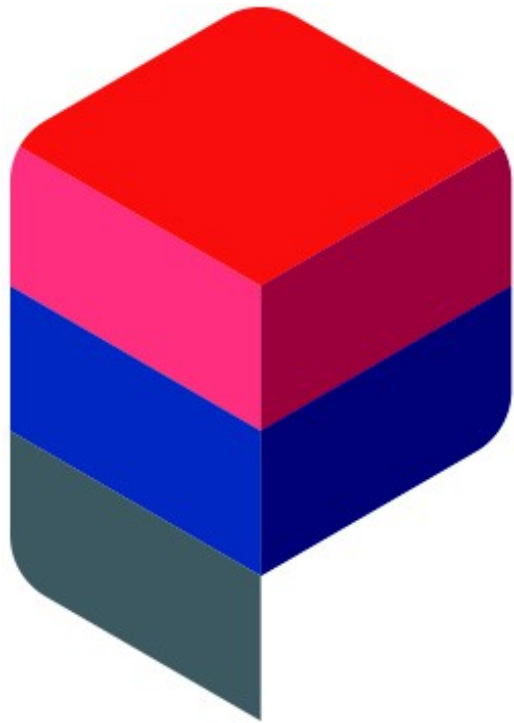




**performing
databases**

YOUR machine and MY database!

– a performing relationship?

Martin Klier

Performing Databases GmbH
Mitterteich



ORACLE[®]
ACE Director



performing
databases

Martin Klier // twitter.com/MartinKlierDBA
www.performing-databases.com

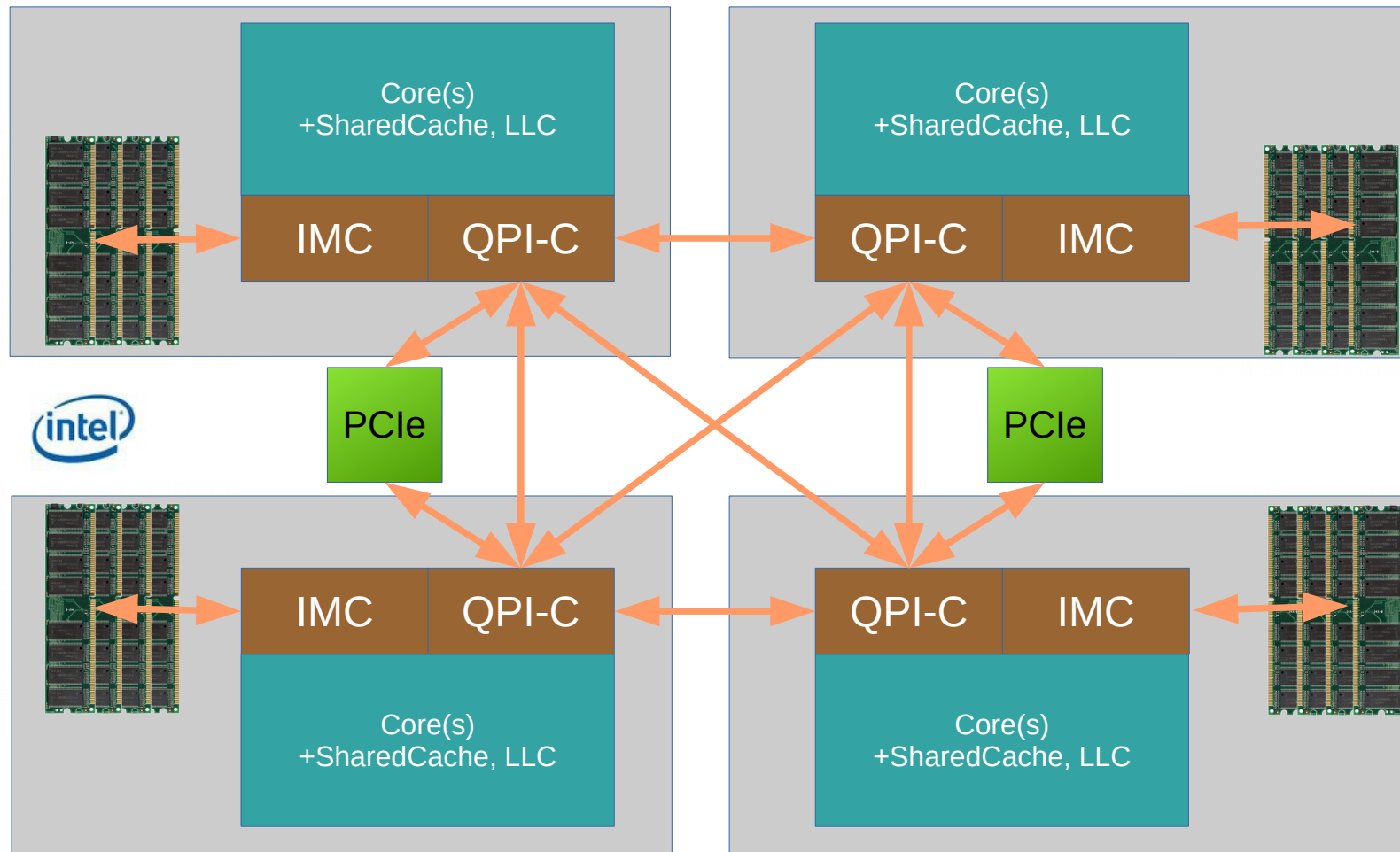
Today ...

- Introduction
- NUMA + Huge Pages
- Disk IO
- Concurrency
- Collaborate!

Server and CPU

NUMA

Non-uniform Memory Access

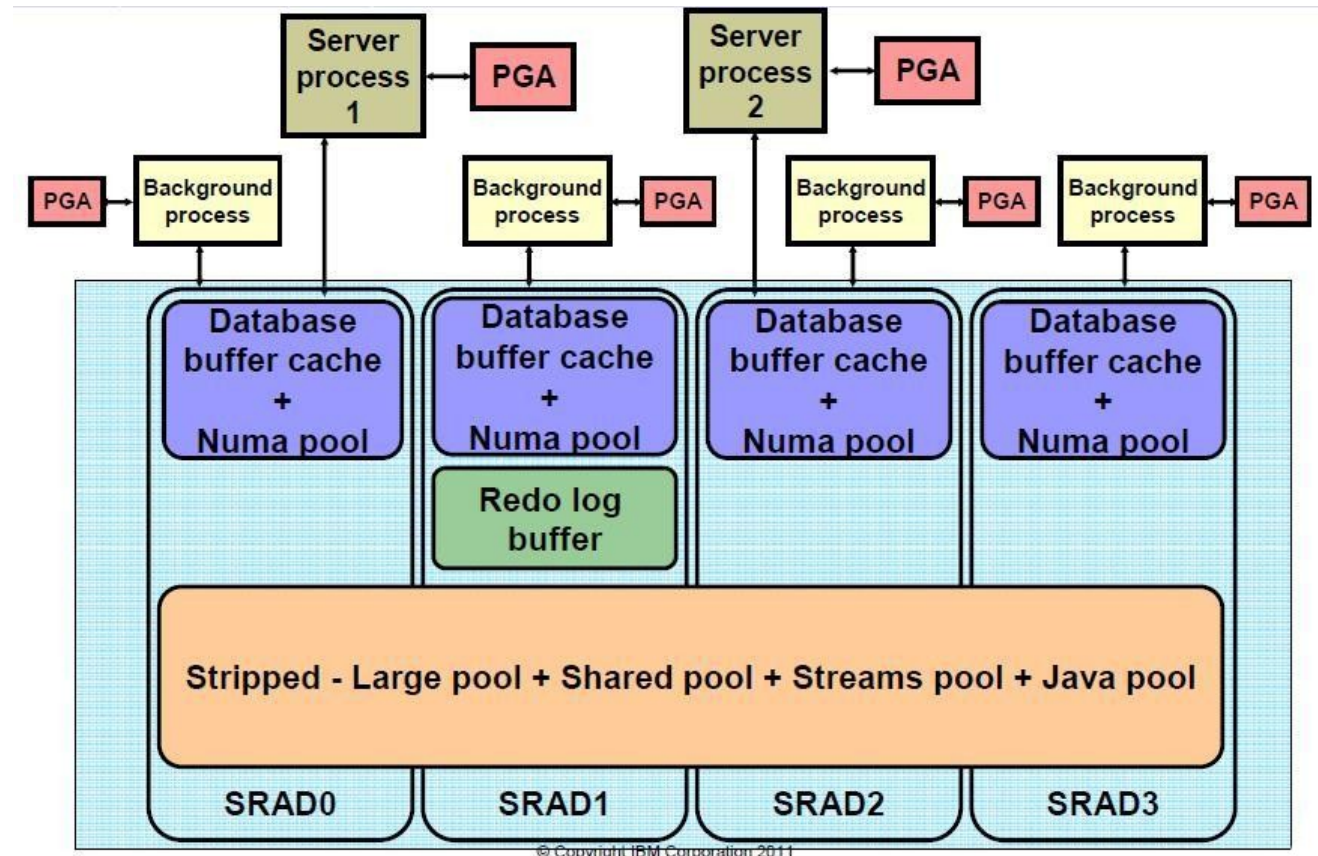


NUMA

Non-uniform Memory Access

`_enable_NUMA_support = TRUE`
MOS Doc ID 864633.1

- Multiple Buffer Caches
- Striped pools
- => cross context :((
- => pool access :(



NUMA

Non-uniform Memory Access

```
[root@ora05 ~]# numactl --hardware
available: 2 nodes (0-1)
node 0 size: 32756 MB
node 0 free: 608 MB
node 1 size: 28672 MB
node 1 free: 1343 MB
node distances:
node    0    1
  0:   10   21
  1:   21   10
```

```
select * from V$SGA_DYNAMIC_COMPONENTS;
```

frageergebnis x

SQL | Alle Zeilen abgerufen: 14 in 0,02 Sekunden

COMPONENT	CURRENT_SIZE	MIN_SIZE	MAX_SIZE
1 shared pool	3489660928	2952790016	3489660928
2 large pool	67108864	0	67108864
3 java pool	67108864	67108864	67108864
4 streams pool	134217728	0	134217728
5 DEFAULT buffer cache	26038239232	26038239232	26709327872



NUMA

Non-uniform Memory Access

```
[root@ora05 ~]# ipcs -ma
```

```
----- Shared Memory Segments -----
```

key	shmid	owner	perms	bytes	nattch	status
0x740301e9	2457600	root	600	4	0	
0x00000000	2752513	root	644	80	2	
0x00000000	2785282	root	644	16384	2	
0x00000000	2818051	root	644	280	2	
0x00000000	2883588	oracle	640	4096	0	
0x00000000	2916357	oracle	640	4096	0	
0xed304ac0	2949126	oracle	640	4096	0	
0x00000000	3735559	oracle	640	138412032	464	
0x00000000	3768328	oracle	640	8388608	464	
0x00000000	3801097	oracle	640	13555990528	464	
0x00000000	3833866	oracle	640	13623099392	464	
0x00000000	3866635	oracle	640	2684354560	464	
0xc93391ac	3899404	oracle	640	2097152	464	

