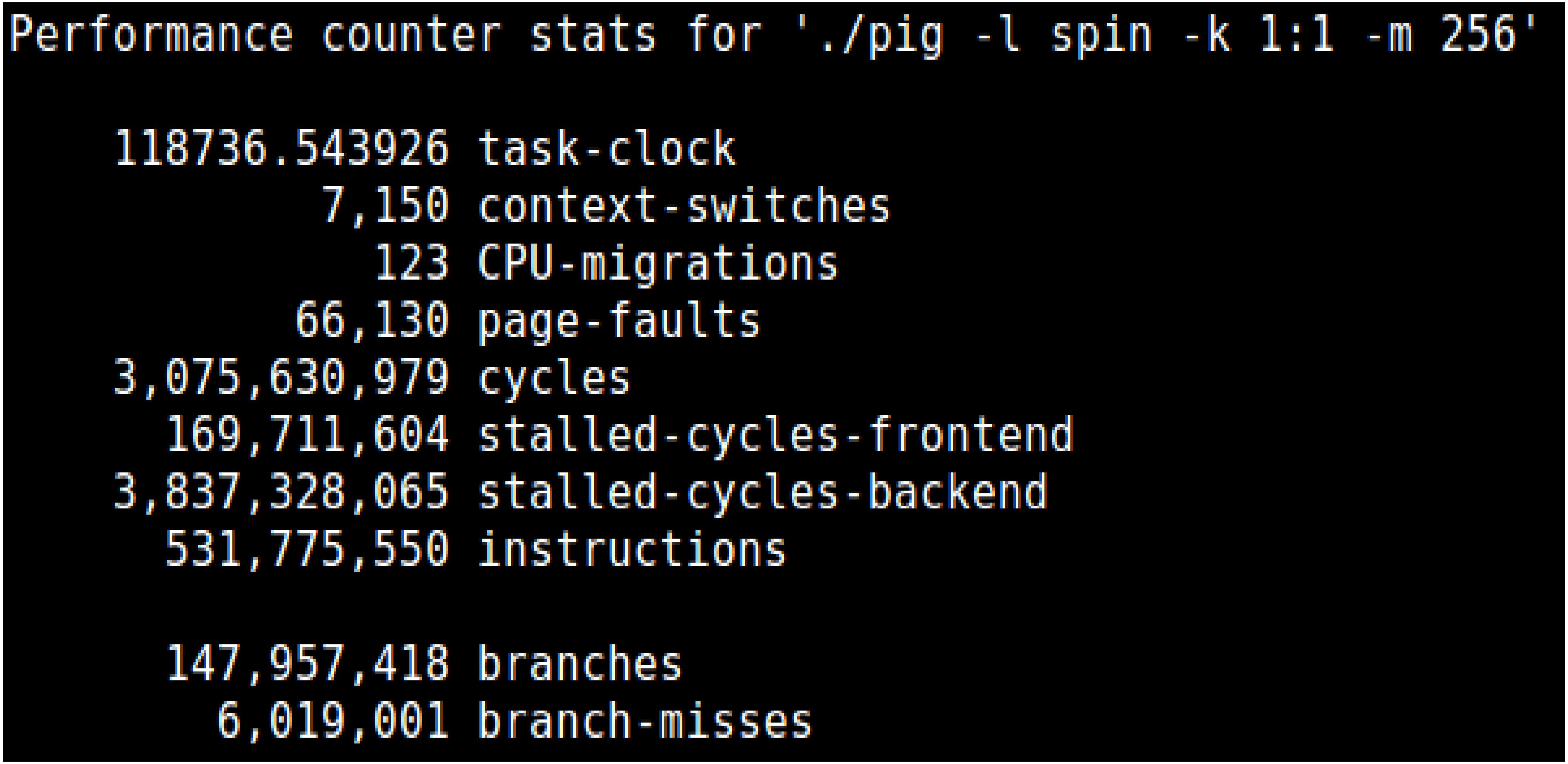


```
Samples: 10K of event 'cycles', Event count (approx.): 5973713325
 34.35%         httpd  [kernel.kallsyms]   [k] avtab_search_node
 12.70%         httpd  [kernel.kallsyms]   [k] _spin_lock
  8.61%         httpd  [kernel.kallsyms]   [k] tg_load_down
  7.42%         httpd  [kernel.kallsyms]   [k] _spin_lock_irq
  5.79%          init  [kernel.kallsyms]   [k] intel_idle
  3.92%         httpd  [kernel.kallsyms]   [k] _spin_lock_irqsave
  1.75%         httpd  [kernel.kallsyms]   [k] sidtab_search_core
  1.74%         httpd  [kernel.kallsyms]   [k] load_balance_fair
  1.18%         httpd  [kernel.kallsyms]   [k] tg_nop
  1.13%          init  [kernel.kallsyms]   [k] _spin_lock
```

Tracing with 'perf': perf stat

- Attach to existing PID and report all kmem tracepoints:

```
# perf stat -a ./my_cmd
```

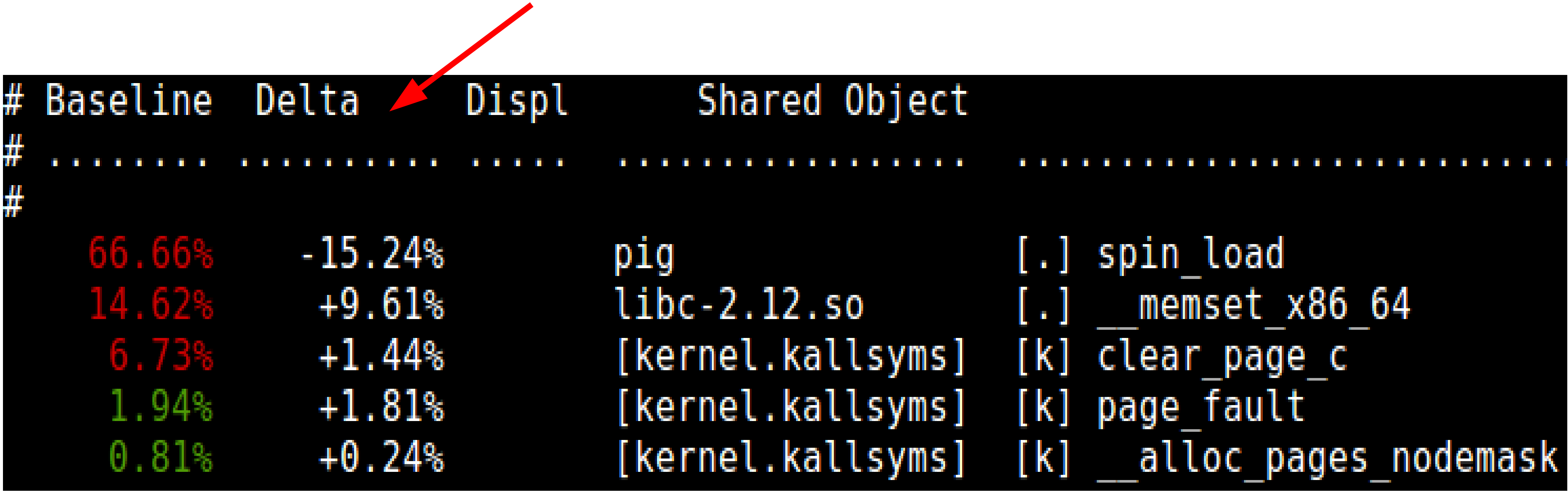


```
Performance counter stats for './pig -l spin -k 1:1 -m 256'

    118736.543926 task-clock
            7,150 context-switches
              123 CPU-migrations
           66,130 page-faults
    3,075,630,979 cycles
      169,711,604 stalled-cycles-frontend
    3,837,328,065 stalled-cycles-backend
      531,775,550 instructions

      147,957,418 branches
        6,019,001 branch-misses
```

Tracing with 'perf': perf diff

- Show differences between 2 perf.data recordings
  - Run perf record twice, each with different tuning

