Tuning Red Hat Platform for Databases

Sanjay Rao
Principal Software Engineer
May 3 2017

RHEL Performance Evolution

RHEL5	RHEL6	RHEL7	RH Cloud
Static Hugepages	Transparent Hugepages	Tuned - throughput-performance (default)	RHEV tuned profile
CPU Sets	Tuned - Choose Profile	Automatic NUMA-balancing kernel scheduler	RHEL OSP7 Tuned, NUMA, SR-IOV
Ktune on/off	NUMAD - userspace	Containers/Docker	RHEL Atomic Host, Atomic Ent
CPU Affinity (taskset)	cgroups	Irqbalance - NUMA enhanced	OpenShift v3
NUMA Pinning (numactl)	irqbalance - NUMA enhanced		CloudForms
irqbalance			

#1 performance and price/performance on non-clustered TPC-H@1000GB

HPE, **Microsoft**, and **Red Hat** deliver first-ever result with **SQL Server 2017 Enterprise Edition**

Winning partnerships!

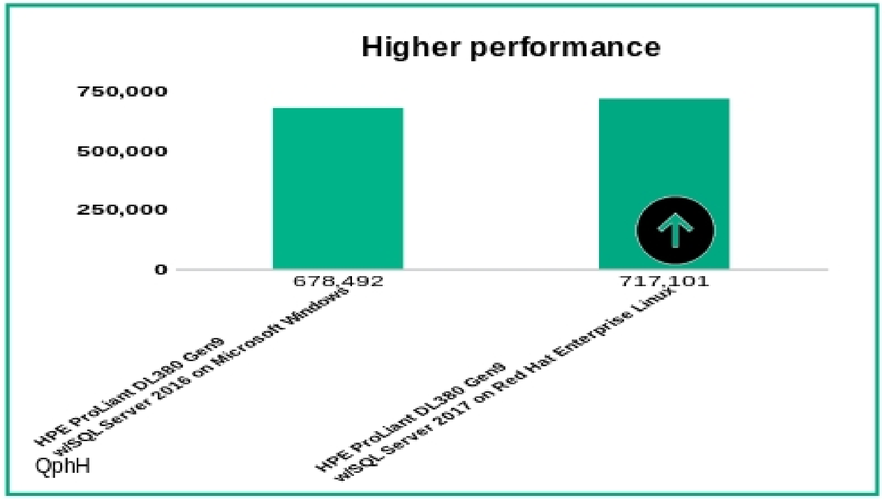
Microsoft

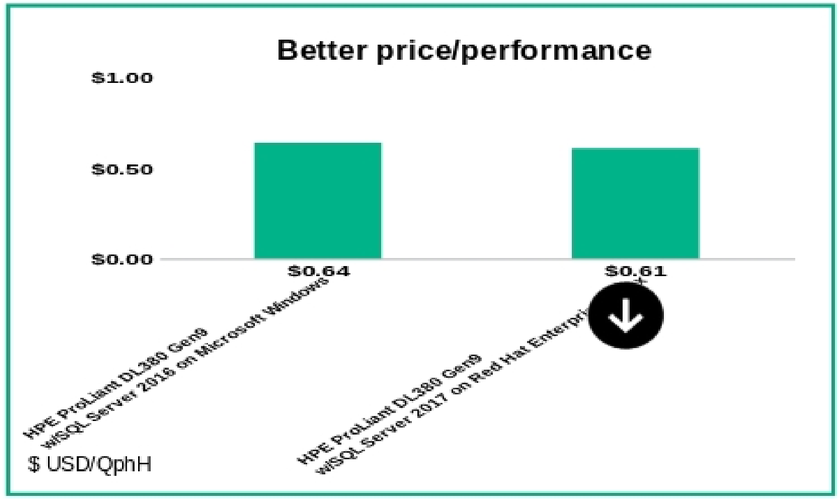
redhat.

HPE ProLiant DL380 Gen9 | **SQL Server 2017 Enterprise Edition** | **Red Hat Enterprise Linux 7**

Key performance takeaways

- SQL Server 2017 on Red Hat Enterprise Linux surpasses the previous #1 TPC-H@1000GB result achieved with SQL Server 2016
  - 6% higher performance
  - 5% better price/performance
- The first and only result with Microsoft SQL Server 2017 Enterprise Edition
- Results achieved on similarly configured servers with two Intel® Xeon® E5-2699 v4 processors





Read the performance brief at hpe.com/servers/benchmarks.

Hewlett Packard Enterprise

Microsoft

redhat.

What is “tuned” ?

- Tuning framework that dynamically modifies system parameters that affect performance
- Pre-existing list of profiles for different sub-systems / application classes
- Existing profiles can be modified (not recommended)
- Custom profiles can be created
- Installed by default in RHEL 7
  - Desktop/Workstation: balanced
  - Server/HPC: throughput-performance
- Can be rolled back

tuned profile list

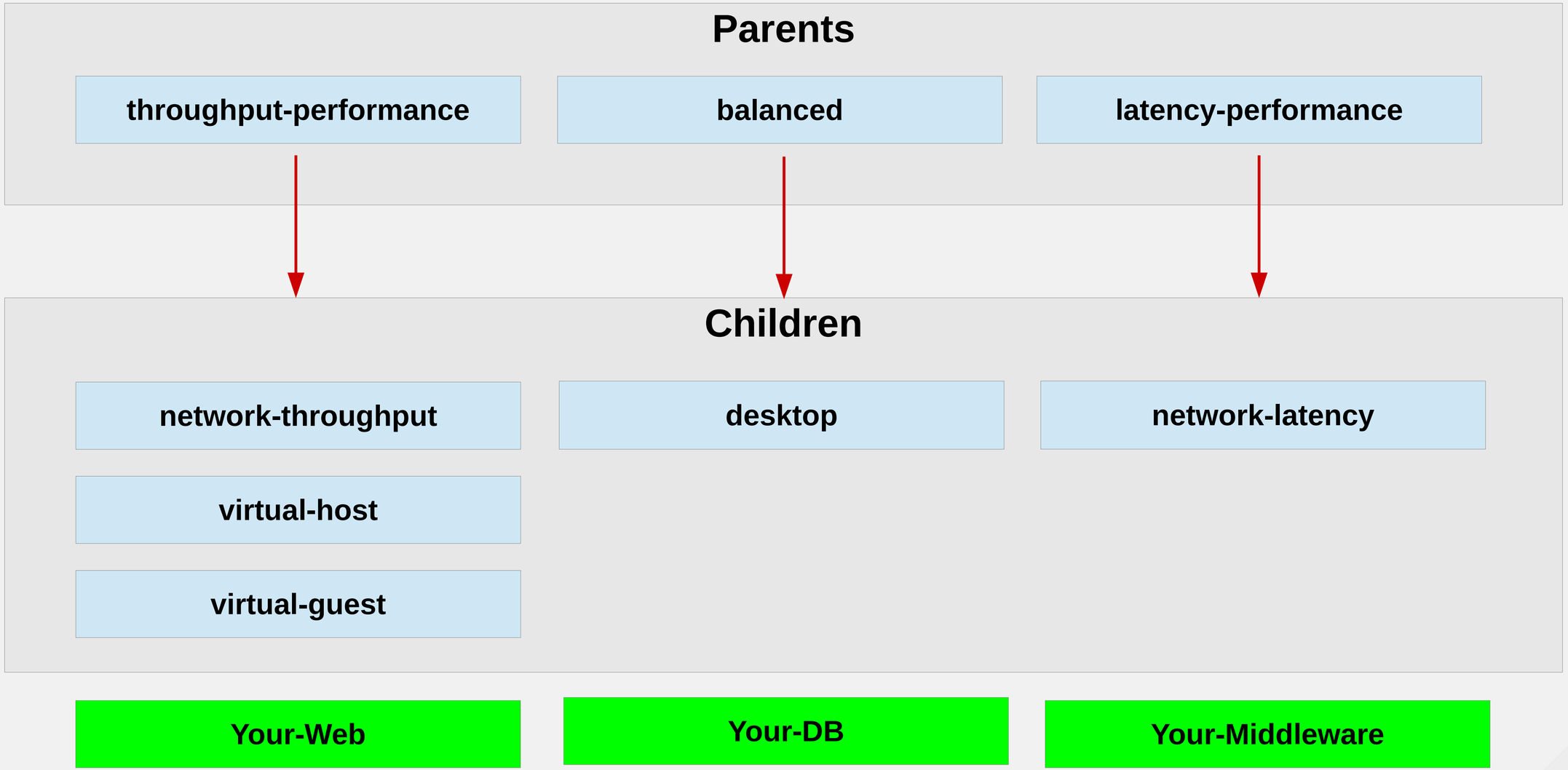
```
# tuned-adm list
    Available profiles:
    - balanced
    - desktop
    - latency-performance
    - network-latency
    - network-throughput
    - powersave
    - sap
    - throughput-performance
    - virtual-guest
    - virtual-host
    Current active profile: throughput-performance
```

Tuned: Updates for RHEL7

- Re-written for maintainability and extensibility.
  - Configuration is now consolidated a single tuned.conf file
    - **/usr/lib/tuned/<profile-name>/tuned.conf**
  - Detail config
  - Optional hook/callout capability
  - Adds concept of Inheritance (just like httpd.conf)
  - Profiles updated for RHEL7 features and characteristics

- How do I create my own “tuned” profile on RHEL?
  - https://access.redhat.com/solutions/731473

Tuned: Profile Inheritance



Tuned: Throughput Profiles - RHEL7

Tunable	Units	Balanced	throughput-performance	network-throughput
Inherits From/Notes				throughput-performance
sched_min_ granularity_ns	nanoseconds	auto-scaling	10000000	
sched_wakeup_granularity_ns	nanoseconds	3000000	15000000	
dirty_ratio	Percent	20	40	
dirty_background_ratio	Percent	10	10	
swappiness	Weight 1-100	60	10	
I/O Scheduler (Elevator)		deadline		
Filesystem Barriers	Boolean	Enabled		
CPU Governor		ondemand	performance	
Disk Read-ahead	KB	128	4096	
Disable THP	Boolean	Enabled		
Energy Perf Bias		normal	performance	
kernel.sched_migration_cost_ns	nanoseconds	500000		
min_perf_pct (intel_pstate only)	Percent	auto-scaling	100	
tcp_rmem	Bytes	auto-scaling		Max=16777216
tcp_wmem	Bytes	auto-scaling		Max=16777216
udp_mem	Pages	auto-scaling		Max=16777216

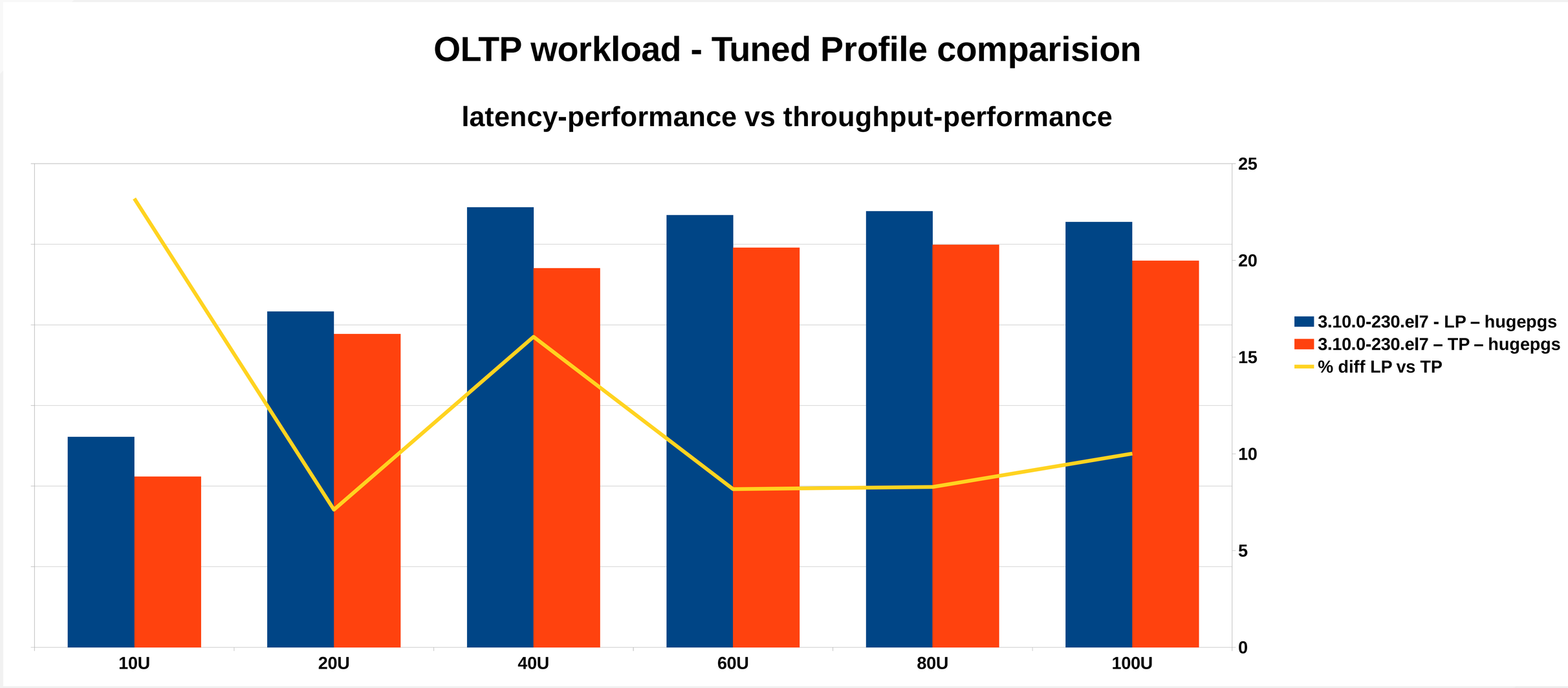
Tuned: Latency Profiles - RHEL7

Tunable	Units	Balanced	latency-performance	network-latency
Inherits From/Notes				**latency-performance**
sched_min_ granularity_ns	nanoseconds	auto-scaling	10000000	
sched_wakeup_granularity_ns	nanoseconds	3000000		10000000
dirty_ratio	percent	20	10	
dirty_background_ratio	percent	10	3	
swappiness	Weight 1-100	60	10	
I/O Scheduler (Elevator)		deadline		
Filesystem Barriers	Boolean	Enabled		
CPU Governor		ondemand	performance	
Disable THP	Boolean	N/A	**No**	Yes
CPU C-States		N/A	Locked @ 1	
Energy Perf Bias		normal	performance	
kernel.sched_migration_cost_ns	nanoseconds	N/A	5000000	
min_perf_pct (intel_pstate only)	percent		100	
net.core.busy_read	microseconds			50
net.core.busy_poll	microseconds			50
net.ipv4.tcp_fastopen	Boolean			Enabled
kernel.numa_balancing	Boolean			Disabled

