Red Hat Linux Performance

Marc Skinner

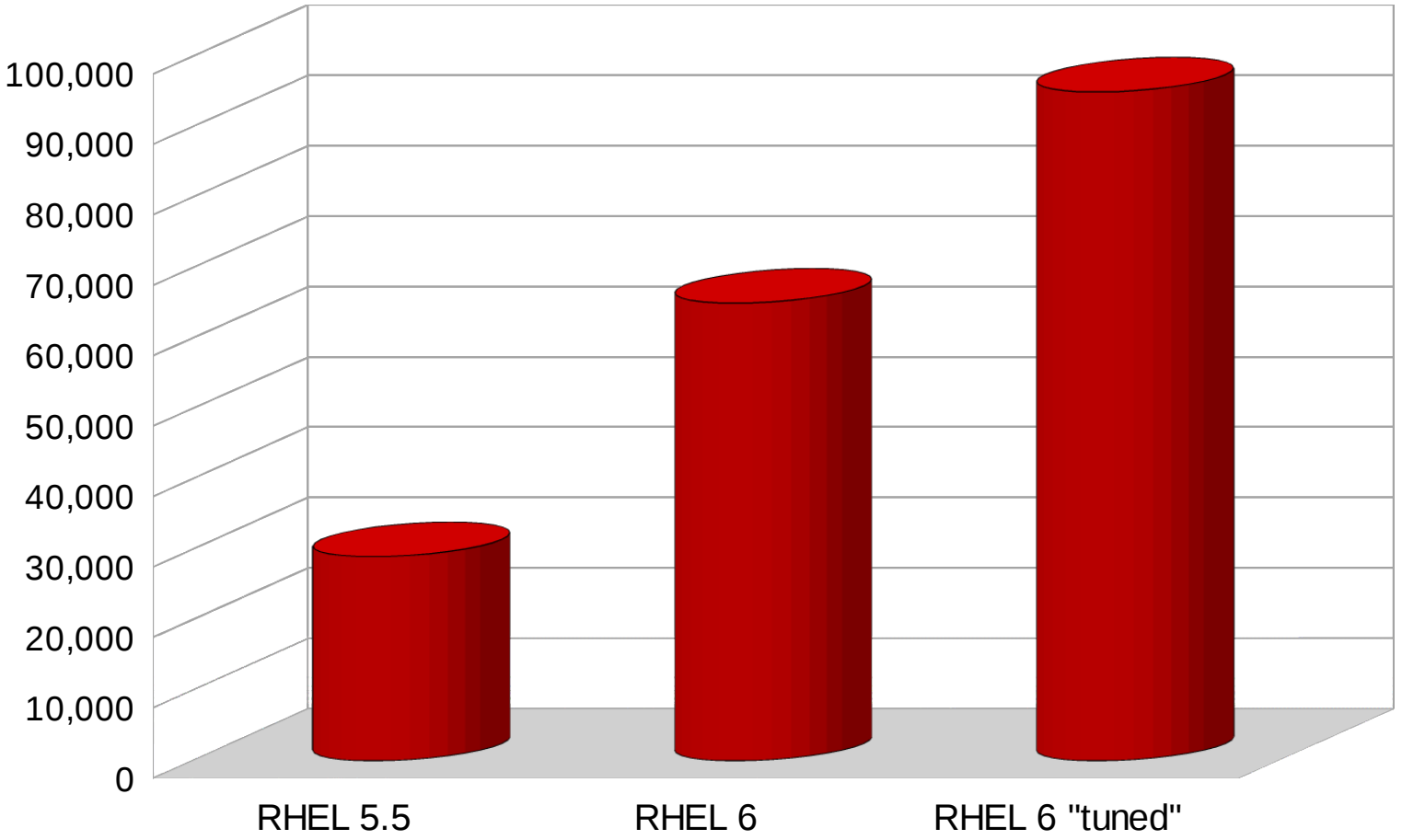
Performance in the past

- Recompile the kernel
- Have your masters or PHD in performance theory
- Be a kernel developer

RHEL 6 now includes tuning resources

- Tuned – kernel and block device optimizations
- Transparent Huge Pages – memory optimizations
- Numad – multi-socket process and memory optimizations
- CPU power scaling

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



HP DL980 64-core/256GB/30 FC/480 lun AIM7 results w/ “tuned”
AIM7 file server performance – jobs per minute

Tuned

The “tuned” package

```
yum -y install tuned

# tuned-adm list
Available profiles:
- virtual-host
- laptop-ac-powersave
- virtual-guest
- server-powersave
- desktop-powersave
- sap
- default
- throughput-performance
- latency-performance
- laptop-battery-powersave
- spindown-disk
- enterprise-storage

Current active profile: virtual-host
```

```
# tuned-adm list
# tuned-adm active
# tuned-adm profile <profile-name>
# tuned-adm off
```

Change profiles with cron

Create a new cronjob

0 8 * * 1-5 /usr/bin/tuned-adm profile throughput-performance
0 18 * * 1-5 /usr/bin/tuned-adm profile server-powersave