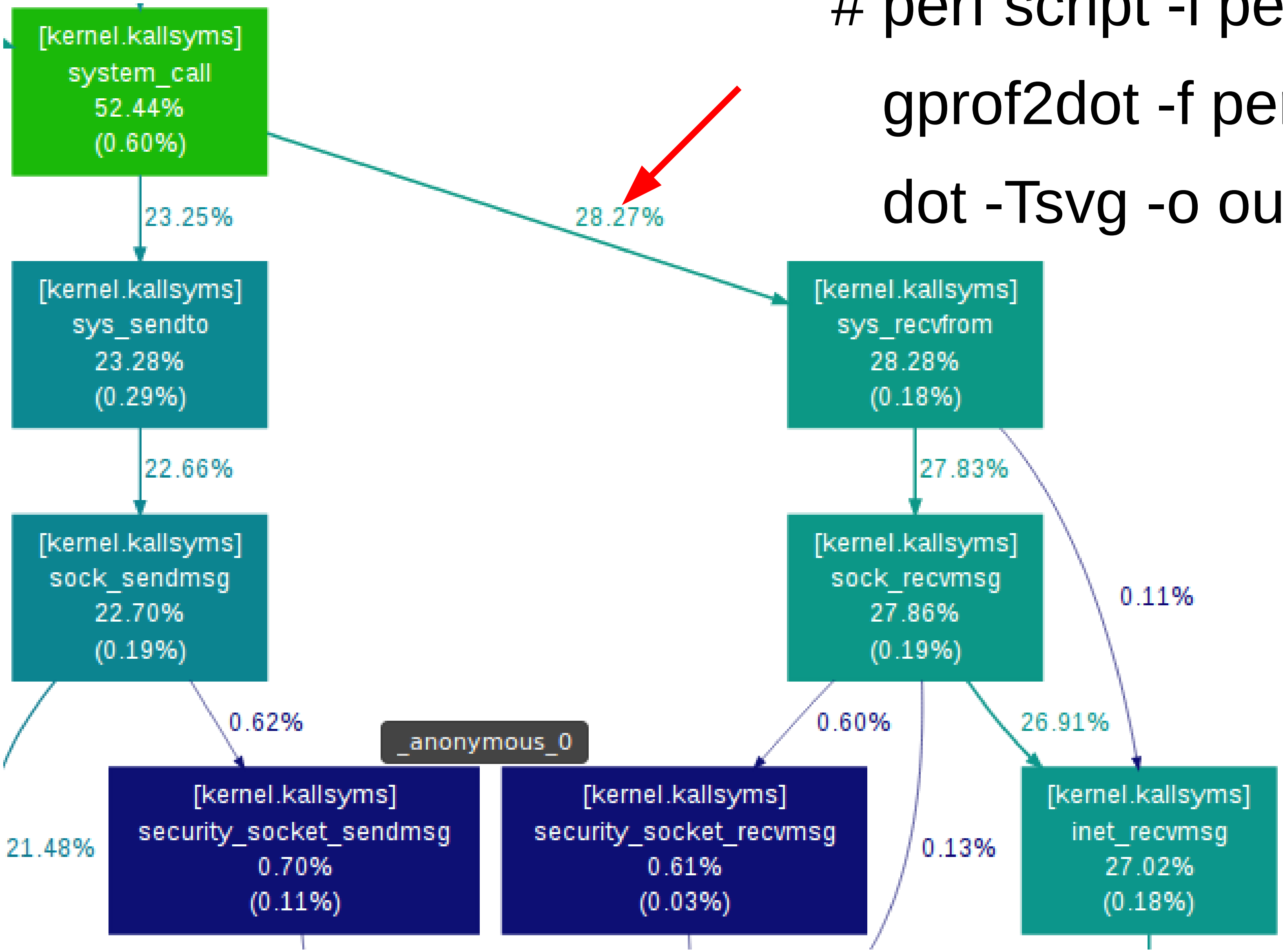
```
# perf diff -M perf.data.old perf.data
```



```
# Baseline  Delta     Displ       Shared Object
# ......... ........... .....  .................  ............................
#
    66.66%    -15.24%          pig                [.] spin_load
    14.62%     +9.61%          libc-2.12.so       [.] __memset_x86_64
     6.73%     +1.44%          [kernel.kallsyms]  [k] clear_page_c
     1.94%     +1.81%          [kernel.kallsyms]  [k] page_fault
     0.81%     +0.24%          [kernel.kallsyms]  [k] __alloc_pages_nodemask
```

Tracing with 'perf': perf script + gprof2dot

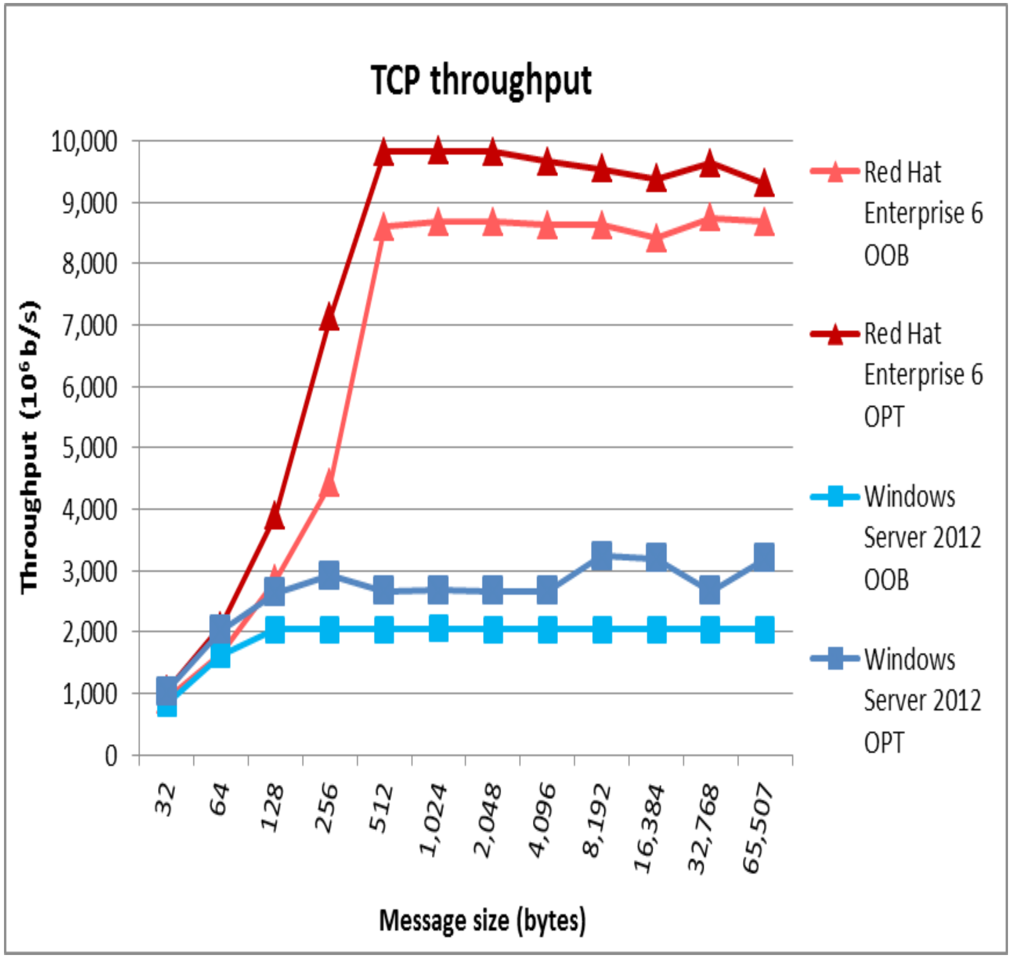
```
# perf script -i perf.data | \
  gprof2dot -f perf | \
  dot -Tsvg -o output.svg
```

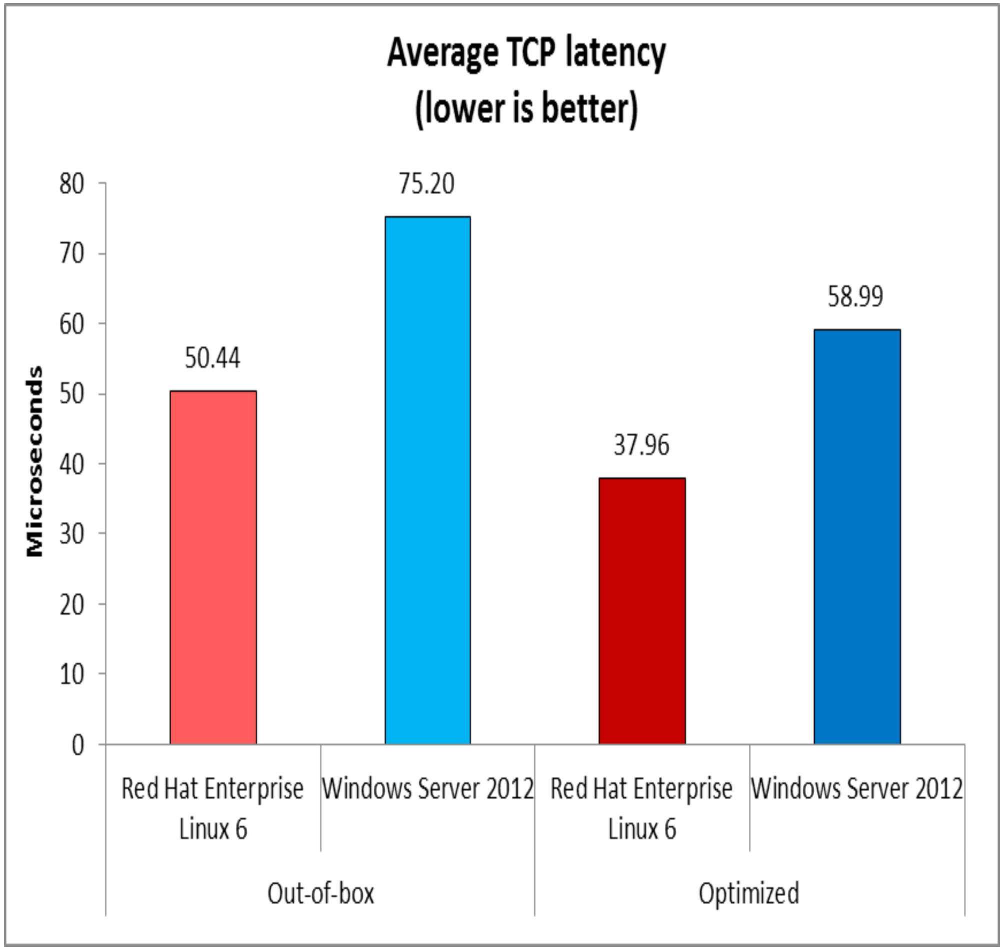


Interesting new Network/Perf things in RHEL6.4

- tuna included
- latency-performance “tuned” profile beefed up
  - Lock C-states
  - Disable Transparent Hugepages
- turbostat included in cpupowerutils package
- hwloc now reports PCI bus topology
- PTP Tech Preview
- Mellanox Infiniband SR-IOV Tech Preview

