COLLABORATE14

IOUG
independent oracle users group

# YOUR machine and MY database - a performing relationship!? (#141)

Martin Klier
Senior / Lead DBA
Klug GmbH integrierte Systeme


Las Vegas, April 10th, 2014


# Agenda


- Introduction
- NUMA + Huge Pages
- Disk IO
- Concurrency
- Engineers to work together


# Speaker




- Martin Klier (twitter: **@MartinKlierDBA**)
- Lead DBA for Oracle at Klug-IS
- Focus on Performance, Tuning and High Availability
- Linux since 1997, Oracle since 2003

ORACLE®

- Email: martin.klier@klug-is.de
- Weblog: http://www.usn-it.de

- Klug GmbH integrierte Systeme (http://www.klug-is.de)
  92552 Teunz, GERMANY

ORACLE Gold Partner

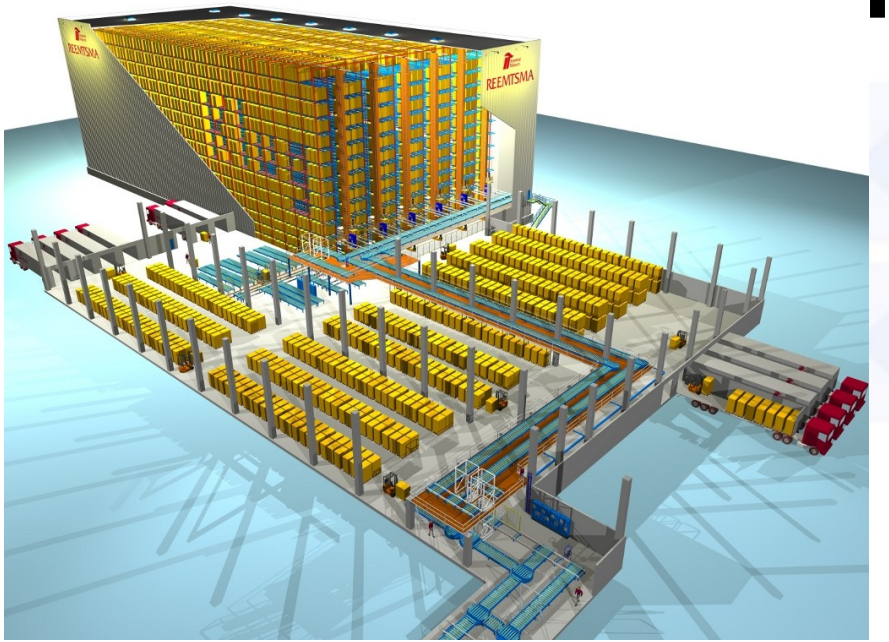
- Specialist leading in complex intralogistical solutions
- Planning and design of automated intralogistics systems, focus on software and system control / PLC
- >300 successful major projects in Europe, America, Asia

COLLABORATE14

Klug integrierte Systeme

IOUG independent oracle users group

# iWACS®




COLLABORATE14

DOAG

Deutsche ORACLE-Anwendergruppe e.V.

IOUG

independent oracle users group

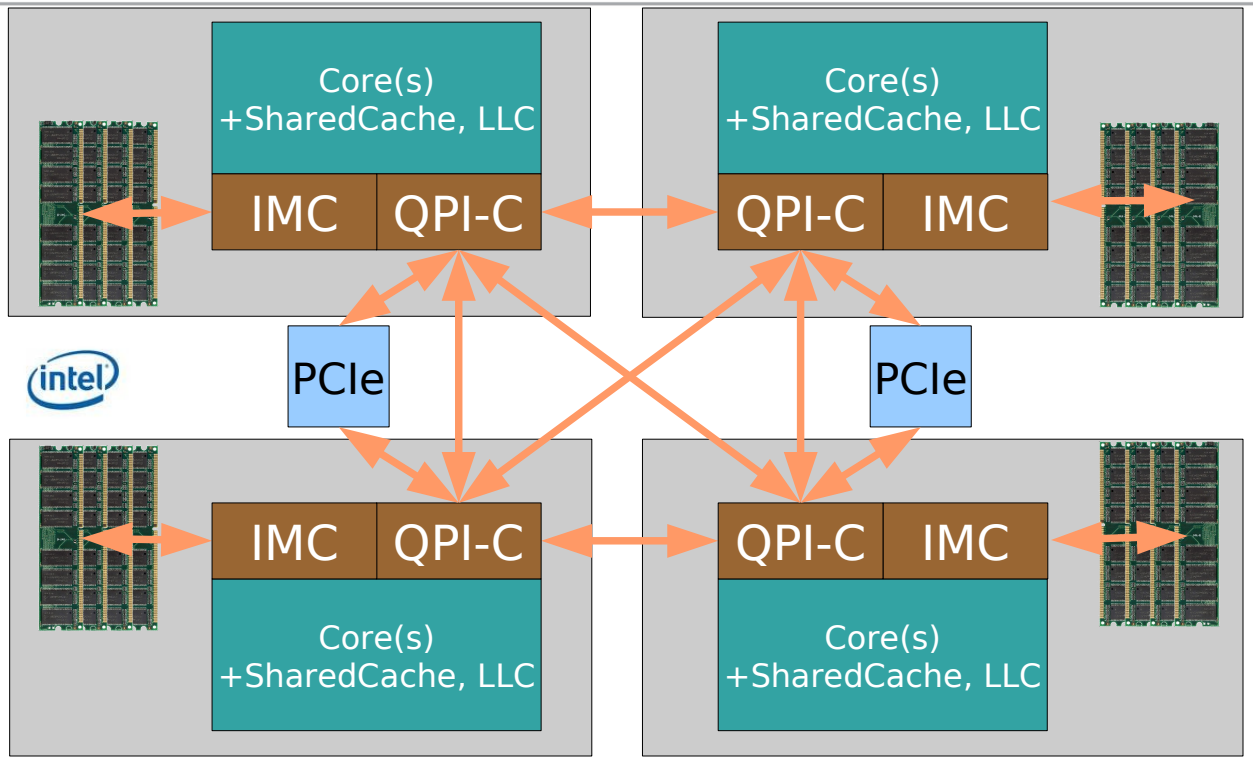
- DOAG - Deutsche Oracle Anwendergruppe (German Oracle User's Group)
- **Biggest** Oracle Technology **Conference in Europe** (2,000+ attendees)
- Save the date:
  **Nuremberg, November 18th - 21st 2014**

# Server / CPU

# NUMA



**Non Unified Memory Access**



# NUMA



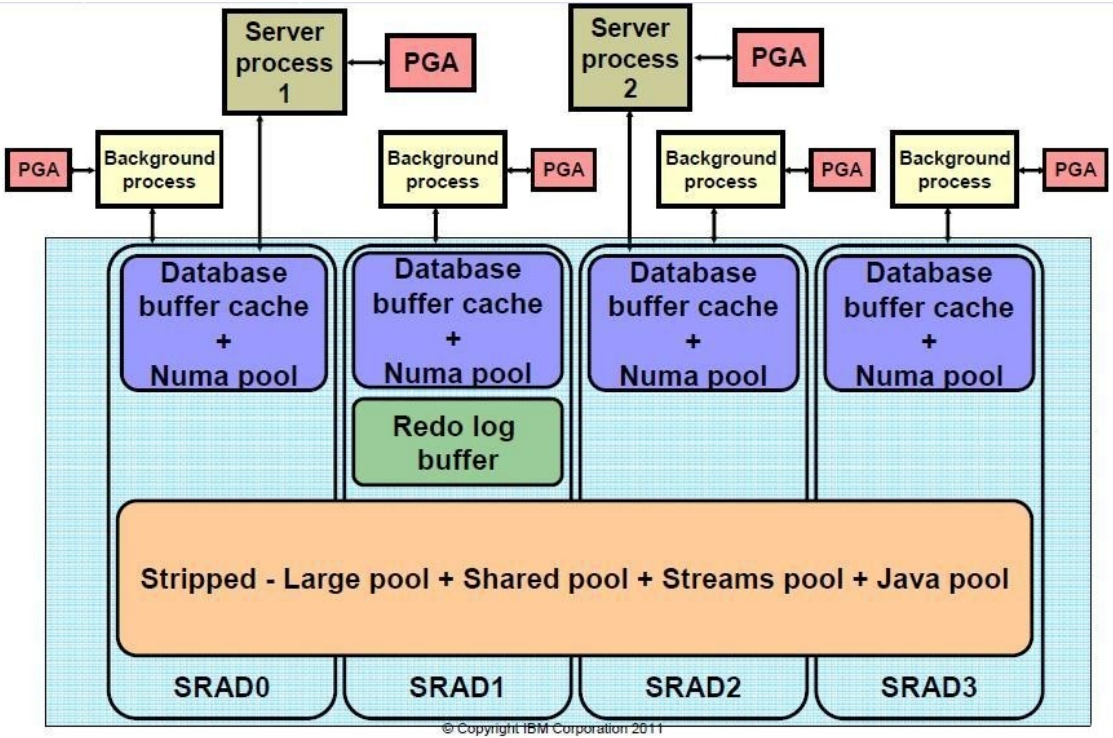
*_enable_NUMA_support = TRUE*

*MOS Doc ID 864633.1*

- Multiple Buffer Caches
- Striped pools

  => cross context :((

  => pool access :(



# NUMA



```
[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

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

# NUMA



