RED HAT SUMMIT

10 YEARS *and counting*

SAN FRANCISCO | APRIL 14-17, 2014

Performance Analysis and Tuning – Part 2

D. John Shakshober (Shak)
Sr Consulting Eng / Director Performance Engineering

Larry Woodman
Senior Consulting Engineer / Kernel VM

Jeremy Eder
Principal Software Engineer/ Performance Engineering

redhat.

Agenda: Performance Analysis Tuning Part II

- Part I
  - RHEL Evolution 5->6->7 – out-of-the-box tuned for Clouds - “tuned”
  - Auto_NUMA_Balance – tuned for NonUniform Memory Access (NUMA)
  - Cgroups / Containers
  - Scalabilty – Scheduler tunables
  - Transparent Hugepages, Static Hugepages 4K/2MB/1GB
- **Part II**
  - **Disk and Filesystem IO - Throughput-performance**
  - **Network Performance and Latency-performance**
  - **System Performance/Tools – perf, tuna, systemtap, performance-co-pilot**
- **Q & A**

RED HAT SUMMIT

10 YEARS *and counting*

SAN FRANCISCO | APRIL 14-17, 2014

Disk I/O in RHEL

redhat.

RHEL “tuned” package

```
Available profiles:
- balanced
- desktop
- latency-performance
- network-latency
- network-throughput
- throughput-performance
- virtual-guest
- virtual-host
Current active profile: throughput-performance
```

Tuned: Profile throughput-performance

throughput-performance

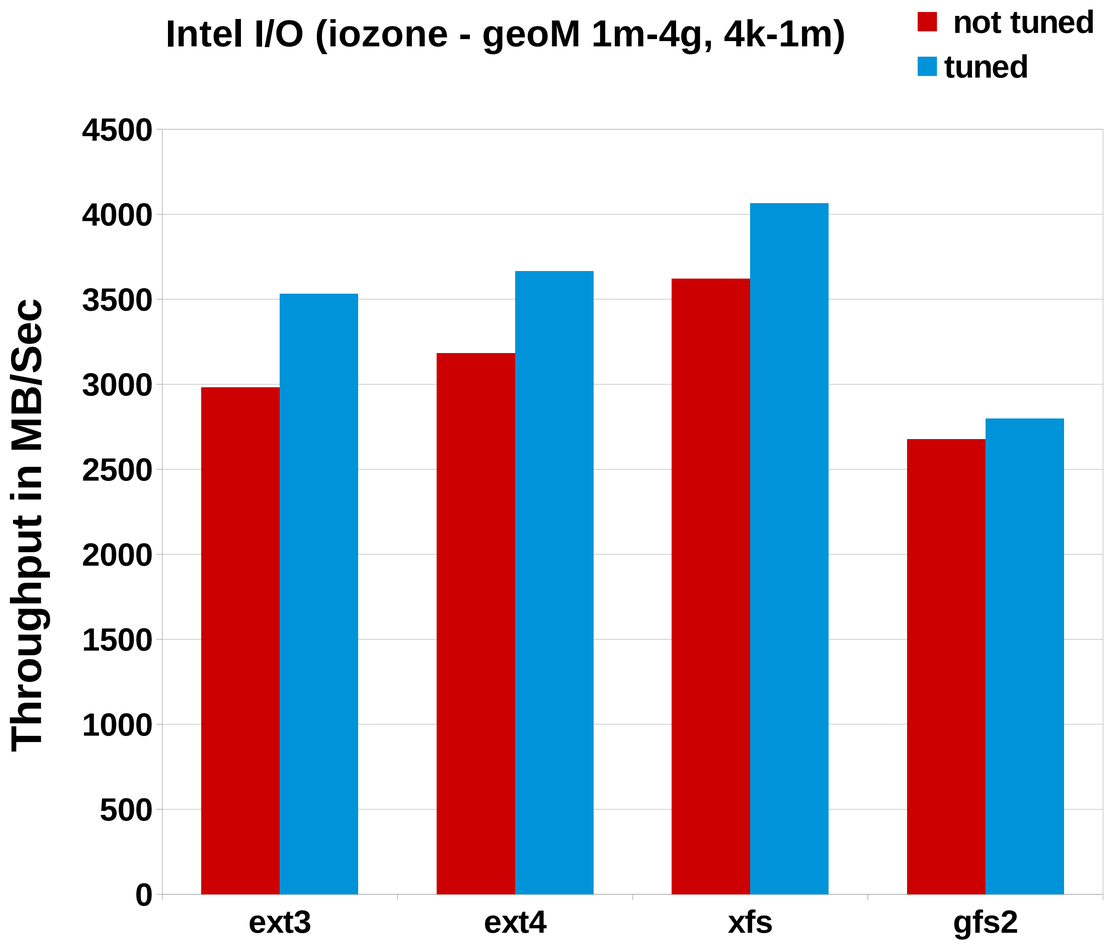
```
governor=performance
energy_perf_bias=performance
min_perf_pct=100
readahead=4096
kernel.sched_min_granularity_ns = 10000000
kernel.sched_wakeup_granularity_ns = 15000000
vm.dirty_background_ratio = 10
vm.swappiness=10
```

I/O Tuning – Understanding I/O Elevators

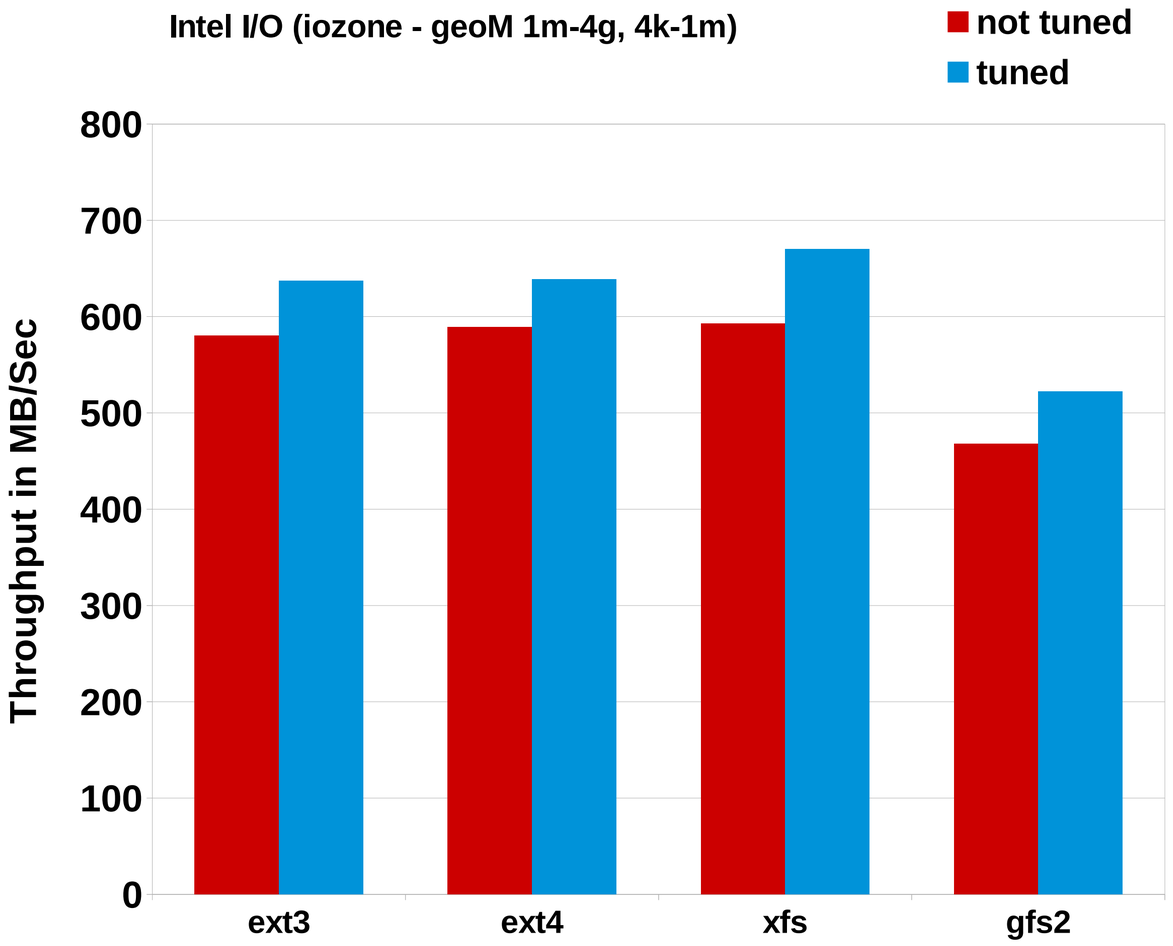
- Deadline – new RHEL7 default for all profiles
  - Two queues per device, one for read and one for writes
  - I/Os dispatched based on time spent in queue
- CFQ – used for system disks off SATA/SAS controllers
  - Per process queue
  - Each process queue gets fixed time slice (based on process priority)
- NOOP – used for high-end SSDs (Fusion IO etc)
  - FIFO
  - Simple I/O Merging
  - Lowest CPU Cost