RHEL vs Windows Server 2012 Comparison Network



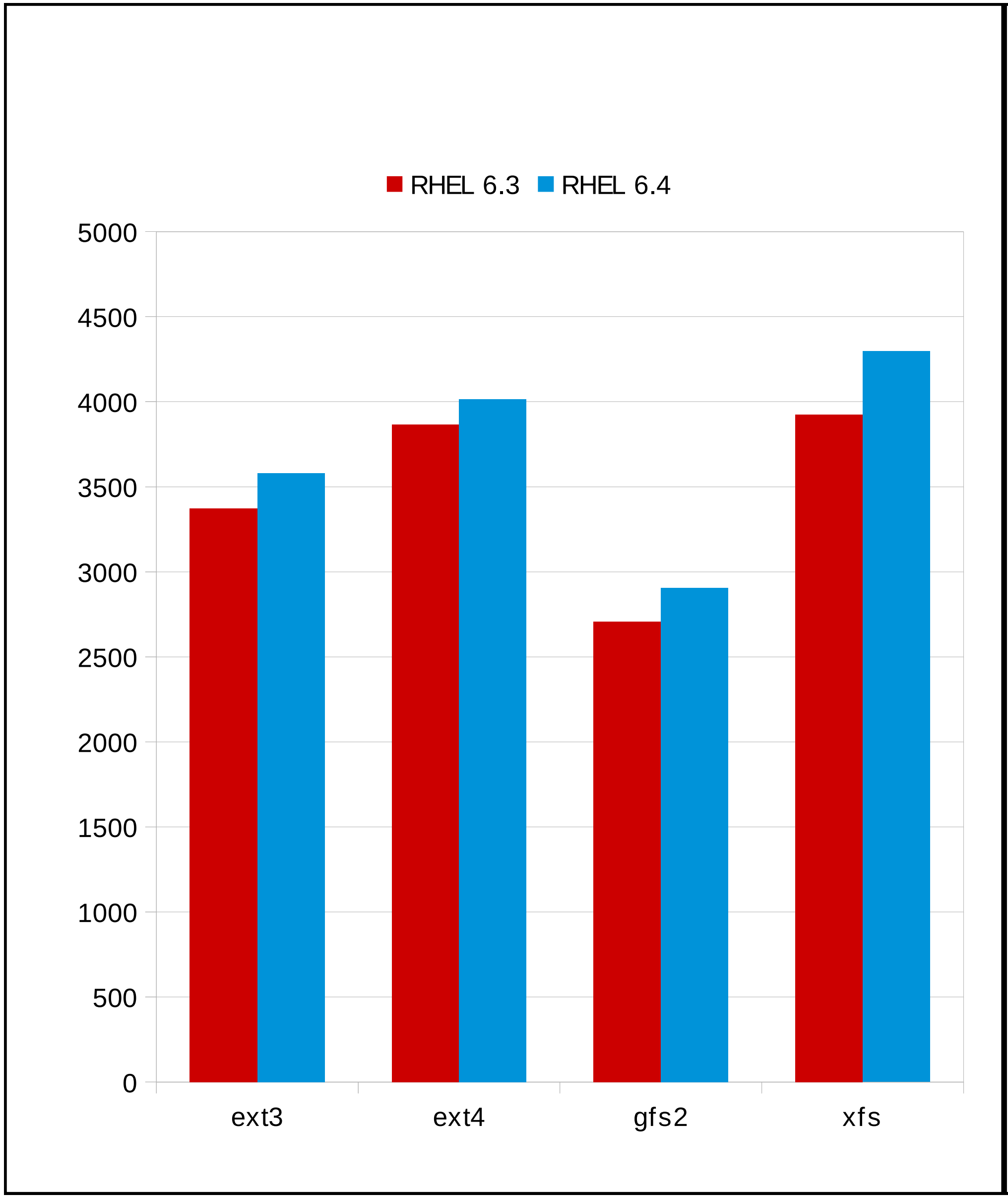


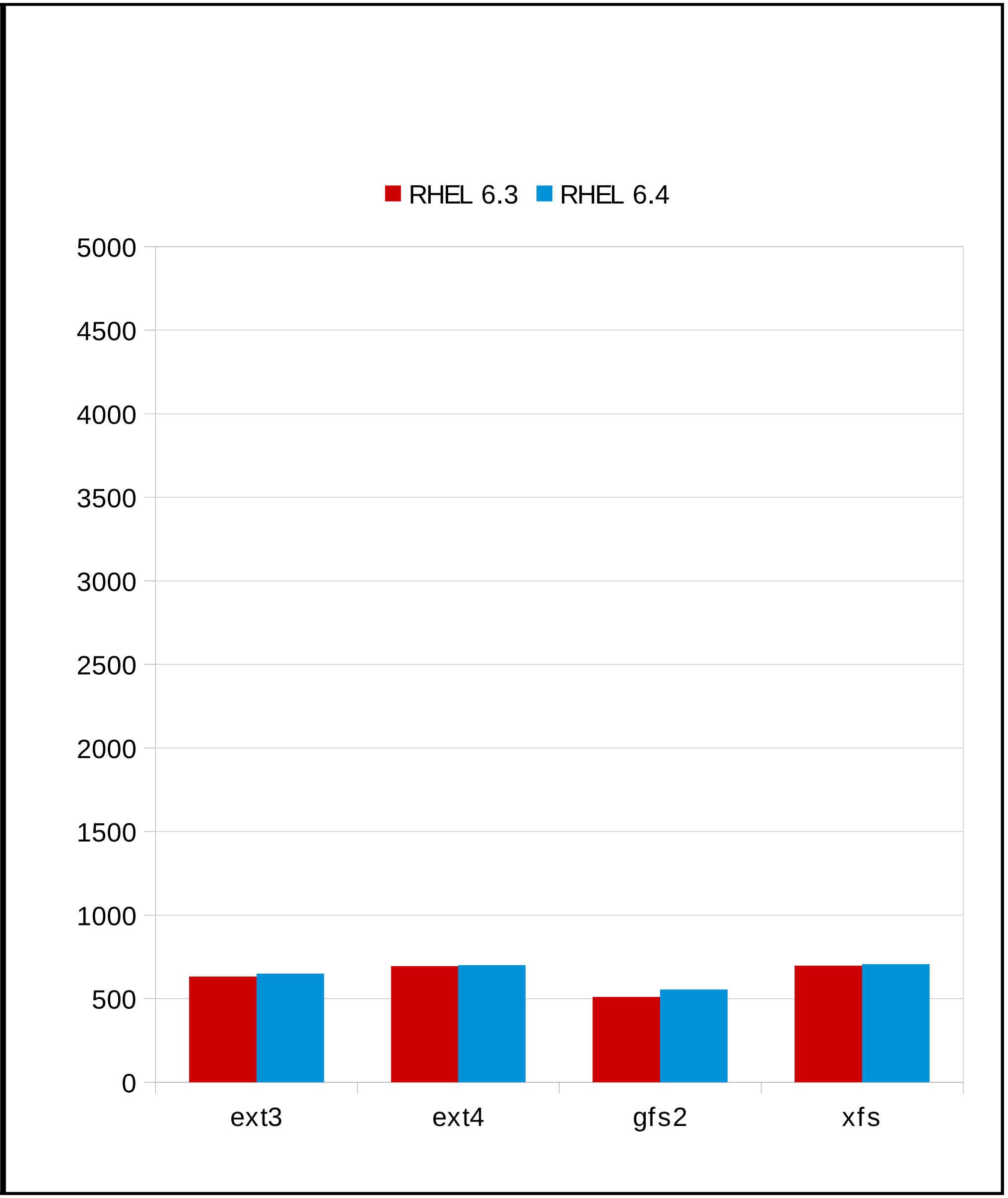
- In both OOB and Optimized cases Red Hat Enterprise Linux delivers better throughput and lower latency to critical network-heavy applications
- http://www.principledtechnologies.com/RedHat/RHEL6_network_0613.pdf

I/O Tuning – Understanding I/O Elevators

- Deadline
  - Two queues per device, one for read and one for writes
  - I/Os dispatched based on time spent in queue
- CFQ
  - Per process queue
  - Each process queue gets fixed time slice (based on process priority)
- Noop
  - FIFO
  - Simple I/O Merging
  - Lowest CPU Cost

Iozone Performance Comparison EXT4/XFS/GFS





SAS Application on Standalone Systems

Picking a RHEL File System

xfs **most** recommended

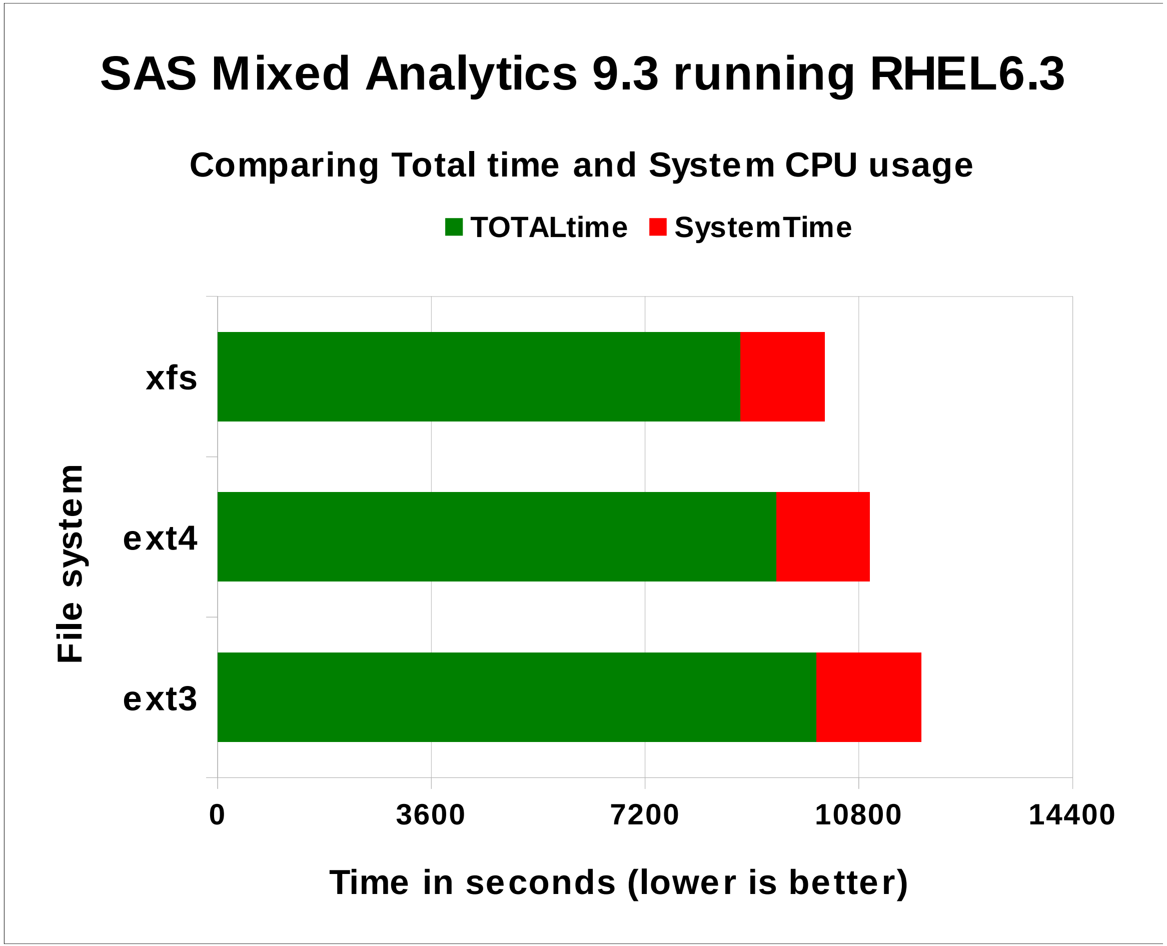
- Max file system size 100TB
- Max file size 100TB
- Best performing