```
[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
```

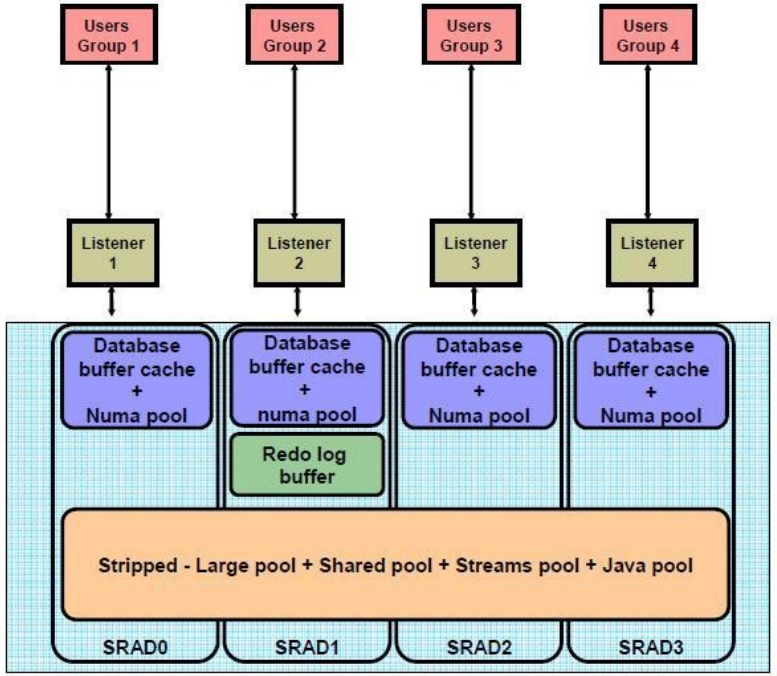
One buffer cache for each node

13GB+13GB=26 GB

# NUMA



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



# NUMA



## Non-NUMA

| node1 per second | node0 per second | |
|---|---|---|
| 6364,2 | 16011,4 | numa_hit |
| 0,2 | 0,7 | numa_miss |
| 0,7 | 0,2 | numa_foreign |
| 0 | 0 | interleave_hit |
| 6364,2 | 16011,3 | local_node |
| 0,2 | 0,7 | other_node |

## with NUMA

| Node1 diff p s | Node0 diff p s | |
|---|---|---|
| 3538,6 | 10179 | numa_hit |
| 0 | 0 | numa_miss |
| 0 | 0 | numa_foreign |
| 0 | 0 | interleave_hit |
| 3538,6 | 10178,9 | local_node |
| 0 | 0,1 | other_node |

With my workload and only one listener:

**Saved <1 page alloc miss per second**

?

# NUMA



```
[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 |

**Fits** into RAM of one node.

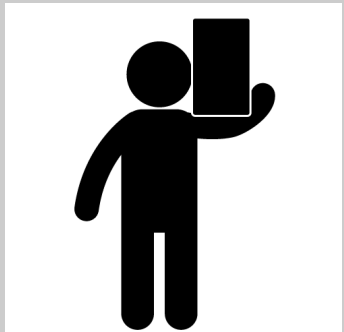
**OS** NUMA optimization at work.

26 GB

# NUMA



## Relevance-and-care chart

| System Admin | DBA | Developer |
| --- | --- | --- |
|  |  |  |

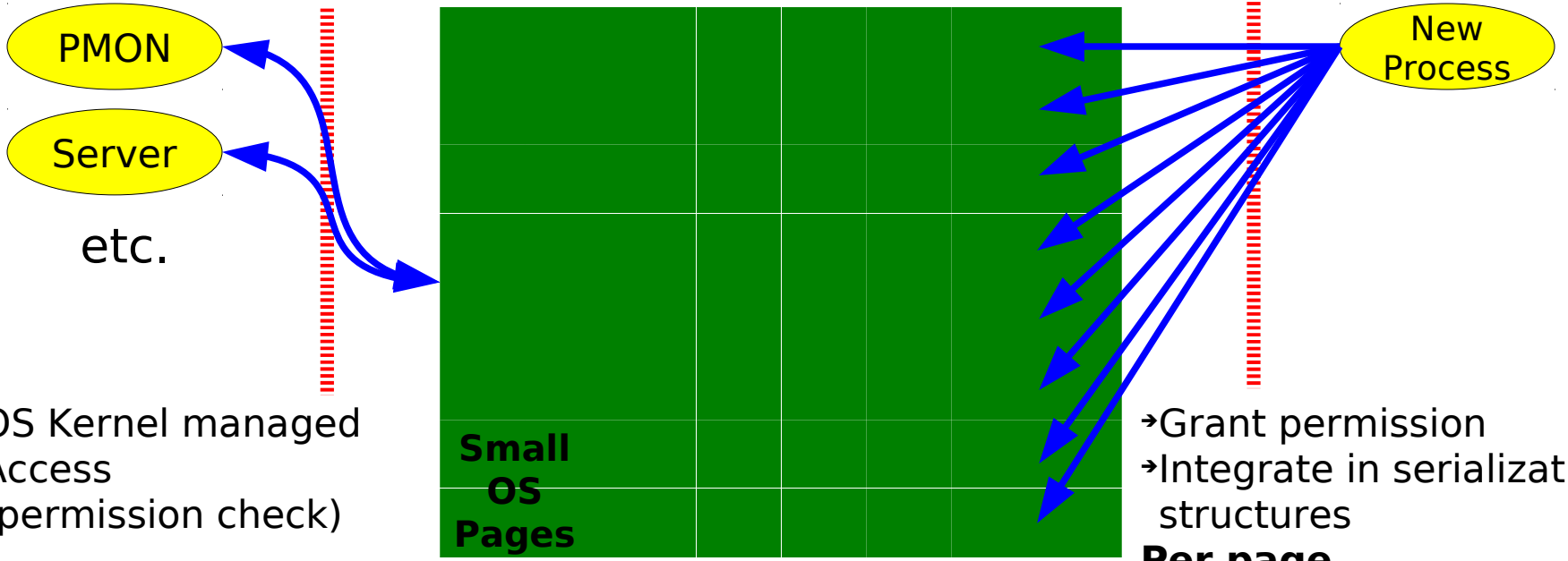
## Suggestions NUMA

- 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)

# Server / RAM

# RAM






OS Kernel managed Access (permission check)

→Grant permission
→Integrate in serialization structures
**Per page**

# RAM

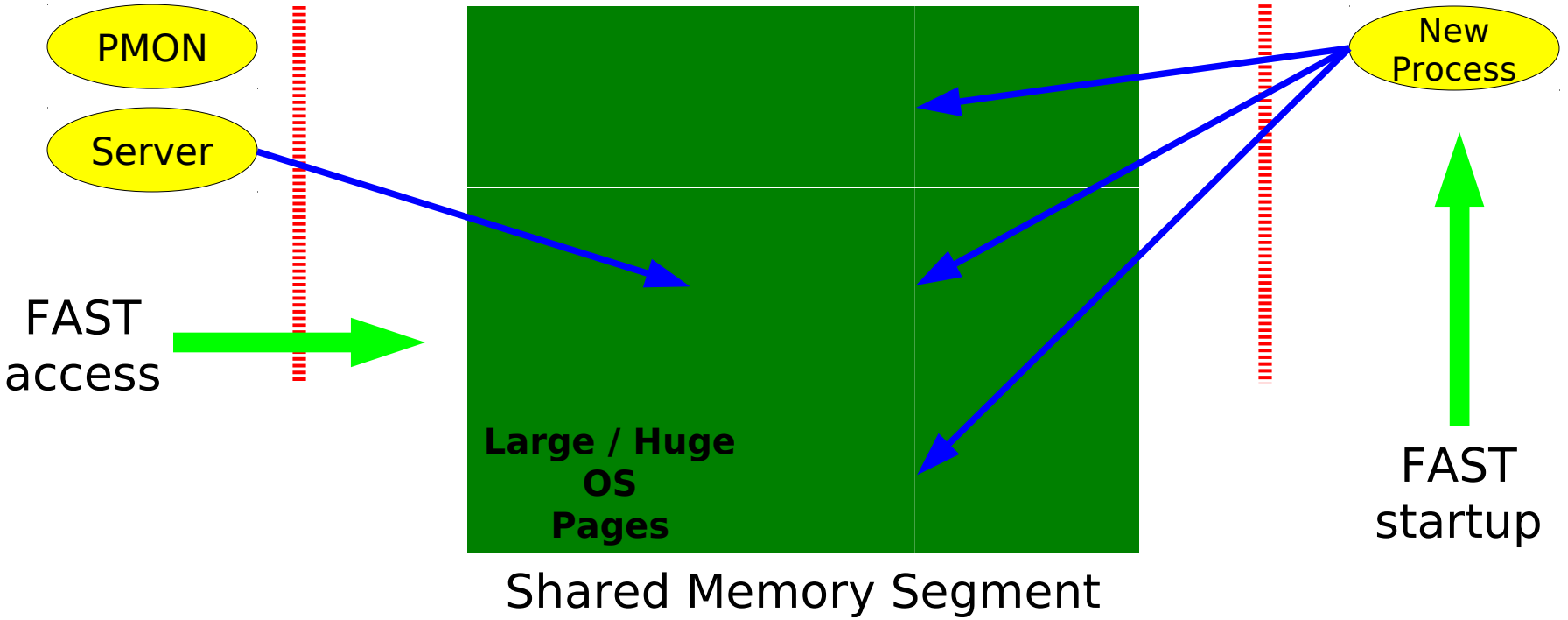


## Problems

- Memory Fragmentation
- Wasting CPU with page alloc
- OS_THREAD_STARTUP waits

# Huge Pages




PMON
Server
FAST access
Large / Huge OS Pages
New Process
FAST startup
Shared Memory Segment

