



RHEL 7 Performance Tuning

Joe Mario

Senior Principal Software Engineer

Sept 22, 2016

Agenda

- The RH performance team
- Problem areas where we spend most of our time:
 - System tuning
 - Numa issues
- RHEL 7 performance enhancements
- Where to learn more.
- Backup slides – if interest and time permits.

• Performance Engineering Team

**30+ Engineers around the globe
Striving for best performance for**

- **Micro-Benchmarks**
- **Applications/Benchmarks**
- **Application Scaling**
- **OS Scaling**
- **New performance feature support**
- **Partner support**
- **Customer support**

...

RHEL Platform(s) Performance Coverage

Benchmarks

- CPU – linpack, Imbench
- Memory – Imbench, McCalpin STREAM
- Disk IO – iozone, fio – SCSI, FC, iSCSI
- Filesystems – iozone, ext3/4, xfs, gfs2, gluster
- Networks – netperf – 10/40Gbit, Infiniband/RoCE, Bypass
- Bare Metal, RHEL6/7 KVM
- White box AMD/Intel, with our OEM partners
- TPC-C, TPC-H based workloads for database testing
- YCSB and SPECvirt for virtualization testing

Application Performance

- Linpack MPI, SPEC CPU, SPECjbb 05/13
- AIM 7 – shared, filesystem, db, compute
- Database: DB2, Oracle 11/12, Sybase 15.x, MySQL, MariaDB, PostgreSQL, Mongo
- OLTP – Bare Metal, KVM, RHEV-M clusters – TPC-C/Virt
- DSS – Bare Metal, KVM, RHEV-M, IQ, TPC-H/Virt
- SPECsfs NFS
- SAP – SLCS, SD, HANA

RHEL Performance Evolution

RHEL5

Static Hugepages

CPU Sets

Ktune on/off

CPU Affinity
(taskset)

NUMA Pinning
(numactl)

irqbalance

RHEL6

Transparent
Hugepages

Tuned - Choose
Profile

NUMAD -
userspace

cgroups

irqbalance -
NUMA enhanced

RHEL7

Tuned -
throughput-
performance
(default)

Automatic NUMA-
balancing kernel
scheduler

Containers/Docker

Irqbalance -
NUMA enhanced

RH Cloud

RHEV tuned
profile

RHEL OSP7
Tuned, NUMA,
SR-IOV

RHEL Atomic
Host, Atomic Ent

OpenShift v3

CloudForms

Agenda

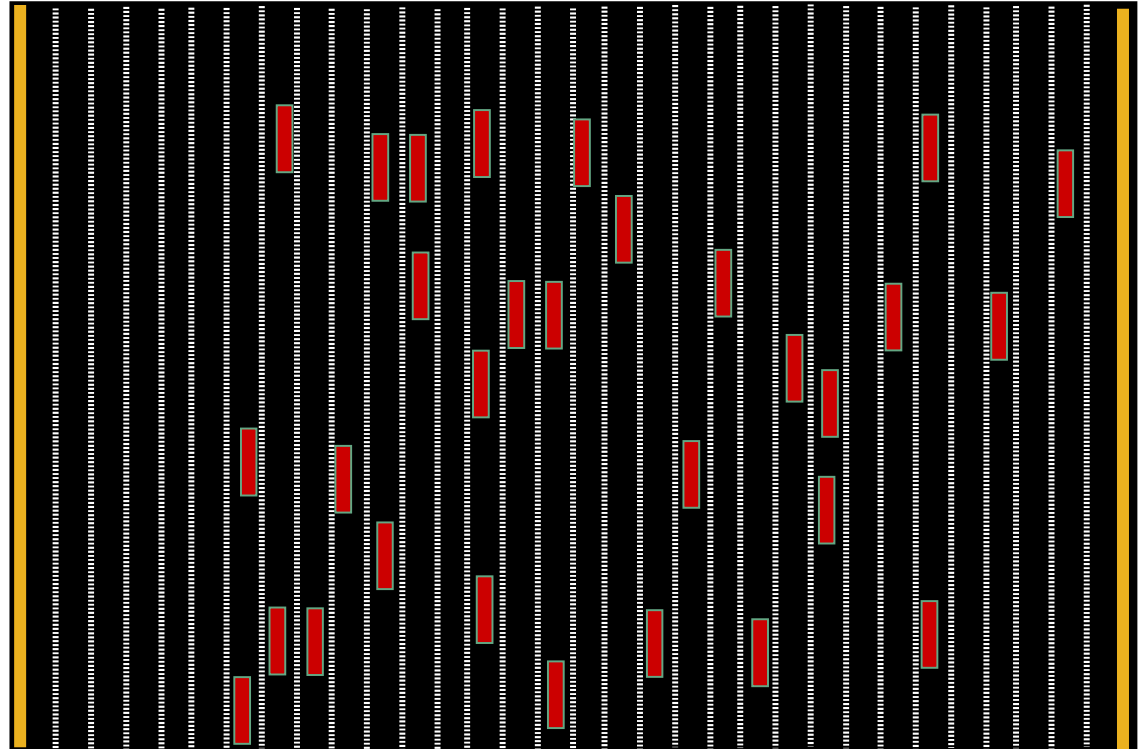
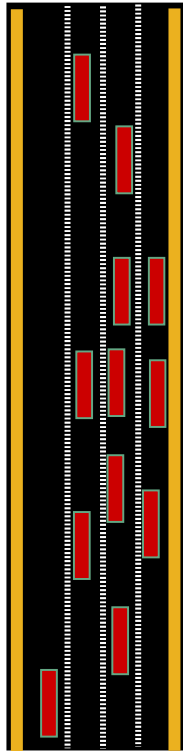
Problem areas where we spend most of our time.

First:

System tuning – with “tuned”

Performance Metrics

Latency==Speed. Throughput==Bandwidth



Latency – Speed Limit

- Ghz of CPU, Memory PCI
- Small transfers, disable aggregation – TCP nodelay
- Dataplane optimization DPDK

Throughput – Bandwidth - # lanes in Highway

- Width of data path / cachelines
- Bus Bandwidth, QPI links, PCI 1-2-3
- Network 1 / 10 / 40 Gb – aggregation, NAPI
- Fiberchannel 4/8/16, SSD, NVME Drivers

What is “tuned” ?

- Tuning profile delivery mechanism
- Red Hat ships *tuned profiles* that improve performance for many workloads...hopefully yours!
- Customize your own profile.

Is your OS is optimally tuned?

What is my system currently tuned to?

tuned-adm active

Current active profile: balanced

How do I change my current tuning setting?

tuned-adm profile network-latency

Tuned (cont)

tuned-adm list

Available profiles:

- balanced
- desktop
- latency-performance
- network-latency
- network-throughput
- powersave
- throughput-performance
- virtual-guest
- virtual-host

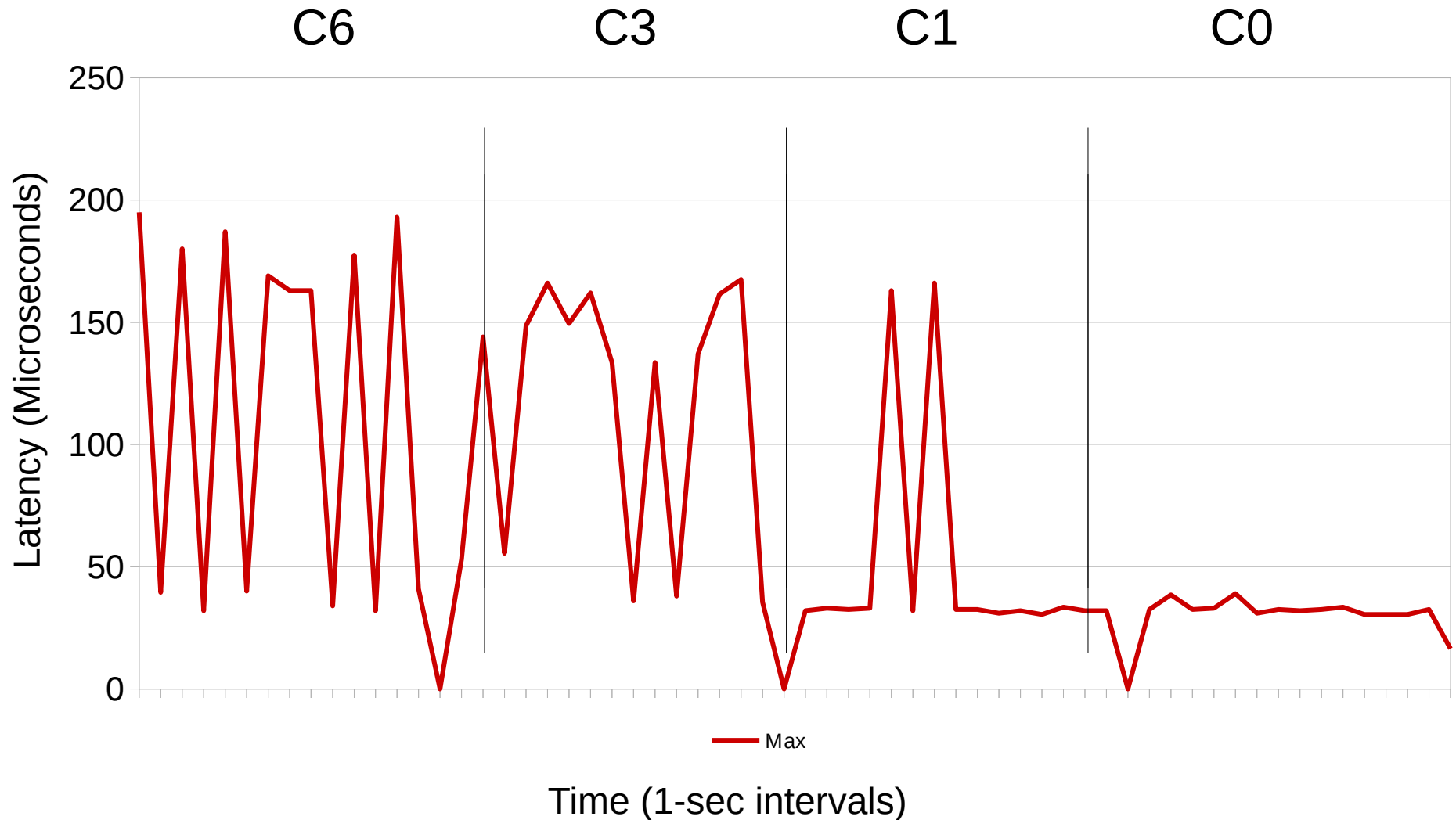
Current active profile: balanced

Tuned Updates for RHEL7

- Installed by default!
- Profiles updated for RHEL7 features and characteristics
- Profiles automatically set based on installation
 - Desktop/Workstation: balanced profile
 - Server/HPC: throughput-performance profile
- Optional hook/callout capability
- Concept of Inheritance (just like `httpd.conf`)

Tuned: Network Latency Performance Boost

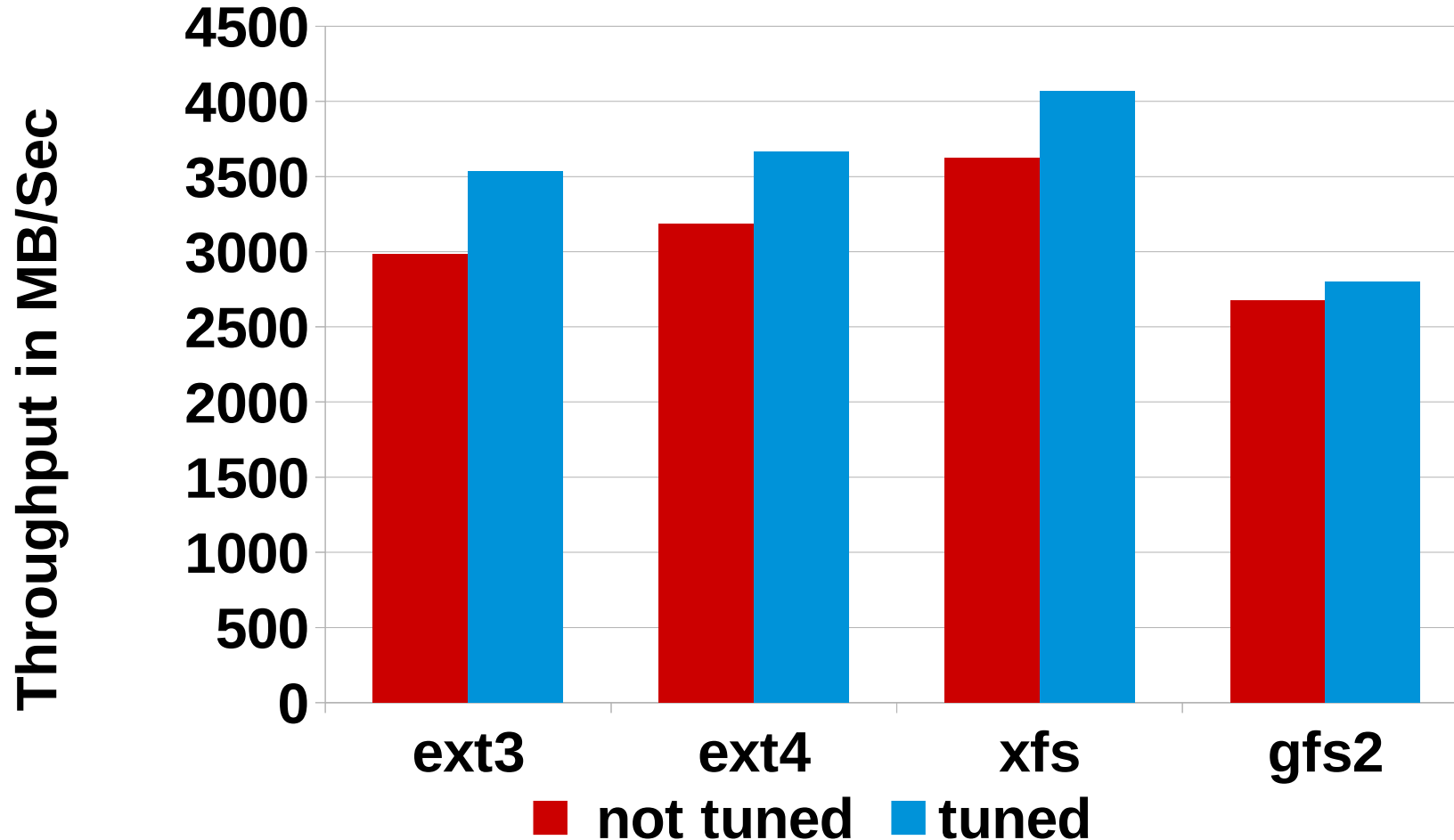
C-state lock improves determinism, reduces jitter



Tuned: Storage Performance Boost

RHEL7 File System In Cache Perf

Intel I/O (iozone - geoM 1m-4g, 4k-1m)



Larger is better

throughput-performance (RHEL7 default)

- governor=**performance**
- energy_perf_bias=**performance**
- min_perf_pct=**100**
- readahead=**4096**
- kernel.sched_min_granularity_ns = **10000000**
- kernel.sched_wakeup_granularity_ns = **15000000**
- vm.dirty_background_ratio = **10**
- vm.swappiness=**10**

Tuned: Profile Inheritance (throughput)