Iozone Performance Effect of TUNED EXT4/XFS/GFS

RHEL7 RC 3.10-111 File System In Cache Performance



RHEL7 3.10-111 File System Out of Cache Performance



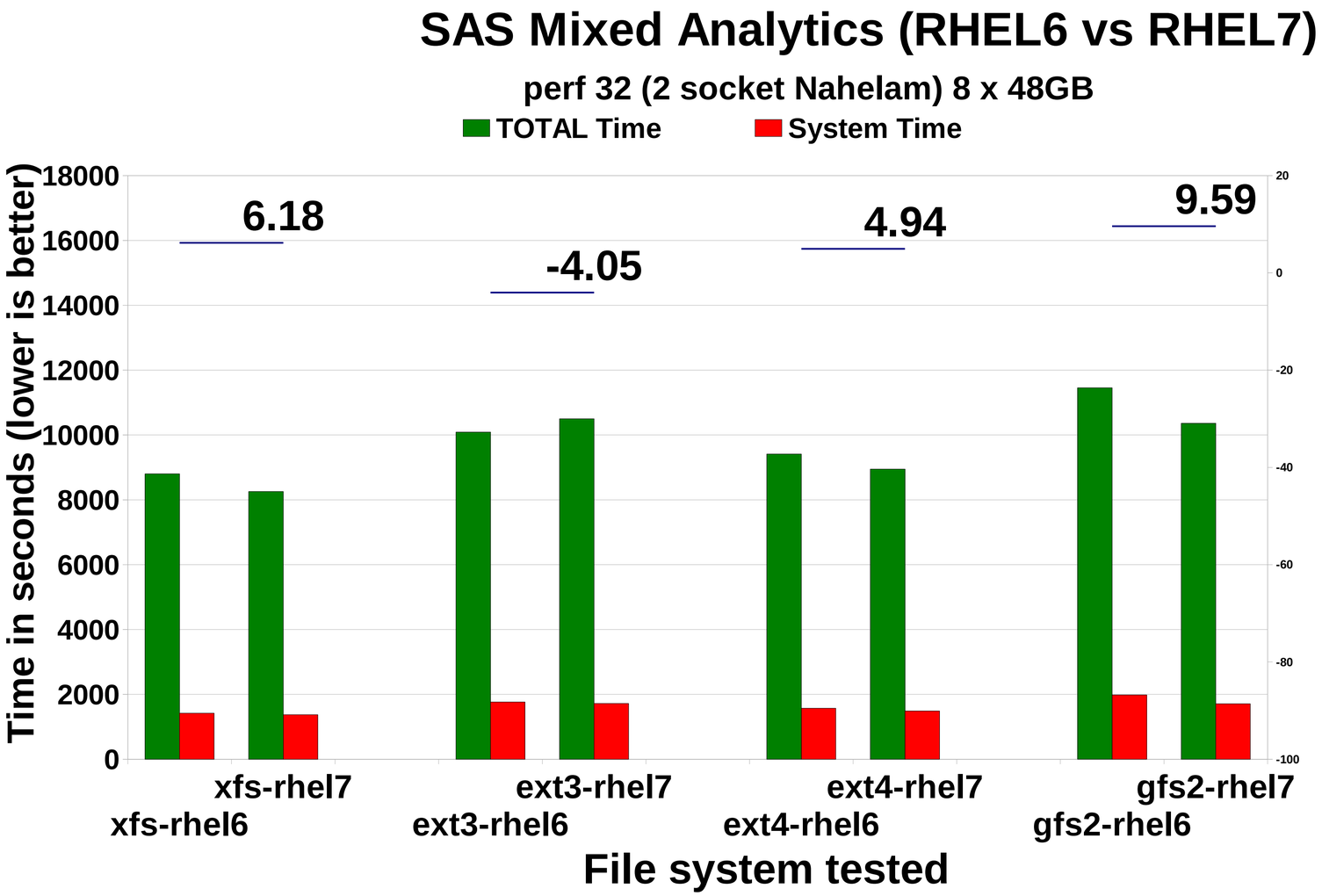
SAS Application on Standalone Systems

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



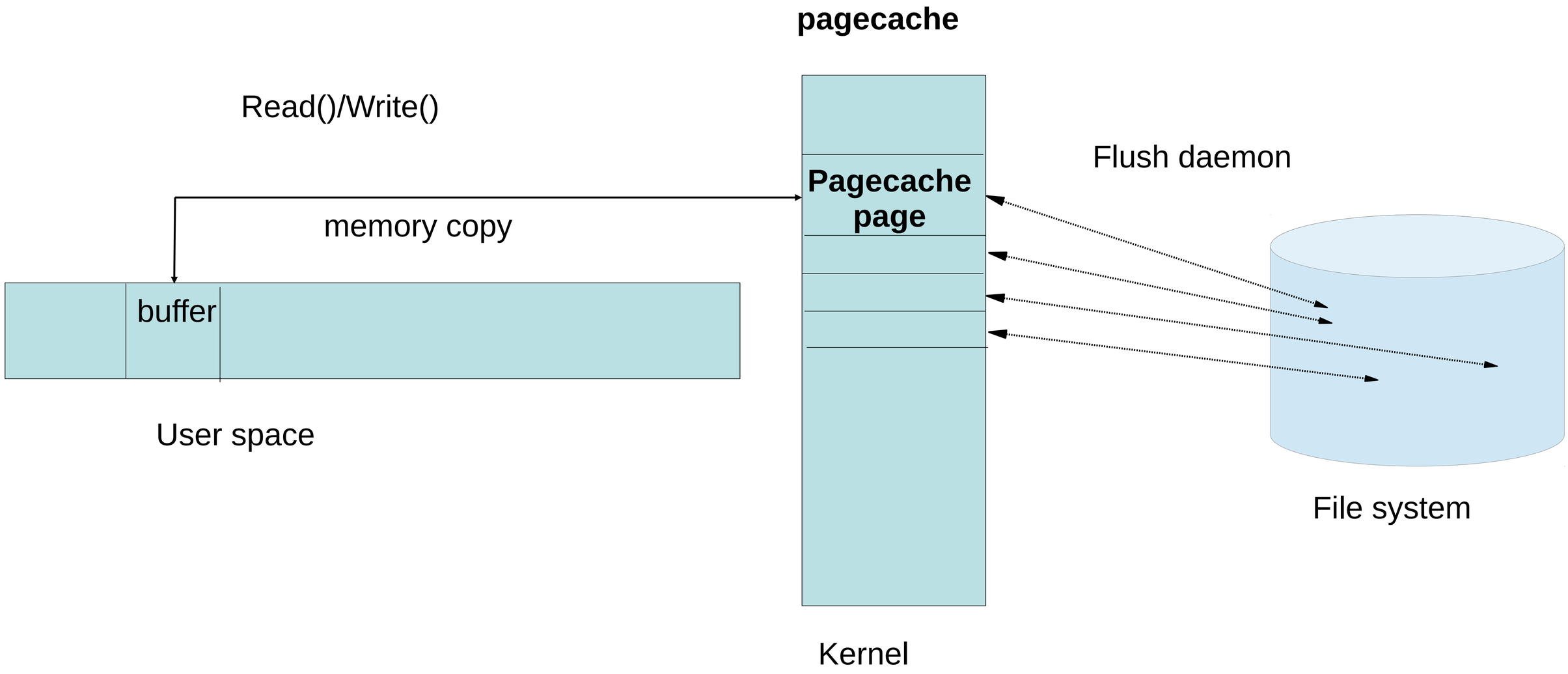
Tuning Memory – Flushing Caches

- 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

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

Per file system flush daemon



swappiness

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

vfs_cache_pressure

- Controls how aggressively the kernel reclaims memory in slab caches.
- Increasing causes the system to reclaim inode cache and dentry cache.
- Decreasing causes inode cache and dentry cache to grow.

min_free_kbytes

Directly controls the page reclaim watermarks in KB

Defaults are higher when THP is enabled

```
# echo 1024 > /proc/sys/vm/min_free_kbytes
----------------------------------------------------------
Node 0 DMA free:4420kB min:8kB low:8kB high:12kB
Node 0 DMA32 free:14456kB min:1012kB low:1264kB high:1516kB
----------------------------------------------------------

echo 2048 > /proc/sys/vm/min_free_kbytes
----------------------------------------------------------
Node 0 DMA free:4420kB min:20kB low:24kB high:28kB
Node 0 DMA32 free:14456kB min:2024kB low:2528kB high:3036kB
----------------------------------------------------------
```

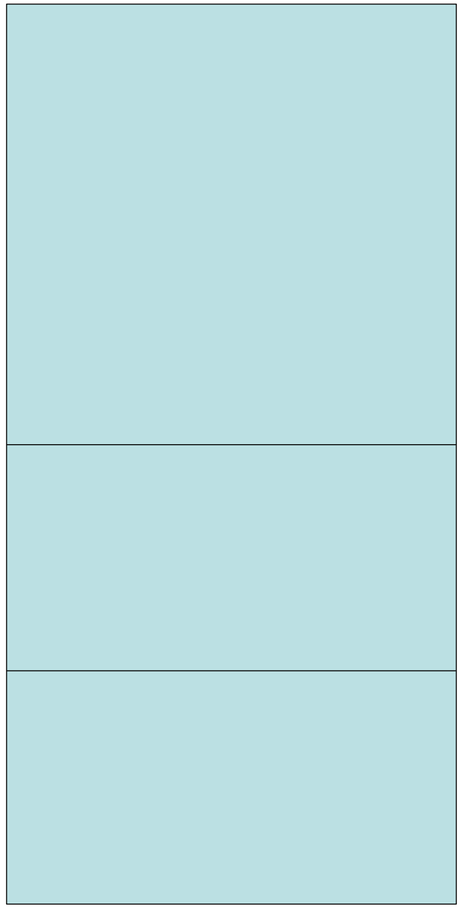
dirty_background_ratio, dirty_background_bytes

- Controls when dirty pagecache memory starts getting written.
- Default is 10%
- Lower
  - flushing starts earlier
  - less dirty pagecache and smaller IO streams
- Higher
  - flushing starts later
  - more dirty pagecache and larger IO streams
- **dirty_background_bytes over-rides when you want < 1%**

dirty_ratio, dirty_bytes

- Absolute limit to percentage of dirty pagecache memory
- Default is 20%
- Lower means clean pagecache and smaller IO streams
- Higher means dirty pagecache and larger IO streams
- dirty_bytes overrides when you want < 1%

dirty_ratio and dirty_background_ratio



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

RED HAT SUMMIT

10 YEARS *and counting*

SAN FRANCISCO | APRIL 14-17, 2014

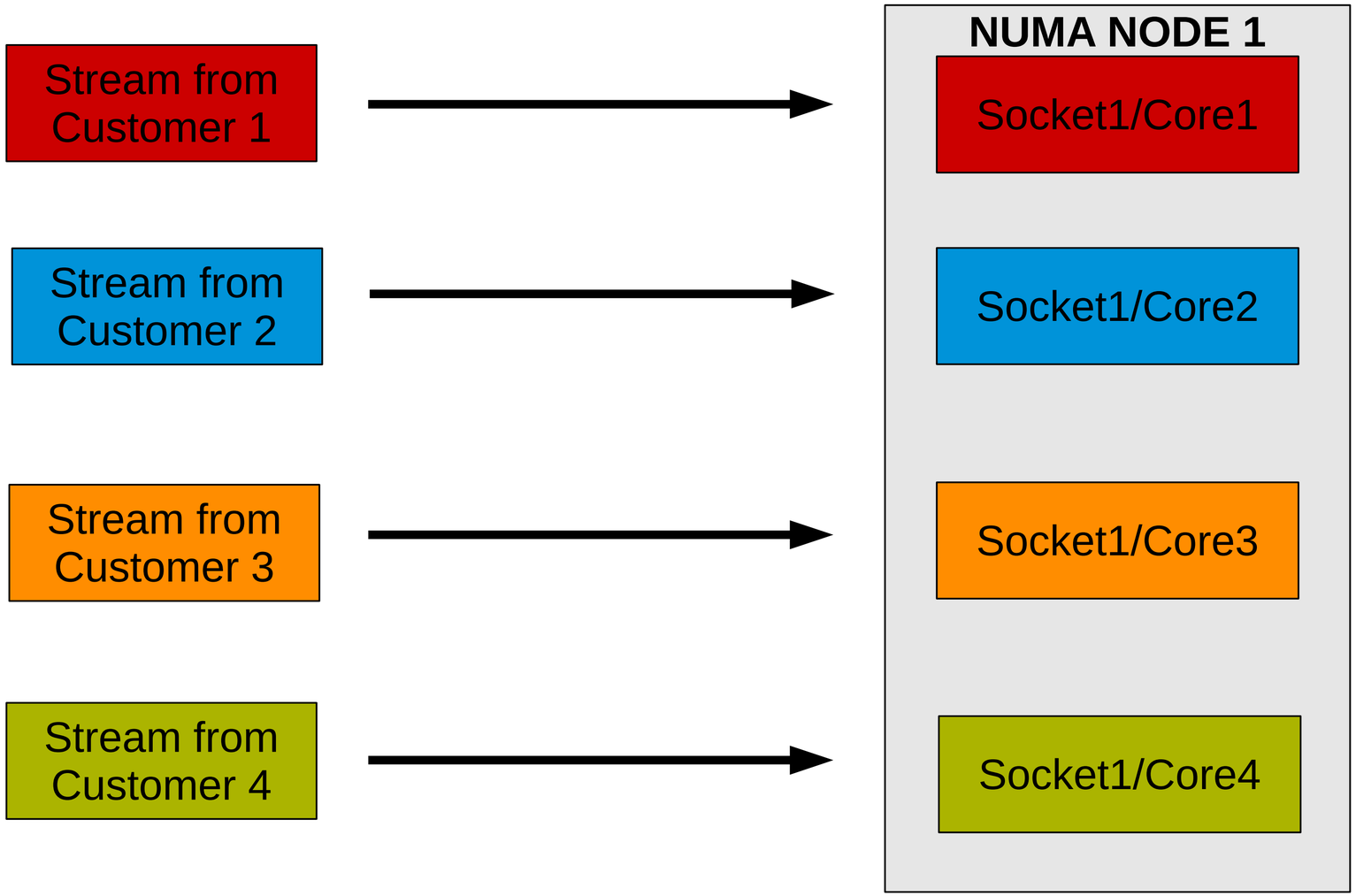
Network Performance Tuning

redhat.