Create your own profiles

/etc/tune-profiles/<*profile-name*>

tuned.conf
ktune.sysconfig
sysctl.ktune
ktune.sh

tuned.conf

Enable/Disable Power Management Plugins
- CPU
- Disk
- Network

Example: Aggressive Link Power Management

ktune.sysconfig

Enable/Disable ktune daemon
Configure disk elevators

Disk Elevators

- `deadline`
- `cfq` [DEFAULT]
- `noop`

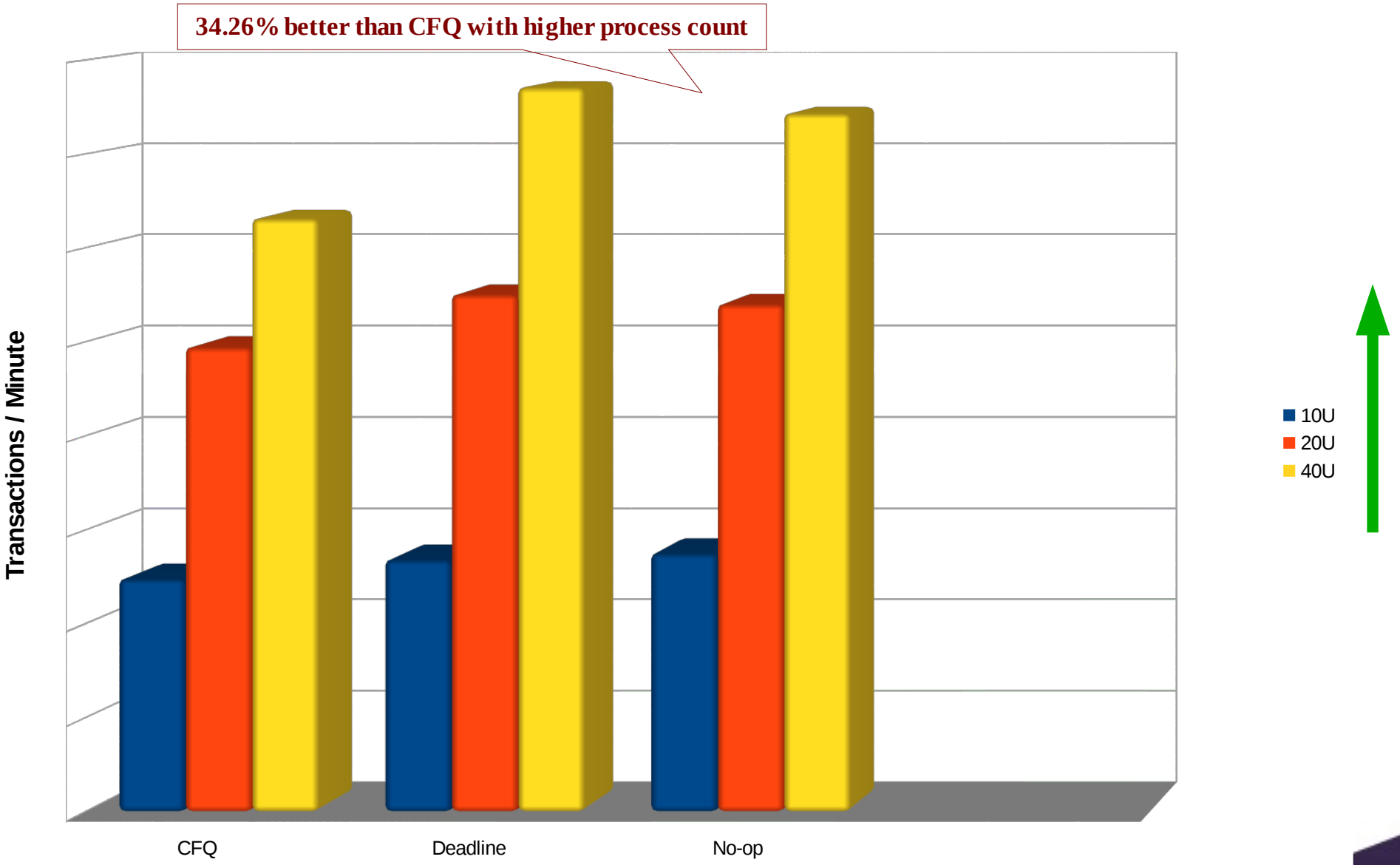
I/O Tuning – Understanding I/O Elevators

- **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)

