RED HAT SUMMIT

LEARN. NETWORK. EXPERIENCE OPEN SOURCE.

June 11-14, 2013
Boston, MA

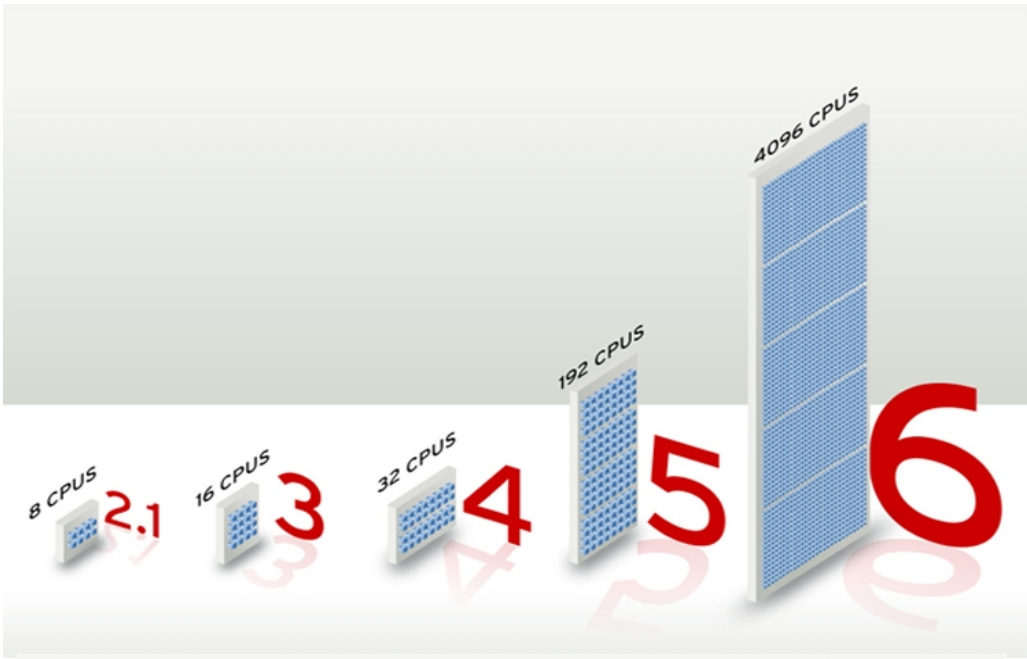
redhat.

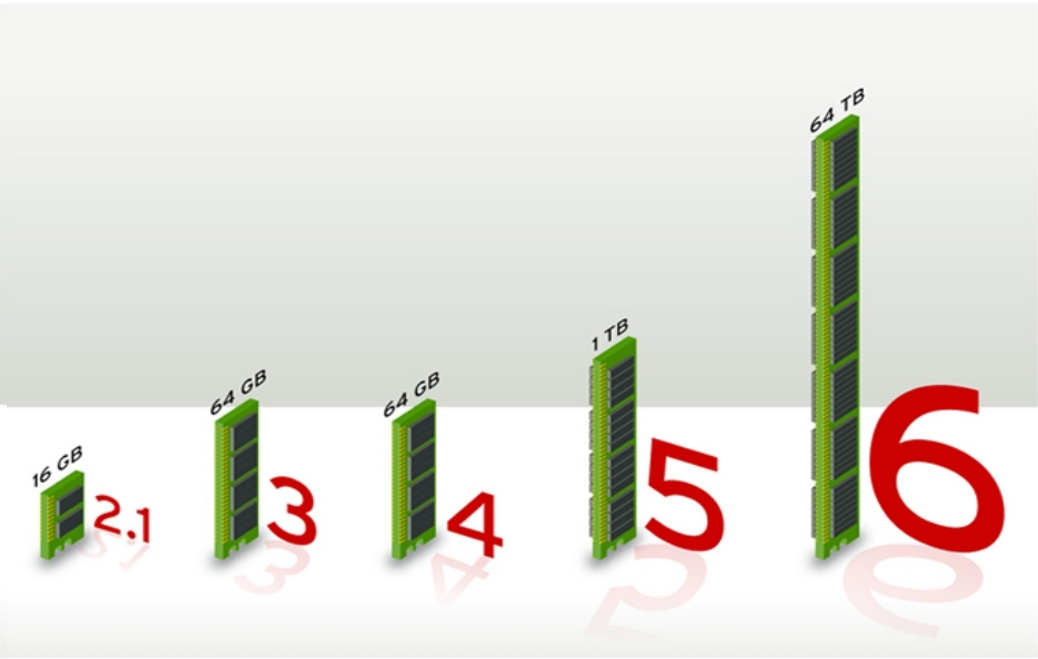
Tuning Red Hat Enterprise Linux for Databases

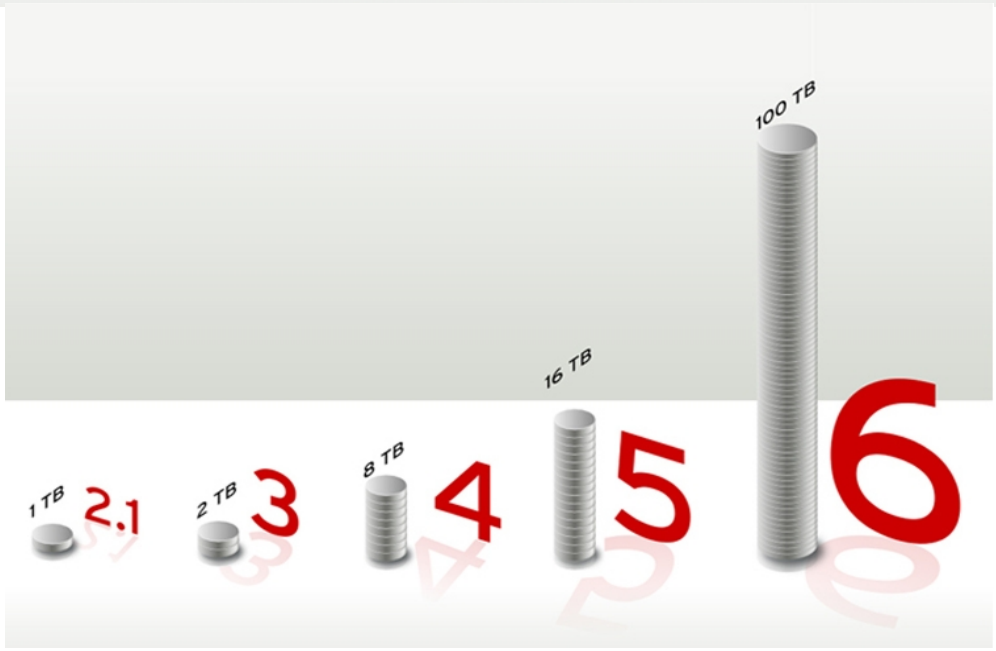
Sanjay Rao
Principal Performance Engineer, Red Hat
June 13, 2013

Objectives of this session

- **Share tuning tips**
  - RHEL 6 scaling
  - Aspects of tuning
  - Tuning parameters
  - Results of the tuning
    - Bare metal
    - KVM Virtualization
- **Tools**







Scalability

RHEL 6 is a lot more scalable but it also offers many opportunites for tuning

What To Tune

- I/O
- Memory
- CPU
- Network

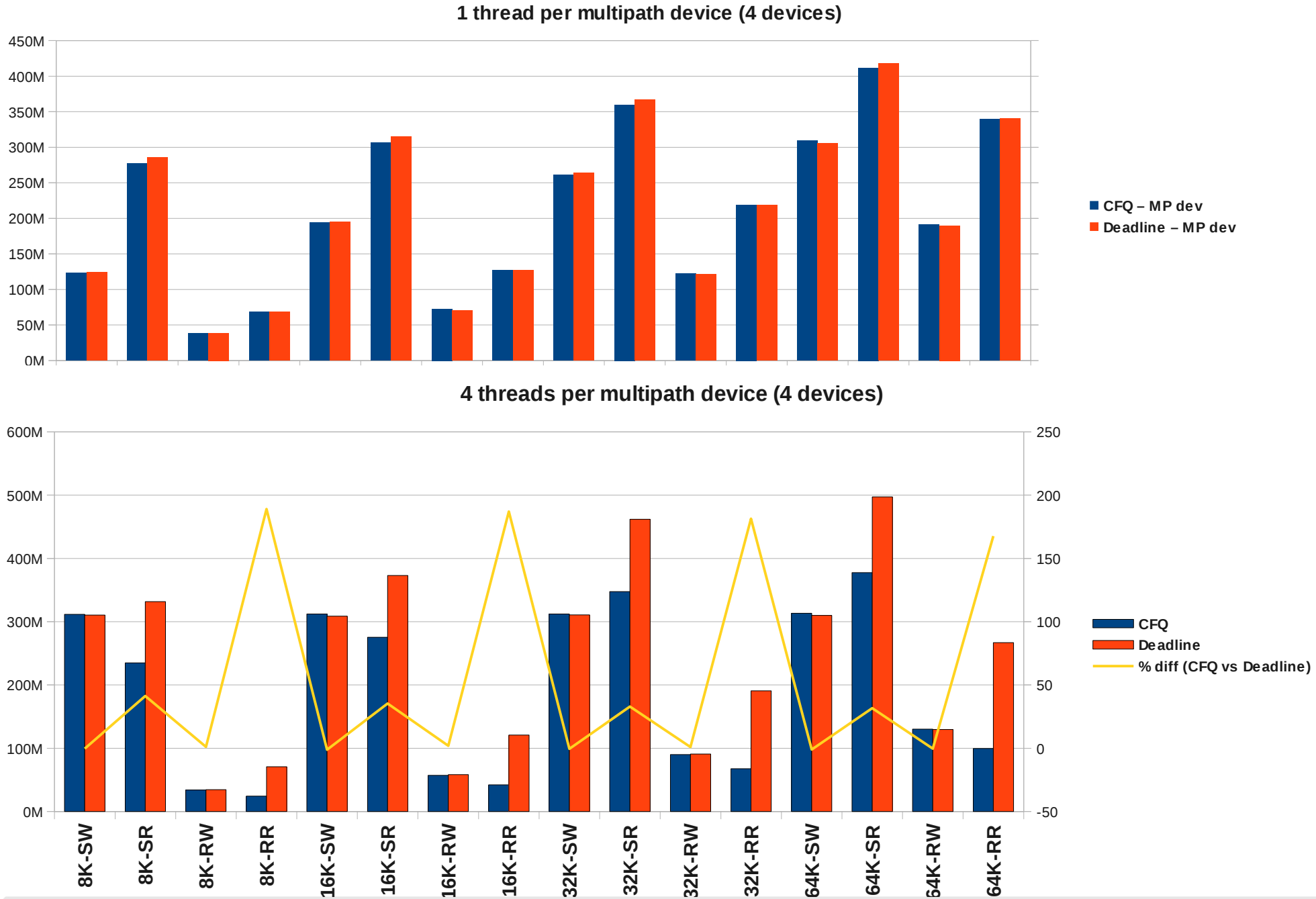
I/O Tuning – Hardware

- **Know Your Storage**
  - SAS or SATA? (Performance comes at a premium)
  - Fibre Channel, Ethernet or SSD?
  - Bandwidth limits (I/O characteristics for desired I/O types)
- **Multiple HBAs**
  - Device-mapper multipath
    - Provides multipathing capabilities and LUN persistence
    - Check for your storage vendors recommendations (upto 20% performance gains with correct settings)
- **How to profile your I/O subsystem**
  - Low level I/O tools – dd, iozone, dt, etc.
  - I/O representative of the database implementation

I/O Tuning – Understanding I/O Elevators

- **Deadline**
  - Two queues per device, one for read and one for writes
  - I/Os dispatched based on time spent in queue
  - Used for multi-process applications and systems running enterprise storage
- **CFQ**
  - Per process queue
  - Each process queue gets fixed time slice (based on process priority)
  - Default setting - Slow storage (SATA)
- **Noop**
  - FIFO
  - Simple I/O Merging
  - Lowest CPU Cost
  - Low latency storage and applications (Solid State Devices)

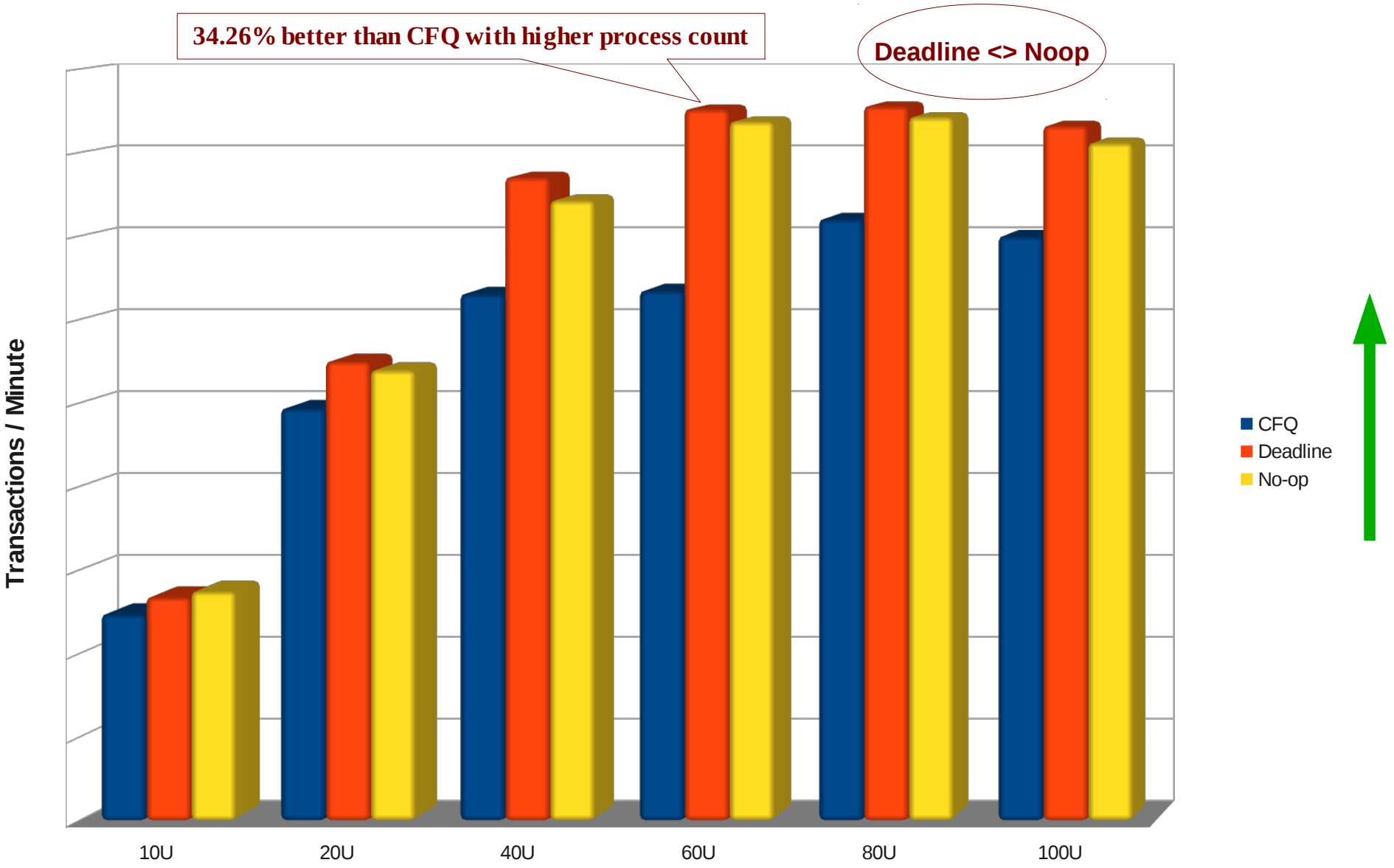
CFQ vs Deadline



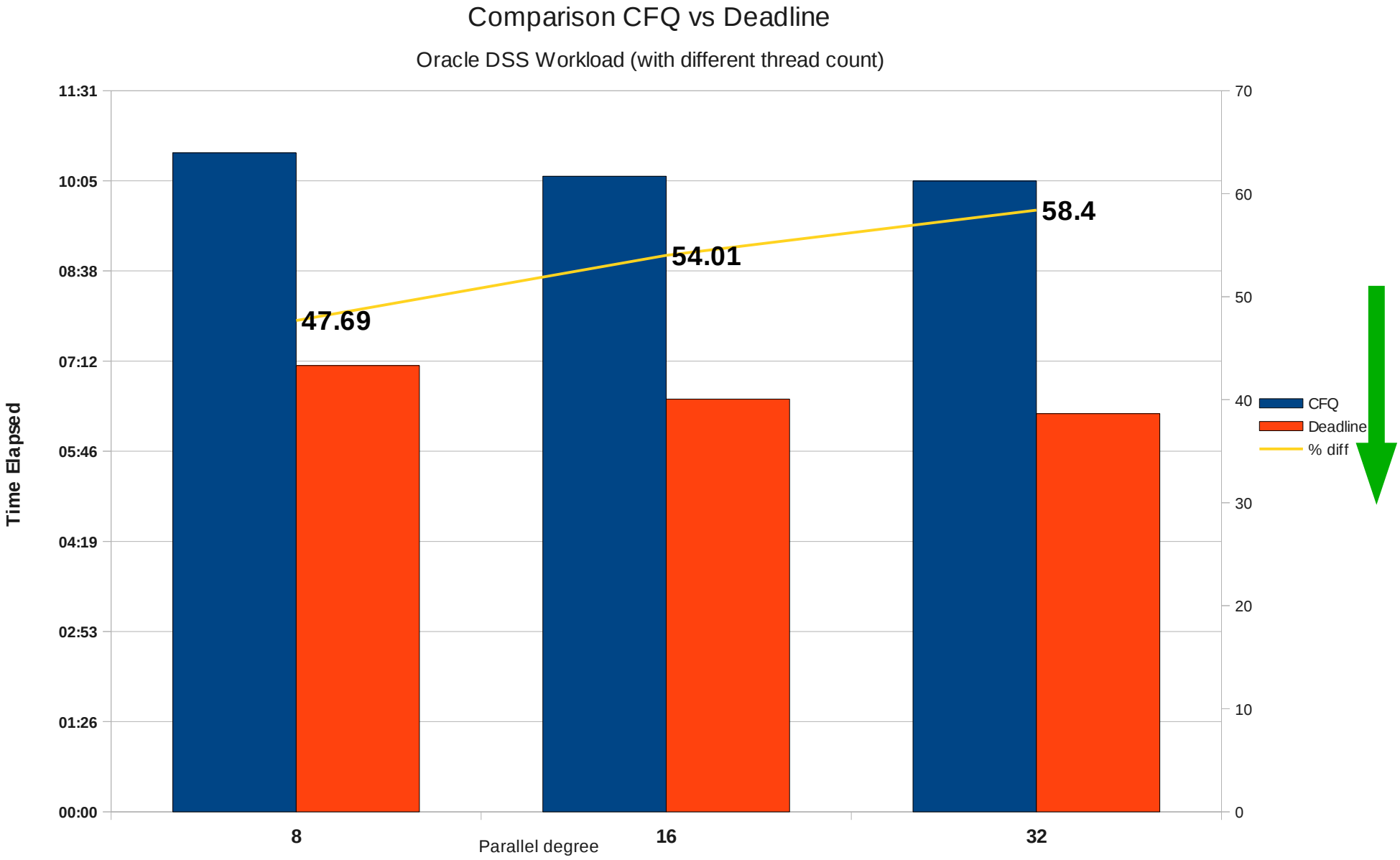
I/O Tuning – Configuring I/O Elevators

- **Boot-time**
  - Grub command line – elevator=deadline/cfq/noop
- **Dynamically, per device**
  - echo “deadline” > /sys/class/block/sda/queue/scheduler
- **tuned (RHEL6 utility)**
  - tuned-adm profile throughput-performance
  - tuned-adm profile enterprise-storage