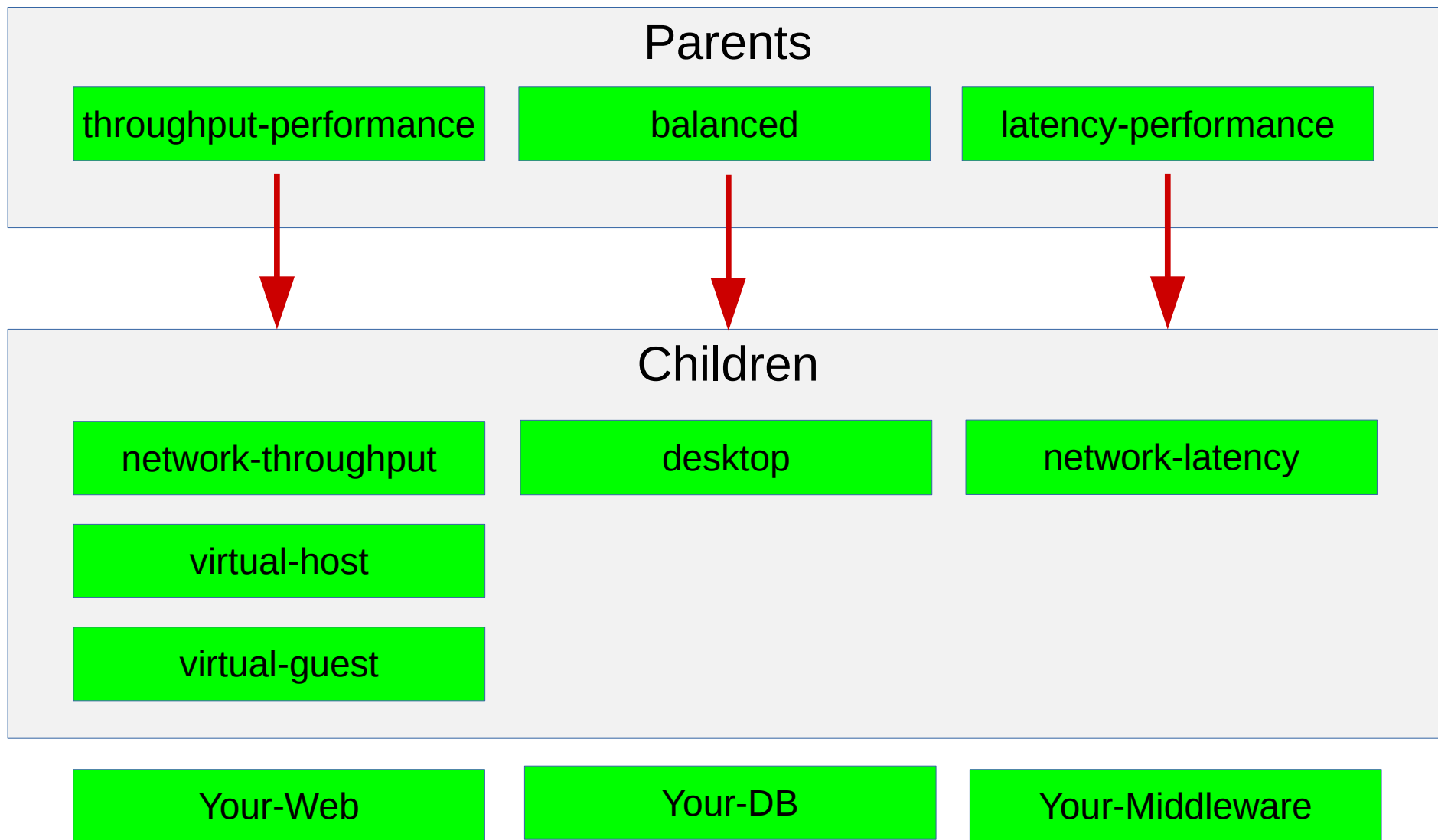
throughput-performance

```
governor=performance
energy_perf_bias=performance
min_perf_pct=100
readahead=4096
kernel.sched_min_granularity_ns = 10000000
kernel.sched_wakeup_granularity_ns = 15000000
vm.dirty_background_ratio = 10
vm.swappiness=10
```

network-throughput

```
net.ipv4.tcp_rmem="4096 87380 16777216"
net.ipv4.tcp_wmem="4096 16384 16777216"
net.ipv4.udp_mem="3145728 4194304 16777216"
```

Tuned: Profile Inheritance



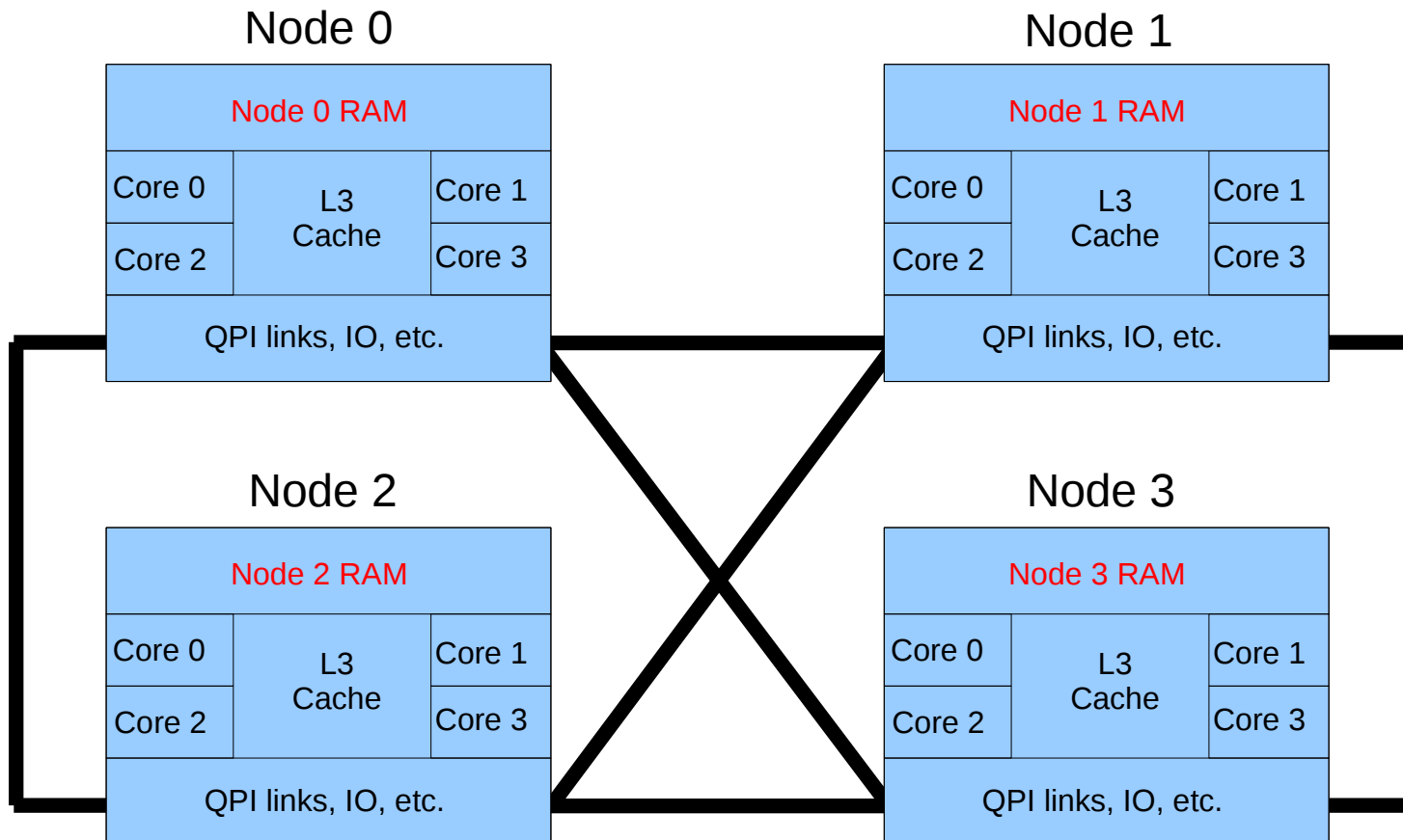
Agenda

Two hottest categories where we spend most of our time:

Second:

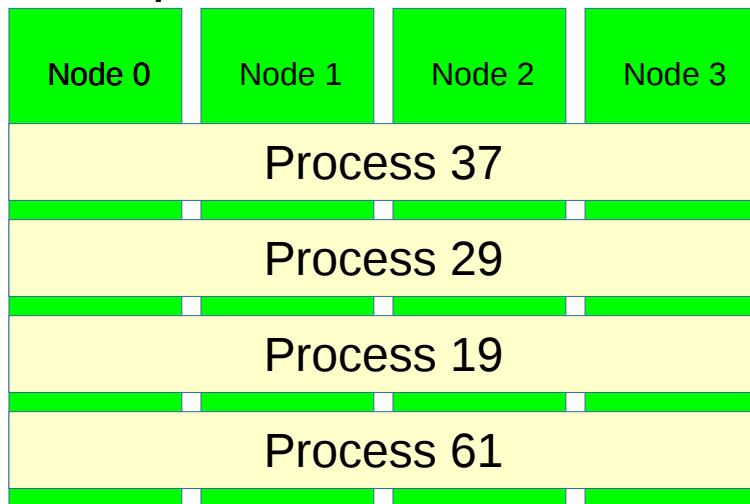
NUMA issues

Typical NUMA System

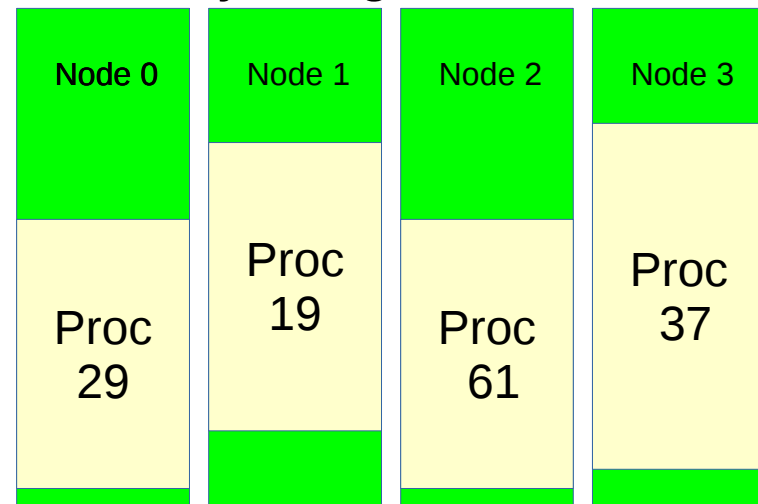


Goal: Align process memory and CPU threads within nodes

Before: processes use cpus & memory from multiple nodes.



After: processes have more “localized” cpu & memory usage.



Numa Due Diligence

- Know your hardware
 - lstopo
 - numactl --hardware
 - Install adapters “close” to the CPU that will run the critical application
 - When BIOS reports locality, irqbalance handles NUMA/IRQ affinity automatically.

Numa Due Diligence (cont)

- Know your application's memory usage
 - `numastat -m cv <proc_name>`
- Understand where processes are executing and the memory they access
 - Run “top”, then enter “f”, then select “Last used cpu” field
 - `ps -T -o pid,tid,psr,comm`
- Use process placement tools
 - `numactl, taskset`
 - `mbind, set_mempolicy, sched_setaffinity, pthread_setaffinity_np`
- Virtualized environments – just as important!

numastat: per-node meminfo (new)

```
# numastat -mczs
```

	Node 0	Node 1	Total
	-----	-----	-----
MemTotal	65491	65536	131027
MemFree	60366	59733	120099
MemUsed	5124	5803	10927
Active	2650	2827	5477
FilePages	2021	3216	5238
Active(file)	1686	2277	3963
Active(anon)	964	551	1515
AnonPages	964	550	1514
Inactive	341	946	1287
Inactive(file)	340	946	1286
Slab	380	438	818
SReclaimable	208	207	415
SUnreclaim	173	230	403
AnonHugePages	134	236	370

numastat – per-PID mode

```
# numastat -c java (default scheduler - non-optimal)
```

```
Per-node process memory usage (in MBs)
```

PID	Node 0	Node 1	Node 2	Node 3	Total
57501 (java)	755	1121	480	698	3054
57502 (java)	1068	702	573	723	3067
57503 (java)	649	1129	687	606	3071
57504 (java)	1202	678	1043	150	3073
Total	3674	3630	2783	2177	12265

```
# numastat -c java (numabalance close to opt)
```

```
Per-node process memory usage (in MBs)
```

PID	Node 0	Node 1	Node 2	Node 3	Total
56918 (java)	49	<u>2791</u>	56	37	2933
56919 (java)	<u>2769</u>	76	55	32	2932
56920 (java)	19	55	77	<u>2780</u>	2932
56921 (java)	97	65	<u>2727</u>	47	2936
Total	2935	2987	2916	2896	11734

Visualize CPUs via Istopo

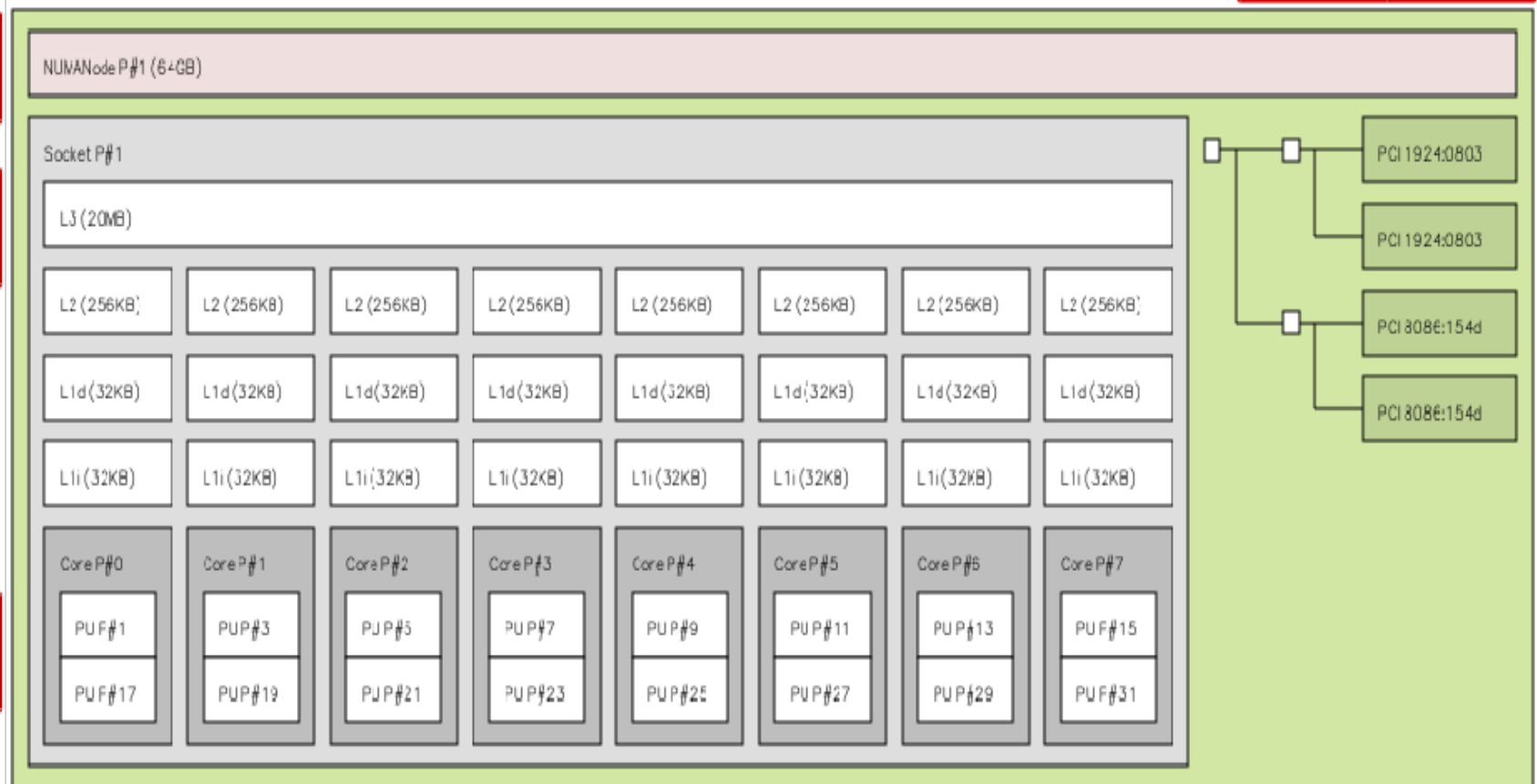
Istopo

PCIe

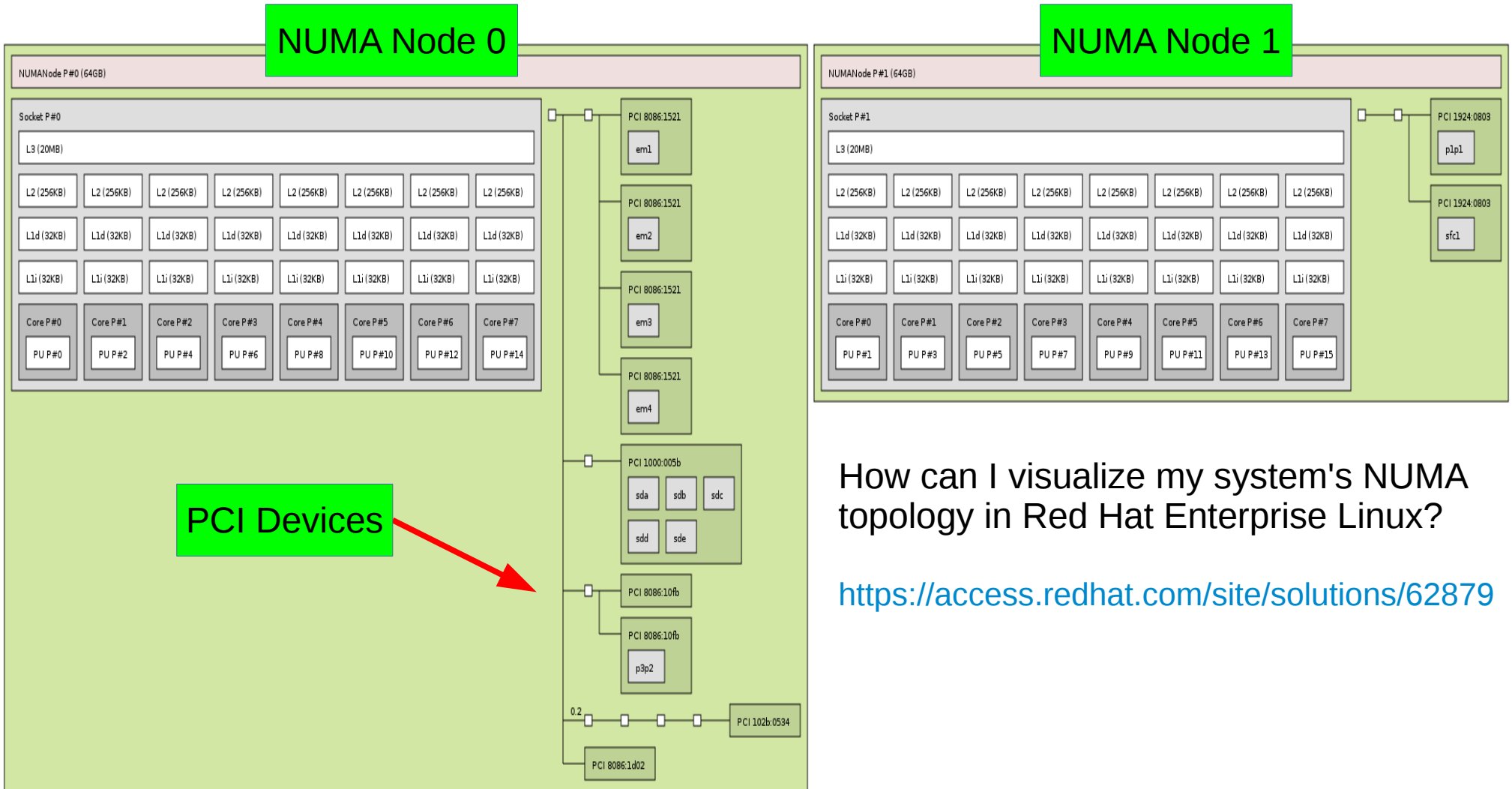
NUMA

CACHE

HT



Visualize NUMA Topology: Istopo



How can I visualize my system's NUMA topology in Red Hat Enterprise Linux?