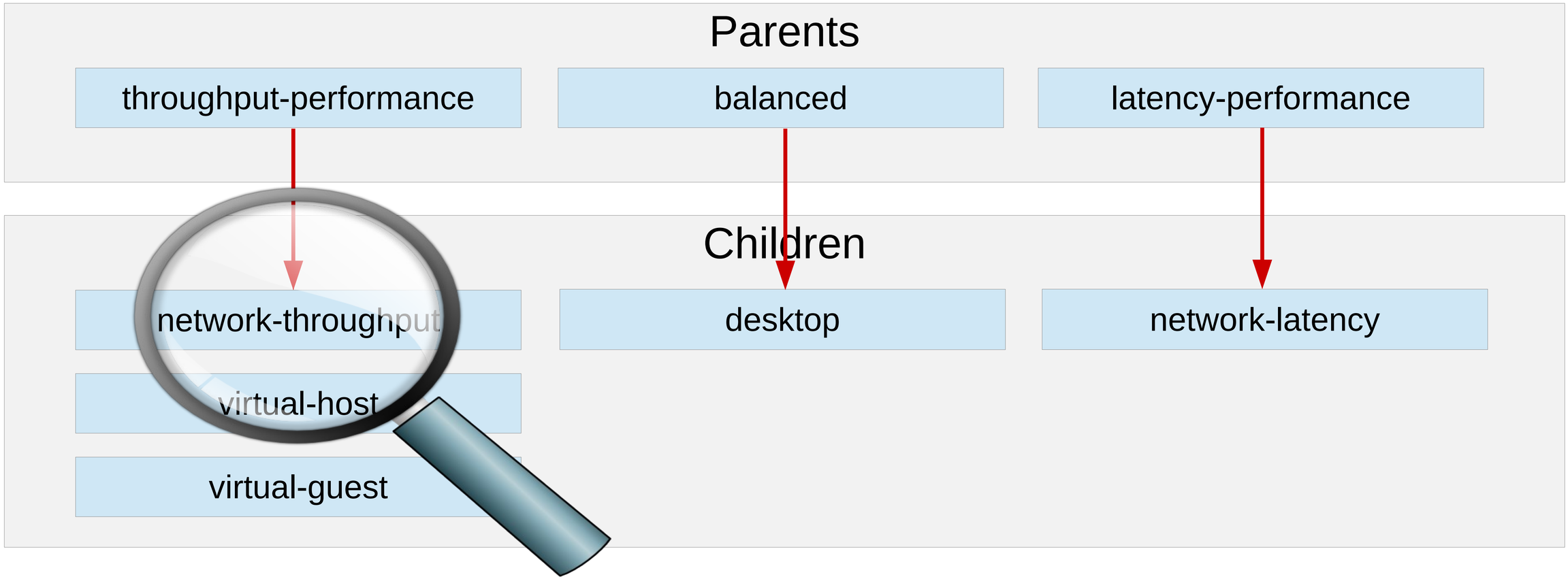
RHEL7 Networks

- IPv4 Routing Cache replaced with Forwarding Information Base
  - Better scalability, determinism and security
- Socket BUSY_POLL (aka low latency sockets)
- 40G NIC support, bottleneck moves back to CPU :-)
- VXLAN Offload (for OpenStack)
- NetworkManager: nmcli and nmtui

Locality of Packets



Tuned: Profile Inheritance



Tuned: Profile Inheritance (throughput)

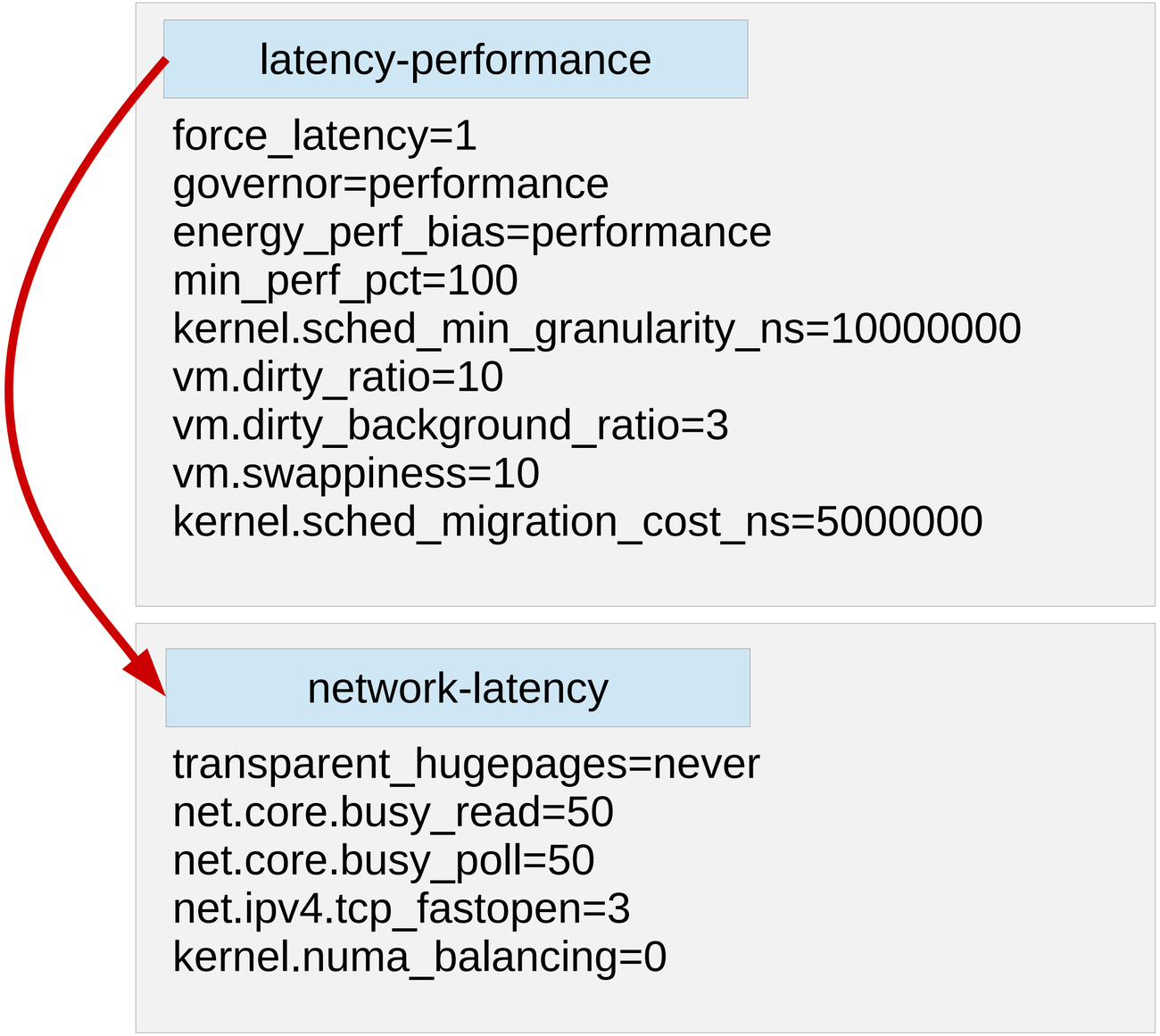
throughput-performance

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 (latency)



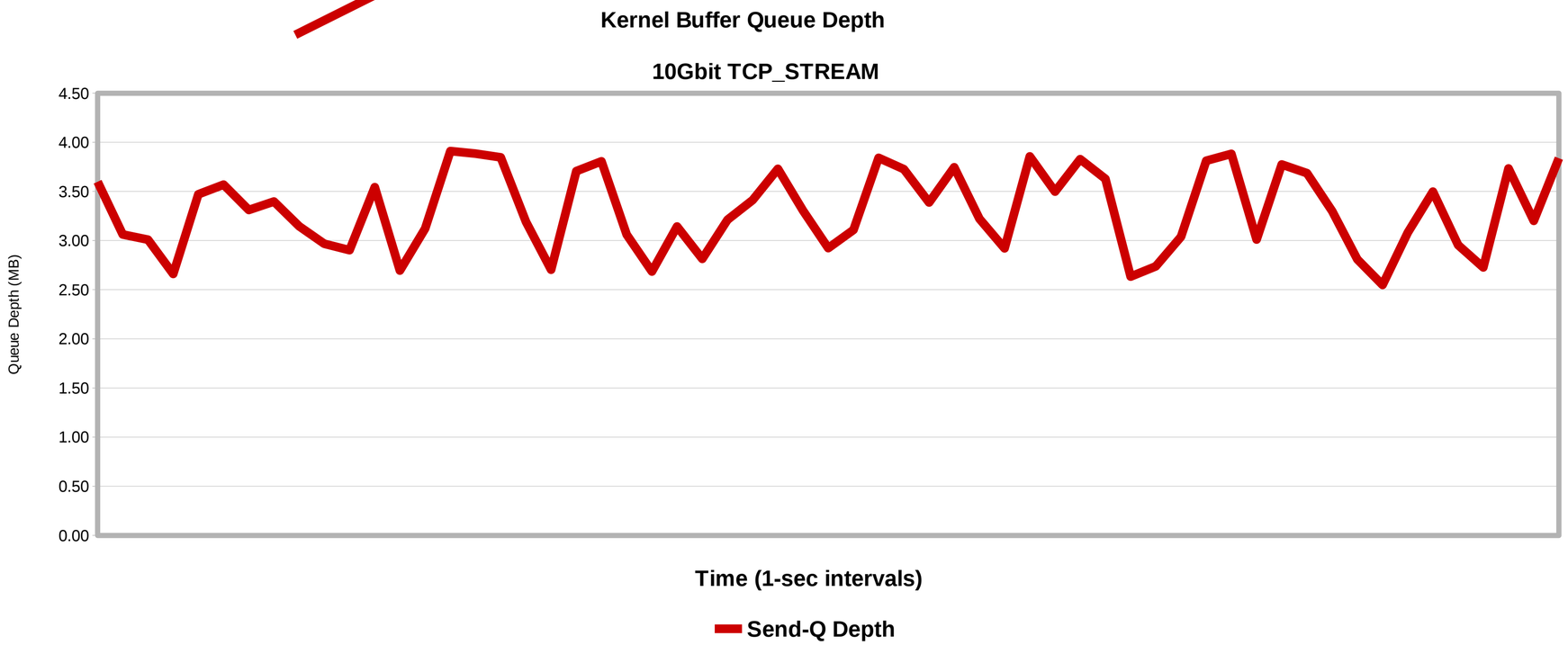
Networking performance – System setup

- 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
  - NIC Offloads favor throughput

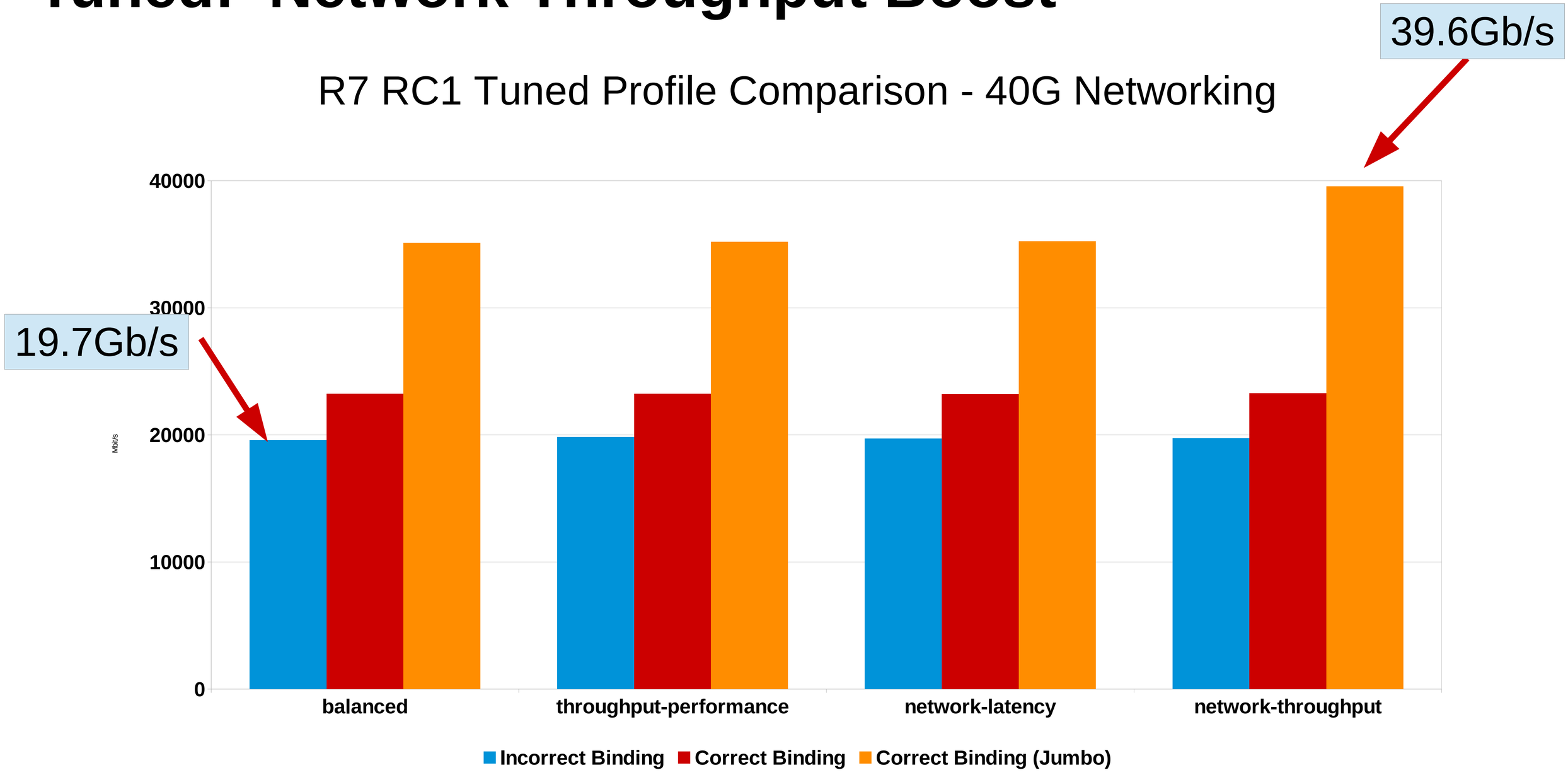
Network Tuning: Buffer Bloat

```
# 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

Tuned: Network Throughput Boost



redhat.

netsniff-ng: ifpps

- Aggregate network stats to one screen
- Can output to .csv

