Oracle DB and AIX
Best Pratices for Performance & tuning

Session ID: PE129

Ronan Bourlier & Loïc Fura

5.3

Agenda

- CPU
  - Power 7
- Memory
  - AIX VMM tuning
  - Active Memory Expansion
- IO
  - Storage consideration
  - AIX LVM Striping
  - Disk/Fiber Channel driver optimization
  - Virtual Disk/Fiber channel driver optimization
  - AIX mount option
  - Asynchronous IO
- NUMA Optimization
- Other Tips

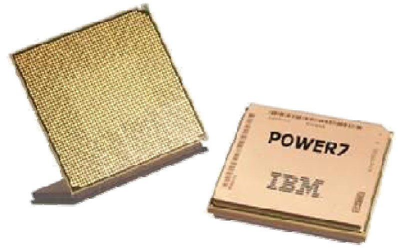
Agenda

- CPU
  - Power 7
- Memory
  - AIX VMM tuning
  - Active Memory Expansion
- IO
  - Storage consideration
  - AIX LVM Striping
  - Disk/Fiber Channel driver optimization
  - Virtual Disk/Fiber channel driver optimization
  - AIX mount option
  - Asynchronous IO
- NUMA Optimization
- Other Tips

Power 7 (Socket/Chip/Core/Threads)

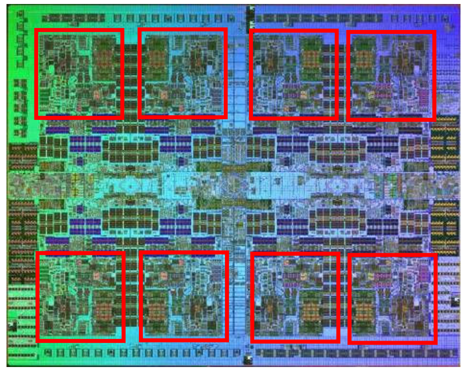
Power Hardware

Each Power7 socket = chip 1 Power7 Chip



Each Power7 Chip =

8 x Power7
4Ghz Cores



Each Power7 Core =
4 HW SMT threads

SMT SMT SMT SMT

Software

lcpu lcpu lcpu lcpu

In AIX, when SMT is enable,
each SMT thread is seen like a logical CPU

32 sockets = 32 chips = 256 cores = 1024 SMT threads = 1024 AIX logical CPU

Power7 specific tuning

- Use SMT4

Give a cpu boost performance to handle more concurrent threads in parallel

- Disabling HW prefetching.

Usually improve performance on Database Workload on big SMP Power system (> P750)

`# dscrctl -n -b -s 1` **(this will dynamically disable HW memory prefetch and keep this configuration across reboot)**

`# dscrctl -n -b -s 0` **(to reset HW prefetching to default value)**

- Use Terabyte Segment aliasing – Enabled by default on AIX 7.1

Improve CPU performance by reducing SLB miss (segment address resolution)

`# vmo -p -o esid_allocator=1` **(To enable it on AIX 6.1)**

- Use Large Pages (16MB memory pages)

Improve CPU performance by reducing TLB miss (Page address resolution)

Configure larges pages (xxxx= # of segments of 16M you want)

`# vmo -r -o lgpg_regions=xxxx -o lgpg_size=16777216`

Enable Oracle userid to use Large Pages

`# chuser capabilities=CAP_NUMA_ATTACH,CAP_BYPASS_RAC_VMM,CAP_PROPAGATE` **oracle**

`# export ORACLE_SGA_PGSZ=16M` **before starting oracle (with oracle user)**

check large page usage for Oracle user

`# svmon -mwU oracle`

Power7 Automatic System Optimization

- ASO – Active System Optimizer
  - Available since AIX 7.1 TL1 SP1
  - Start commands
    - `asoo -p -o aso_active=1`
    - `startsrc -s aso`
  - Monitor processes activity
  - Automatic tuning of
    - cache affinity
    - memory affinity

- DSO – Dynamic System Optimizer
  - Available with AIX 7.1 TL1 SP7 and AIX 6.1 TL8 SP1 (enterprise edition)
  - Add following features to ASO :
    - Automatic 16MB pages
    - HW Prefetching automatic tuning

- **Monitoring**
  - `/var/log/aso/aso.log (aso status runing/stopped)`
  - `/var/log/aso/aso_process.log (aso actions)`

Agenda

- CPU
  - Power 7
- Memory
  - AIX VMM tuning
  - Active Memory Expansion
- IO
  - Storage consideration
  - AIX LVM Striping
  - Disk/Fiber Channel driver optimization
  - Virtual Disk/Fiber channel driver optimization
  - AIX mount option
  - Asynchronous IO
- NUMA Optimization
- Other Tips

Virtual Memory Management

1 - AIX is started, applications load some computational pages into the memory.

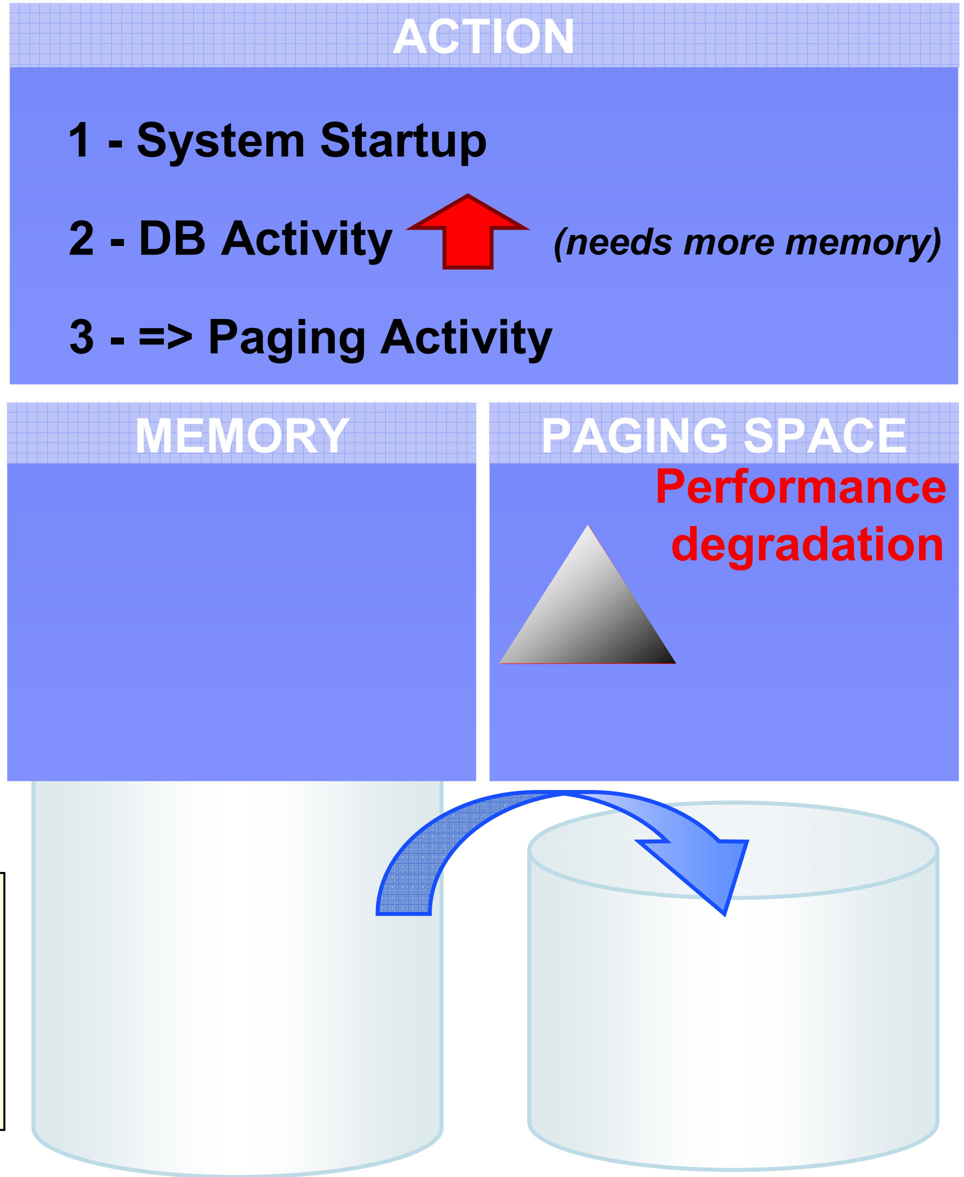
As a UNIX system, AIX will try to take advantage of the free memory by using it as a cache file to reduce the IO on the physical drives.