One buffer cache
for each node



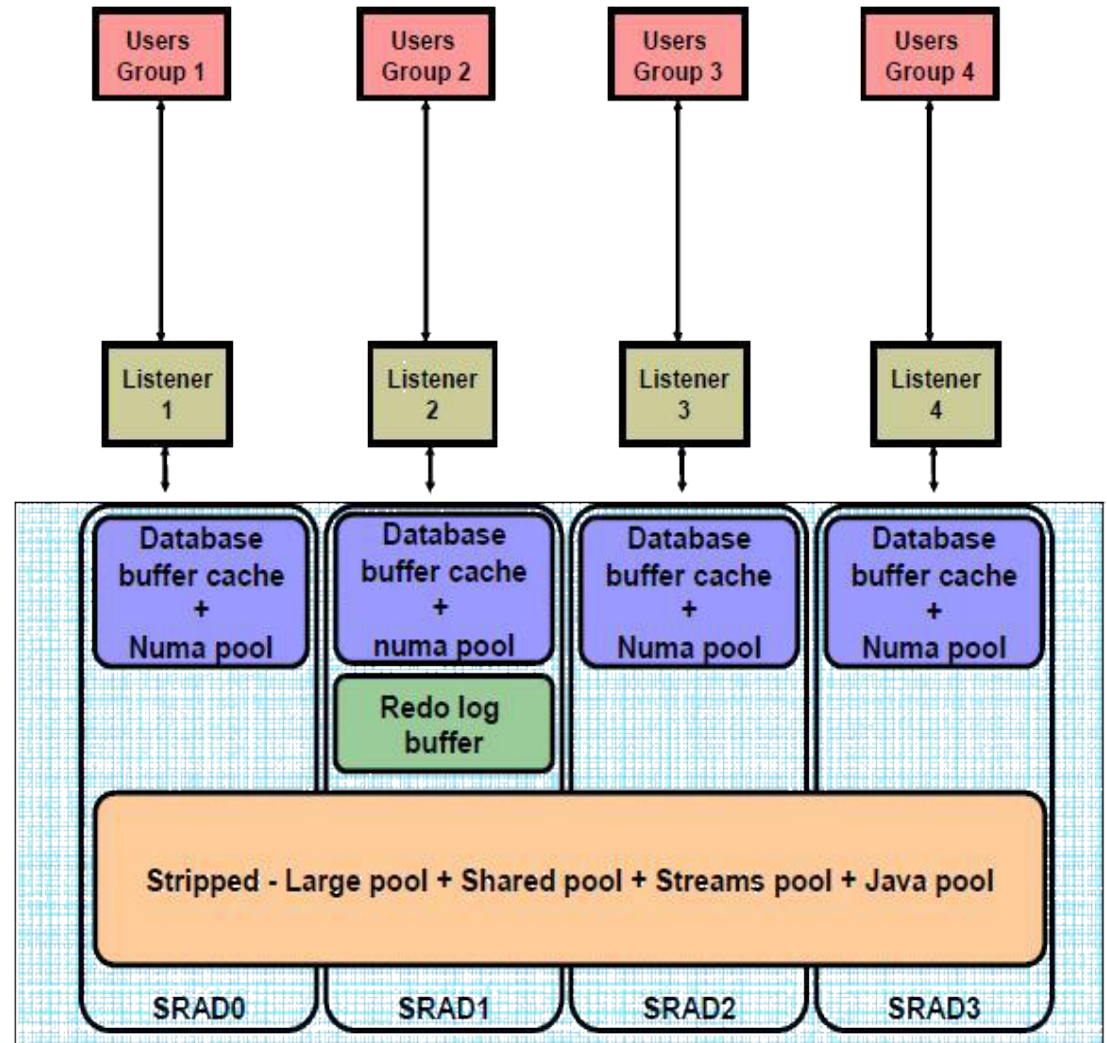
13GB+13GB=26 GB

NUMA

Non-uniform Memory Access

`_enable_NUMA_support = TRUE`
MOS Doc ID 864633.1

- Partitioned access
- Can be up to 40% faster
- But....



© Copyright IBM Corporation 2011

NUMA

Non-uniform Memory Access

Non-NUMA

with NUMA

node1 per second	node0 per second		Node1 diff p s	Node0 diff p s	
6364,2	16011,4	numa_hit	3538,6	10179	numa_hit
0,2	0,7	numa_miss	0	0	numa_miss
0,7	0,2	numa_foreign	0	0	numa_foreign
0	0	interleave_hit	0	0	interleave_hit
6364,2	16011,3	local_node	3538,6	10178,9	local_node
0,2	0,7	other_node	0	0,1	other_node

With my workload and only one listener:
Saved <1 page alloc miss per second



NUMA

Non-uniform Memory Access

```
[root@ora05 ~]# numactl --hardware
```

```
available: 2 nodes (0-1)
```

```
node 0 size: 32756 MB
```

```
node 0 free: 608 MB
```

```
node 1 size: 28672 MB
```

```
node 1 free: 1343 MB
```

```
node distances:
```

```
node    0    1
```

```
  0:   10   21
```

```
  1:   21   10
```

So WHY?

```
select * from V$SGA_DYNAMIC_COMPONENTS;
```

frageergebnis *

SQL | Alle Zeilen abgerufen: 14 in 0.02 Sekunden

COMPONENT	CURRENT_SIZE	MIN_SIZE	MAX_SIZE
1 shared pool	3489660928	2952790016	3489660928
2 large pool	67108864	0	67108864
3 java pool	67108864	67108864	67108864
4 streams pool	134217728	0	134217728
5 DEFAULT buffer cache	26038239232	26038239232	26709327872

26 GB

My Buffer Cache **Fits** into RAM of one node anyway.

LINUX NUMA optimization at work.

NUMA

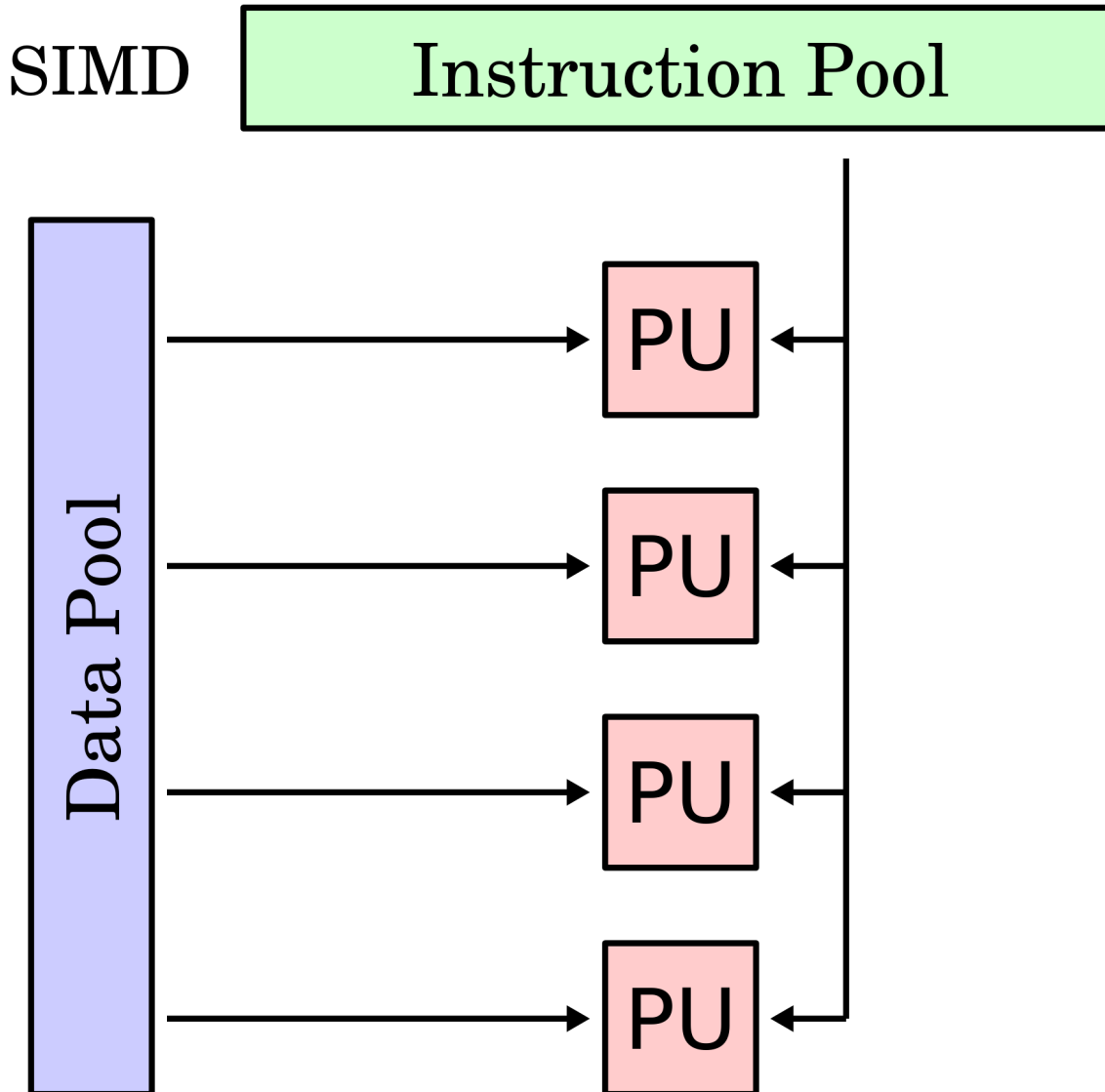
Non-uniform Memory Access

- Useful in big environments only
[think: DB consolidation]
- Make friends with the system admin,
have a joint opinion
- Test thoroughly and quantify use vs. effort
[think: bugs]