<https://access.redhat.com/site/solutions/62879>

NUMA layout via numactl

```
# numactl --hardware
```

```
available: 4 nodes (0-3)
```

```
node 0 cpus: 0 4 8 12 16 20 24 28 32 36
```

```
node 0 size: 65415 MB
```

```
node 0 free: 63482 MB
```

```
node 1 cpus: 2 6 10 14 18 22 26 30 34 38
```

```
node 1 size: 65536 MB
```

```
node 1 free: 63968 MB
```

```
node 2 cpus: 1 5 9 13 17 21 25 29 33 37
```

```
node 2 size: 65536 MB
```

```
node 2 free: 63897 MB
```

```
node 3 cpus: 3 7 11 15 19 23 27 31 35 39
```

```
node 3 size: 65536 MB
```

```
node 3 free: 63971 MB
```

```
node distances:
```

```
node  0  1  2  3
```

```
  0:  10  21  21  21
```

```
  1:  21  10  21  21
```

```
  2:  21  21  10  21
```

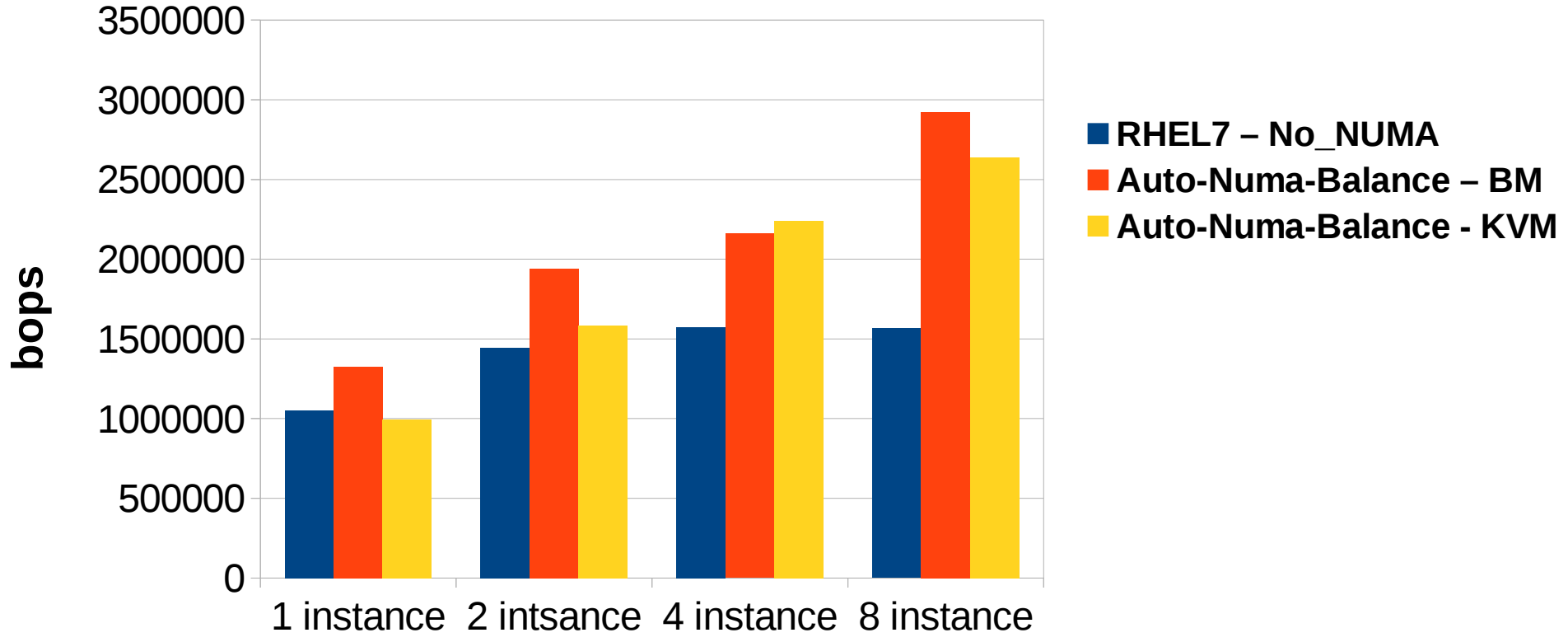
```
  3:  21  21  21  10
```


RHEL NUMA Scheduler

- RHEL6
 - numactl, numastat enhancements
 - numad – usermode tool, dynamically monitor, auto-tune
- RHEL7 – auto numa balancing
 - Moves tasks (threads or processes) closer to the memory they are accessing.
 - Moves application data to memory closer to the tasks that reference it.
 - A win for most apps.
 - Enable / Disable
 - `sysctl kernel.numabalancing={0,1}`

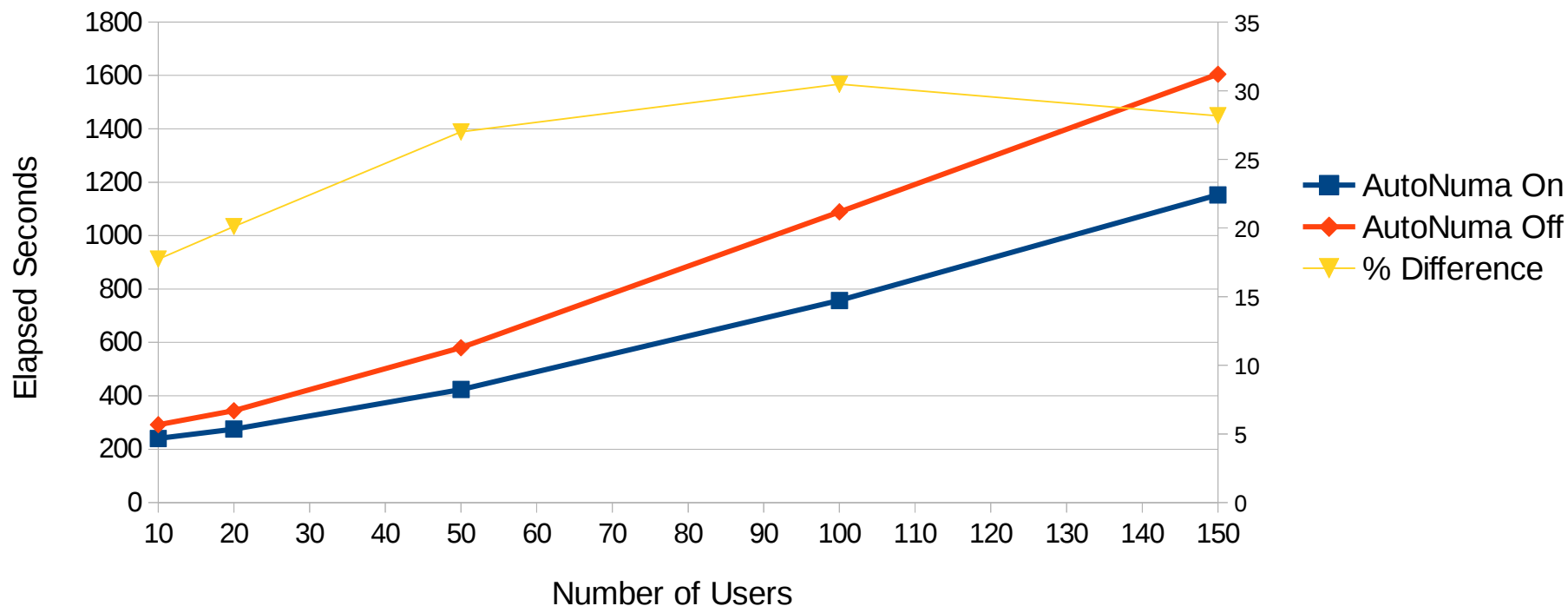
RHEL7 Auto-Numa-Balance SPECjbb2005 multi-instance - bare metal + kvm

8 socket, 80 cpu, 1TB mem



RHEL 7 AutoNuma kernel scheduler benefits (ideal case)

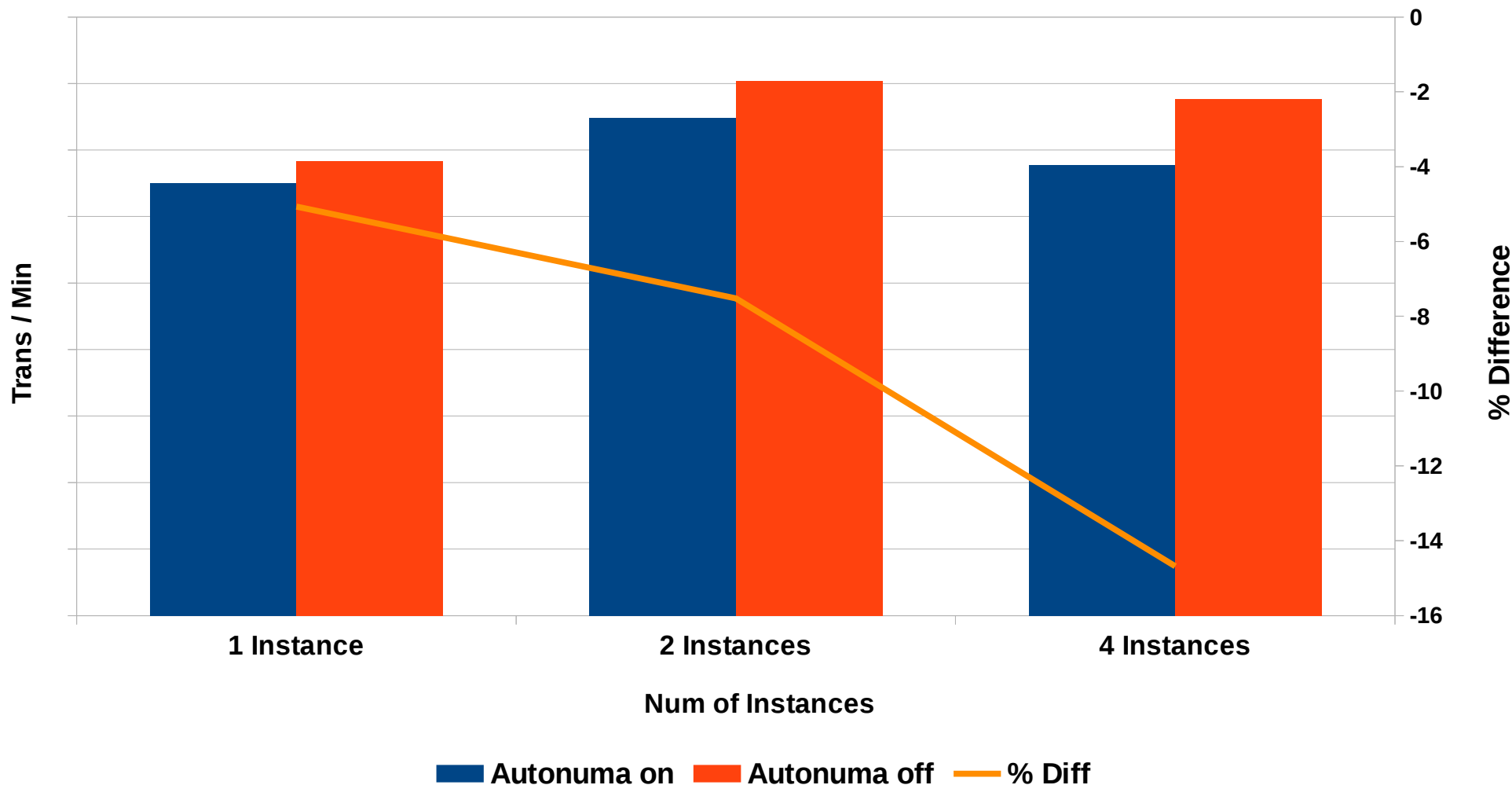
SAP HANA benchBWEMLSim - MultiProvider Elapsed Time



HANA sps09 – prior to the sps10 “numa-aware” HANA

Database OLTP workload on 4 Socket System - 4 NUMA nodes - 64 cpus - 128G memory

1, 2 and 4 instances with and without Autonuma for 100 User set

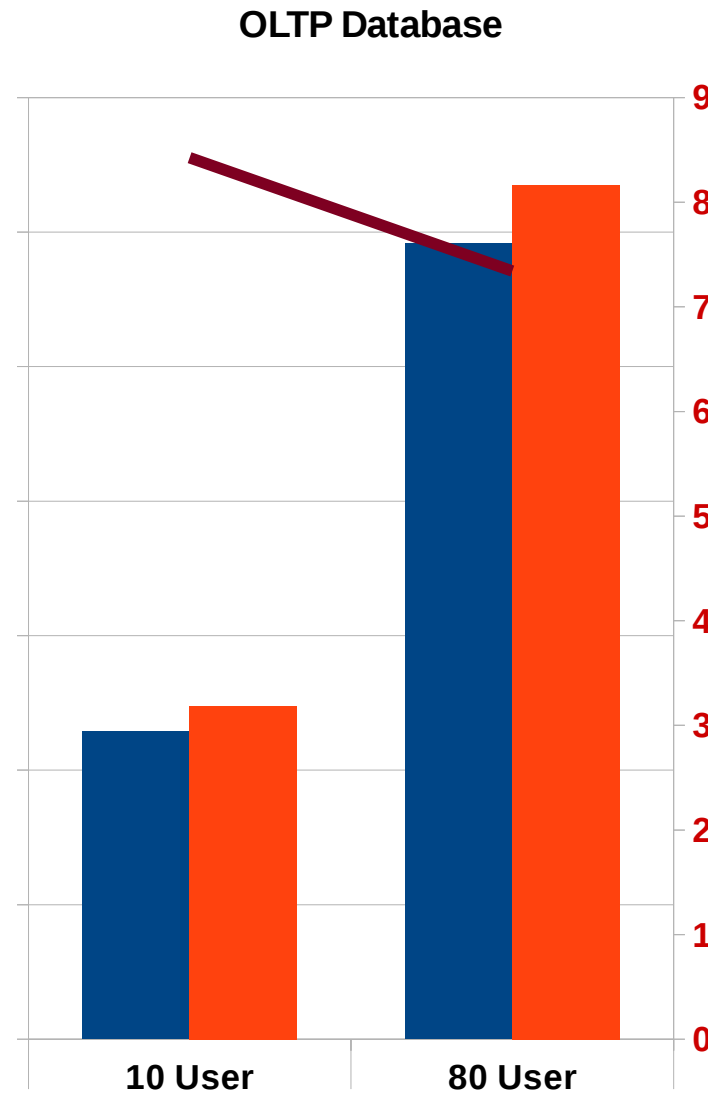


- With hugepages, there is no performance difference with Autonuma on or off because the memory is wired down
- `# echo 0 > /proc/sys/kernel/numa_balancing`

Memory Tuning – Huge Pages

- 2M pages vs 4K standard linux page
- Virtual to physical page map is 512 times smaller
- TLB can map more physical pages, resulting in fewer misses
- Traditional Huge Pages always pinned
- Most databases support Huge pages
- 1G pages supported on newer hardware
- Transparent Huge Pages in RHEL6 (cannot be used for Database shared memory – only for process private memory)

Memory Tuning – huge pages on Bare Metal



Agenda

Sampling of performance features in RHEL 7

Performance Optimizations in RHEL7

Networking

- Full support for PTP1588v2
 - Precision Time Protocol (finally in RHEL)
 - More accurate and better fault tolerance
- Route cache → F.I.B. routing cache algorithm
 - FIB - more secure, scalable, but initially slower
 - Scaling problems fixed in RHEL 7.2 (much faster routing perf)
- irqbalance handles NUMA
- busy_poll
 - Polling vs interrupts - big win
-

Performance Optimizations in RHEL7

Networking (continued)

- `tcp_fastopen`
 - Reduce 1 round trip of handshake setting up TCP connection.
- `nohz_full` (tickless while active)
 - Timer ticks only on boot cpu or selected cpus
- **Byte Queue Limits**
 - Control bufferbloat in network queues
 - Helps tune high prio packets to get delivered w/reasonable latency
- **TCP Small Queues.**
 - Initially in RHEL 6.5, but set it too small (hurt perf)
 - Corrected it in RHEL 6.6, 6.7, & 7.x

Performance Optimizations in RHEL7

Memory

- Automatic NUMA Balancing
- Tunable workqueues (writeback)

CPU

- Support for all new CPUs
- AVX2 instruction support
- RHEL-RT - sync w/RHEL 7.x releases.