2 - The activity is increasing, the DB needs more memory but there is no free pages available. LRUD *(AIX page stealer)* is starting to free some pages into the memory.

3 – On older version of AIX (< AIX 6.1) with default settings, LRUD will page out some computational pages instead of removing only pages from the File System Cache.

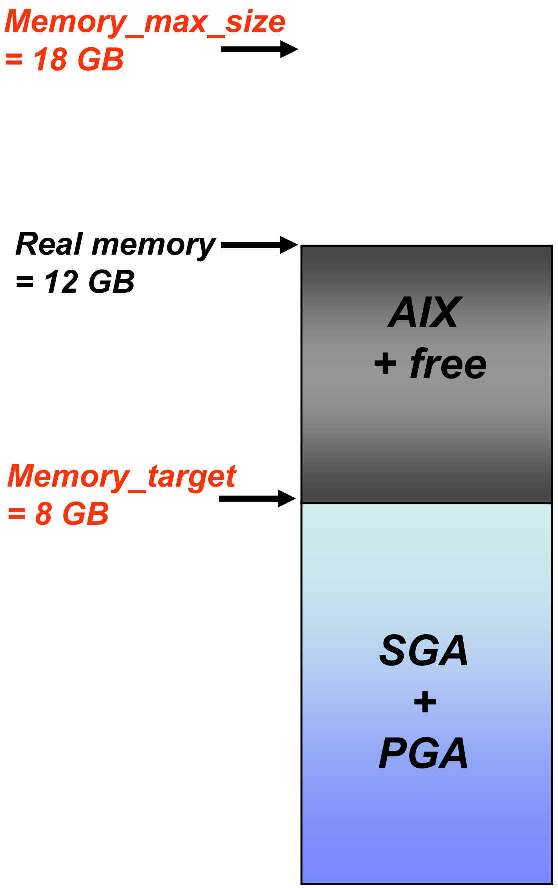
- *Objective :*

Tune the VMM to protect computational pages **(Programs, SGA, PGA) from being paged out and force the LRUD to** steal pages from FS-CACHE only.



Memory : Use jointly AIX dynamic LPAR and Oracle dynamic allocation of memory + AMM

Initial configuration



Scenario :

Oracle tuning advisor indicates that SGA+PGA need to be increased to 11GB:

memory_target **can be increased dynamically to 11GB but real memory is only 12GB, so it needs to be increased as well.**

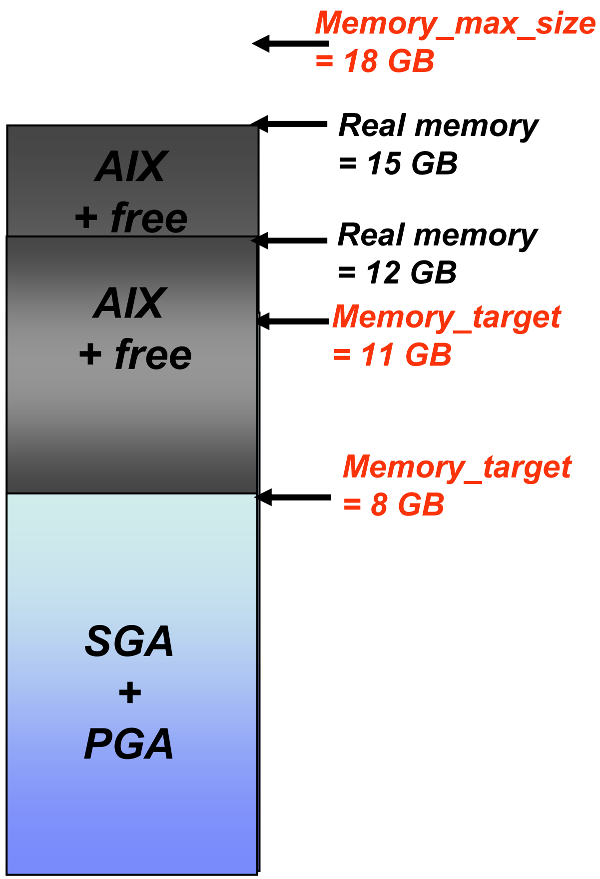
- **Step 1 - Increase physical memory allocated to the LPAR to 15 GB**

```
chhwres -r mem -m <system> -o a -p
<LPAR name> -q 3072
```

- **Step 2 - Increase SGA+PGA allocated to the instance of the database to 11GB**

```
alter system set  memory_target=11GB;
```

Final configuration



➔ ***Memory allocated to the system has been increased dynamically, using AIX DLPAR***

➔ ***Memory allocated to Oracle (SGA and PGA) has been increased on the fly***

Agenda

- CPU
  - Power 7
- Memory
  - AIX VMM tuning
  - Active Memory Expansion
- IO
  - Storage consideration
  - AIX LVM Striping
  - Disk/Fiber Channel driver optimization
  - Virtual Disk/Fiber channel driver optimization
  - AIX mount option
  - Asynchronous IO
- NUMA Optimization
- Other Tips

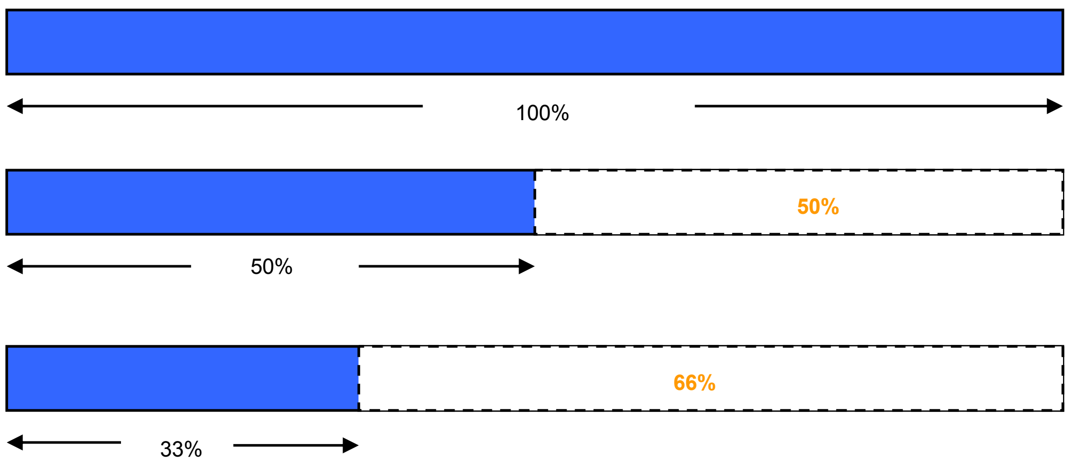
Power 7 : AME example

Active Memory Expansion is a POWER7 new feature that expands system's effective memory capacity by dynamicall compressing real memory. Its activation is on LPAR level and transparent for applications.

AME goal is to improve the system memory usage. It allows to increase the global system's throughtput and/or reduces the Memory/core ratio application requirements with a low impact on performances.