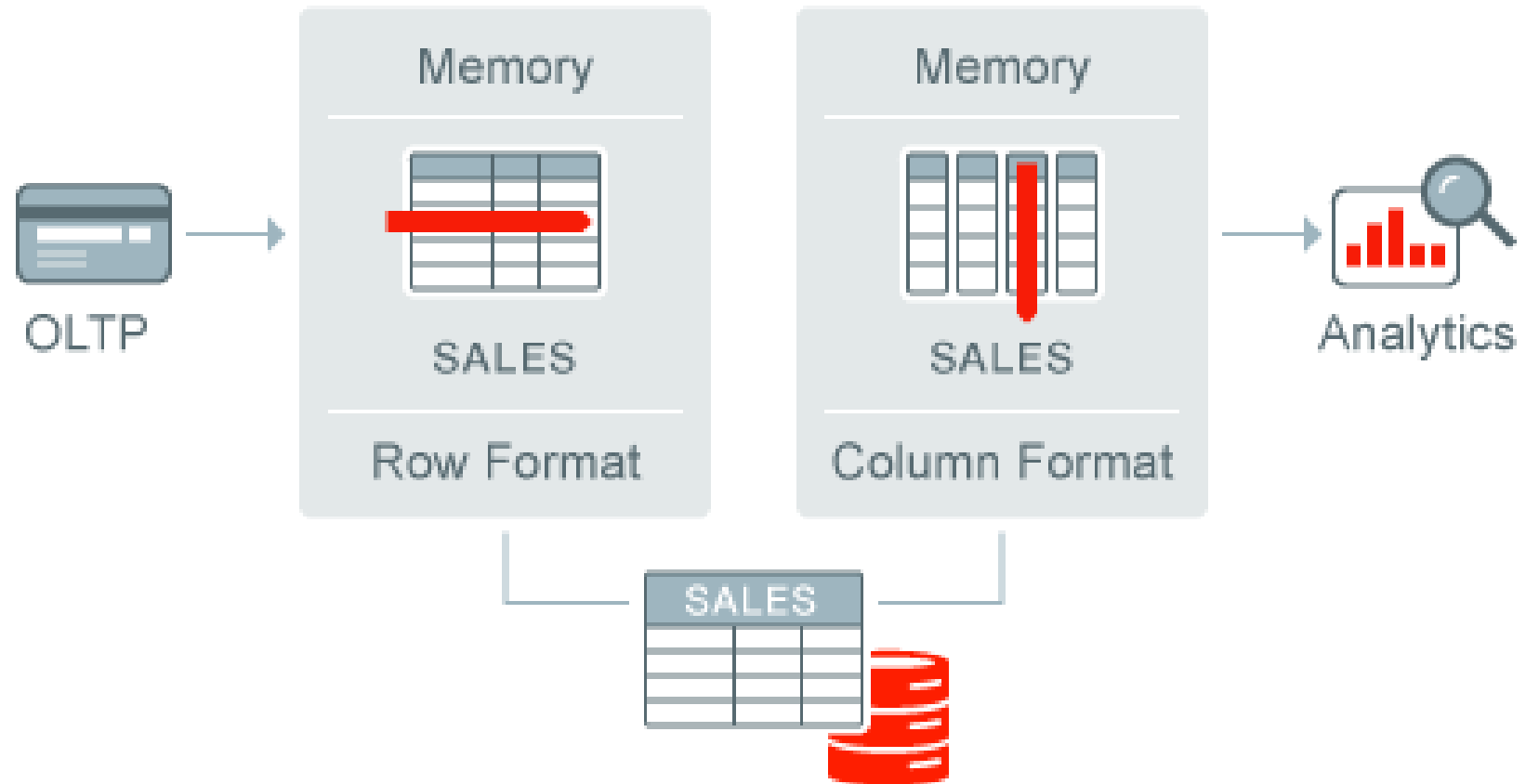
SIMD

Single Instruction Multiple Data

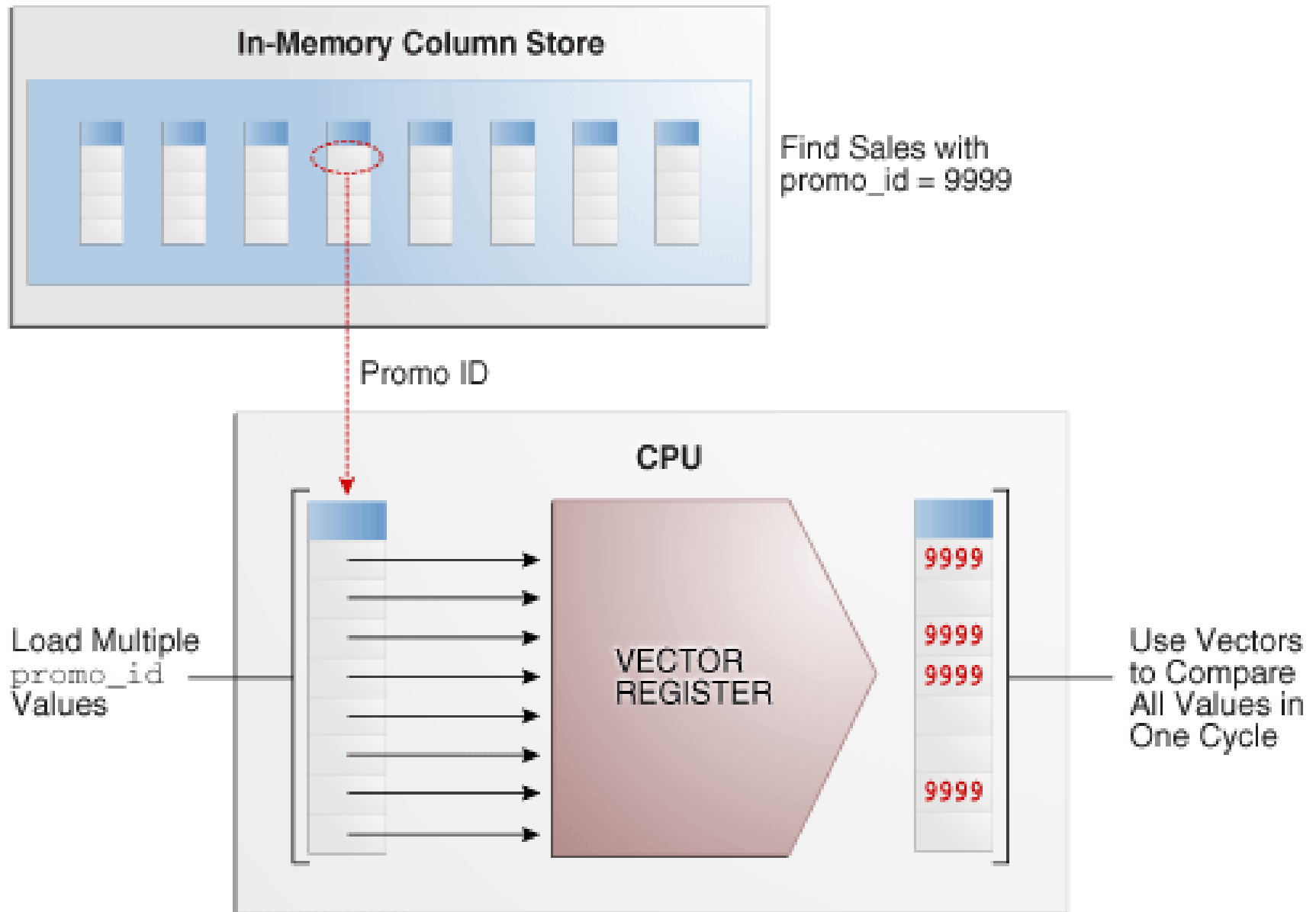
SIMD



Oracle InMemory Column Store



Oracle IMCS using SIMD

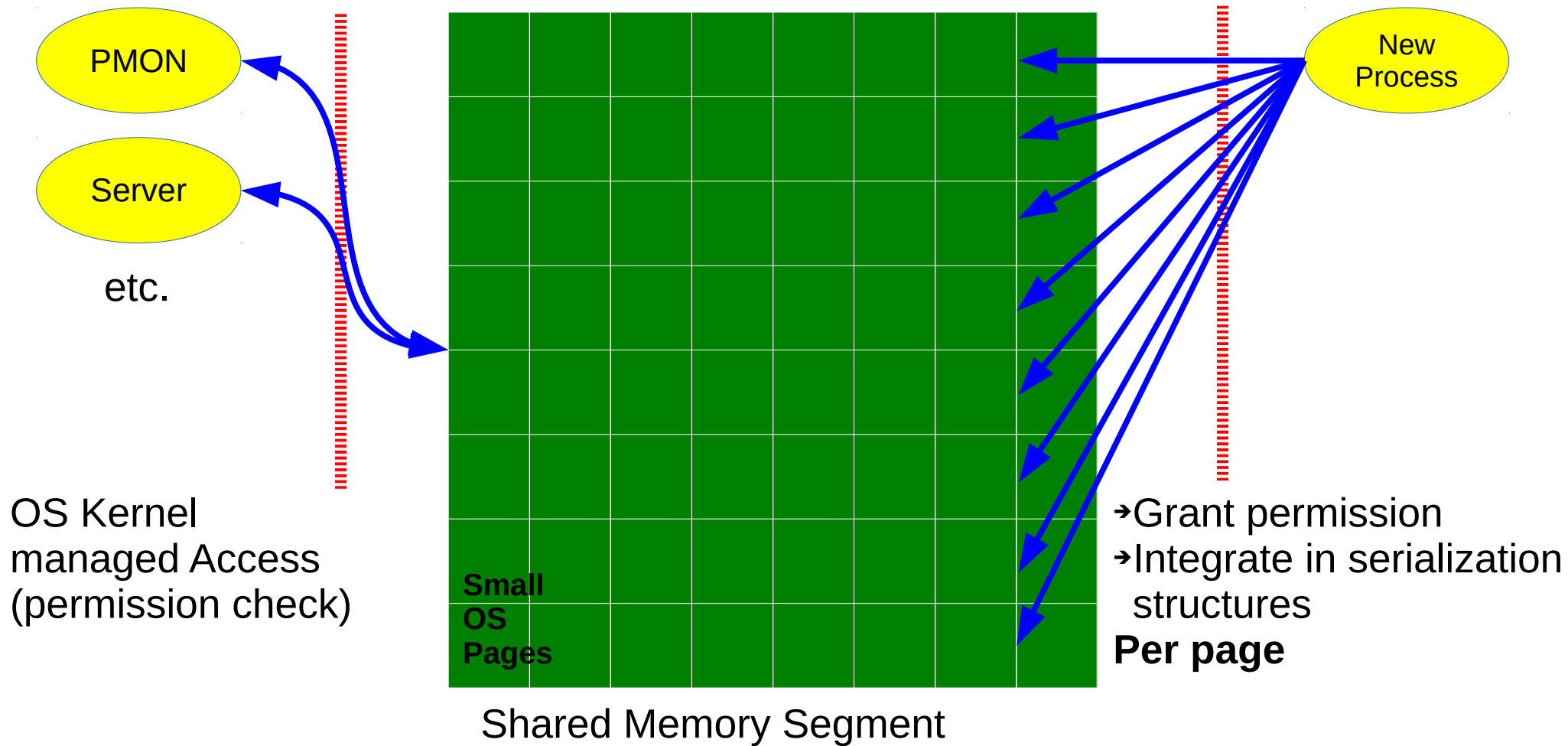


Server and RAM

RAM



RAM

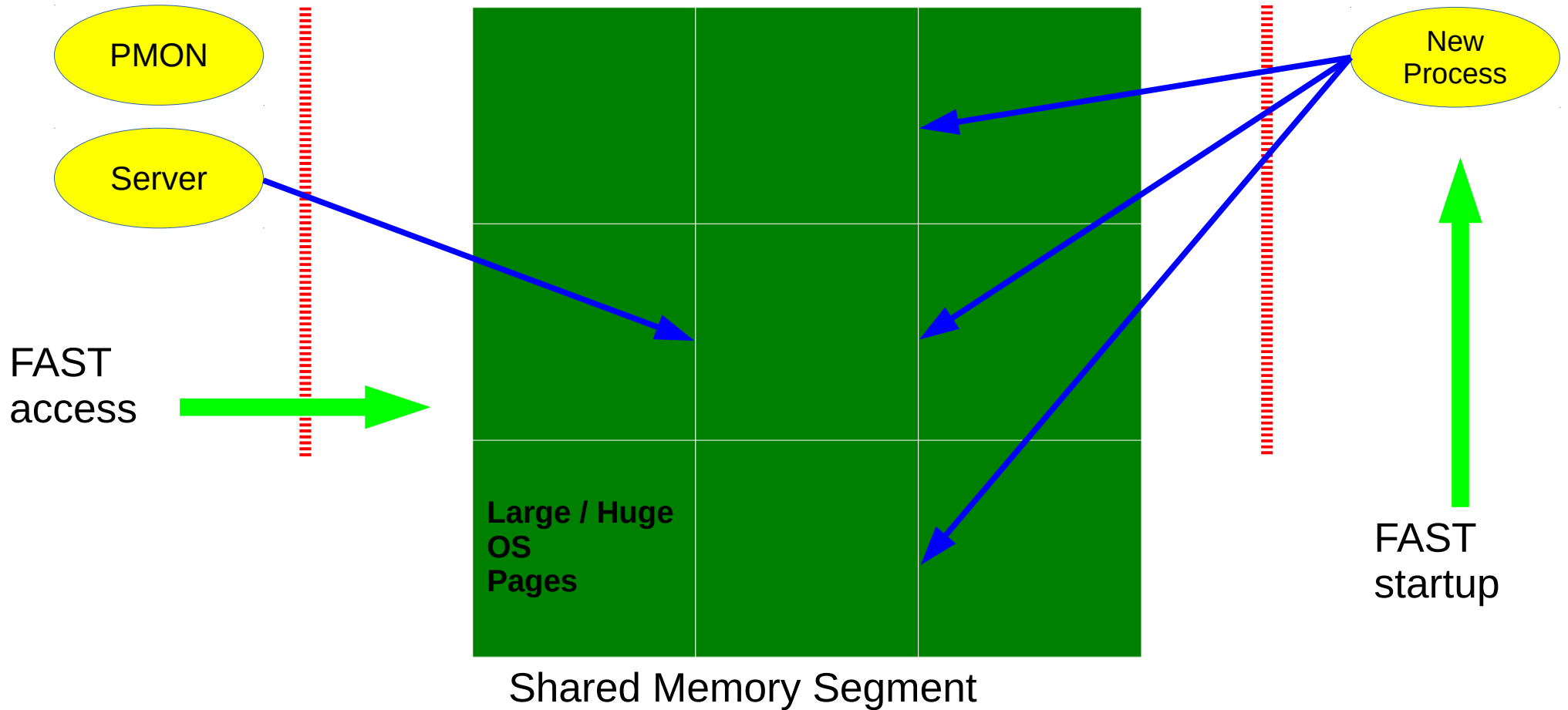


RAM

Challenges

- Memory Fragmentation (whatever that means)
- Wasting CPU with page alloc
- OS_THREAD_STARTUP waits

Huge Pages



Huge Pages

```
[root@ora05 ~]# cat /proc/meminfo
MemTotal:        61956468 kB
MemFree:         2470660 kB
Buffers:         435624 kB
Cached:         14848192 kB
SwapCached:      1200 kB
---
HugePages_Total: 17408
HugePages_Free:  3164
HugePages_Rsvd:  67
HugePages_Surp:  0
Hugepagesize:    2048 kB
DirectMap4k:     6144 kB
```

$$(17408 - 3164) * 2048 \text{ kB} = 28 \text{ GB}$$

Starting ORACLE instance (normal)

***** Large Pages Information *****

Total Shared Global Region in Large Pages = 28 GB (100%)

Large Pages used by this instance: 14337 (28 GB)

Large Pages unused system wide = 3071 (6142 MB) (alloc incr 64 MB)

Large Pages configured system wide = 17408 (34 GB)