Power Management

- intel_pstate
- tuned does most heavy lifting

Low Latency Performance Tuning Guide for Red Hat Enterprise Linux 7

- Tactical tuning overview for latency-sensitive workloads.
- Emphasizes impactful new features included in RHEL7:
 - CPU/power management
 - NUMA
 - tuned profiles
 - scheduling
 - network tunables
 - kernel timers.
 - "de-jittering" CPU cores
 - tracing techniques

<https://access.redhat.com/articles/1323793>

I/O Tuning Database Layout - monitoring I/O - iostat -dmxz 3

LVM - Fibre Channel

Device:	rrqm/s	wrqm/s	r/s	w/s	rMB/s	wMB/s	avgrq-sz	avgqu-sz	await	r_await	w_await	svctm	%util
dm-2	17.20	2418.60	383.00	971.40	9.47	45.55	83.20	3.97	2.92	8.95	0.55	0.71	96.60
dm-4	19.00	2420.60	379.80	966.00	9.67	45.57	84.06	4.04	2.99	9.28	0.52	0.73	98.22
dm-15	21.20	2426.60	396.00	982.80	9.47	45.55	81.73	4.22	3.06	9.19	0.59	0.71	98.32
dm-2	16.00	2462.00	344.00	986.20	8.68	46.58	85.08	3.72	2.80	9.38	0.51	0.71	94.86
dm-4	19.20	2468.20	366.60	989.60	9.42	46.58	84.58	3.95	2.92	9.42	0.51	0.72	97.60
dm-15	18.20	2461.20	367.40	989.60	9.26	46.58	84.27	3.80	2.81	8.82	0.57	0.72	98.10
dm-2	17.00	2349.80	320.80	909.60	8.15	45.08	88.60	3.31	2.69	8.55	0.62	0.79	96.86
dm-4	15.40	2333.60	327.00	911.00	8.29	45.08	88.29	3.44	2.79	8.90	0.59	0.78	97.16
dm-15	17.80	2351.20	331.60	913.00	8.49	45.08	88.15	3.32	2.67	8.29	0.63	0.76	94.86

LVM - Mix (Fibre Channel - SSD)

dm-2	3.60	1848.20	391.60	826.20	9.14	35.35	74.83	3.48	2.85	7.77	0.52	0.79	96.08
dm-4	4.60	1851.60	397.40	839.40	9.34	35.34	73.98	3.35	2.71	7.36	0.50	0.78	96.56
dm-15	2.60	1895.40	391.00	849.00	9.57	35.34	74.17	3.39	2.74	7.57	0.52	0.78	97.10
fioa	0.00	0.00	357.00	1753.00	8.94	35.34	42.99	0.00	0.08	0.18	0.05	0.00	0.00
dm-2	2.80	1795.40	339.80	812.80	8.25	35.16	77.14	3.05	2.65	7.69	0.54	0.83	95.40
dm-4	4.40	1819.80	349.20	828.00	8.53	35.16	76.02	3.15	2.68	7.77	0.53	0.81	95.30
dm-15	3.60	1853.00	351.60	834.00	8.53	35.17	75.49	3.08	2.60	7.51	0.53	0.80	94.52
fioa	0.00	0.00	353.80	1715.40	8.51	35.15	43.21	0.01	0.08	0.17	0.07	0.00	0.18
dm-2	2.60	1775.00	325.80	797.80	7.57	34.55	76.76	2.91	2.59	7.58	0.55	0.85	95.02
dm-4	3.00	1786.40	317.20	814.40	7.57	34.55	76.24	2.79	2.46	7.38	0.55	0.83	94.08
dm-15	3.00	1809.60	311.40	814.00	7.75	34.57	77.01	2.76	2.46	7.42	0.56	0.85	95.24
fioa	0.00	0.00	299.40	1676.40	7.57	34.55	43.66	0.00	0.08	0.18	0.06	0.00	0.00

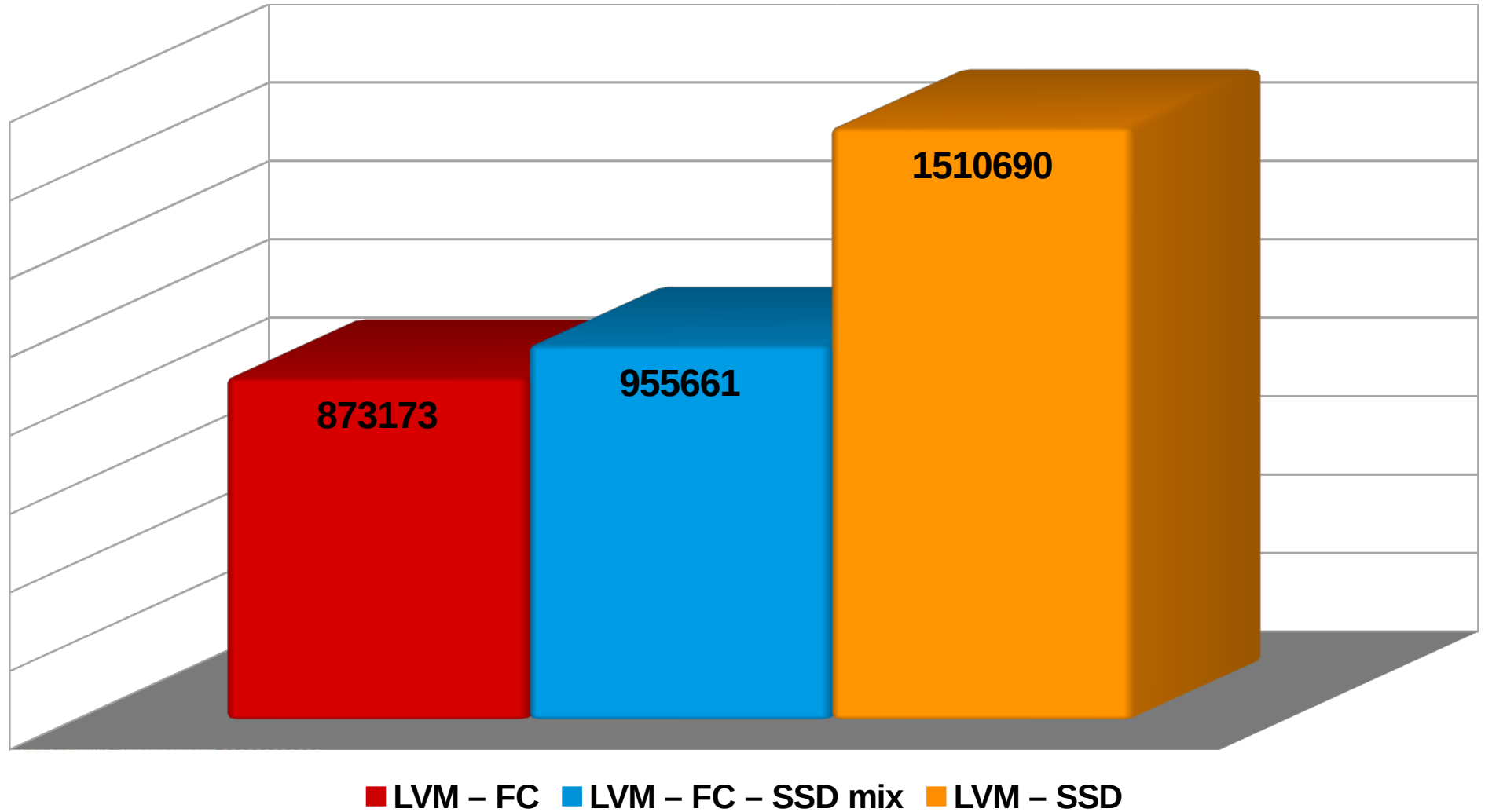
LVM - SSDs

fioa	0.00	0.00	332.60	7110.80	8.87	79.43	24.30	0.12	0.06	0.20	0.05	0.02	11.54
fiob	0.00	0.00	319.40	7060.40	8.59	79.39	24.42	0.79	0.08	0.28	0.07	0.03	24.48
fioa	0.00	0.00	313.00	7083.80	7.72	78.31	23.82	0.01	0.06	0.22	0.06	0.00	0.36
fiob	0.00	0.00	299.80	7109.60	7.83	78.33	23.81	0.23	0.08	0.27	0.07	0.02	11.32
fioa	0.00	0.00	306.00	7049.80	8.06	79.19	24.29	0.00	0.06	0.22	0.06	0.00	0.22
fiob	0.00	0.00	291.60	7060.00	7.88	79.19	24.26	0.98	0.08	0.28	0.07	0.07	49.28

I/O Tuning – Database Layout

OLTP Workload - Using volume manager

Single Instance - LVM - Fibre Channel and SSD



Agenda

New or enhanced performance features in RHEL 7 (and RHEL 6.7)

- dmcache

What is dmcache?

Maps block devices onto higher-level virtual block devices.

Allows fast storage, such as SSDs, to act as a cache for slower storage, such as hard disk drives.

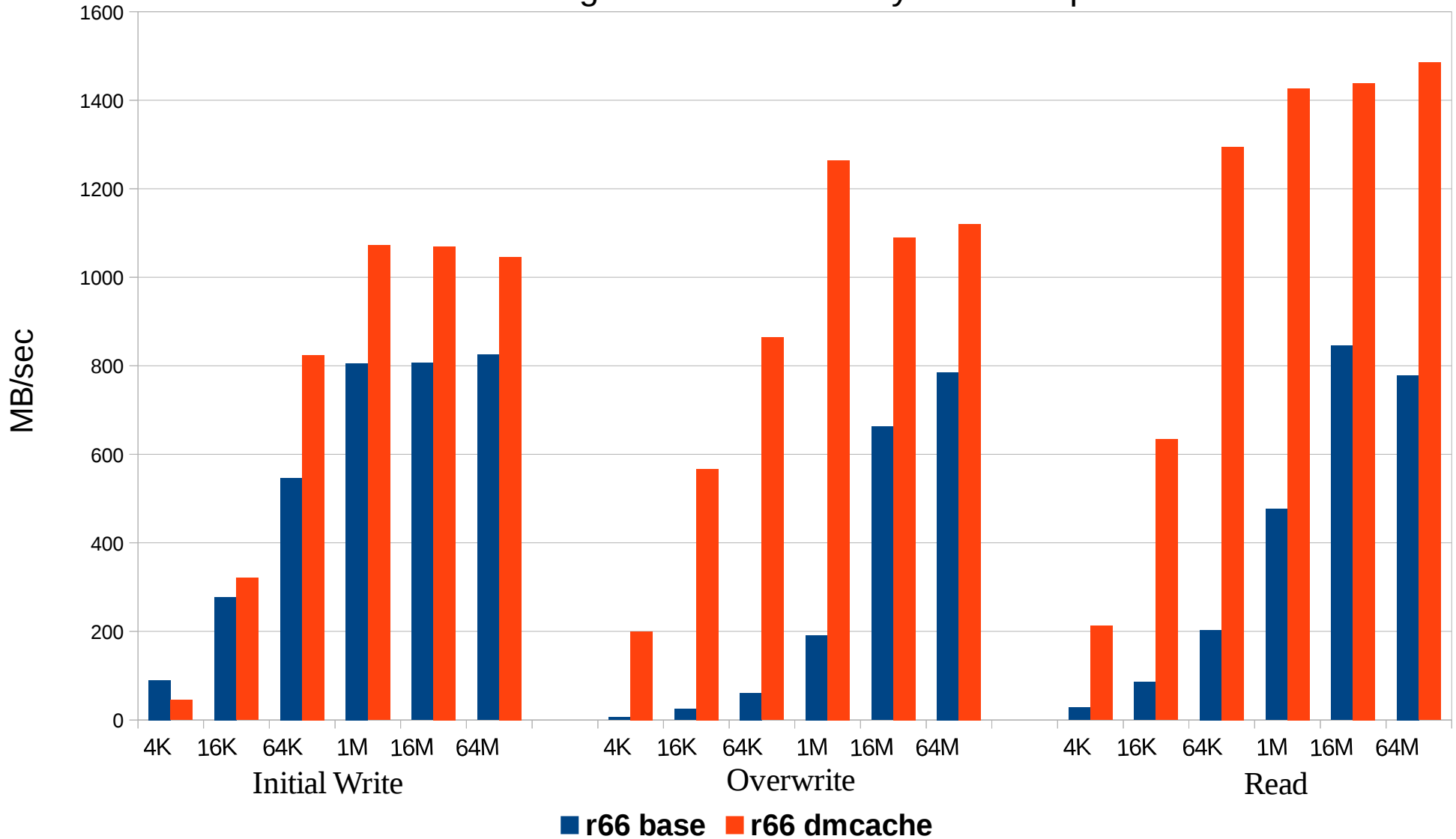
Result - performance improvement.

Supported as of RHEL 7.1 (TechPreview earlier)

docache

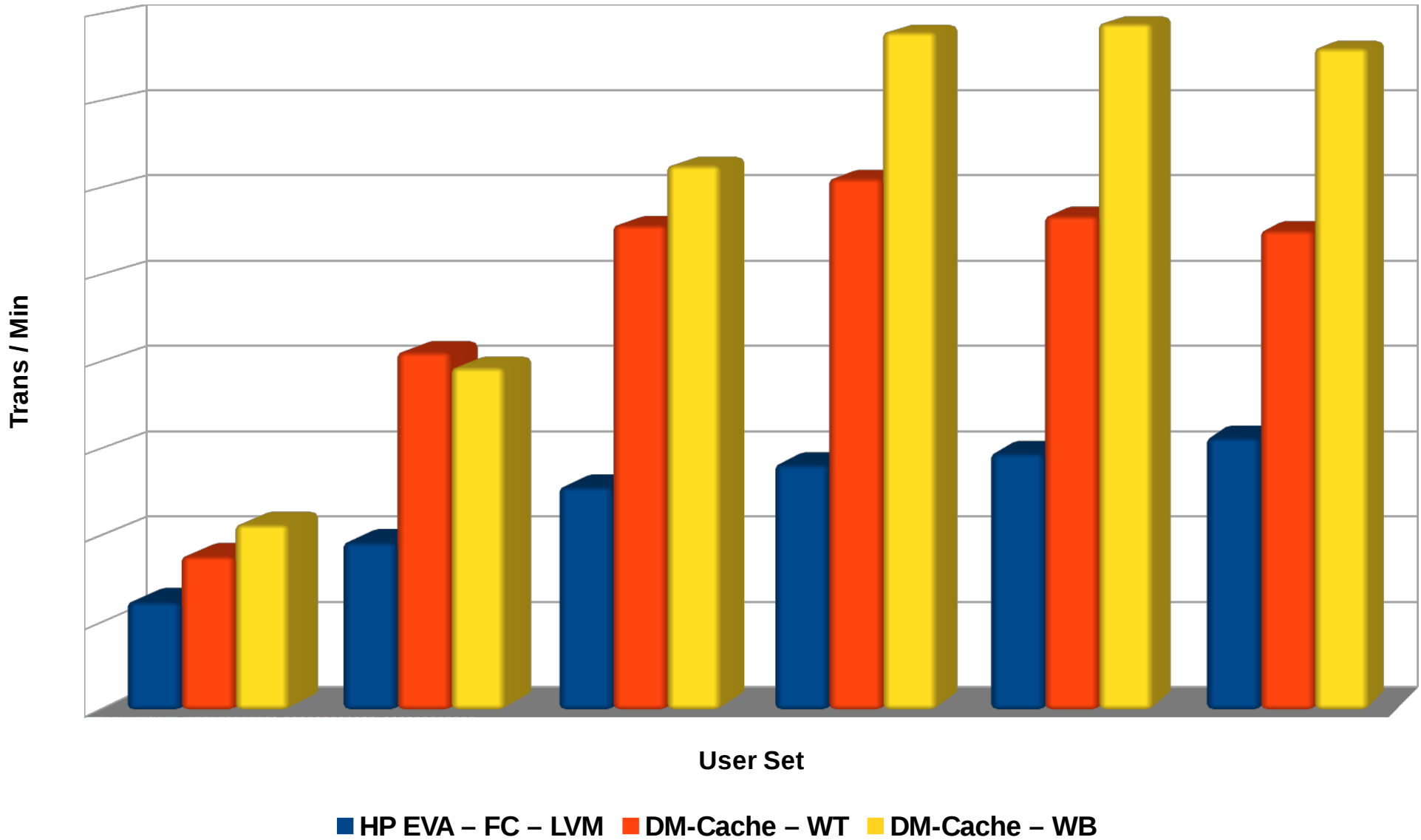
RHEL6.6 base vs. docache

Data file - Random I/O - Using SAP HANA's file system acceptance test.



