



Red Hat Customer Convergence

#rhconvergence



Performance Tuning and Analysis of Red Hat Enterprise Linux 7

Joe Mario
Senior Principal Software Engineer
Feb 2015

Agenda

- Red Hat Performance Team
- RHEL7.1, RHEL6.6 items
- Tuned
- NUMA Tuning
- perf
- Linux Containers

Red Hat Performance Engineering

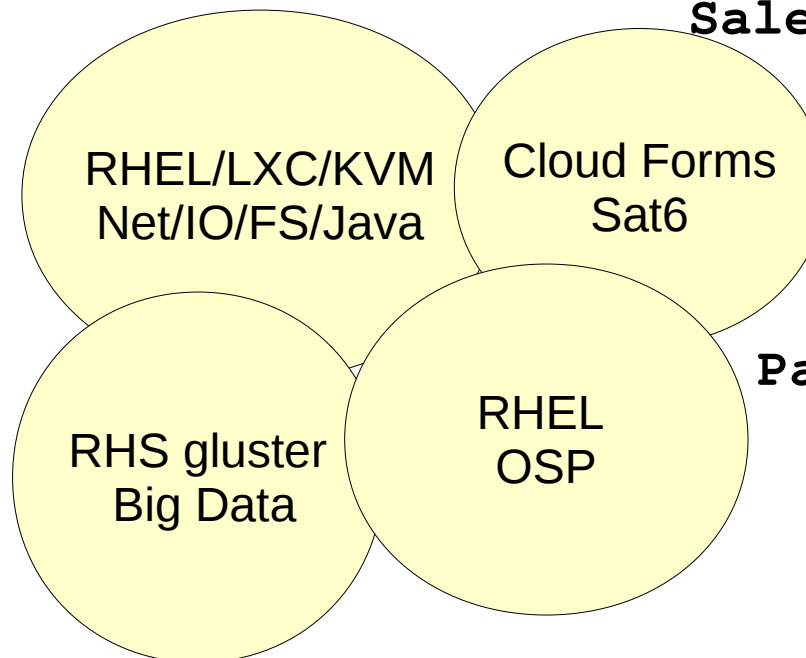
Benchmarking SPEC/TPC/STAC

Field Support

ISV / Applications

Sales / Field Consulting

New Technology
cpu/nic/ssd



Partner / HW OEM / Chips

Competitive Analysis

Reference Architectures
Perf Briefs / Guides

RHEL Platform(s) Performance Coverage

Benchmarks – code path coverage

- 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

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
- STAC = FSI (STAC-N)

RHEL Performance Evolution

Red Hat Enterprise Linux Performance Evolution

RHEL5

Static Hugepages

Ktune – on/off

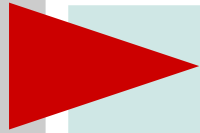
CPU Affinity
(taskset)

NUMA Pinning
(numactl)

irqbalance

Red Hat Enterprise Linux Performance Evolution

RHEL5

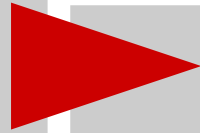


RHEL6

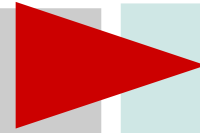
Static Hugepages	Transparent Hugepages
Ktune – on/off	Tuned – choose profile
CPU Affinity (taskset)	CPU Affinity (ts/numactl)
NUMA Pinning (numactl)	NUMAD – userspace tool
irqbalance	irqbalance – NUMA enhanced

Red Hat Enterprise Linux Performance Evolution

RHEL5



RHEL6



RHEL7

Static Hugepages
Ktune – on/off

CPU Affinity
(taskset)

NUMA Pinning
(numactl)

irqbalance

Transparent
Hugepages

Tuned – choose
profile

CPU Affinity
(ts/numactl)

NUMAD – userspace
tool

irqbalance – NUMA
enhanced

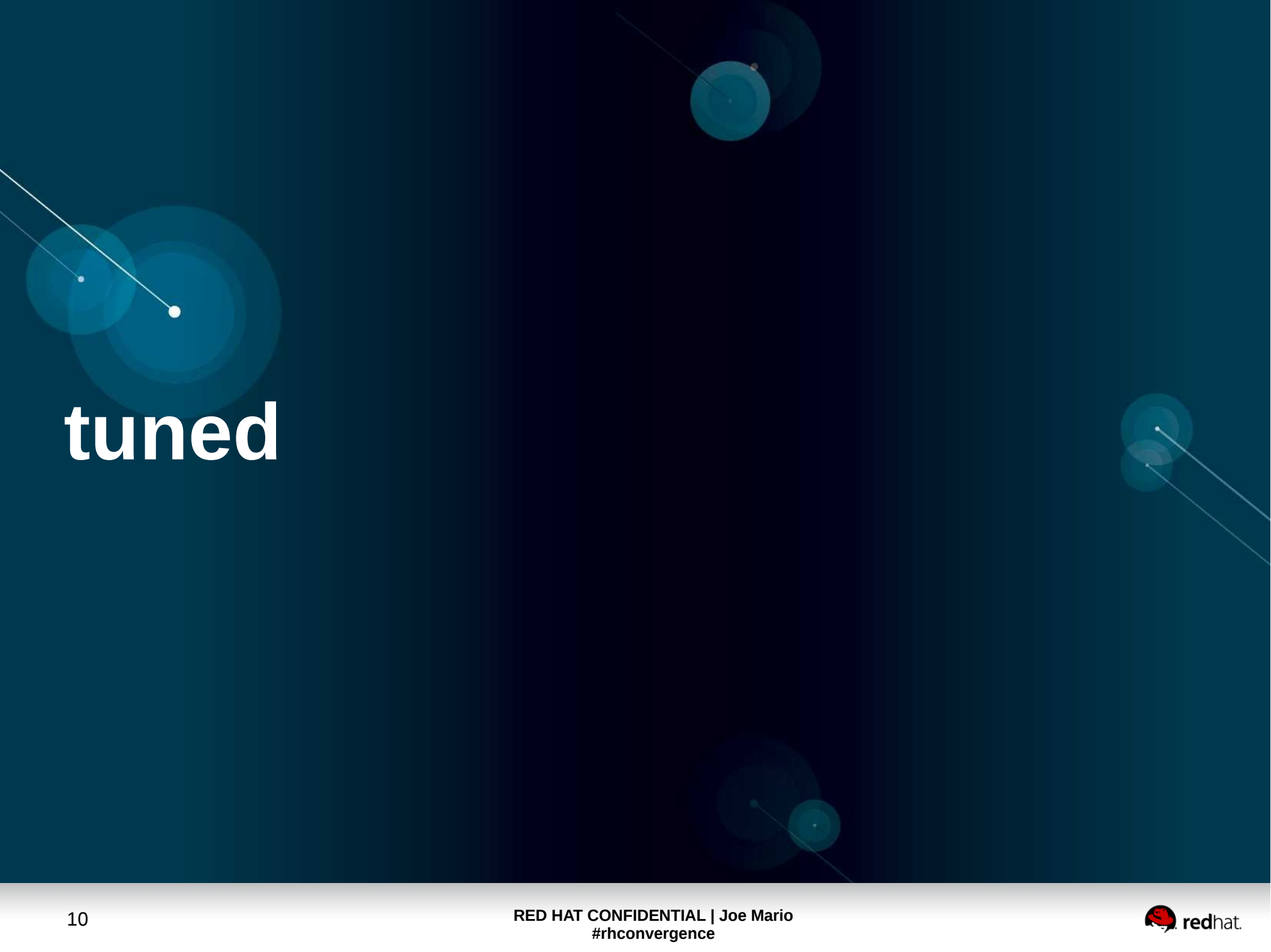
Transparent
Hugepages

Tuned – throughput-
performance (default)

CPU Affinity
(ts/numactl)

Autonuma-Balance

irqbalance – NUMA
enhanced



tuned

What is “tuned” ?

Tuning profile delivery mechanism

Red Hat ships *tuned profiles* that improve performance for many workloads...hopefully yours!

Tuned: Updates for RHEL7

- Re-written for maintainability and extensibility.
 - Configuration is now consolidated a single `tuned.conf` file
 - Optional hook/callout capability
 - Adds concept of Inheritance (just like `httpd.conf`)
 - Profiles updated for RHEL7 features and characteristics

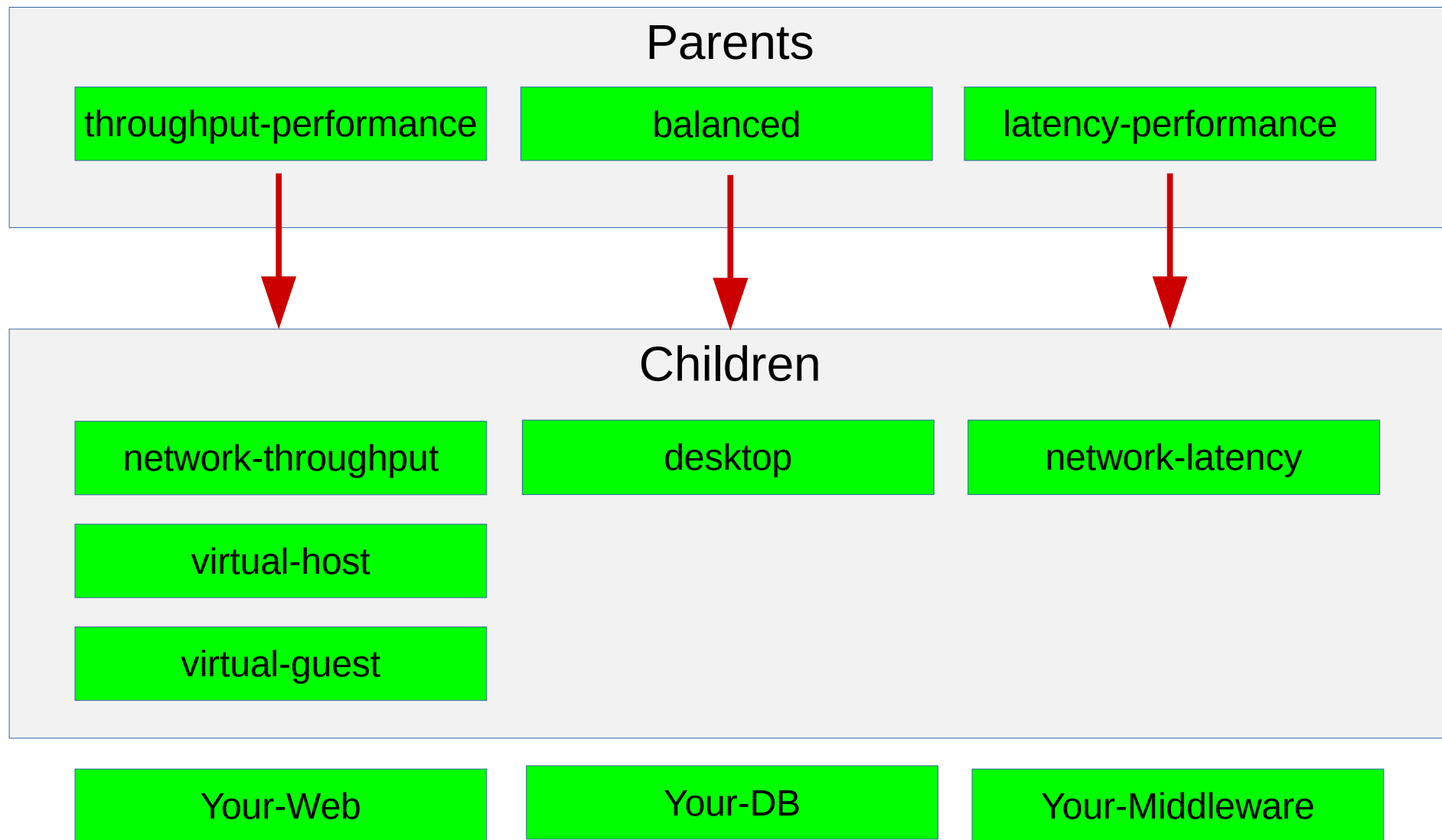
RHEL “tuned” package

Available profiles:

- balanced
- desktop
- latency-performance
- **myprofile** << Easy to add your own
- network-latency
- network-throughput
- throughput-performance
- virtual-guest
- virtual-host

Current active profile: **myprofile**

Tuned: Profile Inheritance



Tuned: RHEL Atomic Profile Inheritance

throughput-performance

governor=performance
energy_perf_bias=performance
min_perf_pct=100
transparent_hugepages=always
readahead=4096
sched_min_granularity_ns = 10000000
sched_wakeup_granularity_ns = 15000000
vm.dirty_ratio = 40
vm.dirty_background_ratio = 10
vm.swappiness=10

virtual-guest

vm.dirty_ratio = 30
vm.swappiness = 30

atomic-host

avc_cache_threshold=65536
nf_conntrack_hashsize=131072
kernel.pid_max=131072
net.netfilter.nf_conntrack_max=1048576

atomic-guest

avc_cache_threshold=65536
nf_conntrack_hashsize=131072
kernel.pid_max=131072
net.netfilter.nf_conntrack_max=1048576

NoFile=1048576

Tuned Profiles throughout Red Hat's Product Line

RHEL7 Desktop/Workstation

balanced

RHEL7 Server/HPC

throughput-performance

RHEL6/7 KVM Host, Guest

virtual-host/guest

RHEV

virtual-host

Red Hat Storage

rhs-high-throughput, virt

RHEL OSP (compute node)