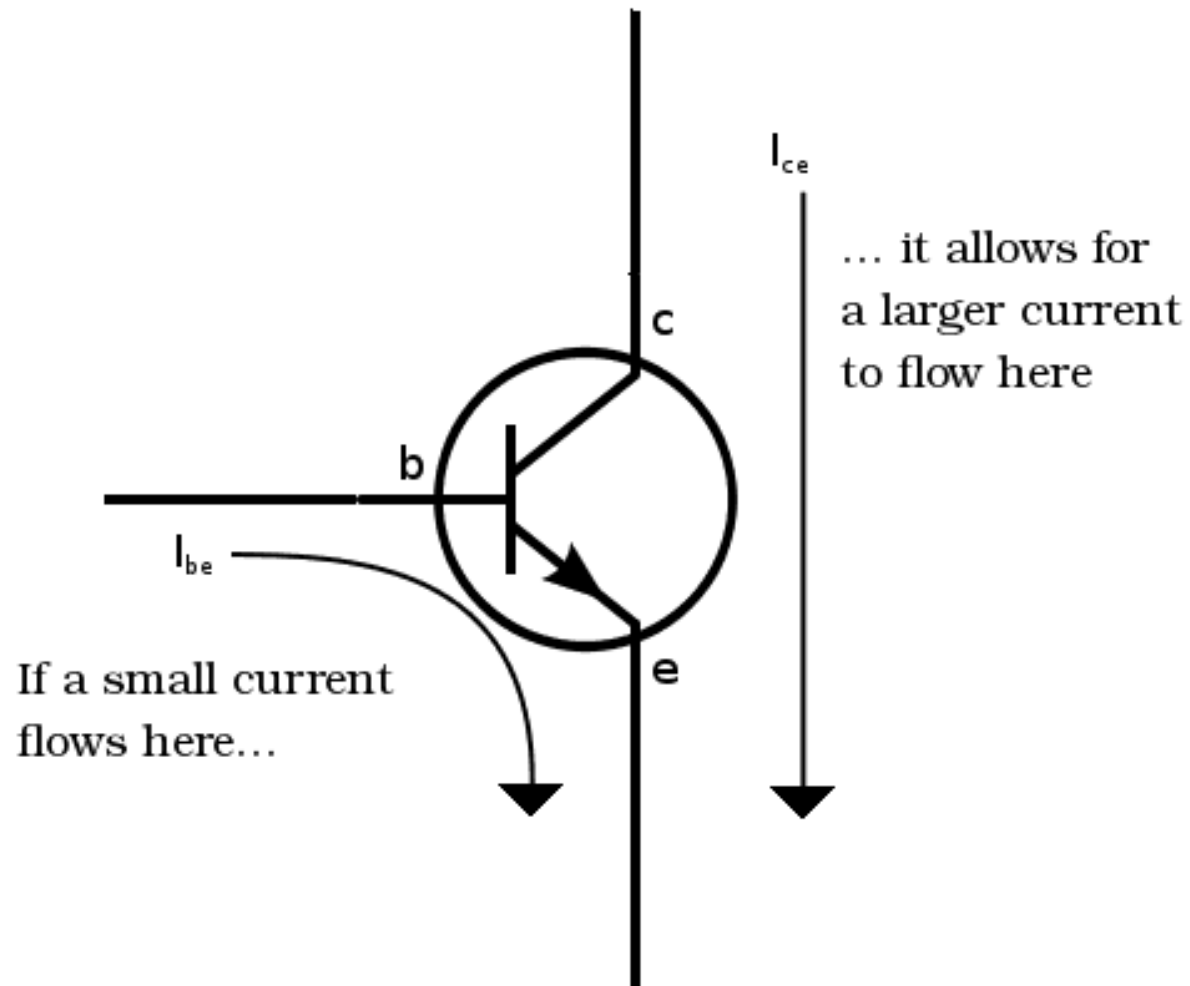
Large Page size = 2048 KB

Huge Pages

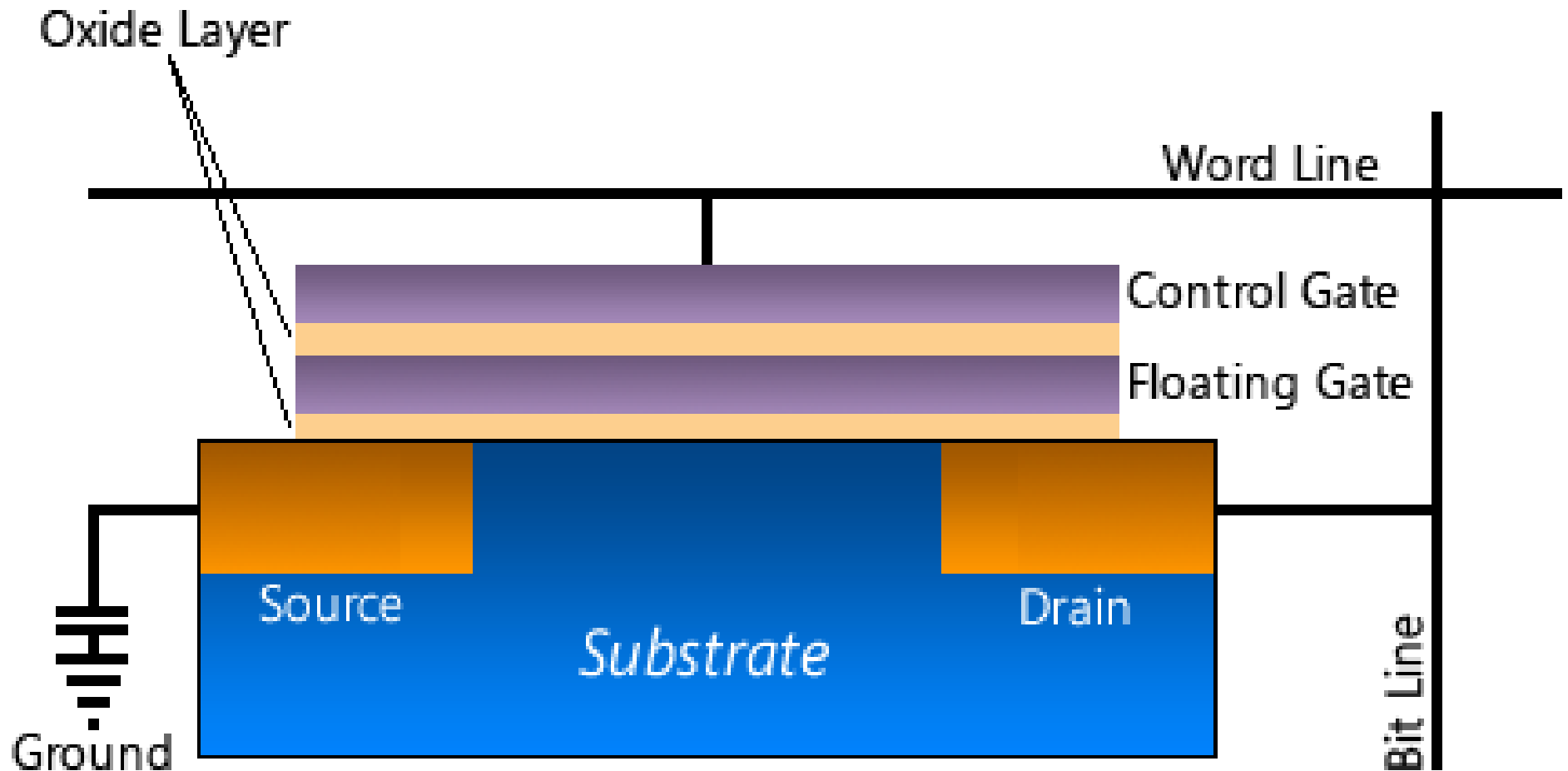
- Useful with SGA $\geq 16\text{GB}$
- Use largest available & sane page size
- Talk your sysadmin into DOing IT
- Combine with `PRE_PAGE_SGA=TRUE`