```
3.10.0-119.el7.x86_64, sfc1 (sfc 10000Mbit/s link:yes), t=1000ms, cpus=5+1/16
(consider to increase your sampling interval, e.g. -t 10000)
rx:          107.498 MiB/t     479466 pkts/t          0 drops/t          0 errors/t
tx:          107.147 MiB/t     491346 pkts/t          0 drops/t          0 errors/t

rx:       101161.313 MiB    477098891 pkts            0 drops            0 errors
tx:       102582.672 MiB    507362796 pkts            0 drops            0 errors

sys:                            317 procs            5 running          0 iowait

mem:         128727M total      3193M used        1438M active        392M inactive
swap:          4095M total         0M used           0M cached

cpu 1 +:        28.7% usr/t      71.3% sys/t        0.0% idl/t         0.0% iow/t
cpu 3 |:        17.8% usr/t      82.2% sys/t        0.0% idl/t         0.0% iow/t
cpu 9 |:       100.0% usr/t       0.0% sys/t        0.0% idl/t         0.0% iow/t
cpu11 |:       100.0% usr/t       0.0% sys/t        0.0% idl/t         0.0% iow/t
cpu 0 |:         0.0% usr/t       0.0% sys/t      100.0% idl/t         0.0% iow/t
cpu15 -:         0.0% usr/t       0.0% sys/t      100.0% idl/t         0.0% iow/t
avg:            16.3%             9.8%              73.9%               0.0%

cpu15 +:       318764 irqs/t             13574 sirq rx/t                16 sirq tx/t
cpu 1 |:            0 irqs/t                 7 sirq rx/t                 4 sirq tx/t
cpu 3 |:            0 irqs/t                 0 sirq rx/t                 0 sirq tx/t
cpu 9 |:            0 irqs/t                 0 sirq rx/t                 0 sirq tx/t
cpu11 |:            0 irqs/t                 0 sirq rx/t                 0 sirq tx/t
cpu14 -:            0 irqs/t                 0 sirq rx/t                 0 sirq tx/t
avg:            19922.8                   848.9                         1.2

cpu15 +:    617304867 irqs
cpu14 |:    151434664 irqs
cpu 1 |:         4847 irqs
cpu 3 |:            0 irqs
cpu 9 |:            0 irqs
cpu13 -:            0 irqs
avg:         48046523.6
```

Pkts/sec

netsniff-ng: ifpps

- Aggregate network stats to one screen
- Can output to .csv

```
3.10.0-119.el7.x86_64, sfc1 (sfc 10000Mbit/s link:yes), t=1000ms, cpus=5+1/16
(consider to increase your sampling interval, e.g. -t 10000)
rx:          107.498 MiB/t     479466 pkts/t          0 drops/t          0 errors/t
tx:          107.147 MiB/t     491346 pkts/t          0 drops/t          0 errors/t

rx:       101161.313 MiB    477098891 pkts            0 drops            0 errors
tx:       102582.672 MiB    507362796 pkts            0 drops            0 errors

sys:                                                  5 running          0 iowait

mem:         128727M total      3193M used        1438M active        392M inactive
swap:          4095M total         0M used           0M cached

cpu 1 +:        28.7% usr/t      71.3% sys/t        0.0% idl/t         0.0% iow/t
cpu 3 |:        17.8% usr/t      82.2% sys/t        0.0% idl/t         0.0% iow/t
cpu 9 |:       100.0% usr/t       0.0% sys/t        0.0% idl/t         0.0% iow/t
cpu11 |:       100.0% usr/t       0.0% sys/t        0.0% idl/t         0.0% iow/t
cpu 0 |:         0.0% usr/t       0.0% sys/t      100.0% idl/t         0.0% iow/t
cpu15 -:         0.0% usr/t       0.0% sys/t      100.0% idl/t         0.0% iow/t
avg:            16.3%             9.8%             73.9%               0.0%

cpu15 +:       318764 irqs/t            13574 sirq rx/t                16 sirq tx/t
cpu 1 |:            0 irqs/t                7 sirq rx/t                 4 sirq tx/t
cpu 3 |:            0 irqs/t                0 sirq rx/t                 0 sirq tx/t
cpu 9 |:            0 irqs/t                0 sirq rx/t                 0 sirq tx/t
cpu11 |:            0 irqs/t                0 sirq rx/t                 0 sirq tx/t
cpu14 -:            0 irqs/t                0 sirq rx/t                 0 sirq tx/t
avg:            19922.8                   848.9                        1.2

cpu15 +:    617304867 irqs
cpu14 |:    151434664 irqs
cpu 1 |:         4847 irqs
cpu 3 |:            0 irqs
cpu 9 |:            0 irqs
cpu13 -:            0 irqs
avg:         48046523.6
```

Pkts/sec

Drops/sec

netsniff-ng: ifpps

- Aggregate network stats to one screen
- Can output to .csv

```
3.10.0-119.el7.x86_64, sfc1 (sfc 10000Mbit/s link:yes), t=1000ms, cpus=5+1/16
(consider to increase your sampling interval, e.g. -t 10000)
rx:          107.498 MiB/t     479466 pkts/t          0 drops/t          0 errors/t
tx:          107.147 MiB/t     491346 pkts/t          0 drops/t          0 errors/t

rx:       101161.313 MiB    477098891 pkts            0 drops            0 errors
tx:       102582.672 MiB    507362796 pkts            0 drops            0 errors

sys:                                                  5 running          0 iowait

mem:         128727M total      3193M used        1438M active        392M inactive
swap:          4095M total         0M used           0M cached

cpu 1 +:        28.7% usr/t      71.3% sys/t        0.0% idl/t         0.0% iow/t
cpu 3 |:        17.8% usr/t      82.2% sys/t        0.0% idl/t         0.0% iow/t
cpu 9 |:       100.0% usr/t       0.0% sys/t        0.0% idl/t         0.0% iow/t
cpu11 |:       100.0% usr/t       0.0% sys/t        0.0% idl/t         0.0% iow/t
cpu 0 |:         0.0% usr/t       0.0% sys/t      100.0% idl/t         0.0% iow/t
cpu15 -:         0.0% usr/t       0.0% sys/t      100.0% idl/t         0.0% iow/t
avg:            16.3%             9.8%             73.9%               0.0%

cpu15 +:       318764 irqs/t            13574 sirq rx/t                16 sirq tx/t
cpu 1 |:            0 irqs/t                7 sirq rx/t                 4 sirq tx/t
cpu 3 |:            0 irqs/t                0 sirq rx/t                 0 sirq tx/t
cpu 9 |:            0 irqs/t                0 sirq rx/t                 0 sirq tx/t
cpu11 |:            0 irqs/t                0 sirq rx/t                 0 sirq tx/t
cpu14 -:            0 irqs/t                0 sirq rx/t                 0 sirq tx/t
avg:            19922.8                   848.9                         1.2