ext4 recommended

- Max file system size 16TB
- Max file size 16TB

ext3 not recommended

- Max file system size 16TB
- Max file size 2TB



RHEL6 “tuned” package

```
# yum install tune*
# tuned-adm profile enterprise-storage
# tuned-adm list
Available profiles:
- latency-performance
- default
- enterprise-storage
- virtual-guest
- throughput-performance
- virtual-host

Current active profile: enterprise-storage
# tuned-adm profile default   (to disable)
```

tuned Profile Comparison Matrix

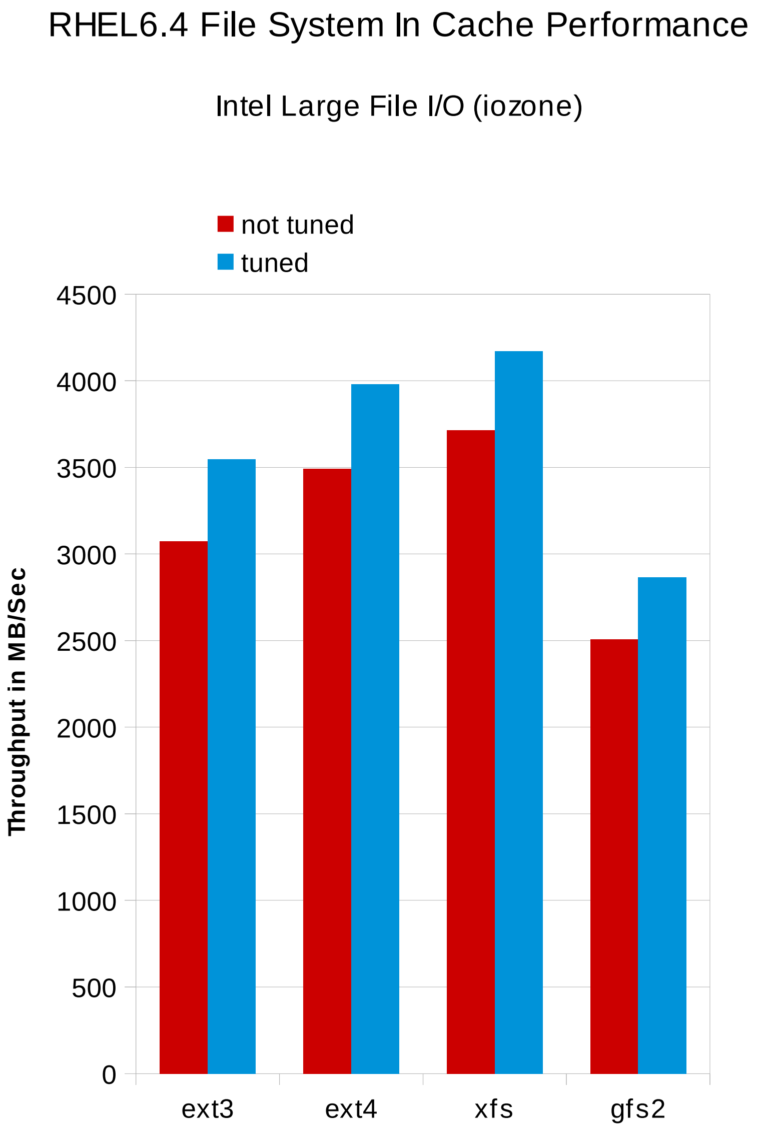
Tunable	default	**enterprise-storage**	virtual-host	virtual-guest	latency-performance	throughput-performance
kernel.sched_min_granularity_ns	4ms	**10ms**	10ms	10ms		10ms
kernel.sched_wakeup_granularity_ns	4ms	**15ms**	15ms	15ms		15ms
vm.dirty_ratio	20% RAM	**40%**	10%	40%		40%
vm.dirty_background_ratio	10% RAM		5%			
vm.swappiness	60		10	30		
I/O Scheduler (Elevator)	CFQ	deadline	deadline	deadline	deadline	deadline
Filesystem Barriers	On	**Off**	Off	Off		
CPU Governor	ondemand	**performance**			performance	performance
Disk Read-ahead		4x				
Disable THP					Yes	
Disable C-States					Yes	

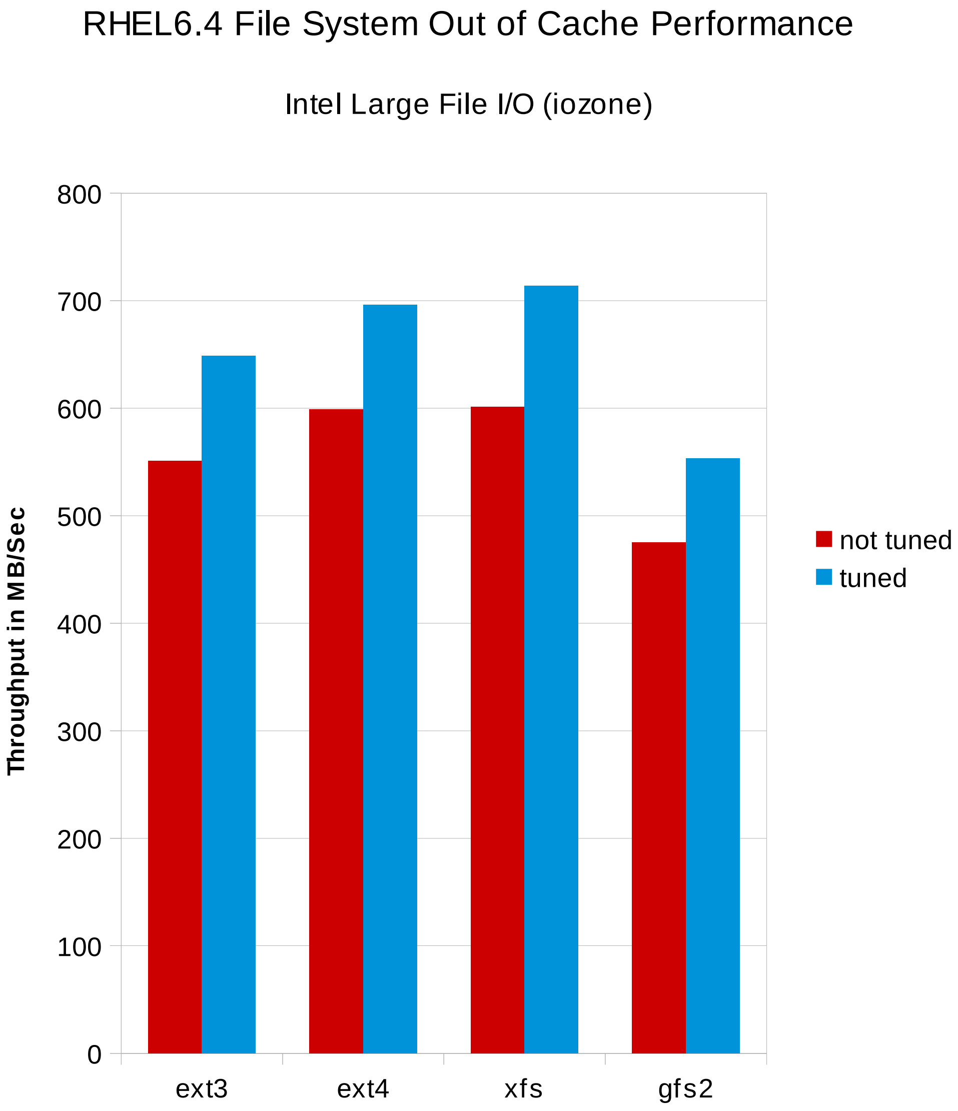
https://access.redhat.com/site/solutions/369093

Tuning Memory – Flushing Caches

- Drop unused Cache – to control pagecache dynamically
  - ✔ Frees most pagecache memory
  - ✔ File cache
  - ✗ If the DB uses cache, may notice slowdown
- NOTE: Use for benchmark environments.
- **Free pagecache**
  - **# sync;** echo 1 > /proc/sys/vm/drop_caches
- **Free slabcache**
  - # sync; echo 2 > /proc/sys/vm/drop_caches
- **Free pagecache and slabcache**
  - # sync; echo 3 > /proc/sys/vm/drop_caches