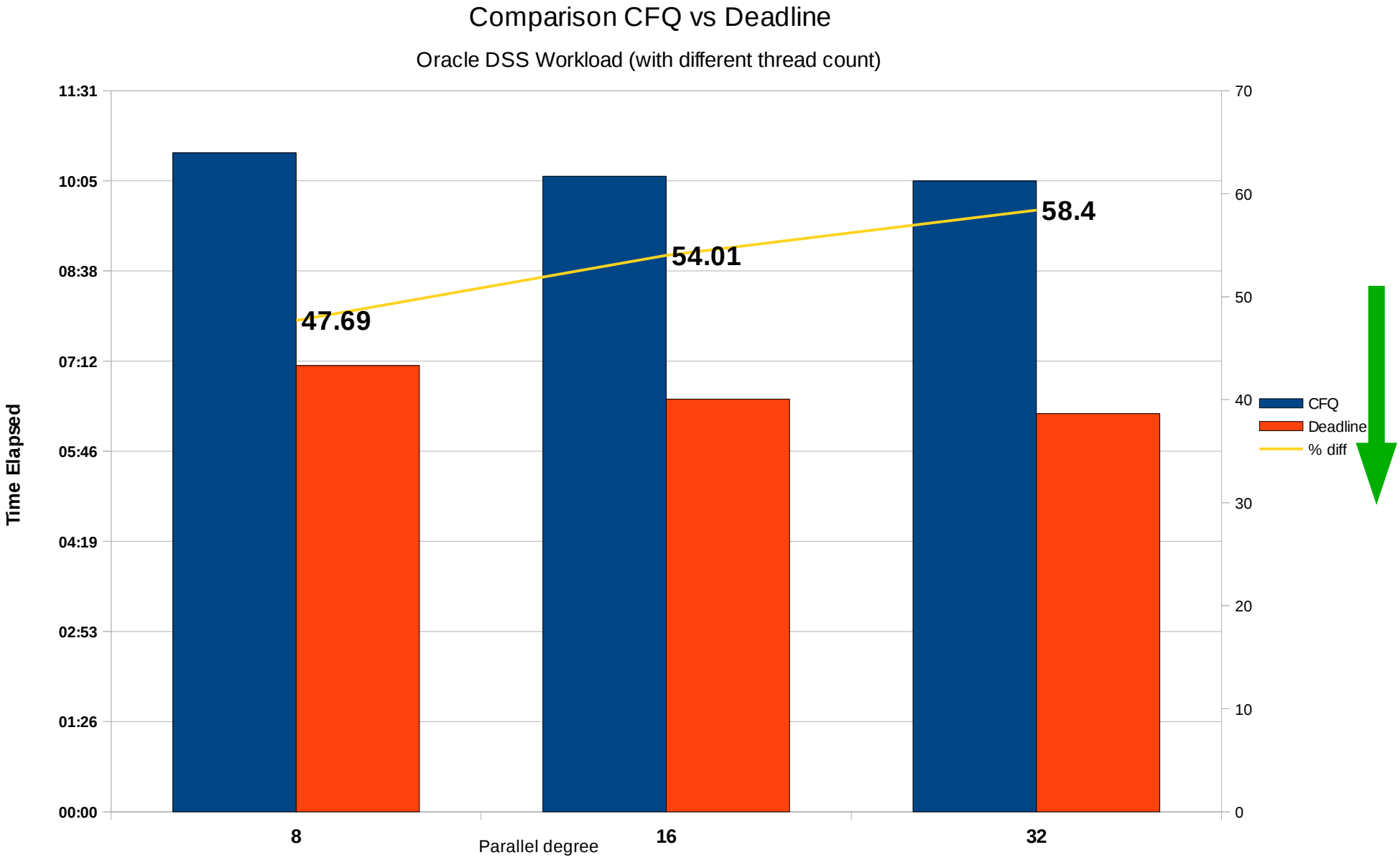
I/O Tuning – Configuring I/O Elevators

- **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



sysctl.ktune

`sysctl` settings

ktune.sh

start() and stop()
/etc/tune-profiles/functions

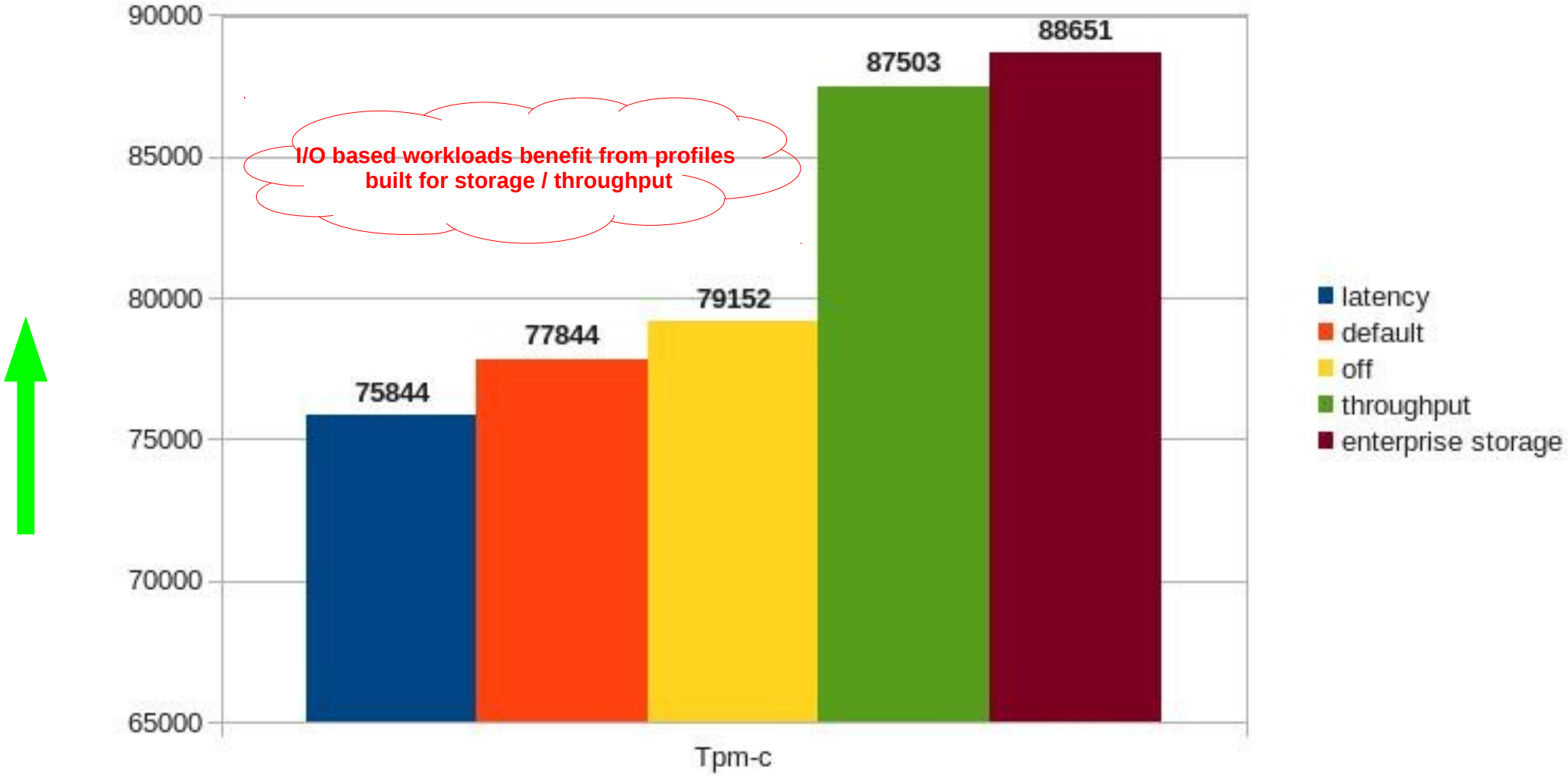
/etc/tune-profiles/functions

- {set,restore)_transparent_hugepages
- {set,restore}_cpu_governor
- {enable,disable}_wifi
- {enable,disable}_cd_polling
- And many more

“tuned” Profile Summary

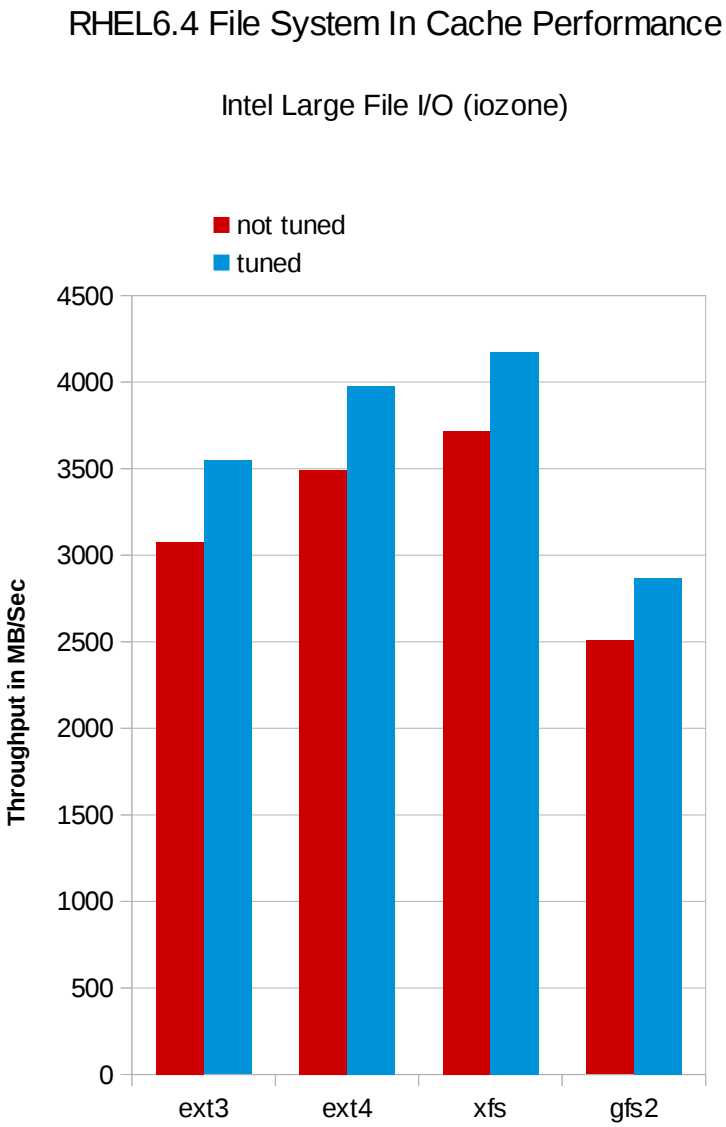
Tunable	default	enterprise-storage	virtual-host	virtual-guest	latency-performance	throughput-performance
kernel.sched_min_granularity_ns	4ms	10ms	10ms	10ms		10ms
kernel.sched_wakeup_granularity_ns	4ms	15ms	15ms	15ms		15ms
vm.dirty_ratio	20% RAM	40%	10%	40%		40%
vm.dirty_background_ratio	10% RAM		5%	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