cpu15 +:    617304867 irqs
cpu14 |:    151434664 irqs
cpu 1 |:         4847 irqs
cpu 3 |:            0 irqs
cpu 9 |:            0 irqs
cpu13 -:            0 irqs
avg:         48046523.6
```

Pkts/sec

Drops/sec

Hard/Soft IRQs/sec

Network Tuning: Low Latency TCP

- set TCP_NODELAY (Nagle)
- Experiment with ethtool offloads
- tcp_low_latency tiny substantive benefit found
- Ensure kernel buffers are “right-sized”
  - Use ss (Recv-Q Send-Q)
  - Don't setsockopt unless you've really tested
- Review old code to see if you're using setsockopt
  - Might be hurting performance

Network Tuning: Low Latency UDP

- Mainly about managing bursts, avoiding drops
  - rmem_max/wmem_max
- TX
  - netdev_max_backlog
  - txqueuelen
- RX
  - netdev_max_backlog
  - ethtool -g
  - ethtool -c
  - netdev_budget
- Dropwatch tool in RHEL

Full DynTicks (nohz_full)

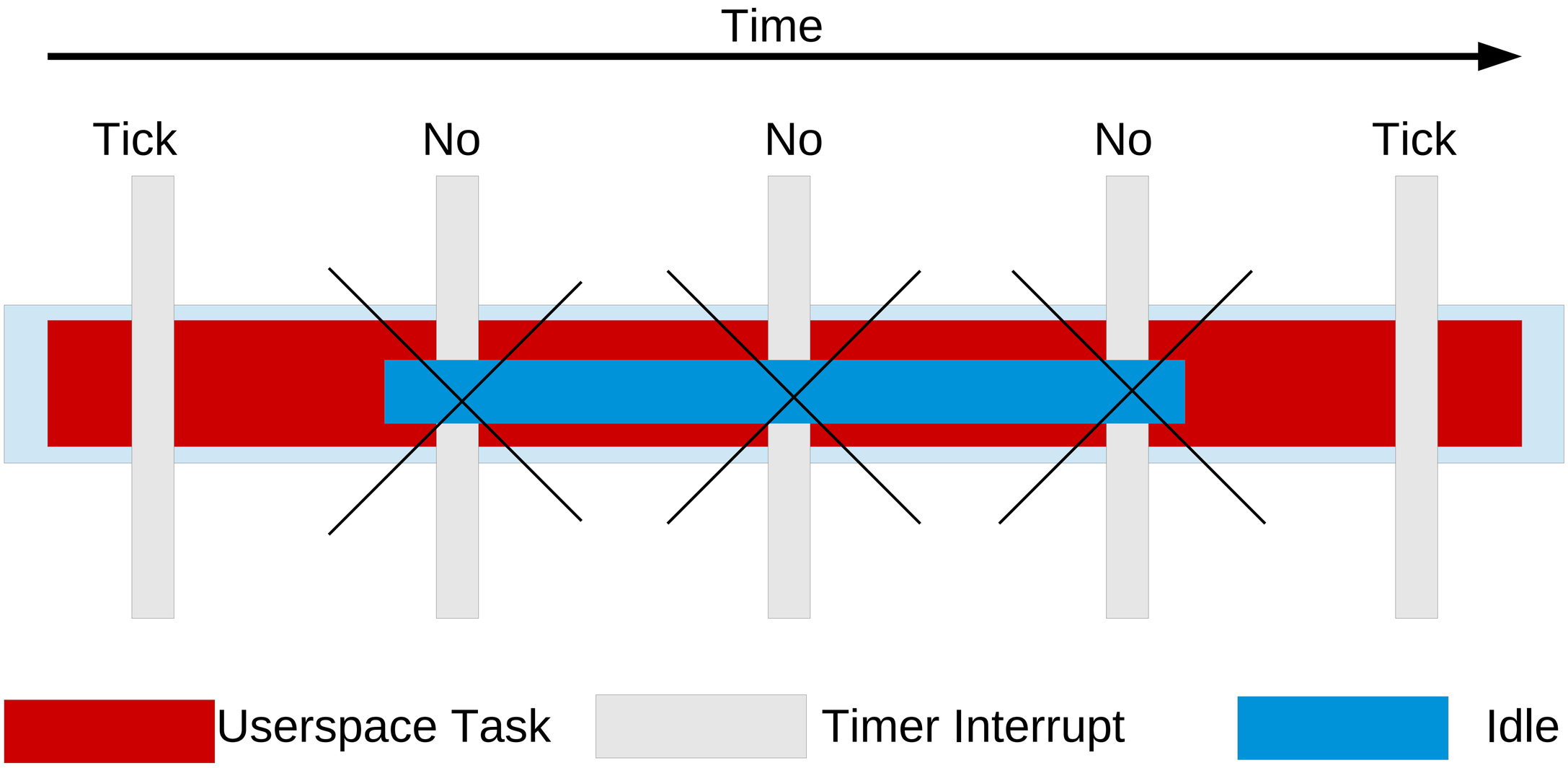
Full DynTicks Patchset

- 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

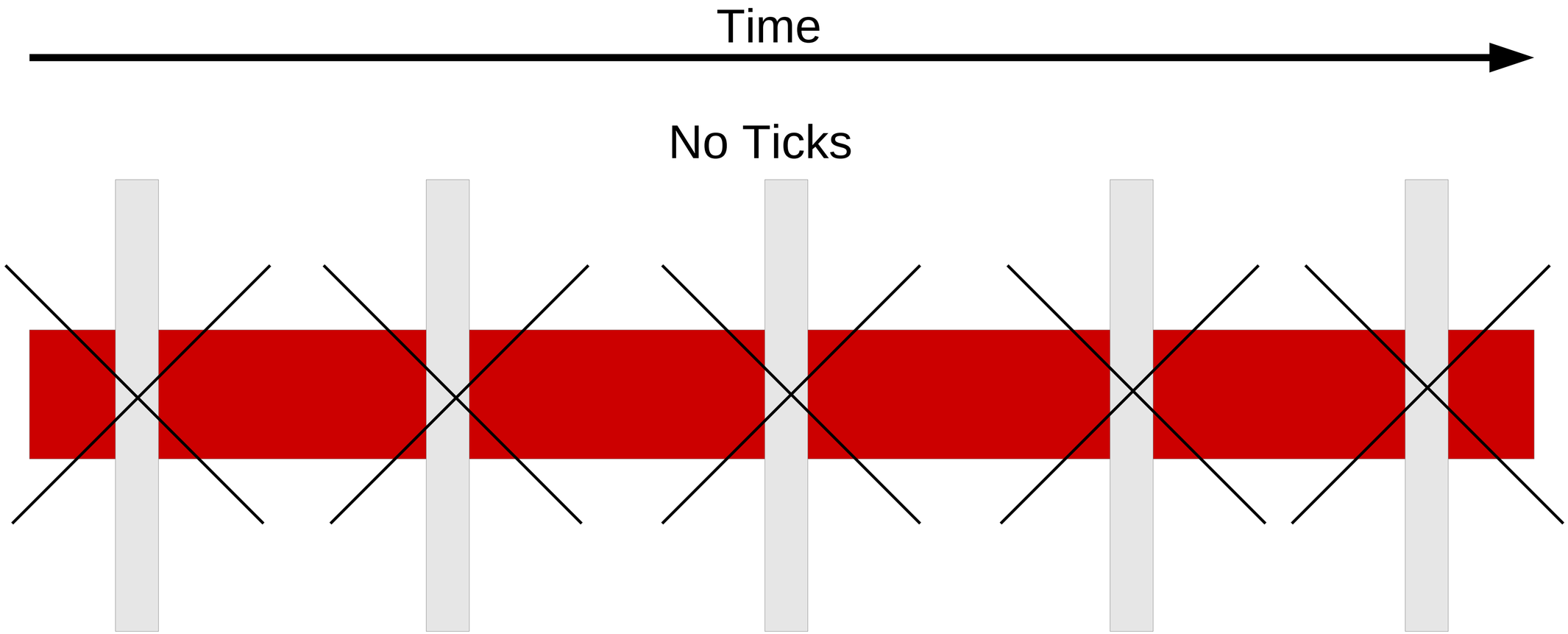
Kernel Tick:

- timekeeping (gettimeofday)
- Scheduler load balancing
- Memory statistics (vmstat)

RHEL6 and 7 Tickless



nohz_full



Userspace Task

Timer Interrupt

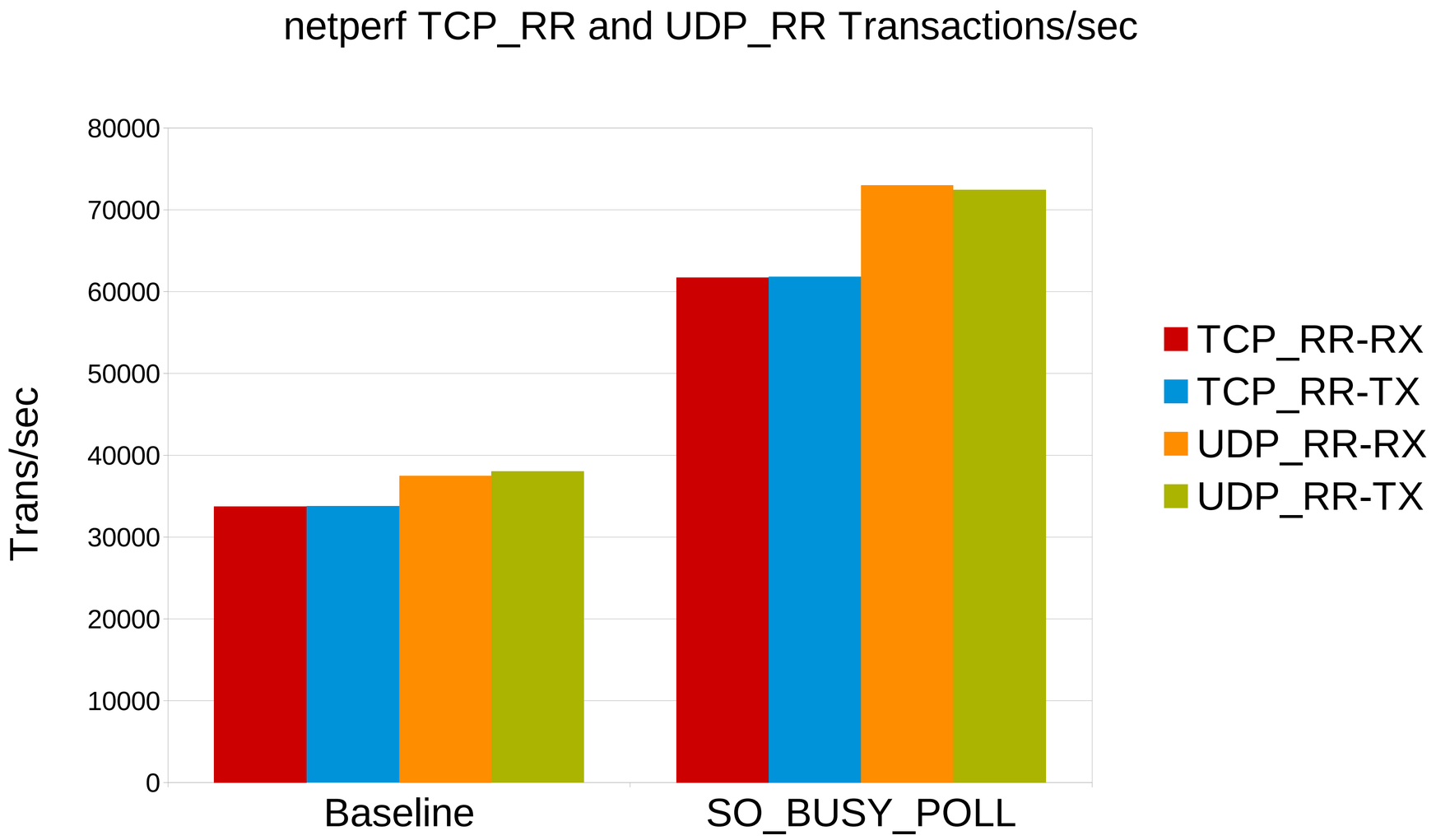
Tickless doesn't require idle...

Busy Polling

SO_BUSY_POLL Socket Option

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

BUSY_POLL Socket Option



Power Management

Power Management: P-states and C-states

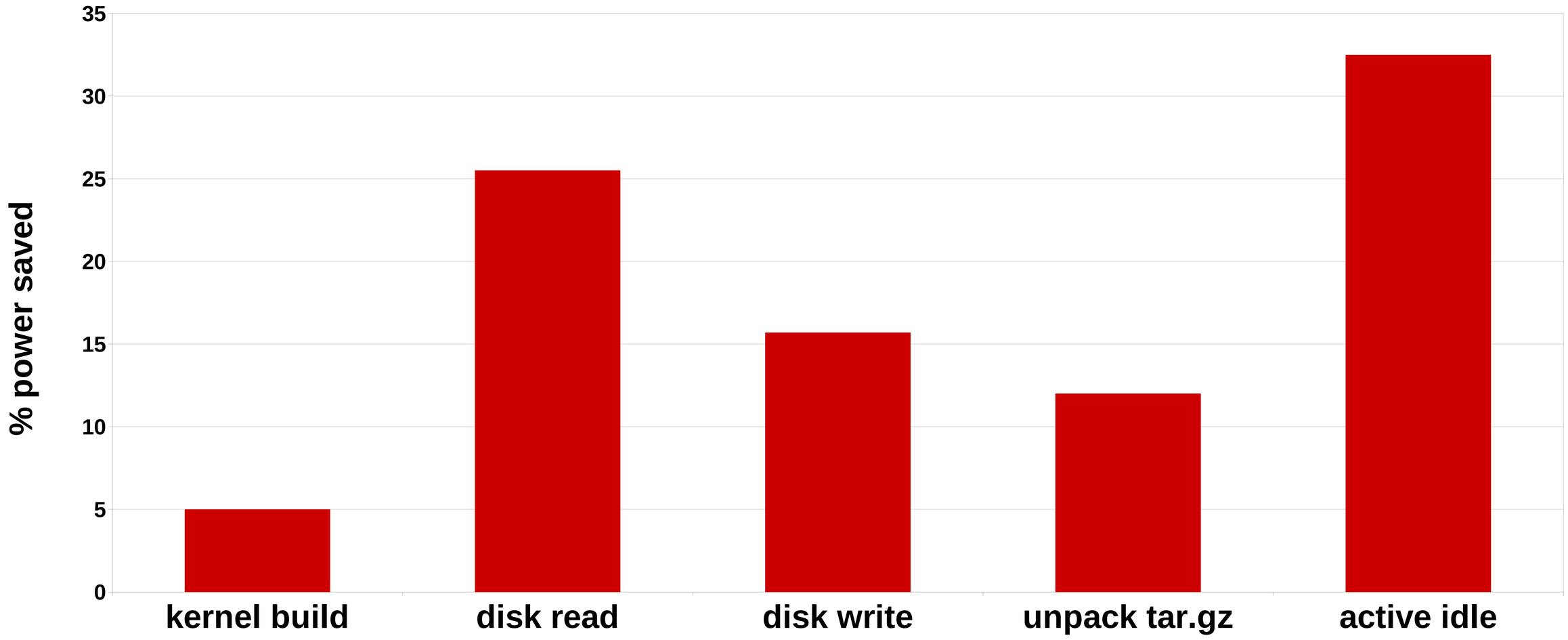
- P-state: CPU Frequency
  - Governors, Frequency scaling
- C-state: CPU Idle State
  - Idle drivers

Introducing intel_pstate P-state Driver

- New Default Idle Driver in RHEL7: intel_pstate (not a module)
  - CPU governors replaced with sysfs min_perf_pct and max_perf_pct
  - Moves Turbo knob into OS control (yay!)

- Tuned handles most of this for you:
  - Sets min_perf_pct=100 for most profiles
  - Sets x86_energy_perf_policy=performance (same as RHEL6)

Impact of CPU Idle Drives (watts per workload) RHEL7 @ C1



Turbostat shows P/C-states on Intel CPUs

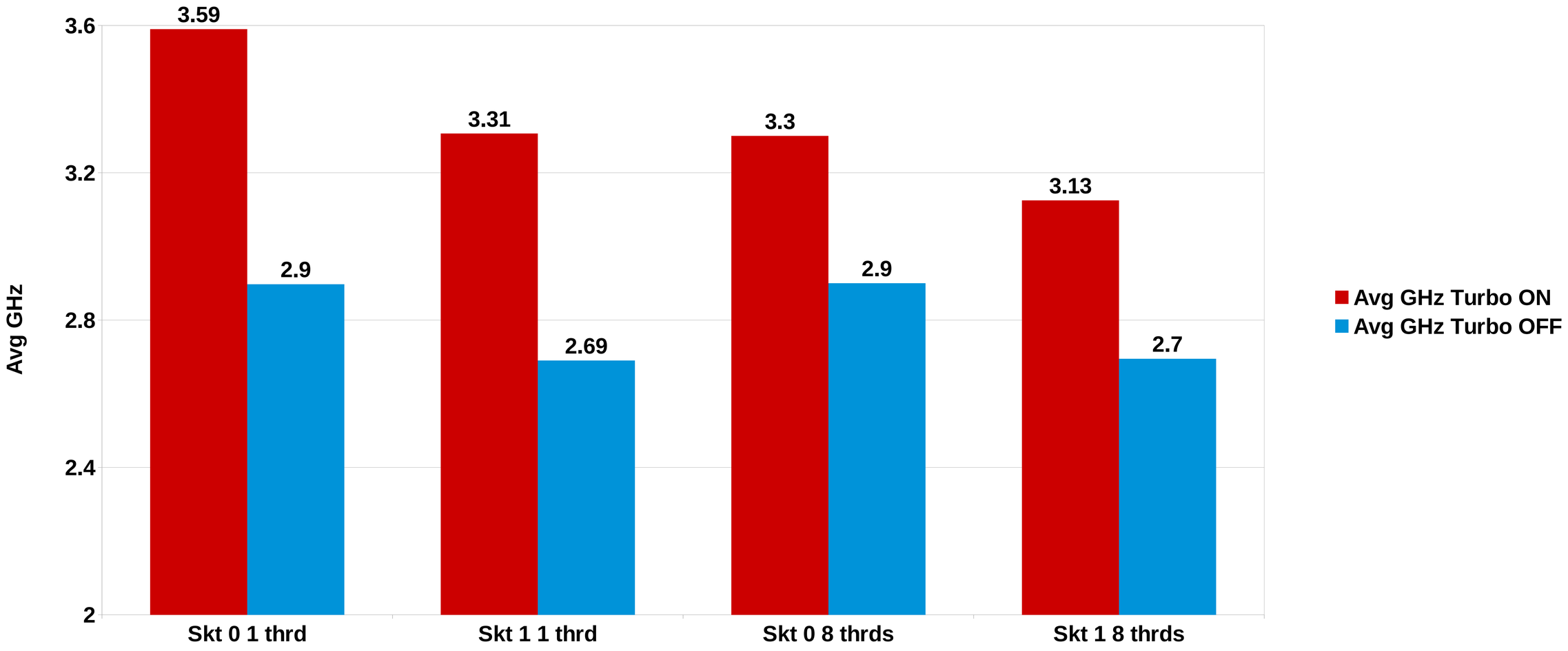
Default

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

Frequency Scaling (Turbo) Varying Load



RED HAT SUMMIT

10 YEARS *and counting*

SAN FRANCISCO | APRIL 14-17, 2014

Analysis Tools

Performance Co-Pilot

redhat.

Performance Co-Pilot (PCP)

(Multi) system-level performance monitoring and management

pmchart – graphical metric plotting tool

- Can plot myriad performance statistics

pmchart – graphical metric plotting tool

- Can plot myriad performance statistics
- Recording mode allows for replay
  - i.e. on a different system
  - Record in GUI, then