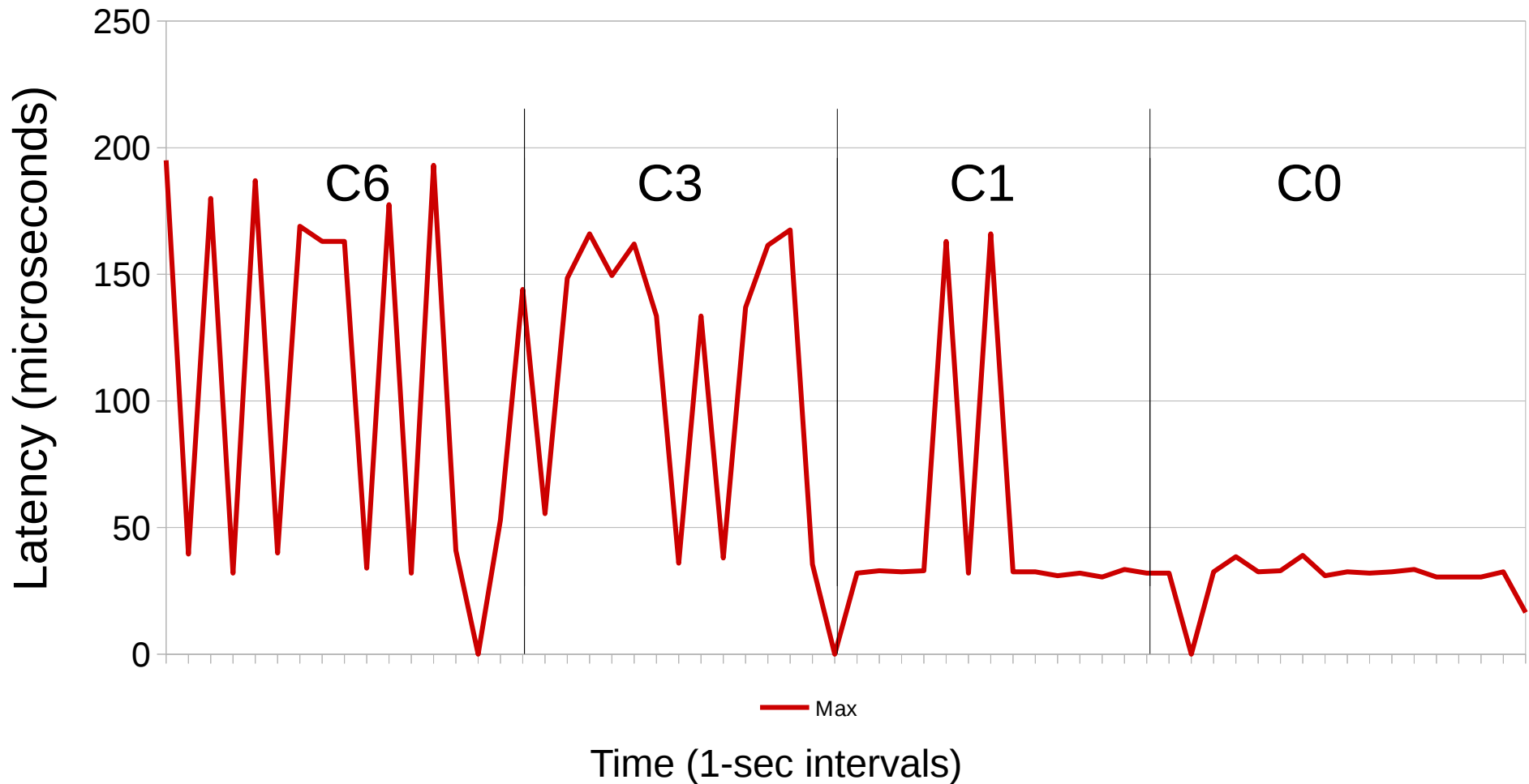
virtual-host

RHEL Atomic

atomic-host/guest

Tuned: Network Latency Performance Boost

C-state lock improves determinism, reduces jitter



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

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

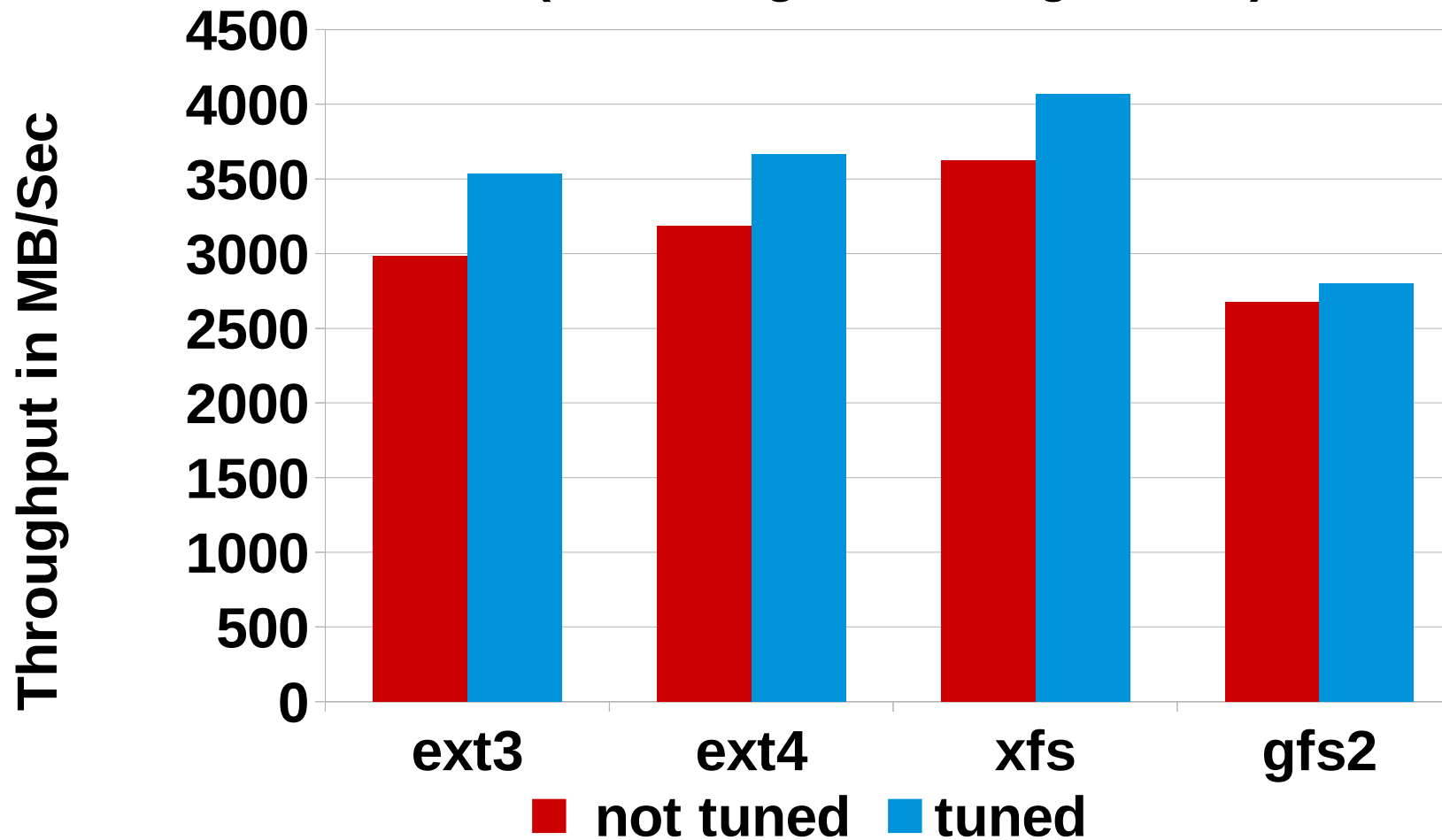
pk	cor	CPU	%c0	GHz	TSC	%c1	%c3	%c6	%c7
0	0	0	0.00	3.30	2.90	100.00	0.00	0.00	0.00
0	1	1	0.00	3.30	2.90	100.00	0.00	0.00	0.00
0	2	2	0.00	3.30	2.90	100.00	0.00	0.00	0.00
0	3	3	0.00	3.30	2.90	100.00	0.00	0.00	0.00



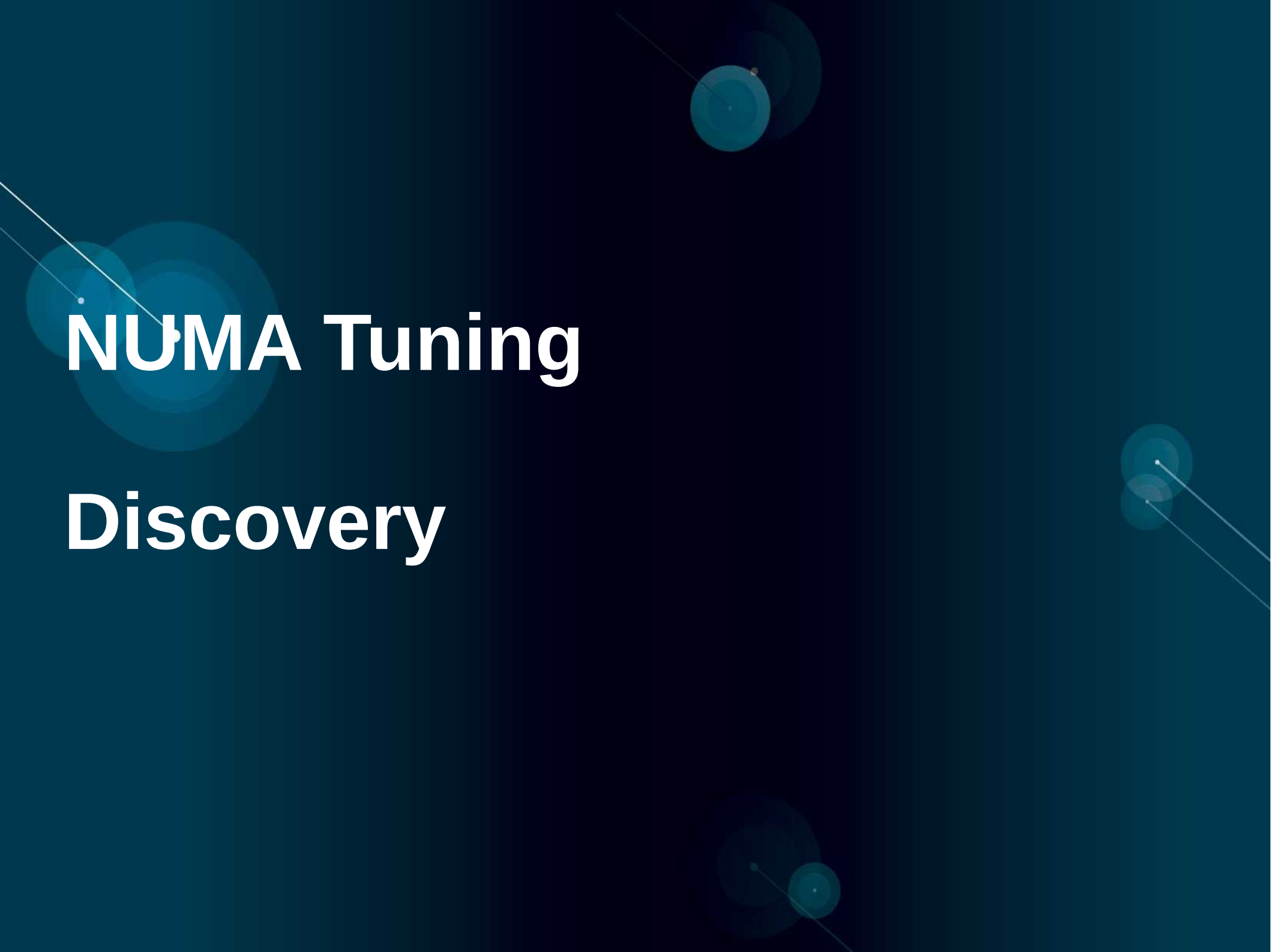
Tuned: Storage Performance Boost

RHEL7 RC1 3.10 File System In Cache Perf

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



Larger is better

The background is a dark teal gradient. It features several abstract elements: a large, semi-transparent teal circle on the left side; a smaller teal circle with a thin white line passing through its center in the top right; another teal circle with a thin white line passing through its center on the right side; and a small teal circle with a thin white line passing through its center in the bottom right. The text 'NUMA Tuning' is written in a bold, white, sans-serif font, positioned in the upper left quadrant of the slide.

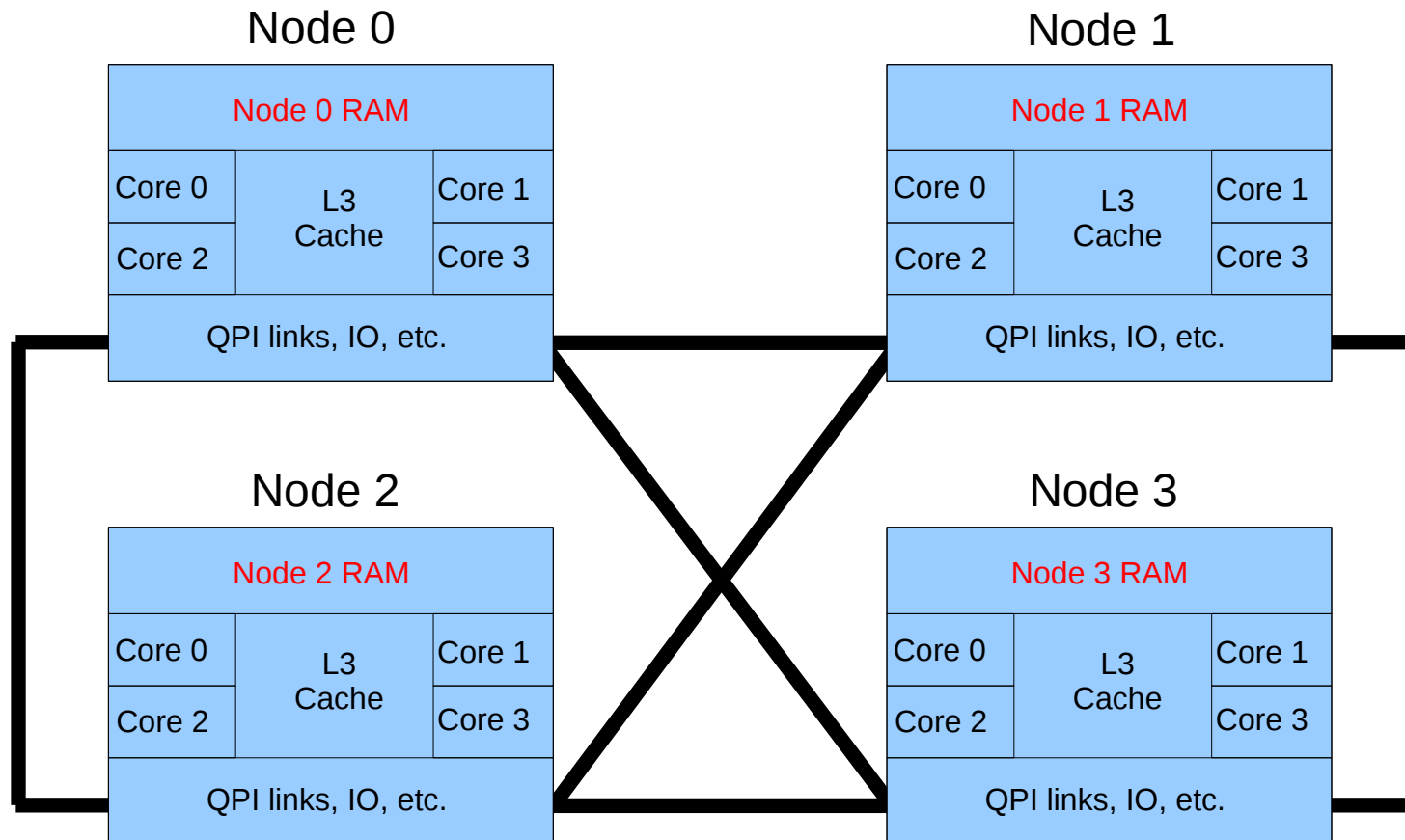
NUMA Tuning

Discovery

What is NUMA?