Iozone Performance Effect of TUNED EXT4/XFS/GFS



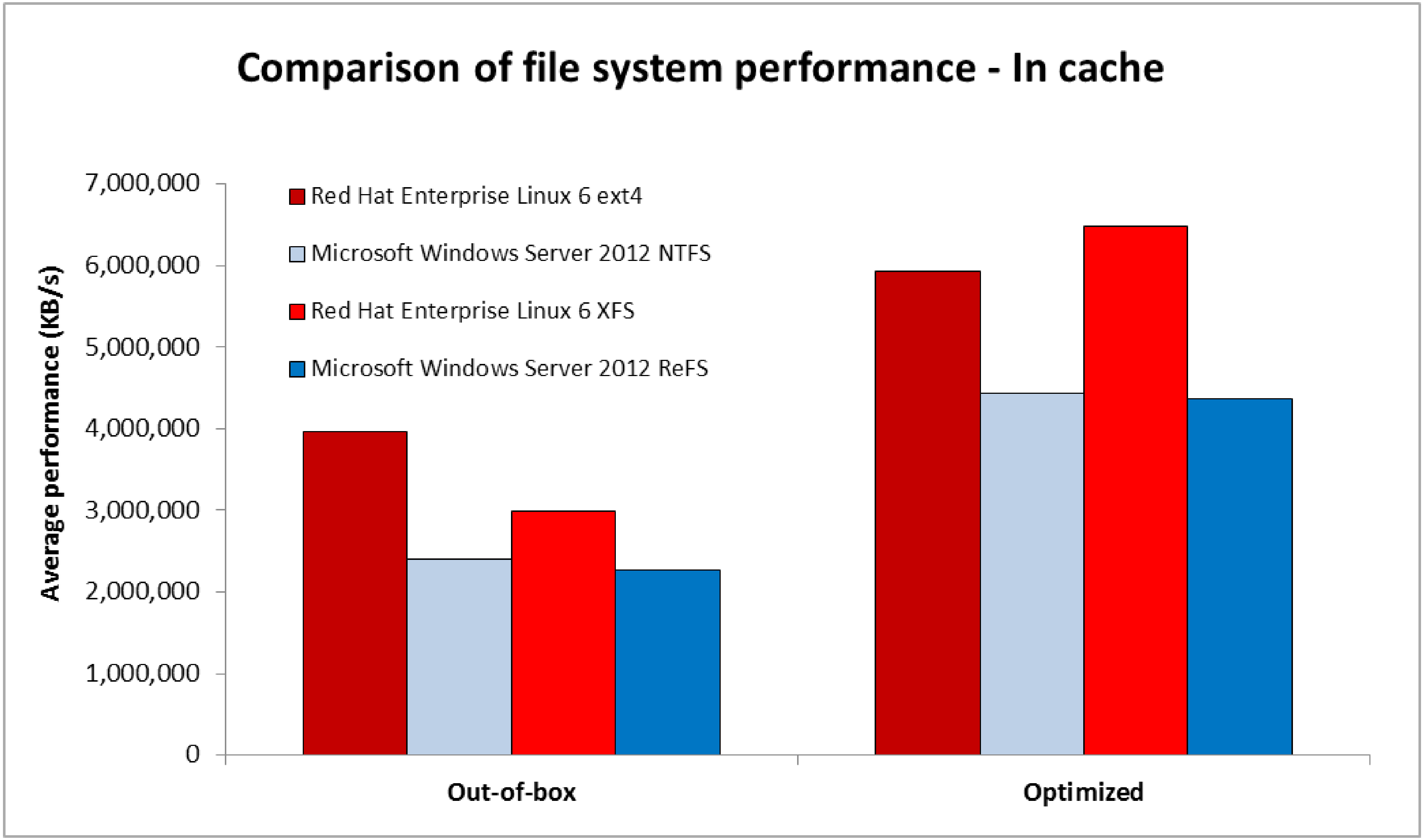


RHEL BIOS and Tuned profiles

BIOS to OS controlled

```
# tuned-adm profile enterprise-storage
kernel.sched_min_granularity_ns = 10000000
kernel.sched_wakeup_granularity_ns = 15000000
vm.dirty_ratio = 40
ELEVATOR="deadline"
 If
 /sys/block/sdX/device/scsi_disk/X:X:X:X/cache_type:write back
 then
 BARRIERS=off (for mounts other than root/boot vols)
# set cpuspeed governors=performance
```

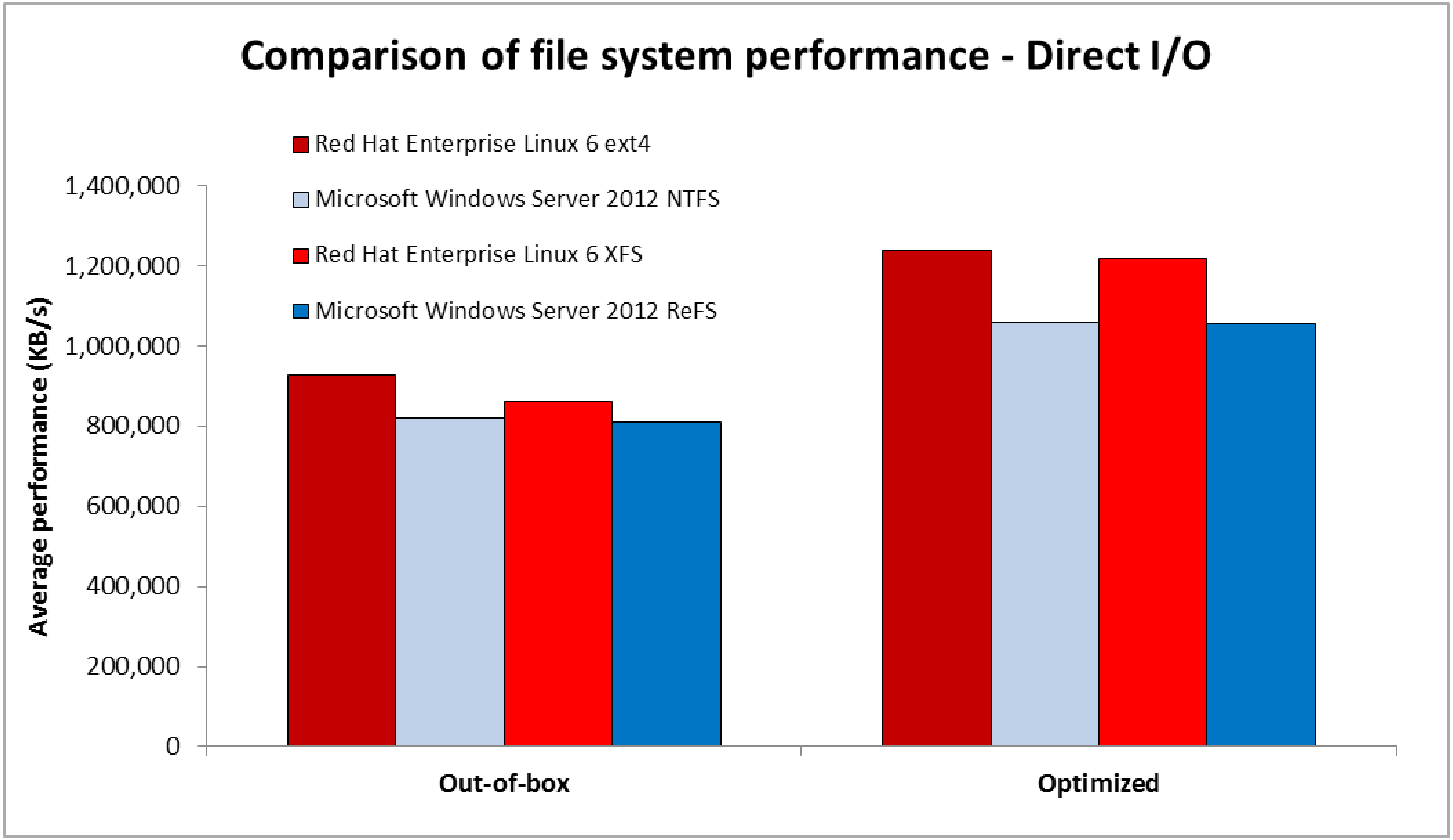
RHEL vs Windows Server 2012 Comparison
File system: in-cache file-access method