AME test on Oracle DB eBS Batch. SGA Size = 112GB

Test (Exp. Factor)	Memory Config	Sample Config
Test 0 – Baseline (N/A)	100%	**Mem Size:** 120GB **AME Factor**: Disabled
Test 1 – 200% Expansion (2.0)	50% / 50%	**Mem Size:** 60GB **AME Factor**: 2.00
Test 2 – 300% Expansion (3.00)	33% / 66%	**Mem Size:** 40GB **AME Factor**: 3.00



Power 7 : AME example 1 (test results)

eBS DB with 24 cores and SGA Size=112GB

TEST	Nb CPU	Physical Memory	AME Factor	BATCH Duration	CPU Consumption
0	24	120 GB	none	124 min	avg: 16.3 cpu
1	24	60 GB	2.0	127 min	avg: 16.8 cpu
2	24	40 GB	3.0	134 min	avg: 17.5 cpu

The impact of AME on batch duration is really low (<10%) with few cpu overhead (7%), even with 3 times less memory.

POWER7+ processor embeds on chip hardware compression, expect less CPU consumption for even more compressed memory

- **Note: This is an illustrative scenario based on using a sample workload. This data represents measured results in a controlled lab environment. Your results may vary.**

Agenda

- CPU
  - Power 7
- Memory
  - AIX VMM tuning
  - Active Memory Expansion
- IO
  - Storage consideration
  - AIX LVM Striping
  - Disk/Fiber Channel driver optimization
  - Virtual Disk/Fiber channel driver optimization
  - AIX mount option
  - Asynchronous IO
- NUMA Optimization
- Other Tips

IO : Database Layout

- **Having a good Storage configuration is a key point :**
  - Because disk is the slowest part of an infrastructure
  - Reconfiguration can be difficult and time consuming

Stripe and mirror everything (S.A.M.E) approach:

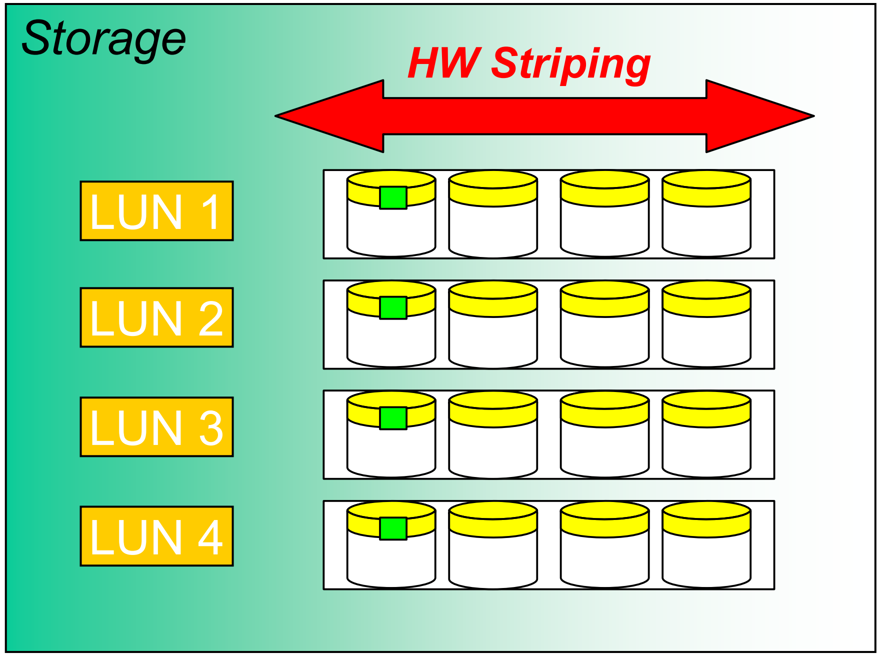
Oracle recommendation

- Goal is to balance I/O activity across all disks, loops, adapters, etc...
- Avoid/Eliminate I/O hotspots
- Manual file-by-file data placement is time consuming, resource intensive and iterative
- Additional advices to implement SAME :
  - apply the SAME strategy to data, indexes
  - if possible separate redologs (+archivelogs)

IO : RAID Policy with ESS, DS6/8K

- **RAID-5 vs. RAID-10 Performance Comparison**

I/O Profile	*RAID-5*	*RAID-10*
Sequential Read	**Excellent**	**Excellent**
Sequential Write	**Excellent**	**Good**
Random Read	**Excellent**	**Excellent**
Random Write	**Fair**	**Excellent**



- With Enterprise class storage (with huge cache), RAID-5 performances are comparable to RAID-10 (for most customer workloads)
- Consider RAID-10 for workloads with a high percentage of random write activity (> 25%) and high I/O access densities (peak > 50%)

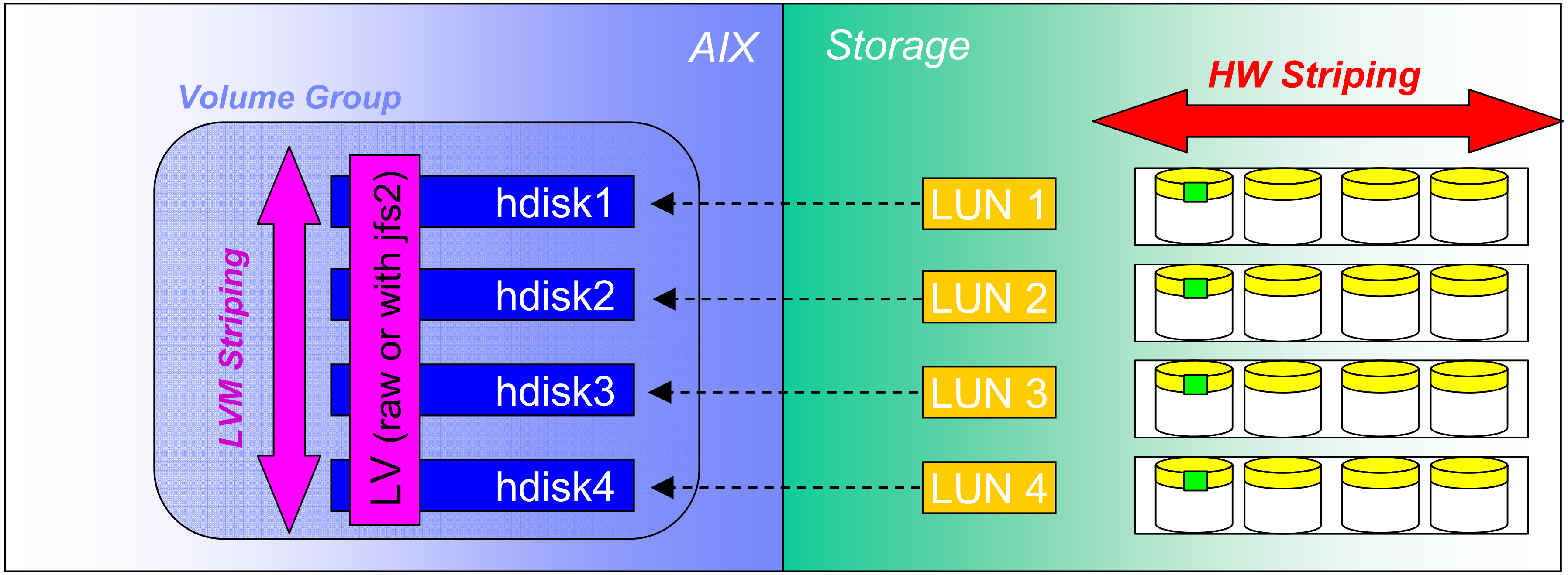
Use RAID-5 or RAID-10 to create striped LUNs

If possible try to minimize the number of LUNs per RAID array to avoid contention on physical disk.