Tuned: Virtualization Profiles - RHEL7

Tunable	Units	throughput-performance	virtual-host	virtual-guest
Inherits From/Notes			throughput-performance	throughput-performance
sched_min_ granularity_ns	nanoseconds	10000000		
sched_wakeup_granularity_ns	nanoseconds	15000000		
dirty_ratio	percent	40		30
dirty_background_ratio	percent	10	5	10
swappiness	Weight 1-100	10		
I/O Scheduler (Elevator)				
Filesystem Barriers	Boolean			
CPU Governor		performance		
Disk Read-ahead	Bytes	4096		
Energy Perf Bias		performance		
kernel.sched_migration_cost_ns	nanoseconds		5000000	
min_perf_pct (intel_pstate only)	percent	100		

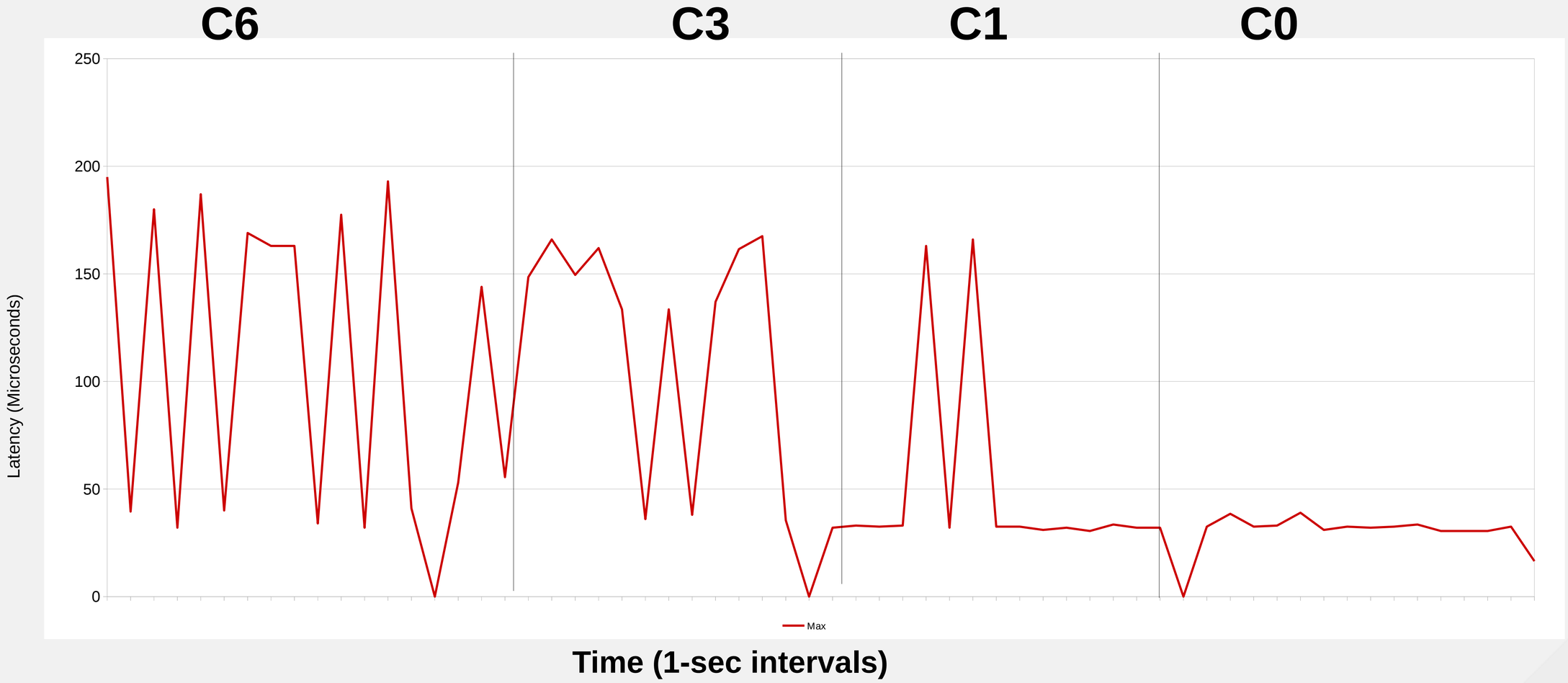
Tuned – Database workload



Please note : The latency profiles tweak the CPU power features and they will not work if the BIOS is not configured properly

Tuned: Network Latency Performance Boost

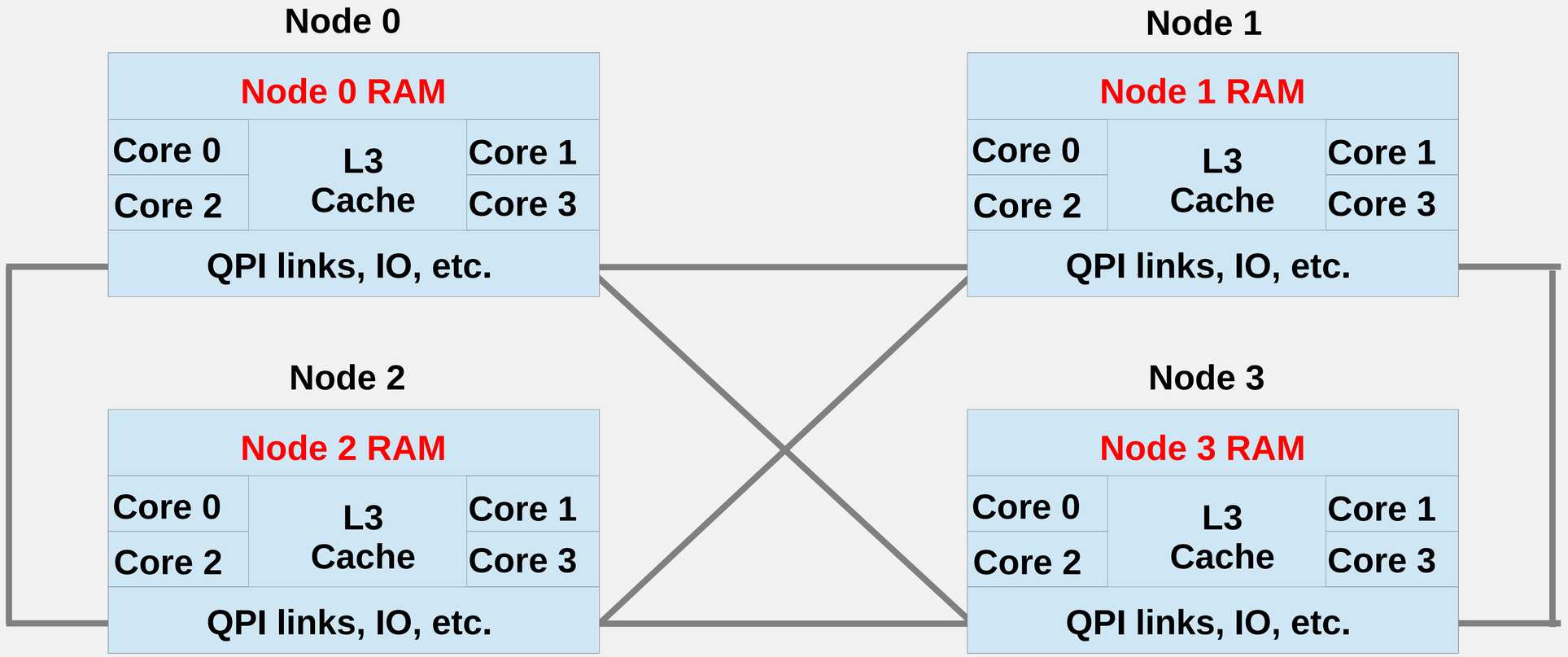
C-state lock improves determinism, reduces jitter



Memory Tuning

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

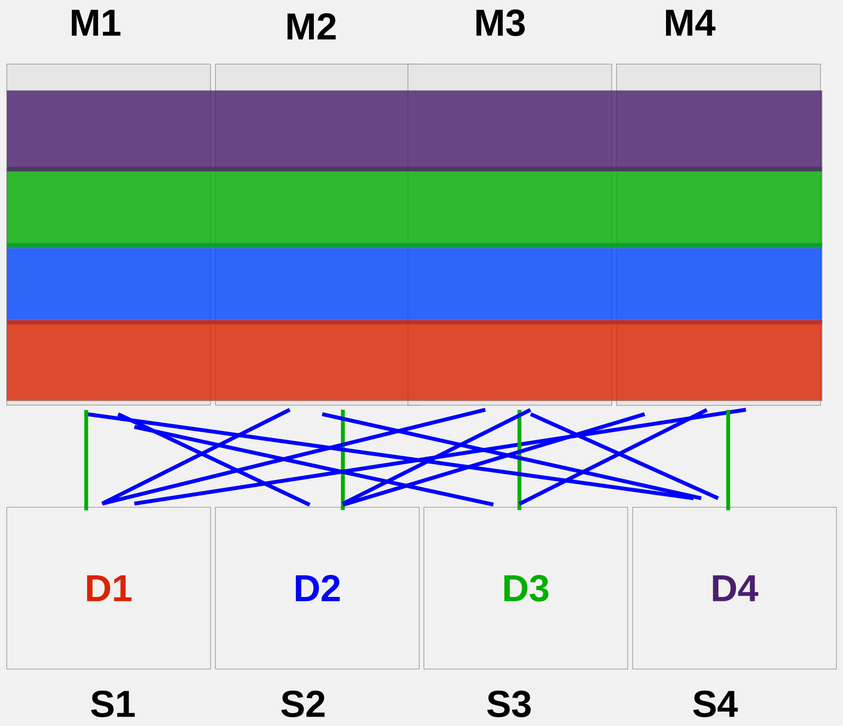
Typical NUMA System



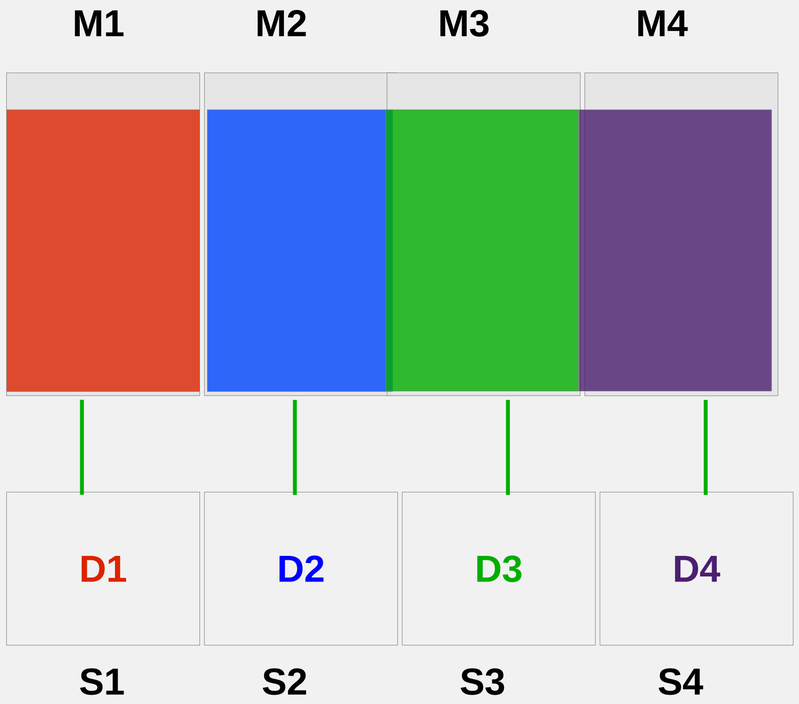
Multi Socket – Multi core architecture

- NUMA required for scaling
- Autonuma code provides performance improvement
- Additional performance gains by enforcing NUMA placement

Understanding NUMA (Non Uniform Memory Access)



No NUMA optimization



NUMA optimization

S – Socket
M – Memory Bank – Attached to each Socket
D – Database

Access path between sockets
Access path between sockets and memory

Numa Due Diligence

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

Numa Due Diligence (cont)