Impact of I/O Elevators – OLTP Workload

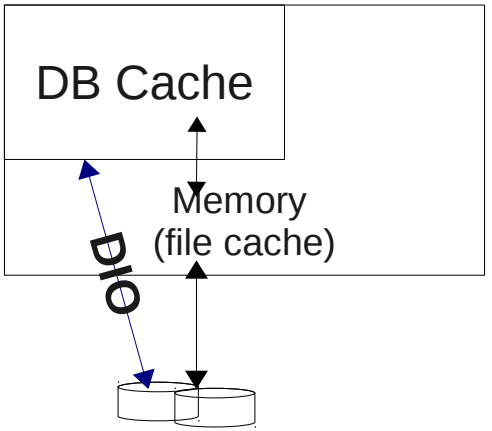


Impact of I/O Elevators – DSS Workload

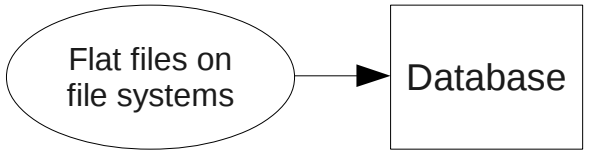


I/O Tuning – File Systems

- **Direct I/O**
  - Avoid double caching
  - Predictable performance
  - Reduce CPU overhead



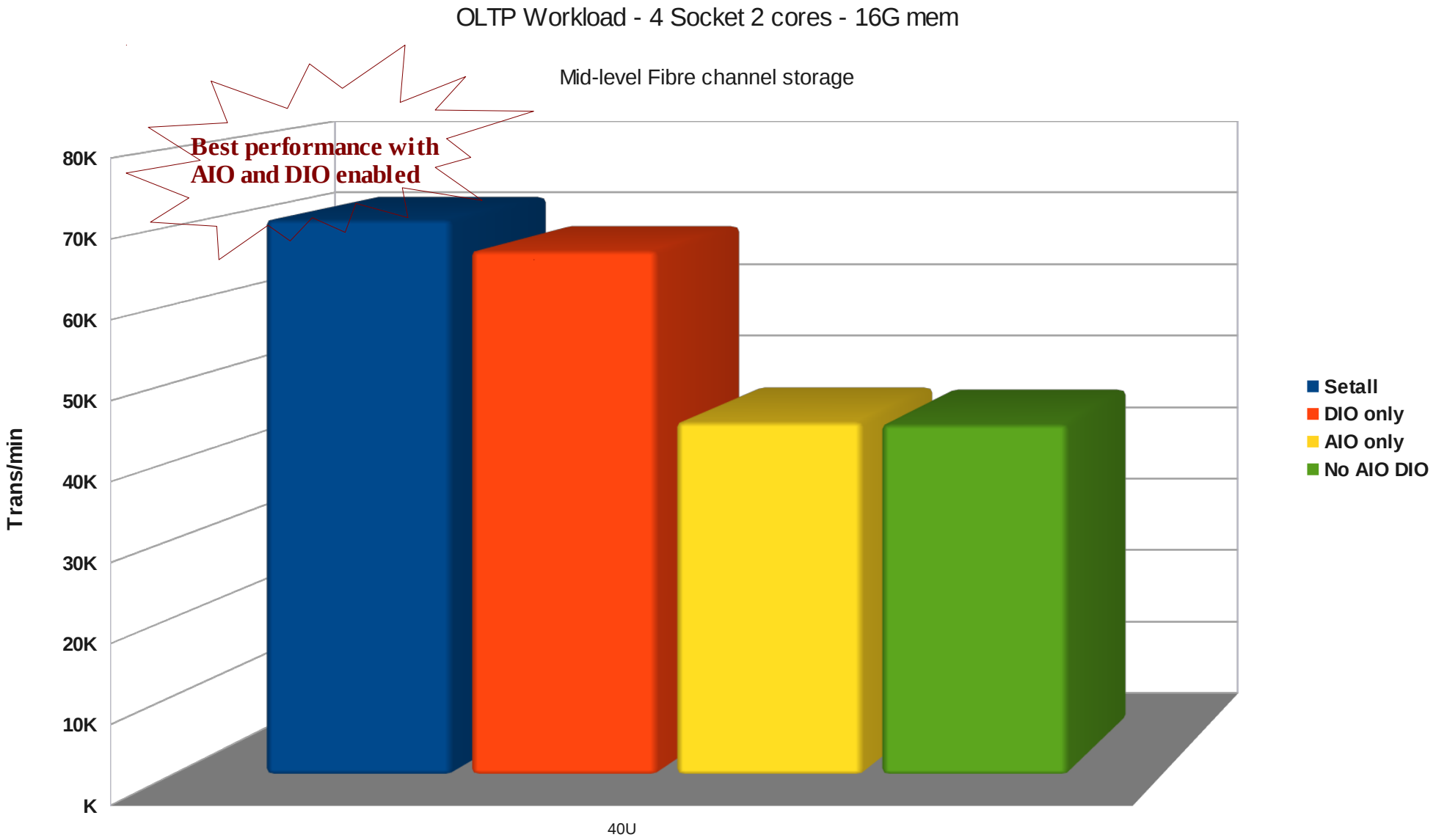
- **Asynchronous I/O**
  - Eliminate synchronous I/O stall
  - Critical for I/O intensive applications



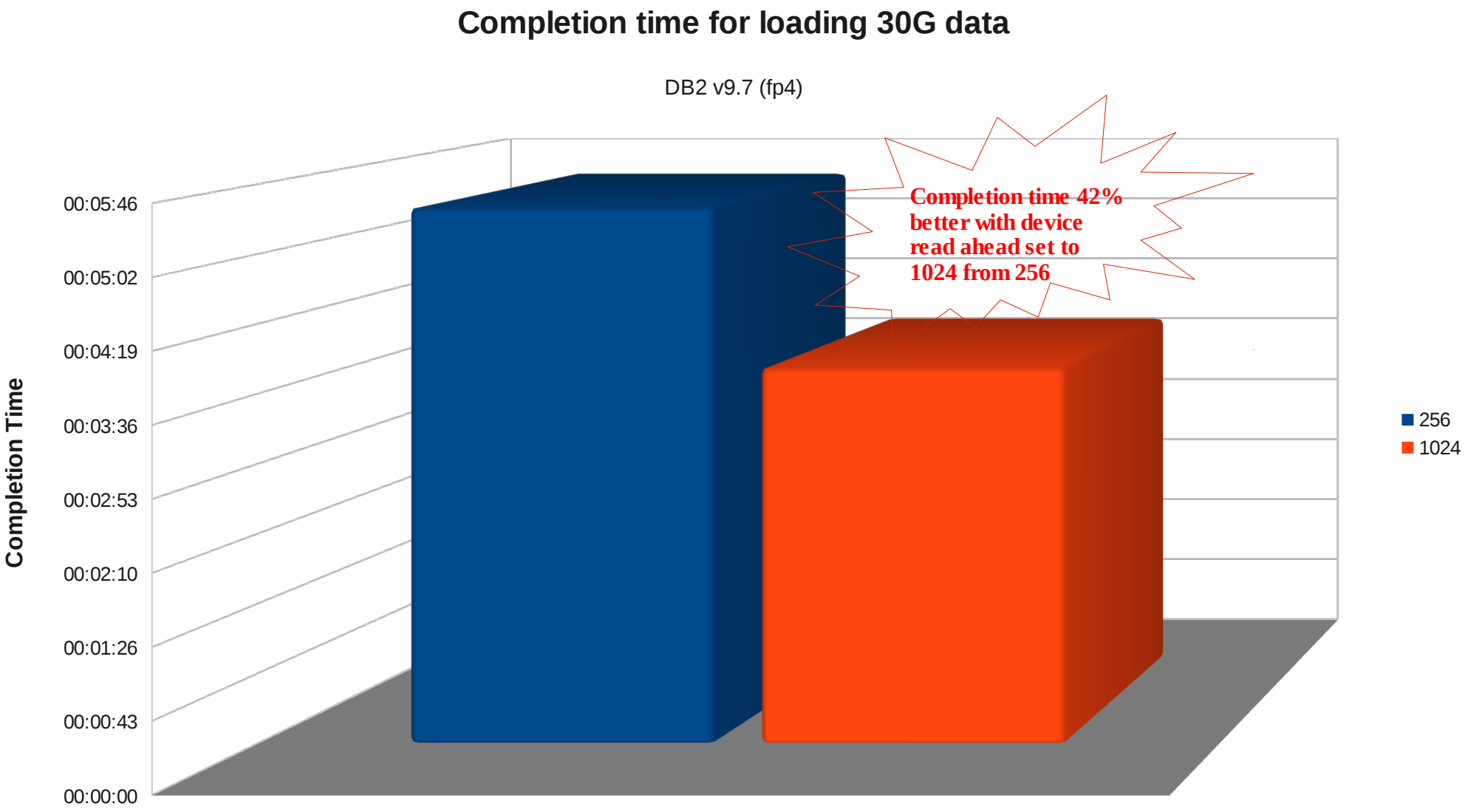
Configure read ahead

- **Configure read ahead** (for sequential read operations)
  - Database (parameters to configure read ahead)
  - Block devices (commands – "**blockdev -- getra / setra**")
  - Configure device read ahead for large data loads
- **Turn off I/O barriers** (RHEL6 and enterprise storage only)

I/O Tuning – Effect of Direct I/O, Asynch I/O



I/O Tuning – Effect of read ahead during data load



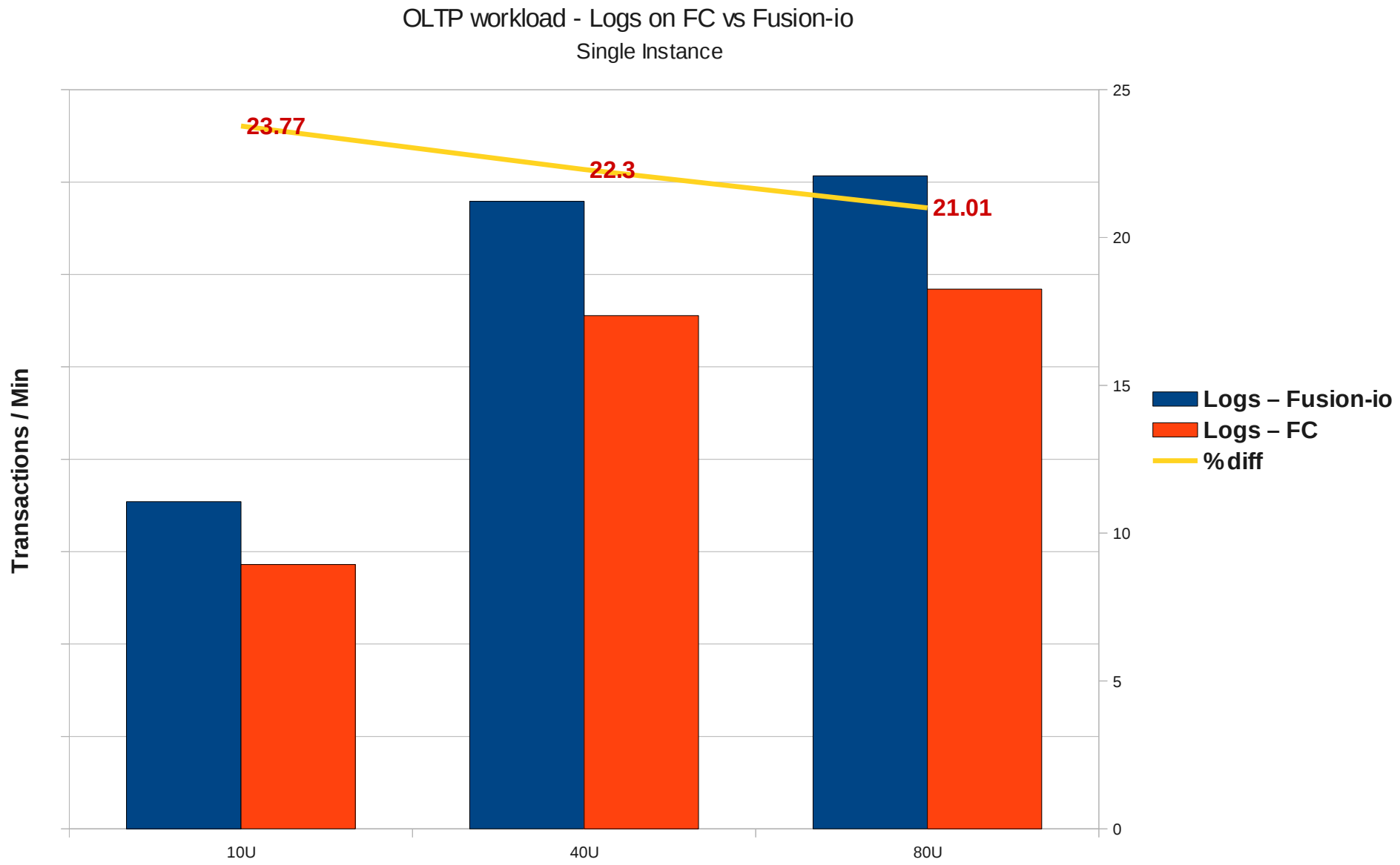
I/O Tuning – Database Layout

- Separate files by I/O (data , logs, undo, temp)
- **OLTP** – data files / undo / logs
  - All transactions generate logs and undo information
- **DSS** – data files / temp files
  - Merge / joins / indexes generally use temp segments
- Use low latency / high bandwidth devices for hot spots
  - Use database statistics

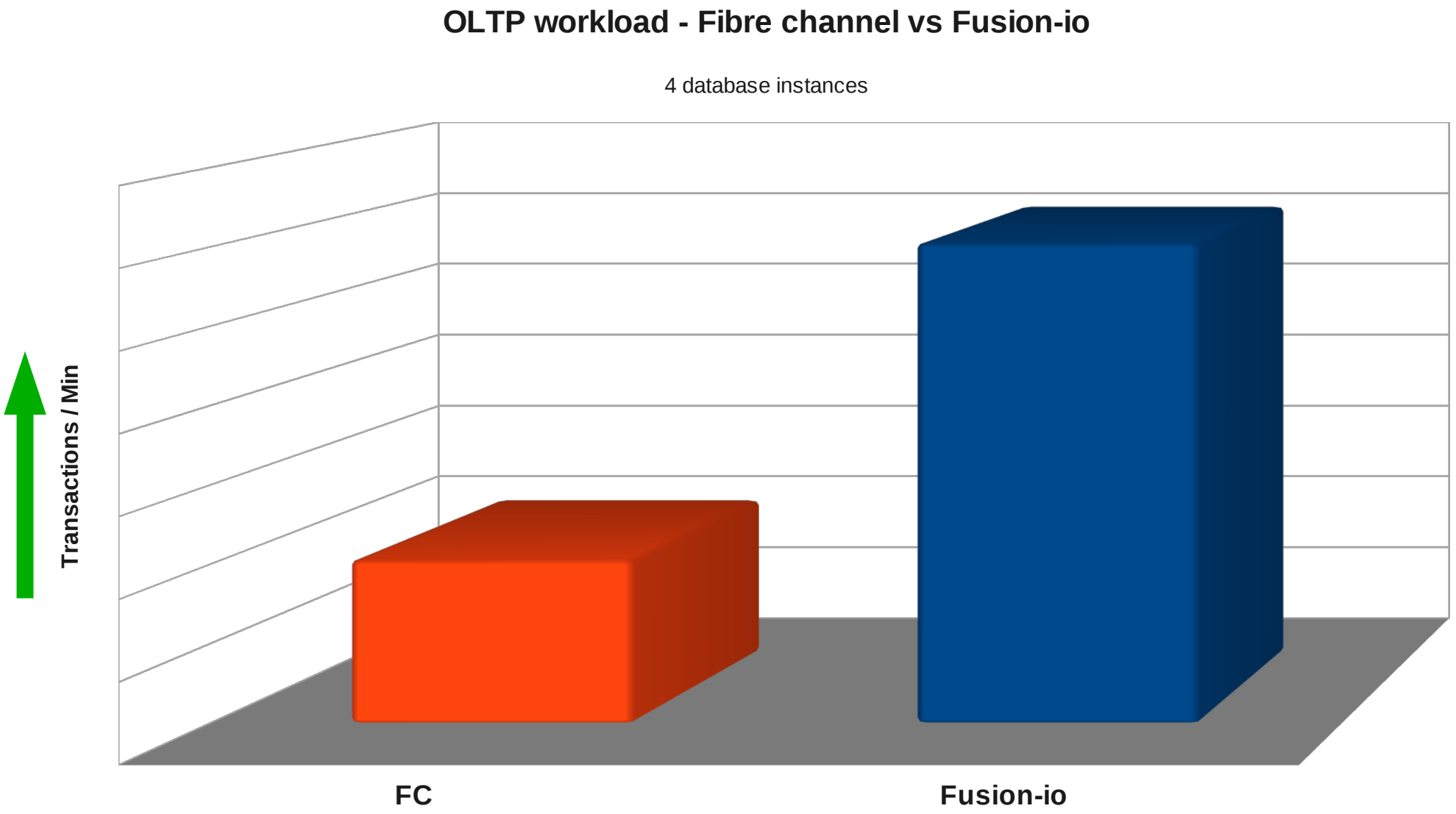
Linux Tool used for Identifying I/O hotspots