Database OLTP workload

DM cache - 20G / Writethrough vs Writeback



Cgroups – RHEL 7 - Systemd

Resource Management

- Memory, cpus, IO, Network
- For performance
- For application consolidation
- Dynamic resource allocation
- **Application Isolation**
 - **Put less-critical or non-critical applications in controlled resource group to prevent them from hogging resources**
- **RHEL 7 Resource Management Guide**
 - https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/pdf/Resource_Management_Guide/Red_Hat_Enterprise_Linux-7-Resource_Management_Guide-en-US.pdf

Valuable Links



Low Latency Tuning Guide for Red Hat Enterprise Linux 7

<https://access.redhat.com/articles/1323793>



Accelerating Red Hat Enterprise Linux 7-based Linux Containers with Solarflare OpenOnload

<https://access.redhat.com/articles/1407003>



How do I create my own tuned profile on RHEL7 ?

<https://access.redhat.com/solutions/731473>

Valuable Links

- [Red Hat Performance Tuning Guide](#)
- [Red Hat Low Latency Tuning Guide](#)
- [Red Hat Virtualization Tuning Guide](#)
- [Resource Management and LXC Guide](#)
- [Comprehensive Overview of Storage Scalability in Docker](#)
- [RHEL Blog / Developer Blog](#)
- Blog: <http://www.breakage.org/> or [@jeremyeder](#)
- Reference Architectures on RH Portal
 - [Deploying Oracle RAC Database 12c on RHEL 7 - Best Practices](#)
- Key RH Summit Presentation:
 - [Performance analysis & tuning of Red Hat Enterprise Linux: Part I](#)
 - [Performance analysis & tuning of Red Hat Enterprise Linux: Part II](#)

Summary Takeaways

- System tuning
 - “tuned” is your friend. Get your tuning right.
 - Find the optimal profile, or create your own.
- Numa
 - Know your use of “numa” is correct
- RHEL 7 performance enhancements
- Pointers to key documentation.
 - Extensive performance briefs, best practices, white papers on RH website.

Questions ?



Backup

- Power management

Power Management: P/C-states

- P-state: CPU Frequency
 - Governors, Frequency scaling
- C-state: CPU Idle State
 - New Default Idle Driver in RHEL7: intel_pstate
 - Replaces acpi-cpufreq driver
 - CPU governors replaced with sysfs {min,max}_perf_pct
 - Moves Turbo knob into OS control (yay!)

Tuned handles
most of this for you

turbostat shows P/C-states on Intel CPUs

turbostat begins shipping in RHEL6.4, cpupowerutils package

tuned-adm profile throughput-performance

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

tuned-adm profile 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

Backup

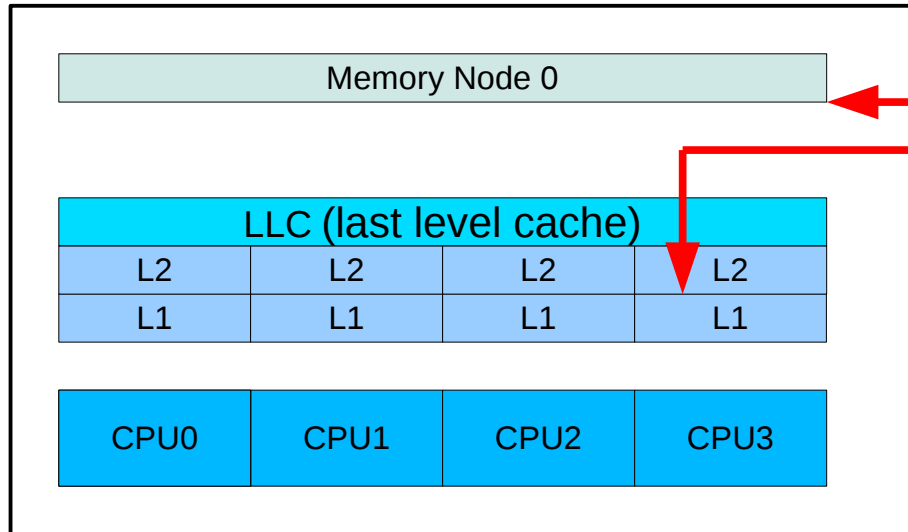
- Detecting “false sharing” cacheline tugging

Cache-to-Cache false sharing tool

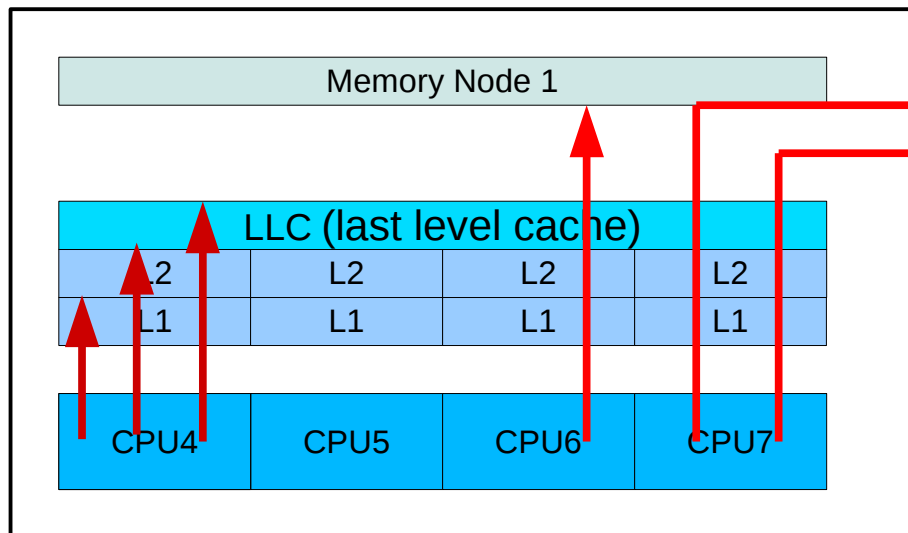
- Red Hat developed an enhancement to the 'perf' tool to detect simultaneous cross-numa cacheline contention.
- Critical to:
 - Shared memory applications
 - Multi-threaded apps spanning multiple numa nodes
- Integrating it into the perf tool (“perf c2c”)
- Should be available in future RHEL7.* release
- A perf binary available today to run on RHEL 7.1 or 7.2

Resolving a memory access

Socket 0

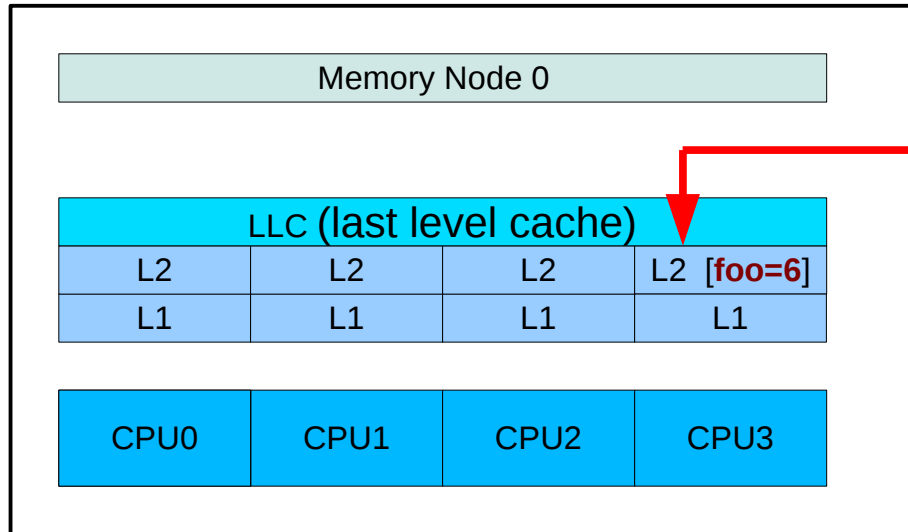


Socket 1

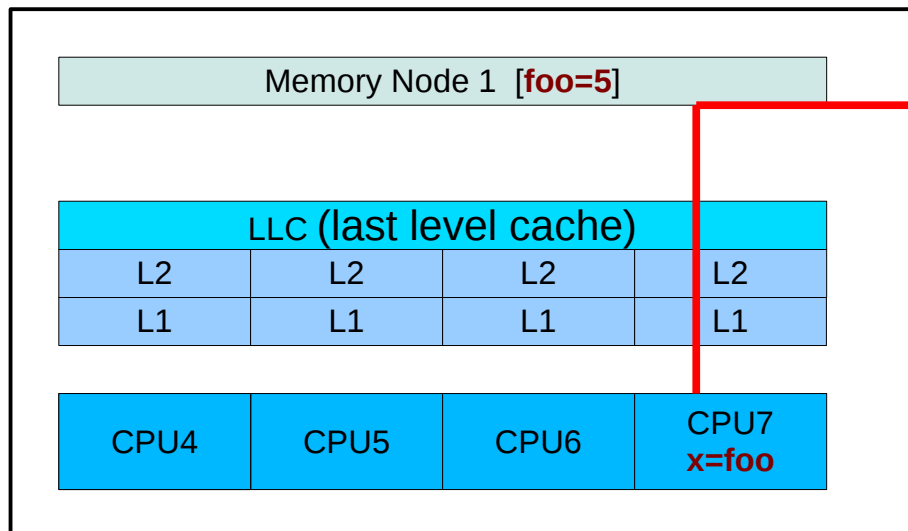


Resolving a memory access

Socket 0



Socket 1



Example data structure

```
struct person {  
    char name[24];  
    int age;  
    int salary;  
};
```

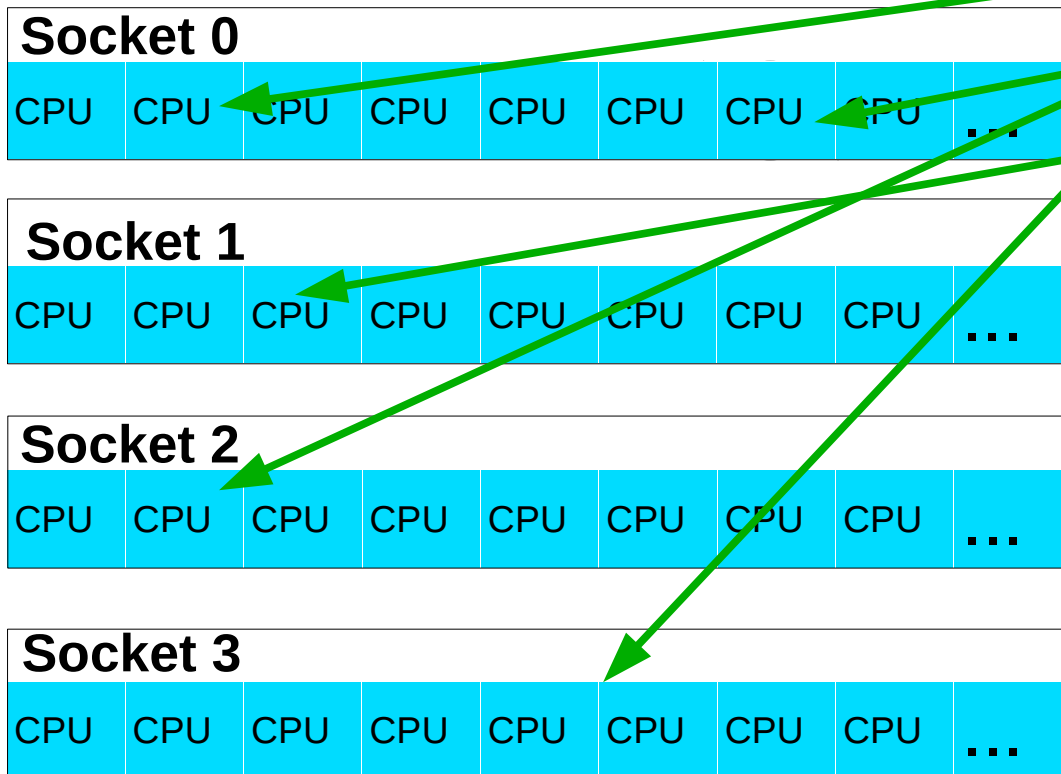
32-byte struct heavily read
from many processes across
all nodes

```
long incoming_cntr;
```

Heavily modified variable

Cacheline Contention - lots of happy readers

64 byte chunk of memory
(size of cacheline)



Offset 0
Offset 8
Offset 16
Offset 24
Offset 32
Offset 40
Offset 48
Offset 56

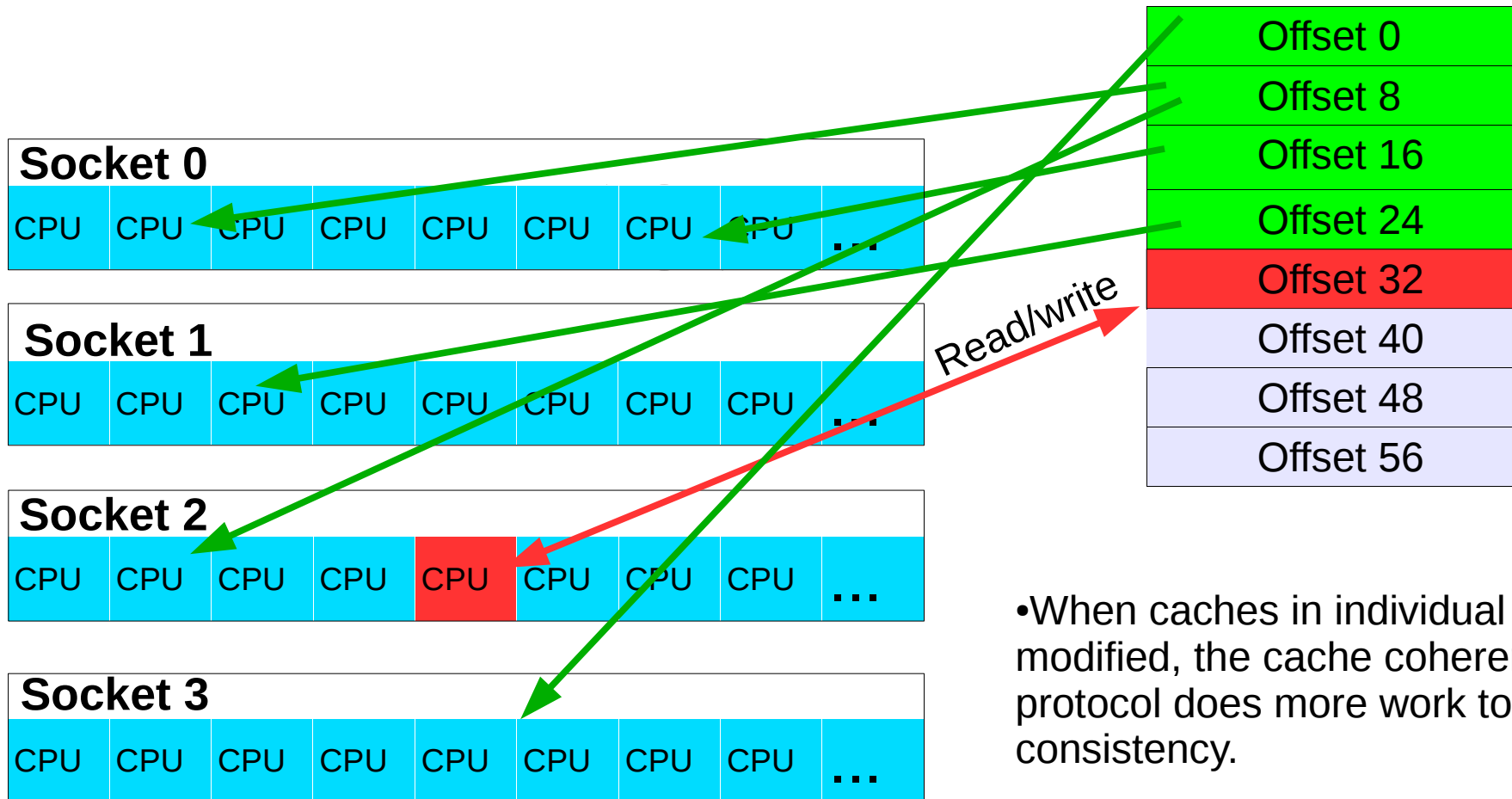
The 4 memory locations are only read.