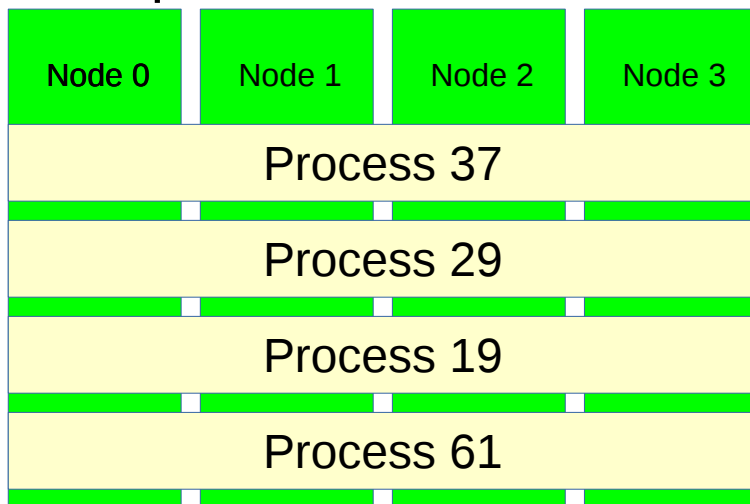
- Non Uniform Memory Access
- A result of making bigger systems more scalable by distributing system memory near individual CPUs....
- Practically all multi-socket systems have NUMA
 - Most servers have 1 NUMA node / socket
 - Recent AMD systems may have 2 NUMA nodes / socket

Typical NUMA System

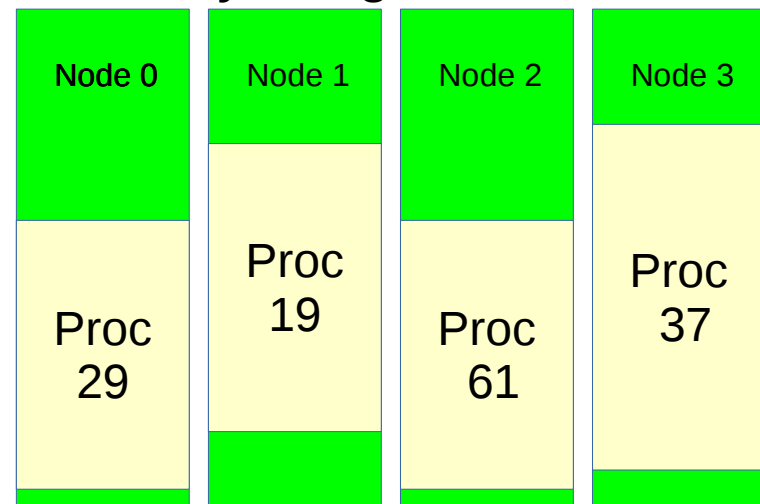


Goal: Align process memory and CPU threads within nodes

Before: processes use cpus & memory from multiple nodes.

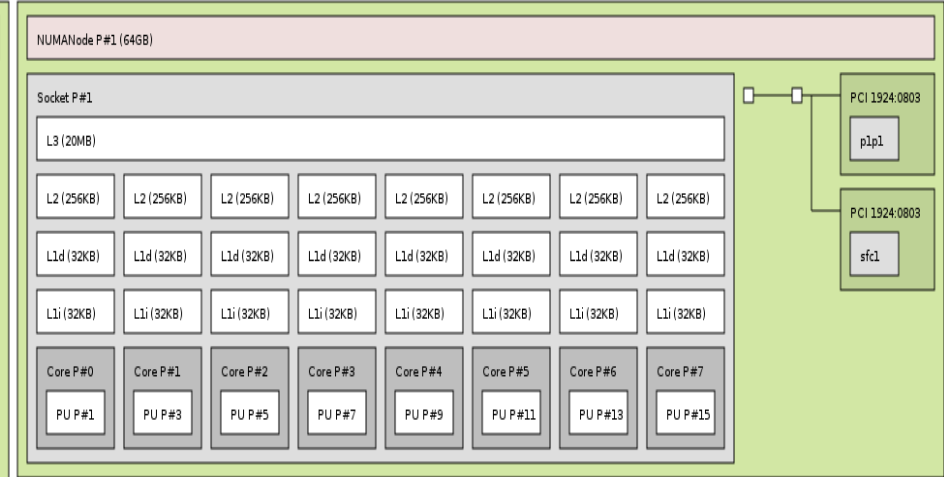
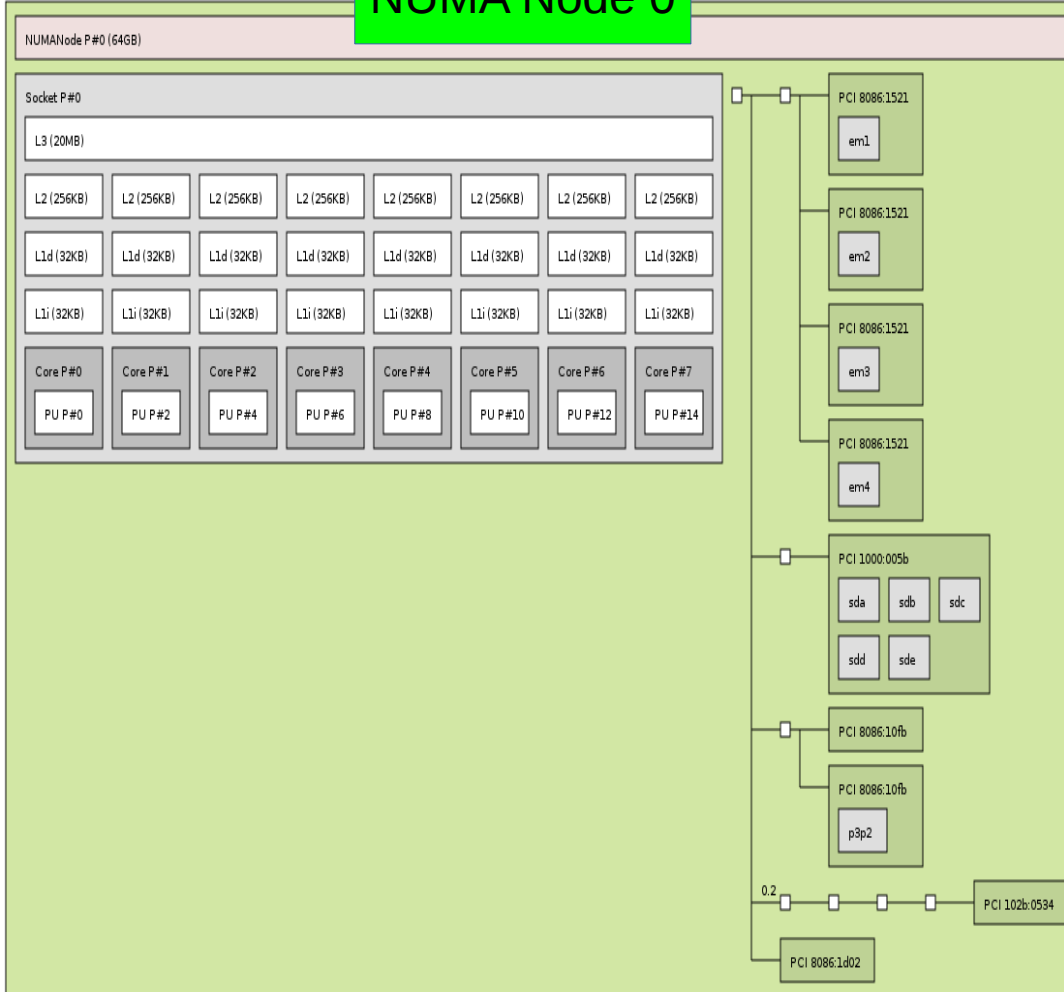


After: processes have more “localized” cpu & memory usage.



Know your hardware: Istopo

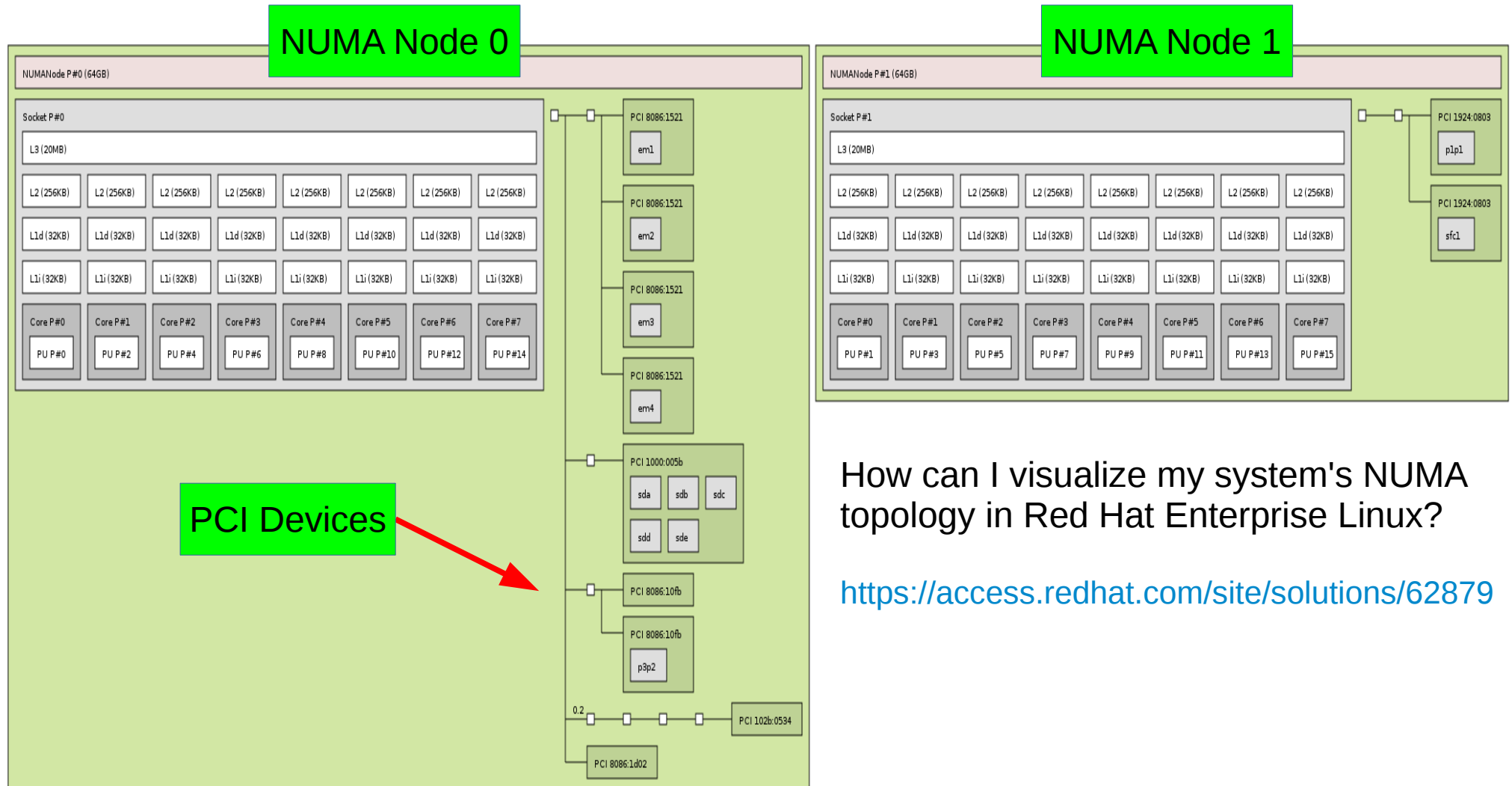
NUMA Node 0



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

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

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
```

NUMA Topology and PCI Bus

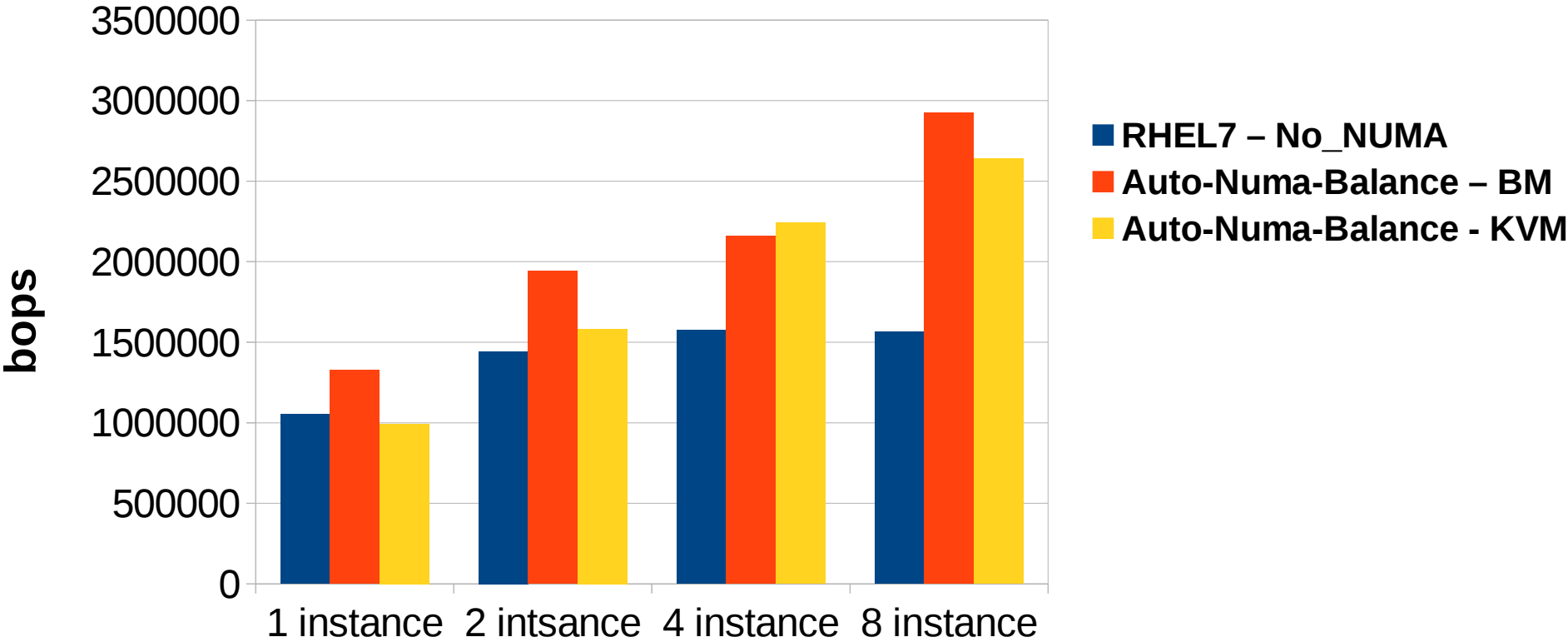
- Install adapters “close” to the CPU that will run the performance critical application.
- When BIOS reports locality, irqbalance handles NUMA/IRQ affinity automatically.

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



NUMA Tools: numastat

Completely rewritten for RHEL6.4

Per-node /proc/meminfo
Backwards compatible

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 – 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	2791	56	37	2933
56919 (java)	2769	76	55	32	2932
56920 (java)	19	55	77	2780	2932
56921 (java)	97	65	2727	47	2936
Total	2935	2987	2916	2896	11734

Overview of Performance Analysis Utilities

perf

Userspace tool to read CPU
counters and kernel tracepoints

perf list

List counters/tracepoints available on your system