Agenda

- CPU
  - Power 7
- Memory
  - AIX VMM tuning
  - Active Memory Expansion
- IO
  - Storage consideration
  - AIX LVM Striping
  - Disk/Fiber Channel driver optimization
  - Virtual Disk/Fiber channel driver optimization
  - AIX mount option
  - Asynchronous IO
- NUMA Optimization
- Other Tips

IO : 2nd Striping (LVM)



1. Luns are striped across physical disks (stripe-size of the physical RAID : ~ 64k, 128k, 256k)
2. LUNs are seen as hdisk device on AIX server.
3. Create AIX Volume Group(s) (VG) with LUNs from multiple arrays
4. Logical Volume **striped** across hdisks (stripe-size : 8M, 16M, 32M, 64M)

   => each read/write access to the LV are well balanced accross LUNs and use the maximum number of physical disks for best performance.

Agenda

- CPU
  - Power 7
- Memory
  - AIX VMM tuning
  - Active Memory Expansion
- IO
  - Storage consideration
  - AIX LVM Striping
  - Disk/Fiber Channel driver optimization
  - Virtual Disk/Fiber channel driver optimization
  - AIX mount option
  - Asynchronous IO
- NUMA Optimization
- Other Tips

IO : Disk Subsystem Advices

- **Check if the definition of your disk subsystem is present in the ODM.**
  - If the description shown in the output of "**lsdev –Cc disk**" the word "**Other**", then it means that AIX doesn't have a correct definition of your disk device in the ODM and use a generic device definition.

```
hdisk7  Available 09-08-02   MPIO Other FC SCSI Disk Drive
hdisk8  Available 09-08-02   MPIO Other FC SCSI Disk Drive
hdisk9  Available 09-08-02   MPIO Other FC SCSI Disk Drive
hdisk10 Available 09-08-02   MPIO Other FC SCSI Disk Drive
hdisk11 Available 09-08-02   MPIO Other EMC SYMMETRIX Disk
hdisk12 Available 09-08-02   MPIO Other EMC SYMMETRIX Disk
hdisk13 Available 09-08-02   MPIO Other EMC SYMMETRIX Disk
hdisk14 Available 09-08-02   MPIO Other EMC SYMMETRIX Disk
```

Generic device definition
bad performance

  - In general, a generic device definition provides **far from optimal performance** since it doesn't properly customize the hdisk device :

    exemple : hdisk are created with a queue_depth=1

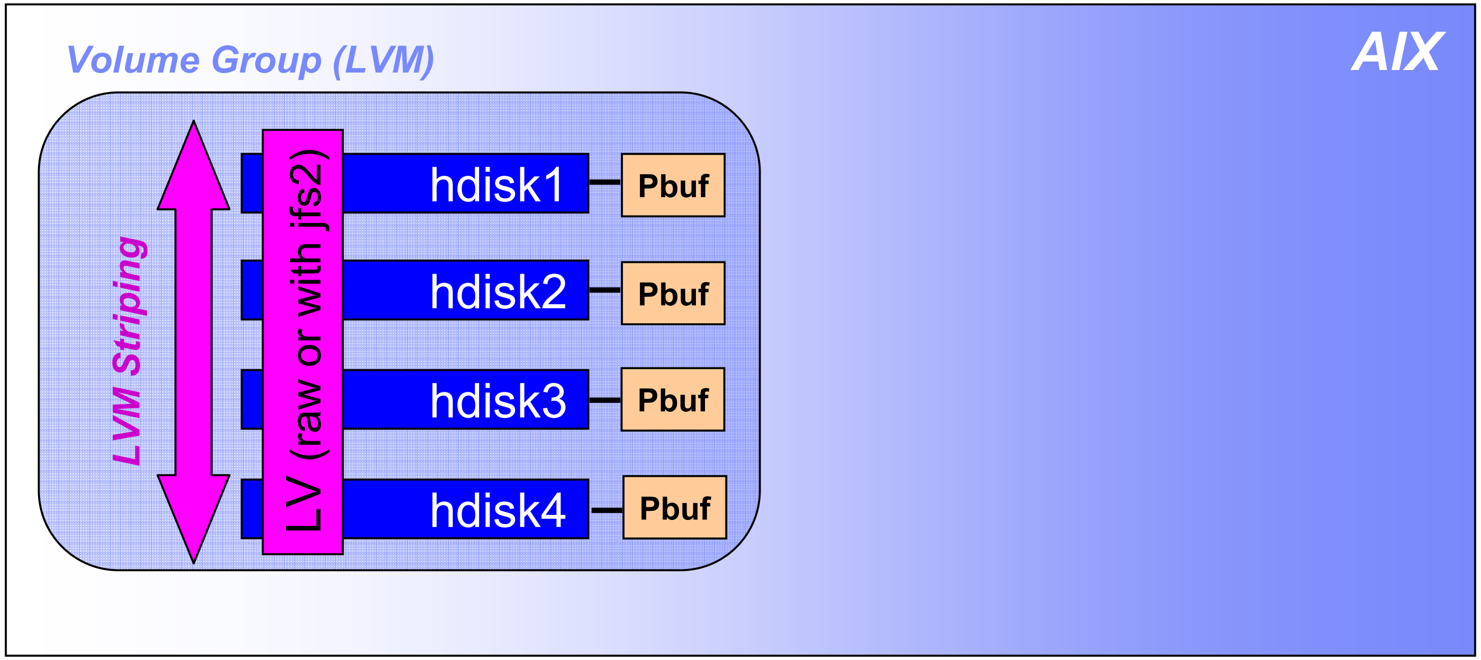
1. Contact your vendor or go to their web site to download the correct ODM definition for your storage subsystem. It will setup properly the "hdisk" accordingly to your hardware for optimal performance.
2. If AIX is connected to the storage subsystem with several Fiber Channel Cards for performance, don't forget to install a **multipath device driver** or **path control module**.

   - sdd or sddpcm for IBM DS6000/DS8000

   - powerpath for EMC disk subsystem

   - hdlm for Hitachi etc....

IO : AIX IO tuning (1) – LVM Physical buffers (pbuf)

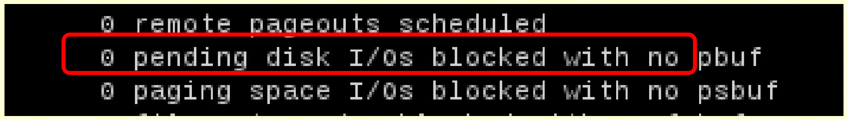


1. Each LVM physical volume as a physical buffer (pbuf)

2. **# vmstat –v** command help you to detect lack of pbuf.