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

Visualize CPUs via lstopo

```
# lstopo
```

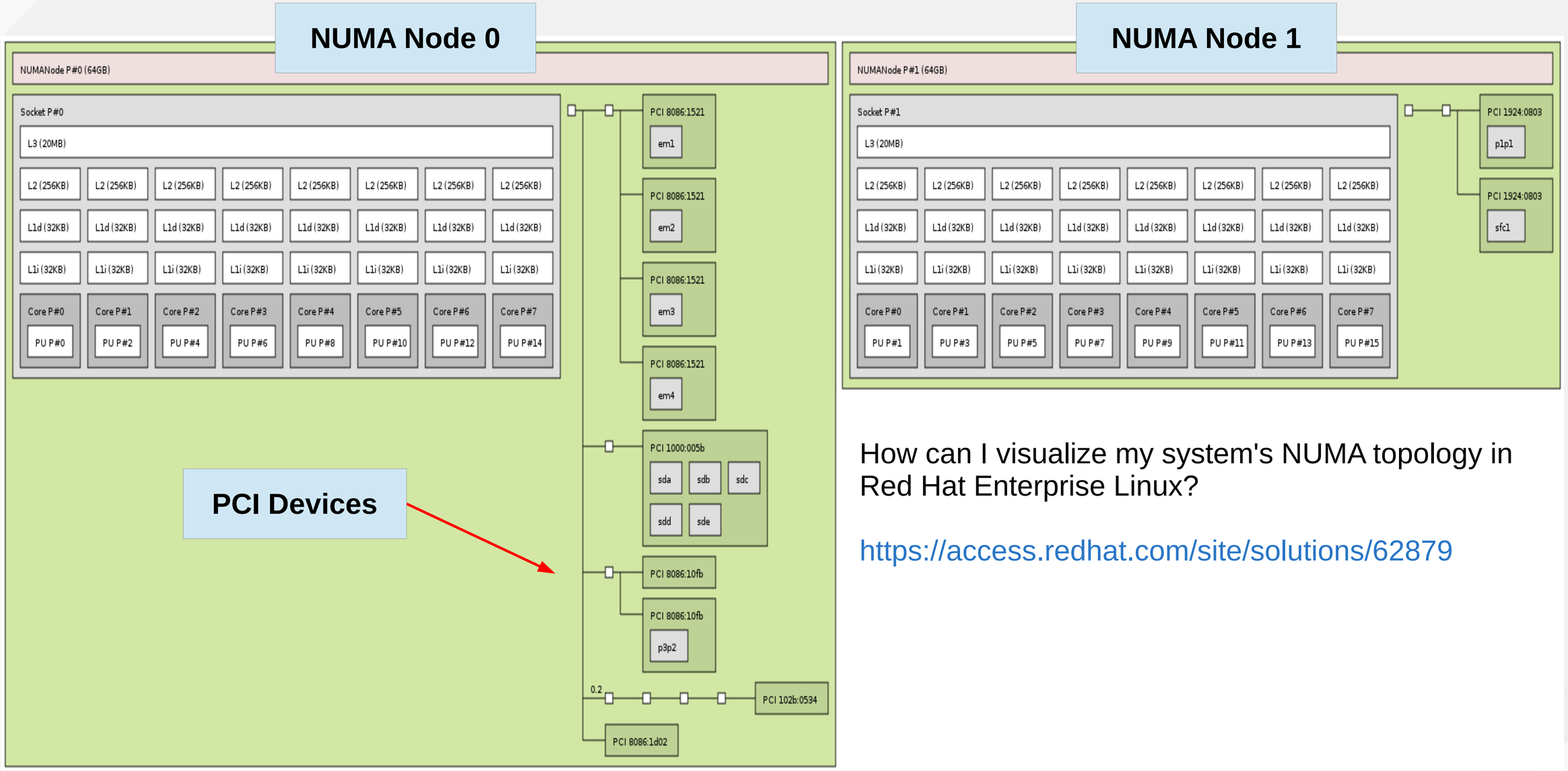
PCIe

NUMA

CACHE

HT

Visualize NUMA Topology: lstopo



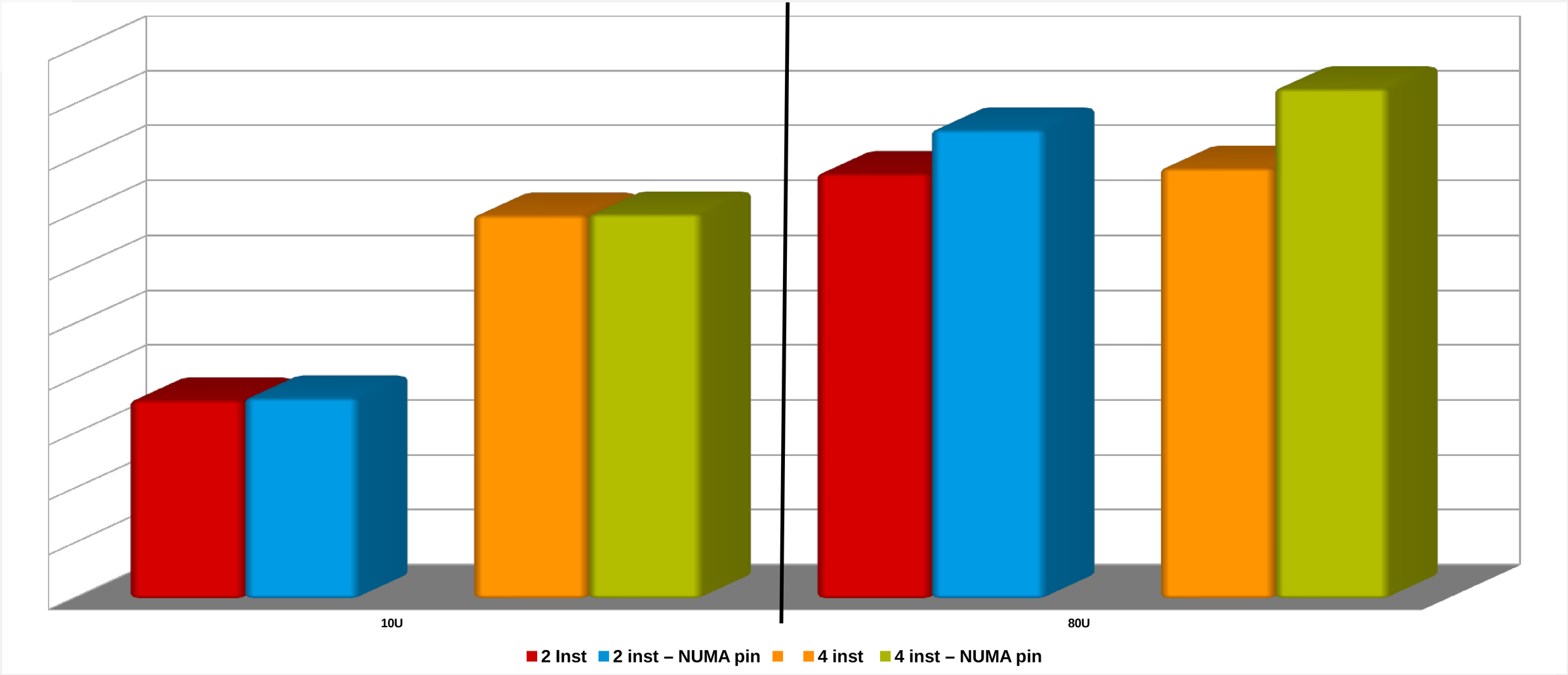
How can I visualize my system's NUMA topology in Red Hat Enterprise Linux?

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

Memory Tuning – NUMA

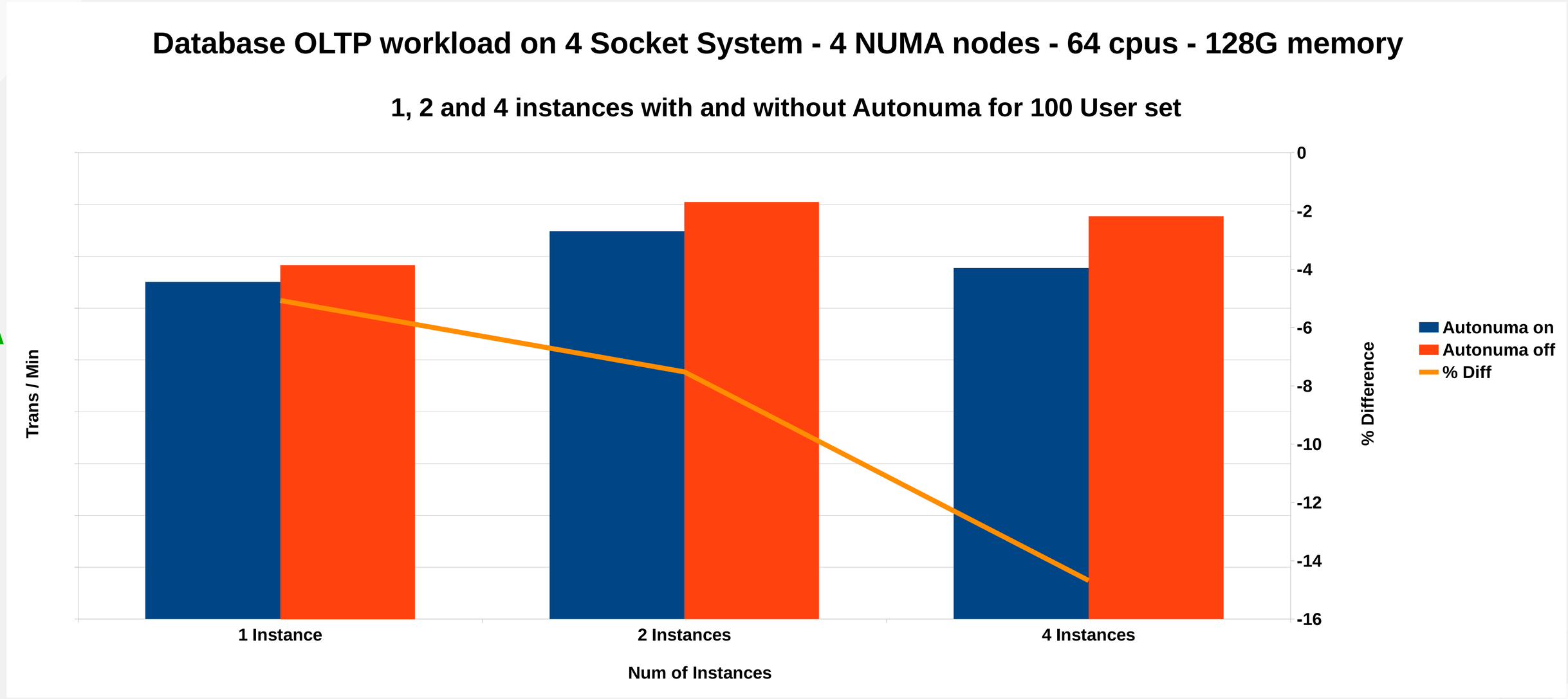
- Enforce NUMA placement
  - Numactl
    - CPU and memory pinning
  - Taskset
    - CPU pinning

Memory Tuning – Effect of NUMA Tuning



The benefits of NUMA pinning is evident when there is a lot of cross NUMA node traffic
In the figure above, we don't see any gain with the 10U runs
We get 10% performance improvement with 2 instaces and 20% imptovements with 4 instances with 80U runs

Memory Tuning – Autonuma



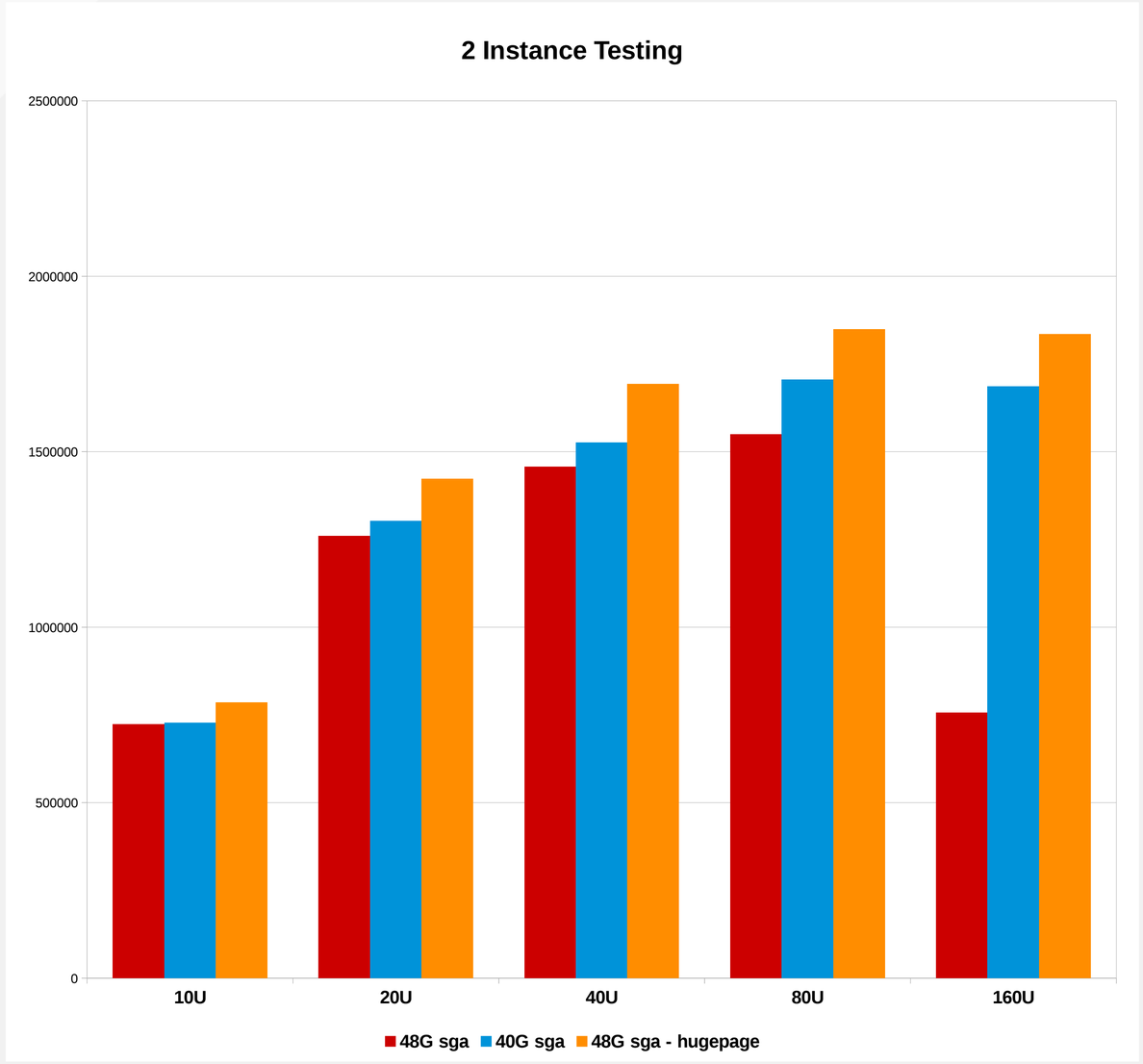
- With hugepages, there is no performance difference with Autonuma on or off because the memory is wired down
- # echo 0 > /proc/sys/kernel/numa_balancing