```
# perf list
```

```
List of pre-defined events (to be used in -e):
```

cpu-cycles OR cycles	[Hardware event]
instructions	[Hardware event]
cache-references	[Hardware event]
cache-misses	[Hardware event]
branch-instructions OR branches	[Hardware event]
branch-misses	[Hardware event]
cpu-clock	[Software event]
task-clock	[Software event]
page-faults OR faults	[Software event]
context-switches OR cs	[Software event]
cpu-migrations OR migrations	[Software event]
minor-faults	[Software event]
major-faults	[Software event]

perf list

grep for something interesting,
maybe to see what numabalance is
doing ?

```
# perf list | grep sched: | grep numa
sched:sched_move_numa          [Tracepoint event]
sched:sched_stick_numa         [Tracepoint event]
sched:sched_swap_numa         [Tracepoint event]
```

perf top

System-wide 'top' view of busy functions

```
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
```



perf record

- *Record system-wide (-a)*

perf record

- Record system-wide (-a)
- *Or record a single command*

perf record

- Record system-wide (-a)
- Or record a single command
- *Or record an existing process (-p)*

perf record

- Record system-wide (-a)
- Or record a single command
- Or record an existing process (-p)
- *Or add call-chain recording (-g)*

perf record

- Record system-wide (-a)
- Or record a single command
- Or record an existing process (-p)
- Or add call-chain recording (-g)
- *Or only record specific events (-e)*

perf record

- Record system-wide (-a)

```
# perf record -a dd if=/dev/zero of=test bs=1M count=1000 conv=fdatasync  
oflag=direct
```

```
1000+0 records in
```

```
1000+0 records out
```

```
1048576000 bytes (1.0 GB) copied, 6.26617 s, 167 MB/s
```

```
[ perf record: Woken up 1 times to write data ]
```

```
[ perf record: Captured and wrote 0.305 MB perf.data (~13311 samples) ]
```

perf report

```
# Overhead Command Shared Object
# .....
#
43.53% dd [kernel.kallsyms] [k] __clear_user
|
--- __clear_user
|
|--99.75%-- read_zero.part.5
|          read_zero
|          vfs_read
|          sys_read
|          system_call_fastpath
|          __GI___libc_read
|
--0.25%-- [...]

5.37% dd [kernel.kallsyms] [k] do_blockdev_direct_IO
|
--- do_blockdev_direct_IO
|   __blockdev_direct_IO
|   xfs_vm_direct_IO
|   generic_file_direct_write
|   xfs_file_dio_aio_write
|   xfs_file_aio_write
|   do_sync_write
```

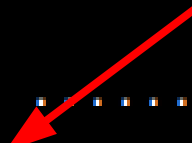
/dev/zero

oflag=direct

perf diff

Compare 2 perf recordings