# Huge Pages



```
[root@ora05 ~]# cat /proc/meminfo
MemTotal:       61956468 kB
MemFree:         2470660 kB
Buffers:          435624 kB
Cached:         14848192 kB
SwapCached:        1220 kB
VmallocTotal: 34359738367 kB
HugePages_Total:   17408
HugePages_Free:     3164
HugePages_Rsvd:       67
HugePages_Surp:        0
Hugepagesize:       2048 kB
DirectMap4k:        6144 kB
```

(17408-3164)*2048kB=28GB

## Alert Log

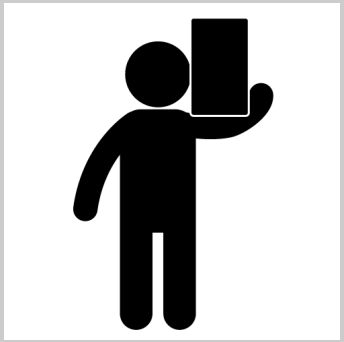
```
Starting ORACLE instance (normal)
****************** Large Pages Information *****************

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

## Relevance-and-care chart

| System Admin | DBA | Developer |
| --- | --- | --- |
|  |  |  |

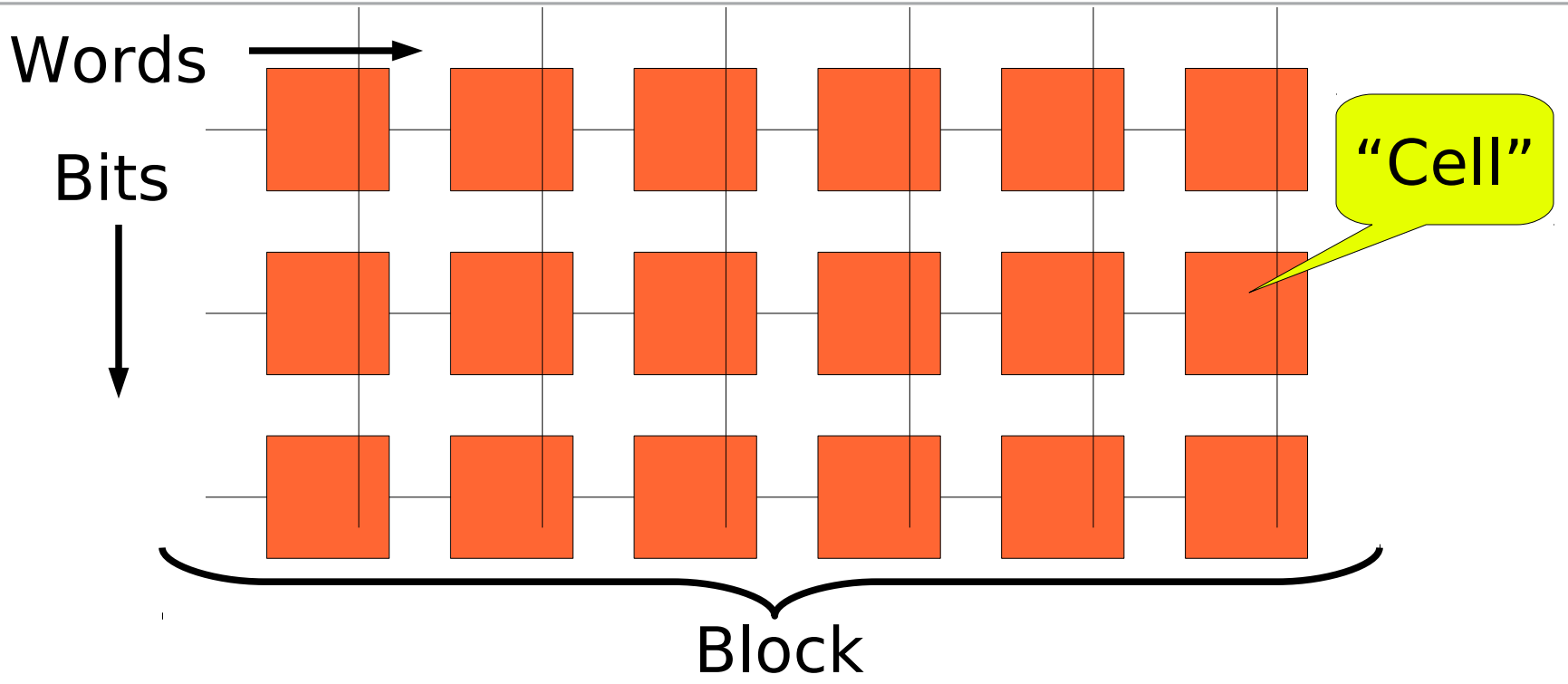
## Suggestions Large/Huge Pages

- Useful with SGA >=16GB
- Use largest available & sane page size
- Talk your sysadmin into **DO**ing **IT**
- Combine with PRE_PAGE_SGA=TRUE

# Storage / SSD

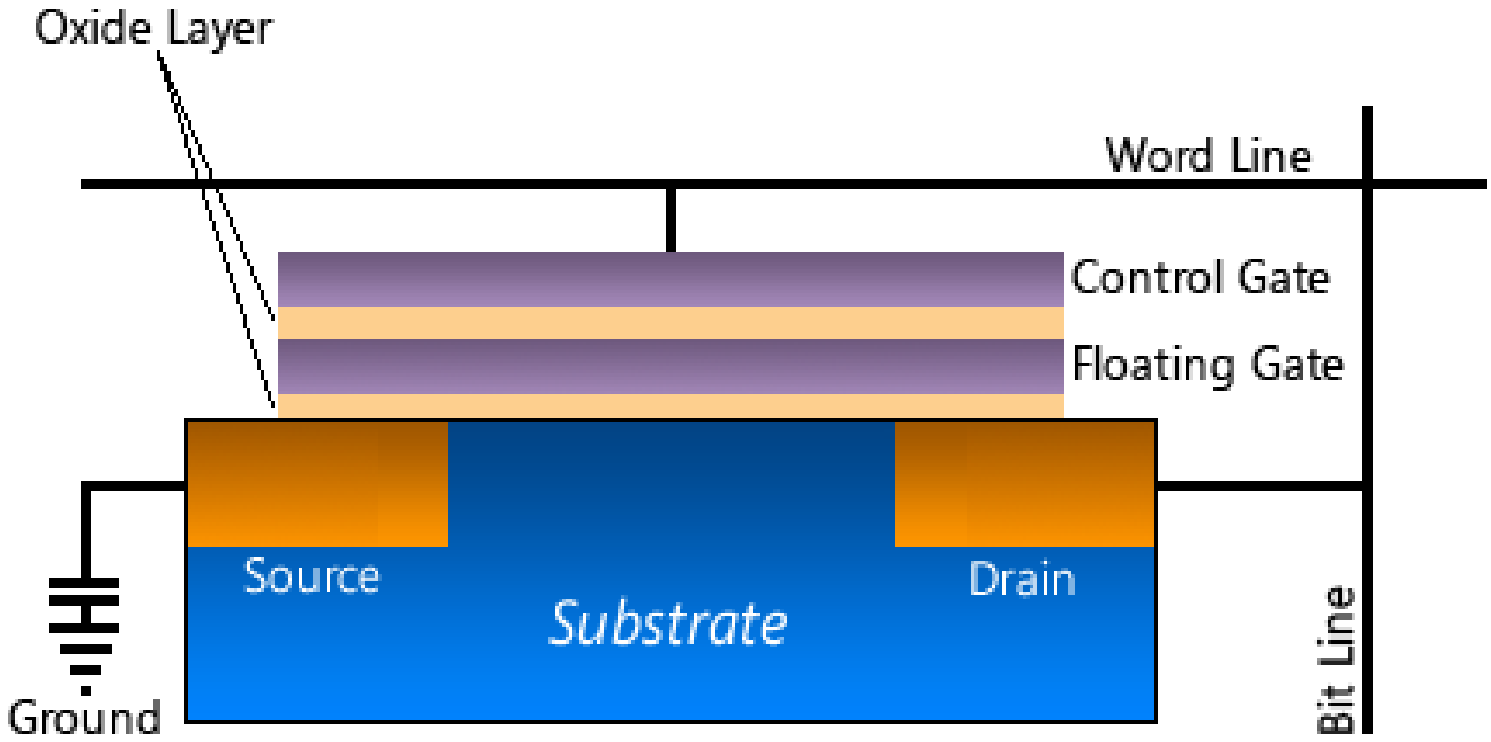
# SSD





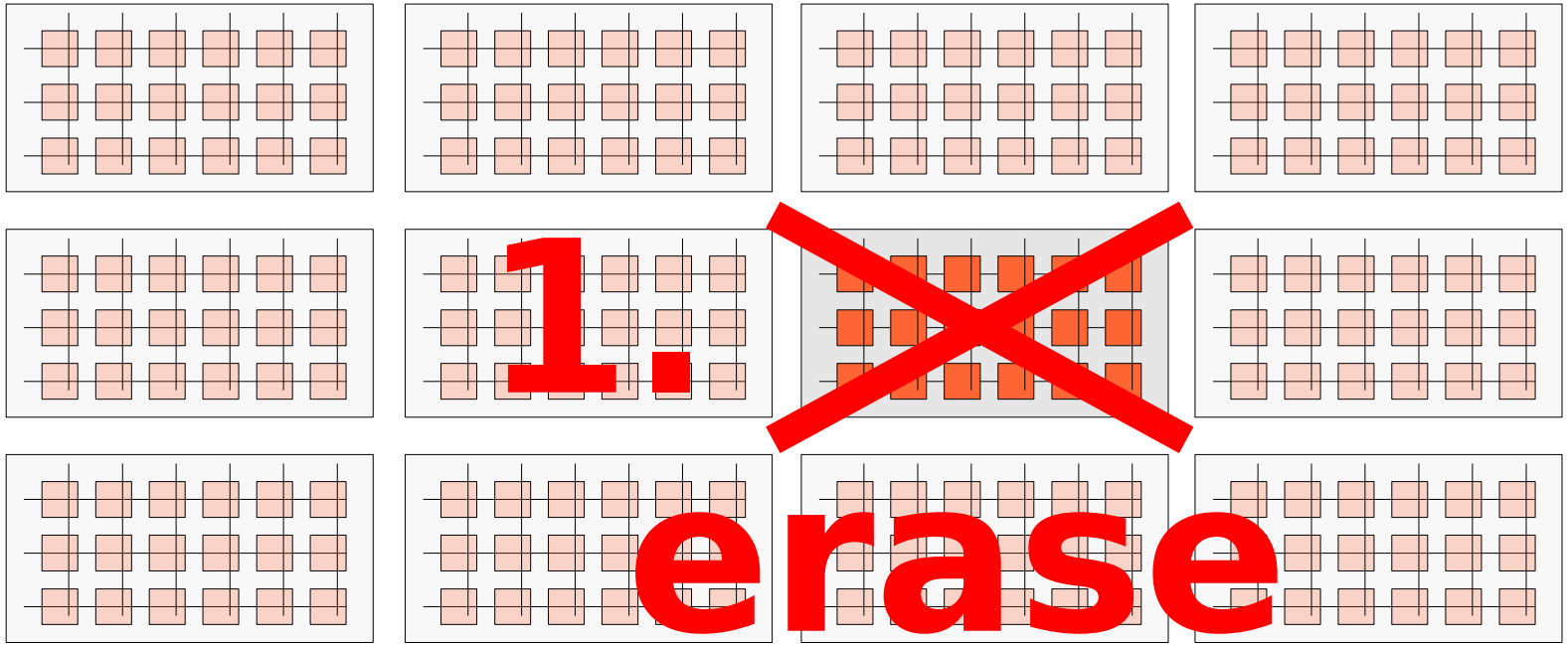
# SSD




Oxide Layer
Word Line
Control Gate
Floating Gate
Source
Substrate
Drain
Ground
Bit Line

# SSD

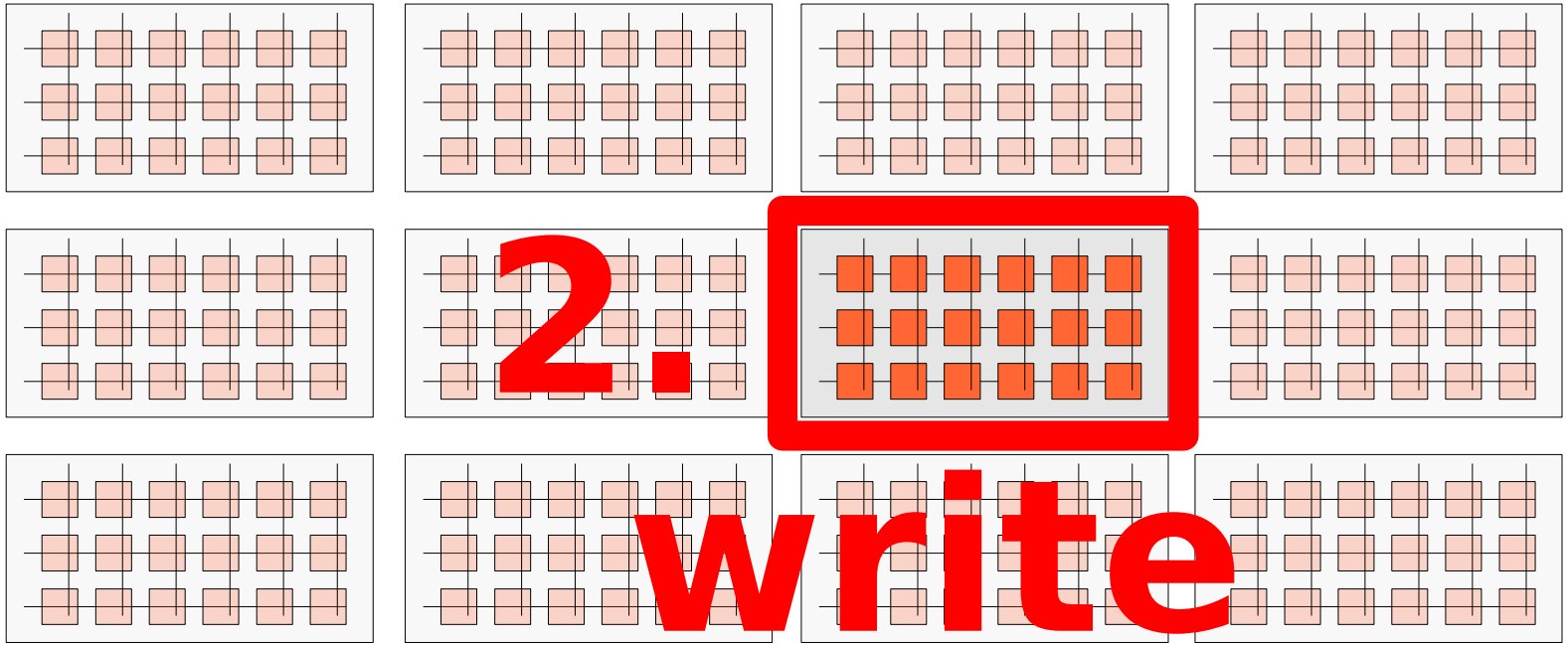





16kB – 512kB pro Block

# SSD

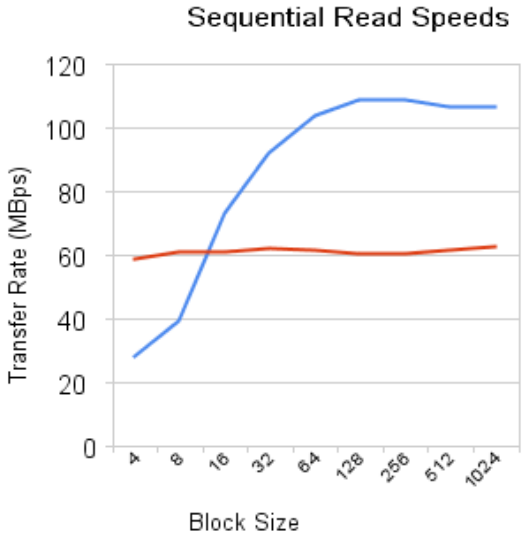




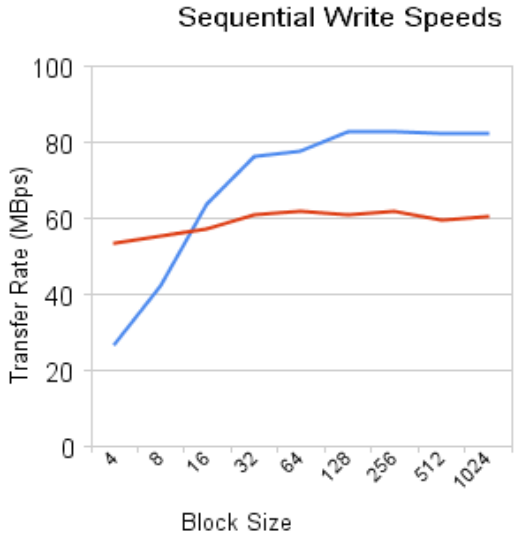

16kB – 512kB pro Block

# SSD






120:60
= *2




85:60
= *1.4

Types and Figures from 2009 - But the terrors are still intact. :)

# SSD