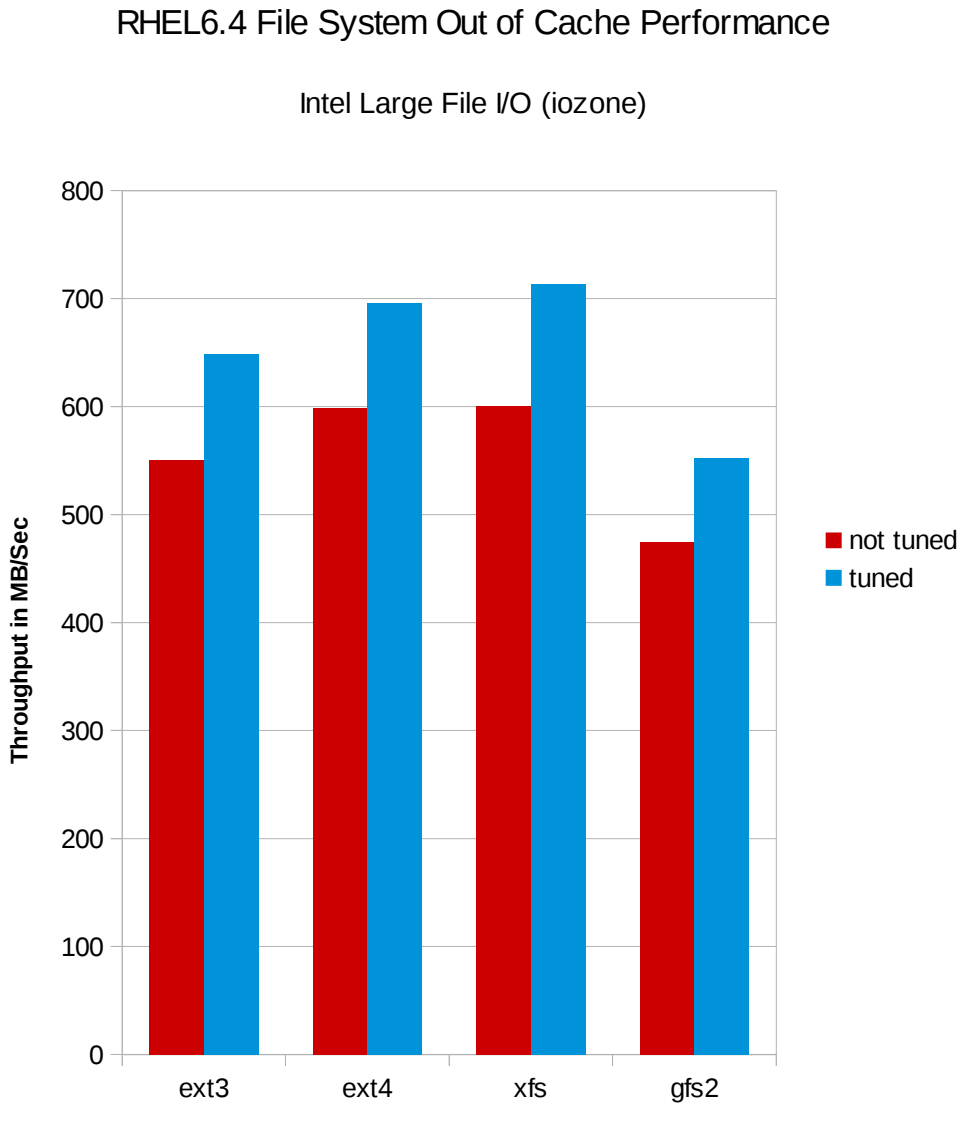
Tuned – BenchmarkSQL 2.3.3 on PostgreSQL 9.2



%15 increase

Iozone Performance Effect of TUNED EXT4/XFS/GFS





Approx %15 increase

Transparent Huge Pages

What is a huge page?

- Memory is managed in pages, traditionally 4K in size
- A huge page can be 2MB or now 1GB with Intel Sandy Bridge and newer
- Less pages, means faster look ups, less misses = performance

Transparent Hugepages

echo never > /sys/kernel/mm/transparent_hugepages=never

```
[root@dhcp-100-19-50 code]# time ./memory 15 0
real    0m12.434s
user    0m0.936s
sys     0m11.416s
```

```
# cat /proc/meminfo
MemTotal:       16331124 kB
AnonHugePages:  0 kB
```

- Boot argument: transparent_hugepages=always (enabled by default)
- # echo always > /sys/kernel/mm/redhat_transparent_hugepage/enabled