```
# pmafm $recording.folio
```

pmchart – graphical metric plotting tool

- Can plot myriad performance statistics
- Recording mode allows for replay
  - i.e. on a different system
  - Record in GUI, then

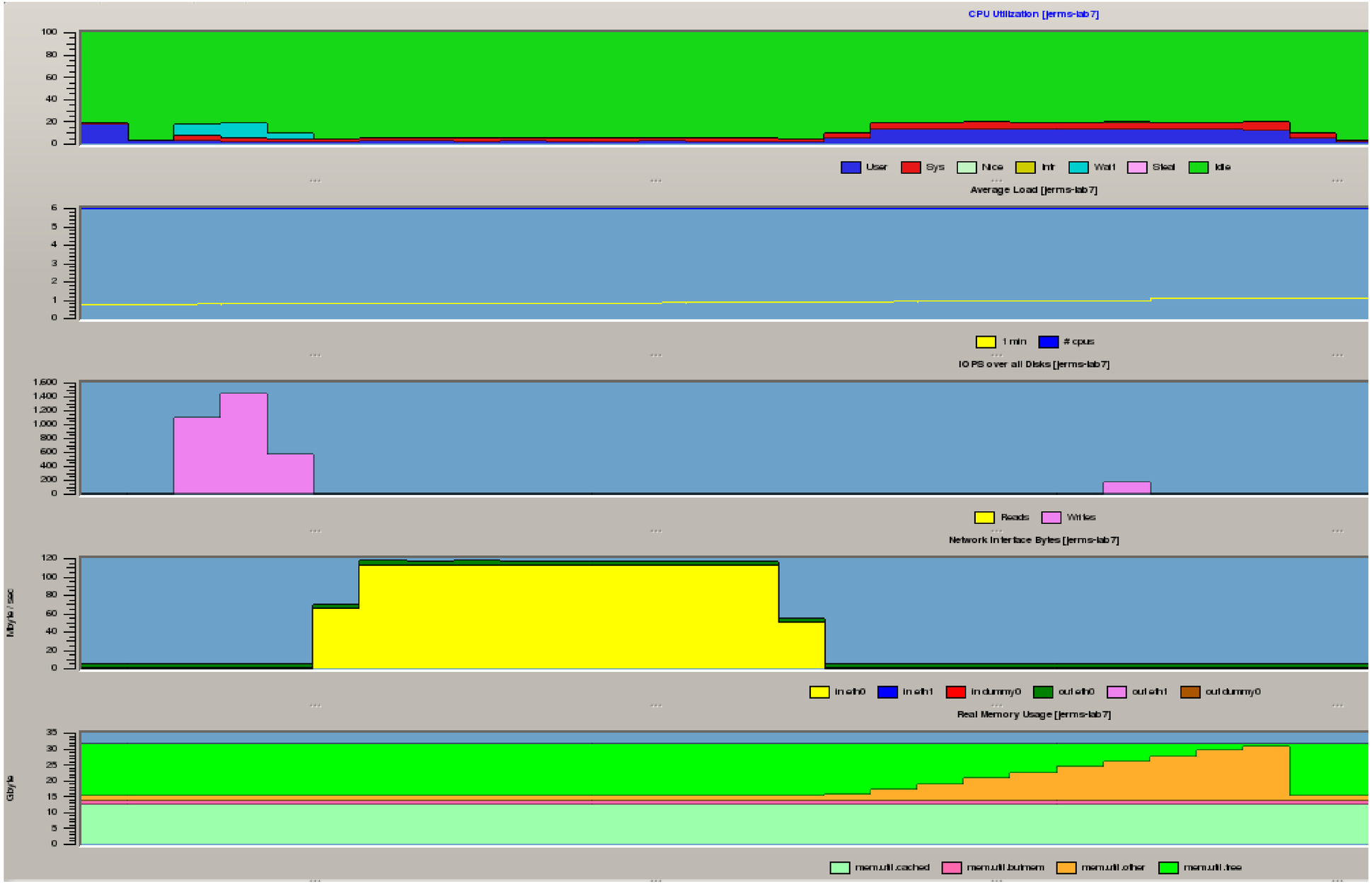
```
# pmafm $recording.folio
```

- Ships with many pre-cooked “views”...for example:
  - ApacheServers: CPU%/Net/Busy/Idle Apache Servers
  - Overview: CPU%/Load/IOPS/Net/Memory

Performance Co-Pilot Demo Script

- Tiny script to exercise 4 food groups...

```
CPU
# stress -t 5 -c 1
DISK
# dd if=/dev/zero of=/root/2GB count=2048 bs=1M oflag=direct
NETWORK
netperf -H rhel7.lab -l 5
# MEMORY
# stress -t 5 --vm 1 –vm-bytes 16G
```



CPU %

Load Avg

IOPS

Network

Memory Allocated

pmcollectl mode

```
#<--------CPU--------> <----------Disks-----------> <----------Network----------> #<--
#cpu sys inter  ctxsw  KBRea KBWrit Writes  KBIn  PktIn  KBOut  PktOut #Free
   0   0   210    179      0      0     64     18     2     17      0       1 32355M
   0   0   202    150      0      0     32     10     1     14      0       1 32355M
   4   1  1678   2073   6876    650    108     14    10     50      5      33 32346M
  17   0  2348    183      0      0     36     10     2     14      0       3 32346M
  17   0  2361    216      0      0     32     10     1     17      0       1 32346M
   7   1  1760   1629    272     20  88356    282    11     63      6      46 32345M
   3   2  1691   2526     40     10 795720   2336     0     11      0       2 32344M
   3   2  1875   2856     28      7 924736   2714     2     18      0       3 32344M
   2   1  5137   5383    460     40 288836    851 35127   2583    161    2473 32345M
   4   3 16997  28627      0      0     56     10 245172  17629   1101   17088 32344M
   3   2 15619  28062      0      0     44     12 242954  17508   1087   16871 32345M
   6   2  4495   7098    104      3     80      9 51692   3781    240    3675 31804M
  17   5  2380    187      0      0     20      5     1     12      0       3 28287M
  17   5  2349    188      0      0     52     15     1     13      0       1 24805M
  17   5  2356    214      0      0     32     10     2     16      0       1 21284M
  17   5  2348    197      0      0     32     10     0      9      0       1 17436M
   9   3  1366    225      0      0     32     10     2     20      0       4 24766M
   1   0   465    516      8      2    992    169     2     25      1      15 32344M
```

CPU

pmcollectl mode

```
#<--------CPU--------> <----------Disks-----------> <----------Network----------> #<--
#cpu sys inter  ctxsw  KBRea  KBWrit Writes  KBIn  PktIn  KBOut  PktOut #Free
   0   0   210    179      0      0     64     18     17      0       1 32355M
   0   0   202    150      0      0     32     10      1     14       0       1 32355M
   4   1  1678   2073   6876    650    108     14     10     50       5      33 32346M
  17   0  2348    183      0      0     36     10      2     14       0       3 32346M
  17   0  2361    216      0      0     32     10      1     17       0       1 32346M
   7   1  1760   1629    272     20  88356    282     11     63       6      46 32345M
   3   2  1691   2526     40     10 795720   2336      0     11       0       2 32344M
   3   2  1875   2856     28      7 924736   2714      2     18       0       3 32344M
   2   1  5137   5383    460     40 288836    851  35127   2583     161    2473 32345M
   4   3 16997  28627      0      0     56     10 245172  17629    1101   17088 32344M
   3   2 15619  28062      0      0     44     12 242954  17508    1087   16871 32345M
   6   2  4495   7098    104      3     80      9  51692   3781     240    3675 31804M
  17   5  2380    187      0      0     20      5      1     12       0       3 28287M
  17   5  2349    188      0      0     52     15      1     13       0       1 24805M
  17   5  2356    214      0      0     32     10      2     16       0       1 21284M
  17   5  2348    197      0      0     32     10      0      9       0       1 17436M
   9   3  1366    225      0      0     32     10      2     20       0       4 24766M
   1   0   465    516      8      2    992    169      2     25       1      15 32344M
```

CPU

IOPS

pmcollectl mode