The data remains in cpu caches for quick access.

Life is good.

Cacheline Contention - add in a hot writer

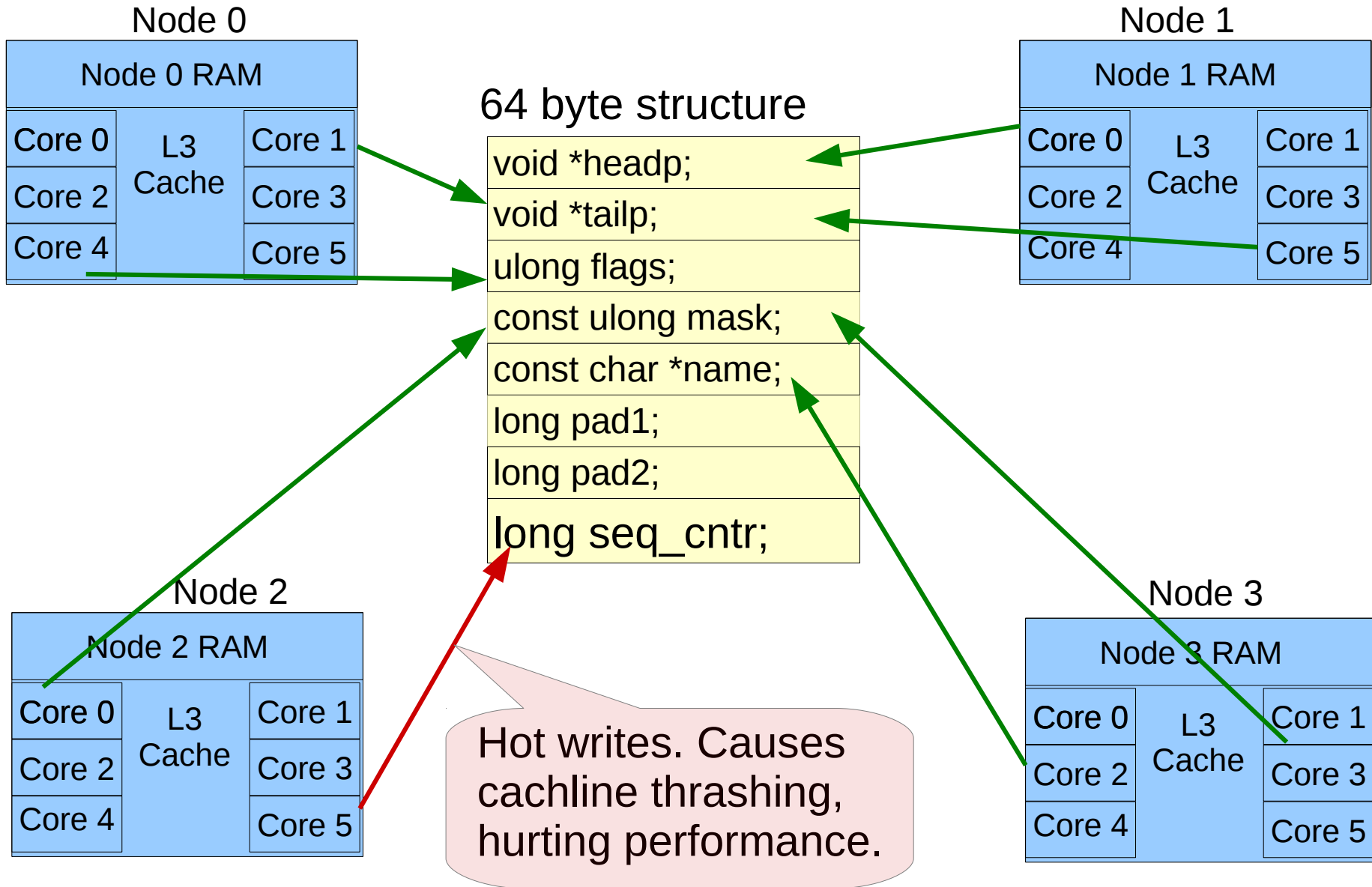
64 byte chunk of memory
(size of cacheline)



- When caches in individual cpus are modified, the cache coherency protocol does more work to maintain consistency.
- Can really hurt performance.

False cacheline sharing.

Add writer (in red) – slows down all readers.



How can you detect these?

The perf tool can find long latency loads

Are they long because the load is getting data from:

- local ram?
- a remote node's ram?
- a modified remote node's cache?
 - If so, who are the “writers” to that cache?

perf “c2c data sharing” tool

```
# perf c2c record sleep 10
```

```
# perf c2c -N report
```

Output from “c2c data sharing” tool (simplified to fit slide)

Cache #	Refs	Stores	Data Address	Pid	Tid	Inst Address	Symbol	Object	CPU Participants
0	118789	273709	0x6023 80	37878					
	17734	136078	0x6023 80	37878	37878	0x401520	read_wrt_thread	a.out	0{0};
	13452	137631	0x6023 88	37878	37883	0x4015a0	read_wrt_thread	a.out	0{1};
	15134	0	0x6023 a8	37878	37882	0x4011d7	reader_thread	a.out	1{5};
	14684	0	0x6023 b0	37878	37880	0x4011d7	reader_thread	a.out	1{6};
	13864	0	0x6023 b8	37878	37881	0x4011d7	reader_thread	a.out	1{7};
1	31	69	0xffff88023960df 40	37878					
	13	69	0xffff88023960df 70	37878	***	0xffffffff8109f8e5	update_cfs_rq_blocked	vmlinux	0{0,1,2}; 1{9,16};
	17	0	0xffff88023960df 60	37878	***	0xffffffff8109fc2e	__update_entity_load_avg	vmlinux	0{1,2}; 1{11,16};
	1	0	0xffff88023960df 78	37878	37882	0xffffffff8109fc4e	__update_entity_load_avg	vmlinux	0{2};

This shows us:

- The hottest contended cachelines
- The process names, data addr, ip, pids, tids
- The node and CPU numbers they ran on,
- And how the cacheline is being accessed (read or write)

Where is your program's memory coming from?

<u>Count</u>	<u>Response Type</u>	
152729	[LOAD,L2,HIT,SNP NONE]	// Loads resolved from the L2 cache
143821	[LOAD,LFB,HIT,SNP NONE]	// Loads resolved from the load fill buffer
116187	[LOAD,RMT_RAM,HIT,SNP NONE,SNP MISS]	// Loads resolved from a remote node's main memory
89248	[LOAD,L1,HIT,SNP NONE]	// Loads that got resolved from the L1 cache
40723	[LOAD,LCL_RAM,HIT,SNP NONE,SNP MISS]	// Loads resolved from local node's memory.
40614	[LOAD,LCL_LLC,HIT,SNP NONE]	// Loads resolved from local last level cache.
826	[STORE,L1,HIT]	// Stores that had L1 cacheline ownership
769	[STORE,L1,MISS]	// Needed to get ownership of cacheline
402	[LOAD,LCL_LLC,HIT,SNP MISS]	
279	[LOAD,LCL_LLC,MISS,SNP NA]	
185	[LOAD,RMT_LLC,HIT,SNP HIT]	
159	[LOAD,LCL_LLC,HIT,SNP HIT]	
134	[LOAD,RMT_LLC,HIT,SNP HITM]	<< False sharing across numa nodes
100	[LOAD,UNCACHED,HIT,SNP NONE]	
40	[LOAD,LCL_LLC,HIT,SNP HITM]	
27	[LOAD,L1,HIT,SNP NONE,LOCKED]	
4	[LOAD,RMT_RAM,HIT,SNP NONE,SNP MISS,LOCKED]	

NUMA backup slides

What is NUMA?

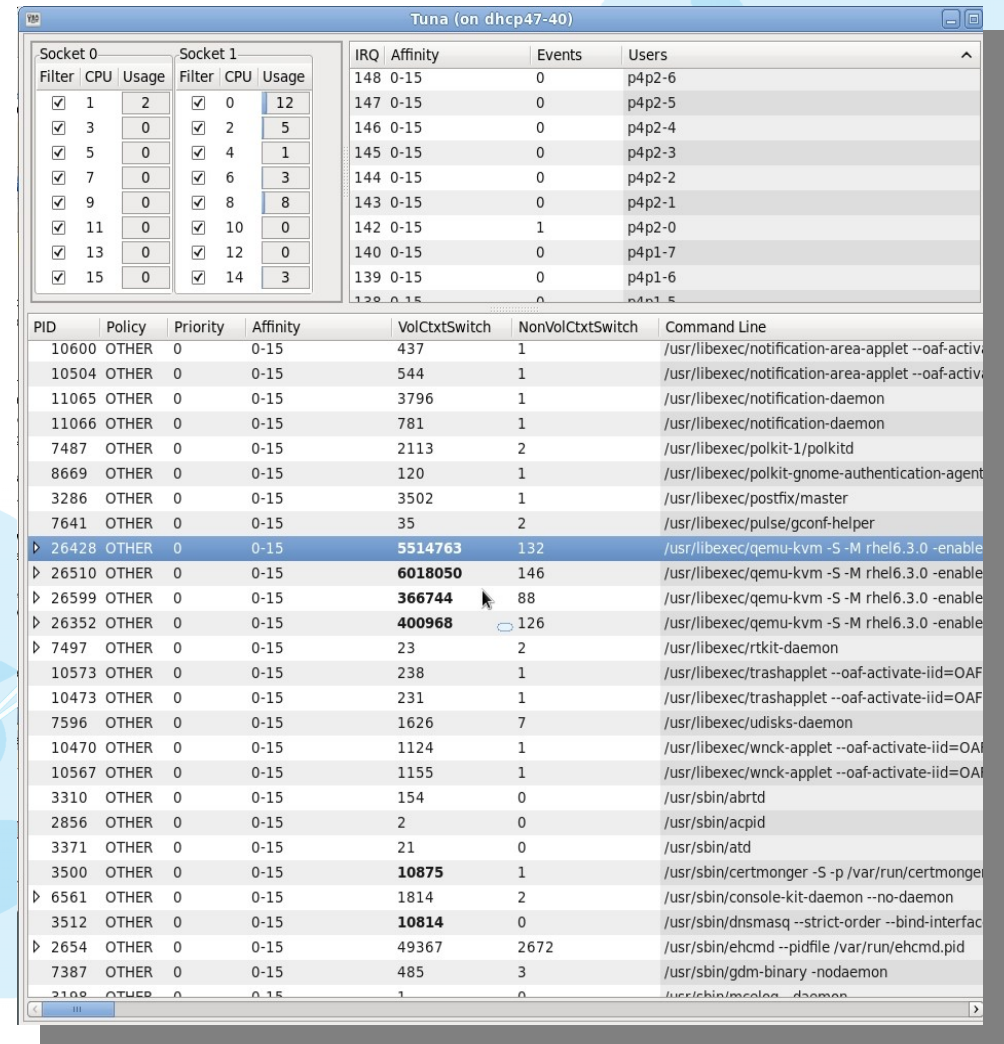
- Non Uniform Memory Access
- A result of making bigger systems more scalable by distributing system memory near individual CPUs....
- Practically all multi-socket systems have NUMA
 - Most servers have 1 NUMA node / socket
 - Recent AMD systems may have 2 NUMA nodes / socket



Tuna backup slides

System Tuning Tool - tuna

- Tool for fine grained control
- Display applications / processes
- Displays CPU enumeration
- Socket (useful for NUMA tuning)
- Dynamic control of tuning
 - Process affinity
 - Parent & threads
 - Scheduling policy
 - Device IRQ priorities, etc



The screenshot shows the Tuna application interface. At the top, it displays 'Tuna (on dhcp47-40)'. Below this, there are two tables for CPU usage on Socket 0 and Socket 1. The Socket 0 table shows CPU 1 with 2% usage, CPU 3 with 0%, CPU 5 with 0%, CPU 7 with 0%, CPU 9 with 0%, CPU 11 with 0%, CPU 13 with 0%, and CPU 15 with 0%. The Socket 1 table shows CPU 0 with 12% usage, CPU 2 with 5%, CPU 4 with 1%, CPU 6 with 3%, CPU 8 with 8%, CPU 10 with 0%, CPU 12 with 0%, and CPU 14 with 3%. To the right of these tables is a table with columns for IRQ, Affinity, Events, and Users. Below these are two main tables. The first table has columns for PID, Policy, Priority, Affinity, VolCtxtSwitch, NonVolCtxtSwitch, and Command Line. The second table is a list of processes with columns for PID, Policy, Priority, Affinity, VolCtxtSwitch, NonVolCtxtSwitch, and Command Line. The process with PID 26428 is highlighted in blue.

Socket 0	Socket 1	IRQ	Affinity	Events	Users
Filter CPU Usage	Filter CPU Usage	148	0-15	0	p4p2-6
<input checked="" type="checkbox"/> 1 2	<input checked="" type="checkbox"/> 0 12	147	0-15	0	p4p2-5
<input checked="" type="checkbox"/> 3 0	<input checked="" type="checkbox"/> 2 5	146	0-15	0	p4p2-4
<input checked="" type="checkbox"/> 5 0	<input checked="" type="checkbox"/> 4 1	145	0-15	0	p4p2-3
<input checked="" type="checkbox"/> 7 0	<input checked="" type="checkbox"/> 6 3	144	0-15	0	p4p2-2
<input checked="" type="checkbox"/> 9 0	<input checked="" type="checkbox"/> 8 8	143	0-15	0	p4p2-1
<input checked="" type="checkbox"/> 11 0	<input checked="" type="checkbox"/> 10 0	142	0-15	1	p4p2-0
<input checked="" type="checkbox"/> 13 0	<input checked="" type="checkbox"/> 12 0	140	0-15	0	p4p1-7
<input checked="" type="checkbox"/> 15 0	<input checked="" type="checkbox"/> 14 3	139	0-15	0	p4p1-6

PID	Policy	Priority	Affinity	VolCtxtSwitch	NonVolCtxtSwitch	Command Line
10600	OTHER	0	0-15	437	1	/usr/libexec/notification-area-applet --oaf-activ
10504	OTHER	0	0-15	544	1	/usr/libexec/notification-area-applet --oaf-activ
11065	OTHER	0	0-15	3796	1	/usr/libexec/notification-daemon
11066	OTHER	0	0-15	781	1	/usr/libexec/notification-daemon
7487	OTHER	0	0-15	2113	2	/usr/libexec/polkit-1/polkitd
8669	OTHER	0	0-15	120	1	/usr/libexec/polkit-gnome-authentication-agent
3286	OTHER	0	0-15	3502	1	/usr/libexec/postfix/master
7641	OTHER	0	0-15	35	2	/usr/libexec/pulse/gconf-helper
▶ 26428	OTHER	0	0-15	5514763	132	/usr/libexec/qemu-kvm -S -M rhel6.3.0 -enable
▶ 26510	OTHER	0	0-15	6018050	146	/usr/libexec/qemu-kvm -S -M rhel6.3.0 -enable
▶ 26599	OTHER	0	0-15	366744	88	/usr/libexec/qemu-kvm -S -M rhel6.3.0 -enable
▶ 26352	OTHER	0	0-15	400968	126	/usr/libexec/qemu-kvm -S -M rhel6.3.0 -enable
▶ 7497	OTHER	0	0-15	23	2	/usr/libexec/rtkit-daemon
10573	OTHER	0	0-15	238	1	/usr/libexec/trashapplet --oaf-activate-iid=OAF
10473	OTHER	0	0-15	231	1	/usr/libexec/trashapplet --oaf-activate-iid=OAF
7596	OTHER	0	0-15	1626	7	/usr/libexec/udisks-daemon
10470	OTHER	0	0-15	1124	1	/usr/libexec/wnck-applet --oaf-activate-iid=OAF
10567	OTHER	0	0-15	1155	1	/usr/libexec/wnck-applet --oaf-activate-iid=OAF
3310	OTHER	0	0-15	154	0	/usr/sbin/abrt
2856	OTHER	0	0-15	2	0	/usr/sbin/acpid
3371	OTHER	0	0-15	21	0	/usr/sbin/atd
3500	OTHER	0	0-15	10875	1	/usr/sbin/certmonger -S -p /var/run/certmonger
▶ 6561	OTHER	0	0-15	1814	2	/usr/sbin/console-kit-daemon --no-daemon
3512	OTHER	0	0-15	10814	0	/usr/sbin/dnsmasq --strict-order --bind-interfac
▶ 2654	OTHER	0	0-15	49367	2672	/usr/sbin/ehcmd --pidfile /var/run/ehcmd.pid
7387	OTHER	0	0-15	485	3	/usr/sbin/gdm-binary -nodaemon
3108	OTHER	0	0-15	1	0	/usr/sbin/mcclm-daemon

Tuna (RHEL6/7)