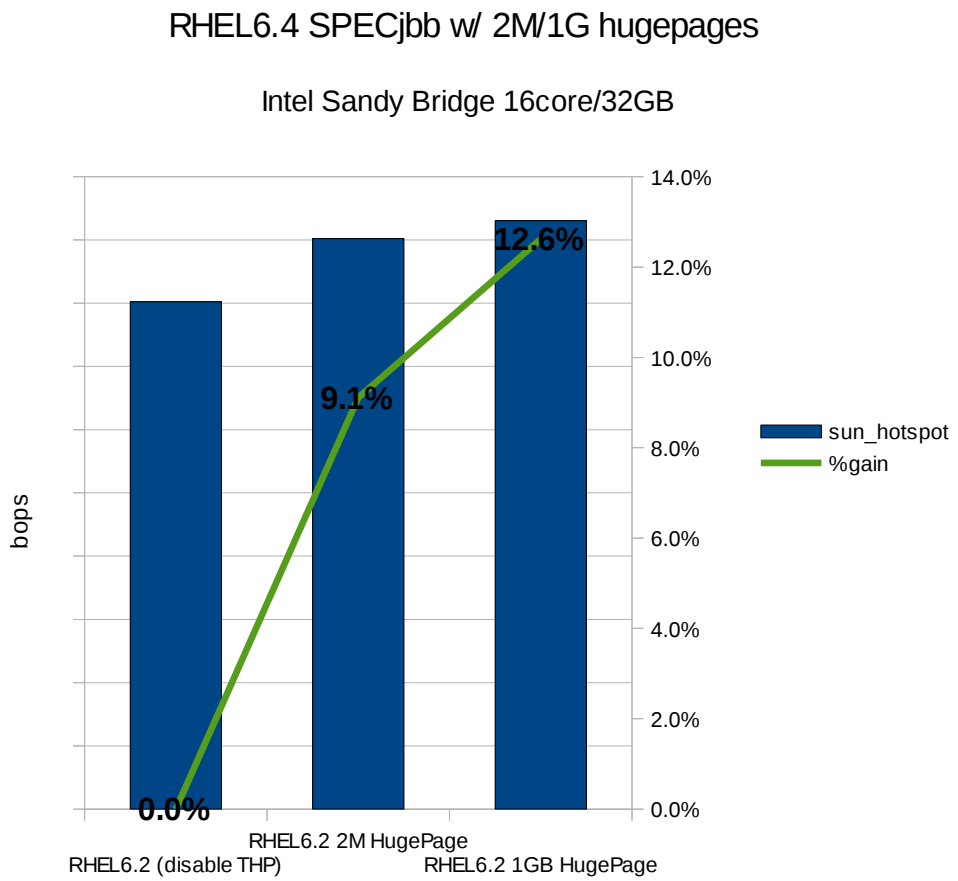
```
# time ./memory 15GB
  real    0m7.024s
  user    0m0.073s
  sys     0m6.847s
```

```
# cat /proc/meminfo
MemTotal:       16331124 kB
AnonHugePages:  15590528 kB
```

SPEEDUP 12.4/7.0 = 1.77x, 56%

Red Hat Enterprise Linux 6 Performance / Sandy Bridge Specjbb Java w/ 1GB huge pages

- Sandy Bridge has 1GB hugepages
  - Support in RHEL5.8 and 6.2
- RHEL6. Transparent Huge pages
  - Use 2M x86_64 page vs 4k page
  - < RHEL6, static use of hugepages
    - Static pages wired-down
    - Need application support DB/Java etc
  - Automatically use huge pages
    - For all annonmous memory
  - Daemon to gather free dynamically
  - %9.1 and %12.6 gain!



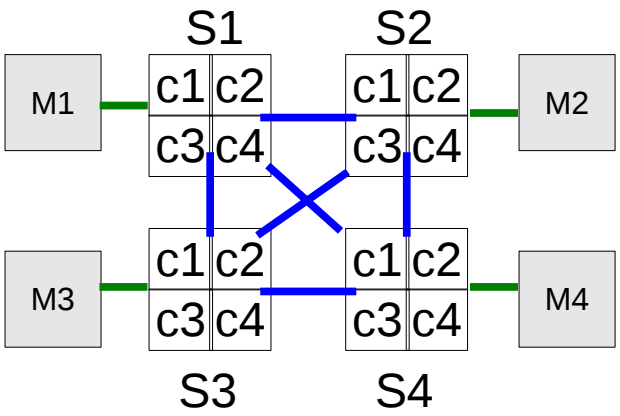
Huge Pages - FYI

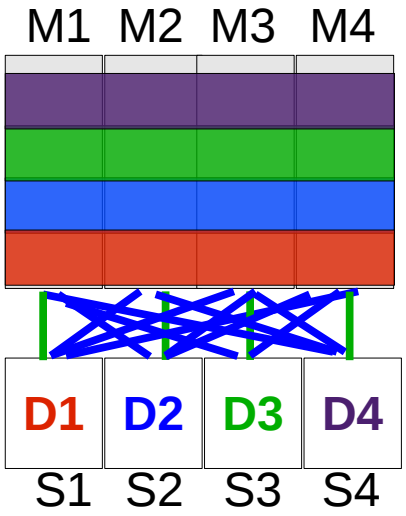
- Most databases support Huge pages
- Transparent Huge Pages in RHEL6 (cannot be used for Database shared memory – only for process private memory)