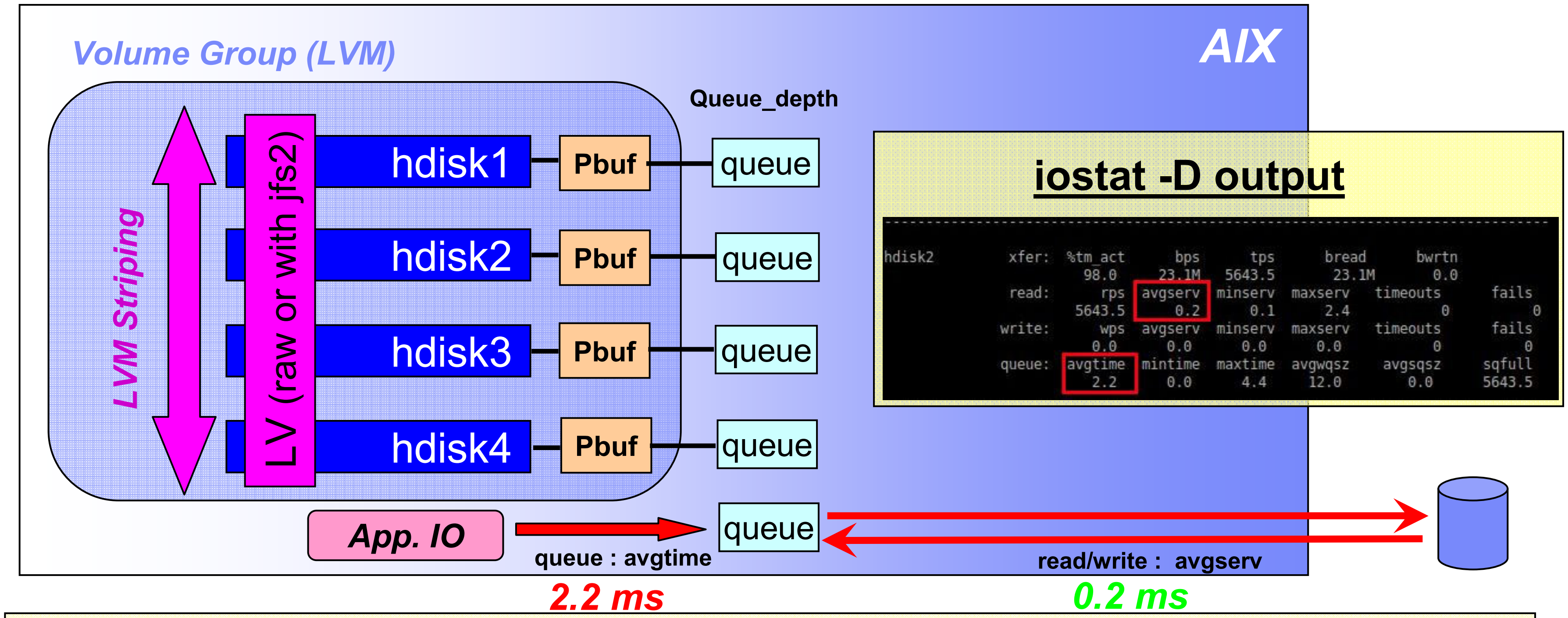
3. If there is a lack of pbuf, 2 solutions:
   - Add more luns (this will add more pbuf)
   - Or increase pbuf size :

     # lvmo –v <vg_name> -o pv_pbuf_count=XXX



Output of “vmstat –v” command

IO : AIX IO tuning (2) – hdisk queue_depth (qdepth)

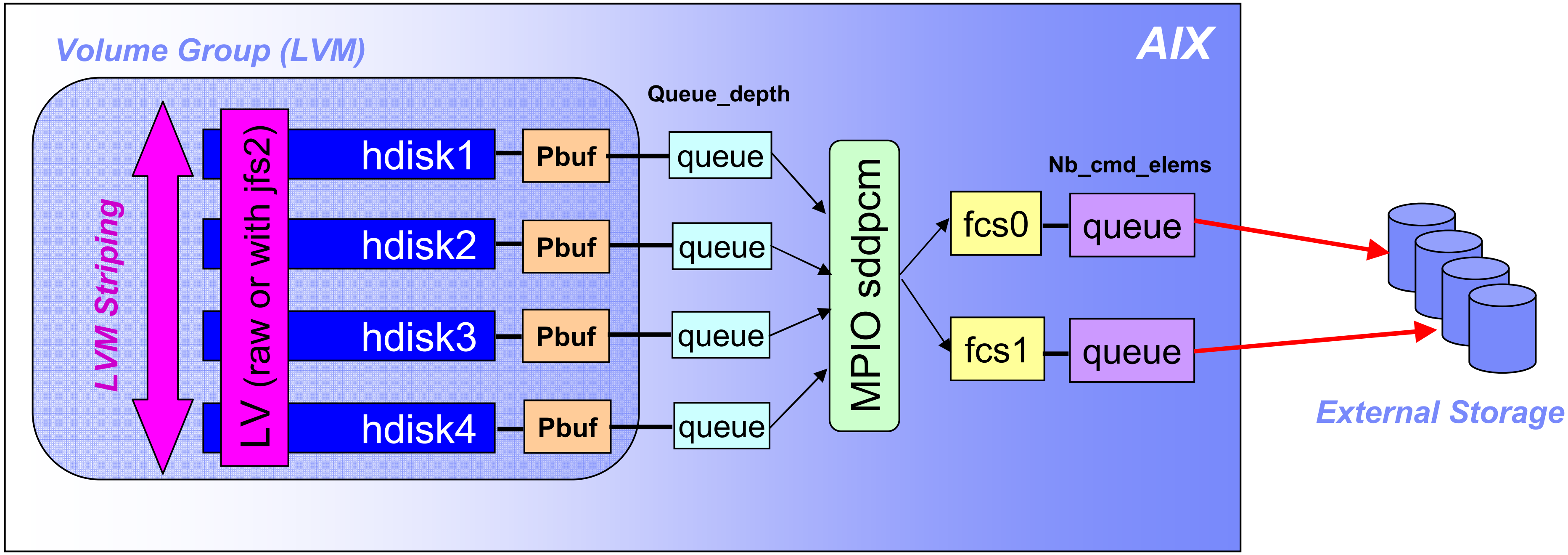


1. Each AIX hdisk has a “Queue” called queue depth. This parameter set the number of // queries that can be send to Physical disk.

2. To know if you have to increase qdepth, use iostat -D and monitor :
   avgserv, avgtime

1. If you have :
   avgserv < 2-3ms => *this mean that Storage behave well (can handle more load)*
   And “**avgtime**” > 1ms => *this mean that disk queue are full, IO wait to be queued*
   => INCREASE hdisk queue depth (`# chdev -l hdiskXX -a queue_depth=YYY`***)***

IO : AIX IO tuning (3) – HBA tuning (num_cmd_elems)



1. Each HBA FC adapter has a queue "nb_cmd_elems". This queue has the same role for the HBS as the qdepth for the disk.

2. **Rule of thumb: nb_cmd_elems= (sum of qdepth) / nb HBA**

3. Changing nb_cmd_elems : **`# chdev –l fcsX –o nb_cmd_elems=YYY`**
You can also change the max_xfer_size=0x200000
and lg_term_dma=0x800000 with the same command

```
FC SCSI Adapter Driver Information
  No DMA Resource Count: 0
  No Adapter Elements Count: 0
  No Command Resource Count: 0
```

fcstat fcs0 output

These changes use more memory and must be made with caution, check first with : `# fcstat fcsX`