```
# perf diff
# Event 'cycles'
#
# Baseline      Delta      Shared Object      Symbol
# .....
#
12.88% -12.27% [kernel.kallsyms] [k] __lookup_mnt
11.97% -11.17% systemd [.] 0x000000000000064968
4.32% +6.43% libdbus-1.so.3.7.4 [.] 0x000000000000029258
4.06% +4.72% dbus-daemon [.] 0x000000000000014a6e
3.79% -3.79% libglib-2.0.so.0.3600.3 [.] 0x000000000000088d6a
3.72% +0.25% [kernel.kallsyms] [k] seq_list_start
```



Perf - analyzing the raw data for mem accesses.

(c2c output, more later)

Count	Response Type	
-----	-----	
152729	[LOAD, L2,HIT ,SNP NONE]	// Hit in the L2 cache
143821	[LOAD, LFB,HIT ,SNP NONE]	// "In flight" loads to Line Fill Buffer
116187	[LOAD, RMT_RAM,HIT ,SNP NONE,SNP MISS]	// Hit in a remote node's ram
89248	[LOAD, L1,HIT ,SNP NONE]	// Hit in the L1 cache
40723	[LOAD, LCL_RAM,HIT ,SNP NONE,SNP MISS]	// Hit in local node's ram
40614	[LOAD, LCL_LLC,HIT ,SNP NONE]	// Hit in local last level cache
826	[STORE,L1,HIT]	
769	[STORE,L1,MISS]	
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]	

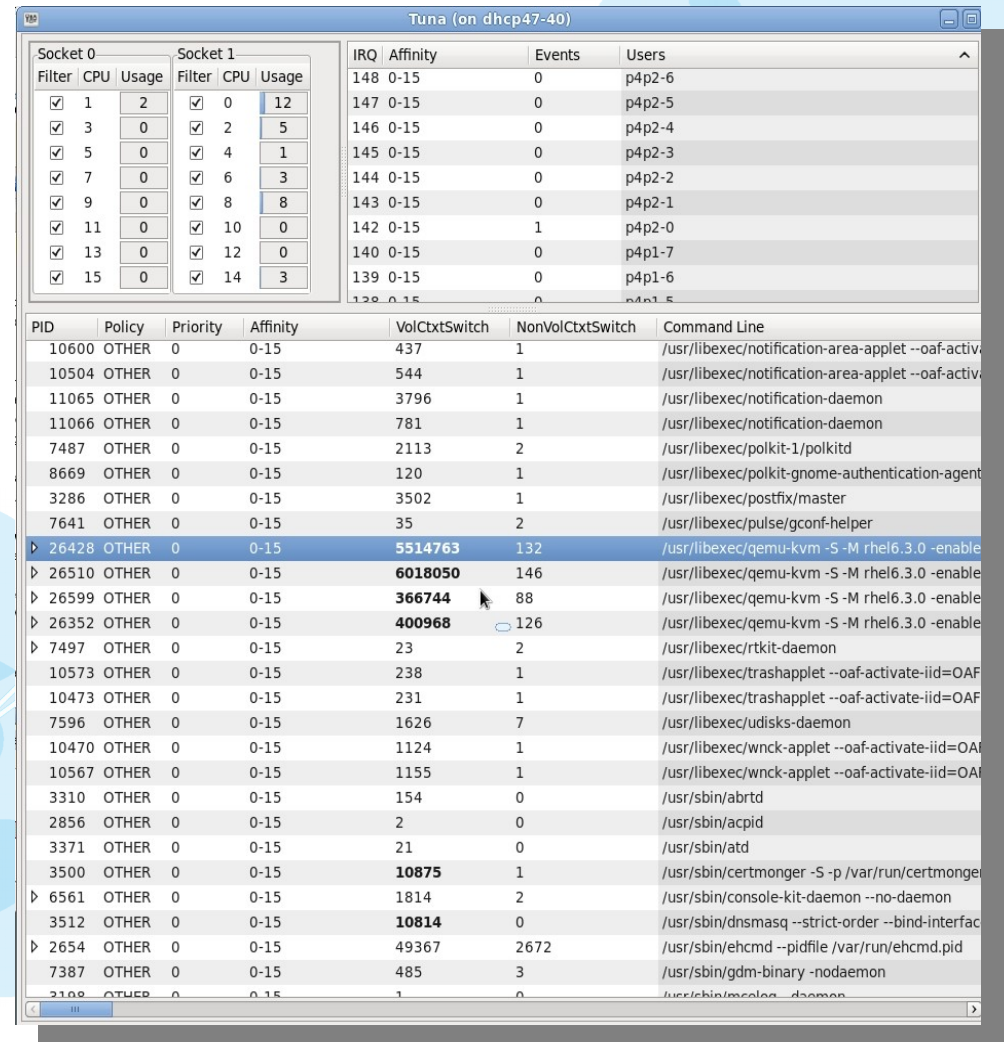
Load latencies – key info to know, not shown here, but in the raw data

The background is a dark teal gradient. It features several abstract elements: a large, semi-transparent teal circle on the left with a white dot at its center and a thin white line extending from it towards the top left; a smaller teal circle at the top center with a white dot and a line extending towards the top right; a cluster of three overlapping teal circles on the right with white dots and lines extending towards the right edge; and a teal circle at the bottom center with a white dot and a line extending towards the bottom right.

Tuna

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



Socket 0			Socket 1		
Filter	CPU	Usage	Filter	CPU	Usage
☒	1	2	☒	0	12
☒	3	0	☒	2	5
☒	5	0	☒	4	1
☒	7	0	☒	6	3
☒	9	0	☒	8	8
☒	11	0	☒	10	0
☒	13	0	☒	12	0
☒	15	0	☒	14	3

IRQ	Affinity	Events	Users
148	0-15	0	p4p2-6
147	0-15	0	p4p2-5
146	0-15	0	p4p2-4
145	0-15	0	p4p2-3
144	0-15	0	p4p2-2
143	0-15	0	p4p2-1
142	0-15	1	p4p2-0
140	0-15	0	p4p1-7
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/ehcnd --pidfile /var/run/ehcnd.pid