Storage and SSD

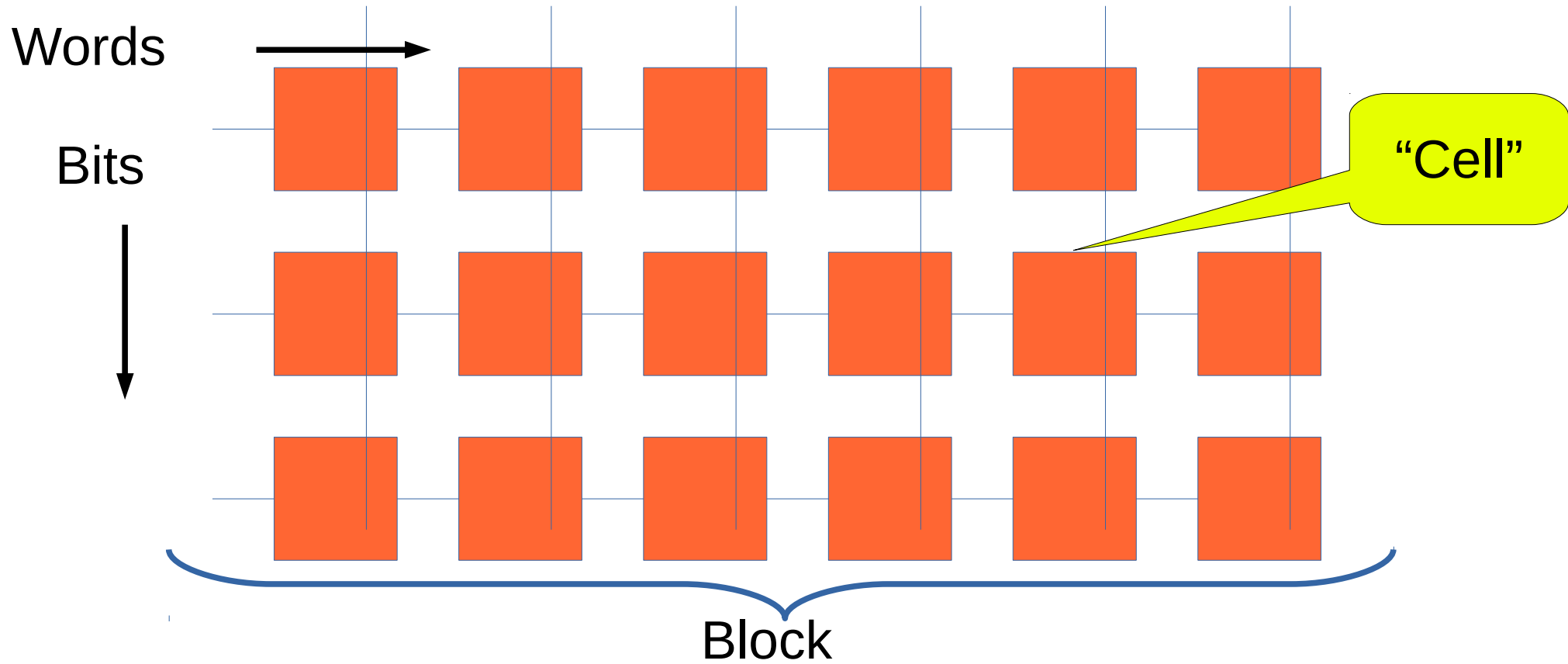
Transistor



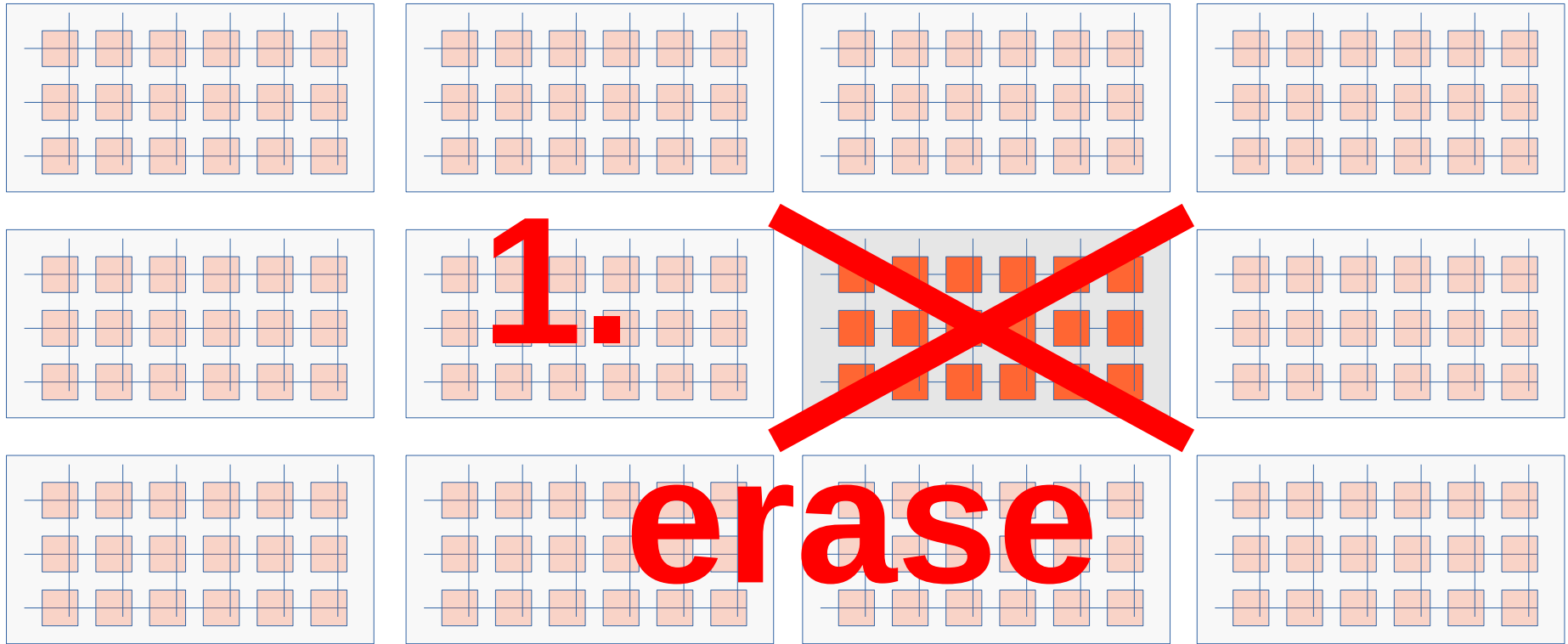
Flash Cell



SSD



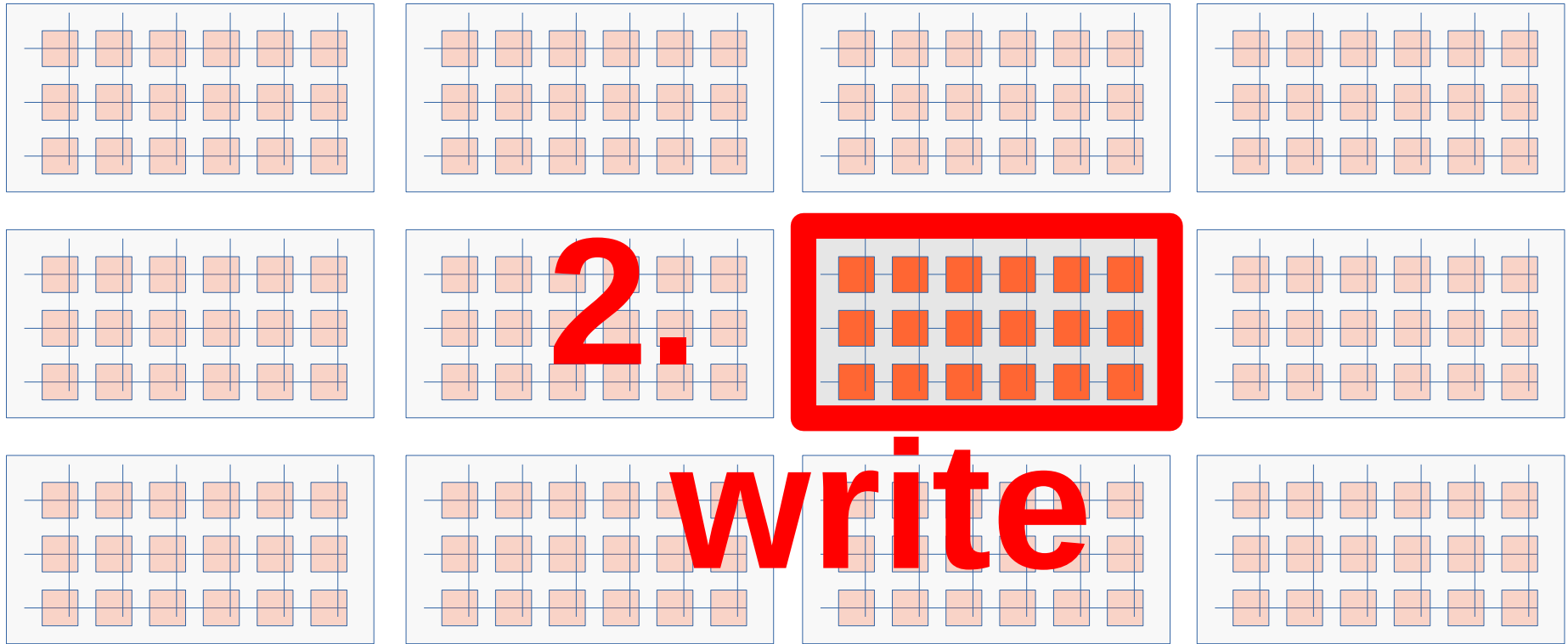
SSD



16kB – 512kB pro Block



SSD



16kB – 512kB pro Block

ORION on SSD

ORION VERSION 11.1.0.7.0

8k/16k blocks

80% write
20% read

Commandline:

-run advanced -testname oltp-write -num_disks 1 -cache_size 8192 -size_small 8 -size_large

This maps to this test:

Test: oltp-write

Small IO size: 8 KB

Large IO size: 16 KB

IO Types: Small Random IOs, Large Random IOs

Simulated Array Type: RAID 0

Stripe Depth: 1024 KB

Write: 80%

Cache Size: 8192 MB

Duration for each Data Point: 30 seconds

Small Columns:, 0

Large Columns:, 0, 1, 2

Total Data Points: 8

Today: 15.000 IOPS

Name: /media/KLMHIGHSPEED/o1_mf_sysaux_4zjblvr4_.dbf Size: 1835016192

1 FILES found.

Maximum Large MBPS=58.51 @ Small=0 and Large=2

Maximum Small IOPS=8171 @ Small=3 and Large=0

Minimum Small Latency=0.14 @ Small=1 and Large=0

8171 IOPS
like 60 HDDs

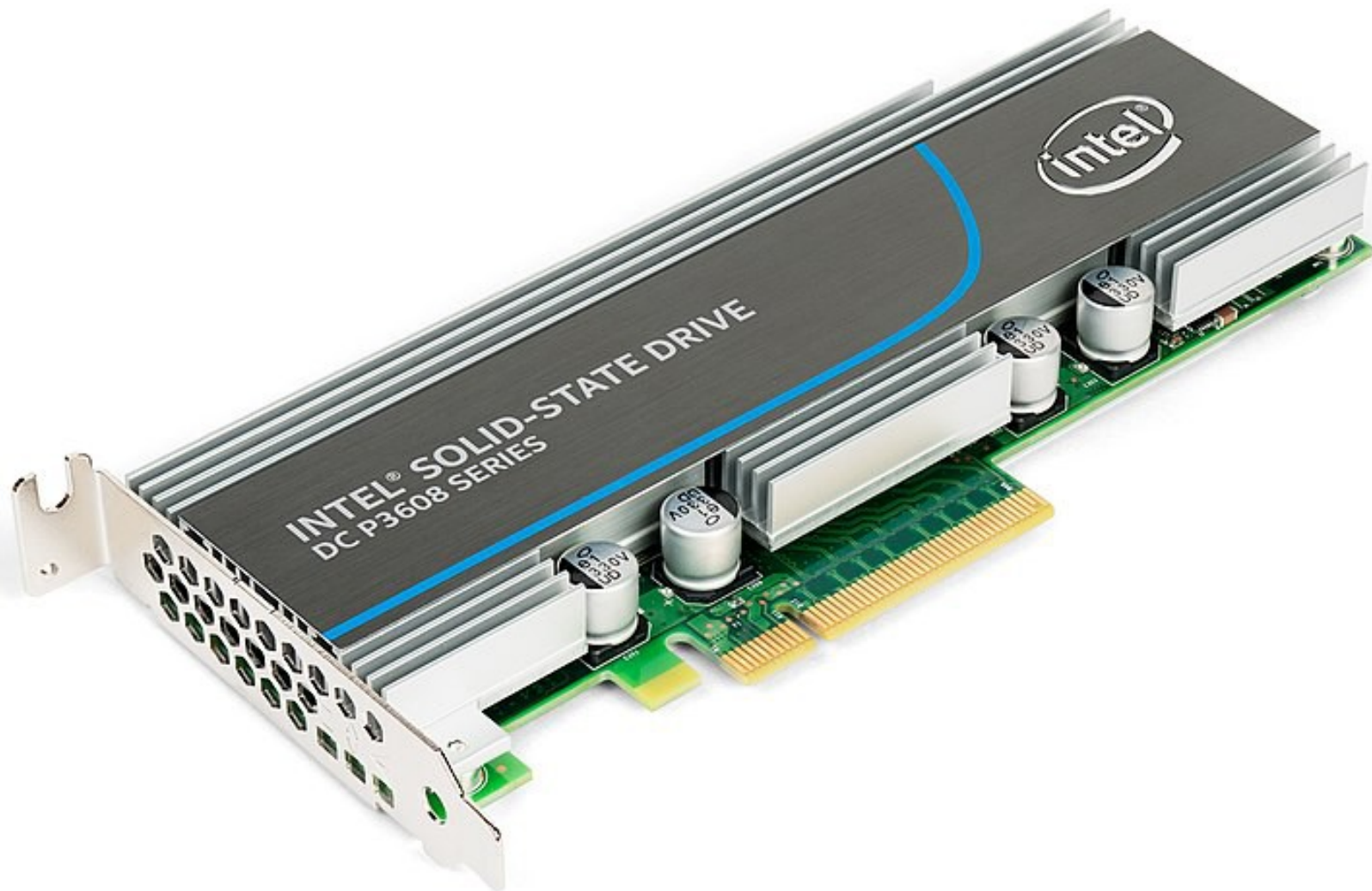


performing
databases

Samsung SSD 840 PRO

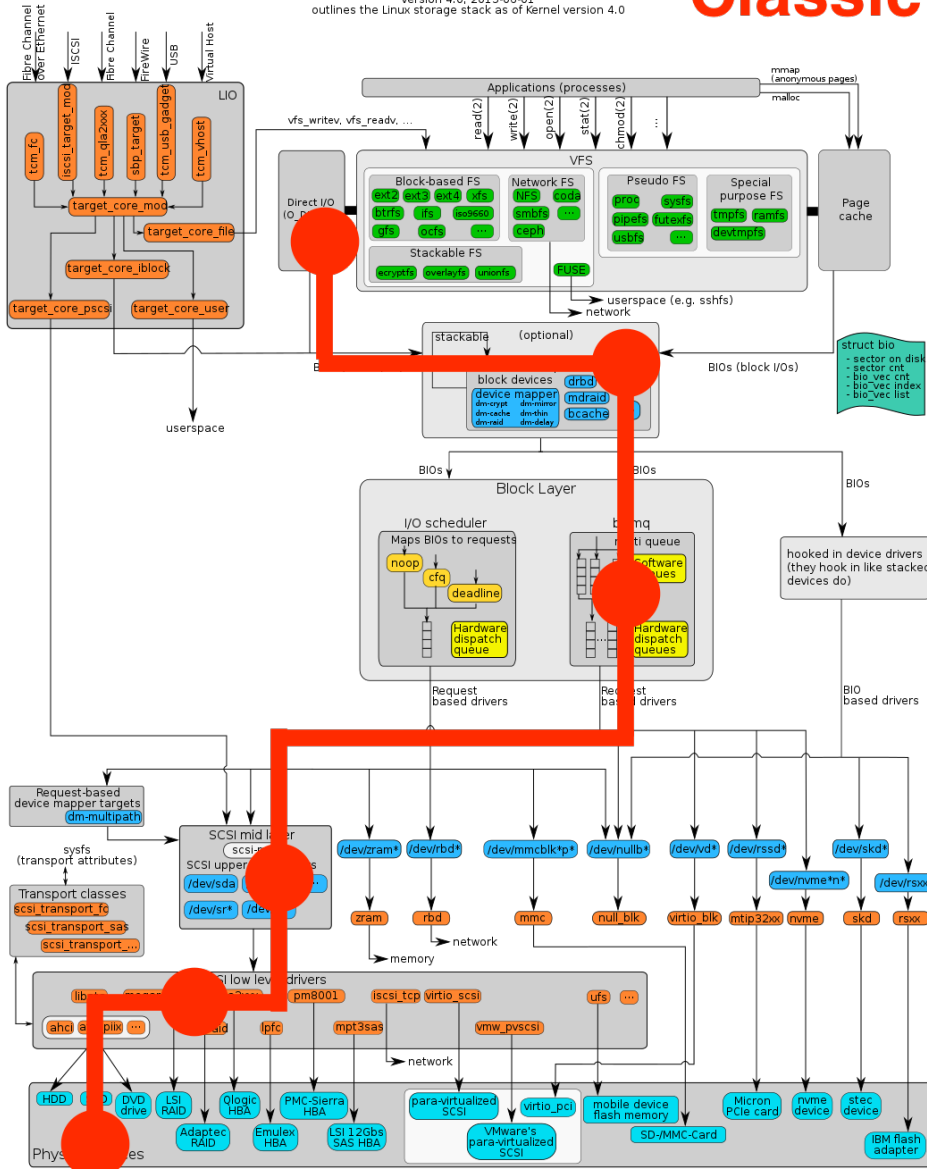
Martin Klier // twitter.com/MartinKlierDBA
www.performing-databases.com