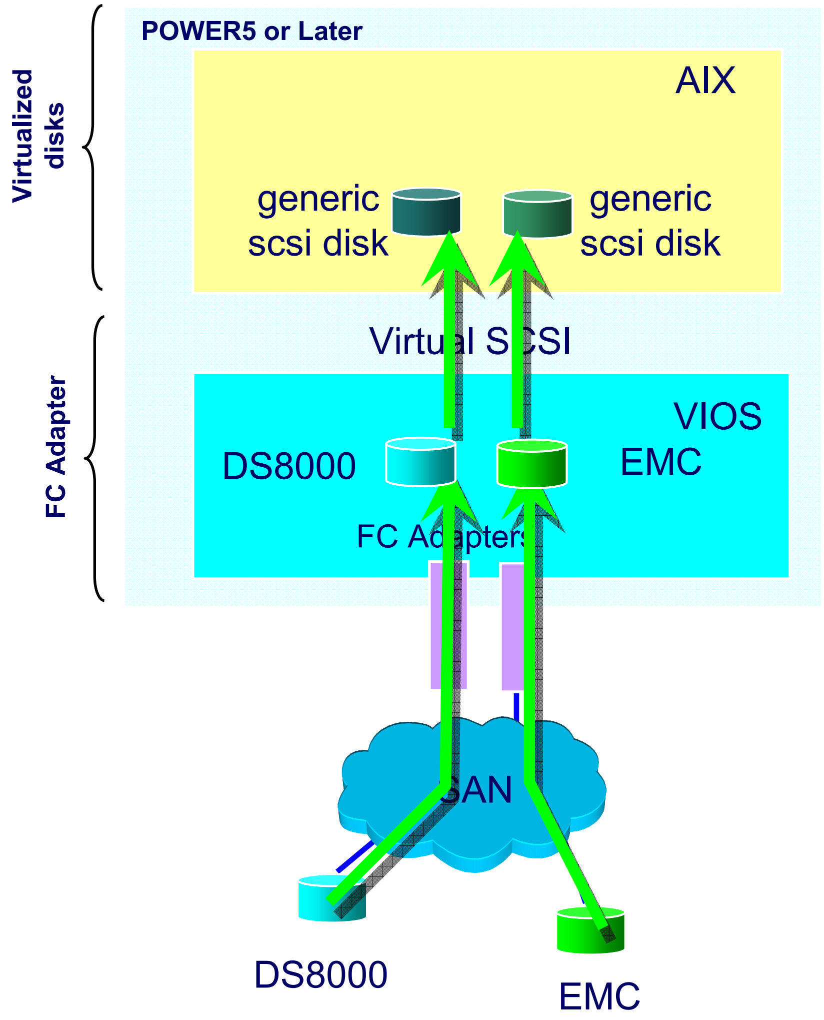
Agenda

- CPU
  - Power 7
- Memory
  - AIX VMM tuning
  - Active Memory Expansion
- IO
  - Storage consideration
  - AIX LVM Striping
  - Disk/Fiber Channel driver optimization
  - Virtual Disk/Fiber channel driver optimization
  - AIX mount option
  - Asynchronous IO
- NUMA Optimization
- Other Tips

Virtual SCSI

Virtual I/O helps reduce hardware costs by sharing disk drives

Virtual SCSI model



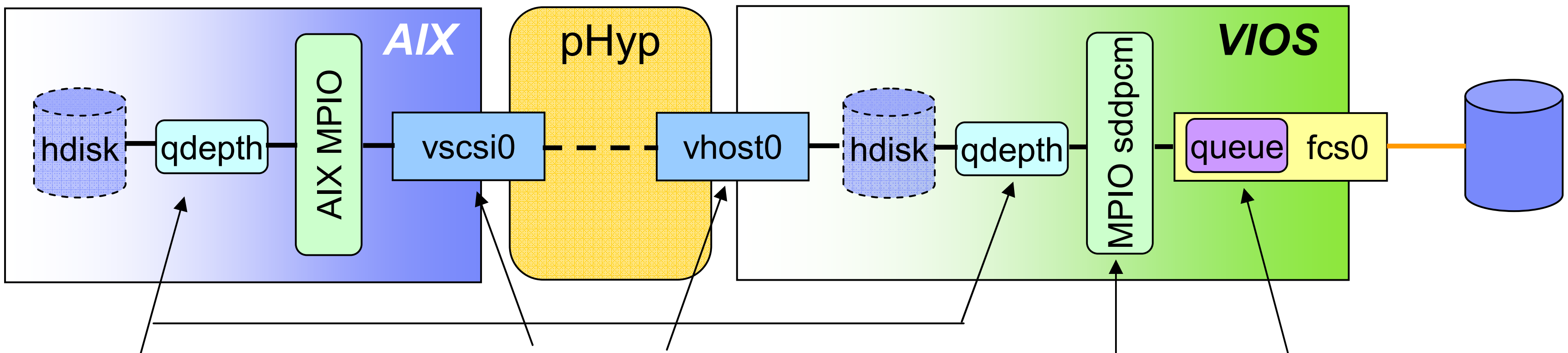
Micro-partition sees disks as vSCSI (Virtual SCSI) devices

- Virtual SCSI devices added to partition via HMC
- LUNs on VIOS accessed as vSCSI disk
- VIOS must be active for client to boot

VIOS owns physical disk resources

- LVM based storage on VIO Server
- Physical Storage can be SCSI or FC
  - Local or remote

VIOS : VSCSI IO tuning



 # lsdev –Cc disk
```
hdisk0 Available Virtual SCSI Disk Drive
```

Storage driver cannot be installed on the lpar
⇒ Default qdepth=3 !!! **Bad performance**

⇒ **# chdev –l hdisk0 –a queue_depth=20**
+ monitor svctime/wait time with nmon to adjust queue depth

* You have to set same queue_depth for the source hdisk on the VIOS

HA: *(dual vios conf.)*
change vscsi_err_recov to fast_fail
 # chdev –l vscsi0 –a vscsi_err_recov=fast_fail

Performance:
No perf tuning can be made;

We just know that each vscsi can handle
512 cmd_elems. *(2 are reserved for the adapter and 3 reserved for each vdisk)*

So, use the following formula to find the number of disk you can attache behind a vscsi adapter.

Nb_luns= (512 – 2) / (Q + 3)
With Q=qdepth of each disk
If more disks are needed => add vscsi

Install Storage Subsystem Driver on the VIO

Monitor VIO FC activity with nmon
(interactive: press “a” or “^”)
(reccord with “-^” option)

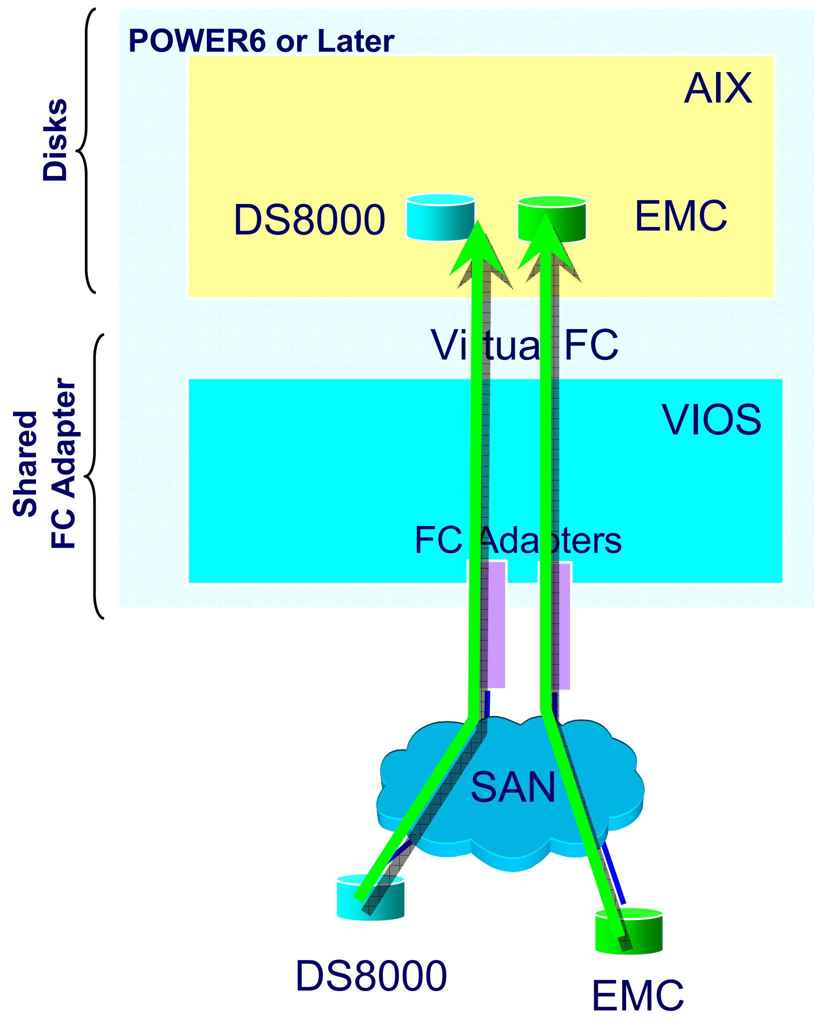
Adapt num_cmd_elems accordingly with sum of qdepth.
check with: **# fcstat fcsX**

HA:
Change the following parameters:
 # chdev –l fscsi0 –a fc_err_recov=fast_fail
 # chdev –l fscsi0 –a dyntrk=yes