```
ORION VERSION 11.1.0.7.0

Commandline:
-run advanced -testname oltp-write -num_disks 1 -cache_size 8192 -size_small 8 -size_large 16 -type rand -simulate raid0 -write 80 -duration 30 -matrix basic

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

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
```

8k/16k blocks

80% write
20% read

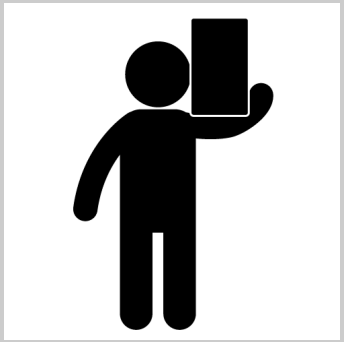
8171 IOPS
like 60 HDDs

*Samsung SSD 840 PRO*

# SSD



## Relevance-and-care chart

| System Admin | DBA | Developer |
| --- | --- | --- |
|  |  |  |

# SSD



## Suggestions

- Know your IO load profile (AWR, nmon)
- Use enterprise-level devices w/ Single Level Cell (SLC)
- SSDs require different lifecycle handling
  in doubt, consider an array of HDDs of same IO power

# Concurrency

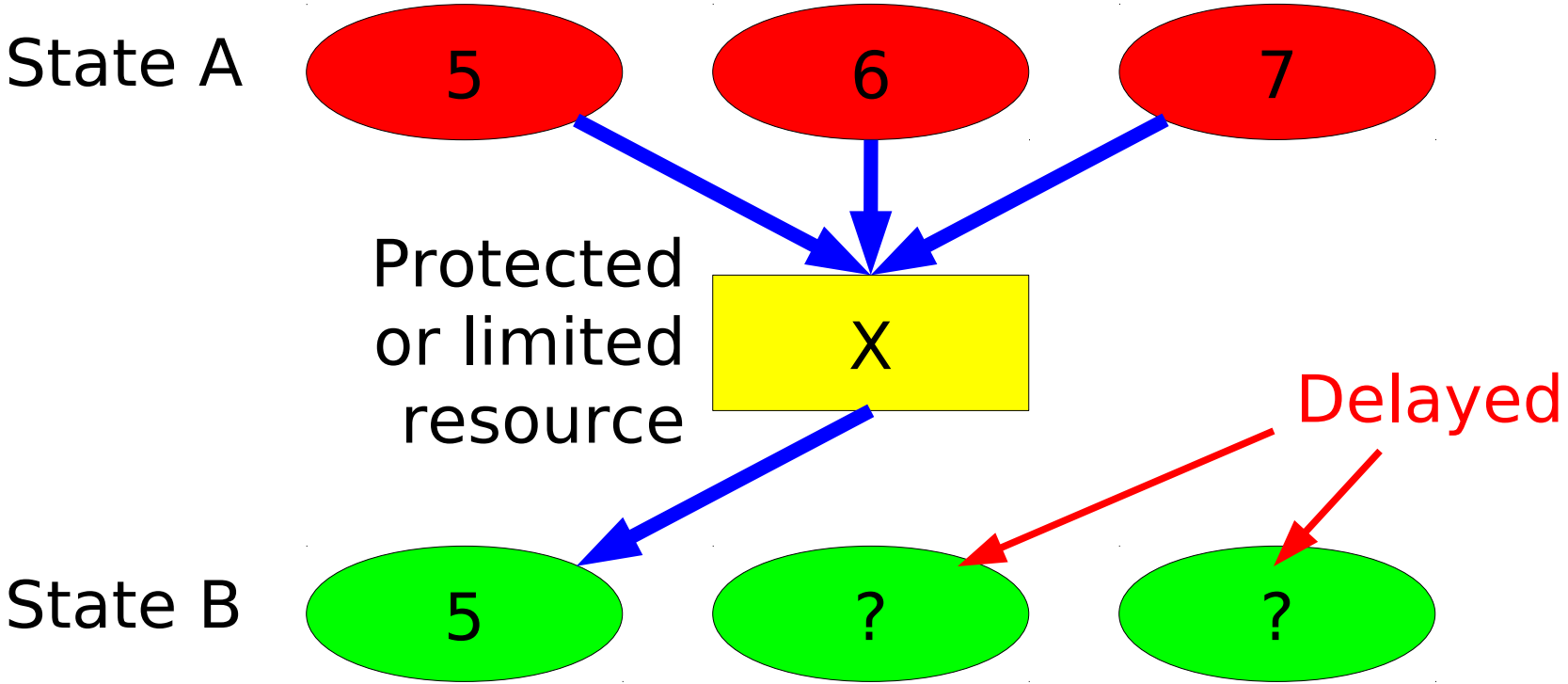
means collisions and serialization

## Occurrence

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

# Concurrency





# 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 |

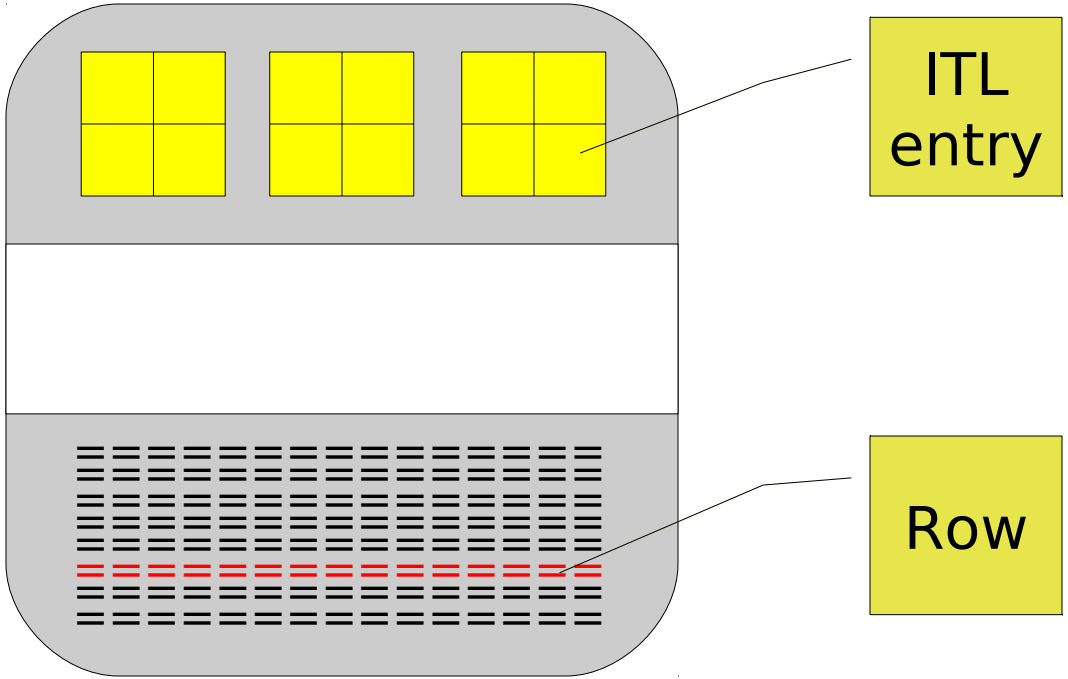
# Block/Buffer



Header

Free Space

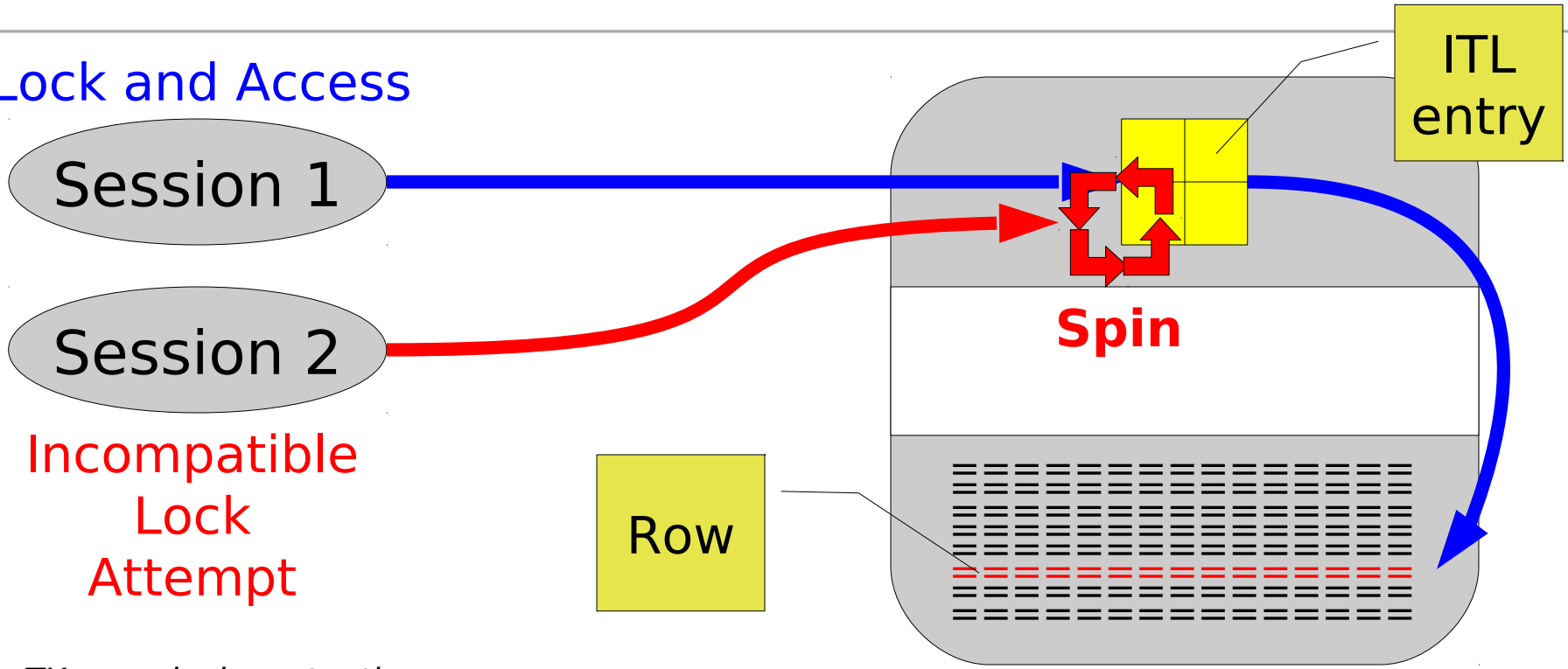