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

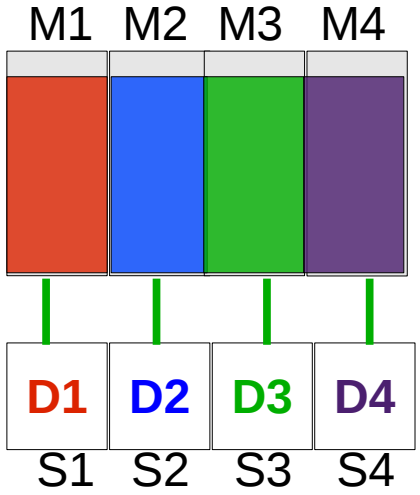
Numad

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

Finding NUMA layout – 4 socket by 16 cores

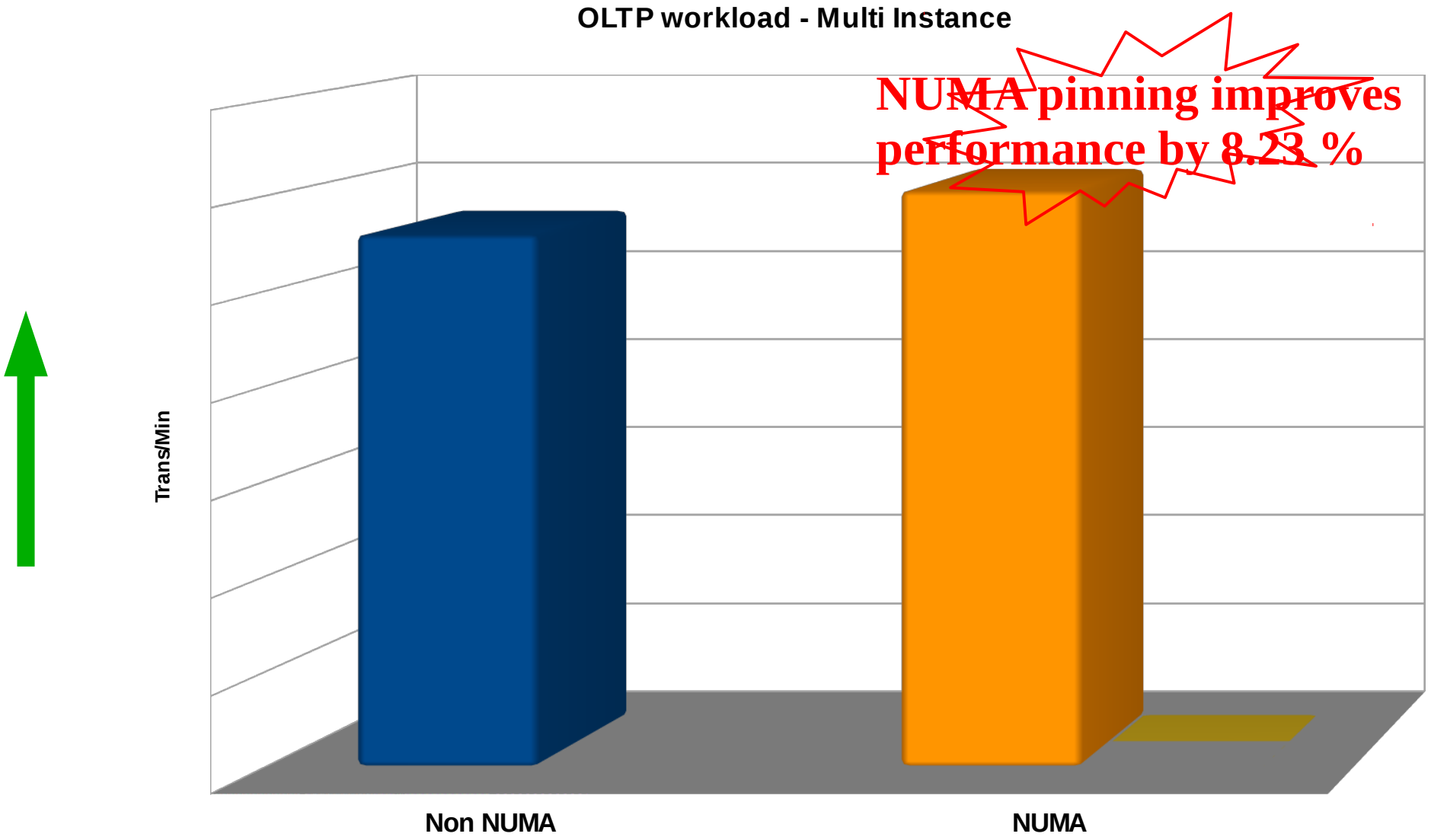
```
[root@perf30 ~]# numactl --hardware
available: 4 nodes (0-3)
node 0 cpus: 0 4 8 12 16 20 24 28 32 36 40 44 48 52 56 60
node 0 size: 32649 MB
node 0 free: 30868 MB
node 1 cpus: 1 5 9 13 17 21 25 29 33 37 41 45 49 53 57 61
node 1 size: 32768 MB
node 1 free: 29483 MB
node 2 cpus: 2 6 10 14 18 22 26 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
```

NUMA Affinity old way

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

- Sets CPU affinity for 'command' to CPU node 1
- Allocates memory out of Memory node 1

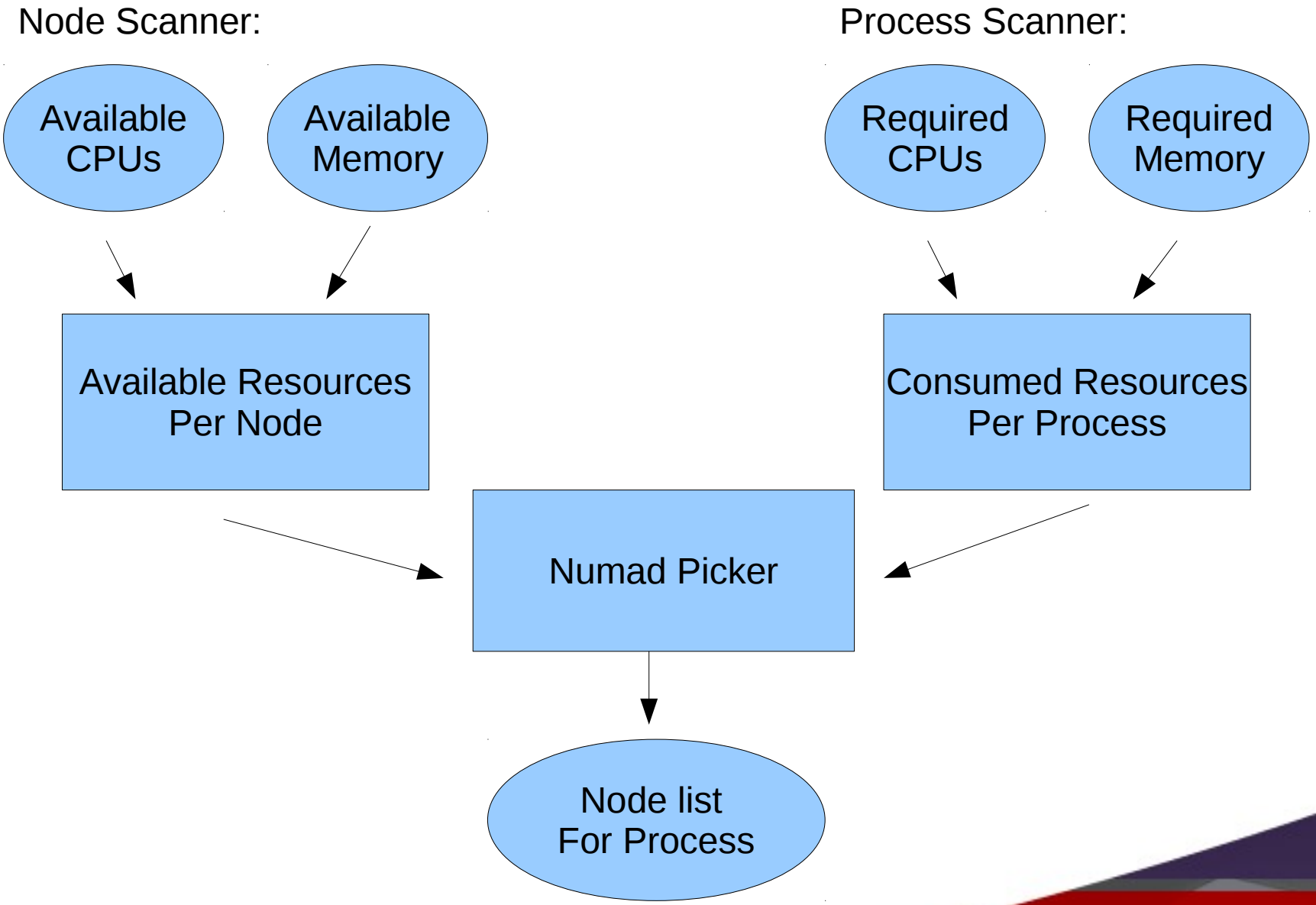
Memory Tuning – Effect of NUMA Tuning



Memory Tuning – NUMA - “numad”