NPIV Simplifies SAN Management

N-Port ID Virtualization

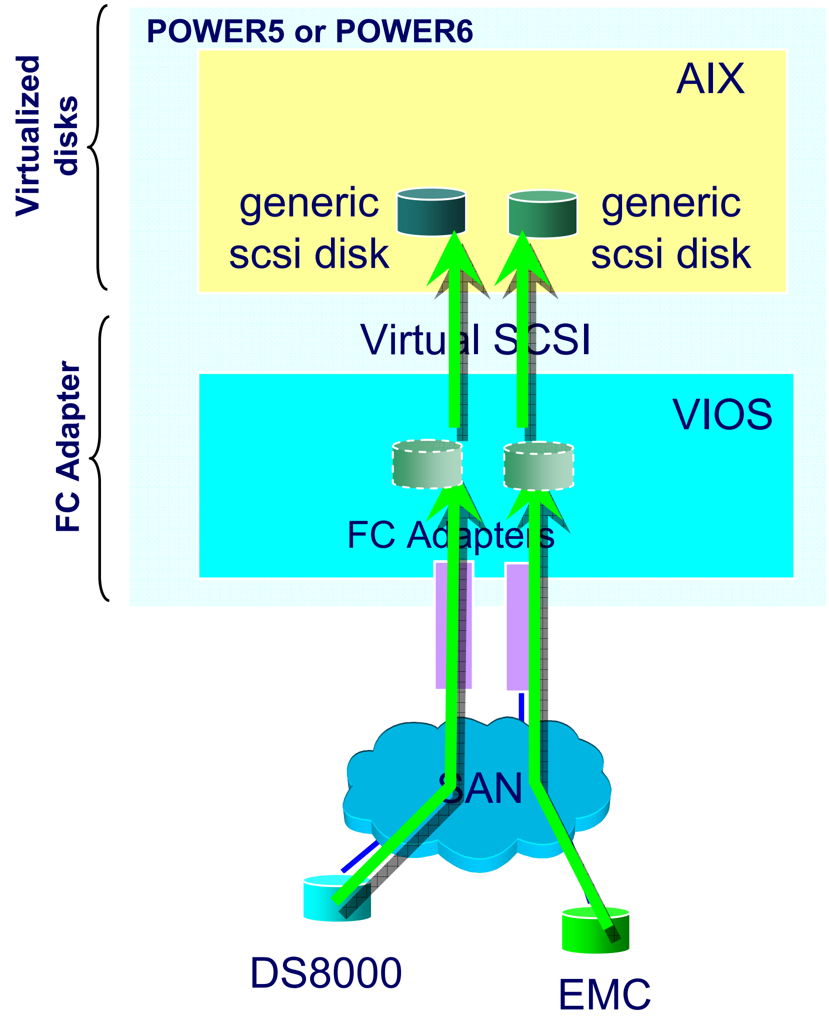
- LPARs own virtual FC adapters
- LPARs have direct visibility on SAN (Zoning/Masking)
- Virtual adapter can be assigned to multiple operating systems sharing the physical adapter
- Tape Library Support



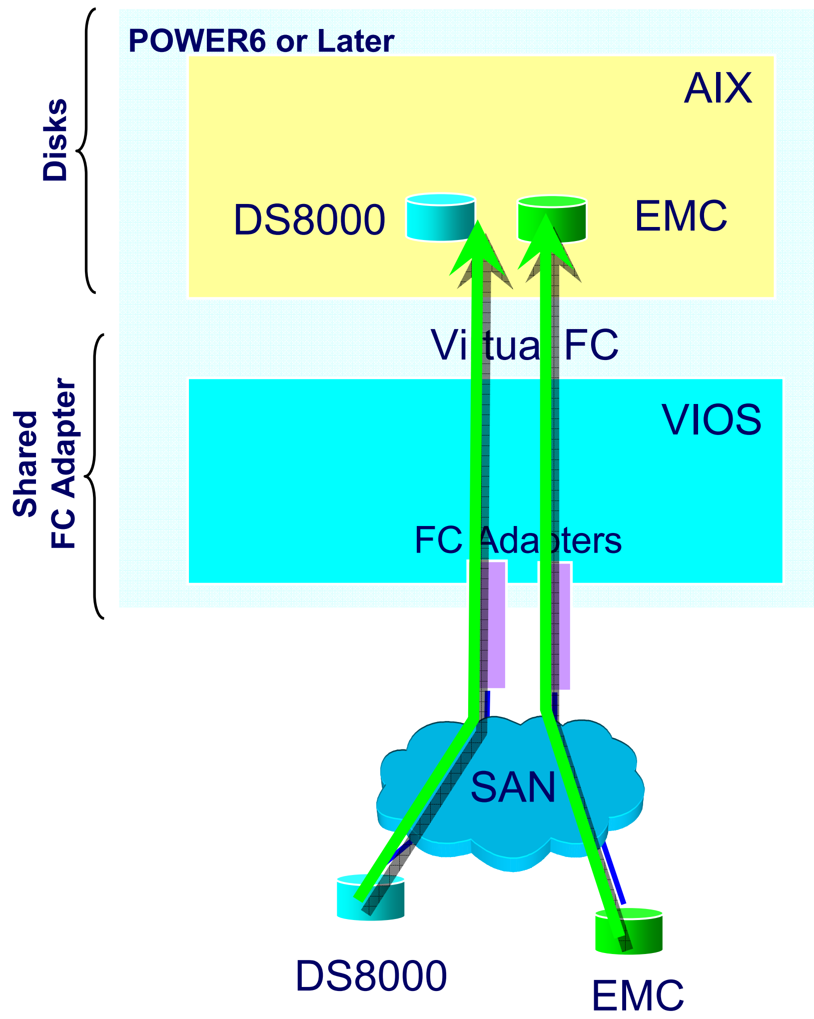
- VIOS owns physical FC adapters
- VIOS virtualizes FC to clients partitions
- VIOS Fiber Channel adapter supports Multiple World Wide Port Names / Source Identifiers
- Physical adapter appears as multiple virtual adapters to SAN / end-point device
- VIOS must be active for client to boot

NPIV Simplifies SAN Management

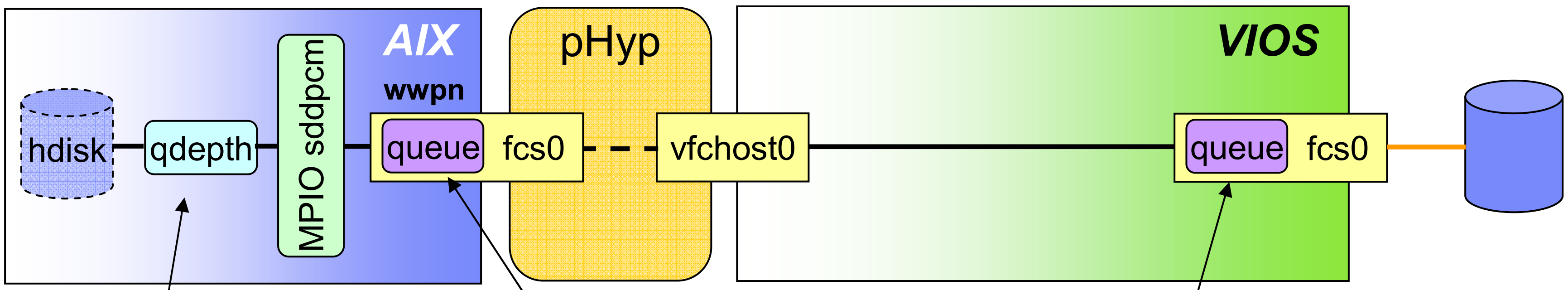
Virtual SCSI model



N-Port ID Virtualization



VIOS: NPIV IO tuning



 # lsdev –Cc disk
hdisk0 Available MPIO FC 2145

Storage driver must be installed on the lpar
⇒Default qdepth is set by the drivers