- *iostat -dmxz <interval>*
  - This shows I/O for all the disks that are in use

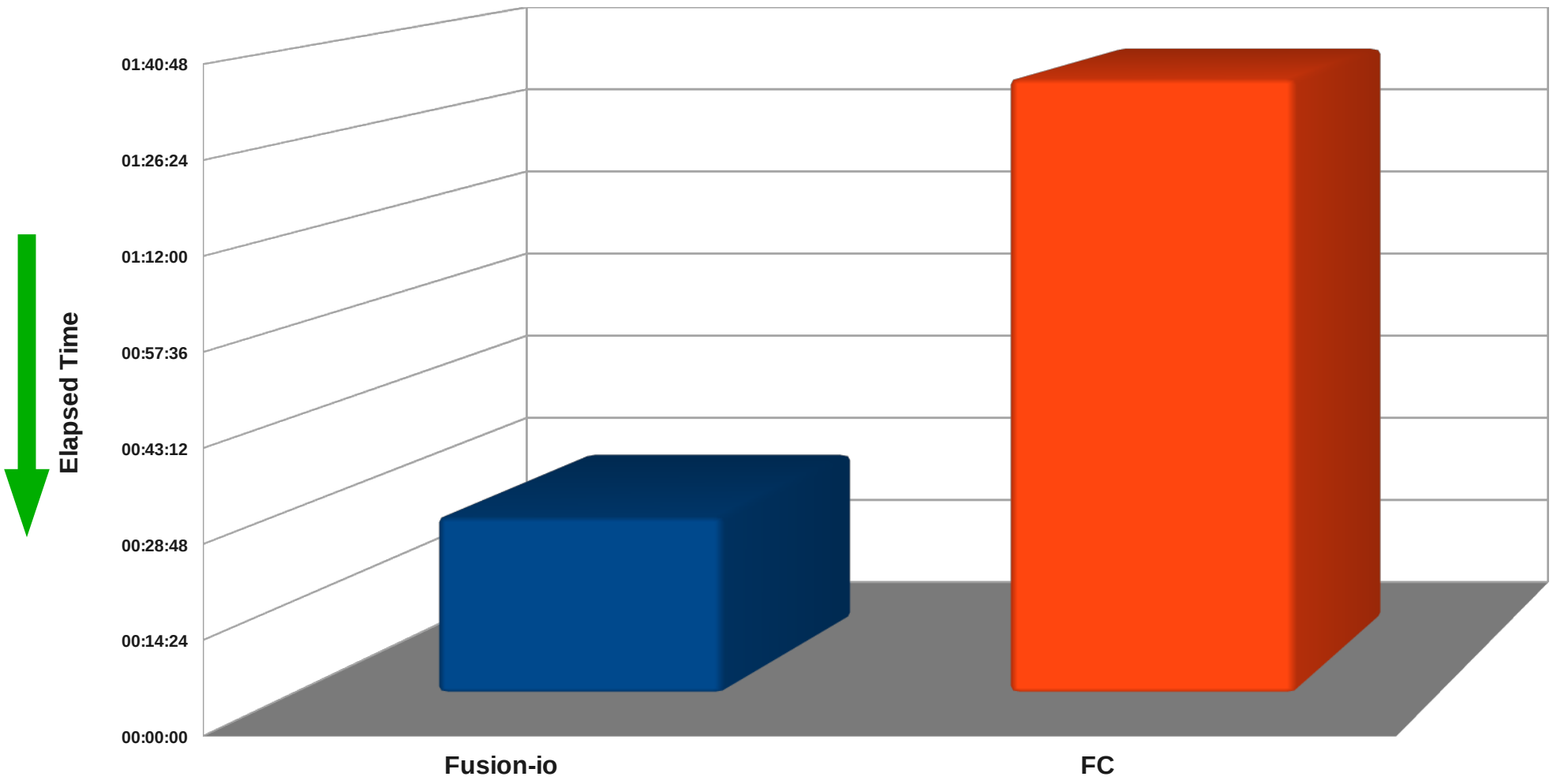
I/O Tuning – OLTP - Logs



I/O Tuning – Storage (OLTP database)



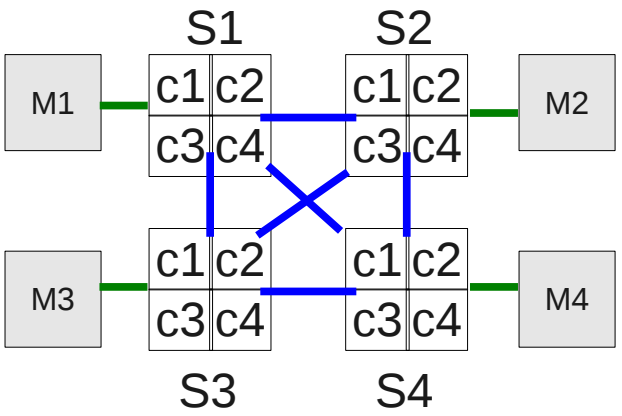
I/O Tuning – DSS - Temp

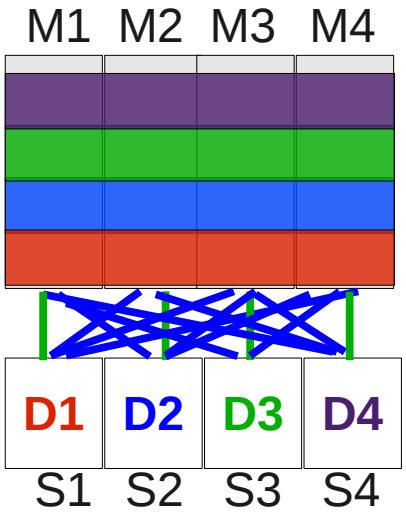


Memory Tuning

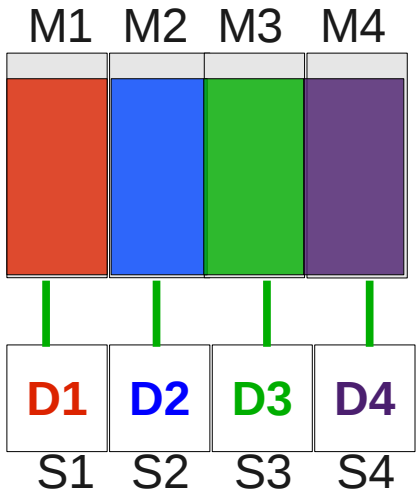
- **NUMA**
- **Huge Pages**
- **Manage Virtual Memory pages**
  - Flushing of dirty pages
  - Swapping behavior

Understanding NUMA (Non Uniform Memory Access)





No NUMA optimization



NUMA optimization

- Multi Socket – Multi core architecture
  - NUMA required for scaling
  - RHEL 5 / 6 completely NUMA aware
  - Additional performance gains by enforcing NUMA placement

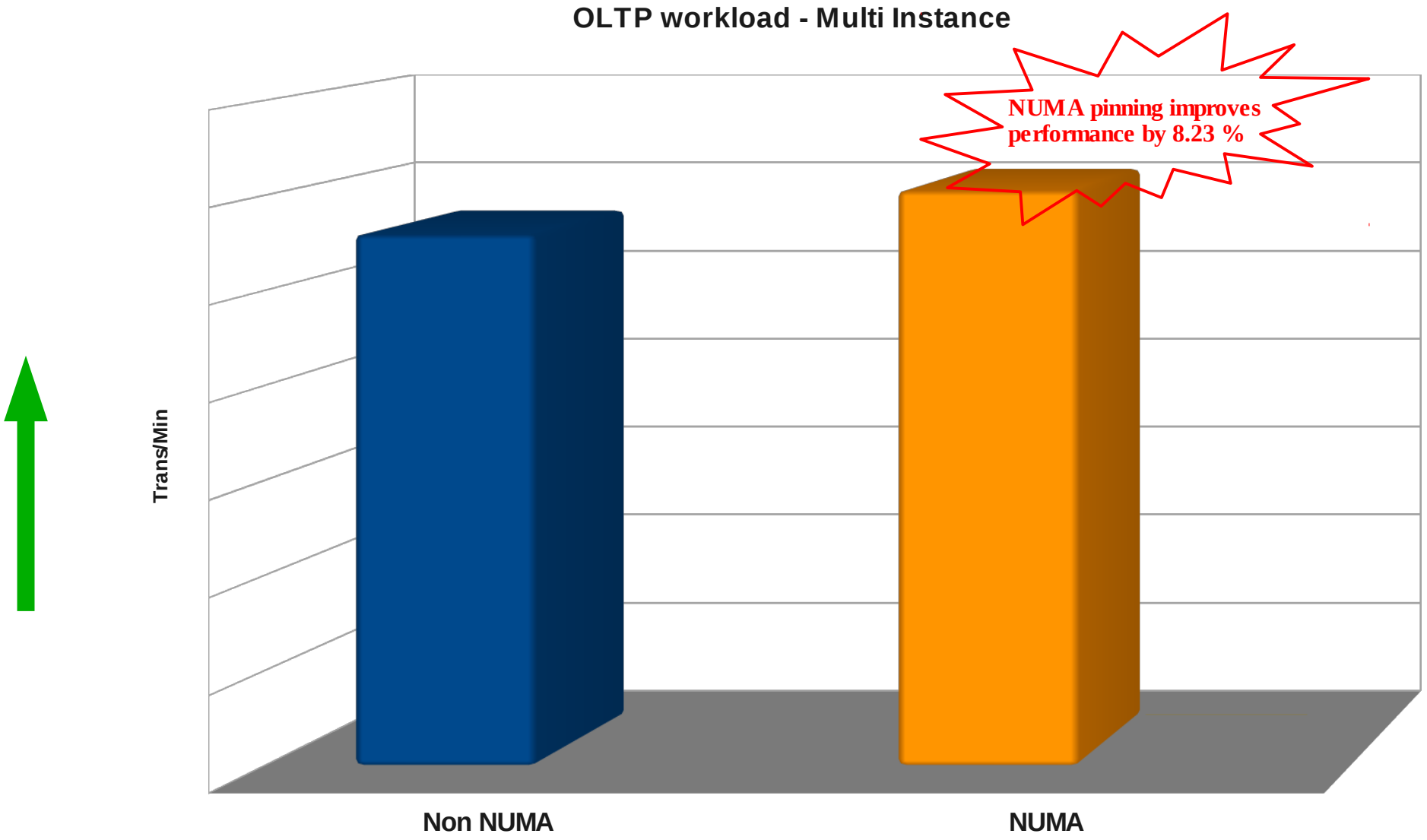
Memory Tuning – Finding NUMA layout

```
[root@perf30 ~]# numactl --hardware
available: 4 nodes (0-3)
node 0 cpus: 0 4 8 12 16 20 24 28 32 36 40 44 48 52 56 60
node 0 size: 32649 MB
node 0 free: 30868 MB
node 1 cpus: 1 5 9 13 17 21 25 29 33 37 41 45 49 53 57 61
node 1 size: 32768 MB
node 1 free: 29483 MB
node 2 cpus: 2 6 10 14 18 22 26 30 34 38 42 46 50 54 58 62
node 2 size: 32768 MB
node 2 free: 31082 MB
node 3 cpus: 3 7 11 15 19 23 27 31 35 39 43 47 51 55 59 63
node 3 size: 32768 MB
node 3 free: 31255 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
```

Memory Tuning – NUMA

- Enforce NUMA placement
  - numactl
    - CPU and memory pinning
  - Taskset
    - CPU pinning
  - cgroups (only in RHEL6)
    - cpusets
    - cpu and memory cgroup
  - Libvirt
    - for KVM guests – CPU pinning

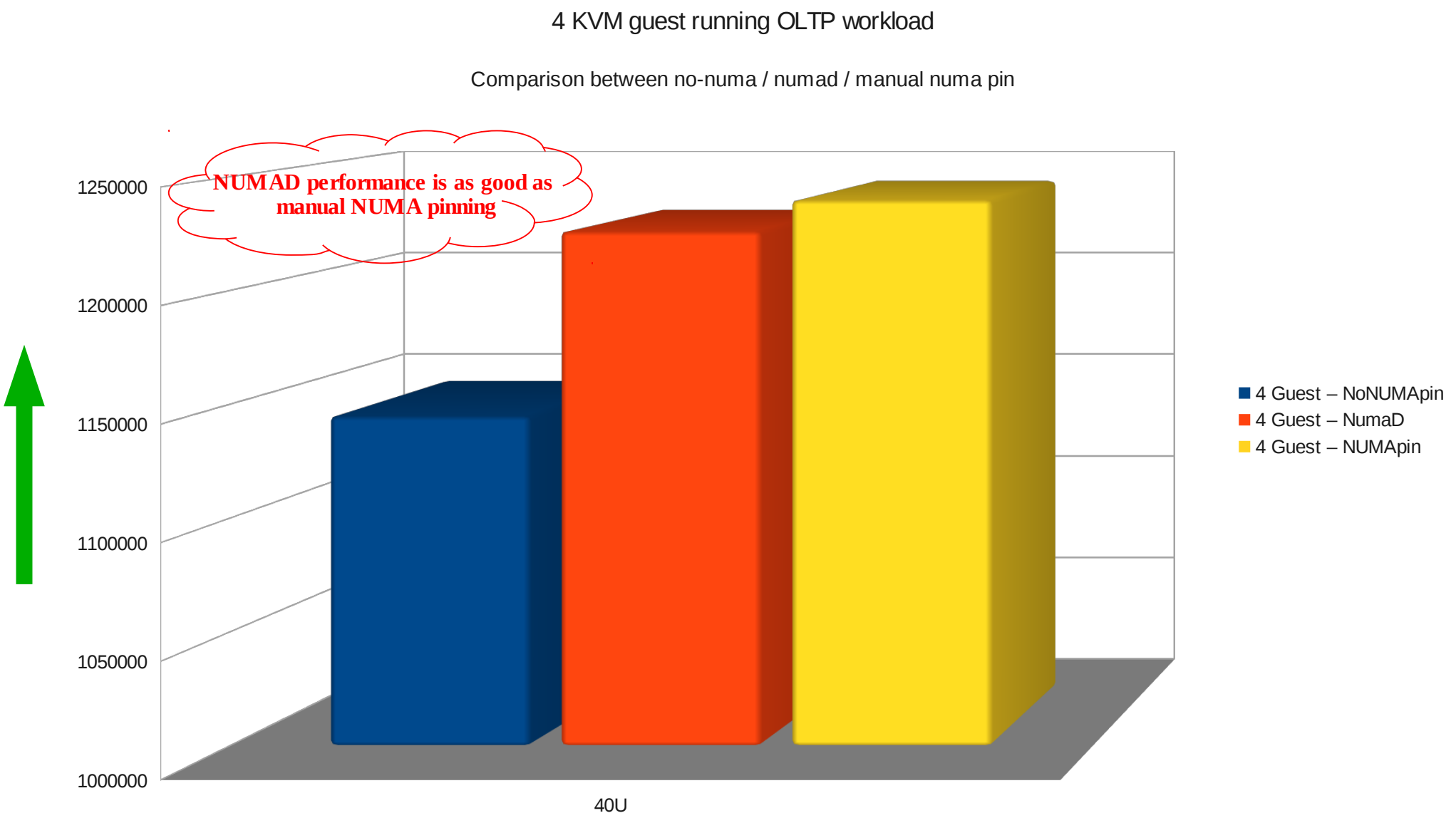
Memory Tuning – Effect of NUMA Tuning



Memory Tuning – NUMA - “numad”

- **What is numad?**
  - User-level daemon to automatically improve out of the box NUMA system performance
  - Added to Fedora 17
  - Added to RHEL 6.3 as tech preview
  - Not enabled by default
- **What does numad do?**
  - Monitors available system resources on a per-node basis and assigns significant consumer processes to aligned resources for optimum NUMA performance.
  - Rebalances when necessary
  - Provides pre-placement advice for the best initial process placement and resource affinity.

Memory Tuning – Effect of “numad”

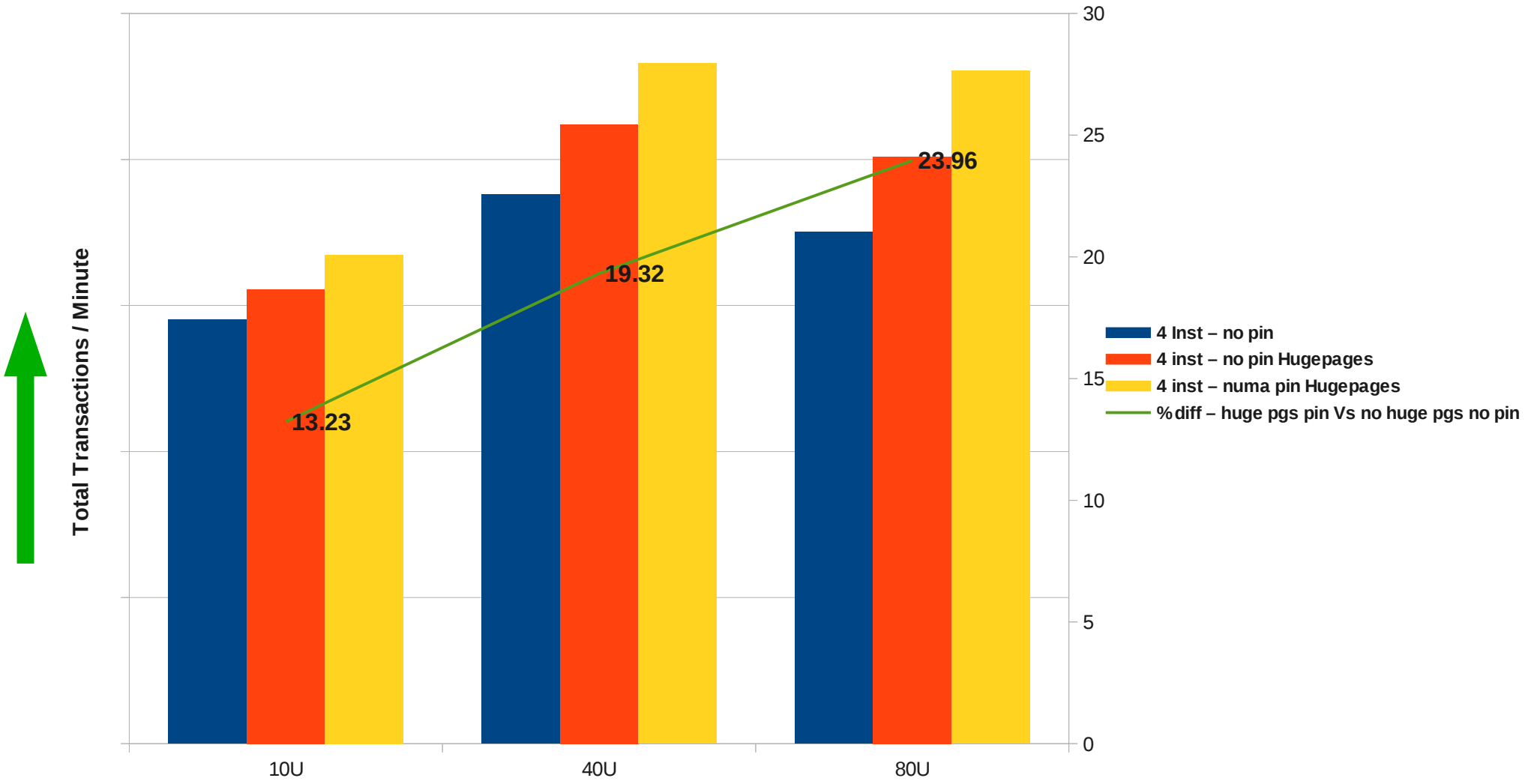


Memory Tuning – Huge Pages

- 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)

- **How to configure Huge Pages (16G)**
  - echo 8192 > /proc/sys/vm/nr_hugepages
  - vi /etc/sysctl.conf (vm.nr_hugepages=8192)

Memory Tuning – Effect of huge pages



Tuning Memory – Flushing Caches

- Drop unused Cache
  - ✔ Frees unused memory
  - ✔ File cache
  - ✗ If the DB uses cache, may notice slowdown

- **Free pagecache**
  - echo 1 > /proc/sys/vm/drop_caches
- **Free slabcache**
  - echo 2 > /proc/sys/vm/drop_caches
- **Free pagecache and slabcache**
  - echo 3 > /proc/sys/vm/drop_caches