7387	OTHER	0	0-15	485	3	/usr/sbin/gdm-binary -nodaemon
2108	OTHER	0	0-15	1	0	/usr/sbin/mcals-daemon

Tuna (RHEL6/7)

1

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

2

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

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	127740	2	hald

Tuna GUI Capabilities Updated for RHEL7

Monitoring **Profile management** Profile editing

Current active tuna profile: example.conf ▾

 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

VM

- vm.dirty_expire_centisecs
- vm.dirty_ratio
- vm.dirty_writeback_centisecs
- vm.laptop_mode
- vm.memory_failure_early_kill
- vm.swappiness

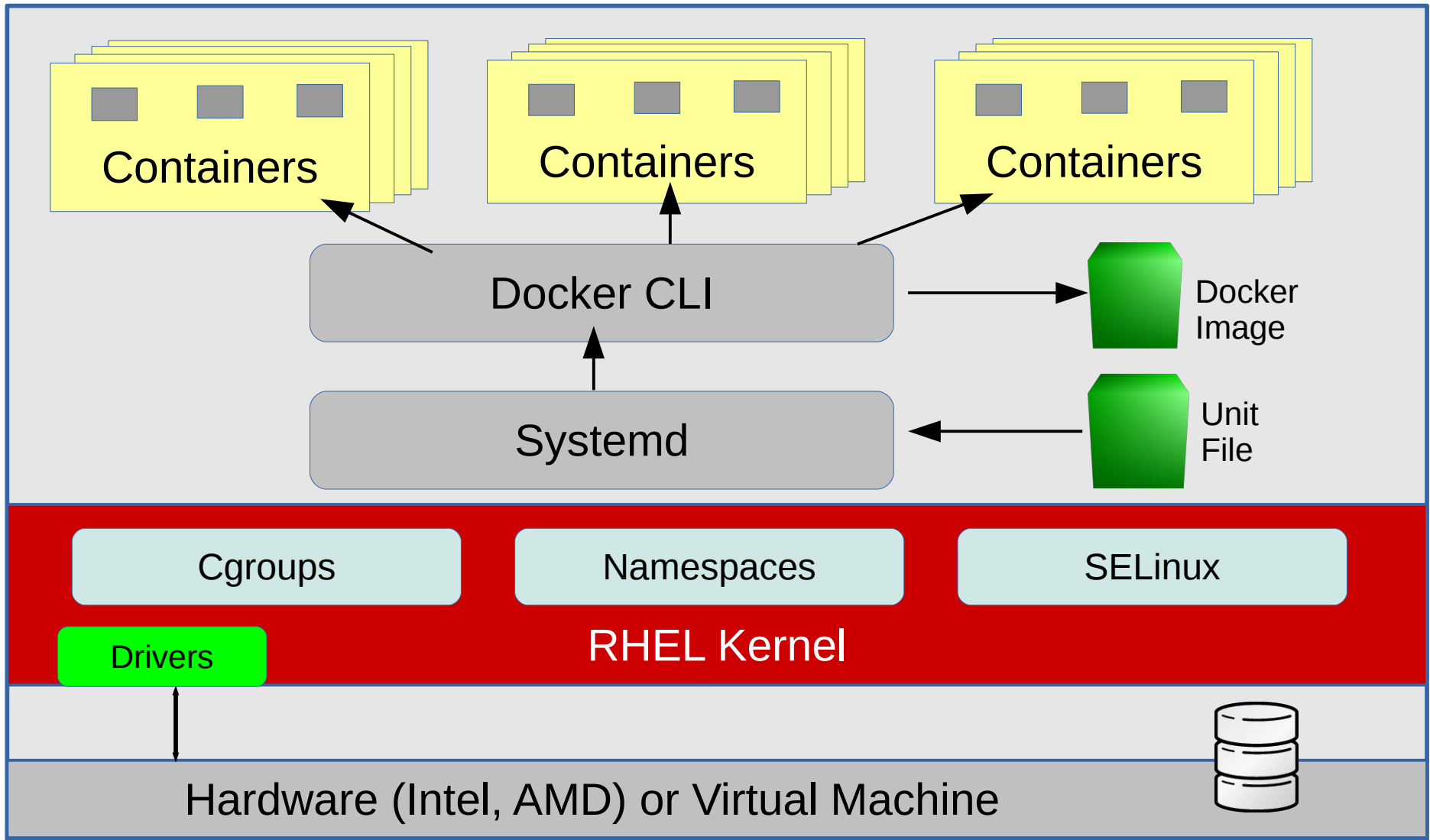
Network IPv6

- ipv6.conf.all.forwarding
- ipv6.conf.default.forwarding
- ipv6.conf.docker0.forwarding
- ipv6.conf.em1.forwarding
- ipv6.conf.em2.forwarding

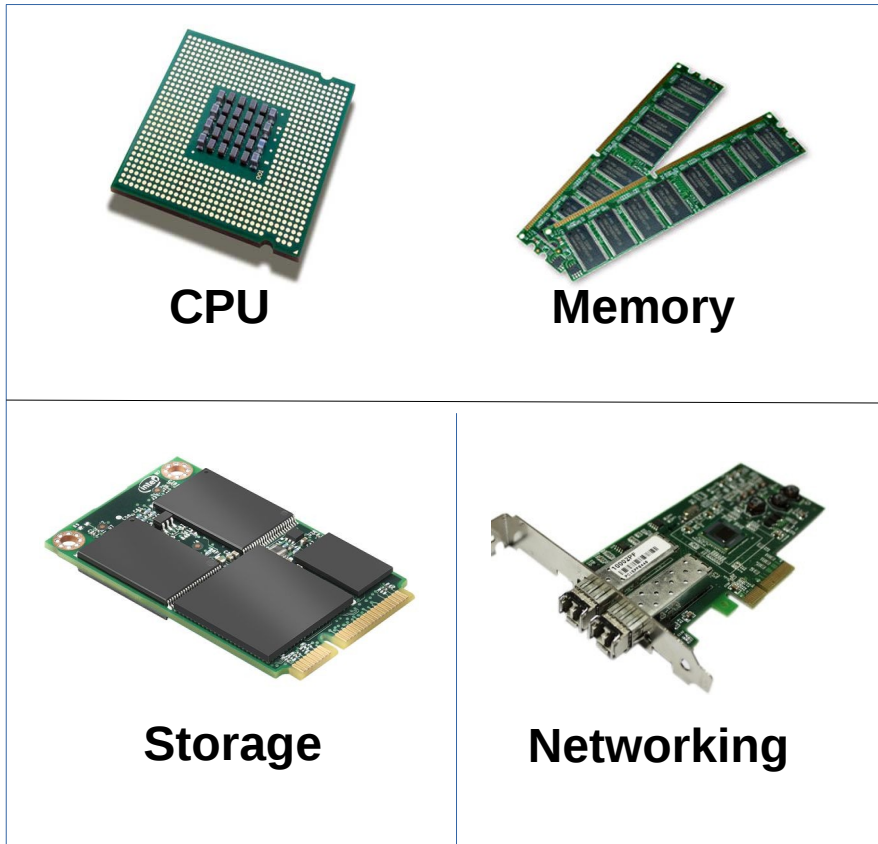
The background is a dark teal gradient. It features several abstract elements: a large, semi-transparent teal circle in the upper left with a white dot at its center and a thin white line extending from it; a smaller teal circle in the upper right with a thin white line extending from it; a cluster of three overlapping teal circles on the right side with thin white lines extending from them; and a small teal circle in the lower right with a thin white line extending from it.

Containers

RHEL 7 Containers Architecture



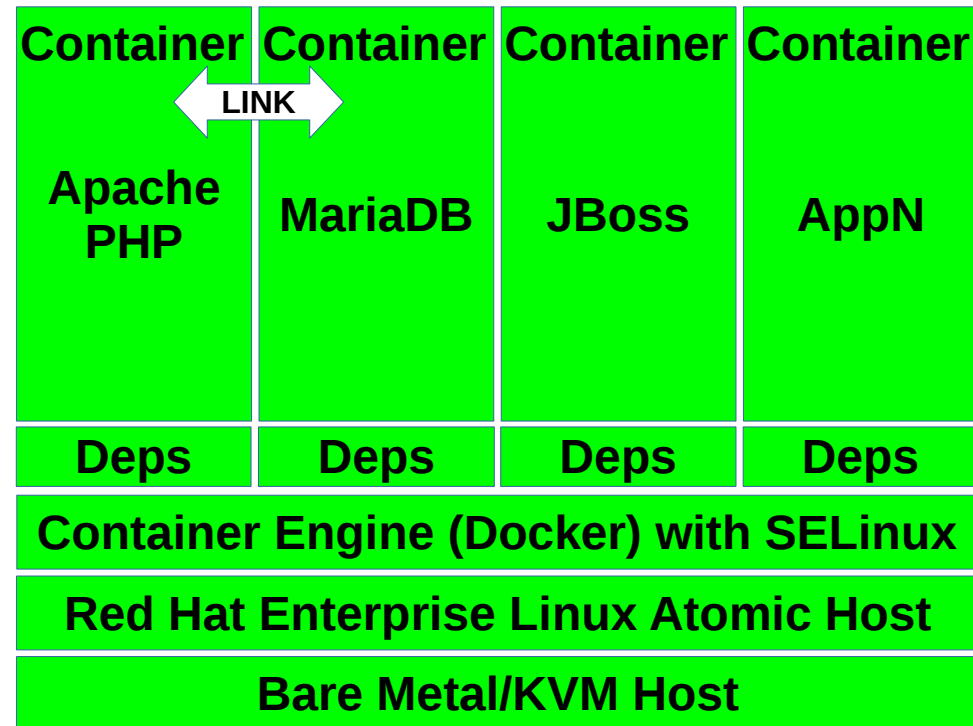
Performance food groups – still relevant in containers



- numa_balancing
- irqbalance
- Storage/Networking passthrough
- Short circuits via shared namespaces
- Network and Storage tuning still applies
- Power management

Density

- Depends on
 - Hardware/Cloud capabilities
 - Workload requirements
 - Overcommit configuration
 - Network capabilities/config
 - ~1000 using Linux Bridge
 - Higher with SDN
 - Storage capabilities/config
 - Loop-LVM: no setup, Good performance...for development
 - Direct-LVM: requires setup, best performance/scale for production



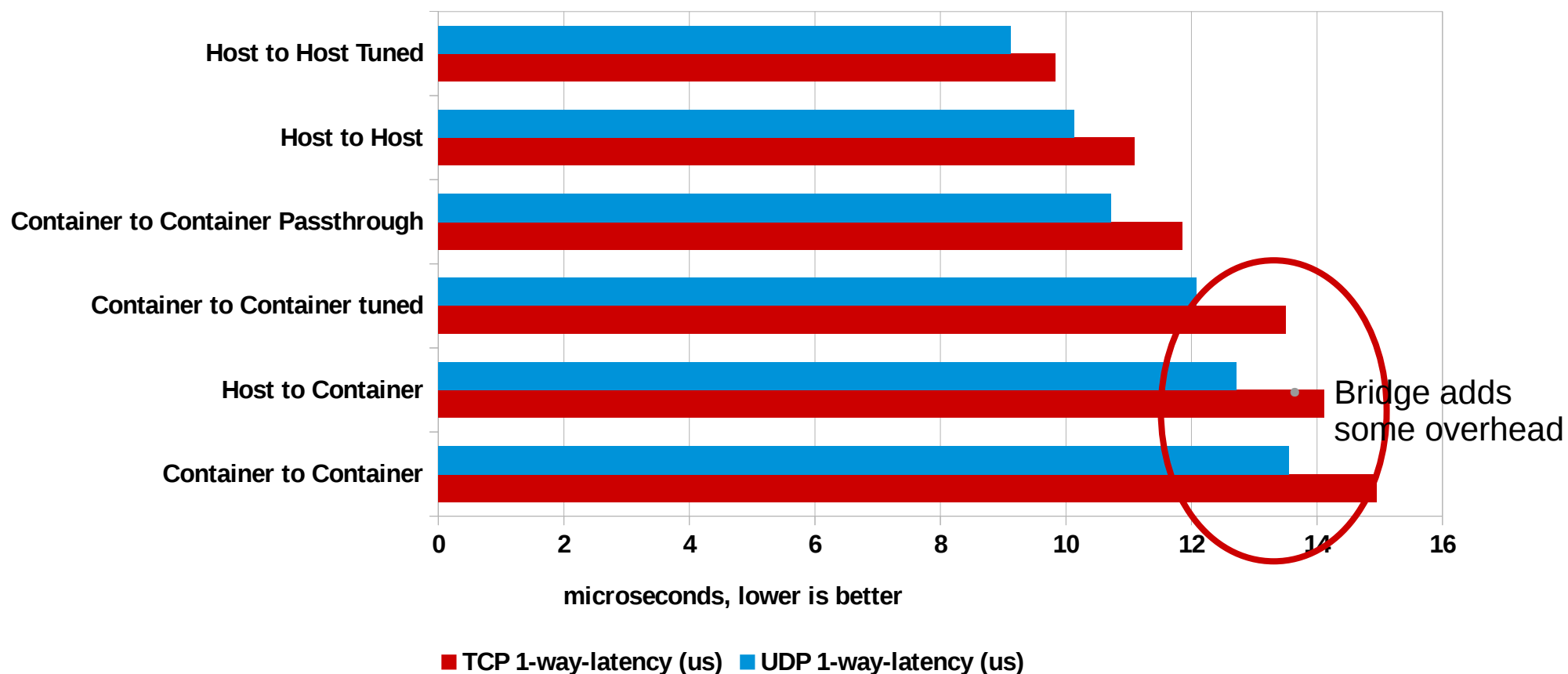
Container Performance vs Bare Metal

Network Performance: netperf, RR tests (latency)

- With cpu pinning and IRQ Affinity in place