```
#<--------CPU--------> <----------Disks-----------> <----------Network----------> #<--
#cpu sys inter  ctxsw  KBRea KBWrit Writes  KBIn  PktIn  KBOut  PktOut #Free
   0   0   210    179      0      0     64    18     17      0       1 32355M
   0   0   202    150      0      0     32    10      1     14       0       1 32355M
   4   1  1678   2073   6876    650    108    14     10     50       5      33 32346M
  17   0  2348    183      0      0     36    10      2     14       0       3 32346M
  17   0  2361    216      0      0     32    10      1     17       0       1 32346M
   7   1  1760   1629    272     20  88356   282     11     63       6       2 32345M
   3   2  1691   2526     40     10 795720  2336      0     11       0       2 32344M
   3   2  1875   2856     28      7 924736  2714      2     18       0       3 32344M
   2   1  5137   5383    460     40 288836   851  35127   2583     161    2473 32345M
   4   3 16997  28627      0      0     56    10 245172  17629    1101   17088 32344M
   3   2 15619  28062      0      0     44    12 242954  17508    1087   16871 32345M
   6   2  4495   7098    104      3     80     9  51692   3781     240    3675 31804M
  17   5  2380    187      0      0     20     5      1     12       0       3 28287M
  17   5  2349    188      0      0     52    15      1     13       0       1 24805M
  17   5  2356    214      0      0     32    10      2     16       0       1 21284M
  17   5  2348    197      0      0     32    10      0      9       0       1 17436M
   9   3  1366    225      0      0     32    10      2     20       0       4 24766M
   1   0   465    516      8      2    992   169      2     25       1      15 32344M
```

CPU

IOPS

NET

pmcollectl mode

```
#<--------CPU--------> <----------Disks-----------> <----------Network----------> #<--
#cpu sys inter  ctxsw  KBRea KBWrit Writes  KBIn  PktIn  KBOut  PktOut #Free
   0   0   210    179      0      0     64    18     17      0       1 32355M
   0   0   202    150      0      0     32    10      1     14       0       1 32355M
   4   1  1678   2073   6876    650    108    14     10     50       5      33 32346M
  17   0  2348    183      0      0     36    10      2     14       0       3 32346M
  17   0  2361    216      0      0     32    10      1     17       0       1 32346M
   7   1  1760   1629    272     20  88356   282     11     63       6       2345M
   3   2  1691   2526     40     10 795720  2336      0     11       0       2344M
   3   2  1875   2856     28      7 924736  2714      2     18       0       3 32344M
   2   1  5137   5383    460     40 288836   851  35127   2583     161    2473 32345M
   4   3 16997  28627      0      0     56    10 245172  17629    1101   17088 32344M
   3   2 15619  28062      0      0     44    12 242954  17508    1087   16871 32345M
   6   2  4495   7098    104      3     80     9  51692   3781     240    3675 31804M
  17   5  2380    187      0      0     20     5      1     12       0       3 28287M
  17   5  2349    188      0      0     52    15      1      1       0       1 24805M
  17   5  2356    214      0      0     32    10      2      1       0       1 21284M
  17   5  2348    197      0      0     32    10      0      9       0       1 17436M
   9   3  1366    225      0      0     32    10      2     20       0       4 24766M
   1   0   465    516      8      2    992   169      2     25       1      15 32344M
```

CPU

IOPS

NET

MEM

pmatop mode

```
ATOP - Mon Apr  7 08:15:04 2014          0:00:05 elapsed

PRC | sys    4.16s | user  16.75s | #proc    332 | #tslpi    37 | #tslpu     0 | #zombie    0
CPU | sys       5% | user     21% | irq       0% | idle     73% | wait      0% |
cpu | sys       0% | user      6% | irq       0% | idle      0% | cpu01     0% | curf  2.9MHz
cpu | sys       5% | user      1% | irq       0% | idle      0% | cpu03     0% | curf  2.9MHz
cpu | sys       0% | user      6% | irq       0% | idle      0% | cpu09     0% | curf  2.9MHz
CPL | avg1    3.99 | avg5    2.54 | avg15   1.13 | csw       423 | intr     2e6 |
MEM | tot  131816M | free 128926M | cache   630M | buff    188M | slab    468M | #shmem   17M
SWP | tot       4G | free      4G |              | vmcom    62G | vmlim    70G |
PAG | scan       0 | steal      0 | stall      0 | swin       0 | swout      0 |
LVM | ot           |              | read       0 | write      1 | MBr/s      0 | MBw/s   0.32
LVM | ot           |              | read       0 | write      3 | MBr/s      0 | MBw/s   0.48
DSK | sda          | busy      0% | read       0 | write      3 | MBr/s      0 | MBw/s      0
DSK | sdb          | busy      0% | read       0 | write      1 | MBr/s      0 | MBw/s   0.48
DSK | sdc          | busy      0% | read       0 | write      0 | MBr/s      0 | MBw/s      0
DSK | sdd          | busy      0% | read       0 | write      0 | MBr/s      0 | MBw/s   0.32
DSK | sde          | busy      0% | read       0 | write      0 | MBr/s      0 | MBw/s      0
NET | transport    | tcpi   1e6M | tcpo   1e6M | udpi      0M | udpo      0M | tcpao     0M |
NET | network      | ipi    1e6M | ipo    1e6M | ipfrw     0M | deliv  1e6M | icmpi     0 | i
NET | sfc1         | pcki   1e6M | pcko   1e6M | si    2 Kbps | so    1 Kpbs | erri     0M |
NET | lo           | pcki     10M | pcko     10M | si    0 Kbps | so    0 Kpbs | erri      0M
NET | em1          | pcki     12M | pcko     12M | si    0 Kbps | so    0 Kpbs | erri      0M

  PID SYSCPU USRCPU VGROW    RGROW  RUID   THR ST EXC S CPU  CMD
28683  4.15s  16.58s      0K      7M  root     6  --  -  S 98% udp_tcp_sock_pr
29276  0.00s  0.16s      0K      0K  root     1  --  -  S  0% pmatop
29208  0.02s  0.00s      0K      0K  root     1  --  -  R  0% pmdaproc
28531  0.01s  0.00s      0K      0K  root     1  --  -  S  0% ssh
```

Tuna

Network Tuning: IRQ affinity

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

- 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

Tuna GUI Capabilities Updated for RHEL7

- Run tuning experiments in realtime
- Save settings to a conf file (then load with tuna cli)

Tuna GUI Capabilities Updated for RHEL7

Monitoring | Profile management | Profile editing

Kernel Monitoring

Socket 0

Filter	CPU	Usage
☑	0	0
☑	2	0
☑	4	0
☑	6	0
☑	8	0
☑	10	16
☑	12	0
☑	14	0

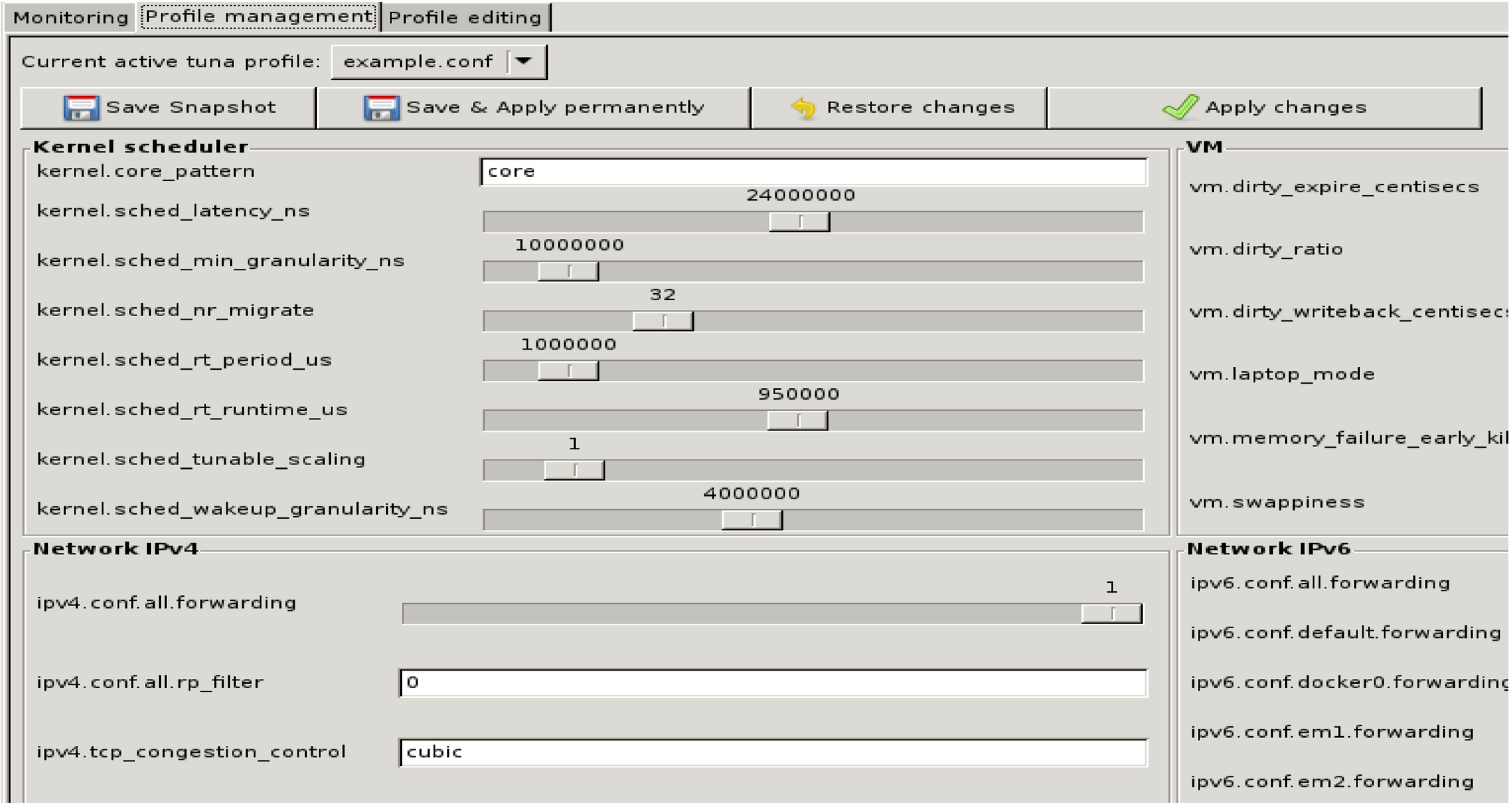
Socket 1

Filter	CPU	Usage
☑	1	0
☑	3	0
☑	5	0
☑	7	0
☑	9	0
☑	11	0
☑	13	0
☑	15	0

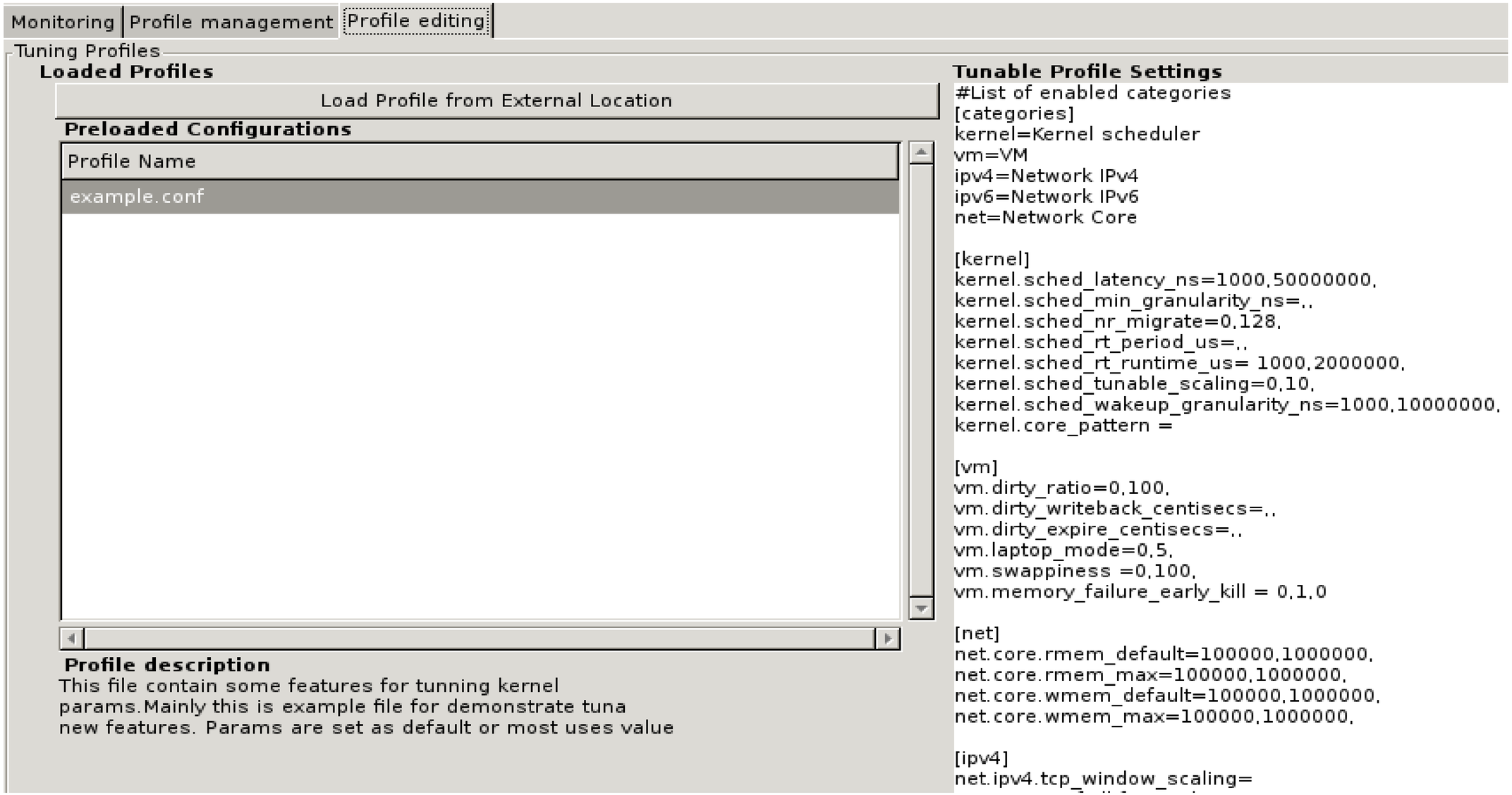
IRQ	Affinity	Events	Users
0	0-15	65	timer
8	1,3,5,7,9,11,13,15	1	rtc0
9	1,3,5,7,9,11,13,15	2	acpi
10	1,3,5,7,9,11,13,15	247	ipmi_si
22	10	37	ehci_hcd:usb2
23	4	136	ehci_hcd:usb1
104	10	0	PCIe PME
105	12	0	PCIe PME
106	14	0	PCIe PME

PID	Policy	Priority	Affinity	VolCtxtSwitch	NonVolCtxtSwitch	CGroup	Command Line
1	OTHER	0	0-15	2416	1907	1:name=systemd:/,2:	/usr/lib/systemd/systemd
2	OTHER	0	0-15	422	0	1:name=systemd:/,2:	kthreadd
3	OTHER	0	0	8300	0	1:name=systemd:/,2:	ksoftirqd/0
5	OTHER	0	0	7	0	1:name=systemd:/,2:	kworker/0:0H
6	OTHER	0	0-15	**14232**	299	1:name=systemd:/,2:	kworker/u64:0
8	FIFO	99	0	227	0	1:name=systemd:/,2:	migration/0

Tuna GUI Capabilities Updated for RHEL7



Tuna GUI Capabilities Updated for RHEL7



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

Tuna – for IRQs

- Move 'p1p1*' IRQs to Socket 1:

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

Core	
0	sfc
1	sfc
2	sfc
3	sfc
4	sfc

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 core/socket isolation