Tuning Memory – swappiness

- Not needed as much in RHEL6
- Controls how aggressively the system reclaims “mapped” memory:
- Default - 60%
- Decreasing: more aggressive reclaiming of unmapped pagecache memory
- Increasing: more aggressive swapping of mapped memory

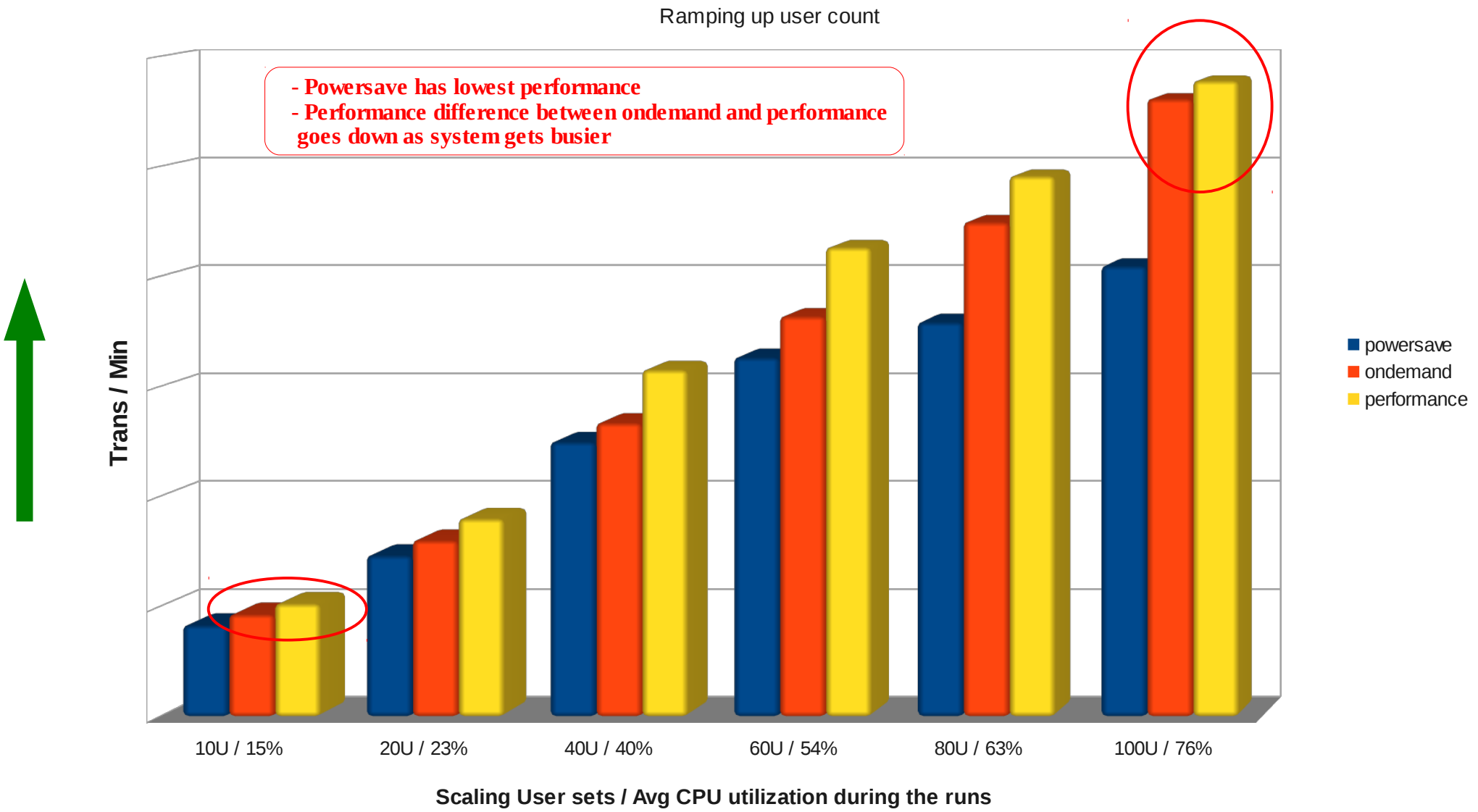
CPU Tuning – Power Savings / cpuspeed

- Power savings mode
  - cpuspeed off
  - performance
  - ondemand
  - powersave

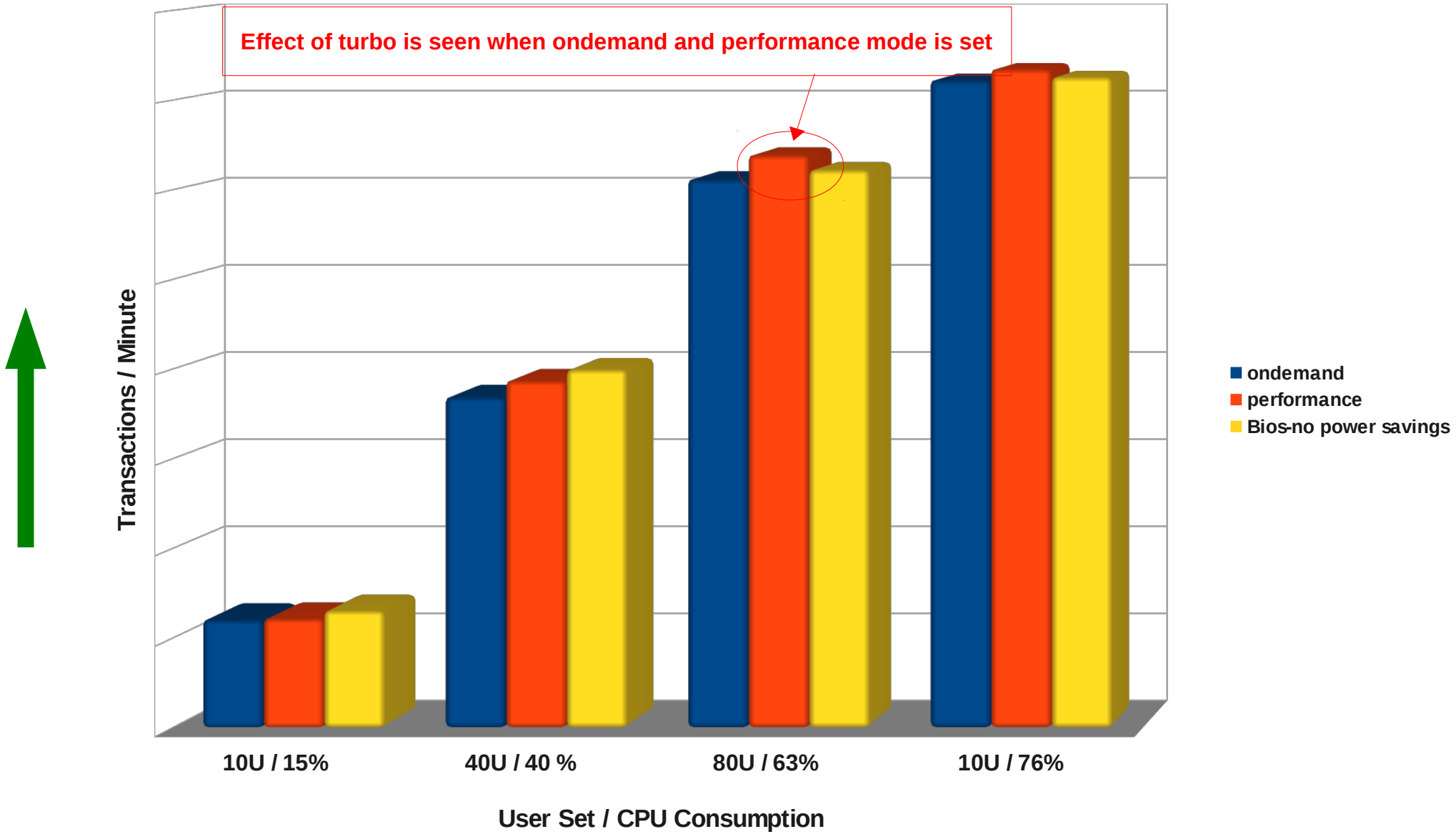
How To

- echo "performance" > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
- best of both worlds – cron jobs to configure the governor mode
- tuned-adm profile server-powersave (RHEL6)

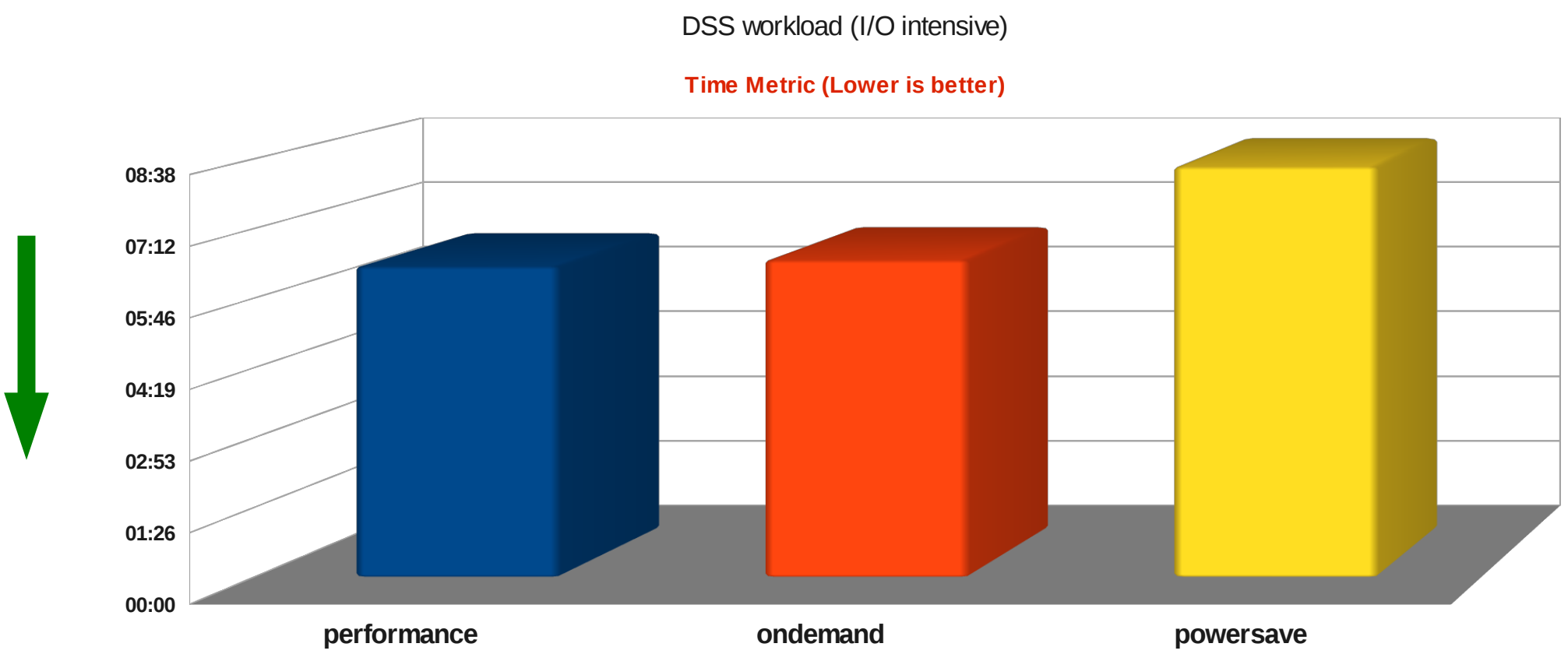
CPU Tuning – Effect of Power Savings - OLTP



CPU Tuning – Effect of Power Savings - OLTP



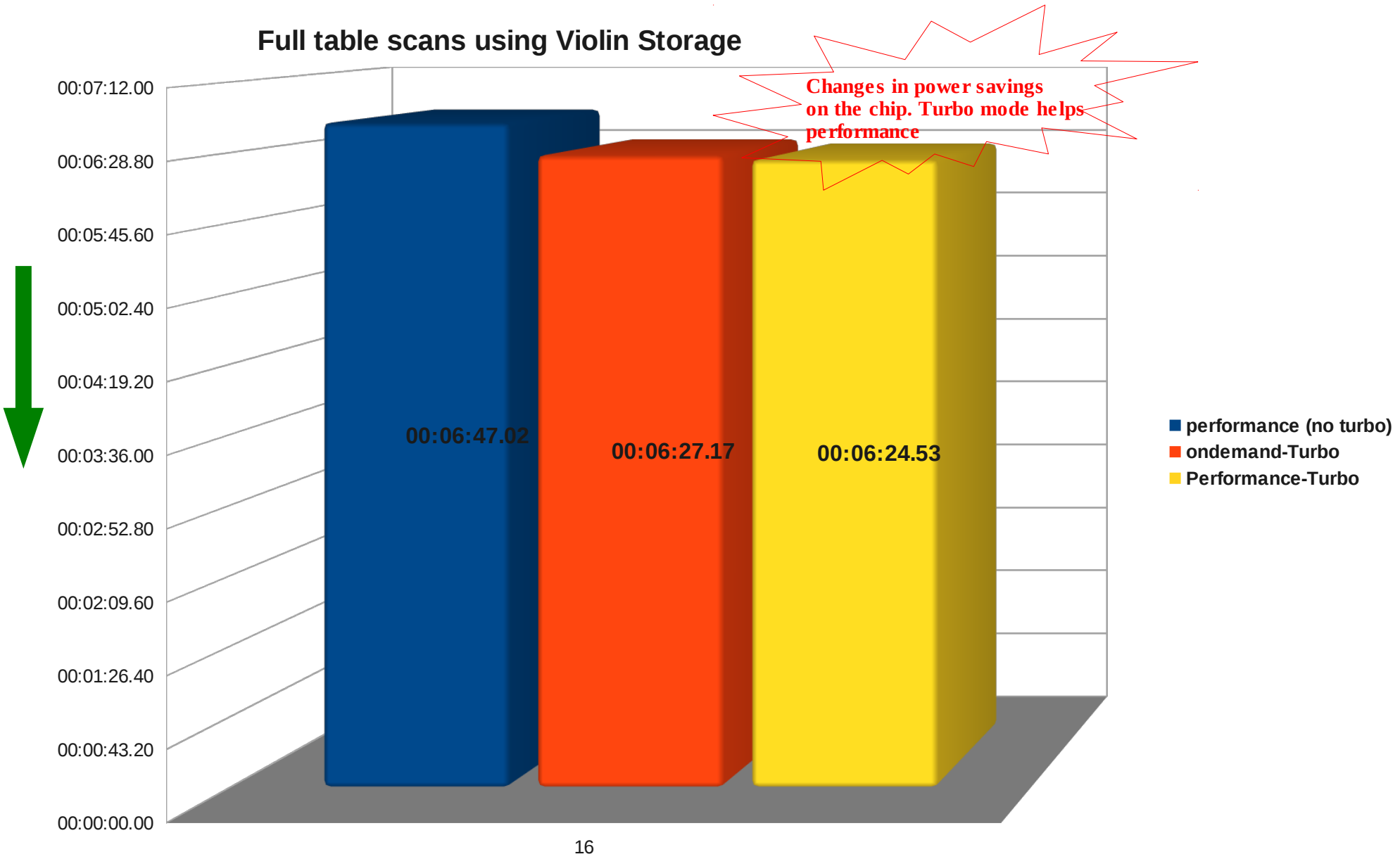
CPU Tuning – Effect of Power Savings - OLTP



vmstat output during test:

7	12	5884	122884416	485900	734376	0	0	184848	39721	9175	37669	4	1	89	6	0
7	12	5884	122885024	485900	734376	0	0	217766	27468	9904	42807	4	2	87	6	0
2	0	5884	122884928	485908	734376	0	0	168496	45375	6294	27759	4	1	90	5	0
7	11	5884	122885056	485912	734372	0	0	178790	40969	9433	38140	4	1	90	5	0
1	15	5884	122885176	485920	734376	0	0	248283	19807	7710	37788	5	2	86	7	0

CPU Tuning – Effect of Power Savings - OLTP



CPU Tuning – C-states

- Various states of the CPUs for power savings
- C0 through C6
- C0 – full frequency (no power savings
- C6 (deep power down mode – maximum power savings)
- OS can tell the processors to transition between these states

How To

- Turn off power savings mode in BIOS (No OS control or Turbo mode)
- ls /sys/devices/system/cpu/cpu0/cpuidle/state*

Linux Tool used for monitoring c-states (only for Intel)

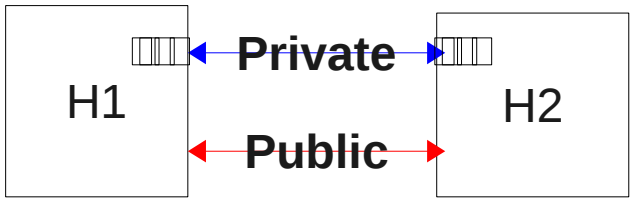
- *turbostat -i <interval>*

RHEL6 Technology Innovation – Networking

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

Network Tuning – Databases

- Network Performance
  - Separate network for different functions (Private network for database traffic)



  - If on same network, use arp_filter to prevent ARP flux
  - echo 1 > /proc/sys/net/ipv4/conf/all/arp_filter