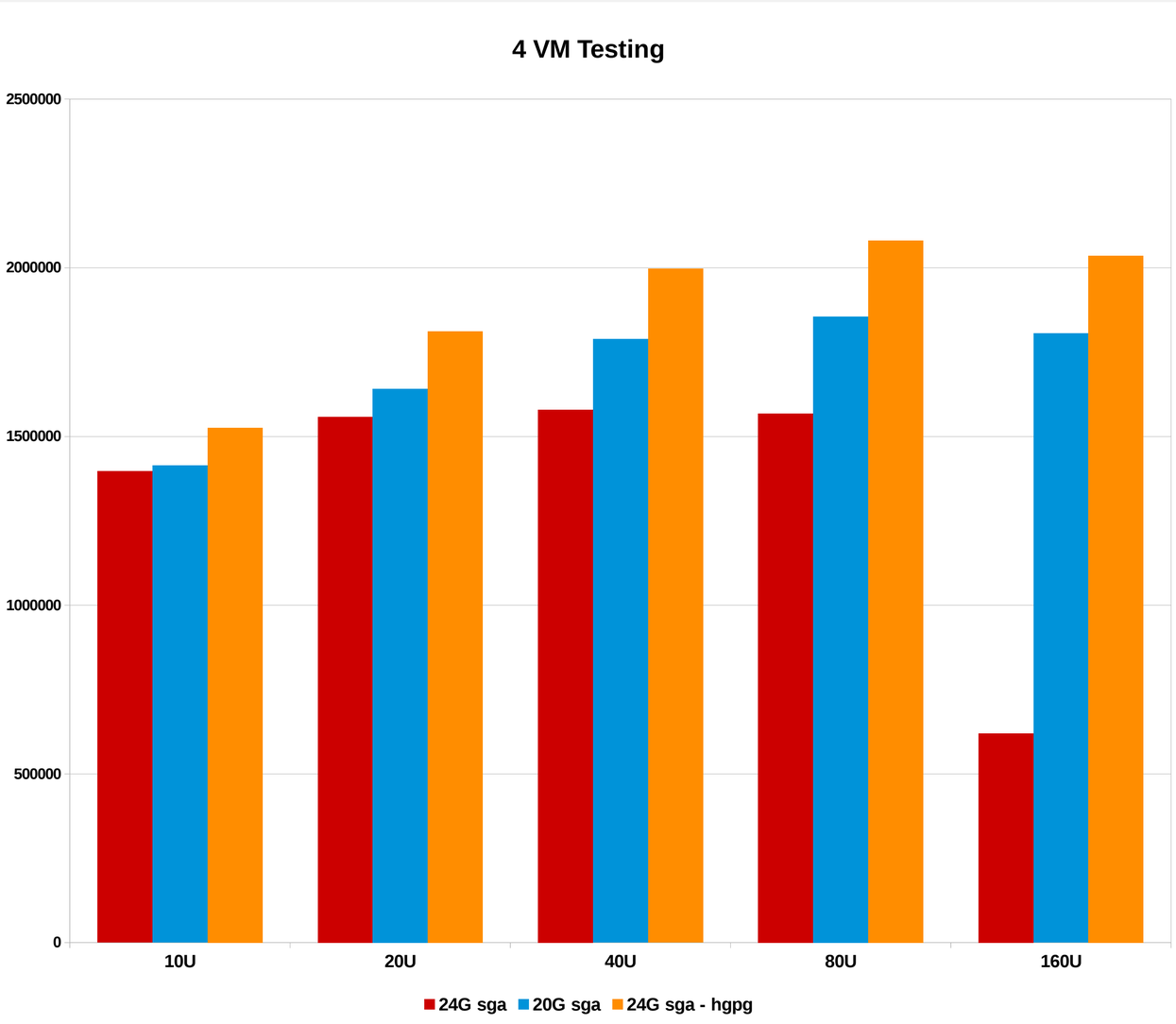
Memory Tuning – Huge Pages

- 2M pages vs 4K standard linux page
- **Virtual to physical page map is 512 times smaller**
- TLB can map more physical pages, resulting in fewer misses
- **Traditional Huge Pages always pinned – Prevents swapping of database shared memory**
- 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 (96G)**
  - echo 49600 > /proc/sys/vm/nr_hugepages
  - vi /etc/sysctl.conf (vm.nr_hugepages=49600)

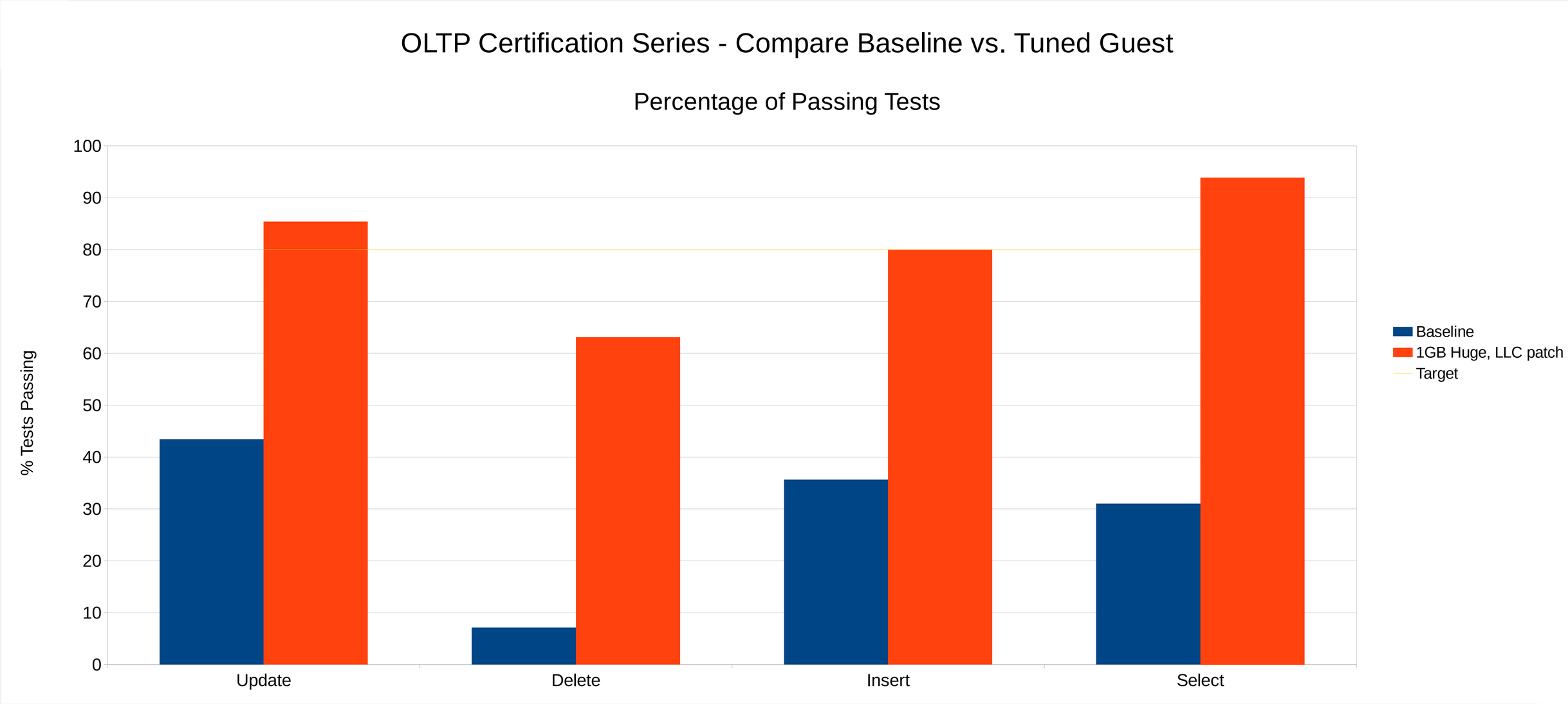
Memory Tuning – huge pages on Bare Metal





The effect of hugepages are more pronounced with multiple databases

Memory Tuning – huge pages for RHV



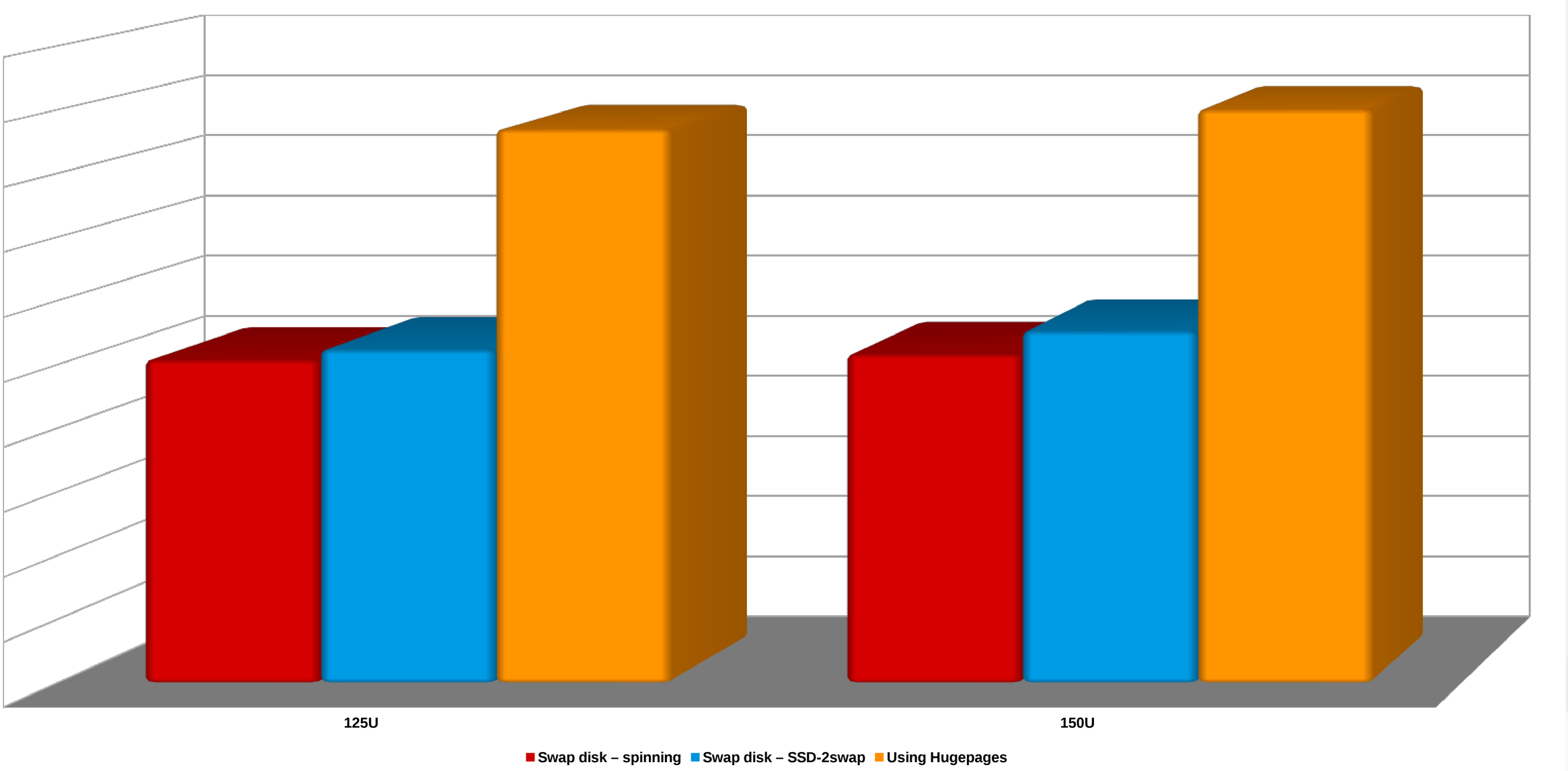
In Memory databases will benefit from hugepages – 1G Pages / SAP HANA –
All the performance was not just from hugepages. Other patches contributed to the gains

Tuning Memory – swappiness

- Not needed as much in RHEL7
- Controls how aggressively the system reclaims “mapped” memory
- Default - 60%
- Decreasing: more aggressive reclaiming of unmapped pagecache memory, thereby delaying swapping
- Increasing: more aggressive swapping of mapped memory