http://www.principledtechnologies.com/RedHat/RHEL6_IO_0613.pdf

RHEL vs Windows Server 2012 Comparison

File system: Direct I/O file-access method

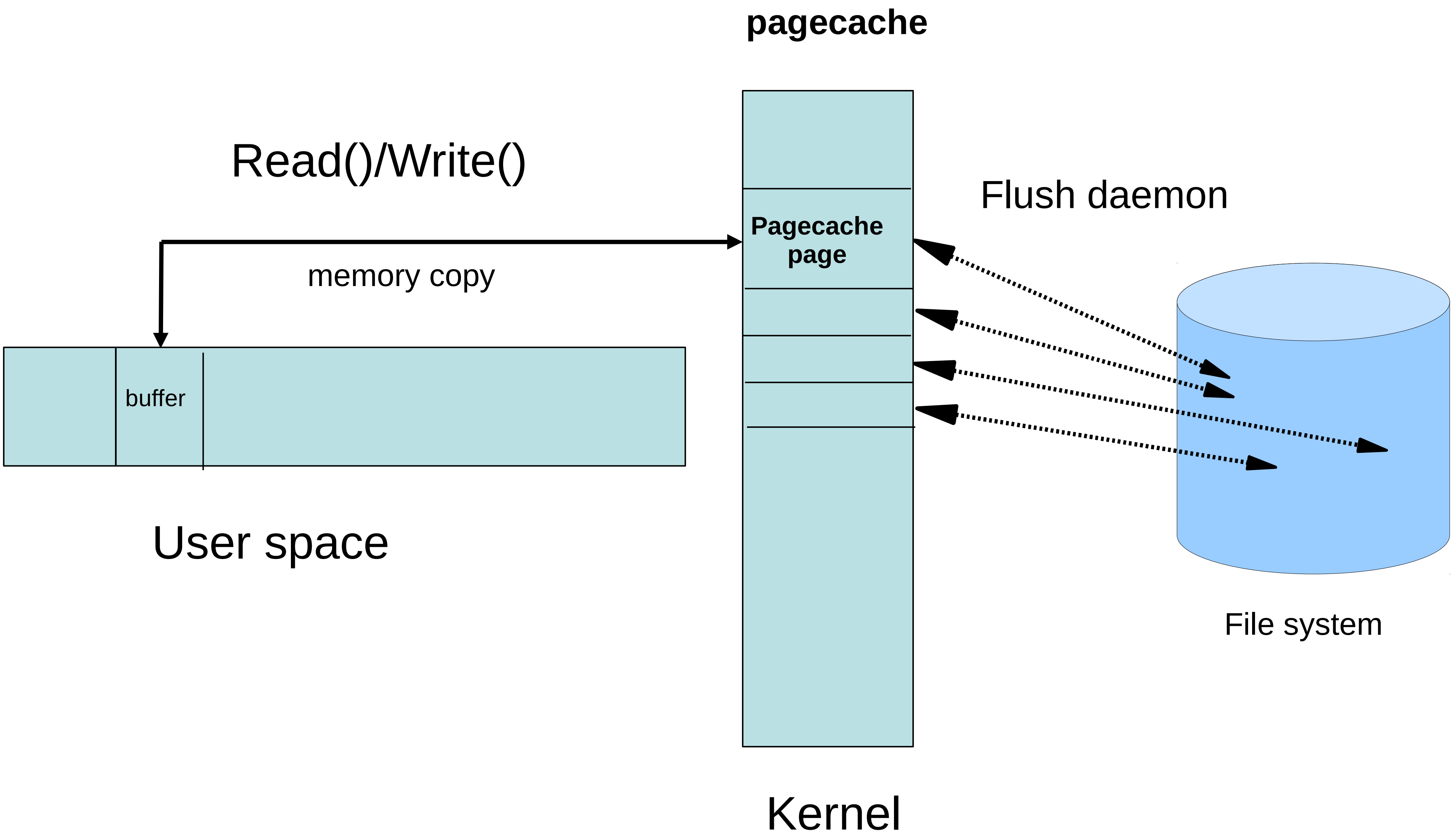


http://www.principledtechnologies.com/RedHat/RHEL6_IO_0613.pdf

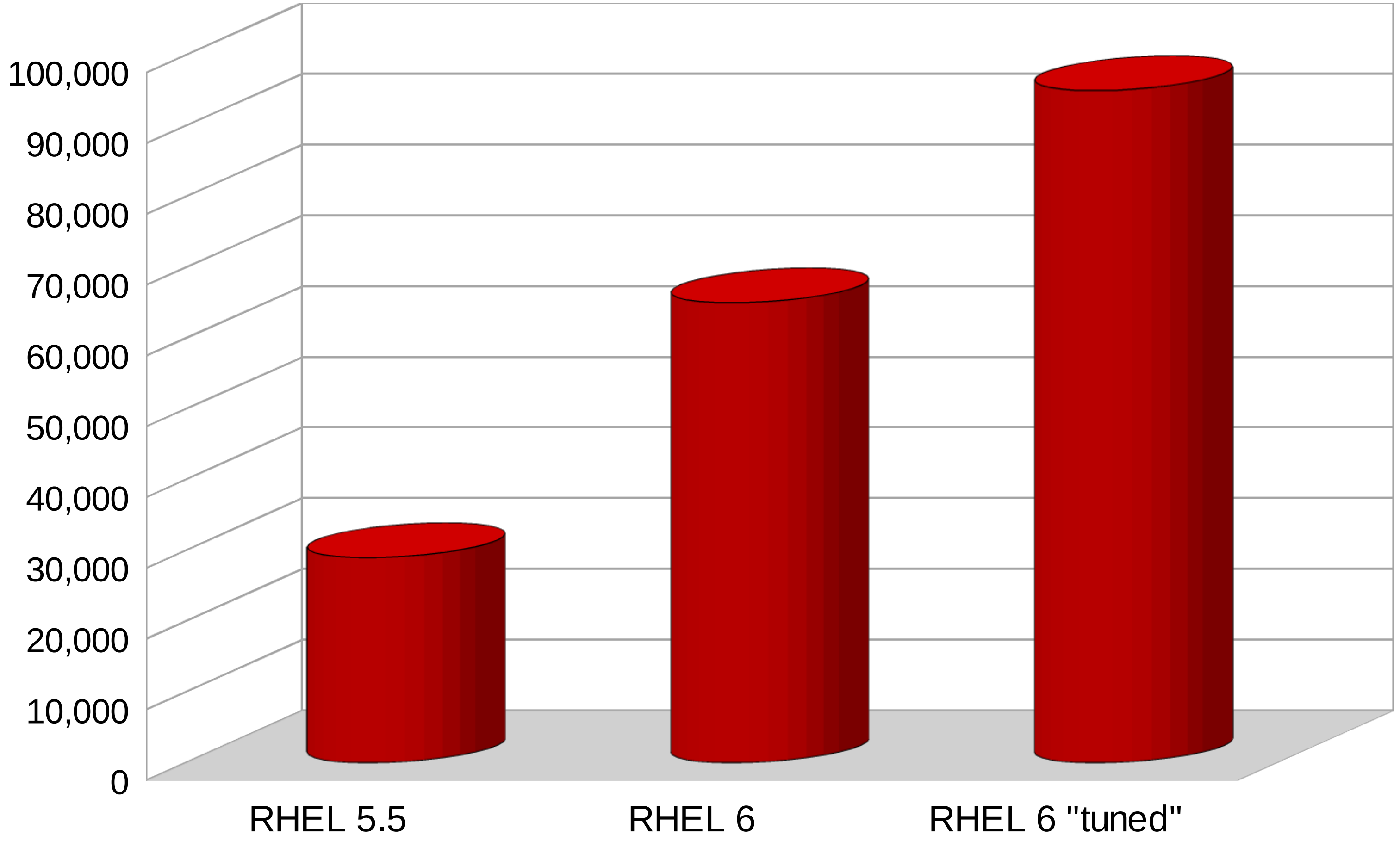
Per device/file/LUN page flush daemon

- Each file system or block device has its own flush daemon
- Allows different flushing thresholds and resources for each daemon/device/file system.
- Prevents some devices from not getting flushed because a shared daemon blocks used all resources
- Replaces pdflushd where a pool of threads flushed all devices.

Per file system flush daemon



High End HP DL 980 AIM7 results w/ “ktune” (r5) “tuned-adm” (r6)



HP DL980 64-core/256GB/30 FC/480 lun   AIM7 results w/ “tuned”

Virtual Memory Manager (VM) Tunables

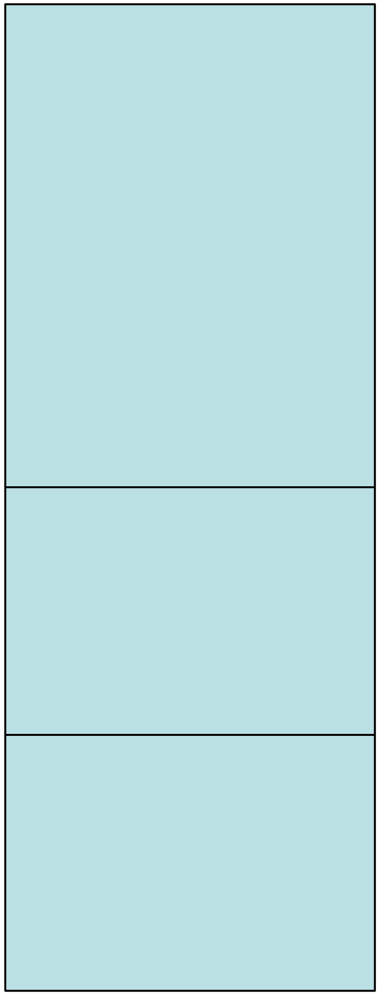
- Reclaim Ratios
  - /proc/sys/vm/swappiness
  - /proc/sys/vm/vfs_cache_pressure
  - /proc/sys/vm/min_free_kbytes

- Writeback Parameters
  - /proc/sys/vm/dirty_background_ratio
  - /proc/sys/vm/dirty_ratio