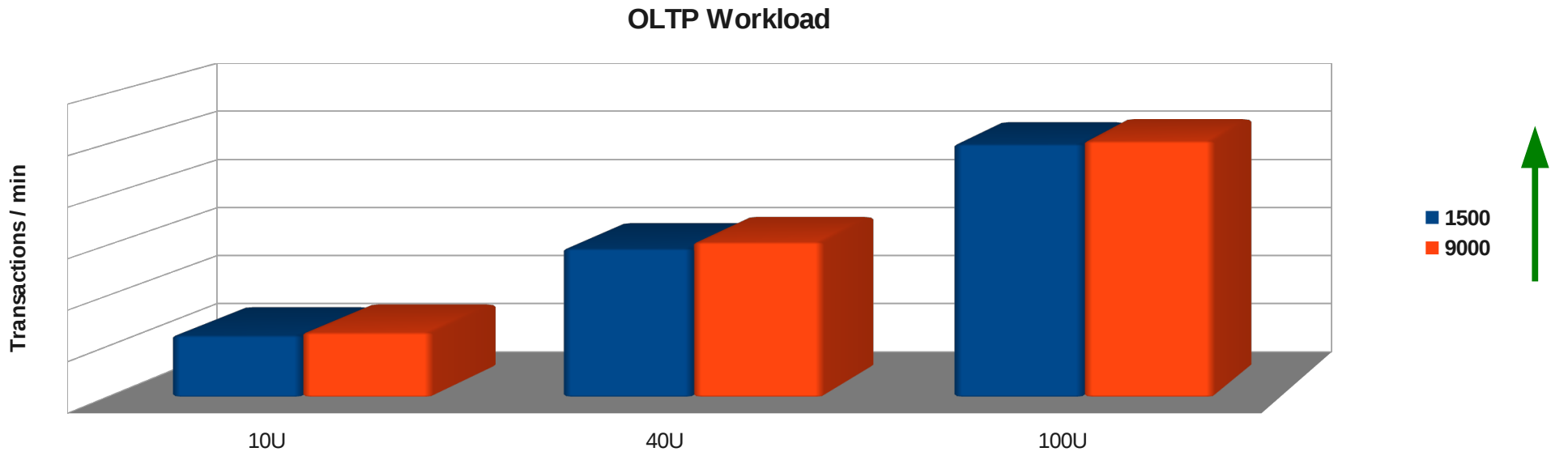
Hardware

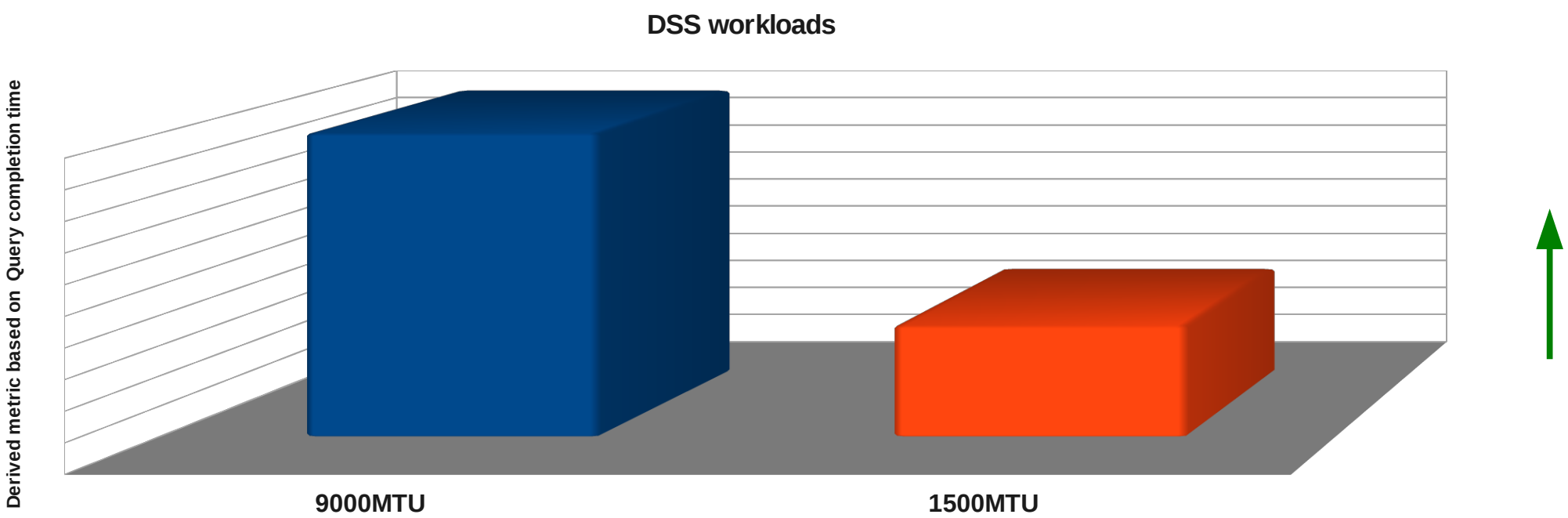
  - 10GigE
    - Supports RDMA w/ RHEL6 high performance networking package (**ROCE**)
  - Infiniband (Consider the cost factor)
- Packet size (Jumbo frames)

Linux Tool used for monitoring network

- *sar -n DEV <interval>*

Network tuning – Jumbo Frames with iSCSI storage





Performance Setting Framework – tuned

tuned for RHEL6

- Configure system for different performance profiles
  - laptop-ac-powersave
  - spindown-disk
  - latency-performance
  - laptop-battery-powersave
  - server-powersave
  - throughput-performance
  - desktop-powersave
  - enterprise-storage
  - Default
- Create your own custom profiles
- Can be rolled back

tuned profile summary...

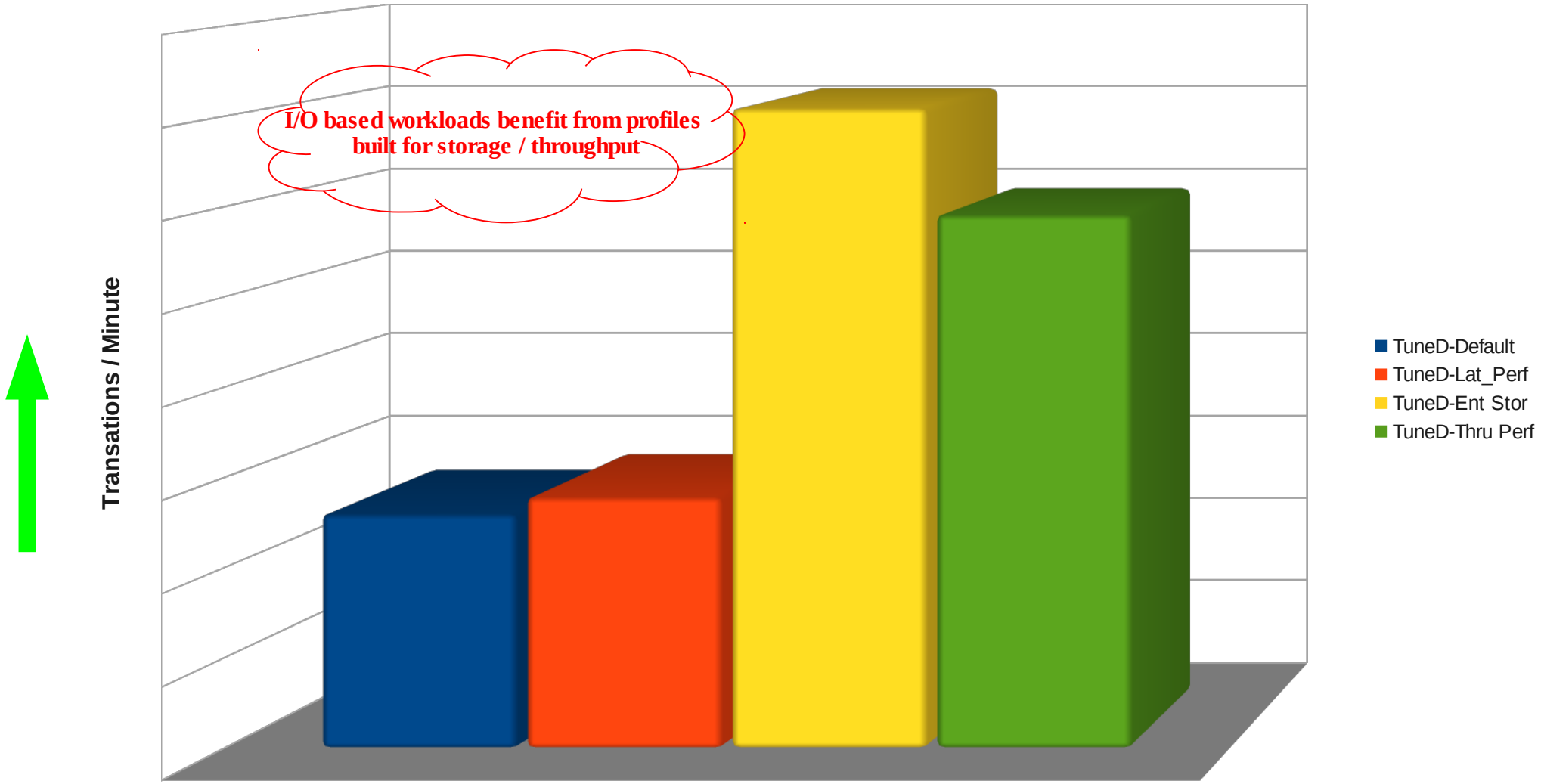
Tunable	default	latency-performance	throughput-performance	enterprise-storage	virtual-host	virtual-guest
kernel.sched_min_granularity_ns	4ms		10ms	10ms	10ms	10ms
kernel.sched_wakeup_granularity_ns	4ms		15ms	15ms	15ms	15ms
vm.dirty_ratio	20% RAM		40%	40%	10%	40%
vm.dirty_background_ratio	10% RAM				5%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

Tuned – Important files / customization

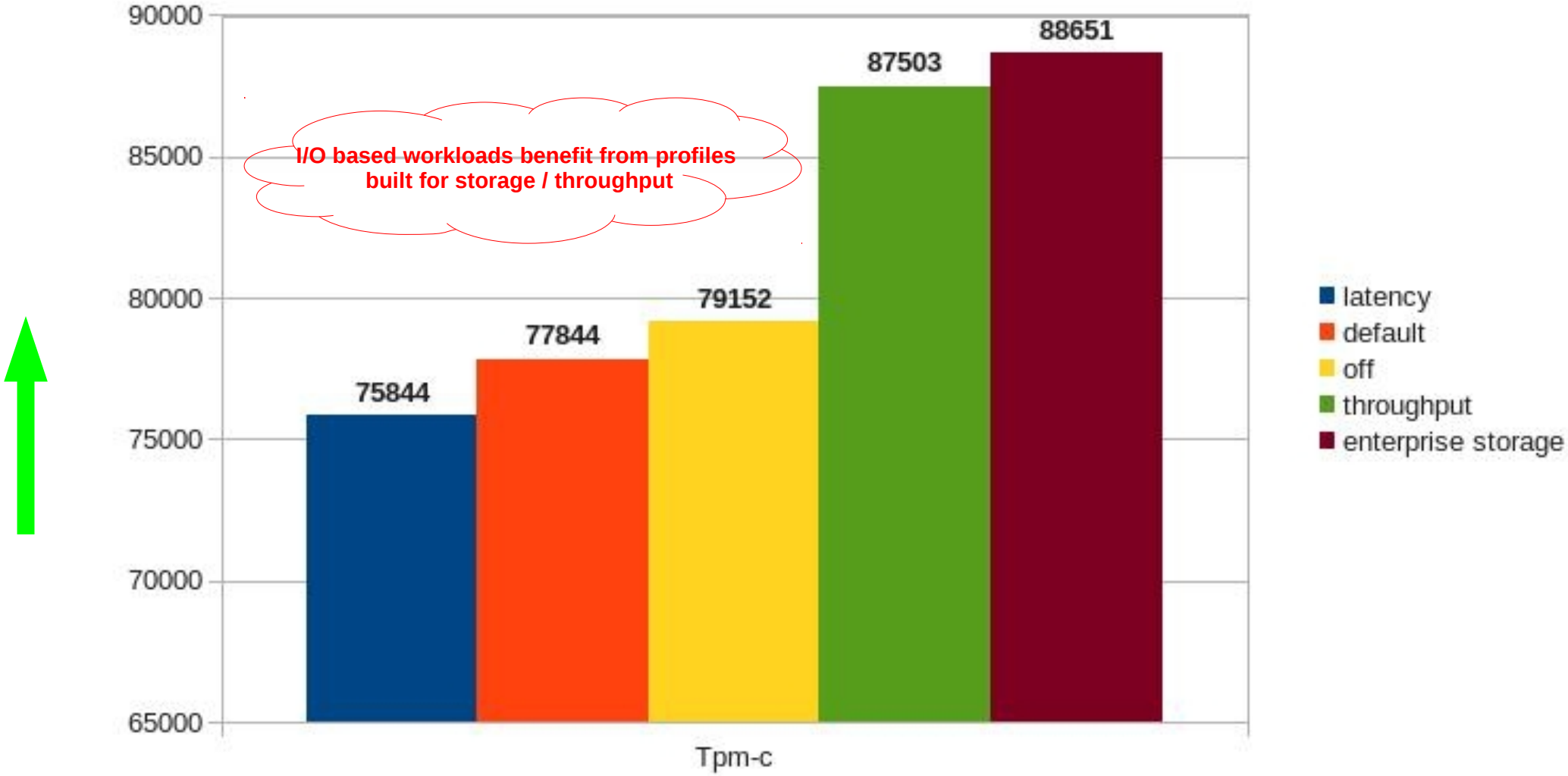
- Turning tuned on
- tuned-adm profile enterprise-storage
- Important tuned files
  - **/etc/tune-profiles** # Directory with config files

  - /etc/tune-profiles/enterprise-storage/ktune.sh
  - /etc/tune-profiles/enterprise-storage/sysctl.ktune
  - /etc/tune-profiles/enterprise-storage/ktune.sysconfig

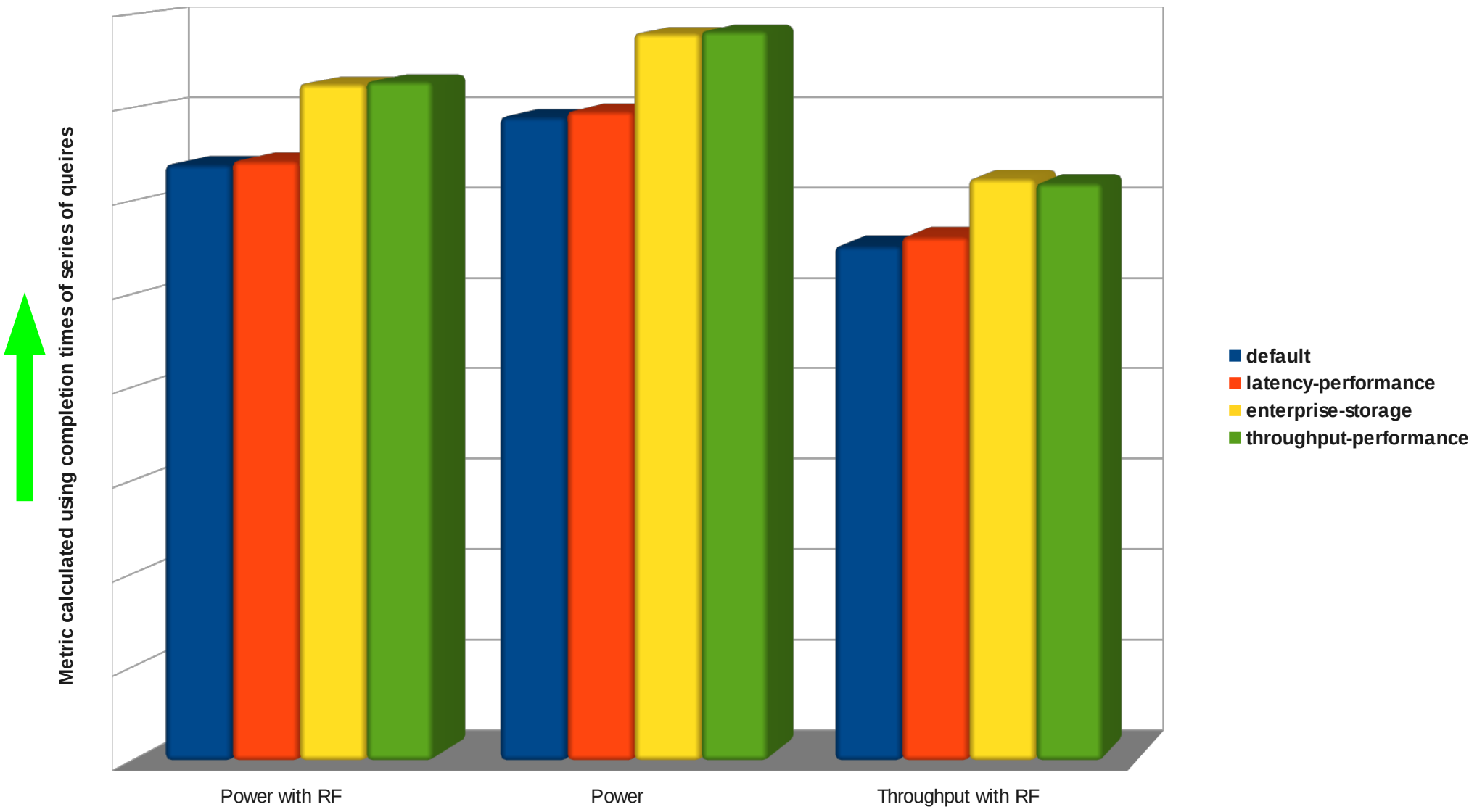
Tuned – OLTP workload



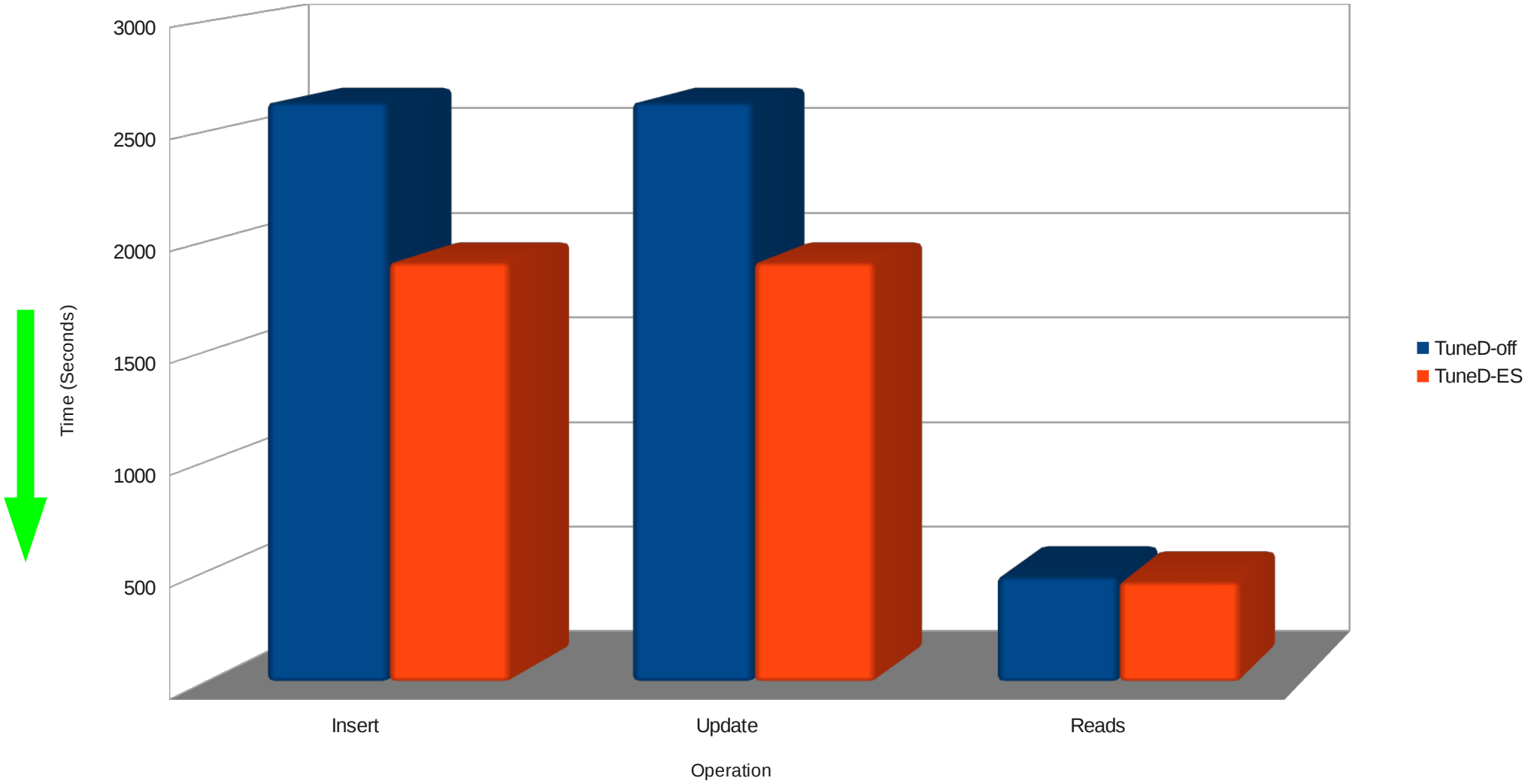
Tuned – BenchmarkSQL 2.3.3 on PostgreSQL 9.2