1

Socket 0			Socket 1			IRQ	Affinity	Events	Users
Filter	CPU	Usage	Filter	CPU	Usage				
<input checked="" type="checkbox"/>	0	29	<input checked="" type="checkbox"/>	1	0	0	0-23	12994	timer
<input checked="" type="checkbox"/>	2	6	<input checked="" type="checkbox"/>	3	0	1	0,2,4,6,8,10	2	i8042
<input checked="" type="checkbox"/>	4	19	<input checked="" type="checkbox"/>	5	0	3	0,2,4,6,8,10	268	serial
<input checked="" type="checkbox"/>	6	0	<input checked="" type="checkbox"/>	7	0	4	0,2,4,6,8,10	1	
<input checked="" type="checkbox"/>	8	0	<input checked="" type="checkbox"/>	9	0	8	0,2,4,6,8,10	1	rtc0
<input checked="" type="checkbox"/>	10	0	<input checked="" type="checkbox"/>	11	0	9	0,2,4,6,8,10	0	acpi
<input checked="" type="checkbox"/>	12	0	<input checked="" type="checkbox"/>	13	0	12	0,2,4,6,8,10	4	i8042
<input checked="" type="checkbox"/>	14	7	<input checked="" type="checkbox"/>	15	0	14	6	0	pata_atiixp
<input checked="" type="checkbox"/>	16	0	<input checked="" type="checkbox"/>	17	0	15	0,2,4,6,8,10	0	pata_atiixp
<input checked="" type="checkbox"/>	18	0	<input checked="" type="checkbox"/>	19	0	16	20	0	radeon,ahci
<input checked="" type="checkbox"/>	20	0	<input checked="" type="checkbox"/>	21	0	22	2	0	ehci_hcd:usb2,ohci_hcd:usb3,ohci_hcd:usb4
<input checked="" type="checkbox"/>	22	0	<input checked="" type="checkbox"/>	23	0	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

2

3

PID	Policy	Priority	Affinity	VolCtxSwitch	NonVolCtxSwitch	Command Line
1	OTHER	0	0-23	1452	55	/sbin/init
383	OTHER	0	0-23	1	0	/sbin/udev -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,+skin
911	OTHER	0	0-23	668	91	/sbin/udev -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	127710	2	hdparm

Tuna GUI Capabilities Updated for RHEL7

The screenshot displays the Tuna GUI interface for RHEL7, showing configuration options for various system parameters. The interface is divided into several sections:

- Monitoring** | **Profile management** | Profile editing
- Current active tuna profile: `example.conf`
- Buttons: Save Snapshot, Save & Apply permanently, Restore changes, Apply changes
- Kernel scheduler**
 - `kernel.core_pattern`: `core`
 - `kernel.sched_latency_ns`: 24000000
 - `kernel.sched_min_granularity_ns`: 10000000
 - `kernel.sched_nr_migrate`: 32
 - `kernel.sched_rt_period_us`: 1000000
 - `kernel.sched_rt_runtime_us`: 950000
 - `kernel.sched_tunable_scaling`: 1
 - `kernel.sched_wakeup_granularity_ns`: 4000000
- Network IPv4**
 - `ipv4.conf.all.forwarding`: 1
 - `ipv4.conf.all.rp_filter`: 0
 - `ipv4.tcp_congestion_control`: `cubic`
- Network IPv6**
 - `ipv6.conf.all.forwarding`
 - `ipv6.conf.default.forwarding`
 - `ipv6.conf.docker0.forwarding`
 - `ipv6.conf.em1.forwarding`
 - `ipv6.conf.em2.forwarding`
- VM**
 - `vm.dirty_expire_centiseccs`
 - `vm.dirty_ratio`
 - `vm.dirty_writeback_centiseccs`
 - `vm.laptop_mode`
 - `vm.memory_failure_early_kill`
 - `vm.swappiness`



Backup slides for RHEL 7 perf features

Latency Performance – System setup

- Evaluate the 2 new tuned profiles for networking
- Disable unnecessary services, runlevel 3
 - Follow vendor guidelines for BIOS Tuning
 - Logical cores? Power Management? Turbo?
- In the OS, consider
 - Disabling filesystem journal
 - SSD/Memory Storage
 - Reducing writeback thresholds if your app does disk I/O
 - Tune writeback workqueue affinity
 - NIC Offloads favor throughput

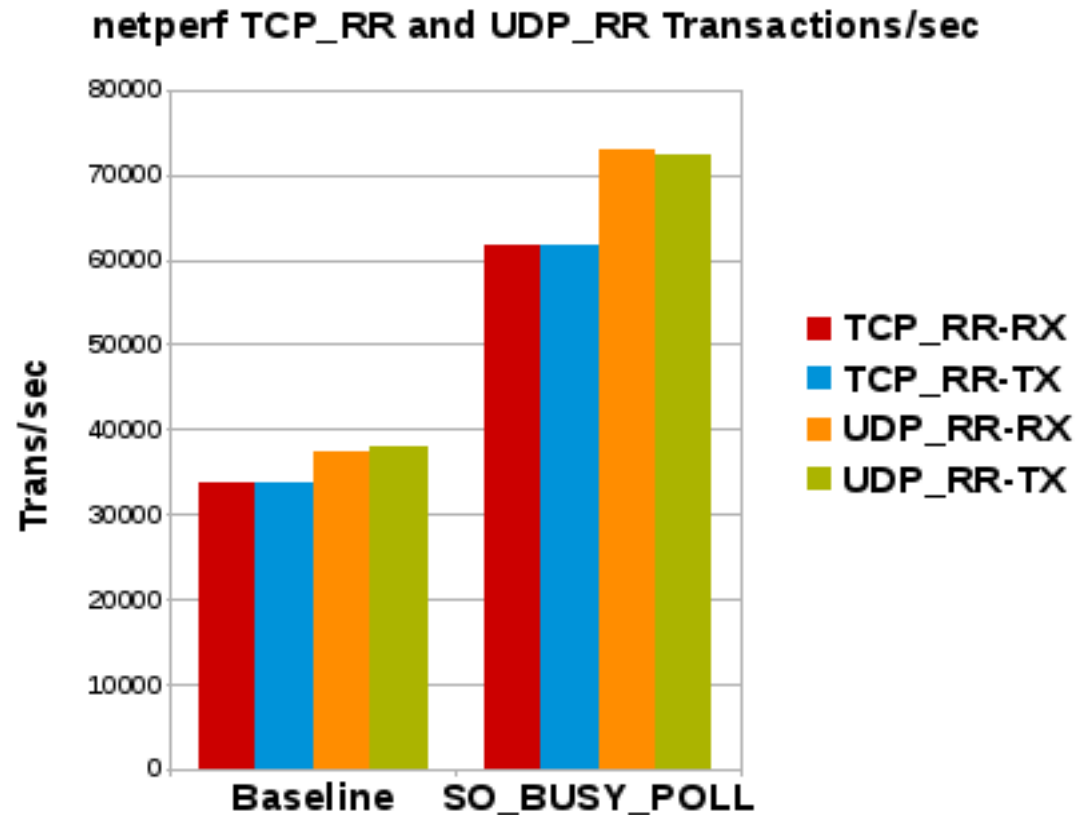
BUSY_POLL Socket Option

- Available in RHEL 7
- Socket-layer code polls receive queue of NIC
- Significant performance improvement
- Avoids interrupts and resulting context switching.
- and NAPI interrupt mitigation
- Retains full capabilities of kernel network stack

Higher is better

RHEL7 BUSY_POLL Socket Option

- Socket-layer code polls receive queue of NIC
- Replaces interrupts and NAPI
- Retains full capabilities of kernel network stack



Higher is better

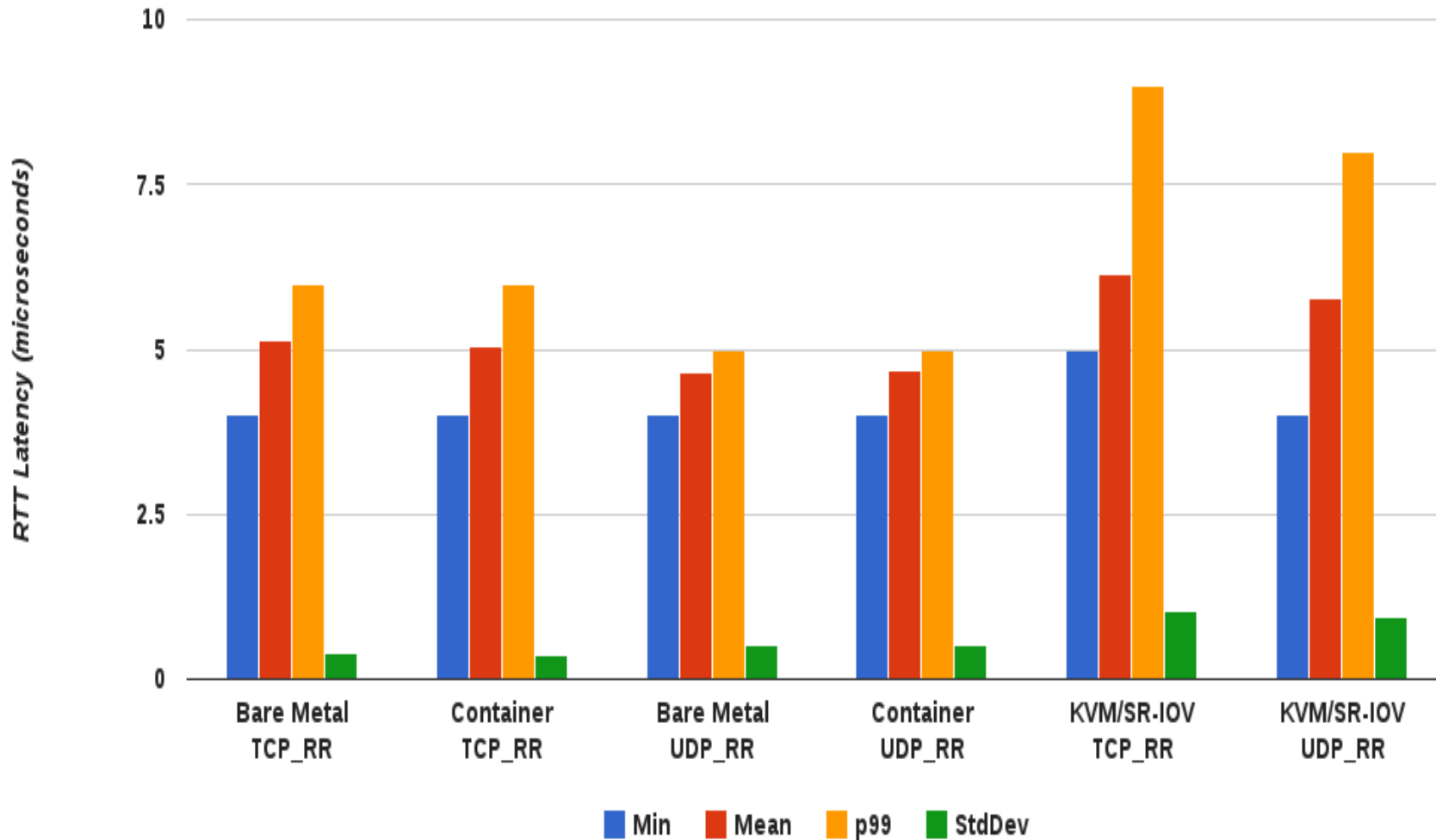
Accelerating Red Hat Enterprise Linux 7-based Linux Containers with Solarflare OpenOnload

- Config and tuning guidance for OpenOnload-accelerated Linux containers
- Emphasizes impactful new features included in RHEL7:
 - Docker
 - Atomic
 - Super-privileged Containers
 - Low Latency Tuning
 - tuned profiles

<https://access.redhat.com/articles/1323793>

RHEL 7.1 + OpenOnload

Bare Metal/Containers/SR-IOV

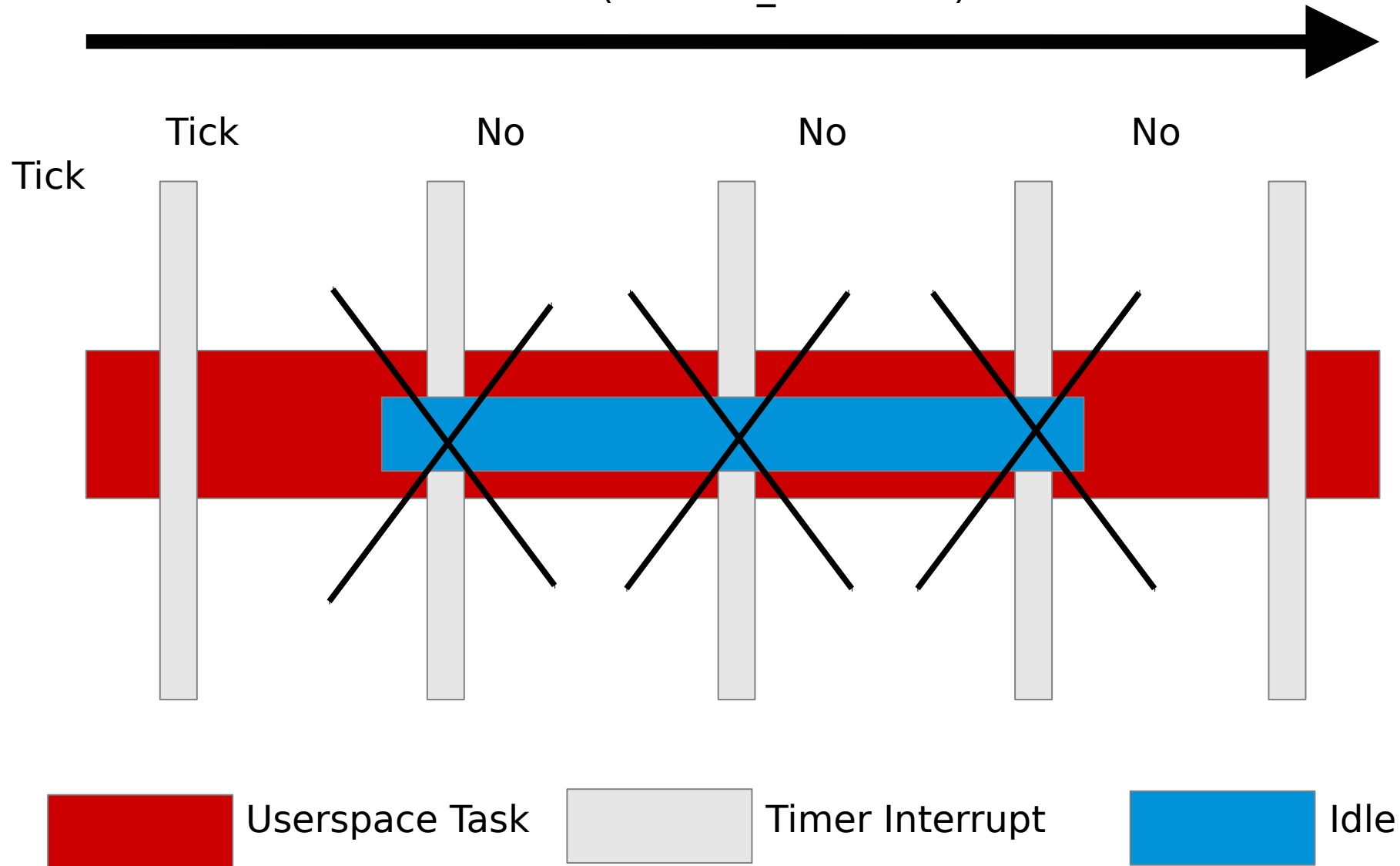


Full DynTicks Patchset

- Patchset Goal:
 - Stop interrupting userspace tasks
 - Move timekeeping to non-latency-sensitive cores
- If `nr_running=1`, then scheduler/tick can avoid that core
- Default disabled...Opt-in via `nohz_full` cmdline option
 - Kernel Tick:
 - timekeeping (`gettimeofday`)
 - Scheduler load balancing
 - Memory statistics (`vmstat`)

RHEL6 and 7 Tickless

Time (CONFIG_HZ=1000)