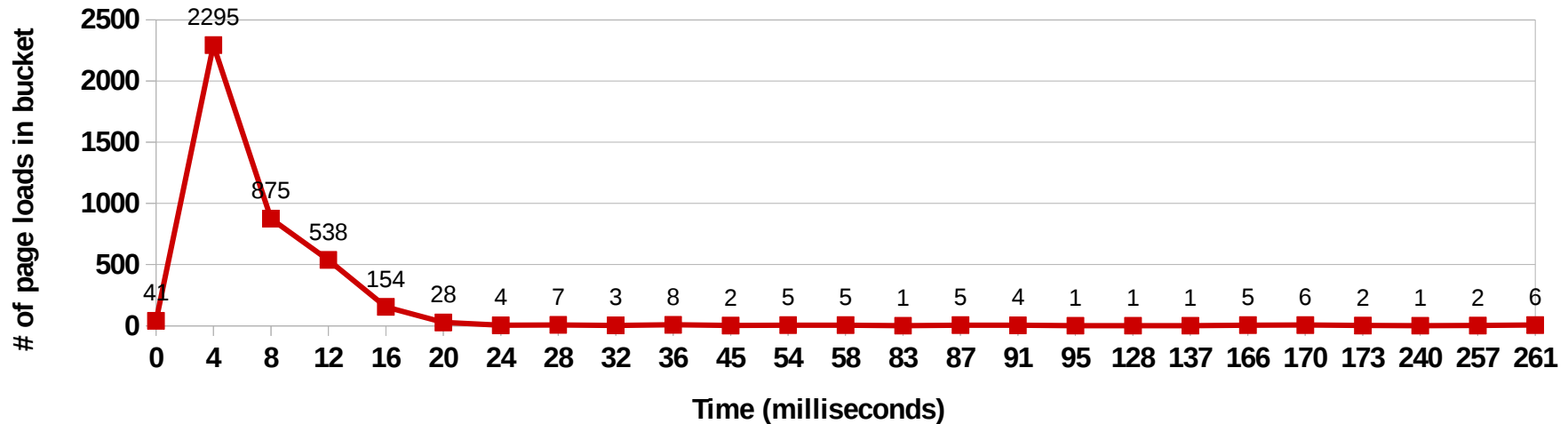
netperf RR tests

Average one-way latency

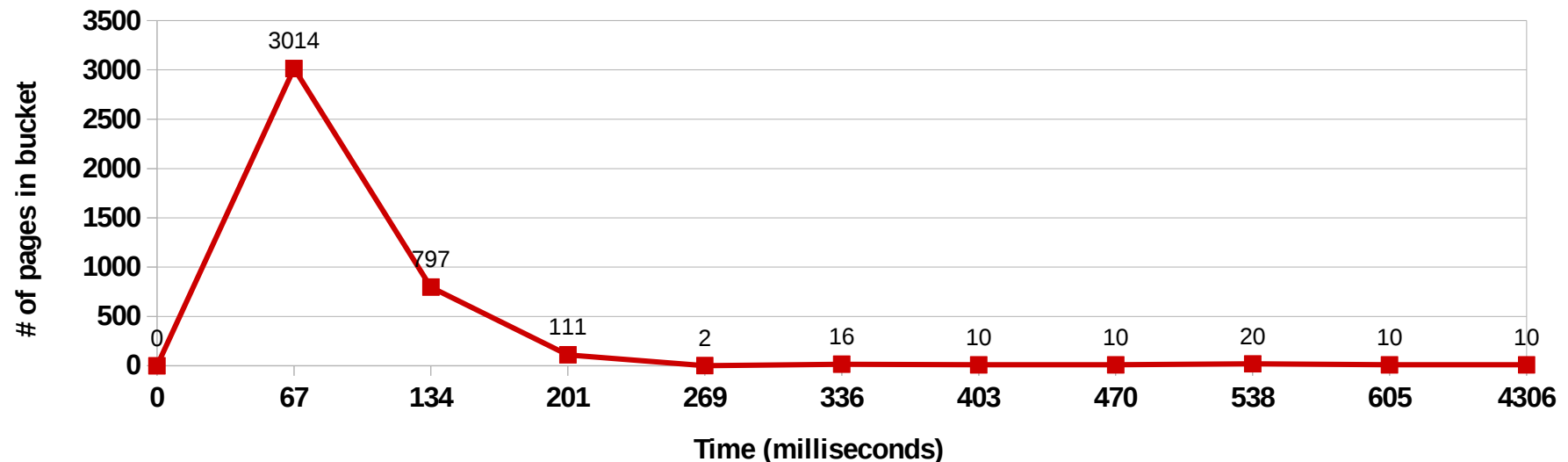


Apache, PHP, MariaDB in a container [remote JMeter]

Histogram of JMeter Page Load Latency
Apache/PHP



Histogram of JMeter Page Load Latency:
Apache, PHP, MariaDB, no_cache_query

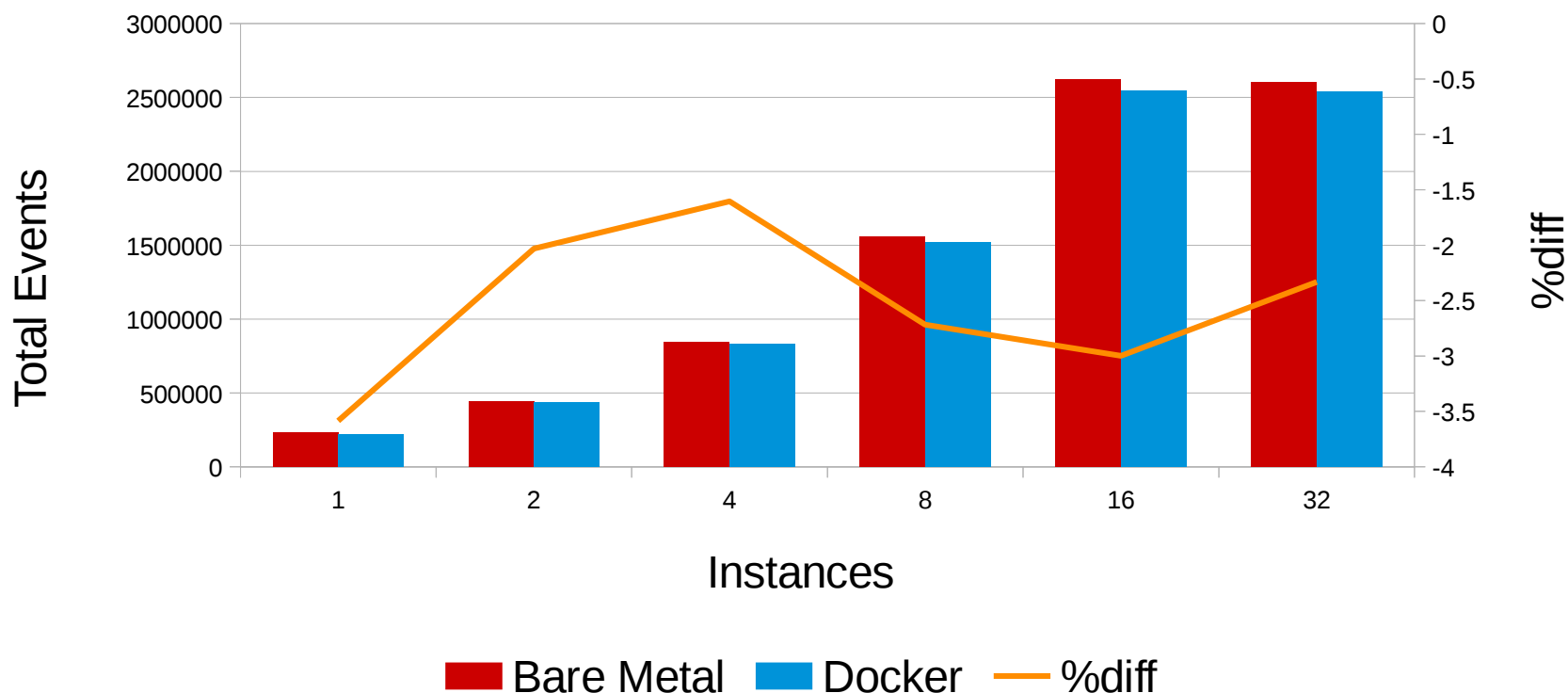


CPU/Memory Performance Test Plan: sysbench oltp

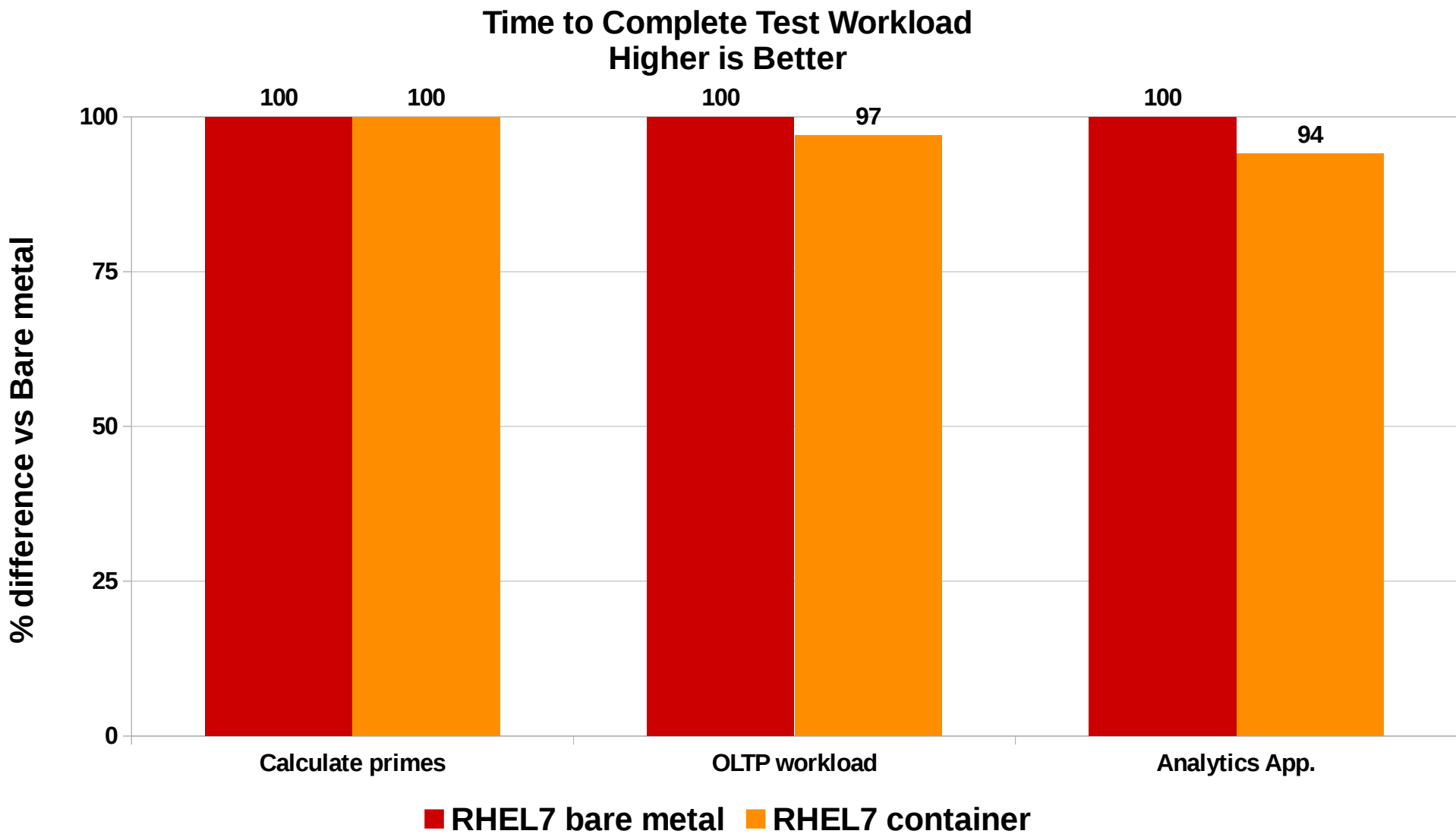
- sysbench --test=oltp mariadb 2M rows, 300s
- Within 1-4% of bare metal.
- 32 threads = overcommit

sysbench --test=oltp (16 core SNB)

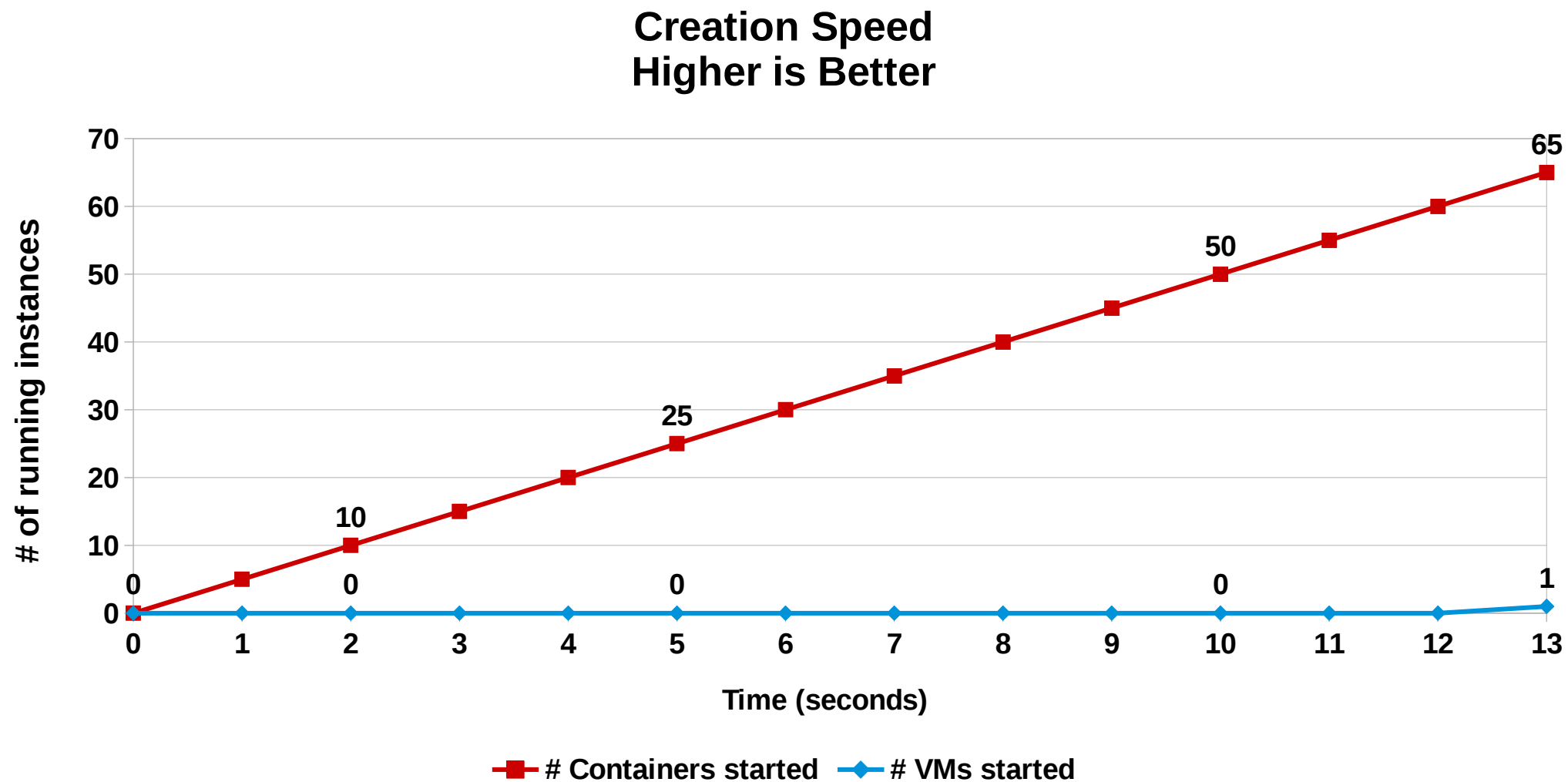
mariadb 2M rows 300s



Container performance across multiple workloads



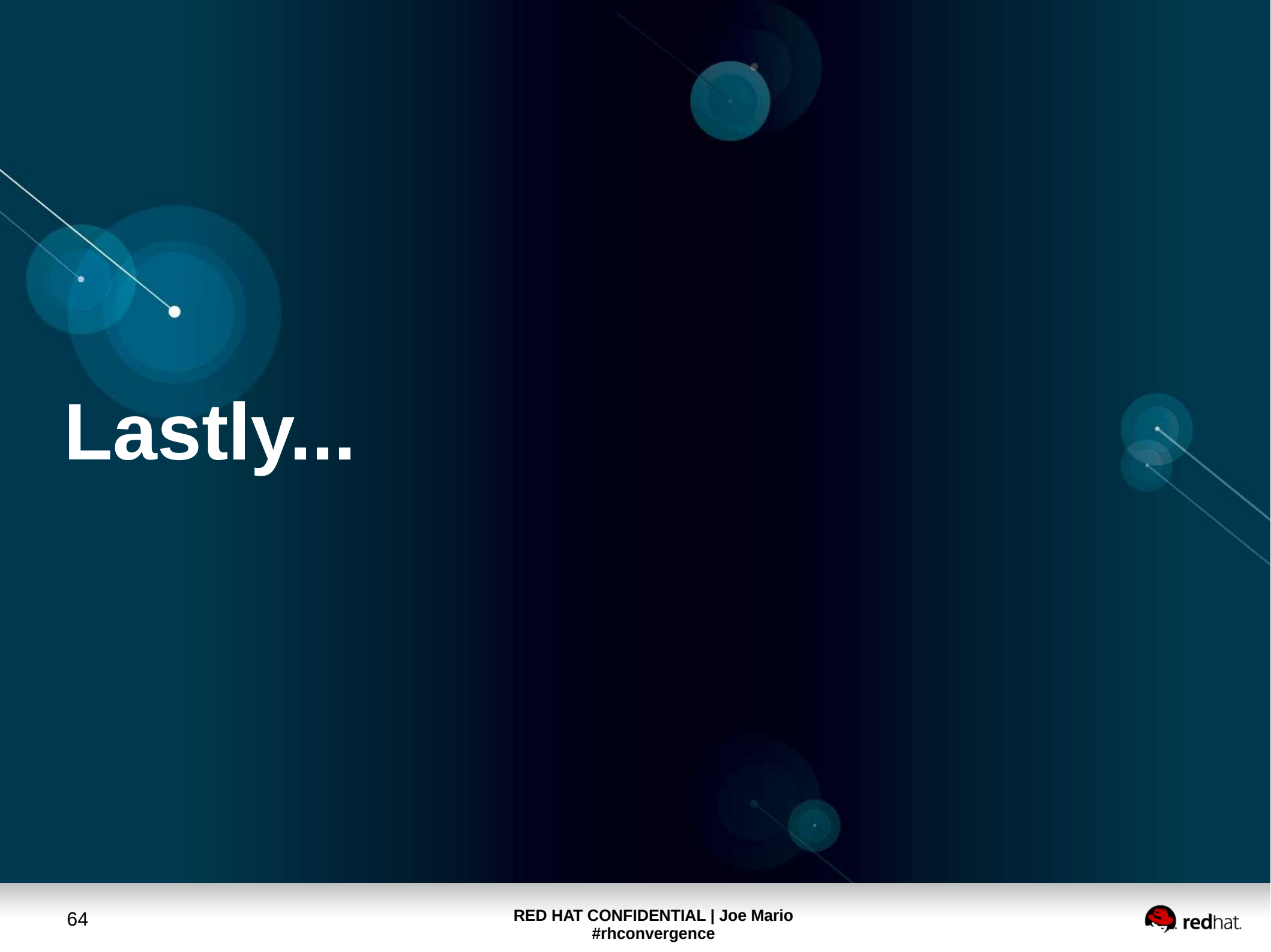
Container creation speed vs VMs



Dockerfile for Performance Analysis

- Includes all the common debug/analysis tools
 - Some tests run with additional tracing
- Sets up the environment
 - Tests must know if they're run in a container or not
- Pulls in necessary git repos
 - Repos include all the tests/workload generators
- Mounts volumes:
 - For test state information, multi-host semaphore/locking
 - Tests write output to persistent storage

Red Hat delivering this in a 'rhel-tools' container image. (GA with Atomic)



Lastly...

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
 - Ex:[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](#)

Useful Utilities

Supportability

- redhat-support-tool
- sos
- kdump
- perf
- psmisc
- strace
- sysstat
- systemtap
- ftrace / trace-cmd
- Util-linux-ng
- pcp

NUMA

- hwloc
- Intel PCM
- numactl
- numad
- numatop (01.org)

Networking

- dropwatch
- ethtool
- netsniff-ng (EPEL6)
- tcpdump
- wireshark/tshark

Power/Tuning

- cpupowerutils (R6)
- kernel-tools (R7)
- powertop
- tuna
- tuned

Storage

- blktrace
- iotop
- iostat

Questions

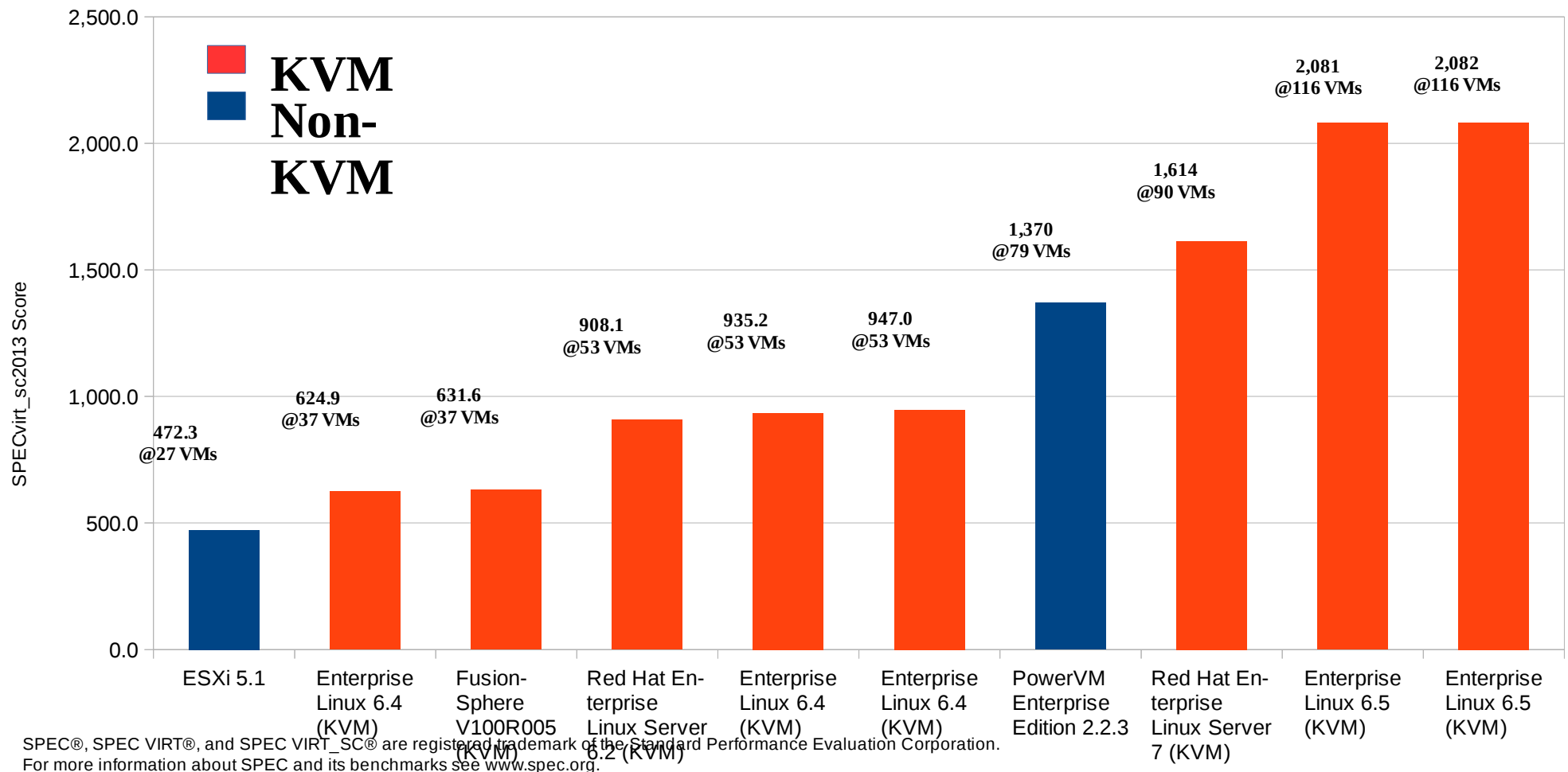
Backup slides

SPECvirt

Latest SPECvirt 2013 results

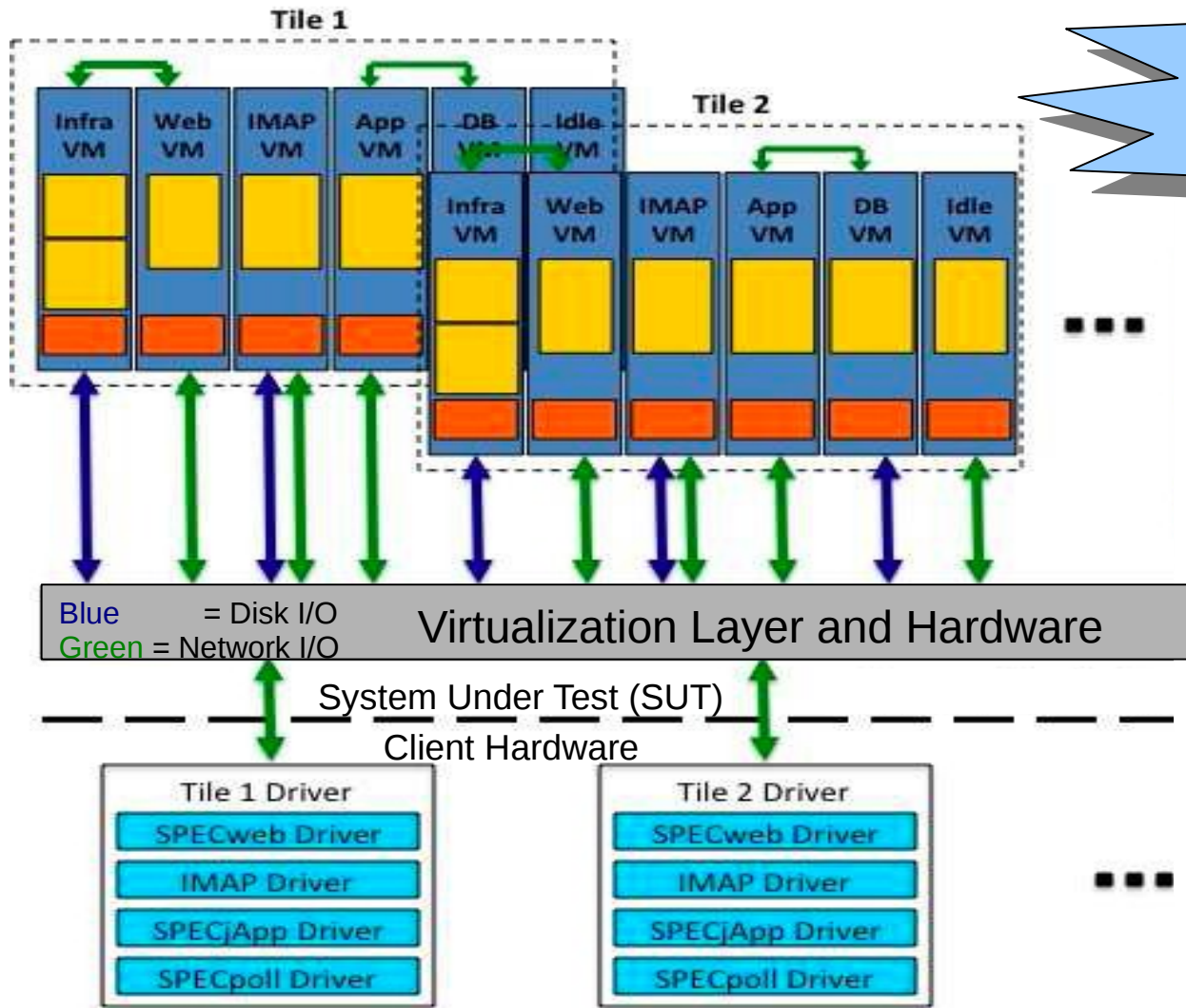
All SPECvirt_sc2013 Results Sorted by Performance

(As of October 6, 2014)



SPECvirt2013:

RHEL 7/6 and KVM Post Industry Leading Results



> 1 SPECvirt Tile/core

Key Enablers:

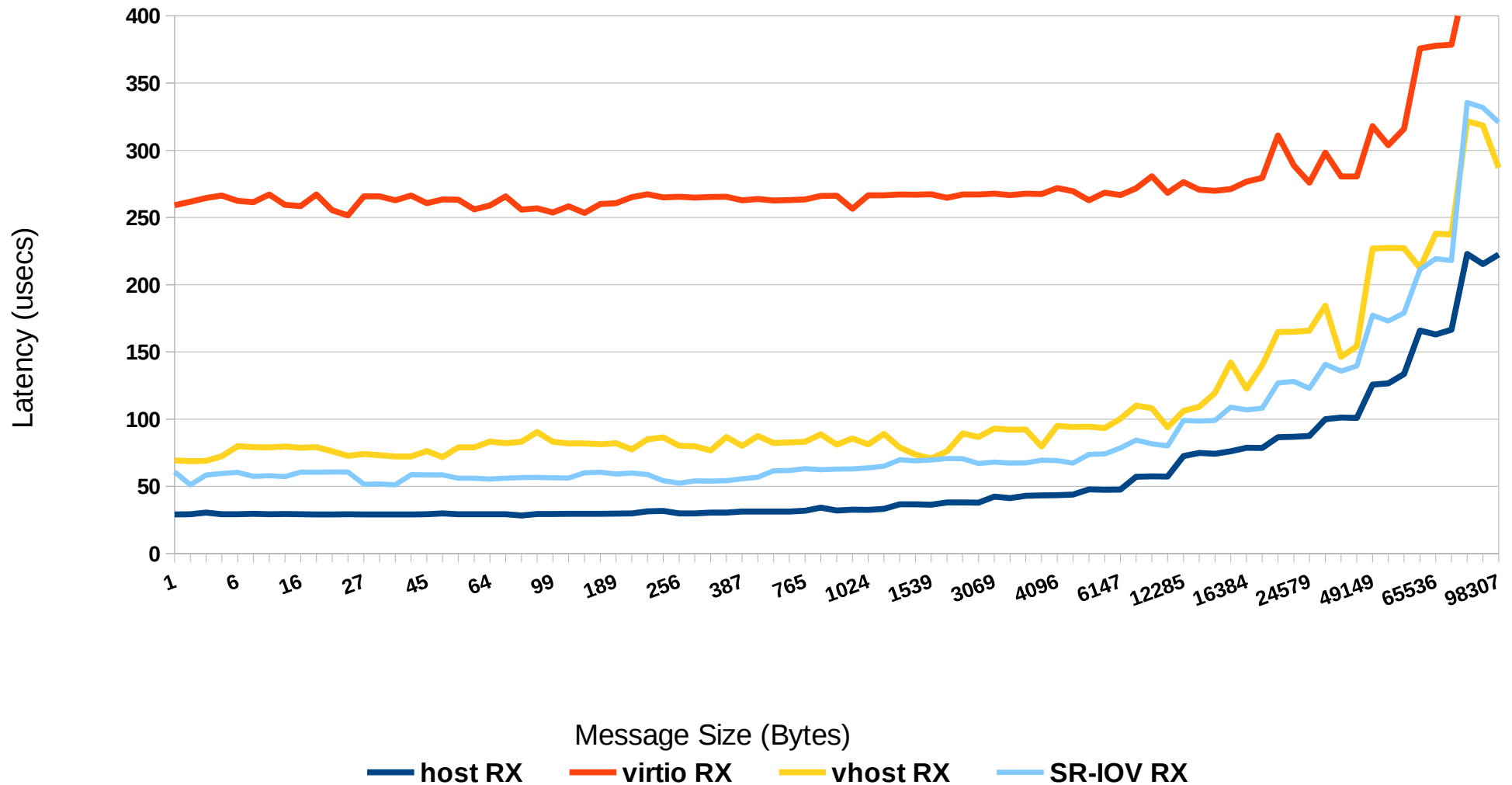
- ✓ SR-IOV
- ✓ Huge Pages
- ✓ NUMA
- ✓ Node Binding

http://www.spec.org/virt_sc2013/results/

Virtualization Tuning – Network – Latency Comparison

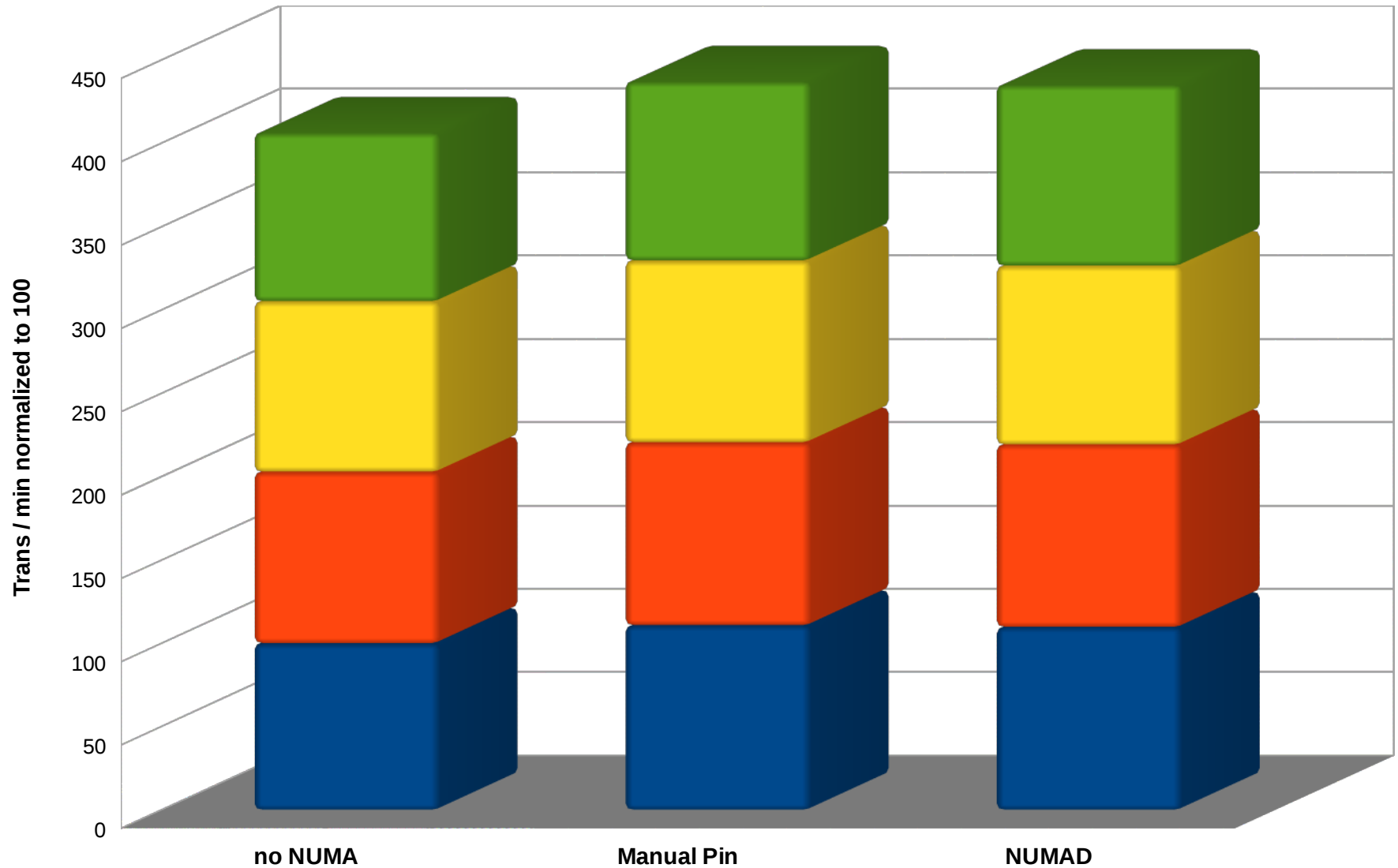
Network Latency by Guest Interface Method

Guest Receive (Lower is better)



Virt Tuning – Using NUMA

4 Virtual Machines running OLTP workload



Backup

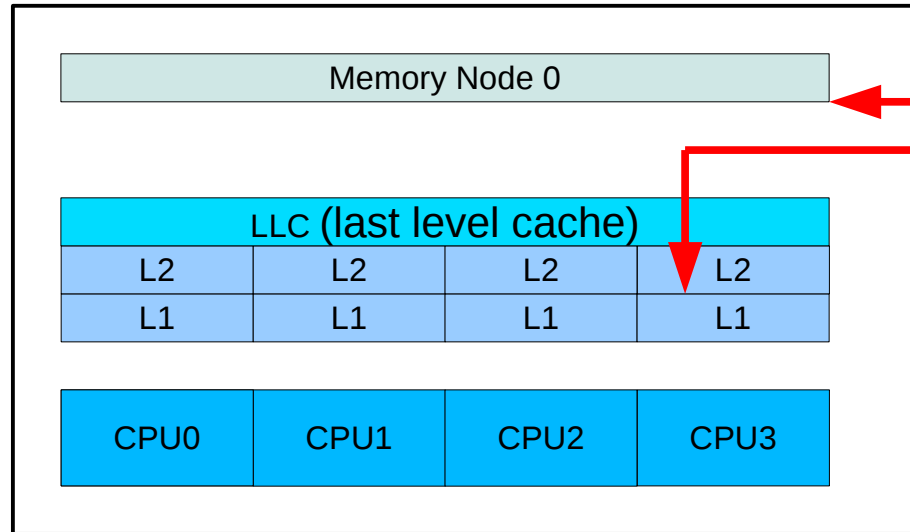
Detecting cache to cache contention.

Cache-to-Cache false sharing tool

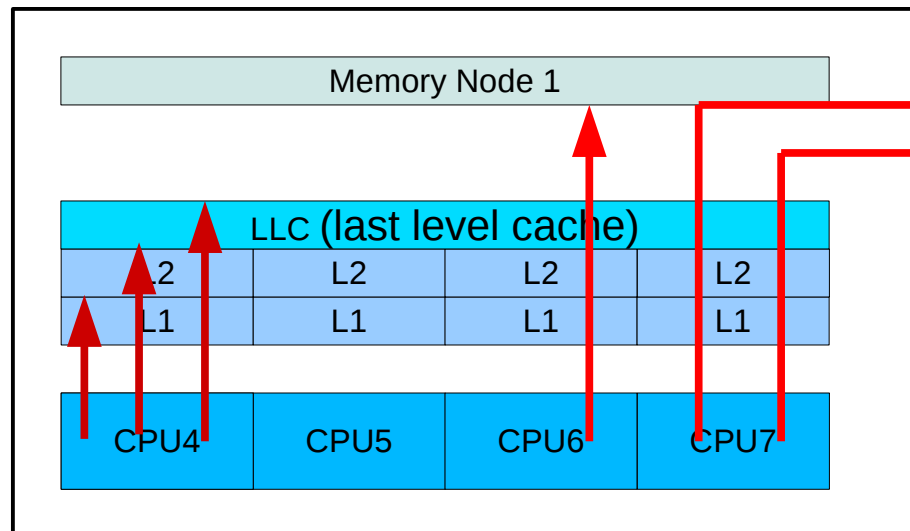
- Red Hat developed an enhancement to the 'perf' tool to detect simultaneous cross-numa cacheline contention.
- Critical to:
 - Shared memory applications
 - Multi-threaded apps spanning multiple numa nodes
- Code being reviewed upstream
- Integrating it into the perf tool (“perf c2c”)
- Expect it to be available in RHEL7.2

Resolving a memory access

Socket 0

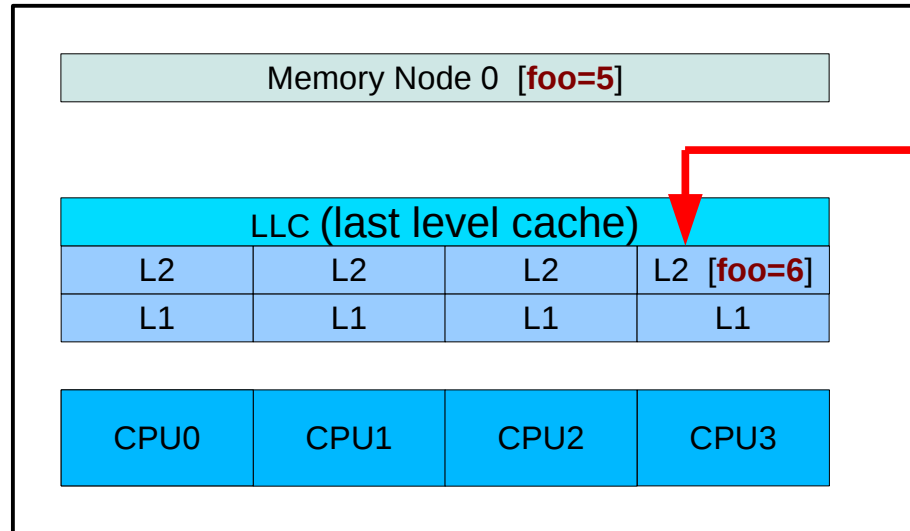


Socket 1

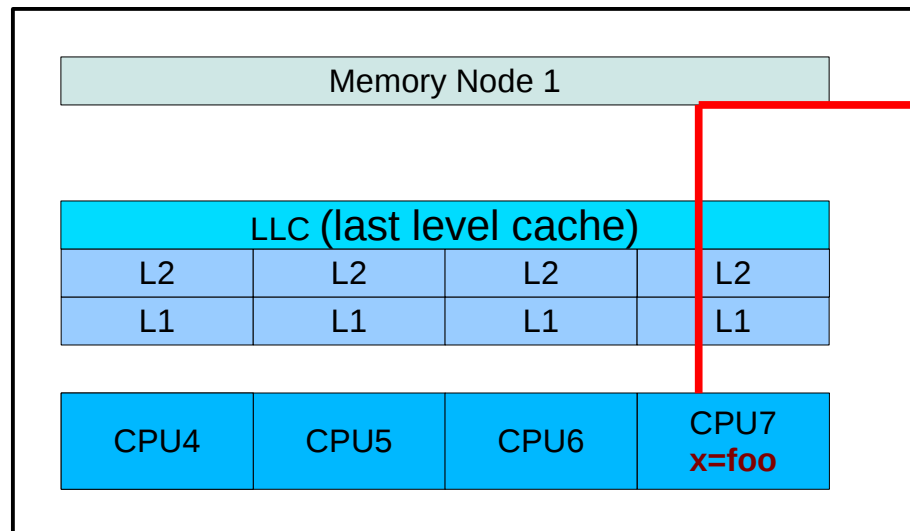


Resolving a memory access

Socket 0

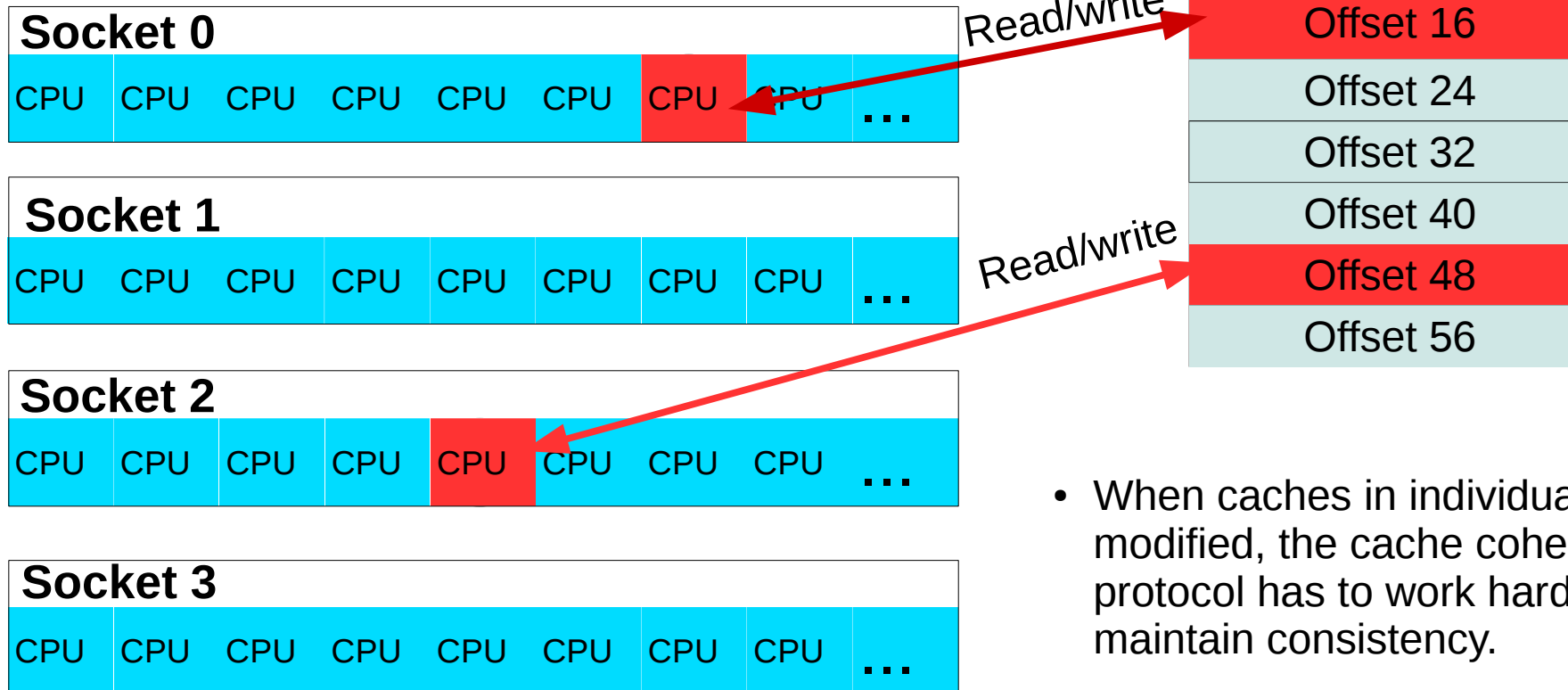


Socket 1



Cacheline Contention – high level

64 byte chunk of memory
(size of cacheline)



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

Output from “c2c data sharing” tool

=====									
Cache									CPU
#	Refs	Stores	Data Address	Pid	Tid	Inst Address	Symbol	Object	Participants
=====									
0	118789	273709	0x602380	37878					
	17734	136078	0x602380	37878	37878	0x401520	read_wrt_thread	a.out	0{0};
	13452	137631	0x602388	37878	37883	0x4015a0	read_wrt_thread	a.out	0{1};
	15134	0	0x6023a8	37878	37882	0x4011d7	reader_thread	a.out	1{5};
	14684	0	0x6023b0	37878	37880	0x4011d7	reader_thread	a.out	1{6};
	13864	0	0x6023b8	37878	37881	0x4011d7	reader_thread	a.out	1{7};
1	31	69	0xffff88023960df40	37878					
	13	69	0xffff88023960df70	37878	***	0xffffffff8109f8e5	update_cfs_rq_blocked_load	vmlinux	0{0,1,2}; 1{14,16};
	17	0	0xffff88023960df60	37878	***	0xffffffff8109fc2e	__update_entity_load_avg_contrib	vmlinux	0{0,1,2}; 1{14,16};
	1	0	0xffff88023960df78	37878	37882	0xffffffff8109fc4e	__update_entity_load_avg_contrib	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)