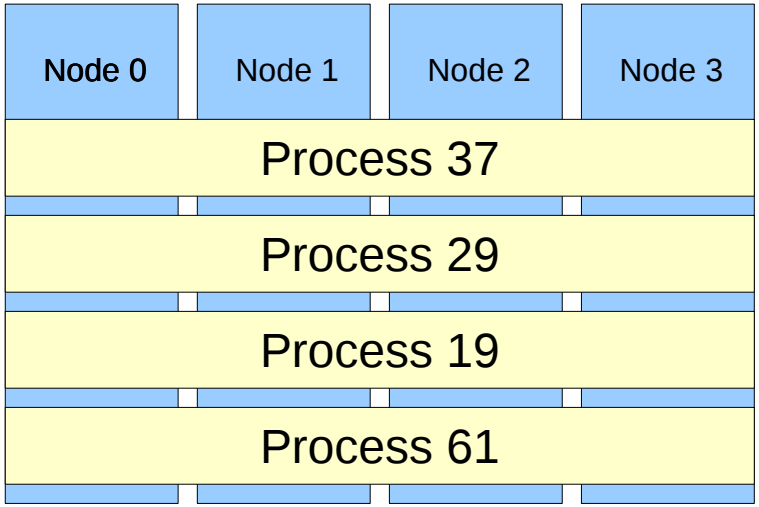
- **What is numad?**
  - User-level daemon to automatically improve out of the box NUMA system performance
  - Added to Fedora 17
  - Added to RHEL 6.3 as tech preview
  - Not enabled by default
- **What does numad do?**
  - Monitors available system resources on a per-node basis and assigns significant 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.

numad matches resource consumers with available resources

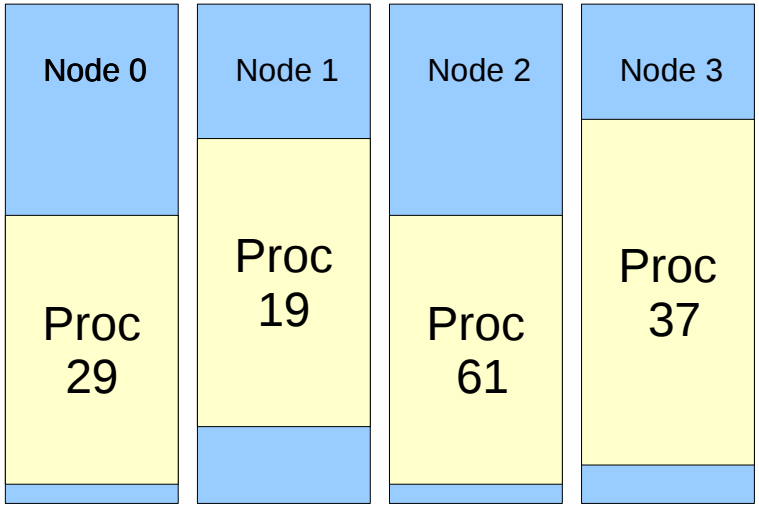


numad aligns process memory and CPU threads within nodes

Before numad



After numad



So, what's the NUMA problem?

- The Linux system scheduler is very good at maintaining responsiveness and optimizing for CPU utilization
- Tries to use idle CPUs, regardless of where process memory is located.... Using remote memory degrades performance!
  - Red Hat is working with the upstream community to increase NUMA awareness of the scheduler and to implement automatic NUMA balancing.
- Remote memory latency matters most for long-running, significant processes, e.g., HPTC, VMs, etc.

Use numastat to see memory layout

- Rewritten for RHEL6.4 to show per-node system and process memory information
- 100% compatible with prior version by default, displaying /sys...node<n>/numastat memory allocation statistics
- Any command options invoke new functionality
  - -m for per-node system memory info
  - <pattern> for per-node process memory info

numastat shows unaligned guests

```
# numastat -c qemu

Per-node process memory usage (in Mbs)

PID              Node 0 Node 1 Node 2 Node 3 Total
---------------  ------ ------ ------ ------ -----
10587 (qemu-kvm)   1216   4022   4028   1456 10722
10629 (qemu-kvm)   2108     56    473   8077 10714
10671 (qemu-kvm)   4096   3470   3036    110 10712
10713 (qemu-kvm)   4043   3498   2135   1055 10730
---------------  ------ ------ ------ ------ -----
Total             11462  11045   9672  10698 42877
```