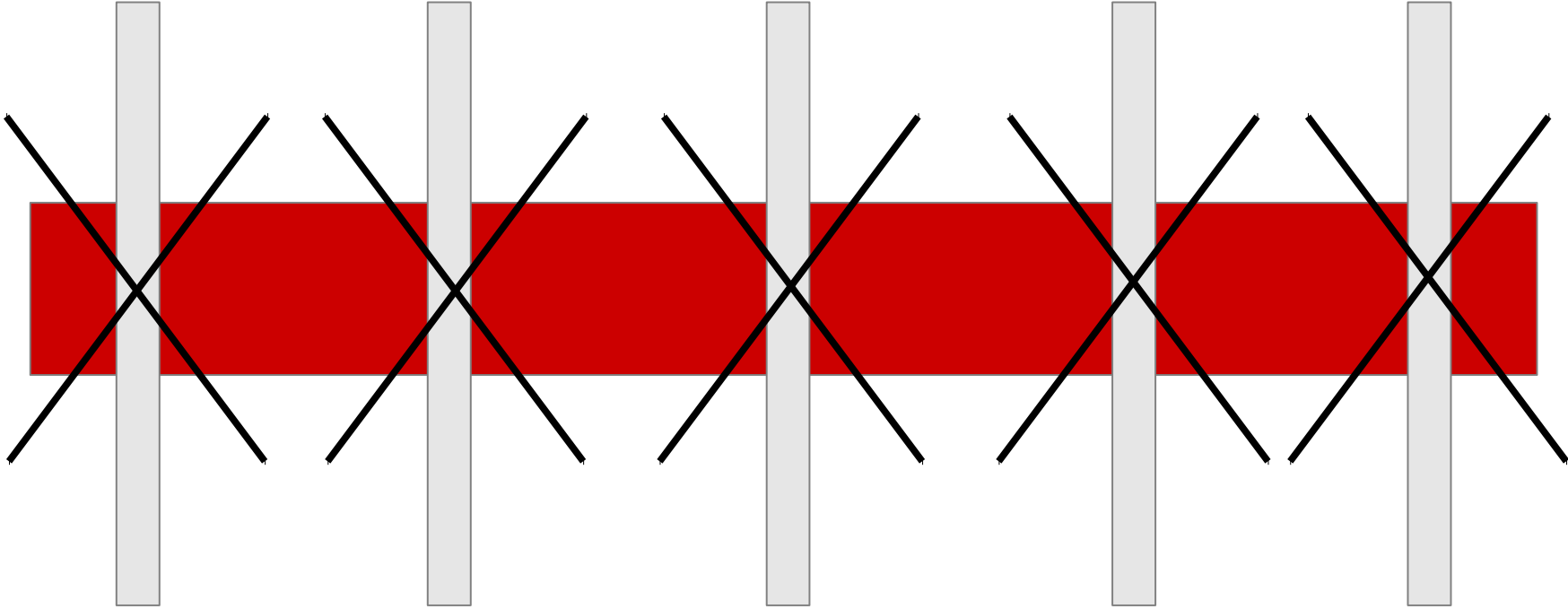


RHEL7 nohz_full

Time (CONFIG_HZ=1000)



No Ticks



Userspace Task



Timer Interrupt

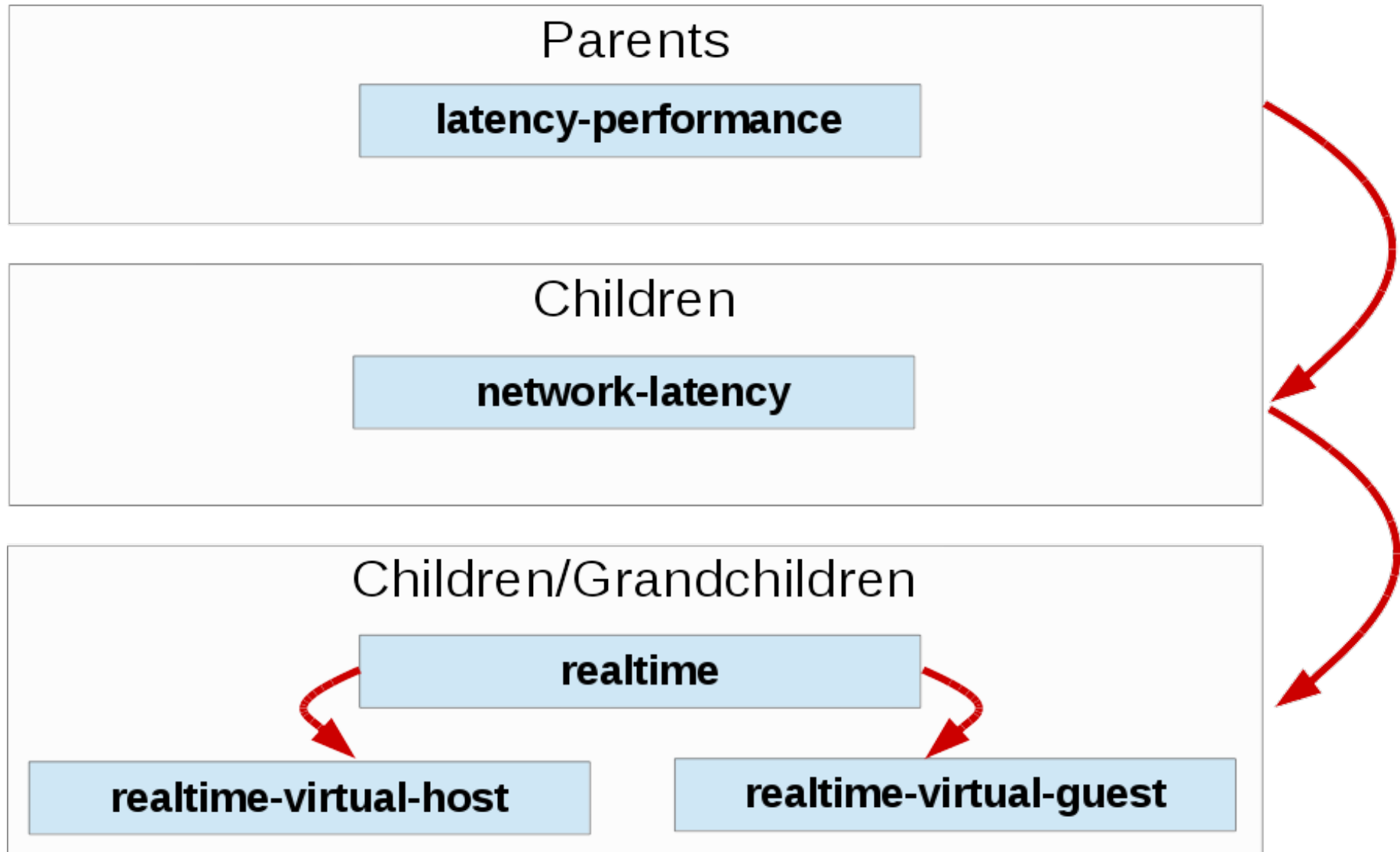


Idle

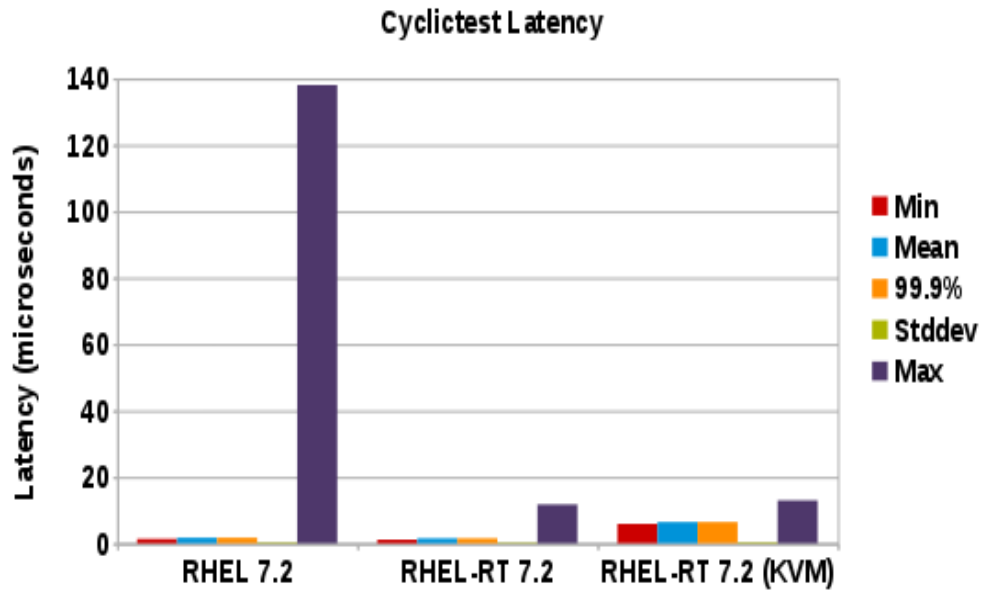
Backup

- NFV/Realtime Performance

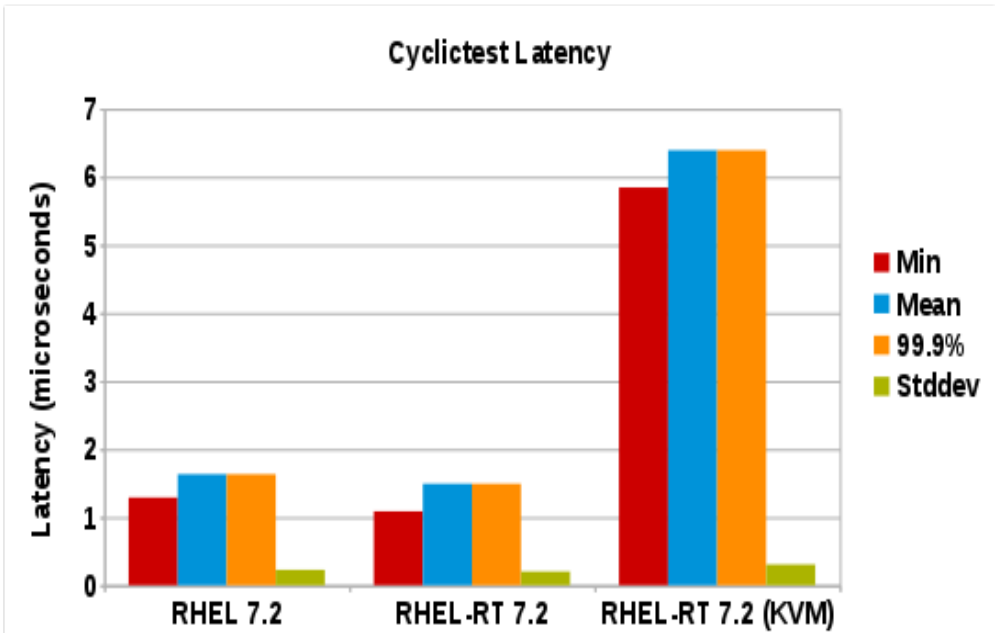
RT, KVM-RT/NFV Tuned Profiles



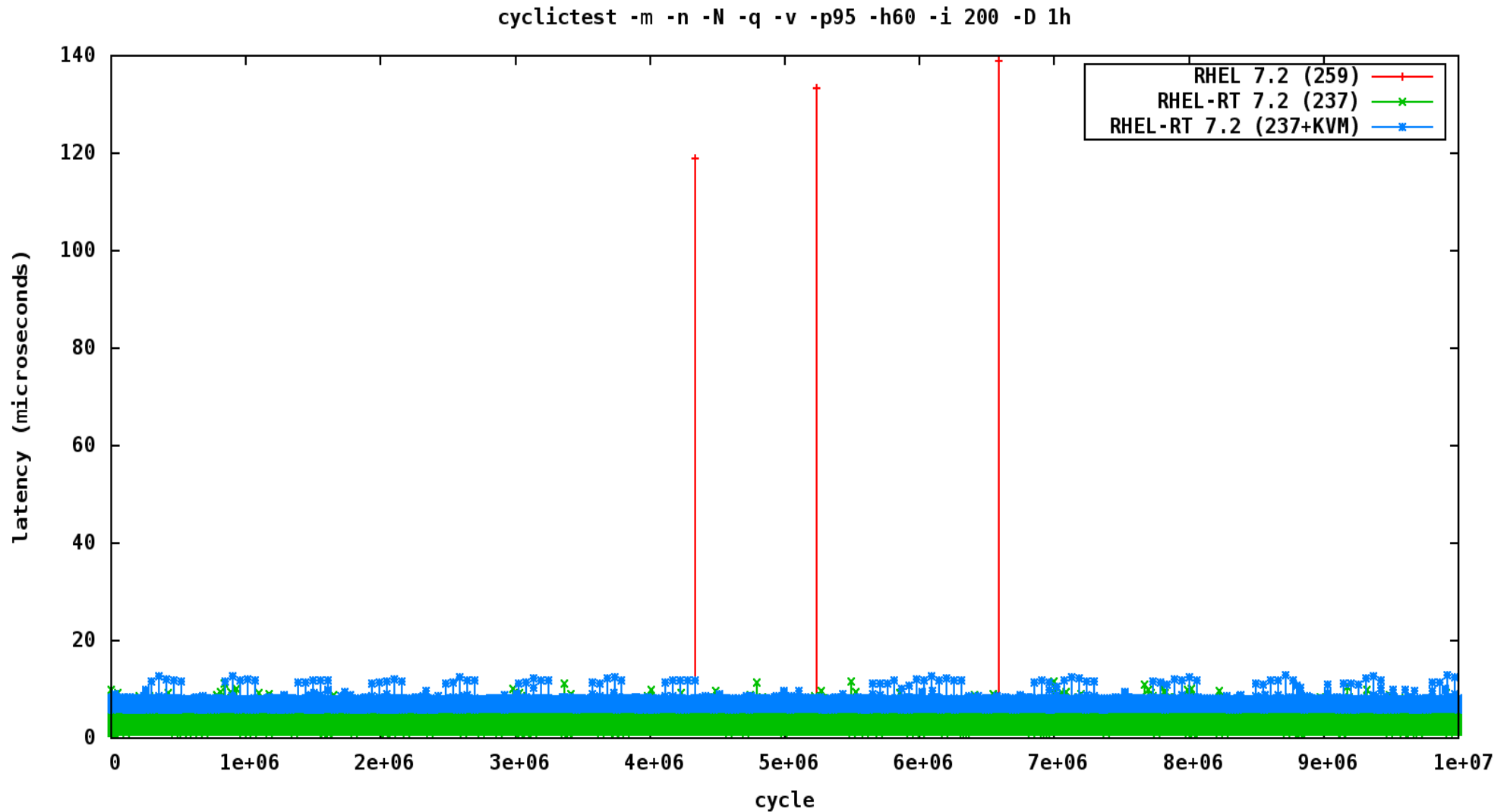
Scheduler Latency (cyclicttest)



Remove maxes to zoom in



Realtime Scheduler Latency Jitter Plot



RHEL 7.x Network Performance

Intel Haswell EP, 12-40Gb ports (6 cards)

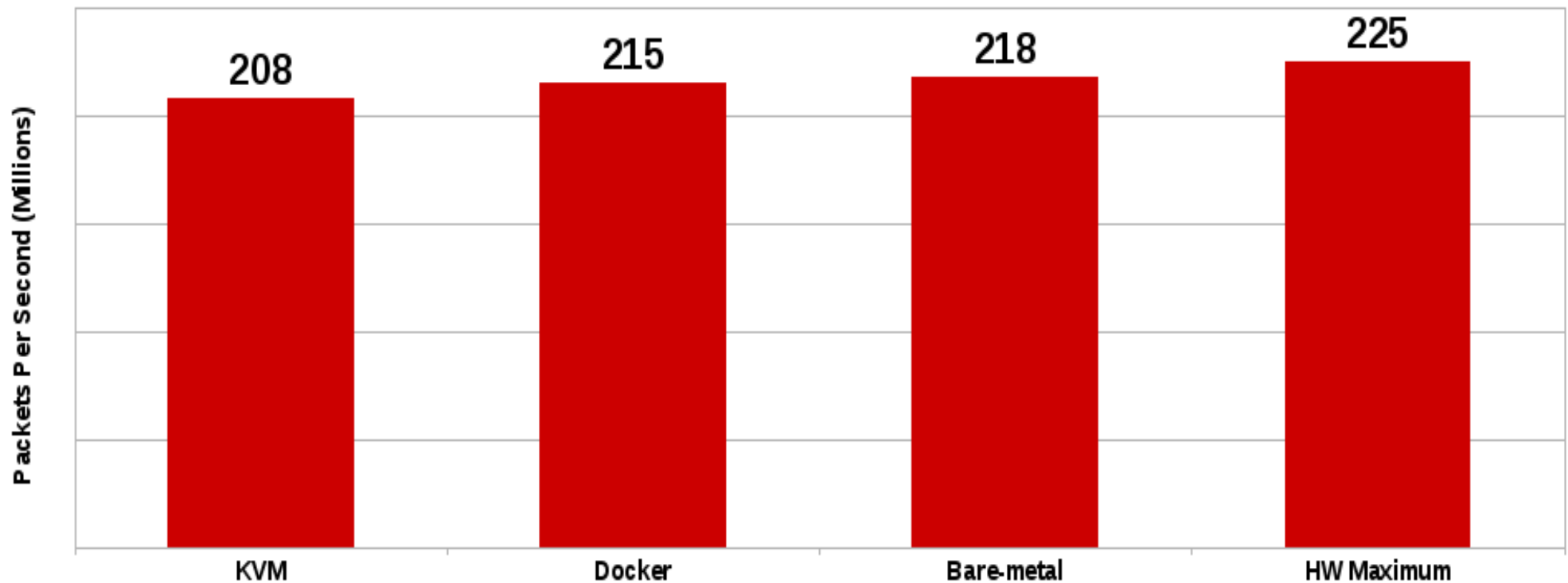


NFV 40G Packets/Sec DPDK (64 byte UDP)

208Mpps+
INTO KVM
DPDK

NFV: Millions of Packets Per Second

RHEL 7.x, L2 Forwarding, 12 x 40Gb NICs



40G Network Data/Tuned Networks

421 Gbps