- **Avoid swapping of database shared memory at all costs**

Memory Tuning – huge pages and swapping

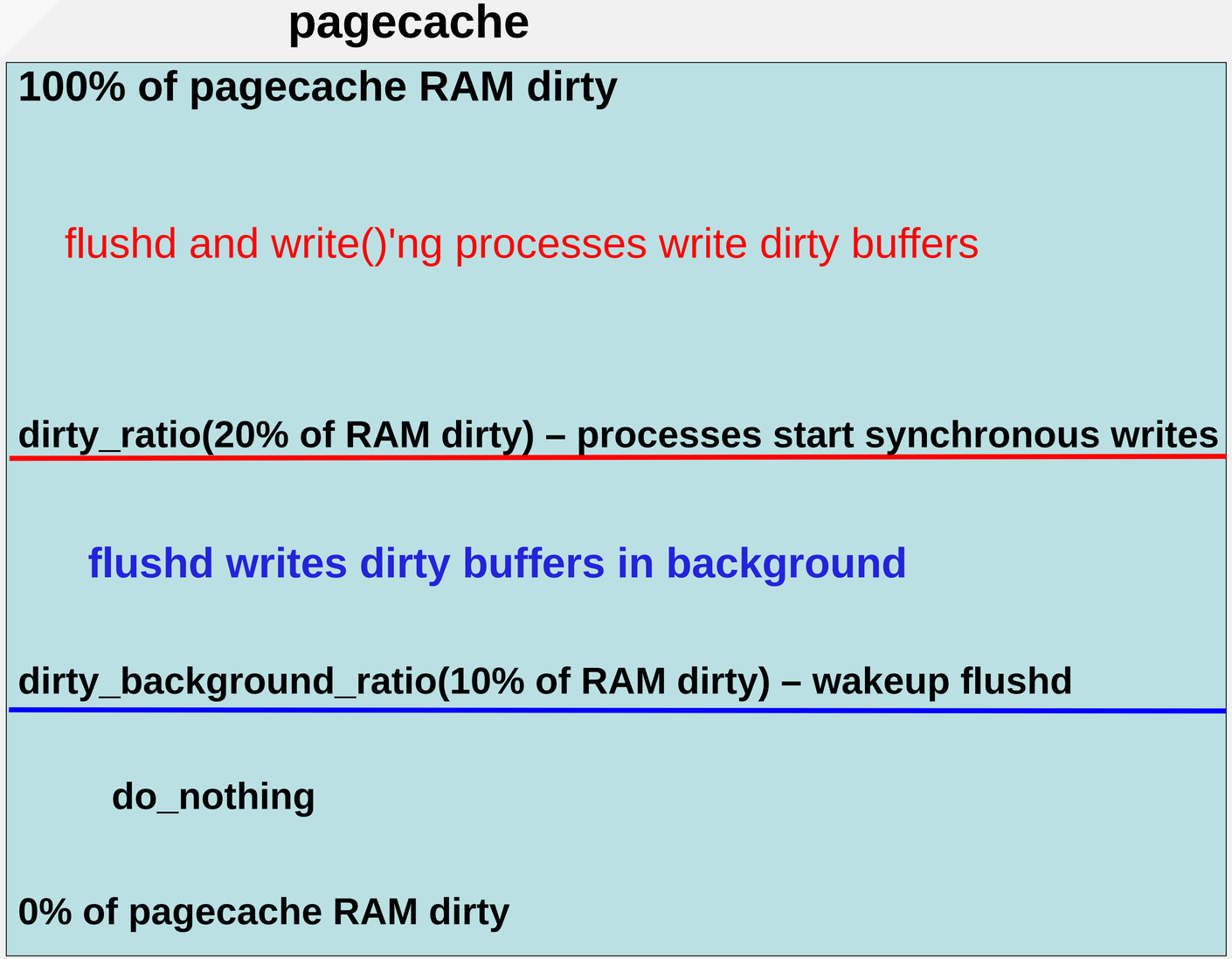


Tuning Memory – Flushing Caches

- 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

dirty_ratio and dirty_background_ratio



If there is a lot of pagecache pressure one would want to start background flushing sooner and delay the synchronous writes. This can be done by

- **Lowering the dirty_background_ratio**
- **Increasing the dirty_ratio**

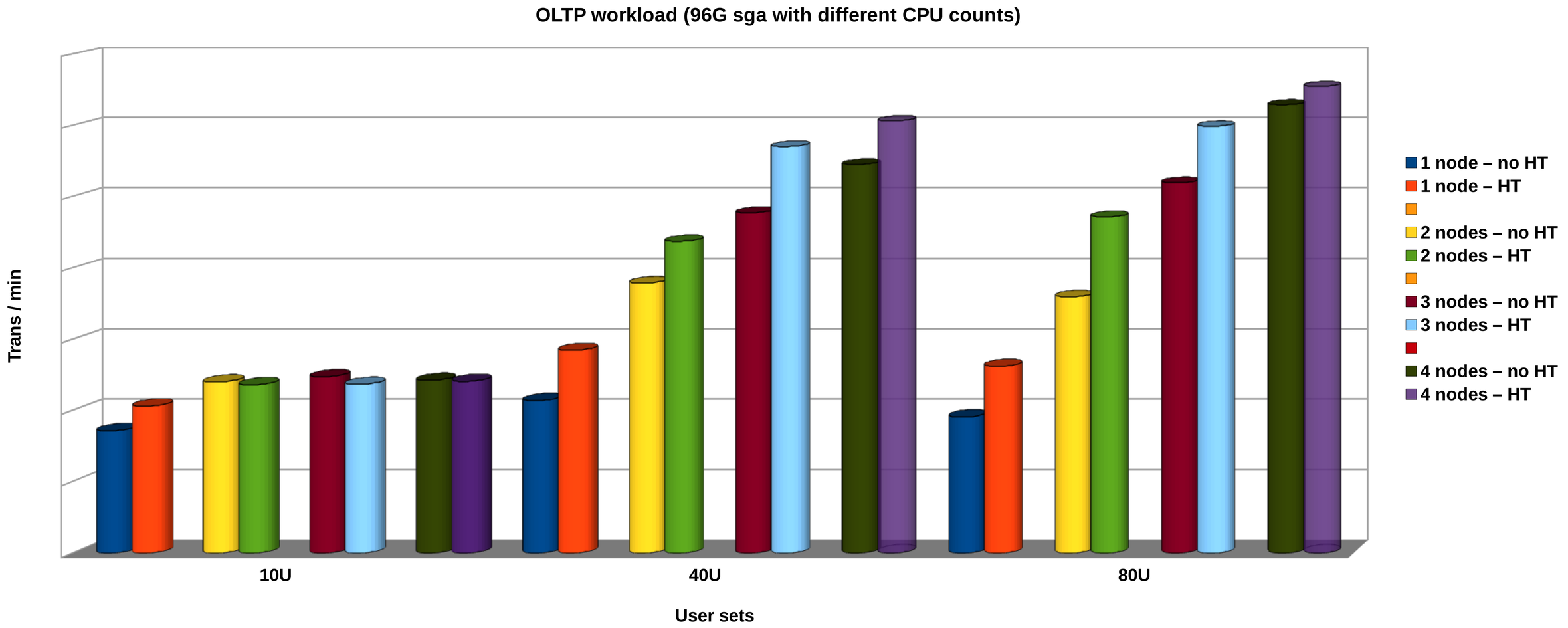
Anything “Hyper” has to be good for perfomance... right?

- Using Hyperthreads improves performance with database workload but the mileage will vary depending on how the database workload scales.

- Having more CPUs sharing the same physical cache can help performance

- Some workloads lend themselves to scaling efficiently and they will do very well with hyperthreads but if the scaling factor for workloads are not linear with physical cpus, it probably won’t be a good candidate for scaling with hyperthreads.

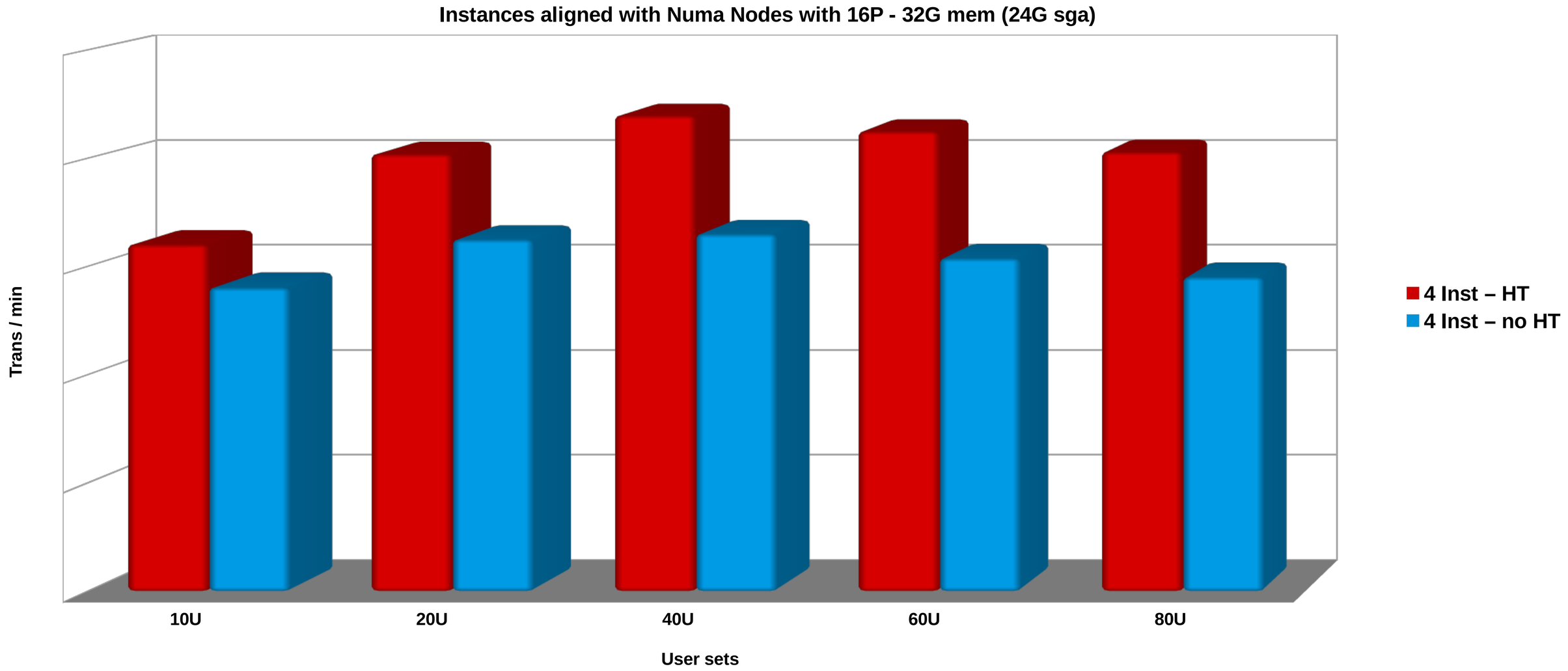
Scaling test - Hyperthread vs no Hyperthead



Single Instance scaled across NUMA nodes, one node at a time. The 1 node test shows the best gain in performance. As more NUMA nodes come into play, the performance difference is hard to predict because of the memory placement and the CPU cache sharing among physical threads and hyperthreads of the CPUs

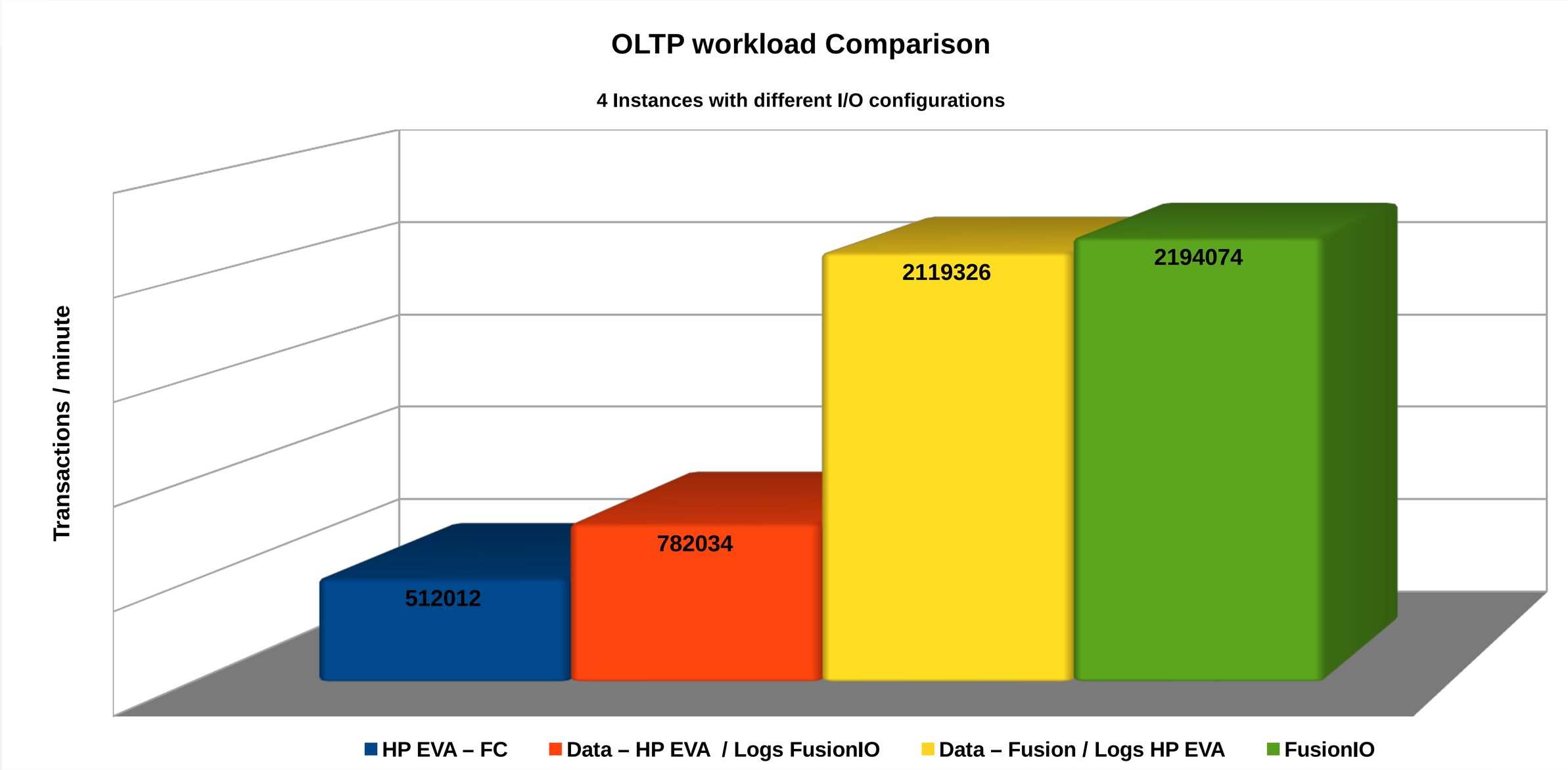
Avg % gain – 1 Node – 30% / 2 Nodes – 20 % / 3 Nodes – 14 % / 4 Nodes – 5.5 %