Tuned – Sybase IQ – DSS Workload – 300GB database



Tuned – MongoDB – YCSB workload - Using Journals



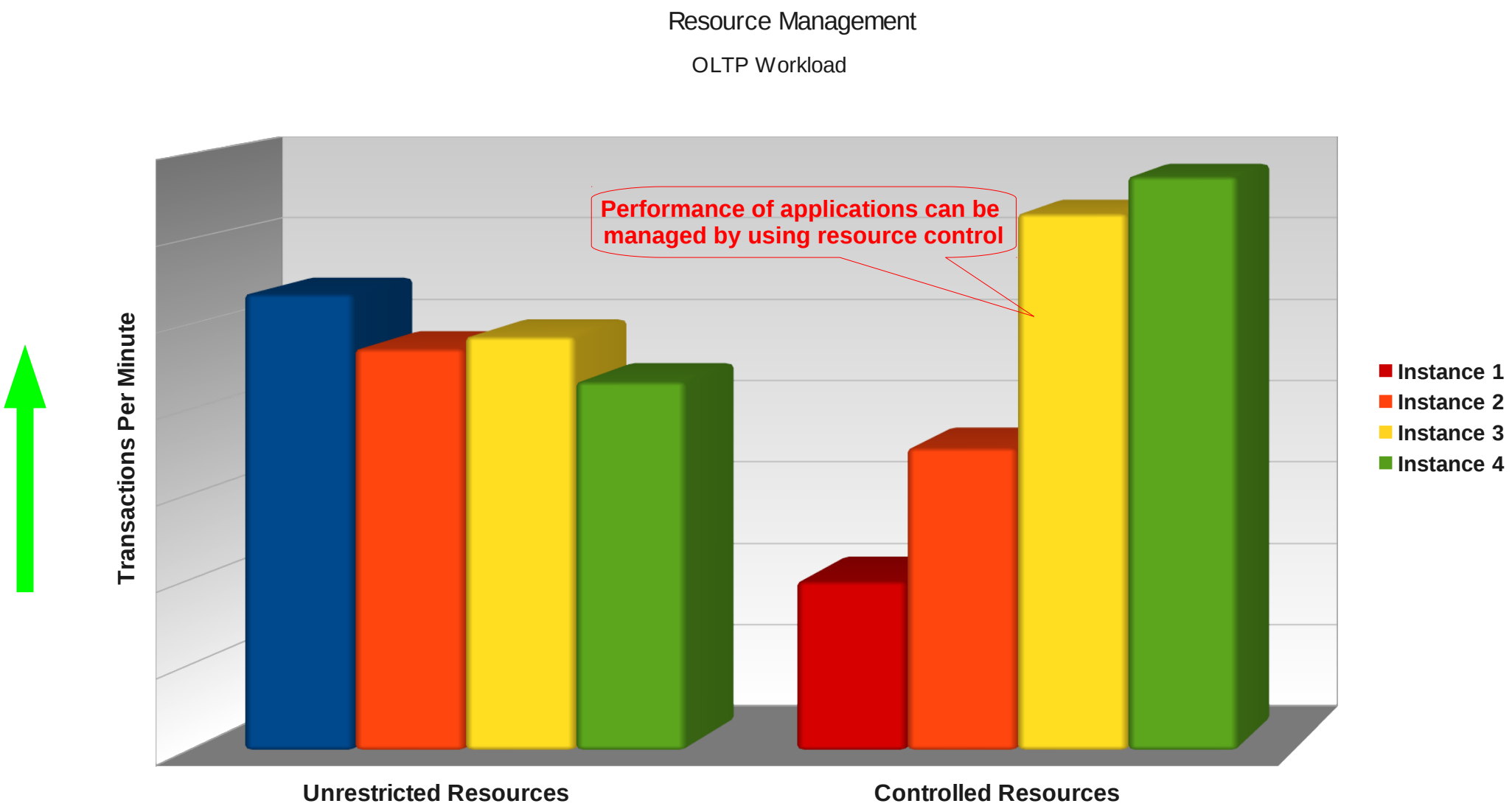
Database Performance

- Application tuning
  - Design
  - Reduce locking / waiting
  - Database tools (optimize regularly)

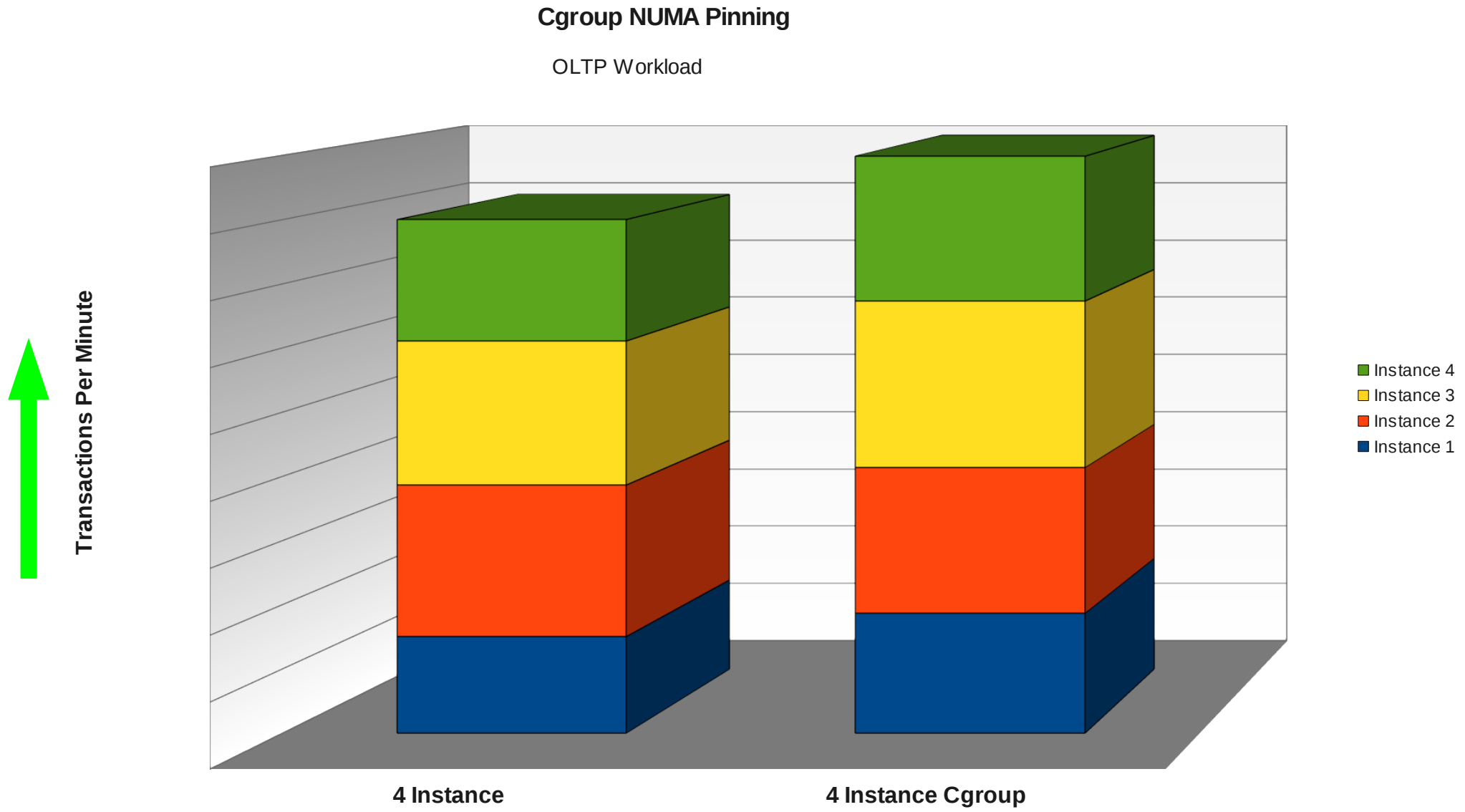
C-group – Resource Management and Performance

- Resource Management
  - Memory, cpus, IO, Network
  - For performance
  - For application consolidation
  - Dynamic resource allocation
- Application Isolation
- I/O Cgroups
  - At device level control the % of I/O for each Cgroup if the device is shared
  - At device level put a cap on the throughput

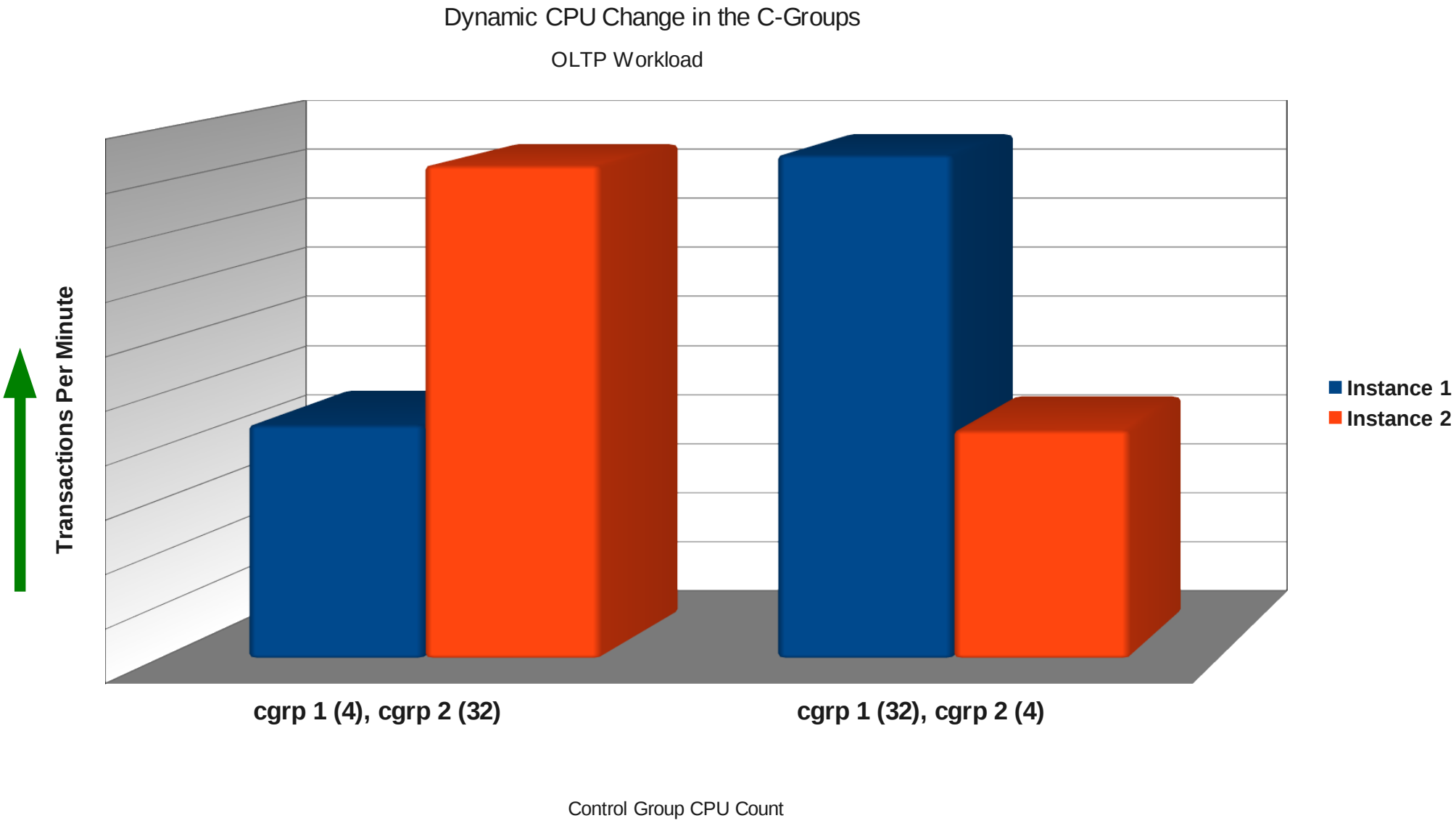
Cgroups – Resource management



Cgroups – NUMA pinning

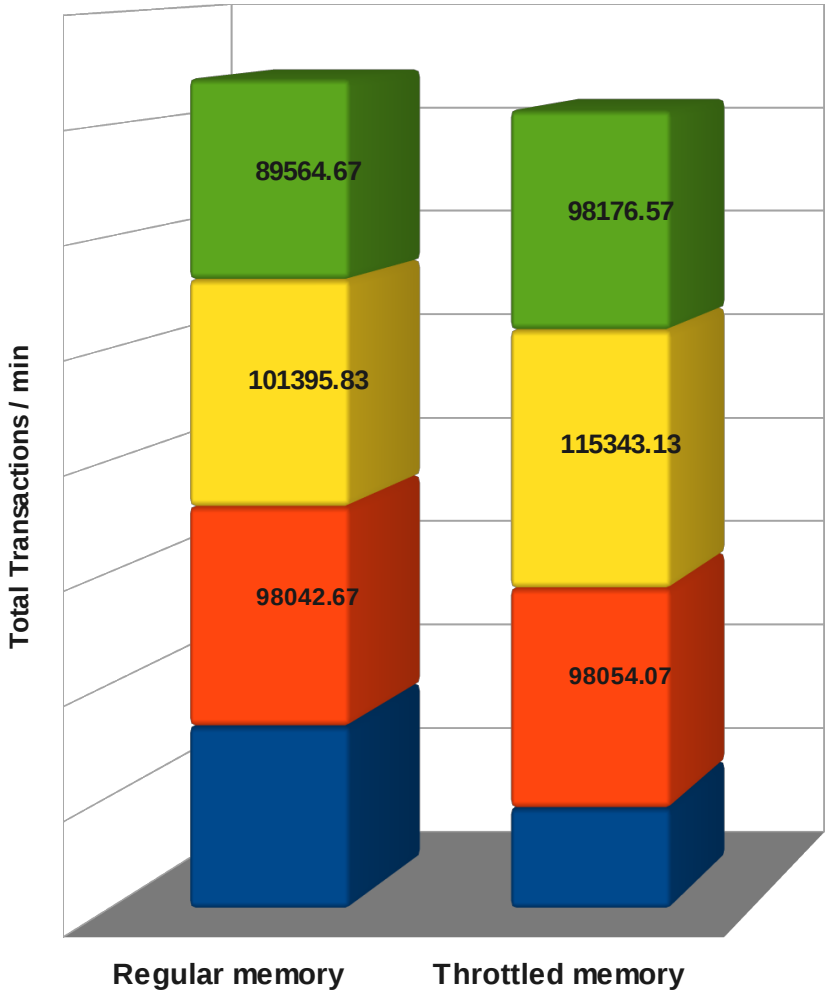


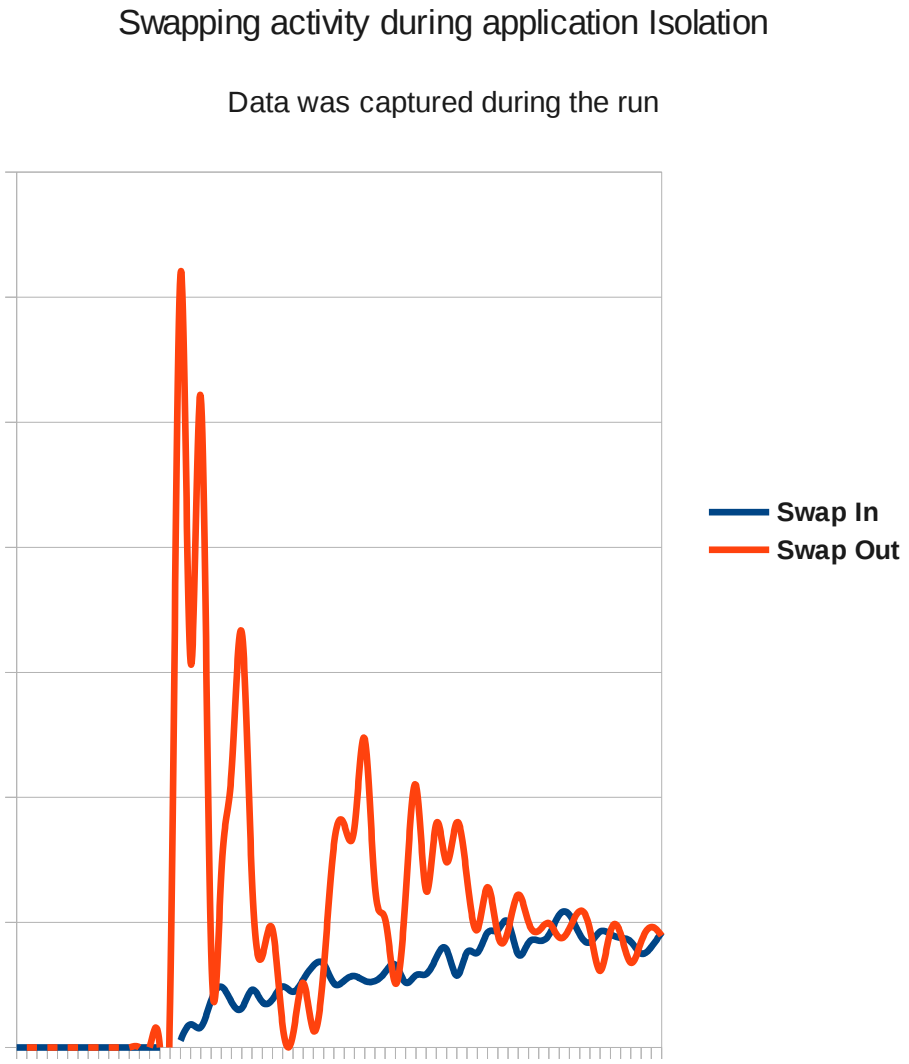
Cgroups – Dynamic Resource Control



Cgroups – Applicaton Isolation

Instance 1 was throttled to show that swapping within a C-group does not affect the performance of applications running in other C-groups





Quick Overview – KVM Architecture

- Guests run as a process in userspace on the host
- A virtual CPU is implemented using a Linux thread
  - The Linux scheduler is responsible for scheduling a virtual CPU, as it is a normal thread
- Guests inherit features from the kernel
  - NUMA
  - Huge Pages
  - Support for new hardware