NVMe SSD

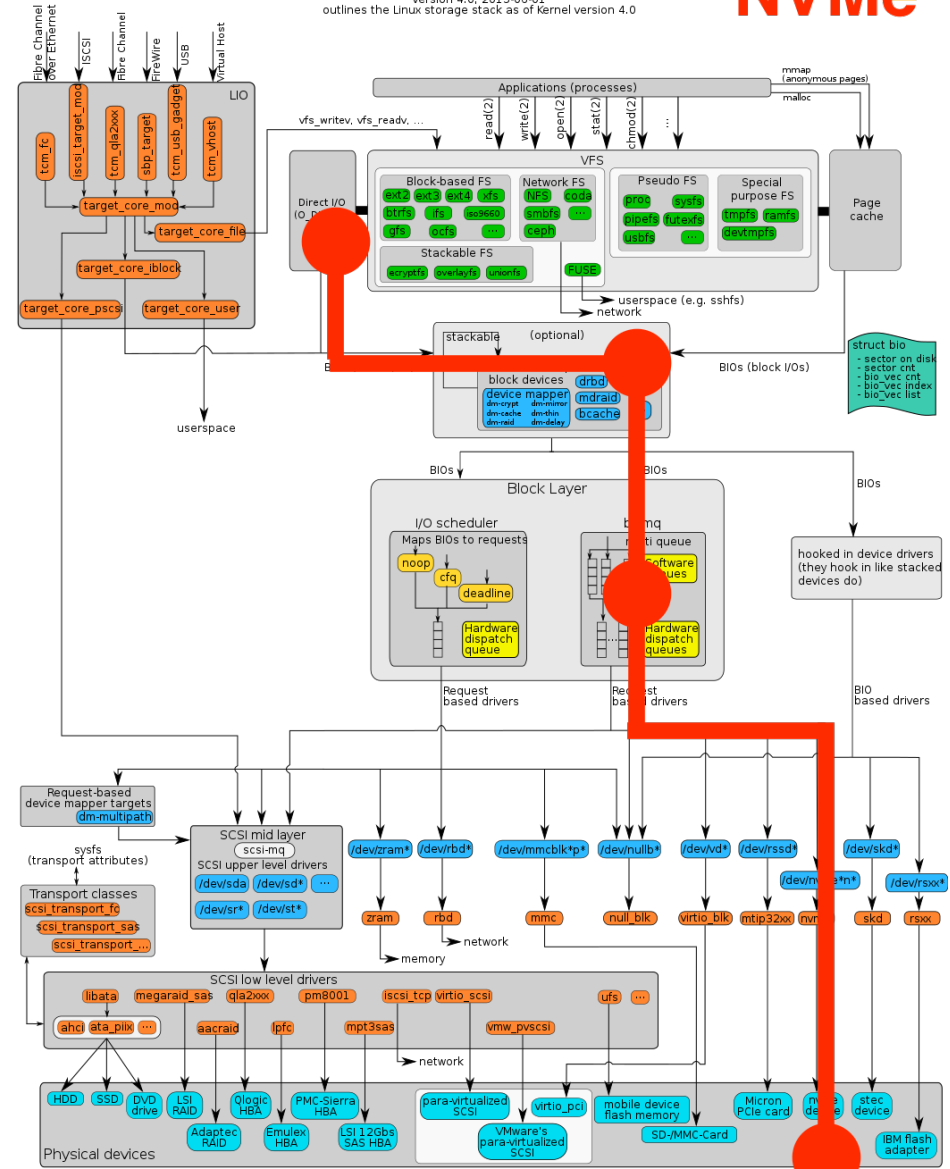


NVMe SSD

The Linux Storage Stack Diagram **Classic**
version 4.0, 2015-06-01
outlines the Linux storage stack as of Kernel version 4.0



The Linux Storage Stack Diagram **NVMe**
version 4.0, 2015-06-01
outlines the Linux storage stack as of Kernel version 4.0



SSD

- Know your IO load profile (AWR, nmon)
- Use enterprise-level devices w/ Single Level Cell (SLC)
- SSDs require different lifecycle handling
=> Controller
- The Path to disk matters!

Concurrency

means collisions and serialisation

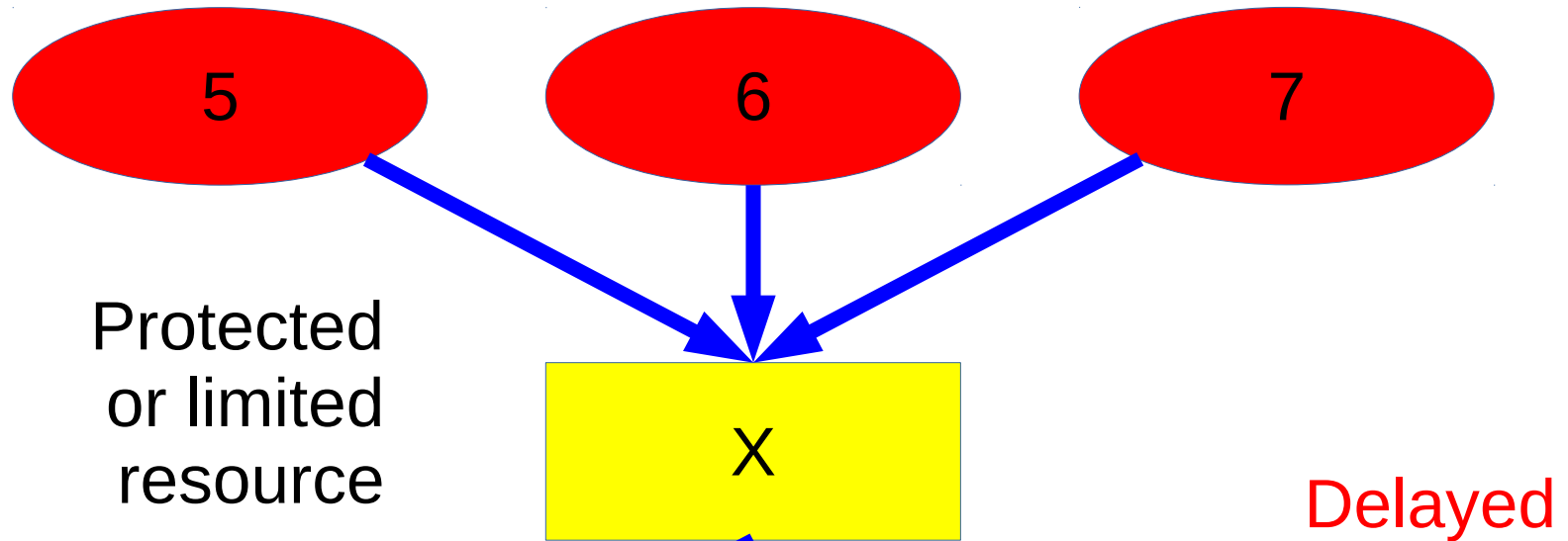


Concurrency

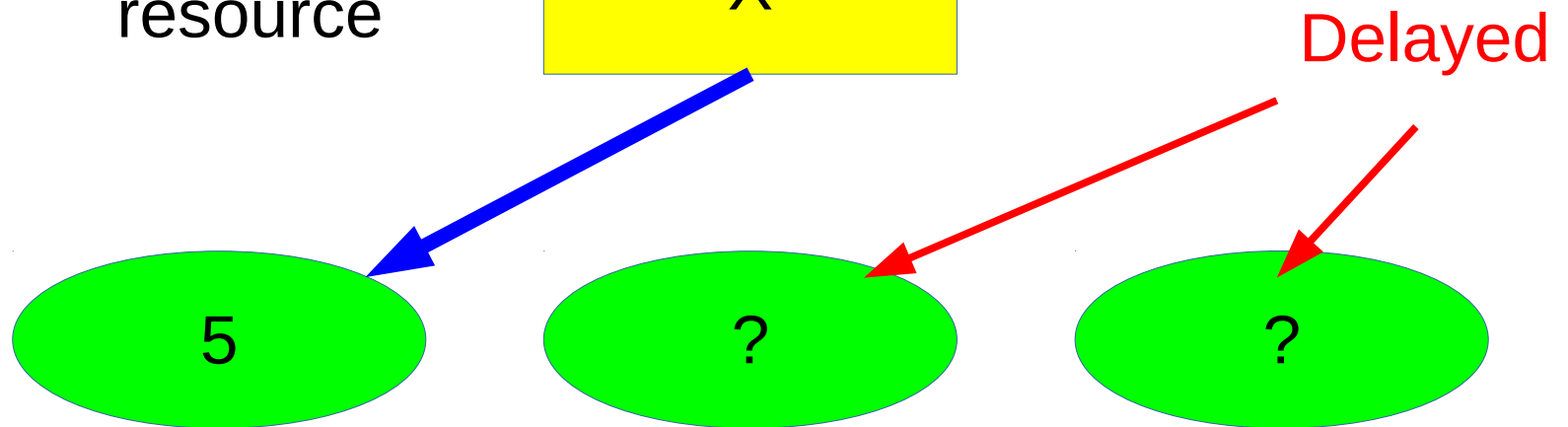
- Data Access (Row Lock, Block Header)
- Shared memory organization (Buffer / Library Cache etc.)
- CPU queueing
- Disk / Network IO

Concurrency

State A



State B



Row Lock

```
update EMPLOYEES a  
  set a.salary = 1200  
  where a.JOB_ID = 20;
```

```
update EMPLOYEES a  
  set a.salary = 1500  
  where a.EMP_ID = 5;
```

Lock

EMPLOYEES

EMP_ID	JOB_ID	SALARY
1	100	2000
3	13	1500
99	22	750
2	12	1200
4	144	1500
5	20	1100 1200 ?
6	233	3000
10	100	2100
22	22	800
19	12	1250
11	144	1330
32	12	1100
8	144	1440
9	20	1150 1200
11	233	2990