```
# tuna -S1 -i
# grep Cpus_allowed_list /proc/`pgrep rsyslogd`/status
Cpus_allowed_list: 0-15
```

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
```

Analysis Tools

perf

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)
- A single command

perf record

- Record system-wide (-a)
- A single command
- An existing process (-p)

perf record

- Record system-wide (-a)
- A single command
- An existing process (-p)
- Add call-chain recording (-g)

perf record

- Record system-wide (-a)
- A single command
- An existing process (-p)
- Add call-chain recording (-g)
- Only specific events (-e)

perf record

- Record system-wide (-a)
- A single command
- An existing process (-p)
- Add call-chain recording (-g)
- Only specific events (-e)

```
# perf record -g dd if=/dev/zero of=1G bs=1M count=10 oflag=direct
10+0 records in
10+0 records out
10485760 bytes (10 MB) copied, 0.0830041 s, 126 MB/s
[ perf record: Woken up 1 times to write data ]
[ perf record: Captured and wrote 0.016 MB perf.data (~715 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

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                       [.] 0x0000000000064968
     4.32%   +6.43%  libdbus-1.so.3.7.4            [.] 0x0000000000029258
     4.06%   +4.72%  dbus-daemon                   [.] 0x000000000014a6e
     3.79%   -3.79%  libglib-2.0.so.0.3600.3       [.] 0x000000000088d6a
     3.72%   +0.25%  [kernel.kallsyms]             [k] seq_list_start
```

perf probe (dynamic tracepoints)

Insert a tracepoint on any function...

Try 'perf probe -F' to list possibilities

```
# perf probe -a account_user_time
# perf record -e probe:account_user_time -aR sleep 1
[ perf record: Woken up 1 times to write data ]
[ perf record: Captured and wrote 0.098 MB perf.data (~4301 samples) ]

# perf report > out ; cat out
...
# Samples: 1  of event 'probe:account_user_time'
# Event count (approx.): 1
#
# Overhead  Command      Shared Object                Symbol
# ........  .......  ..................  ......................
#
   100.00%    sleep  [kernel.kallsyms]  [k] account_user_time
```

My Probe Point

RHEL7 Performance Tuning Summary

- Use “Tuned”, “NumaD” and “Tuna” in RHEL6 and RHEL7
  - Power savings mode (performance), locked (latency)
  - Transparent Hugepages for anon memory (monitor it)
  - numabalance – Multi-instance, consider “NumaD”
  - Virtualization – virtio drivers, consider SR-IOV
- Manually Tune
  - NUMA – via numactl, monitor numastat -c pid
  - Huge Pages – static hugepages for pinned shared-memory
  - Managing VM, dirty ratio and swappiness tuning
  - Use cgroups for further resource management control

Upcoming Performance Talks

- Performance tuning: Red Hat Enterprise Linux for databases
  - Sanjay Rao, Wednesday April 16, 2:30pm
- Automatic NUMA balancing for bare-metal workloads & KVM virtualization
  - Rik van Riel, Wednesday April 16, 3:40pm
- Red Hat Storage Server Performance
  - Ben England, Thursday April 17, 11:00am

Helpful Utilities

Supportability

- redhat-support-tool
- sos
- kdump
- perf
- psmisc
- strace
- sysstat
- systemtap
- trace-cmd
- util-linux-ng

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

Helpful Links

- Official Red Hat Documentation
- Red Hat Low Latency Performance Tuning Guide
- Optimizing RHEL Performance by Tuning IRQ Affinity
- nohz_full
- Performance Co-Pilot
- Perf
- How do I create my own tuned profile on RHEL7 ?
- Busy Polling Whitepaper
- Blog: http://www.breakage.org/ or @jeremyeder

Q & A

Tuned: Profile virtual-host

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-host

vm.dirty_background_ratio = 5
sched_migration_cost_ns = 5000000

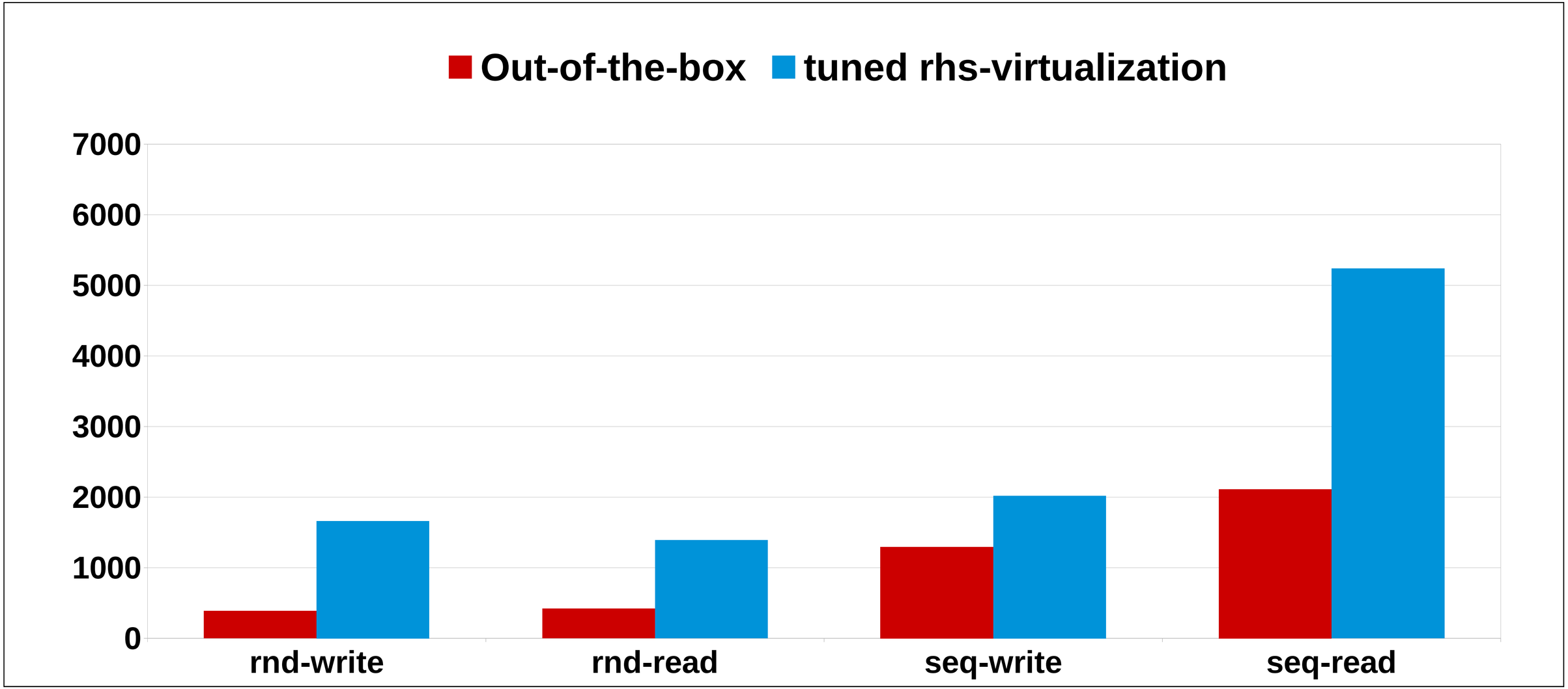
virtual-guest

vm.dirty_ratio = 30
vm.swappiness = 30

RHEL RHS Tuning w/ RHEV/RHEL OSP (tuned)

- **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



RHS Fuse vs libgfapi integration (RHEL6.5 and RHEL7)