- Readahead parameters
  - /sys/block/<bdev>/queue/read_ahead_kb

dirty_ratio and dirty_background_ratio

pagecache



100% of pagecache RAM dirty

flushd and write()'ng processes write dirty buffers

dirty_ratio(20% of RAM dirty) – processes start synchronous writes

flushd writes dirty buffers in background

dirty_background_ratio(10% of RAM dirty) – wakeup flushd

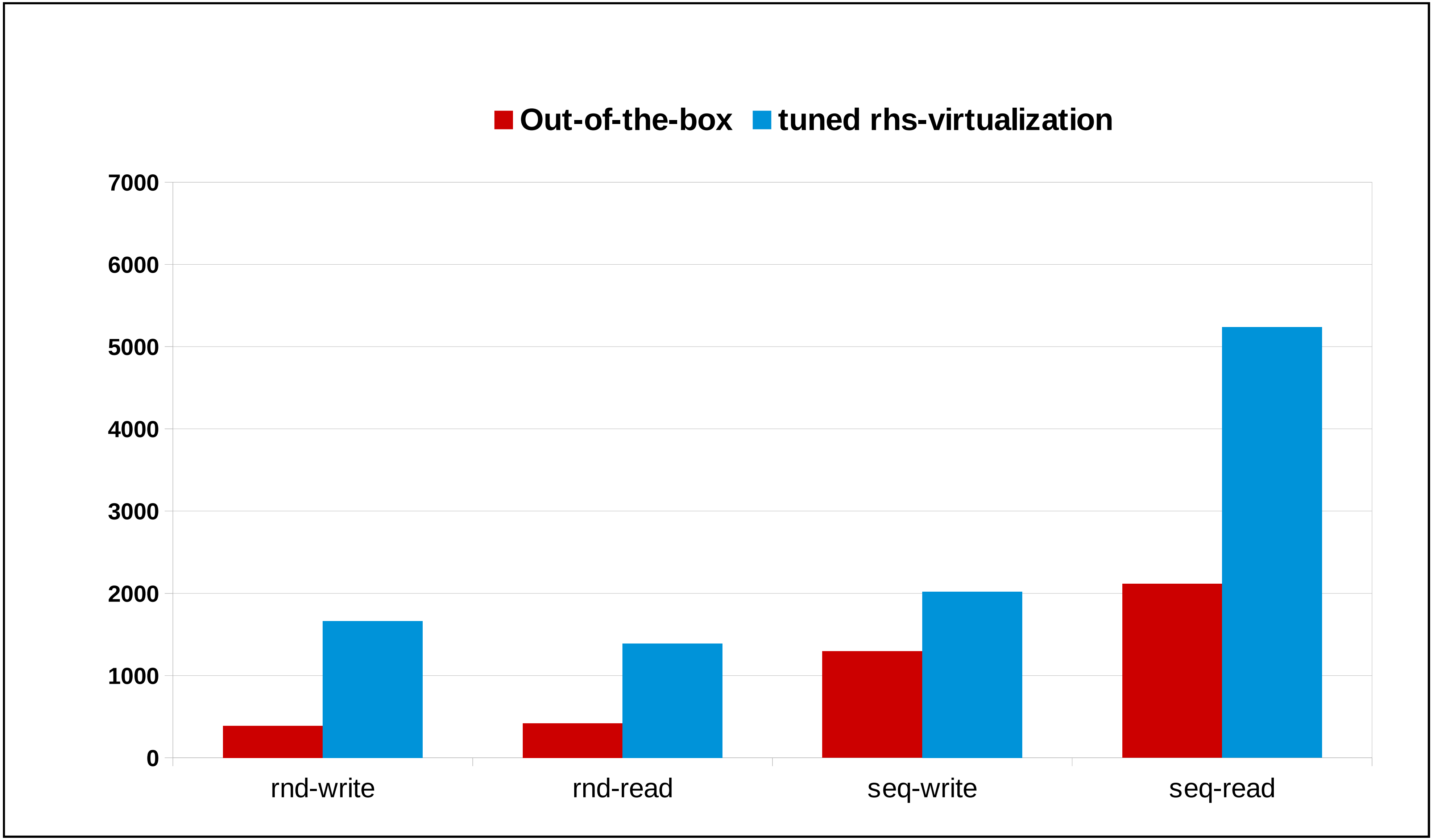
do_nothing

0% of pagecache RAM dirty

KVM / RHS Tuning

- **gluster volume set <*volume*> group virt**
- XFS mkfs -n size=8192, mount inode64, noatime
- RHS server: **tuned-adm profile rhs-virtualization**
  - Increase in readahead, lower dirty ratio's
- KVM host: **tuned-adm profile virtual-host**
  - Better response time shrink guest block device queue
    - `/sys/block/vda/queue/nr_request (16 or 8)`
  - Best sequential read throughput, raise VM read-ahead
    - `/sys/block/vda/queue/read_ahead_kb (4096/8192)`

Iozone Performance Comparison RHS2.1/XFS w/ RHEV



Summary / Questions

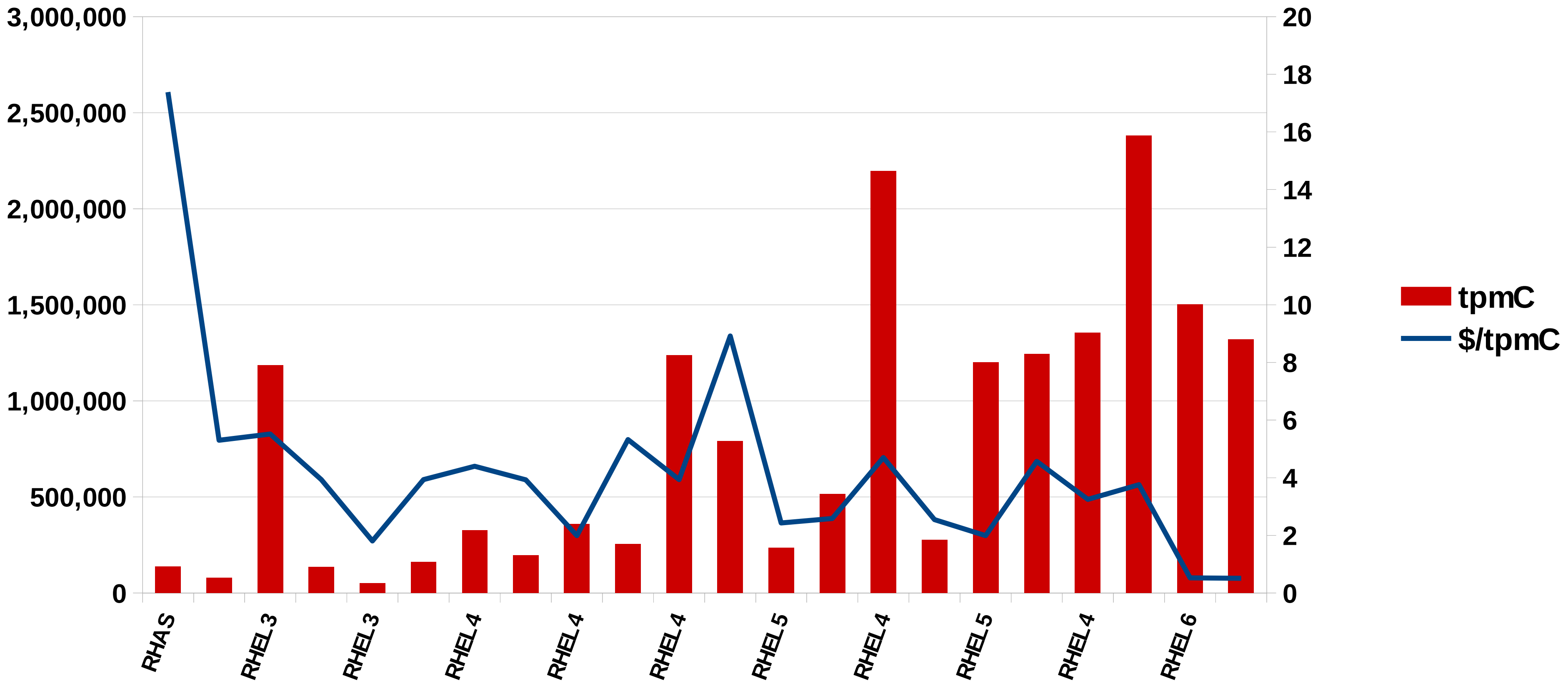
- Red Hat Enterprise Linux 6 Performance Features
  - “TUNED” tool – adjusts system parameters to match environments - throughput/latency.
  - Transparent Huge Pages – auto select large pages for anonymous memory, static hugepages for shared mem
  - Non-uniform Memory Access (NUMA)
    - NUMAstat enhancements
    - NUMActl for manual control
    - NUMAD daemon for auto placement
  - TUNA – integration w/ RHEL6.4

Helpful Links

- Performance Tuning Guide
- Low Latency Performance Tuning Guide
- Optimizing RHEL Performance by Tuning IRQ Affinity
- KVM Performance Guide
- STAC Network I/O SIG

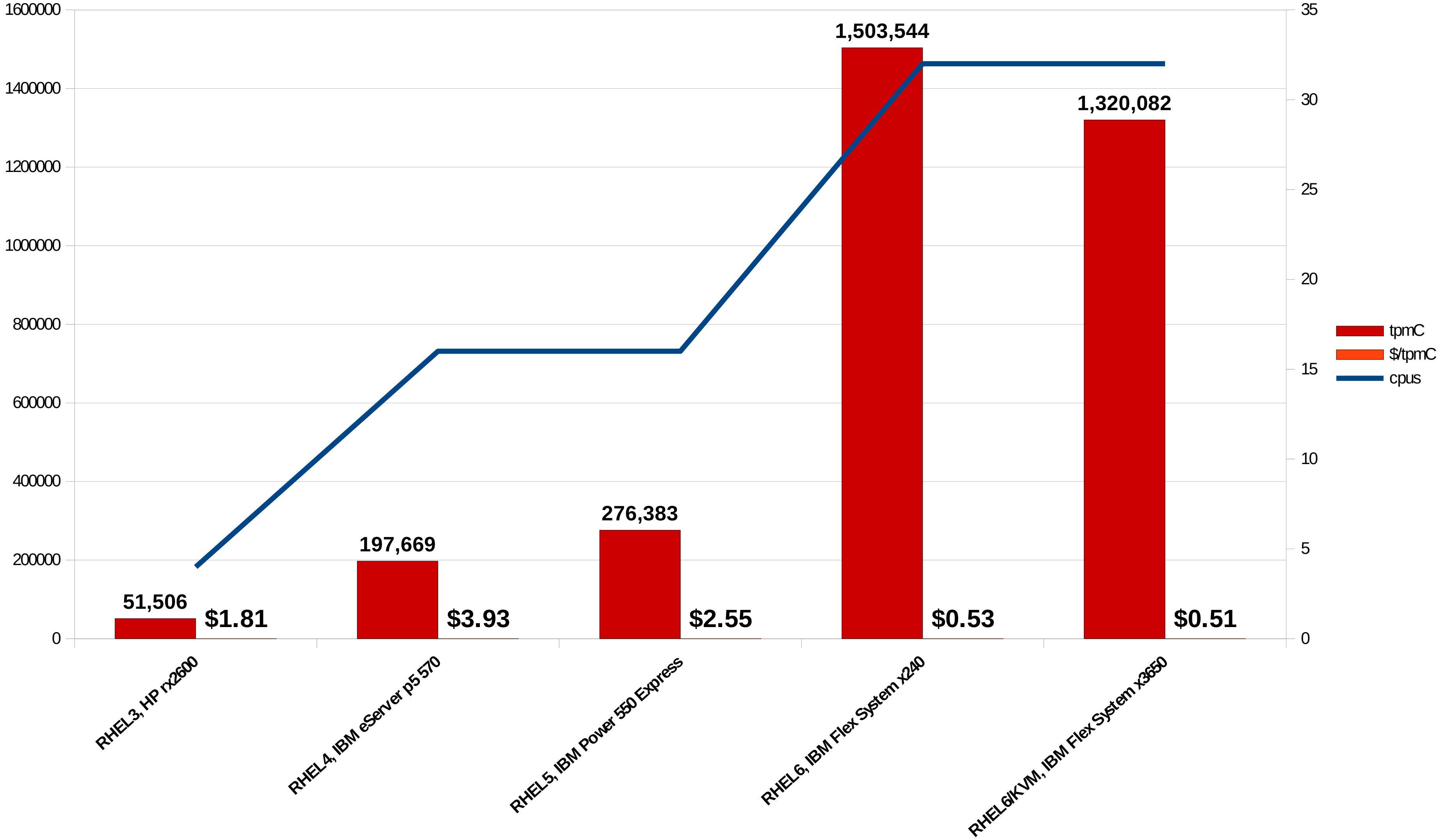
- Blog: http://www.breakage.org/ or @jeremyeder