Virtualization Tuning – Caching

Figure 1

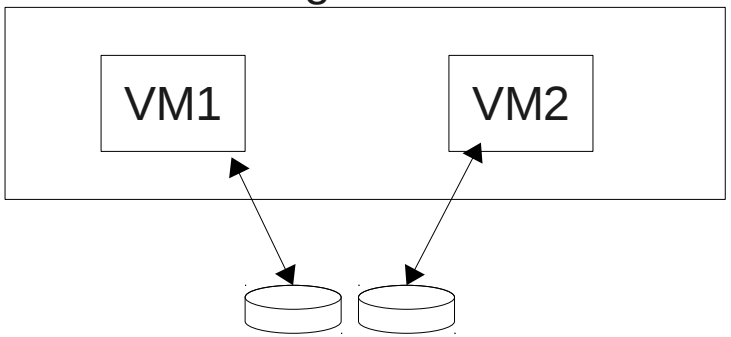
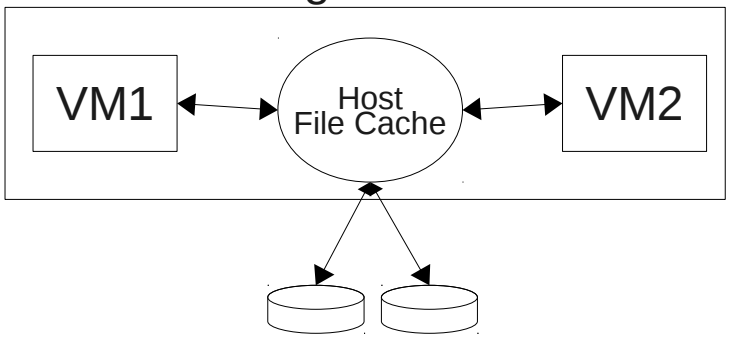
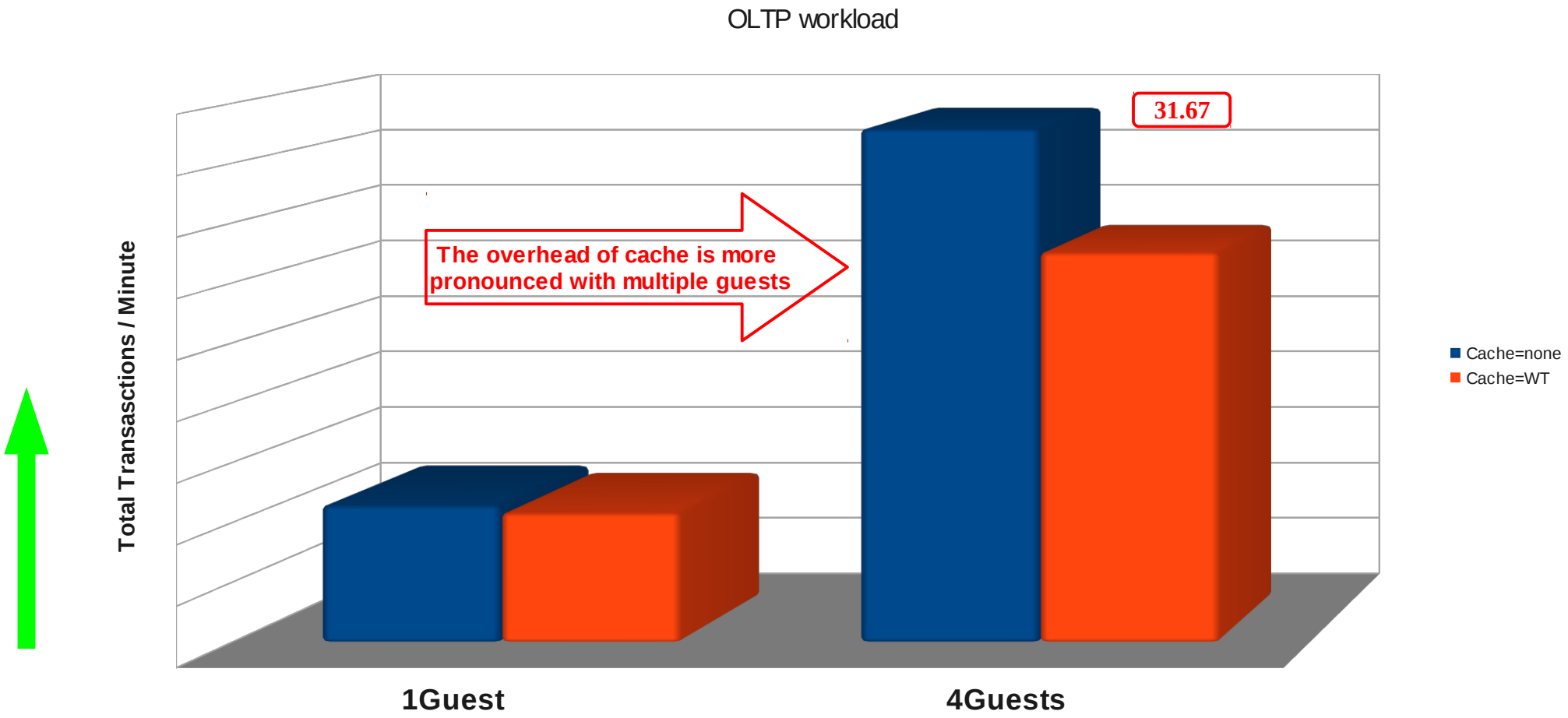


Figure 2



- **Cache = none (Figure 1)**
  - I/O from the guest is not cached on the host
- **Cache = writethrough (Figure 2)**
  - I/O from the guest is cached and written through on the host
    - Works well on large systems (lots of memory and CPU)
    - Potential scaling problems with this option with multiple guests (host CPU used to maintain cache)
    - Can lead to swapping on the host
- **How To**
    - Configure I/O - Cache per disk in qemu command line or libvirt

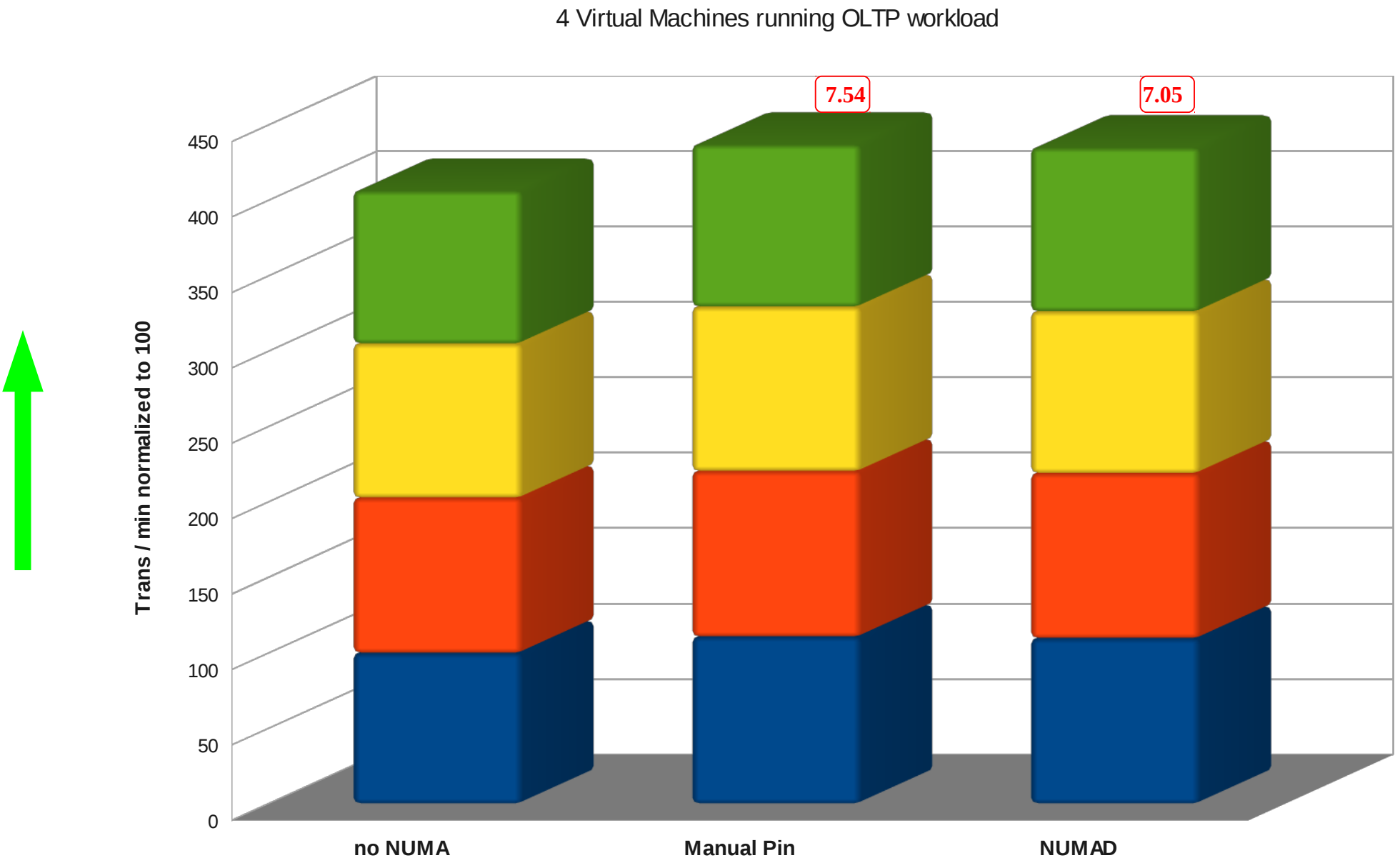
Virt Tuning – Effect of I/O Cache Settings



Configurable per device:

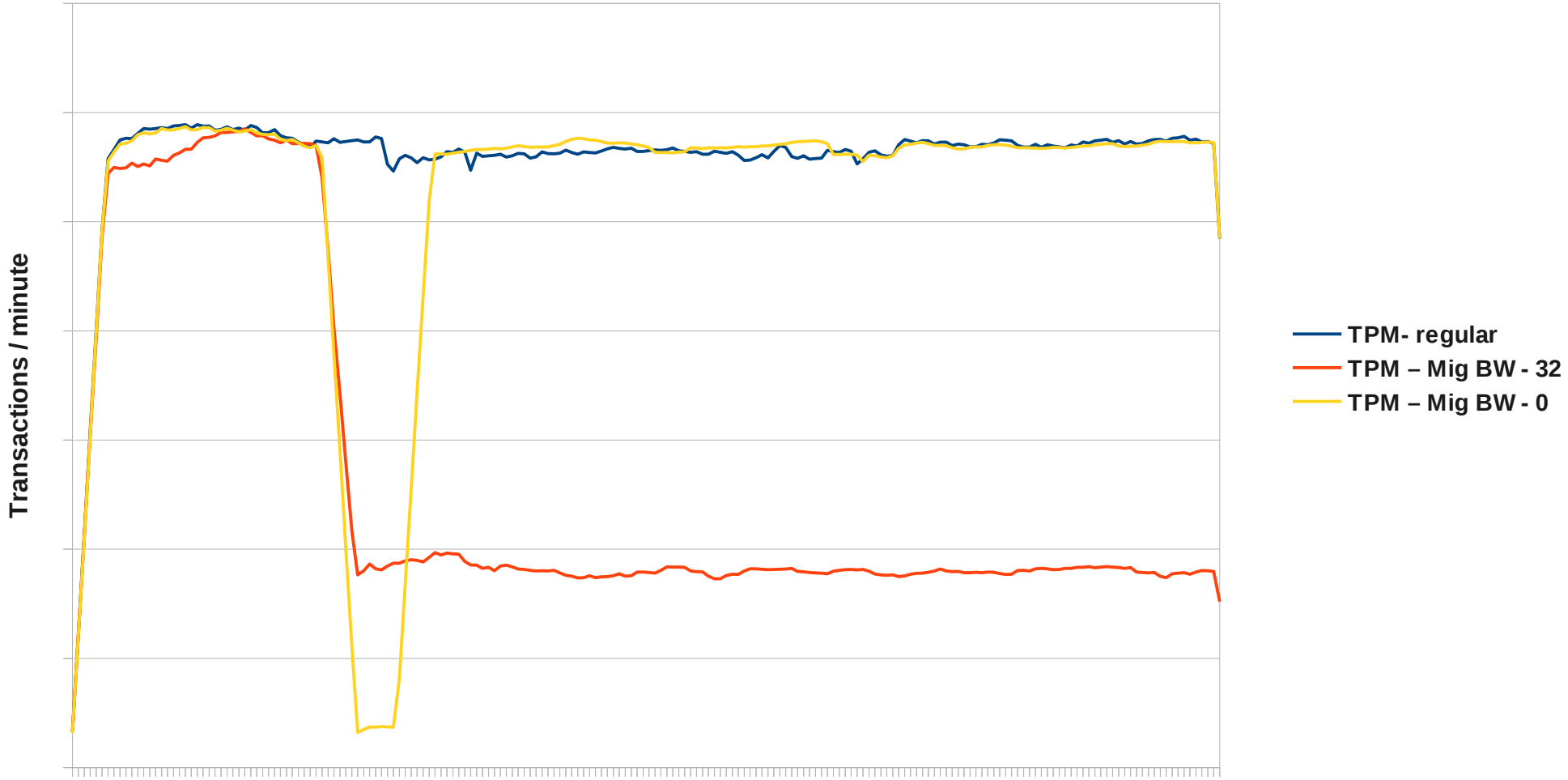
- **Virt-Manager - drop-down option under “Advanced Options”**
- **Libvirt xml file - driver name='qemu' type='raw' cache='writethrough' io='native'**

Virt Tuning – Using NUMA



RHEV – Migration

Migration tuning – configure migration bandwidth to facilitate migration

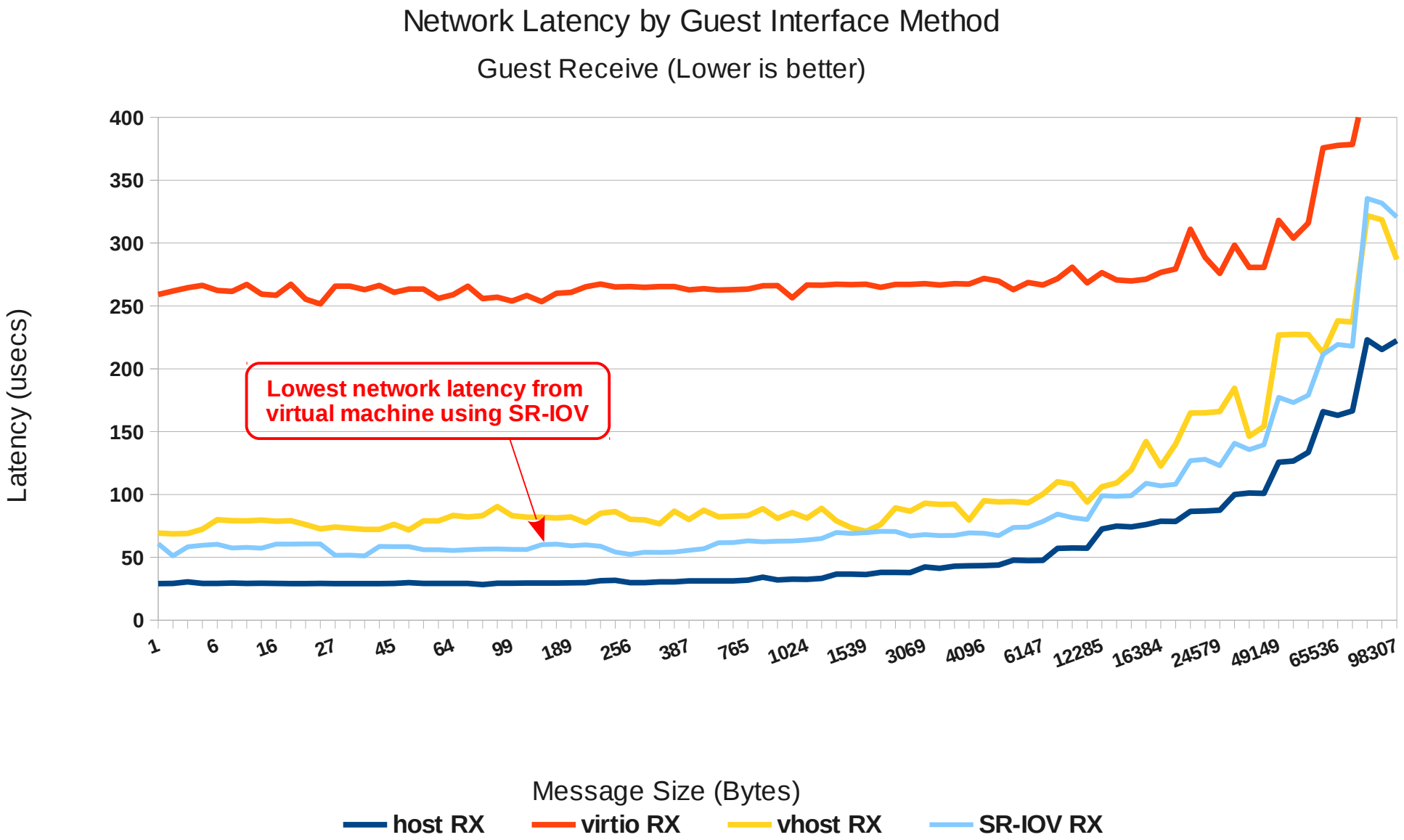


Configure – migration_max_bandwidth = <Value> in /etc/vdsm/vdsm.conf

Virtualization Tuning – Network

- VirtIO
  - ✔ VirtIO drivers for network
- vhost_net (low latency – close to line speed)
  - ✔ Bypass the qemu layer
- PCI pass through
  - ✔ Bypass the host and pass the PCI device to the guest
  - ✔ Can be passed only to one guest
- SR-IOV (Single root I/O Virtualization)
  - ✔ Pass through to the guest
  - ✔ Can be shared among multiple guests
  - ✔ Limited hardware support