Multi Instances of database with and without Hyperthreads



Each of the 4 instances were aligned to an individual NUMA node. This test shows the best gain in performances as other factors influencing performance like NUMA, I/O are not a factor

I/O Tuning – Database Layout



I/O Tuning – Database layout – vmstat

OLTP Workload
Fibre Channel Storage

r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id	wa	st
62	19	5092	44894312	130704	76267048	0	0	8530	144255	35350	113257	43	4	45	9	0
3	20	5092	43670800	131216	77248544	0	0	6146	152650	29368	93373	33	3	53	11	0
21	27	5092	42975532	131620	77808736	0	0	2973	147526	20886	66140	20	2	65	13	0
7	20	5092	42555764	132012	78158840	0	0	2206	136012	19526	61452	17	2	69	12	0
25	18	5092	42002368	132536	78647472	0	0	2466	144191	20255	63366	19	2	67	11	0
4	21	5092	41469552	132944	79111672	0	0	2581	144470	21125	66029	22	2	65	11	0
1	32	5092	40814696	133368	79699200	0	0	2608	151518	21967	69841	23	2	64	11	0
4	17	5092	40046620	133804	80385232	0	0	2638	151933	23044	70294	24	2	64	10	0
17	14	5092	39499580	134204	80894864	0	0	2377	152805	23663	72655	25	2	62	10	0
35	14	5092	38910024	134596	81436952	0	0	2278	152864	24944	74231	27	2	61	9	0
20	13	5092	38313900	135032	81978544	0	0	2091	156207	24257	72968	26	2	62	10	0
1	14	5092	37831076	135528	82389120	0	0	1332	155549	19798	58195	20	2	67	11	0
23	24	5092	37430772	135936	82749040	0	0	1955	145791	19557	56133	18	2	66	14	0
34	12	5092	36864500	136396	83297184	0	0	1546	141385	19957	56894	19	2	67	13	0

OLTP Workload
PCI SSD Storage

r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id	wa	st
77	0	6604	55179876	358888	66226960	0	0	7325	266895	70185	149686	90	7	4	0	0
77	1	6604	50630092	359288	70476248	0	0	6873	306900	70166	149804	88	7	5	0	0
76	3	6604	46031168	360132	74444776	0	0	5818	574286	77388	177454	88	8	4	0	0
81	1	6604	41510608	360512	78641480	0	0	4970	452939	75322	168464	89	7	3	0	0
82	3	6604	35358836	361012	84466256	0	0	4011	441042	74022	162443	88	7	4	0	0
81	3	6604	34991452	361892	84740008	0	0	2126	440876	73702	161618	88	7	5	0	0
79	1	6604	34939792	362296	84747016	0	0	2323	400324	73091	161592	90	6	3	0	0
79	0	6604	34879644	362992	84754016	0	0	2275	412631	73271	160766	89	6	4	0	0
76	2	6604	34844616	363396	84760976	0	0	2275	415777	73019	158614	89	6	4	0	0
61	3	6604	34808680	363828	84768016	0	0	2209	401522	72367	159100	89	6	4	0	0
77	1	6604	34781944	364180	84774992	0	0	2172	401966	73253	159064	90	6	4	0	0
54	4	6604	34724948	364772	84803456	0	0	3031	421299	72990	156224	89	6	4	0	0
80	2	6604	34701500	365500	84809072	0	0	2216	573246	76404	175922	88	7	5	1	0

Memory Stats

I/O Stats

CPU stats

Swap stats

I/O Tuning
Database Layout

monitoring I/O
iostat -dmxz 3

Fibre Channel

Device:	rrqm/s	wrqm/s	r/s	w/s	rMB/s	wMB/s	avgrq-sz	avgqu-sz	await	r_await	w_await	svctm	%util
dm-2	0.00	0.00	0.00	10.20	0.00	0.64	127.64	0.25	24.98	0.00	24.98	1.27	1.30
dm-5	0.00	8.90	0.00	289.00	0.00	2.13	15.11	0.13	0.45	0.00	0.45	0.38	10.94
dm-6	0.00	213.20	0.00	1224.50	0.00	9.31	15.57	0.42	0.34	0.00	0.34	0.32	39.50
dm-7	0.00	72.90	0.00	903.60	0.00	6.06	13.73	0.27	0.29	0.00	0.29	0.27	24.01
dm-8	0.00	1.70	0.00	131.70	0.00	0.91	14.18	0.06	0.44	0.00	0.44	0.43	5.61
dm-18	0.10	0.20	1784.10	1.10	68.06	0.01	78.10	92.18	51.61	51.64	5.27	0.56	100.01
dm-19	0.00	0.10	1684.80	0.90	70.32	0.01	85.45	78.34	46.42	46.45	0.22	0.59	100.01
dm-20	0.00	0.20	1952.20	1.10	74.02	0.01	77.62	84.36	43.25	43.27	1.18	0.51	100.01
dm-21	0.00	0.10	1226.30	0.90	49.89	0.01	83.28	96.24	78.52	78.58	0.78	0.81	100.01
Log on PCI – SSD													
fiod	0.00	1340.20	0.00	1524.70	0.00	47.86	64.29	0.19	0.13	0.00	0.13	0.13	19.31
fioa	0.00	911.10	0.00	1499.10	0.00	31.11	42.49	0.16	0.11	0.00	0.11	0.10	15.74
fioc	0.00	1059.60	0.00	1379.10	0.00	38.69	57.45	0.16	0.11	0.00	0.11	0.11	15.64
fiob	0.00	991.60	0.00	1534.80	0.00	40.31	53.79	0.18	0.12	0.00	0.12	0.11	17.51
dm-18	0.00	0.40	293.70	555.60	9.82	24.92	83.76	19.19	22.96	61.97	2.34	1.15	97.53
dm-19	0.00	0.40	197.70	587.10	6.34	25.86	84.03	18.57	24.42	87.00	3.34	1.23	96.91
dm-20	0.00	0.40	221.40	431.40	7.18	19.94	85.08	19.74	30.86	84.25	3.46	1.50	98.17
dm-21	0.00	0.40	203.10	295.40	6.88	14.16	86.43	22.20	45.48	104.92	4.61	1.98	98.70
Data on PCI – SSD													
fiod	0.00	0.40	800.80	6101.60	6.28	167.21	51.48	10.37	1.50	2.93	1.32	0.14	95.10
fioa	0.00	0.40	671.60	6309.40	5.27	166.26	50.32	6.42	0.92	1.86	0.82	0.12	86.88
fioc	0.00	0.40	779.20	6195.60	6.10	173.99	52.88	10.15	1.46	3.28	1.23	0.14	96.19
fiob	0.00	9.40	813.80	6176.90	6.46	197.63	59.79	9.98	1.43	2.87	1.24	0.14	94.51
dm-18	0.00	1428.90	0.00	1029.70	0.00	46.56	92.60	0.54	0.52	0.00	0.52	0.50	51.89
dm-19	0.00	1475.00	0.00	1100.70	0.00	47.81	88.95	0.57	0.52	0.00	0.52	0.51	55.75
dm-20	0.00	1093.50	0.00	693.20	0.00	45.64	134.83	0.53	0.77	0.00	0.77	0.67	46.76
dm-21	0.00	1312.20	0.00	849.50	0.00	46.20	111.38	0.59	0.69	0.00	0.69	0.61	51.69
dm-1	0.00	0.00	0.00	0.80	0.00	0.02	49.12	0.01	6.25	0.00	6.25	4.25	0.34
SanDisk PCI - SSD													
fioa	0.00	1279.30	522.00	1049.70	13.99	54.21	88.87	0.27	0.17	0.22	0.15	0.13	20.50
fioc	0.00	1359.80	651.70	1247.70	5.10	57.99	68.03	0.34	0.18	0.18	0.18	0.13	23.84
fiob	0.00	1284.30	693.10	1221.10	5.47	85.05	96.84	0.72	0.38	0.26	0.45	0.15	27.88
fiod	0.00	1434.30	740.80	1512.20	5.81	66.73	65.94	0.80	0.36	0.20	0.44	0.11	25.32
fioa	0.00	1292.20	514.60	1043.70	5.24	55.90	80.36	0.30	0.20	0.19	0.20	0.13	19.86
fioc	0.00	909.10	674.20	826.00	5.29	59.69	88.71	0.41	0.28	0.21	0.33	0.15	22.10
fiob	0.00	1380.40	728.20	1458.80	5.71	66.02	67.17	0.99	0.45	0.20	0.58	0.13	27.53
fiod	0.00	1297.10	815.40	1494.90	6.43	65.28	63.57	0.90	0.39	0.21	0.49	0.12	26.78

I/O Tuning
Database Layout

monitoring I/O
iostat -dmxz 3

LVM – Fibre Channel

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

LVM – Mix (Fibre Channel - SSD)

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

LVM - SSDs

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

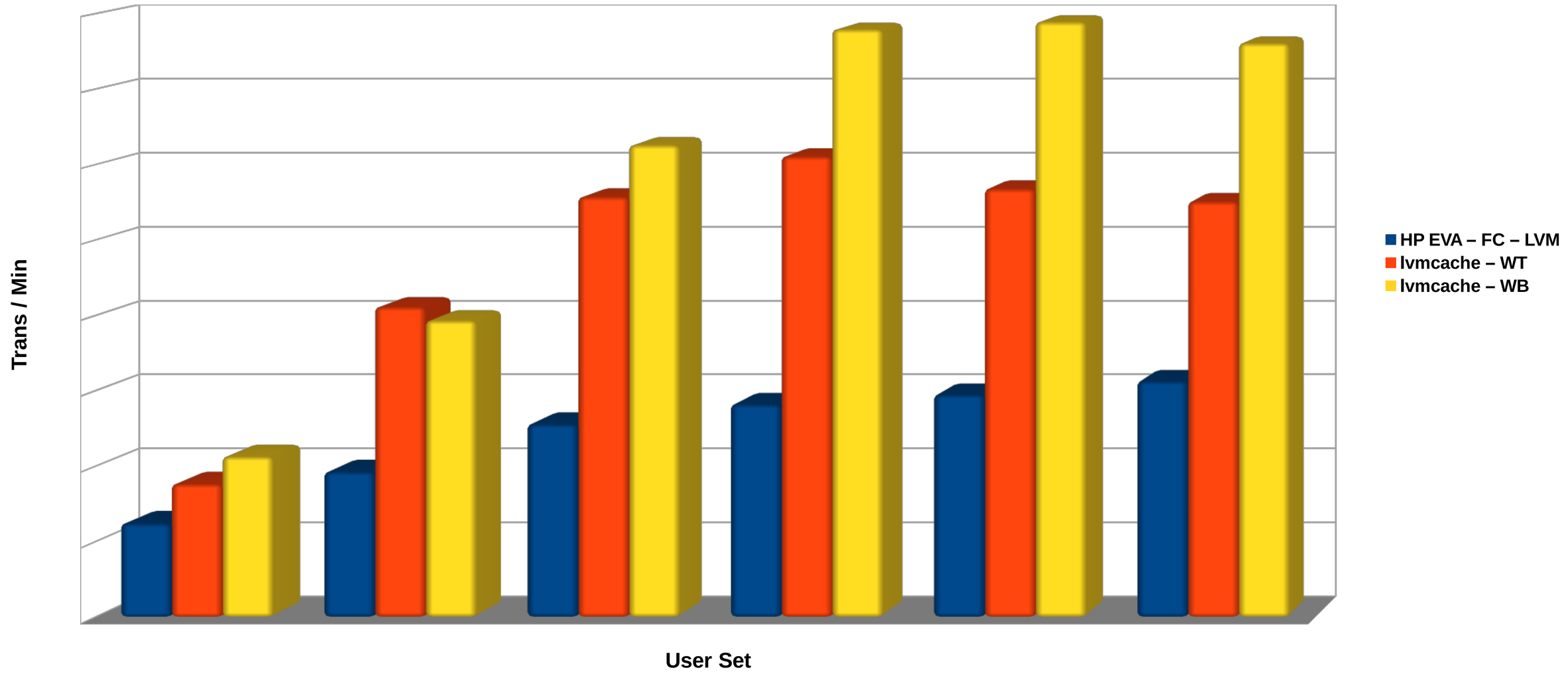
I/O Tuning – lvmcache

- Caching in the device mapper stack to improve performance
- Caching of frequently accessed data in a faster target