RHEL Benchmark 10-year History in TPC-C

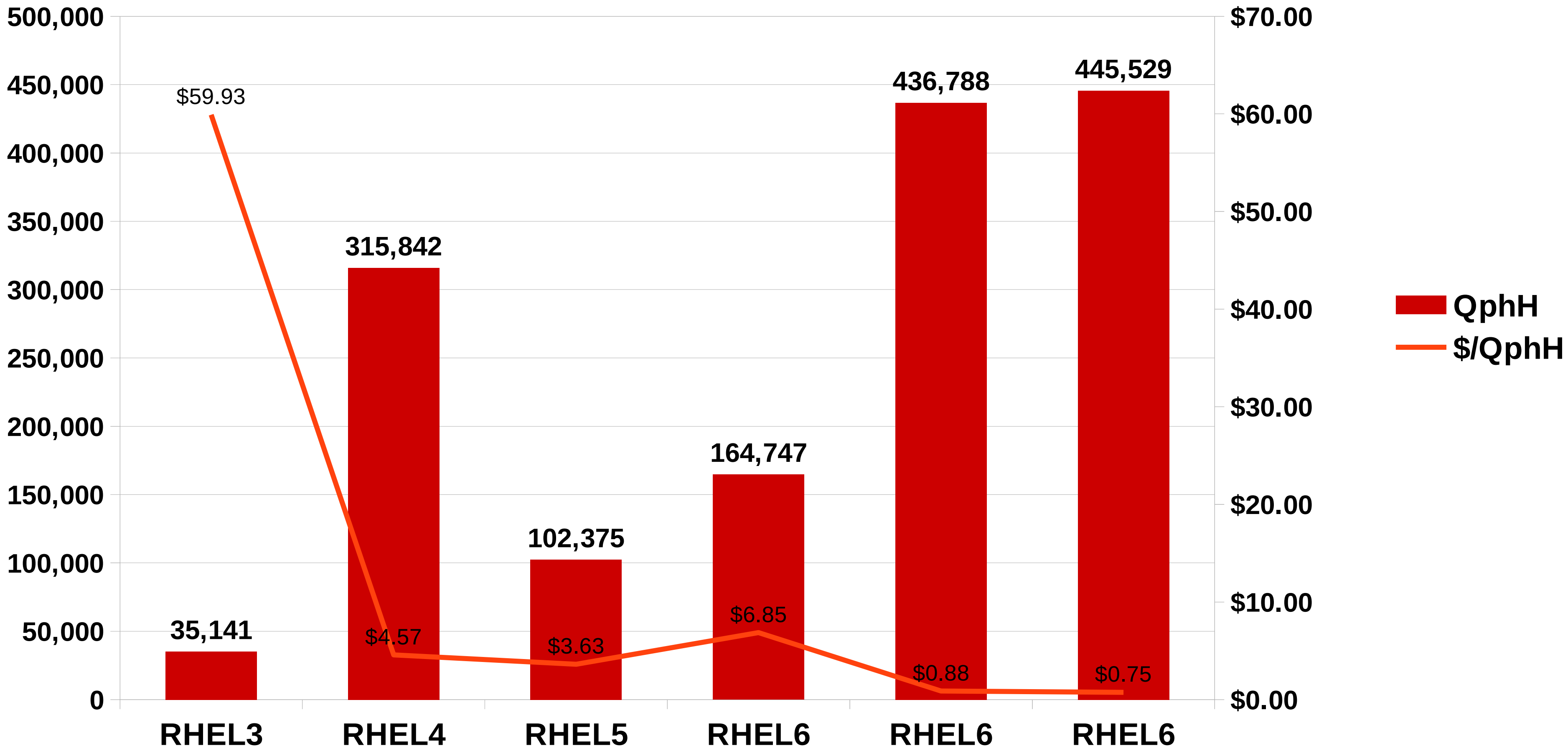


Comparison of TPC-C results using the Red Hat operating system. For more information about the TPC and the benchmark results referenced here see www.tpc.org.

RHEL6 Benchmark TPC-C- 2 socket improvements

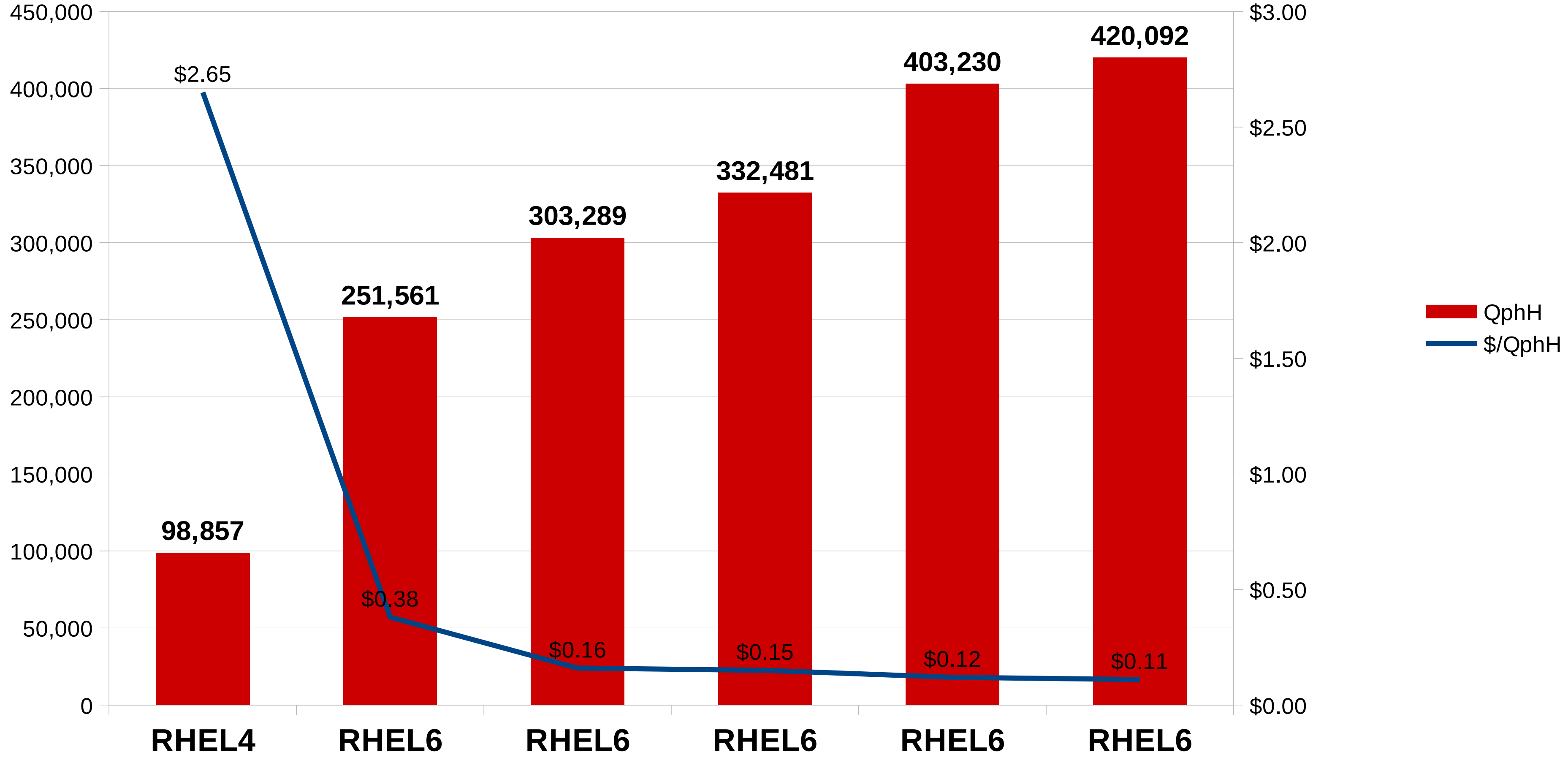


RHEL6 Benchmark 10 year History in TPC-H



Comparison of TPC-C results using the Red Hat operating system. For more information about the TPC and the benchmark results referenced here see www.tpc.org.

RHEL6 Benchmark 10-year History in TPC-H



Comparison of TPC-C results using the Red Hat operating system. For more information about the TPC and the benchmark results referenced here see www.tpc.org.