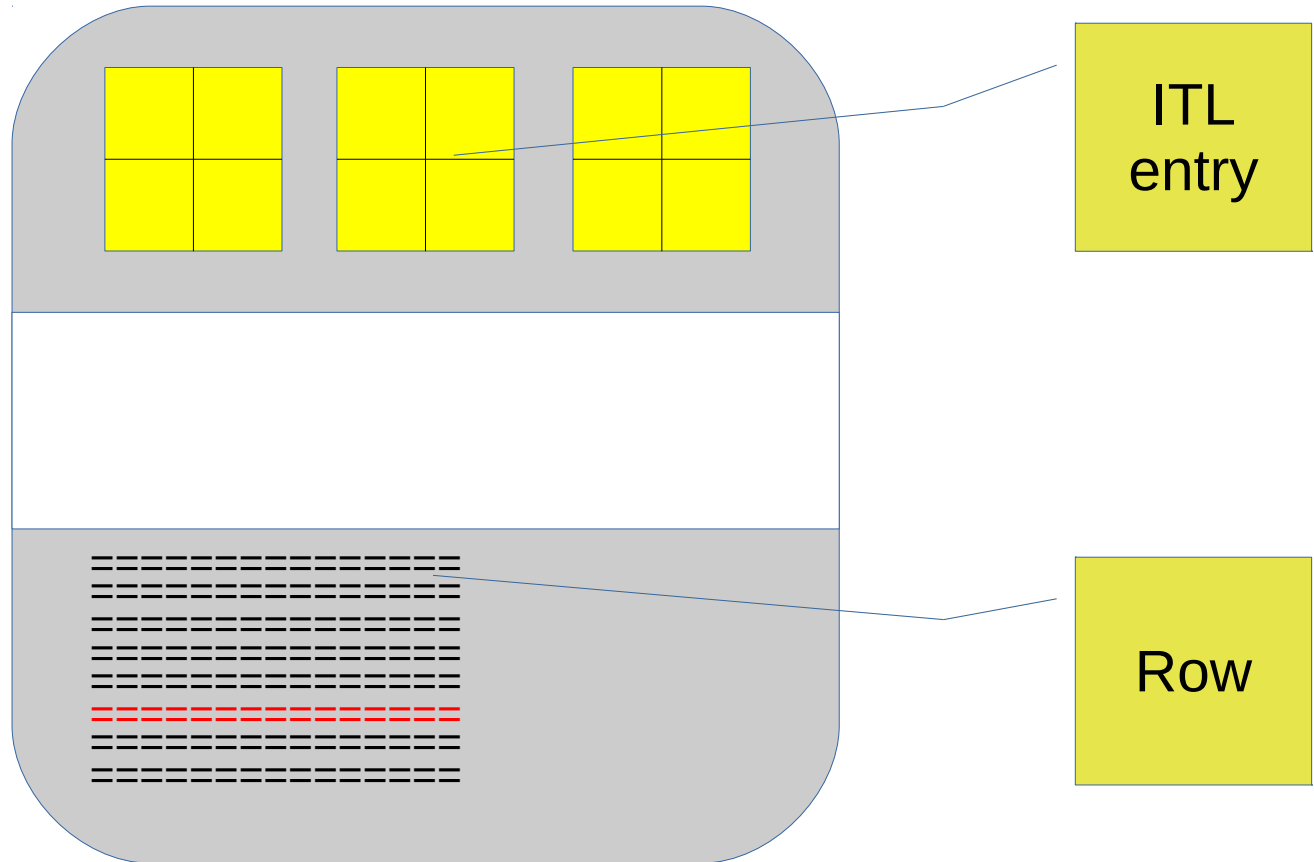


Buffer

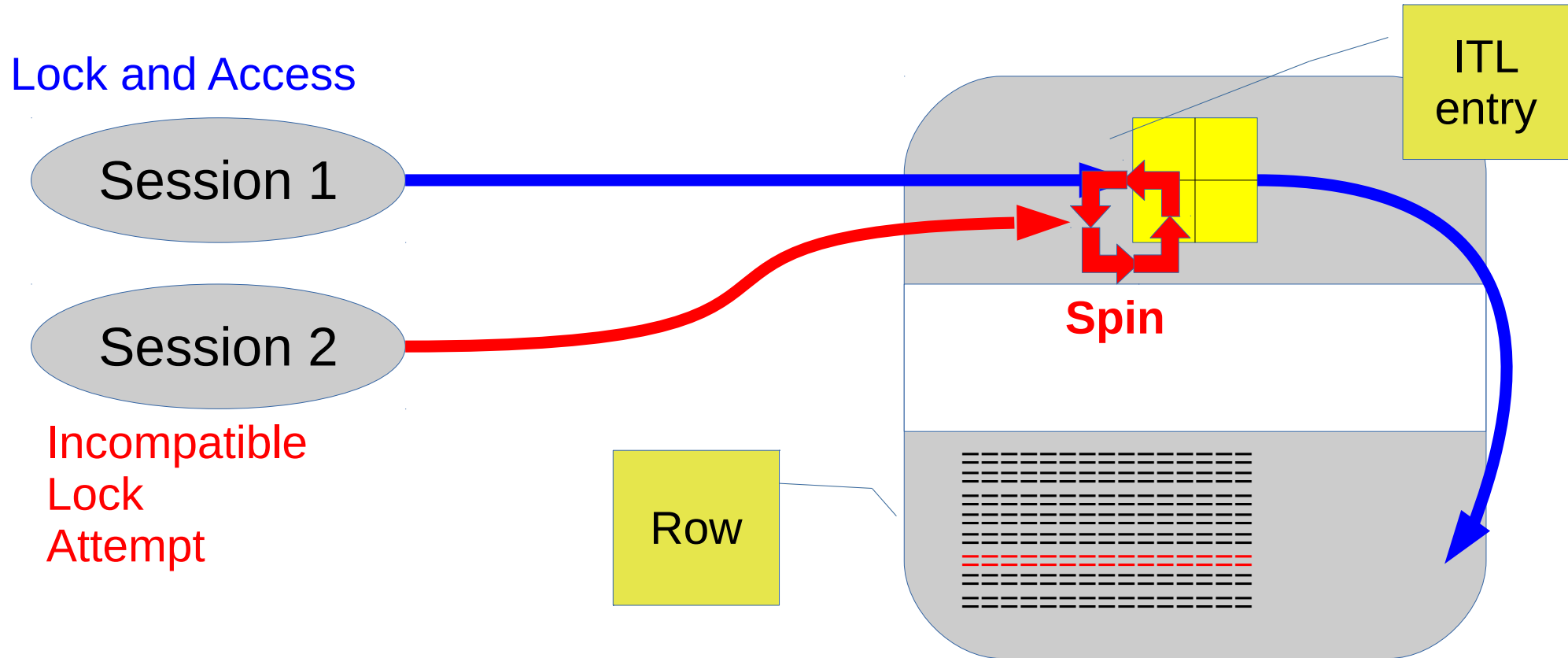
Header

Free Space

Row Space



Row Lock



enq: TX - row lock contention



Concurrency

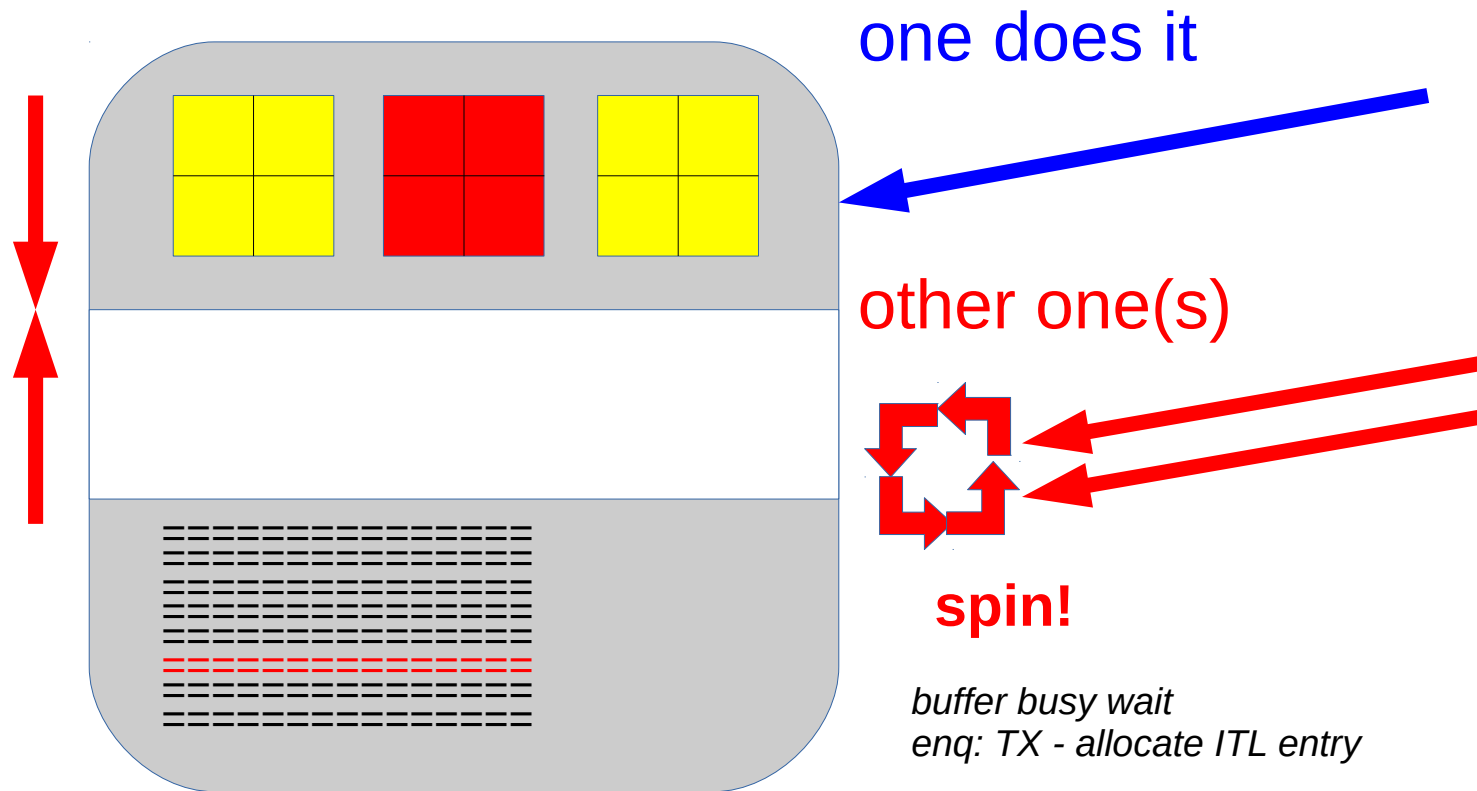
Spinning

- Active checking of a value in memory
- “Wasting” CPU for non-productive work
- Oracle Spin Count limits and Wait Events are a generosity
- to limit, see and measure the impact

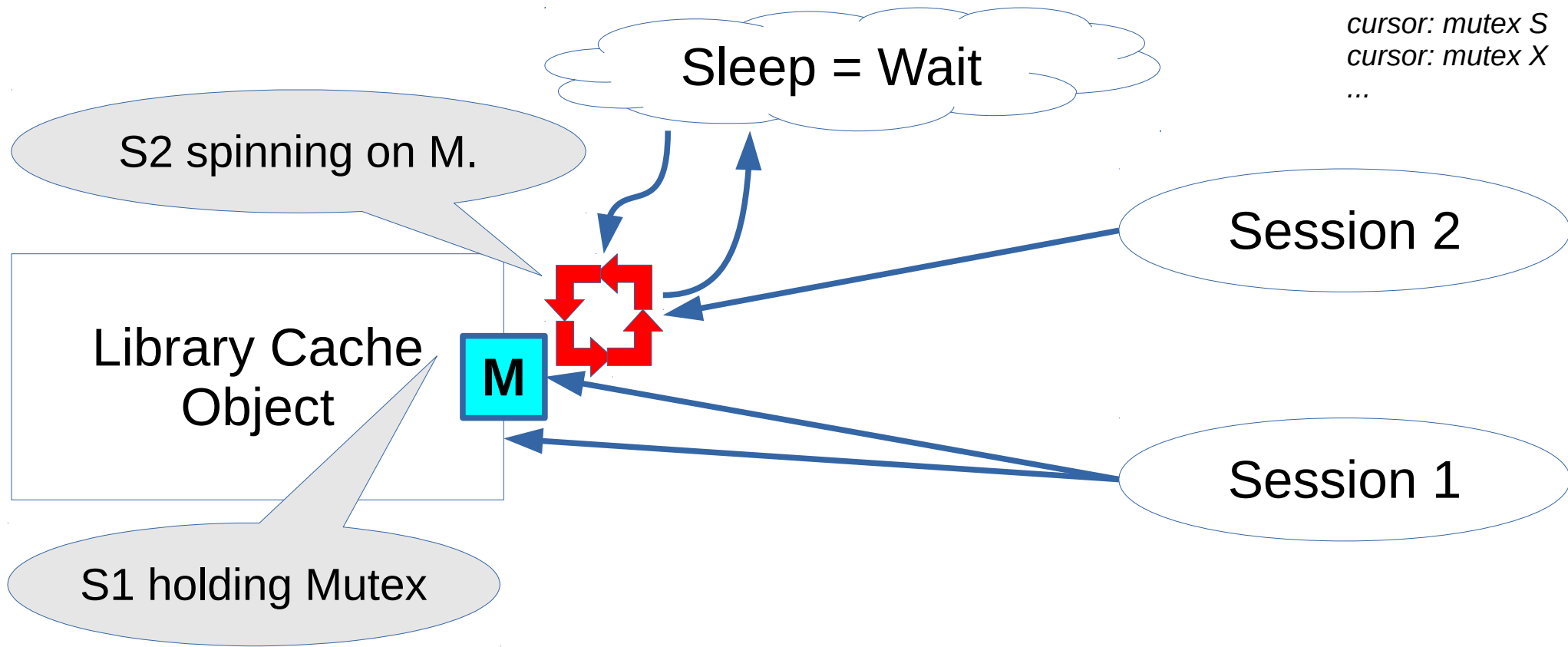
ITL Stress

Resizing

- Limited Space
- Concurrent Buffer modif.