- For more information on how to configure cache device
  - man lvmcache

Database OLTP workload

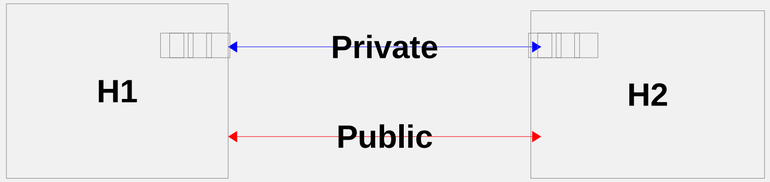
DM cache - 20G / Writethrough vs Writeback



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

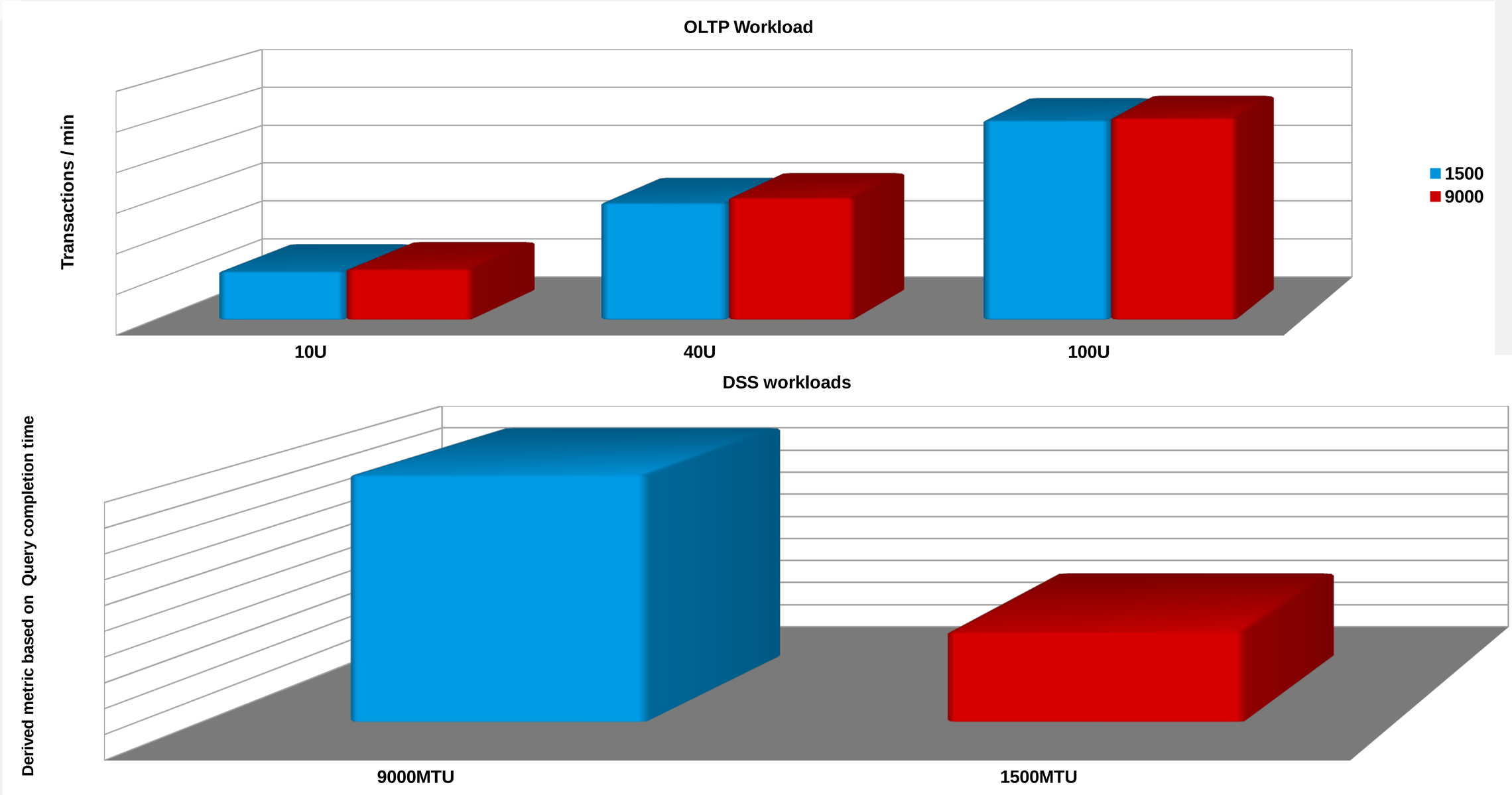


- **Hardware**
  - 10GigE
    - Supports RDMA w/ RHEL6 high performance networking package (**ROCE**)
  - Infiniband (Consider the cost factor)
- Packet size (Jumbo frames)

Linux Tool used for monitoring network

- *sar -n DEV <interval>*

Network tuning – Jumbo Frames with iSCSI storage



Output of “sar -N DEV 3”
For a DSS workload running on iSCSI storage using different MTUs

1500 MTU

	IFACE	rxpck/s	txpck/s	rxkB/s	txkB/s	rxcmp/s	txcmp/s	rxmcst/s
01:40:08 PM								
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

	IFACE	rxpck/s	txpck/s	rxkB/s	txkB/s	rxcmp/s	txcmp/s	rxmcst/s
06:58:43 PM								
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

Database Performance

- **Application tuning**
  - Design
  - Reduce locking / waiting
  - Database tools (optimize regularly)
- **I/O layout**
  - Transaction based
    - Database files / Redo logs / Rollback segments
  - Decision Support
    - Database files / Temp Segments
- Resiliency is not a friend of performance
  - Redo log file size (Find a balance between checkpointing and recovery time)
  - Redundancy (Hardware and Software)
  - Geo replication / Archiving logs

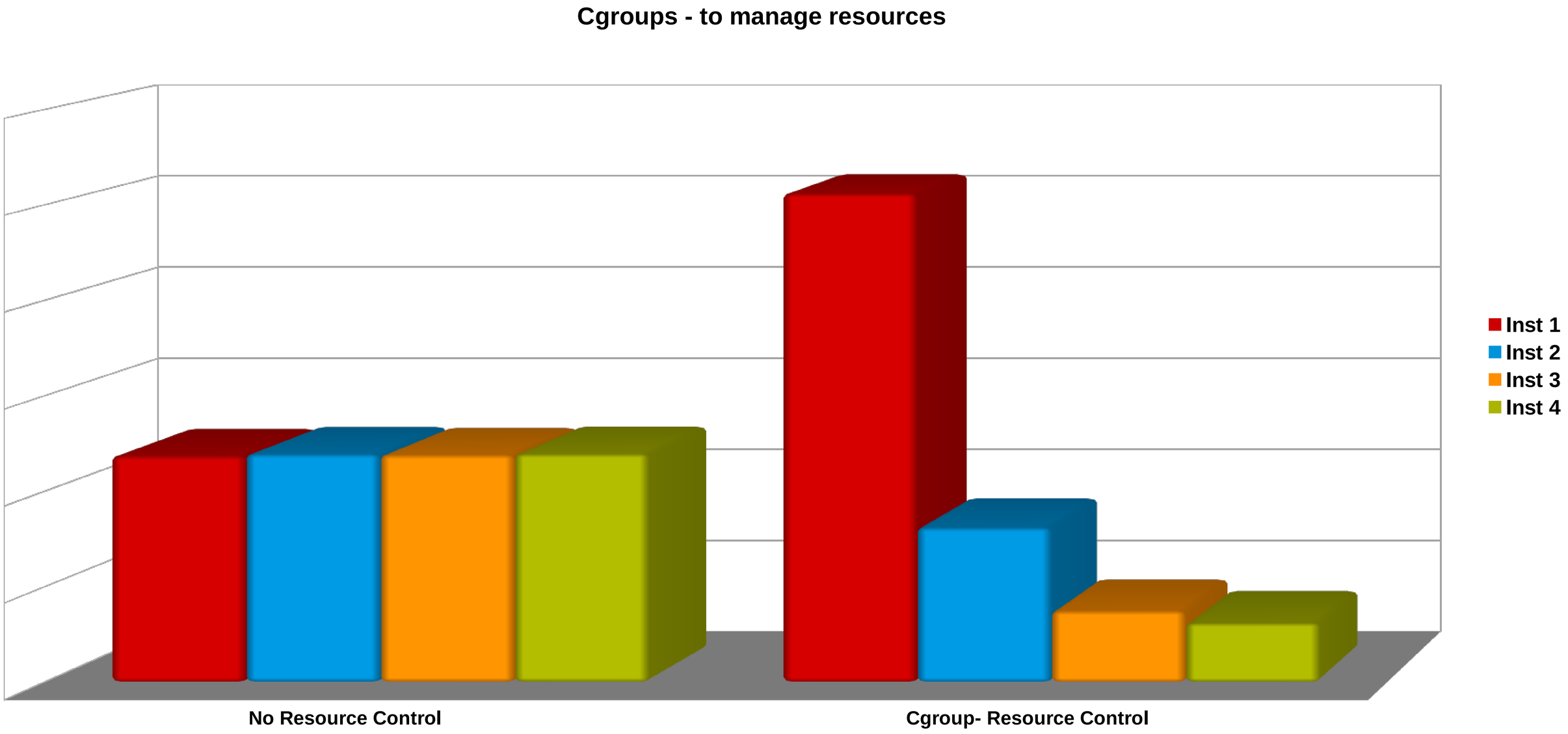
Tuning multiple database with Cgroups – RHEL 7 - Systemd

Resource Management

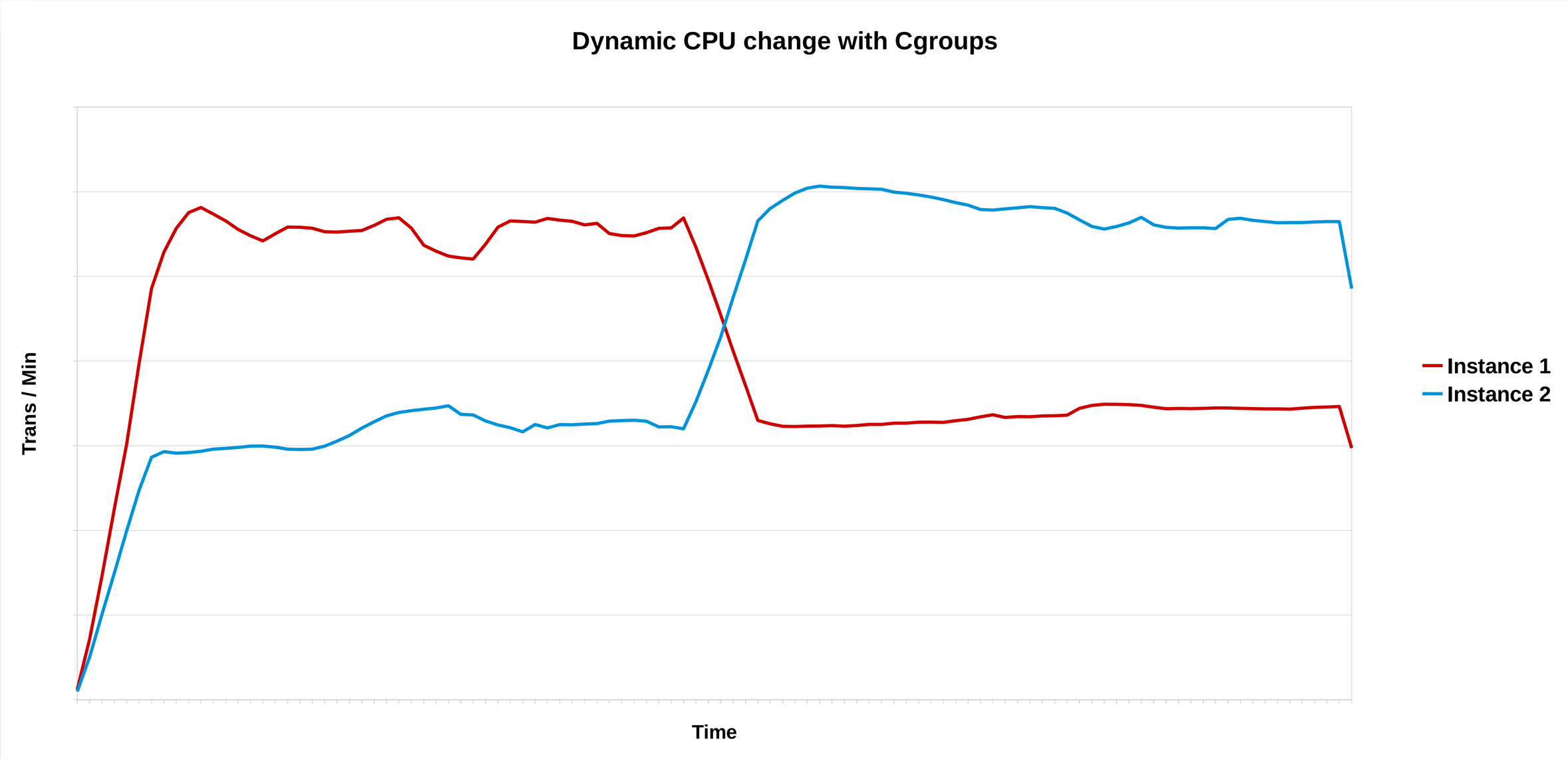
- Memory, cpus, IO, Network
- For performance
- For application consolidation
- Dynamic resource allocation

- **Application Isolation**
  - **Put less-critical or non-critical applications in controlled resource group to prevent them from hogging resources**
- **RHEL 7 Resource Management Guide**
  - https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/pdf/Resource_Management_Guide/Red_Hat_Enterprise_Linux-7-Resource_Management_Guide-en-US.pdf