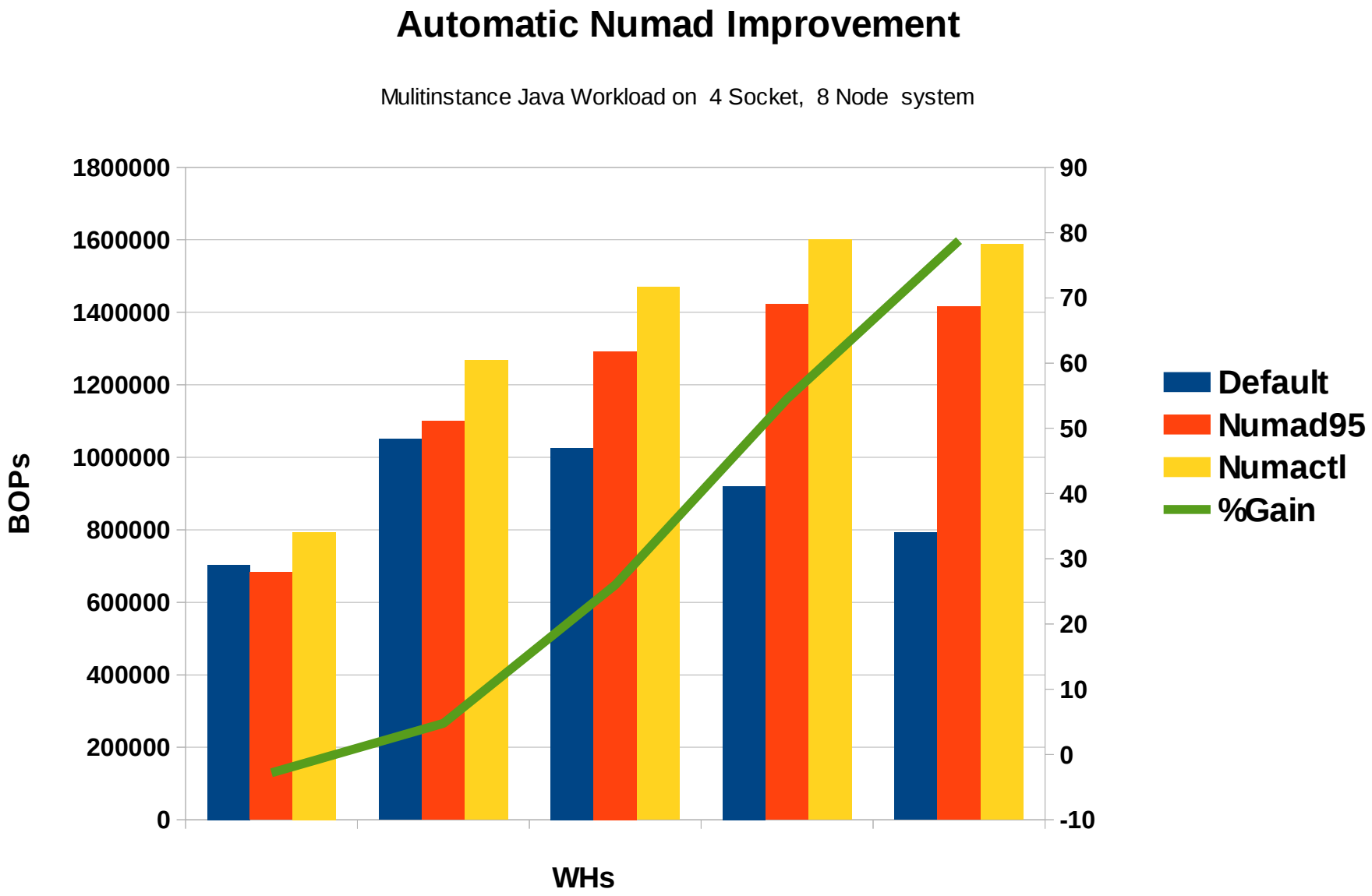
numastat shows aligned guests

```
# numastat -c qemu

Per-node process memory usage (in Mbs)

PID                Node 0 Node 1 Node 2 Node 3 Total
---------------  ------ ------ ------ ------ -----
10587 (qemu-kvm)      0  10723      5      0 10728
10629 (qemu-kvm)      0      0      5  10717 10722
10671 (qemu-kvm)      0      0  10726      0 10726
10713 (qemu-kvm)  10733      0      5      0 10738
---------------  ------ ------ ------ ------ -----
Total             10733  10723  10740  10717 42913
```

Bare Metal - Java Workload



CPU Tuning

CPU Tuning – Power Savings / cpuspeed

- Power savings mode
  - cpuspeed or cpupower
    - 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 – 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

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

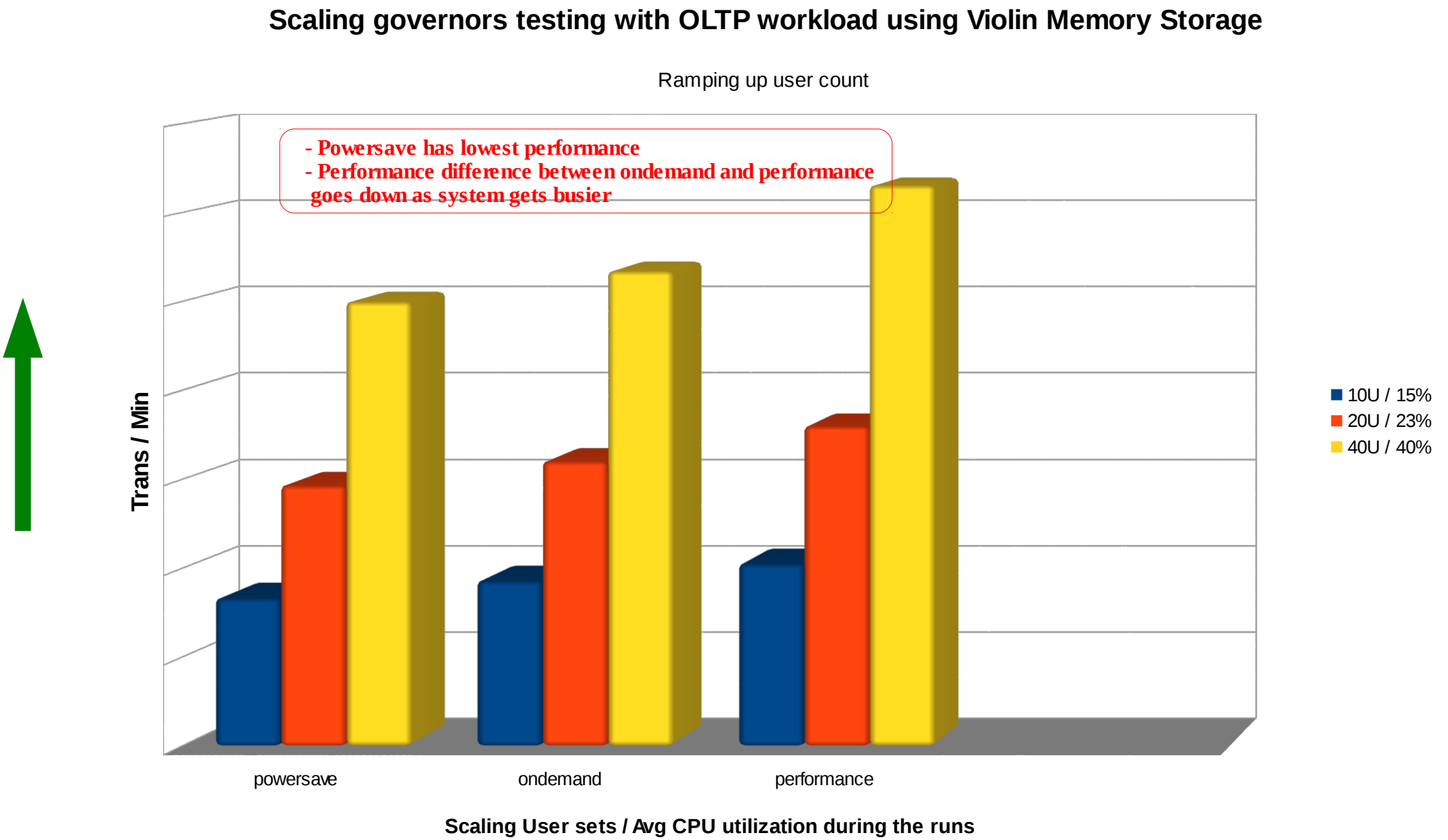
- *turbostat -i <interval>*

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
```

CPU Tuning – Effect of Power Savings - OLTP



Gotchas

- Test your changes
- If an application is not compatible with transparent huge pages, you will find out quickly!

Recommendations

- Use Tuned!
  - %10-%30

Thank you!