Row Space



# Row Lock



## Lock and Access




*enq: TX - row lock contention*

## Spinning means

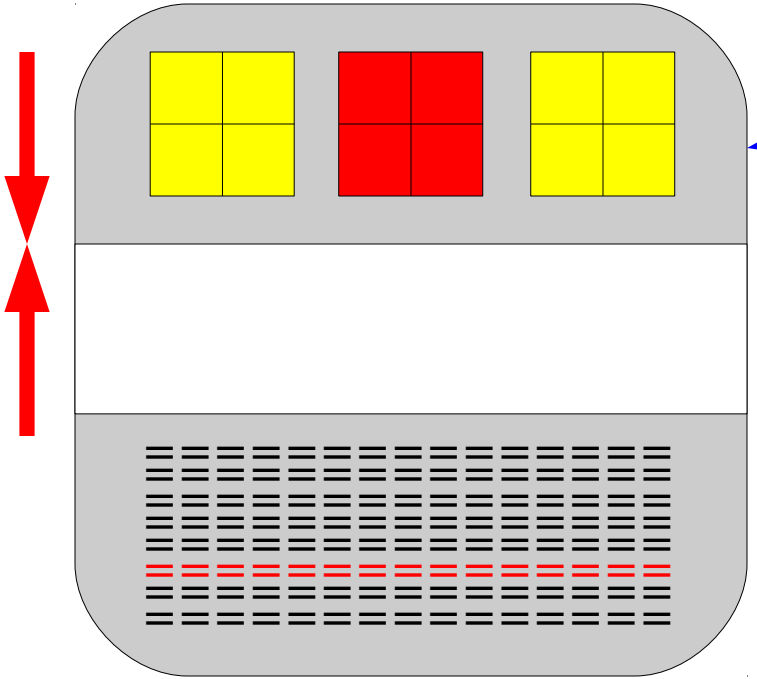
- Active checking of a value in memory
- “Wasting” CPU for non-productive work
- Oracle Spin Count limits and Wait Events are a generosity to limit, see and measure the impact

# ITL Stress



**Resizing**

- Limited Space
- Concurrent Buffer modif.



one does it

other one(s)

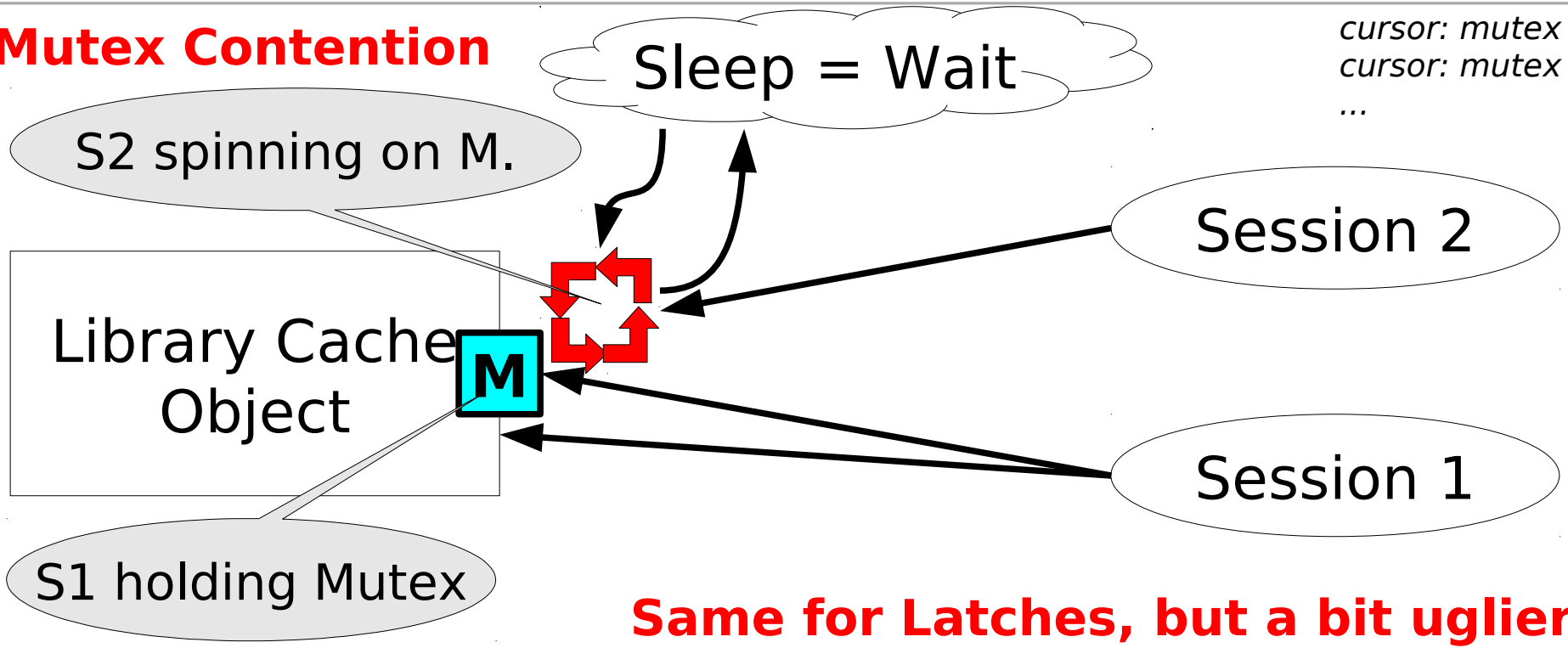
**spin!**

*buffer busy wait*
*enq: TX - allocate ITL entry*

# Mutex



## Mutex Contention



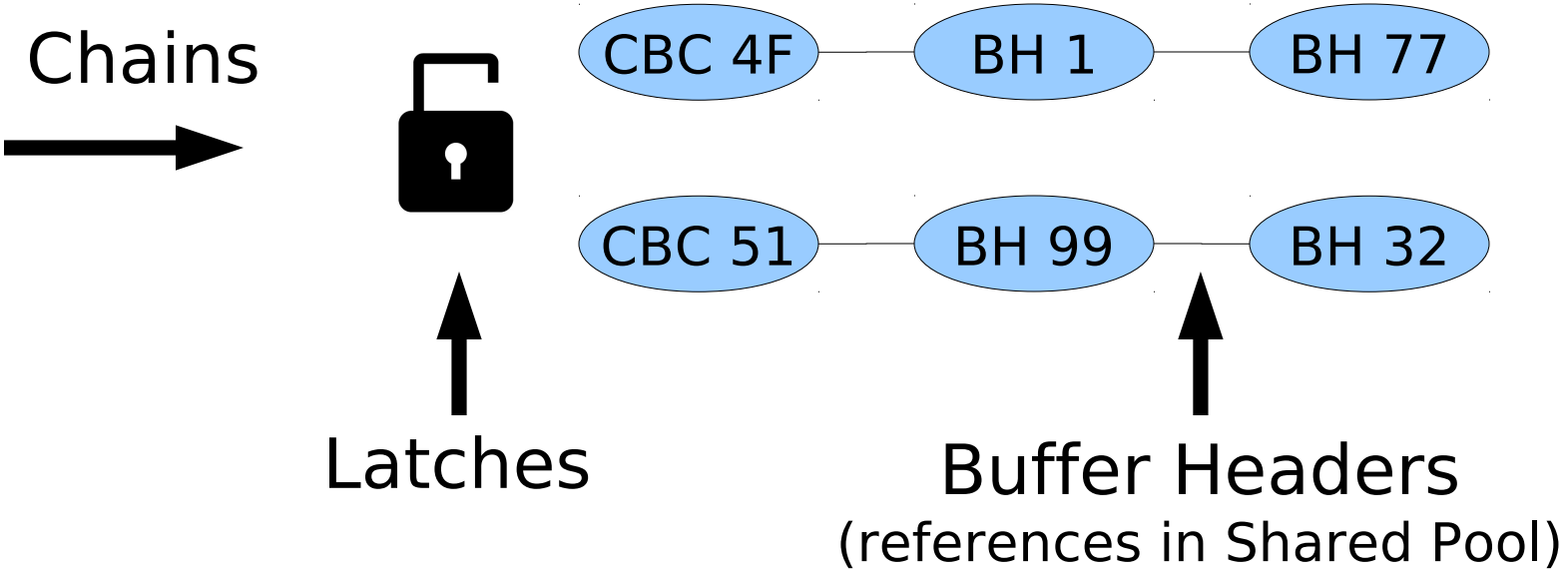

*cursor: mutex S*
*cursor: mutex X*
*...*

**Same for Latches, but a bit uglier.**

# CBC

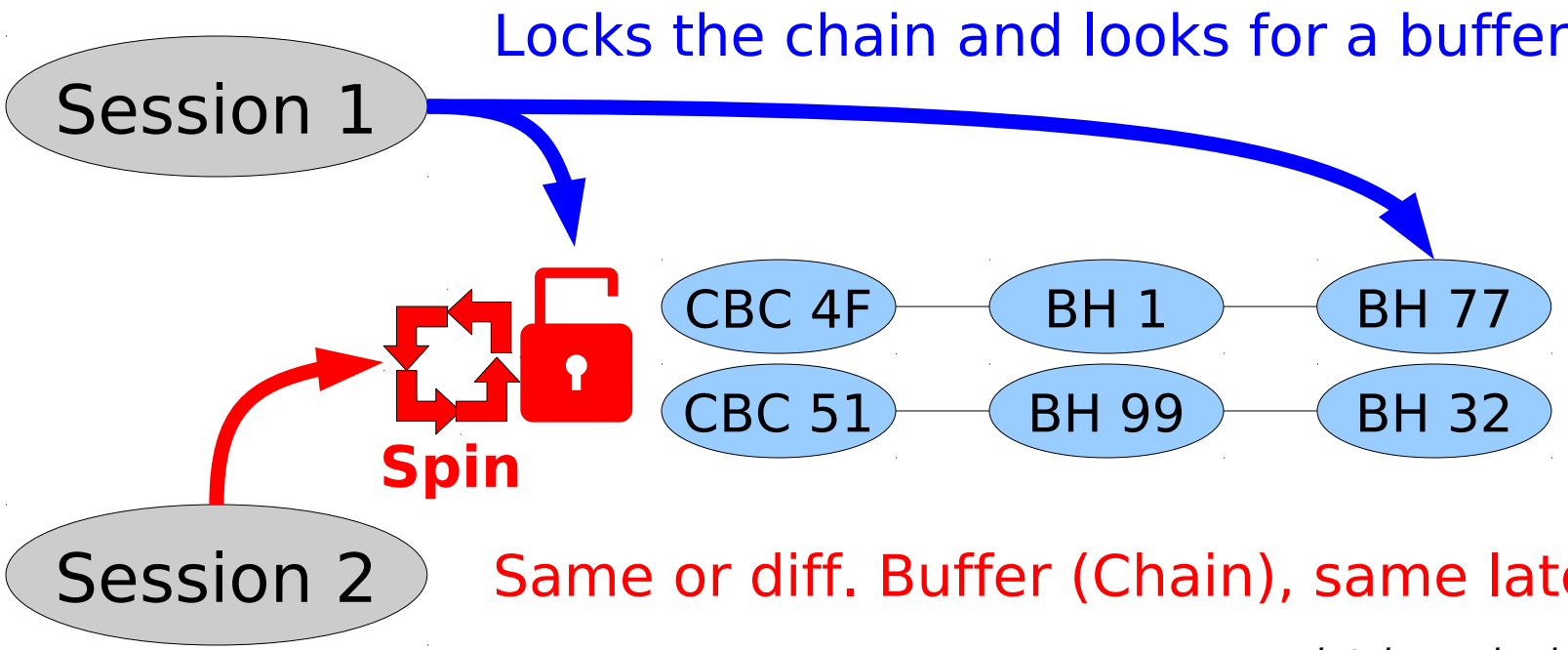


## *Cache Buffer Chains: Is this block in the BC?*



# CBC



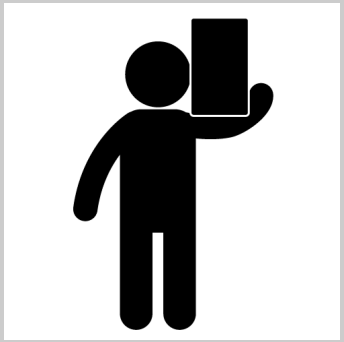



*latch: cache buffer chain*

# Conc. Wait



## Relevance-and-care chart

| System Admin | DBA | Developer |
| --- | --- | --- |
|  |  |  |

## Suggestions

- Check workload (think: SQL efficiency)
  => Reduce logical reads/writes

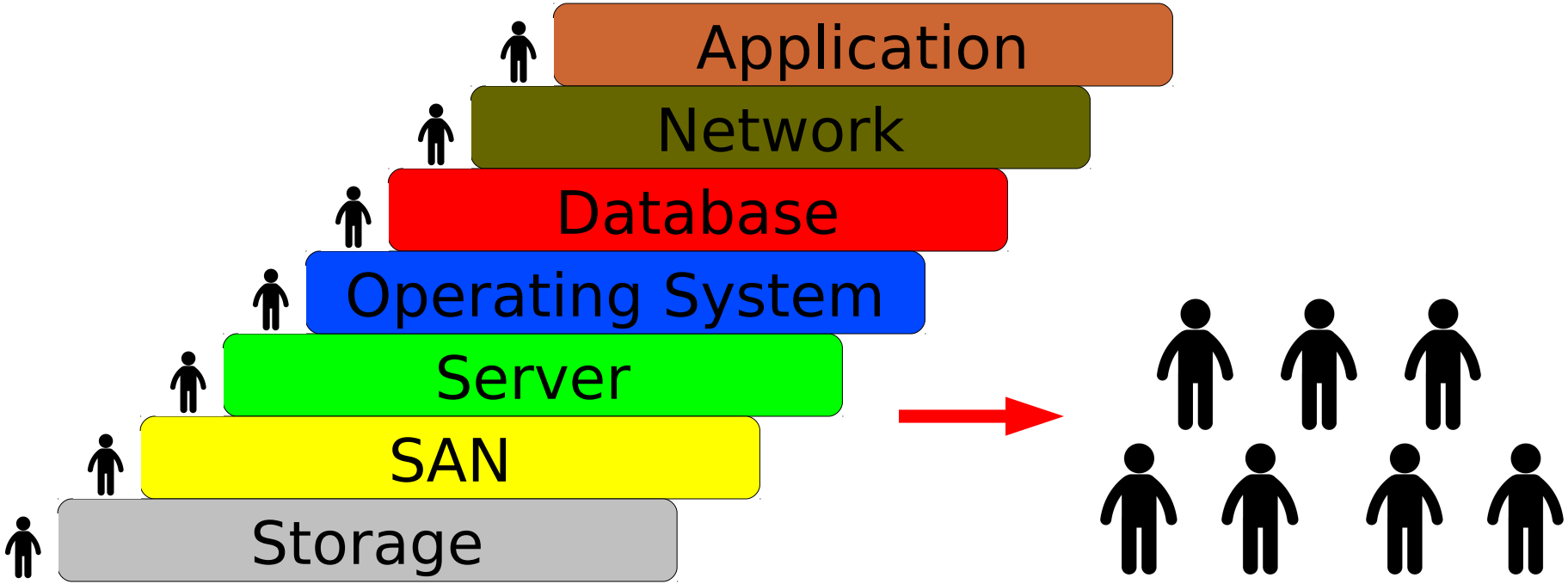