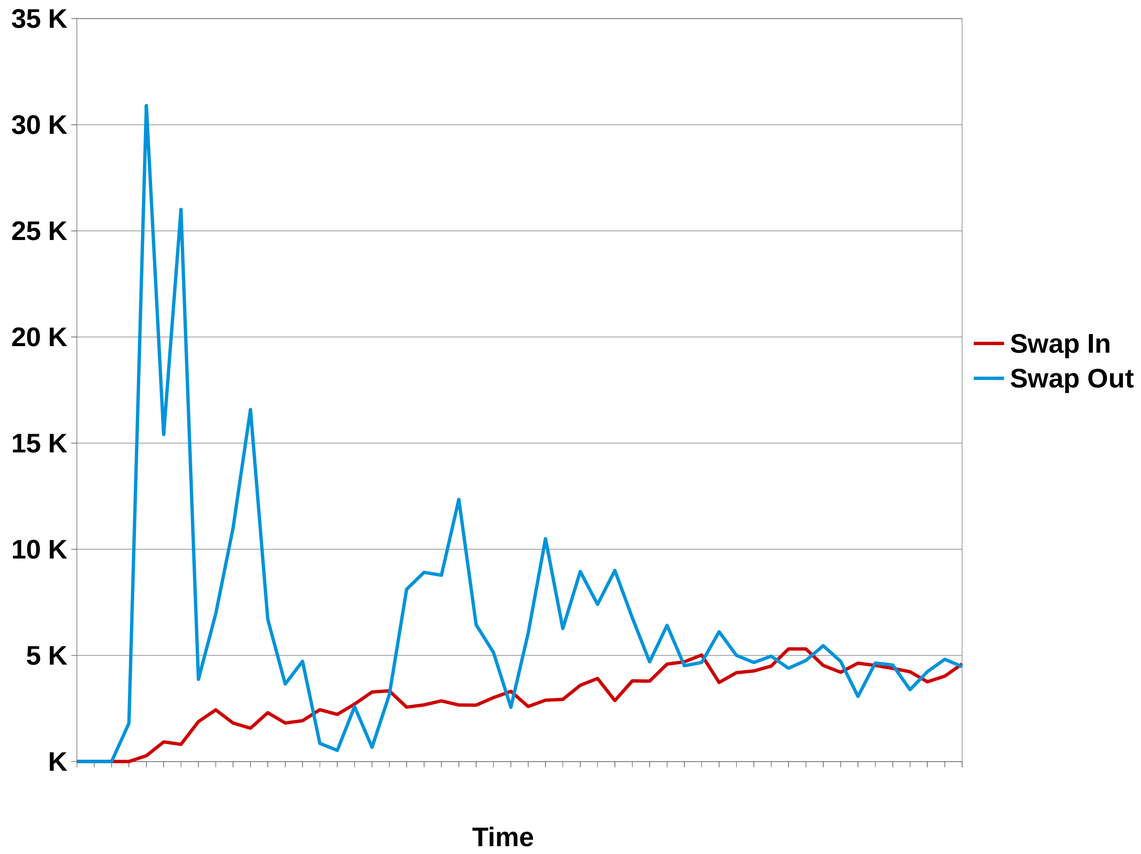
Cgroup – Resource management

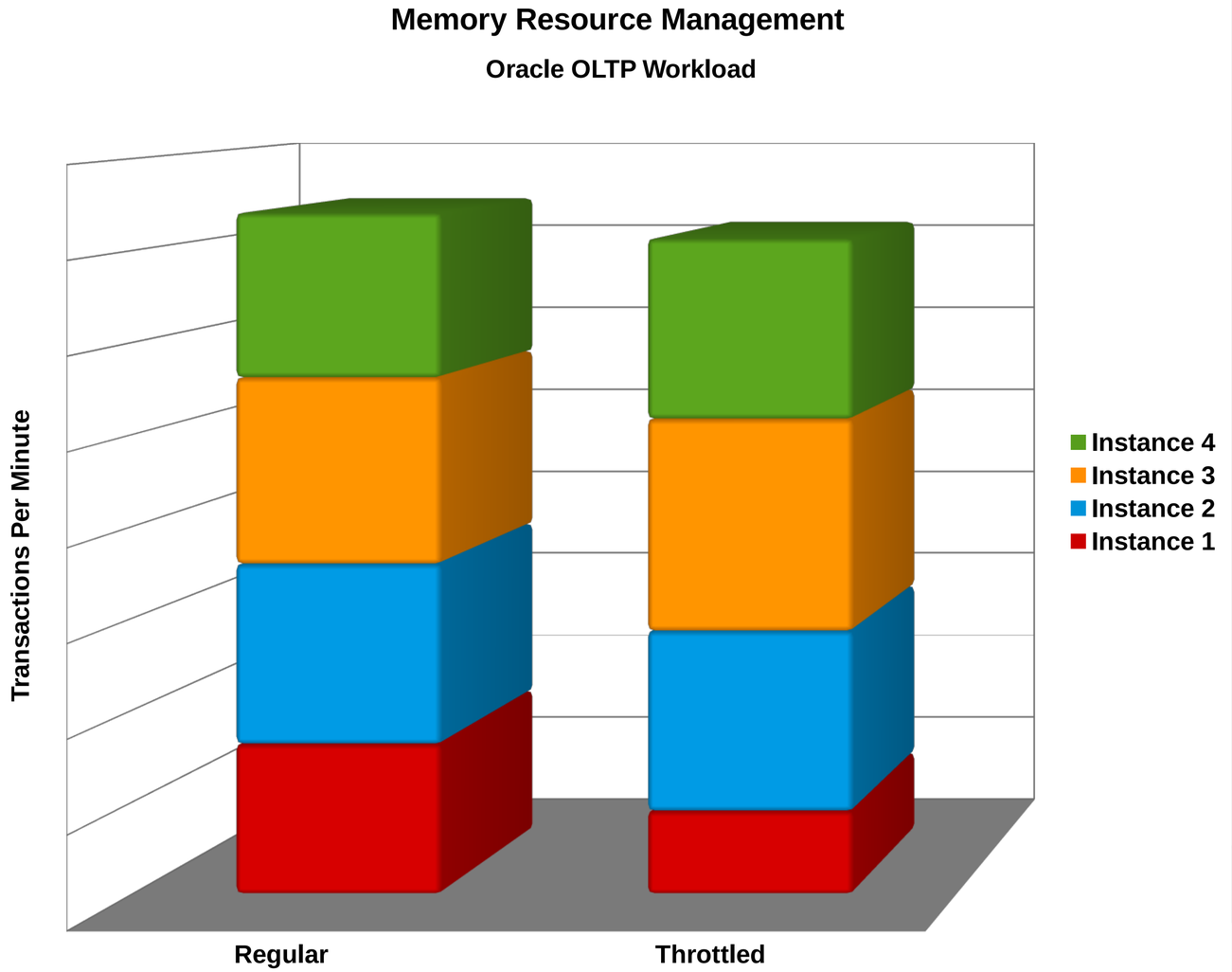


C-group - Dynamic resource control



Cgroup – Application Isolation





Even though one application does not have resources and starts swapping, other applications are not affected

Database in Virtual Machines

- Start with Virtual-Guest tuned profile in the VM
  - Set by default in VMs in Red Hat Enterprise Virtualization
  - Configure if you are running on other Virtualization platforms

- Follow the tuning methodology that we do on bare metal
  - Use Hugepages in VM
  - Use pass through technology (SR-IOV)

Tuned: Profile Inheritance for Virtual Machines

Parents

throughput-performance

Children

network-throughput

virtual-host

virtual-guest

Your-DB

Tunable	Units	throughput-performance	virtual-host	virtual-guest
Inherits From/Notes			throughput-performance	throughput-performance
sched_min_ granularity_ns	nanoseconds	10000000		
sched_wakeup_granularity_ns	nanoseconds	15000000		
dirty_ratio	percent	40		30
dirty_background_ratio	percent	10	5	10
swappiness	Weight 1-100	10		
I/O Scheduler (Elevator)				
Filesystem Barriers	Boolean			
CPU Governor		performance		
Disk Read-ahead	Bytes	4096		
Energy Perf Bias		performance		
kernel.sched_migration_cost_ns	nanoseconds		5000000	
min_perf_pct (intel_pstate only)	percent	100		

Database on RHV

- NUMA optimization (Auto numa / Manual pinning)
- Device Cache settings
  - None – for multiple VMS
  - Writethrough – for fewer VMs to take advantage of host cache
- IO drivers
  - Virtio – Dataplane – if you are using storage pool
  - Virtio-scsi with pass through – if you are using direct luns
- Migration bandwidth
- Use Host CPU pass through when migration is not a consideration
- Do not enable KSM for Linux VMs running databases

Linux Containers

- Keep application and their runtime components together
- Provides isolation[1]
- Image based deployment
- Allows independent updates to containers or platform
- Can be deployed on datacenter systems, private or public cloud
- Makes resource control easy - Cgroup integration
- Very little performance overhead

Mapping *tuned* profiles to Red Hat's product portfolio

RHEL Desktop/Workstation **balanced**	RHEL Server/HPC **throughput-performance**	
RHEL KVM Host, Guest **virtual-host/guest**	RHEV **virtual-host**	RHEL for Real Time **realtime**
Red Hat Storage **rhs-high-throughput, virt**	RHEL OSP (compute node) **virtual-host**	RHEL for Real Time KVM/NFV **realtime-virtual-host/guest**
RHEL Atomic **atomic-host, atomic-guest**	OpenShift **openshift-master, node**	

Performance Monitoring Tools

- Monitoring tools
  - top, vmstat, ps, iostat, netstat, sar, perf
- 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

Multi Instance OLTP Run without Huge Pages

```
File  Edit  View  Search  Terminal  Help
Samples: 9M of event 'cycles', Event count (approx.): 1641057303107
  3.41%  oracle                    [.] kcbgtcr
  1.42%  oracle                    [.] ktrexc
  1.29%  oracle                    [.] __intel_new_memcpy
  1.22%  oracle                    [.] __intel_new_memset
  1.13%  oracle                    [.] kcbgtcr
  1.10%  oracle                    [.] kcbgcur
  1.01%  oracle                    [.] kdxlrs2
  0.90%  oracle                    [.] opiexe
  0.85%  oracle                    [.] kdxbrs1
  0.80%  [kernel]                  [k] page_fault
  0.76%  oracle                    [.] kcbgtcrf
  0.59%  [kernel]                  [k] ipt_do_table
  0.57%  oracle                    [.] ktbgfi
  0.54%  [kernel]                  [k] _raw_spin_lock
  0.53%  oracle                    [.] opipls
  0.48%  oracle                    [.] ktrexc
  0.47%  [kernel]                  [k] __radix_tree_lookup
  0.46%  perf                      [.] 0x0000000000065d66
  0.45%  [kernel]                  [k] tg_load_down
  0.45%  oracle                    [.] __intel_new_memcpy
  0.42%  oracle                    [.] __intel_new_memset
  0.41%  oracle                    [.] kduovw
  0.41%  perf                      [.] dso__find_symbol
  0.40%  oracle                    [.] kslfre
  0.38%  oracle                    [.] kcrfw_redo_gen
  0.38%  oracle                    [.] ktuchg2
  0.37%  oracle                    [.] kcbgcur
  0.37%  [vdso]                    [.] 0x0000000000000cd0
  0.37%  [kernel]                  [k] _raw_spin_lock_irqsave
  0.37%  [kernel]                  [k] update_cfs_rq_blocked_load
  0.37%  oracle                    [.] kslgetl
  0.36%  oracle                    [.] kdudcp
```

Performance Monitoring Tool – perf record / report

Multi Instance OLTP Run with Huge Pages

root@perf30:~

File Edit View Search Terminal Help

```
amples: 1M of event 'cycles', Event count (approx.): 656344167175
 2.62%           oracle  oracle                   [.] kcbgtcr
 1.55%           oracle  oracle                   [.] ktrexc
 1.36%           oracle  oracle                   [.] __intel_new_memcpy
 1.35%           oracle  oracle                   [.] __intel_new_memset
 1.07%           oracle  oracle                   [.] kdxlrs2
 0.96%           oracle  oracle                   [.] kcbgcur
 0.92%           oracle  oracle                   [.] opiexe
 0.91%           oracle  oracle                   [.] kcbgtcr
 0.90%           oracle  oracle                   [.] kdxbrs1
 0.63%           oracle  oracle                   [.] ktbgfi
 0.59%           oracle  oracle                   [.] ktrexc
 0.57%           oracle  oracle                   [.] opipls
 0.54%           oracle  oracle                   [.] kcbgtcrf
 0.52%           oracle  oracle                   [.] __intel_new_memcpy
 0.50%           oracle  oracle                   [.] __intel_new_memset
 0.47%           oracle  [kernel.kallsyms]        [k] tg_load_down
 0.44%           oracle  oracle                   [.] kslfre
 0.44%           oracle  oracle                   [.] kcbchg1_main
 0.44%           oracle  oracle                   [.] kcrfw_redo_gen
 0.44%           oracle  oracle                   [.] kduovw
 0.42%           oracle  [vdso]                   [.] 0x0000000000000cd0
 0.42%           oracle  oracle                   [.] ktuchg2
 0.41%           oracle  oracle                   [.] kdxlrs2
 0.40%           oracle  oracle                   [.] kdudcp
 0.39%          swapper  [kernel.kallsyms]        [k] intel_idle
 0.38%           oracle  oracle                   [.] kpobii
 0.36%  runoastoltpb.ex  libclntsh.so.11.1        [.] ttcacr
 0.35%           oracle  oracle                   [.] kslgetl
 0.34%           oracle  oracle                   [.] kcbgcur
 0.34%           oracle  [kernel.kallsyms]        [k] ipt_do_table
 0.33%           oracle  [kernel.kallsyms]        [k] update_cfs_rq_blocked_load
 0.33%           oracle  oracle                   [.] opiexe
 0.33%           oracle  oracle                   [.] kdimodnu0
 0.32%           oracle  oracle                   [.] kdiins0
```

Valuable Links

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