Virtualization Tuning – Network – Latency Comparison



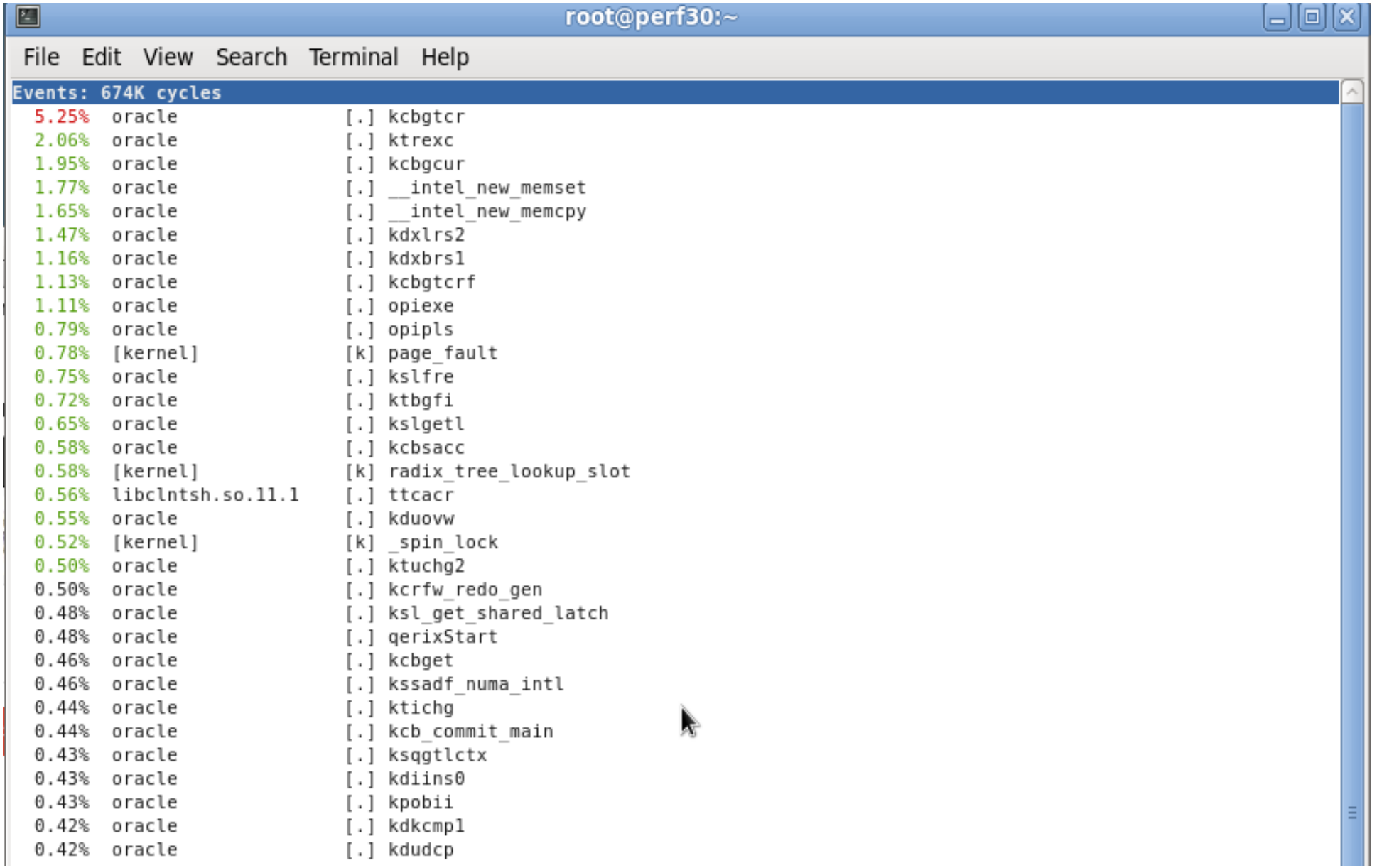
Performance Monitoring Tools

- Monitoring tools
  - top, vmstat, ps, iostat, netstat, sar, perf, turbostat
- Kernel tools
  - /proc, sysctl, AltSysRq
- Networking
  - ethtool, ifconfig
- Profiling
  - oprofile, strace, ltrace, systemtap, perf

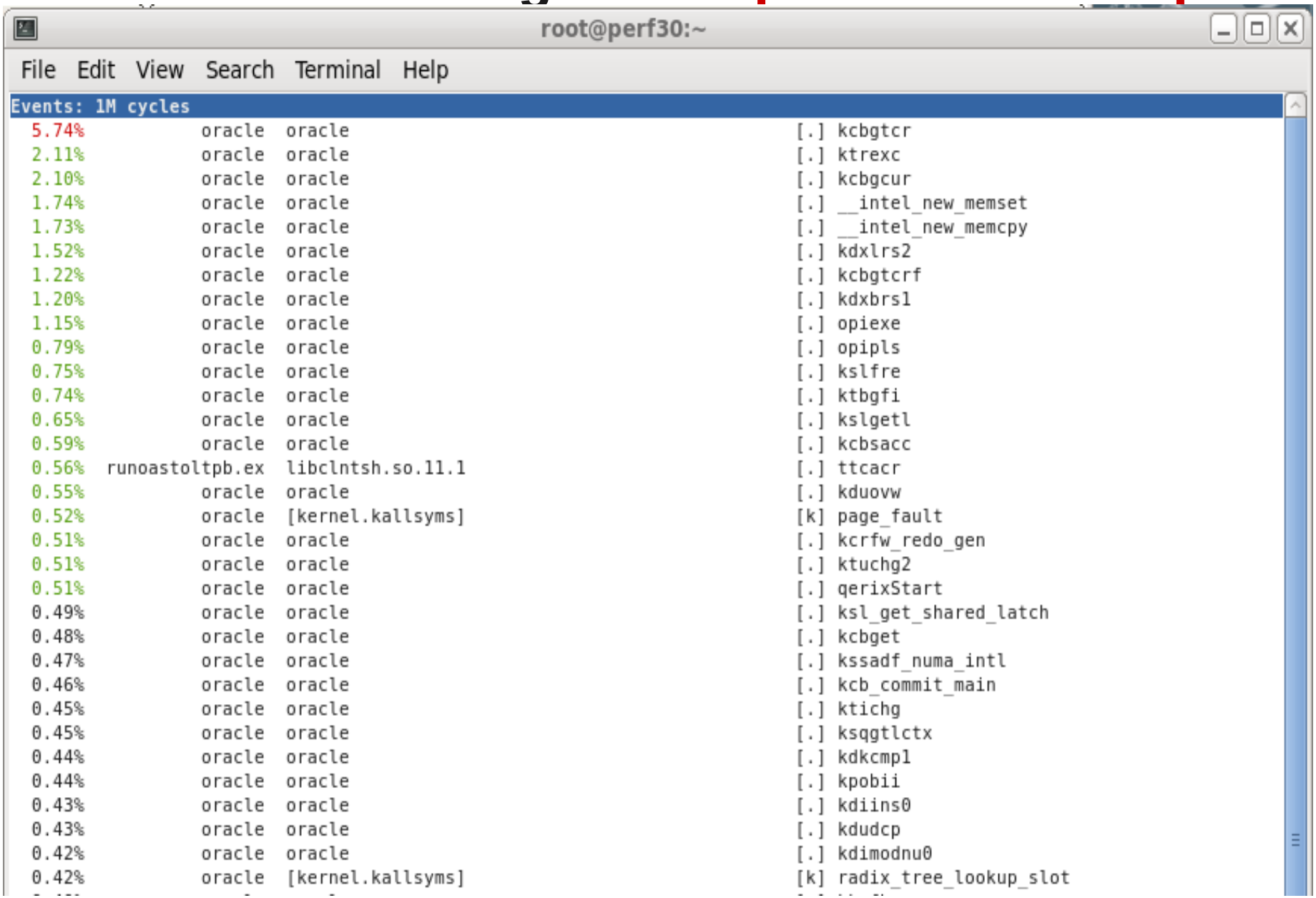
Performance Monitoring Tool – perf

- Performance analysis tool
  - perf top (dynamic)
  - perf record / report (save and replay)
  - perf stat <command> (analyze a particular workload)

Performance Monitoring Tool – perf top



Performance Monitoring Tool – perf record / report



Performance Monitoring Tool – perf stat

- perf stat <command>
  - monitors any workload and collects variety of statistics
  - can monitor specific events for any workload with -e flag (“perf list” give list of events)

“perf stat” - with regular 4k pages

```
oracle@perf30 ~/oast/home> perf stat ./database_workload_command

 Performance counter stats for './database_workload_command’

   1198816.385221 task-clock                #    2.690 CPUs utilized
      24,468,186 context-switches          #    0.020 M/sec
       3,603,875 CPU-migrations            #    0.003 M/sec
         282,197 page-faults               #    0.000 M/sec
2,589,984,107,267 cycles                   #    2.160 GHz                      [83.36%]
2,052,981,463,592 stalled-cycles-frontend  #   79.27% frontend cycles idle     [83.41%]
1,447,156,041,144 stalled-cycles-backend   #   55.88% backend  cycles idle     [66.62%]
  988,260,844,982 instructions             #    0.38  insns per cycle
                                           #    2.08  stalled cycles per insn [83.34%]
  195,178,277,195 branches                 #  162.809 M/sec                    [83.33%]
   14,063,695,242 branch-misses            #    7.21% of all branches          [83.29%]

    445.726643364 seconds time elapsed
```

Performance Monitoring Tool – perf stat

“perf stat” - with 2M huge pages

oracle@perf30 ~/oast/home> perf stat ./database_workload_command

Performance counter stats for './database_workload_command’

```
   1223064.068933 task-clock                #    2.726 CPUs utilized
      25,521,110 context-switches          #    0.021 M/sec
       4,242,520 CPU-migrations            #    0.003 M/sec
         151,366 page-faults               #    0.000 M/sec
2,640,419,666,995 cycles                    #    2.159 GHz                      [83.35%]
2,085,237,230,532 stalled-cycles-frontend   #   78.97% frontend cycles idle     [83.33%]
1,459,622,166,670 stalled-cycles-backend    #   55.28% backend  cycles idle     [66.68%]
1,020,193,451,957 instructions              #    0.39  insns per cycle
                                            #    2.04  stalled cycles per insn [83.32%]
  201,608,008,922 branches                  #  164.838 M/sec                    [83.36%]
   14,310,983,194 branch-misses             #    7.10% of all branches          [83.29%]

     448.643139666 seconds time elapsed
```

Performance Monitoring Tool – sar

Output of "sar -N DEV 3"

For a DSS workload running on iSCSI storage using different MTUs

1500 MTU

```
01:40:08 PM     IFACE   rxpck/s   txpck/s    rxkB/s    txkB/s   rxcmp/s   txcmp/s  rxmcst/s
01:40:11 PM      eth0      0.34      0.34      0.02      0.02      0.00      0.00      0.00
01:40:11 PM      eth5 135016.78  19107.72 199178.19   1338.53      0.00      0.00      0.34

01:40:14 PM      eth0      0.66      0.00      0.05      0.00      0.00      0.00      0.66
01:40:14 PM      eth5 133676.74  18911.30 197199.84   1310.25      0.00      0.00      0.66

01:40:17 PM      eth0      0.67      0.00      0.05      0.00      0.00      0.00      0.67
01:40:17 PM      eth5 134555.85  19045.15 198502.27   1334.19      0.00      0.00      0.33

01:40:20 PM      eth0      1.00      0.00      0.07      0.00      0.00      0.00      0.67
01:40:20 PM      eth5 134116.33  18972.33 197849.55   1325.03      0.00      0.00      1.00
```

9000 MTU

```
06:58:43 PM     IFACE   rxpck/s   txpck/s    rxkB/s    txkB/s   rxcmp/s   txcmp/s  rxmcst/s
06:58:46 PM      eth0      0.91      0.00      0.07      0.00      0.00      0.00      0.00
06:58:46 PM      eth5 104816.36  48617.27 900444.38   3431.15      0.00      0.00      0.91

06:58:49 PM      eth0      0.00      0.00      0.00      0.00      0.00      0.00      0.00
06:58:49 PM      eth5 118269.80  54965.84 1016151.64   3867.91      0.00      0.00      0.50

06:58:52 PM      eth0      0.00      0.00      0.00      0.00      0.00      0.00      0.00
06:58:52 PM      eth5 118470.73  54382.44 1017676.21   3818.35      0.00      0.00      0.98

06:58:55 PM      eth0      0.94      0.00      0.06      0.00      0.00      0.00      0.00
06:58:55 PM      eth5 115853.05  53515.49 995087.67   3766.28      0.00      0.00      0.47
```

Performance Monitoring Tool – vmstat

Output of “vmstat -n 3”

Procs		----------memory----------				---swap--		-----io----		--system--		-----cpu-----				
r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id	wa	st
25	1	39748	1102508	170976	23568380	0	0	7952	163312	40235	97711	81	11	7	1	0
39	4	39748	1081552	171036	23568948	0	0	8150	162642	40035	97585	82	11	6	1	0
54	0	39748	1071600	171064	23569452	0	0	7498	166835	40494	97413	82	12	6	1	0
10	0	39748	1077892	171104	23569980	0	0	6841	159781	40150	95170	83	12	5	1	0
49	2	39748	1139520	171128	23570568	0	0	5950	138597	40117	94040	85	12	4	0	0
46	1	39748	1192436	171144	23571136	0	0	5895	139294	40487	94423	84	12	4	0	0
40	3	39748	1213212	171168	23571660	0	0	5906	136871	40401	94313	84	11	4	0	0
50	2	39748	1210840	171176	23572212	0	0	5890	135288	40744	95360	84	12	4	0	0
52	1	39748	1090252	171200	23572732	0	0	7866	174045	39702	97930	80	11	8	1	0
46	3	39748	1082532	171244	23573300	0	0	7217	174223	40469	95697	82	11	5	1	0
59	3	39748	1129396	171268	23573892	0	0	5682	218917	41571	94576	84	12	4	1	0
46	7	39748	1159372	171284	23574428	0	0	5488	357287	45871	96181	83	12	4	1	0
46	3	39748	1196788	171328	23574912	0	0	5456	257984	45617	97021	84	12	3	1	0
51	0	39748	1199880	171336	23575584	0	0	5518	161104	41572	96639	84	12	4	1	0
42	3	39748	1198720	171352	23576148	0	0	5440	159580	41191	95308	85	11	3	0	0

Performance Monitoring Tool – iostat

Output of “iostat -dmxz 3”

```
Device:         rrqm/s   wrqm/s     r/s     w/s    rMB/s    wMB/s avgrq-sz avgqu-sz   await  svctm  %util
sda               0.00    25.20    0.00    2.40     0.00     0.11    90.67     0.04   17.50  11.50   2.76
dm-0              0.00     0.00    0.00    1.00     0.00     0.00     8.00     0.05   47.00  15.20   1.52
dm-2              0.00     0.00    0.00   26.20     0.00     0.10     8.00     0.43   16.43   0.47   1.24
fioa              0.00    41.80 1057.60 3747.60    28.74   114.75    61.16     1.72    0.36   0.16  76.88

Device:         rrqm/s   wrqm/s     r/s     w/s    rMB/s    wMB/s avgrq-sz avgqu-sz   await  svctm  %util
sda               0.00     2.99    0.00    3.19     0.00     0.02    15.00     0.01    4.50   4.44   1.42
dm-2              0.00     0.00    0.00    5.99     0.00     0.02     8.00     0.01    2.43   2.37   1.42
fioa              0.00    32.93  950.70 3771.46    25.33   127.18    66.14     1.77    0.38   0.16  76.57

Device:         rrqm/s   wrqm/s     r/s     w/s    rMB/s    wMB/s avgrq-sz avgqu-sz   await  svctm  %util
sda               0.00    22.80    0.00    1.60     0.00     0.09   121.00     0.01    6.25   6.12   0.98
dm-2              0.00     0.00    0.00   24.20     0.00     0.09     8.00     0.11    4.69   0.40   0.98
fioa              0.00    40.00  915.00 3868.60    24.10   118.31    60.97     1.63    0.34   0.16  75.34

Device:         rrqm/s   wrqm/s     r/s     w/s    rMB/s    wMB/s avgrq-sz avgqu-sz   await  svctm  %util
sda               0.00    54.20    0.00    1.60     0.00     0.22   278.00     0.01    6.00   5.00   0.80
dm-2              0.00     0.00    0.00   55.60     0.00     0.22     8.00     0.24    4.26   0.14   0.80
fioa              0.00    39.80  862.00 3800.60    21.93   131.67    67.47     1.72    0.37   0.16  75.96

Device:         rrqm/s   wrqm/s     r/s     w/s    rMB/s    wMB/s avgrq-sz avgqu-sz   await  svctm  %util
sda               0.00     2.40    0.00    0.80     0.00     0.01    30.00     0.01    6.75   6.75   0.54
dm-2              0.00     0.00    0.00    3.00     0.00     0.01     8.00     0.01    1.80   1.80   0.54
fioa              0.00    36.00  811.20 3720.80    20.74   116.78    62.15     1.56    0.34   0.16  72.72
```

Performance Monitoring Tool – turbostats (Intel only)

```
# turbostat -i 3
pkg core CPU    %c0   GHz  TSC    %c1    %c3    %c6   %pc3   %pc6
                0.23 1.08 2.26  99.77   0.00   0.00   0.00   0.00
   0    0    0   0.10 1.06 2.26  99.90   0.00   0.00   0.00   0.00
   0    1    4   0.20 1.06 2.26  99.80   0.00   0.00   0.00   0.00
   0    2    8   1.51 1.06 2.26  98.49   0.00   0.00   0.00   0.00
   0    3   12   0.02 1.06 2.26  99.98   0.00   0.00   0.00   0.00
   0    8   16   0.01 1.06 2.26  99.99   0.00   0.00   0.00   0.00
   0    9   20   0.01 1.06 2.26  99.99   0.00   0.00   0.00   0.00
   0   10   24   0.01 1.06 2.26  99.99   0.00   0.00   0.00   0.00
   0   11   28   0.01 1.06 2.26  99.99   0.00   0.00   0.00   0.00
   1    0    2   0.06 1.06 2.26  99.94   0.00   0.00   0.00   0.00
   1    1    6   0.20 1.06 2.26  99.80   0.00   0.00   0.00   0.00
   1    2   10   0.01 1.06 2.26  99.99   0.00   0.00   0.00   0.00
   1    3   14   0.01 1.06 2.26  99.99   0.00   0.00   0.00   0.00
   1    8   18   0.01 1.06 2.26  99.99   0.00   0.00   0.00   0.00
   1    9   22   0.00 1.06 2.26 100.00   0.00   0.00   0.00   0.00
   1   10   26   0.00 1.06 2.26 100.00   0.00   0.00   0.00   0.00
   1   11   30   0.00 1.06 2.26 100.00   0.00   0.00   0.00   0.00
   2    0    1   0.05 1.06 2.26  99.95   0.00   0.00   0.00   0.00
   2    1    5   0.06 1.06 2.26  99.94   0.00   0.00   0.00   0.00
   2    2    9   0.01 1.06 2.26  99.99   0.00   0.00   0.00   0.00
   2    3   13   0.01 1.06 2.26  99.99   0.00   0.00   0.00   0.00
   2    8   17   0.00 1.07 2.26 100.00   0.00   0.00   0.00   0.00
   2    9   21   0.00 1.06 2.26 100.00   0.00   0.00   0.00   0.00
   2   10   25   0.00 1.06 2.26 100.00   0.00   0.00   0.00   0.00
   2   11   29   0.00 1.06 2.26 100.00   0.00   0.00   0.00   0.00
   3    0    3   1.89 1.08 2.26  98.11   0.00   0.00   0.00   0.00
   3    1    7   2.43 1.09 2.26  97.57   0.00   0.00   0.00   0.00
```

The tool is found in cpupowerutils.x86_64 in RHEL6.4

Wrap up – Bare Metal

- I/O
  - Choose the right elevator
  - Eliminated hot spots
  - Direct I/O or Asynchronous I/O
  - Virtualization – Caching
- Memory
  - NUMA
  - Huge Pages
  - Swapping
  - Managing Caches
- RHEL has many tools to help with debugging / tuning

Wrap Up – Bare Metal

- CPU
  - Check cpuspeed settings
- Network
  - Separate networks
  - arp_filter
  - Packet size

New Tools

- tuned
- perf options - top, stats, record, report
- turbostats (Intel only)

Wrap Up – Virtualization

- VirtIO drivers
- aio (native)
- NUMA
- Cache options (none, writethrough)
- Network (vhost-net)

Stay connected through the Red Hat Customer Portal

Maximizing Oracle Database Efficiency for RHEL

Watch video

Improving OpenJDK and Oracle JDK Garbage Collection Performance

Review Tech brief

access.redhat.com