- Be ready for decent diagnosis (think in Wait Events)

# Collaborate

It's all about humans working together

# Layers

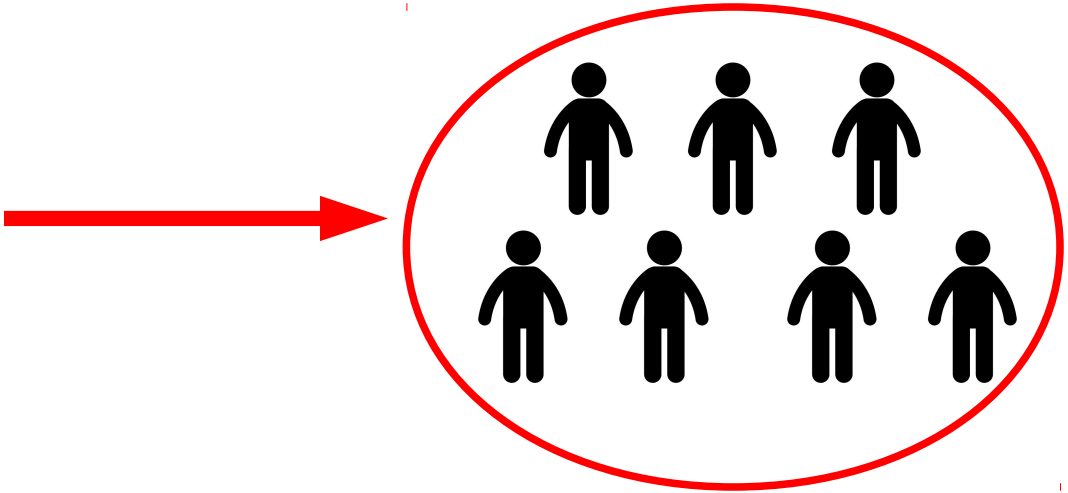



Application
Network
Database
Operating System
Server
SAN
Storage

# People



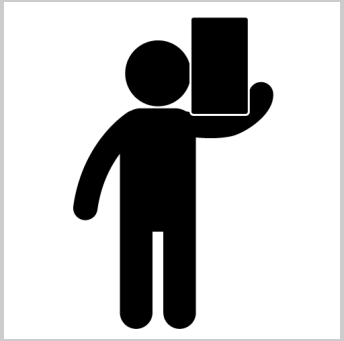
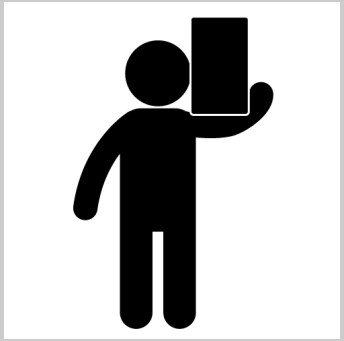
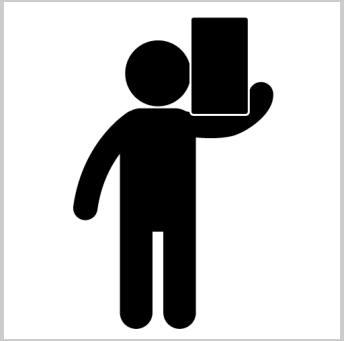
Engineers to work together



# All is slow



## Relevance-and-care chart

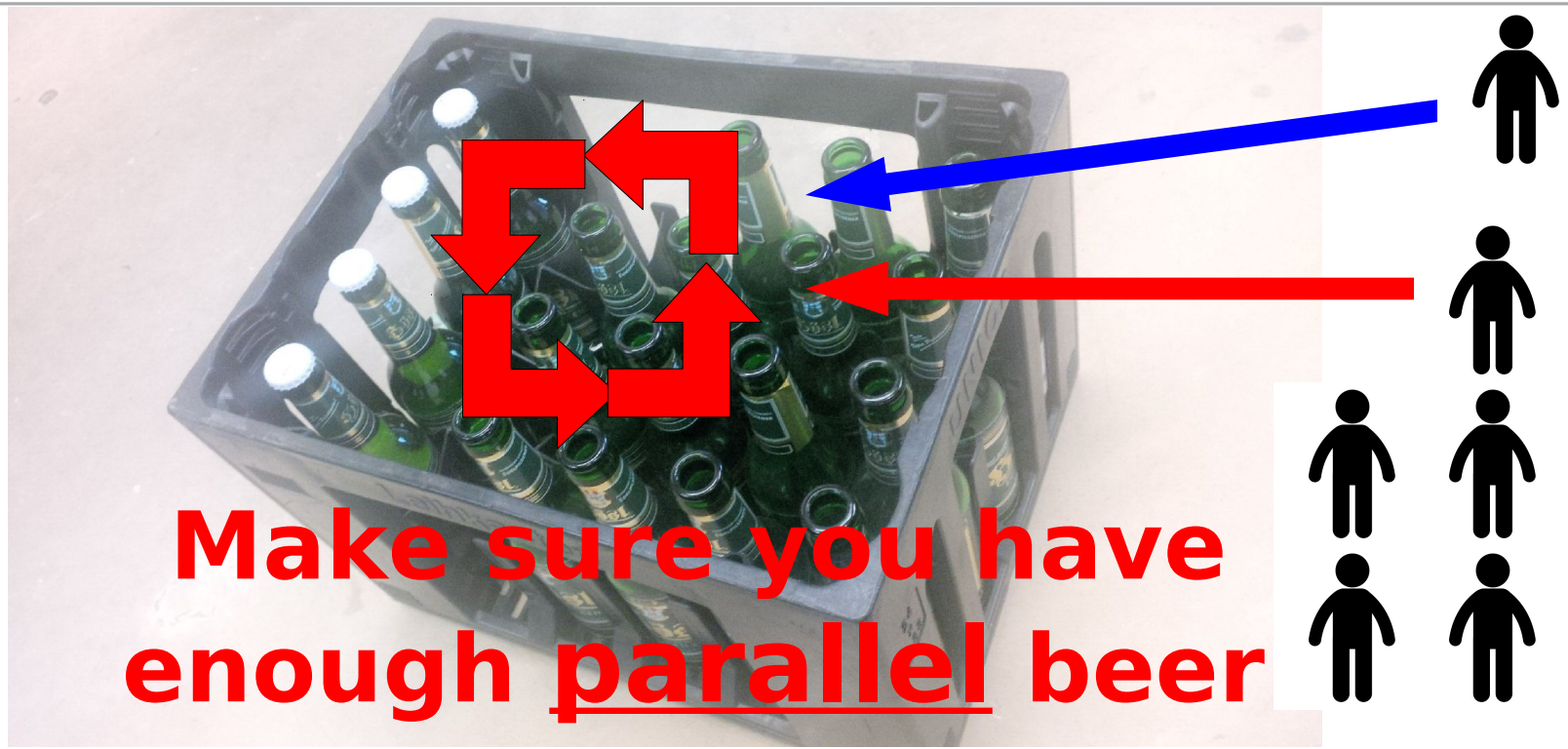
| System Admin | DBA | Developer |
| --- | --- | --- |
|  |  |  |

# It works



## Relevance-and-care chart

| System Admin | DBA | Developer |
| --- | --- | --- |
|  |  |  |


Make sure you have
enough parallel beer

COLLABORATE14

DOAG
Deutsche ORACLE-Anwendergruppe e.V.

IOUG
independent oracle users group

# Thank you very much for your attention!

Martin Klier
Senior / Lead DBA
Klug GmbH integrierte Systeme

Klug integrierte Systeme

Las Vegas, April 26th, 2012

# Read on...



## More resources on this topic

- Kevin Closson, on NUMA and Huge Pages
  https://kevinclosson.wordpress.com/2010/03/18/you-buy-a-numa-system-oracle-says-disable-numa-what-gives-part-i/
  http://kevinclosson.wordpress.com/2010/09/28/configuring-linux-hugepages-for-oracle-database-is-just-too-difficult-part-i/
- Craig Shallahamer, on Cache Buffer Chain visualization
  http://shallahamer-orapub.blogspot.de/2010/09/buffer-cache-visualization-and-tool.html
- Arup Nanda, on ITL / Locks
  http://arup.blogspot.de/2011/01/more-on-interested-transaction-lists.html
- Andrey Nikolaev on Mutexes
  “Exploring mutexes, the Oracle RDBMS retrial spinlocks”
- Ronan Bourlier & Loïc Fura, IBM
  “Oracle DB and AIX Best Practices for Performance & Tuning”
- My Oracle Support
  Doc ID 864633.1 “Enable Oracle NUMA support with Oracle Server Version 11gR2”
  Doc ID 1392497.1 “USE_LARGE_PAGES To Enable HugePages”
  Doc ID 361468.1 “HugePages on Oracle Linux 64-bit”

COLLABORATE14

# Thank you

IOUG independent oracle users group

Many people have helped with suggestions, critics or taking daily work off me during preparation and travel phase.

Guys, you are top!

Special thanks to:
My boss and company, for endorsement
My team, for digging out the interesting stuff