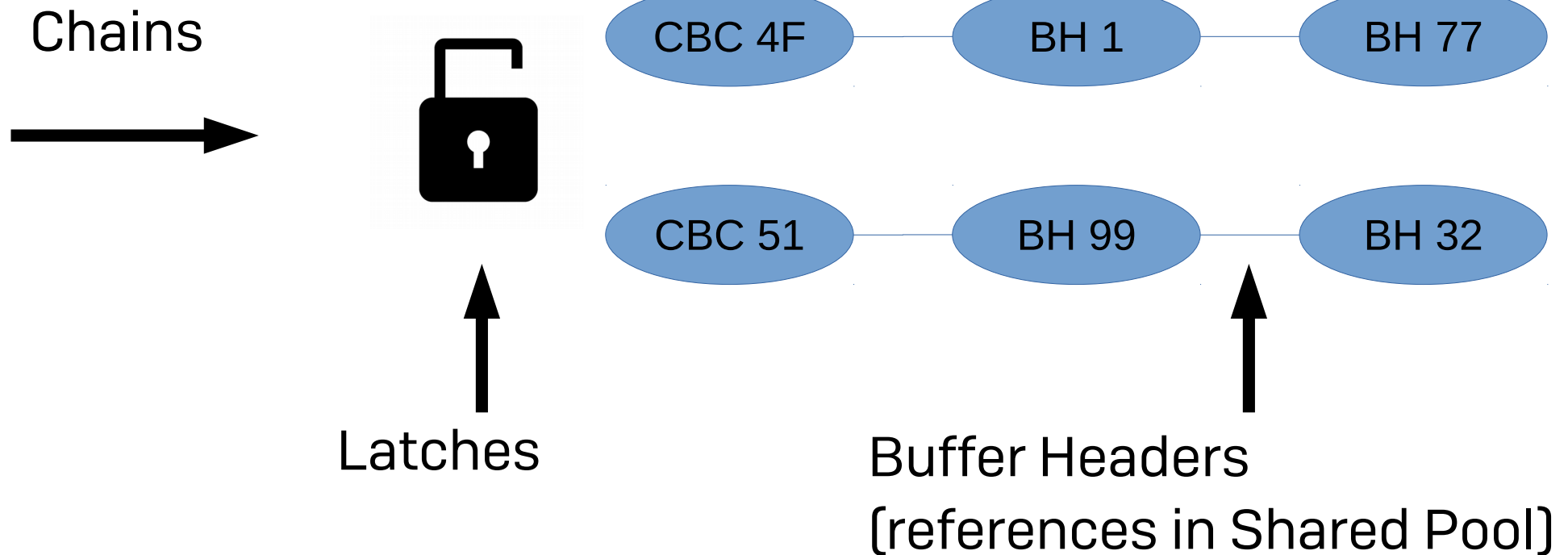


Mutex



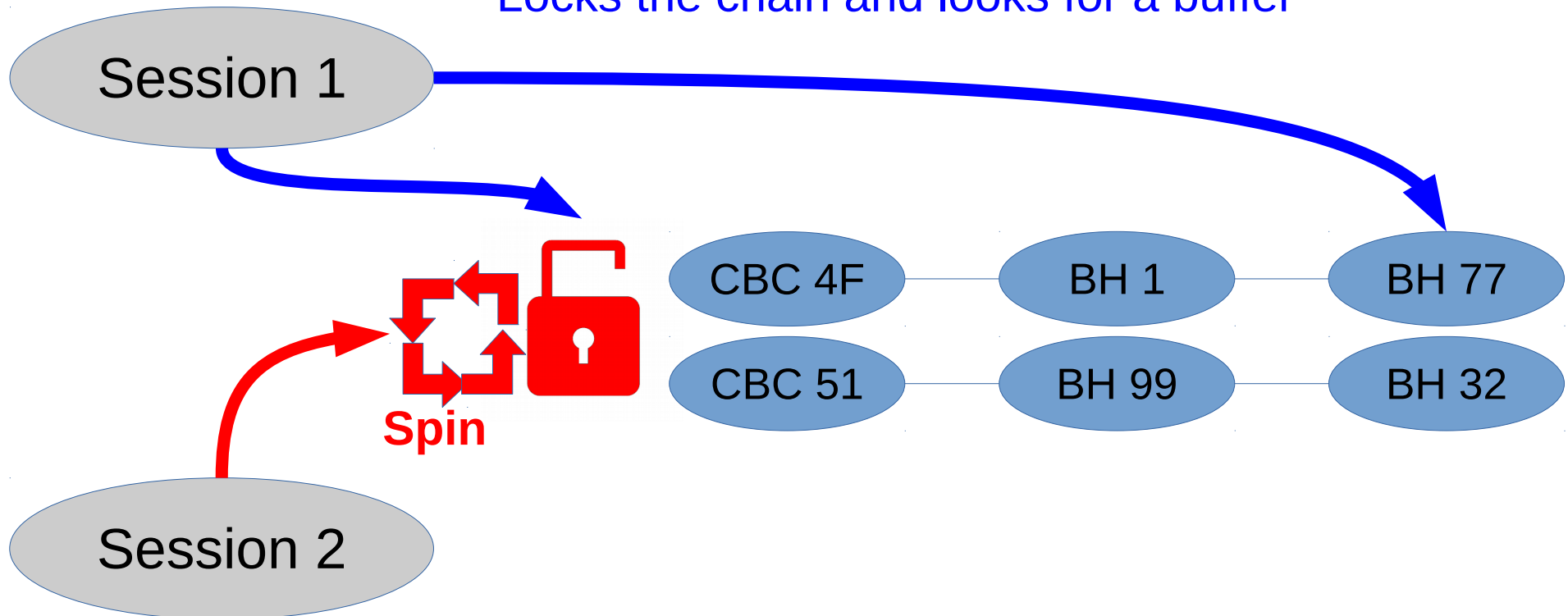
CBC

Cache Buffer Chains: Is this block in the BC?



Concurrency

Locks the chain and looks for a buffer



latch: cache buffer chain

Same or diff. Buffer (Chain), same latch :(

Concurrency

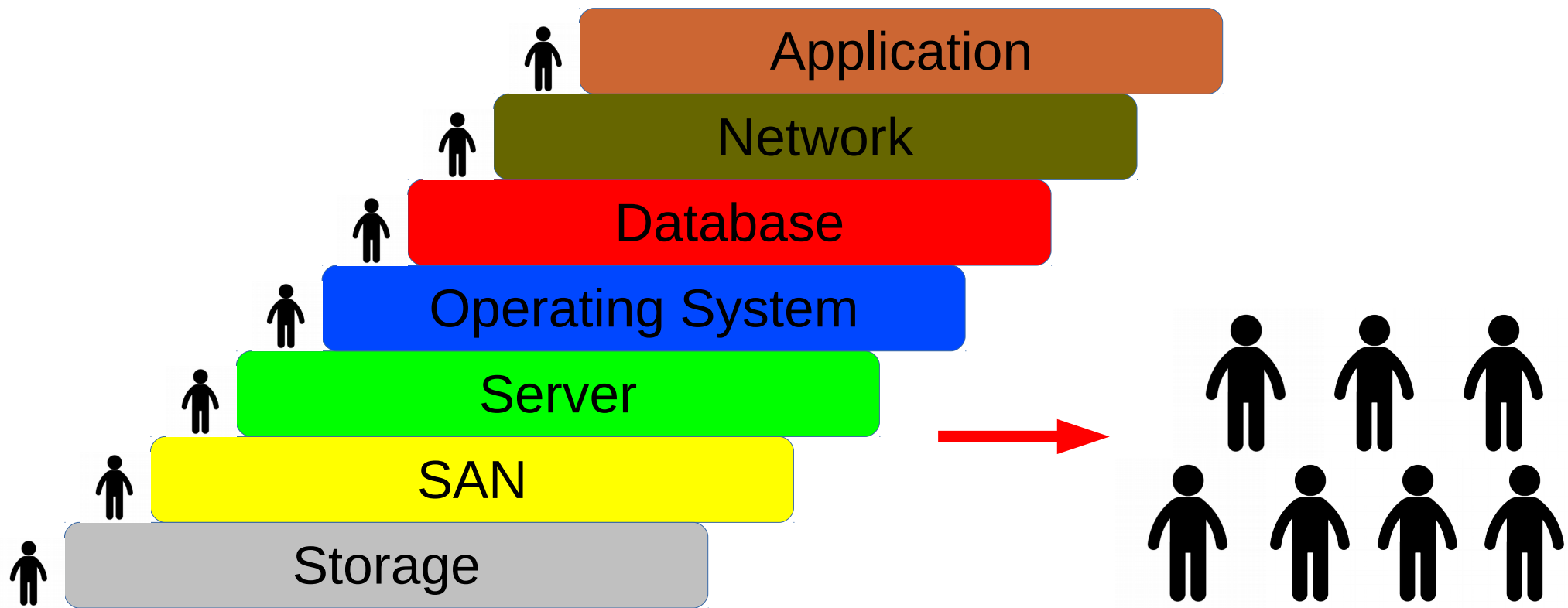
Suggestions

- Check workload (think: SQL efficiency)
=> Reduce logical reads/writes
- Be ready for decent diagnosis (think in Wait Events)

Collaborate!

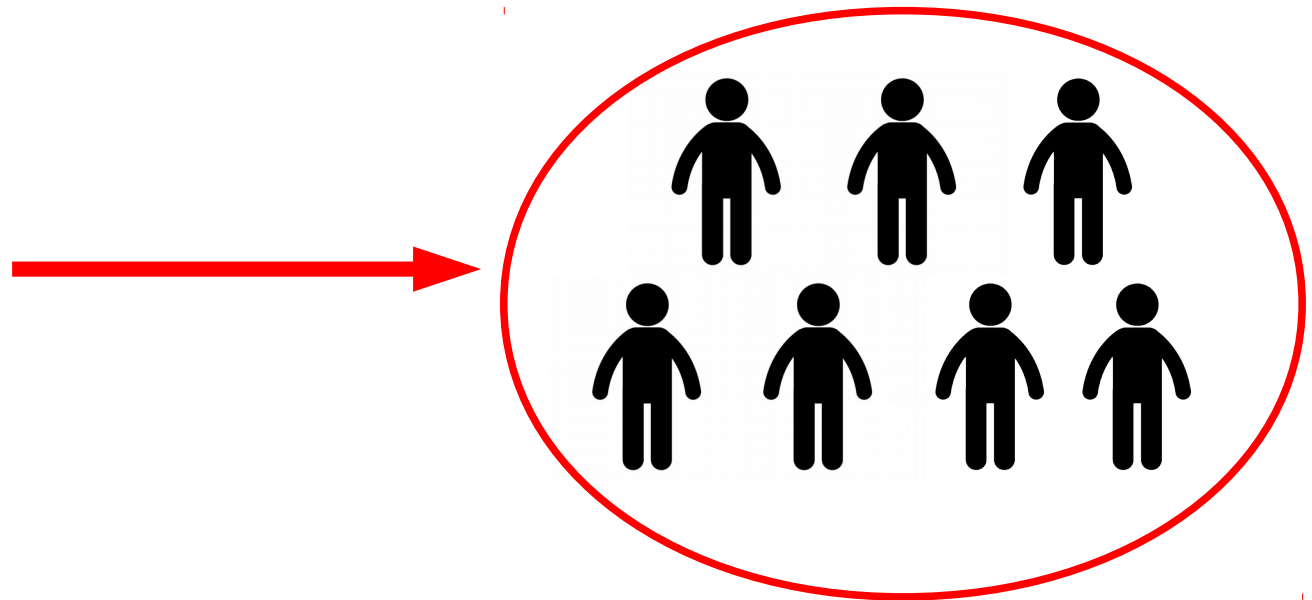
never forget the human factor

Layers



People

Engineers
to work
together



Speaker

- Martin Klier
- Solution Architect and
- Database Expert
- My focus:
 - Performance + Tuning
 - Highly available Systems
 - Cluster and Replication
- Linux since 1997
- Oracle Database since 2003



ORACLE®
ACE Director



performing
databases

Martin Klier // twitter.com/MartinKlierDBA
www.performing-databases.com

Performing Databases

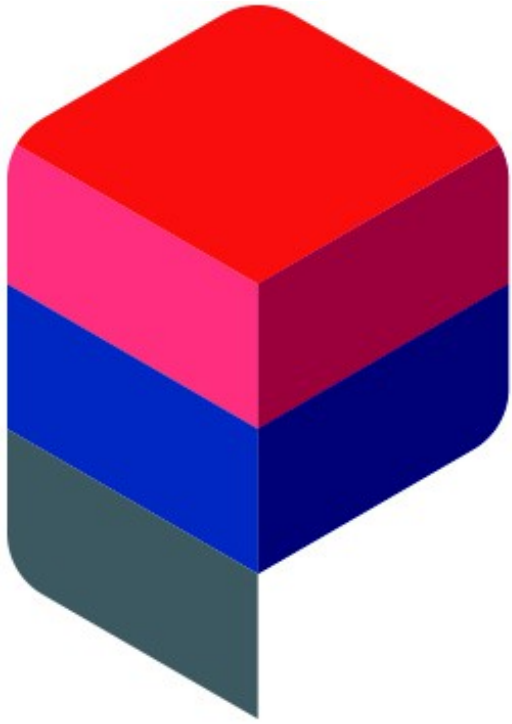
- Experts for Database Technology
 - Concepts and Project Competence
 - Architektur- und System planning
 - Licensing
 - Implementation and Troubleshooting
- Contact
 - Performing Databases GmbH
Wiesauer Straße 27
95666 Mitterteich // Germany
 - Web: <http://www.performing-databases.com>
 - Twitter: @PerformingDB





Download Presentation

<http://www.performing-databases.com>



**performing
databases**