⇒Monitor “svctime” / “wait time” with nmon or iostat to tune the queue depth

__HA__: *(dual vios conf.)*
Change the following parameters:

```
# chdev -l fscsi0 -a fc_err_recov=fast_fail
# chdev -l fscsi0 -a dyntrk=yes
```

__Performance:__
Monitor fc activity with nmon
(interactive: option “a” or “^”)
(reccording : option “-^”)

Adapt num_cmd_elems
Check fcstat fcsX

__Performance:__
Monitor fc activity with nmon
(interactive: **option “^” only**)
(reccording : option “-^”)

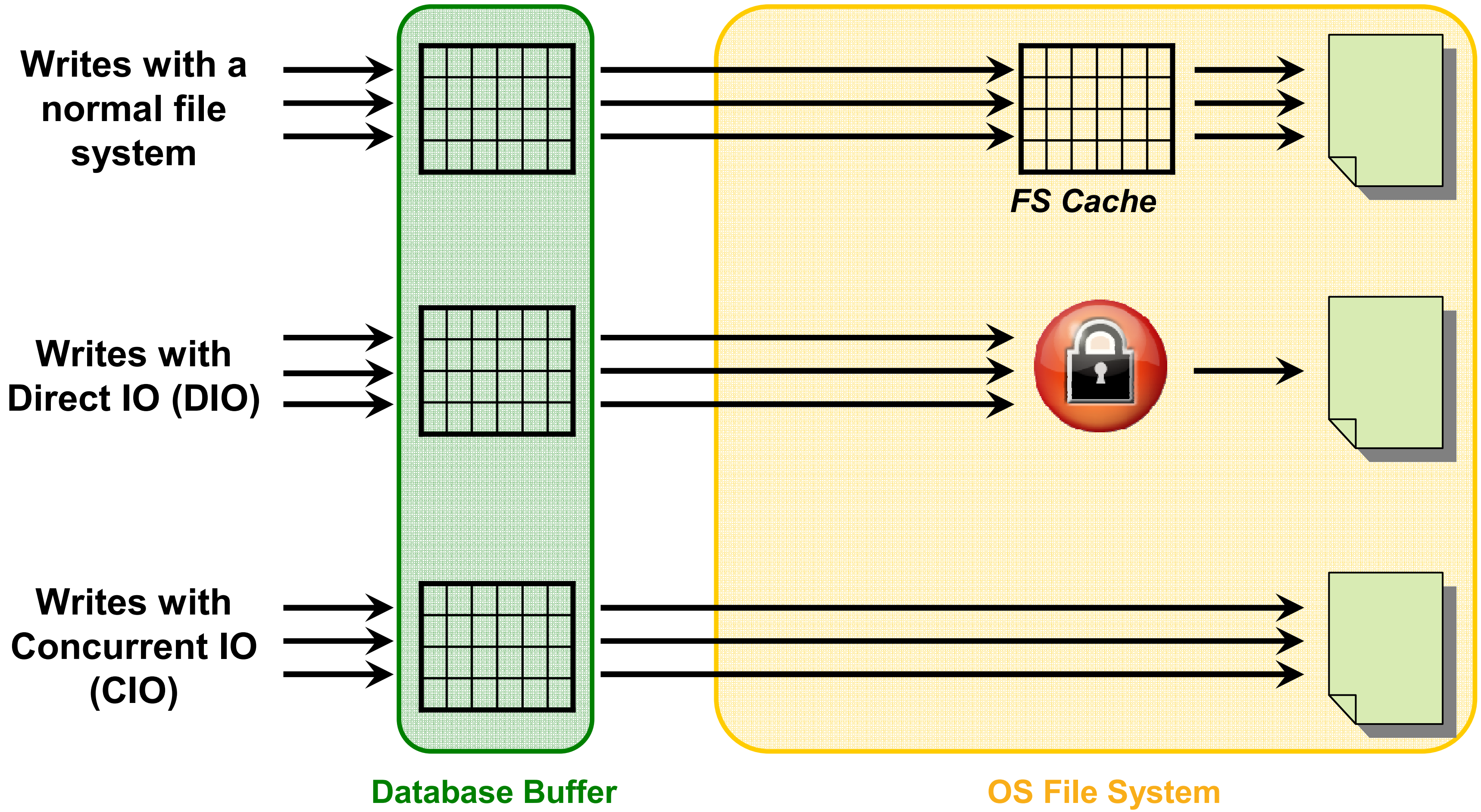
Adapt num_cmd_elems
Check fcstat fcsX

⇒Should be = Sum of vfcs num_cmd_elems connected to the backend device

Agenda

- CPU
  - Power 7
- Memory
  - AIX VMM tuning
  - Active Memory Expansion
- IO
  - Storage consideration
  - AIX LVM Striping
  - Disk/Fiber Channel driver optimization
  - Virtual Disk/Fiber channel driver optimization
  - AIX mount option
  - Asynchronous IO
- NUMA Optimization
- Other Tips

IO : Filesystems Mount Options (DIO, CIO)



IO : Filesystems Mount Options (DIO, CIO)

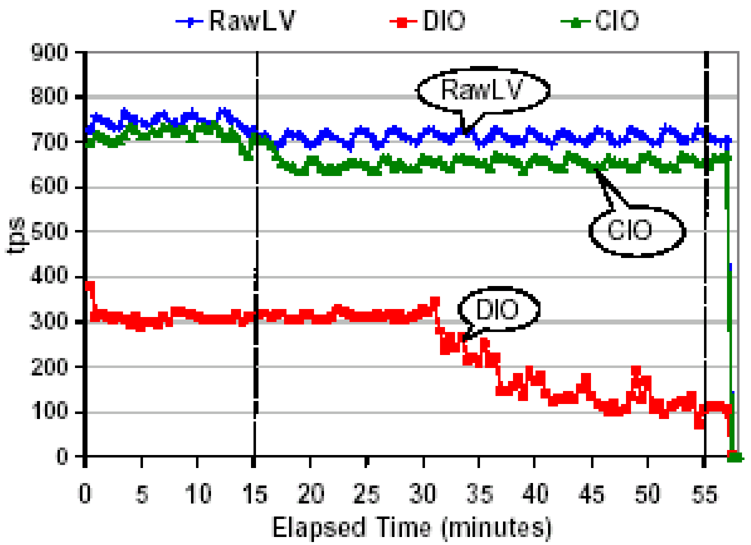
If Oracle data are stored in a Filesystem, some mount option can improve performance :

- **Direct IO (DIO)** – introduced in AIX 4.3.
  - Data is transfered directly from the disk to the application buffer, bypassing the file buffer cache hence avoiding double caching (filesystem cache + Oracle SGA).
  - Emulates a raw-device implementation.
  - To mount a filesystem in DIO

```
$ mount -o dio /data
```

- **Concurrent IO (CIO)** – introduced with jfs2 in AIX 5.2 ML1
  - Implicit use of DIO.
  - **No Inode locking** : Multiple threads can perform reads and writes on the same file at the same time.
  - Performance achieved using CIO is comparable to raw-devices.
  - To mount a filesystem in CIO:

```
$ mount -o cio /data
```



Bench throughput over run duration – higher tps indicates better performance.

IO : Benefits of CIO for Oracle

- **Benefits :**

1. Avoid double caching : Some data are already cache in the Application layer (SGA)
2. Give a faster access to the backend disk and reduce the CPU utilization
3. Disable the inode-lock to allow several threads to read and write the same file (**CIO only**)

- **Restrictions :**

1. Because data transfer is bypassing AIX buffer cache, jfs2 prefetching and write-behind can't be used. These functionnalities can be handled by Oracle.

   ⇒ *(Oracle parameter)* db_file_multiblock_read_count = 8, 16, 32, ... , 128 according to workload

2. When using DIO/CIO, IO requests made by Oracle **must by aligned** with the jfs2 blocksize to avoid a **demoted IO** *(Return to normal IO after a Direct IO Failure)*

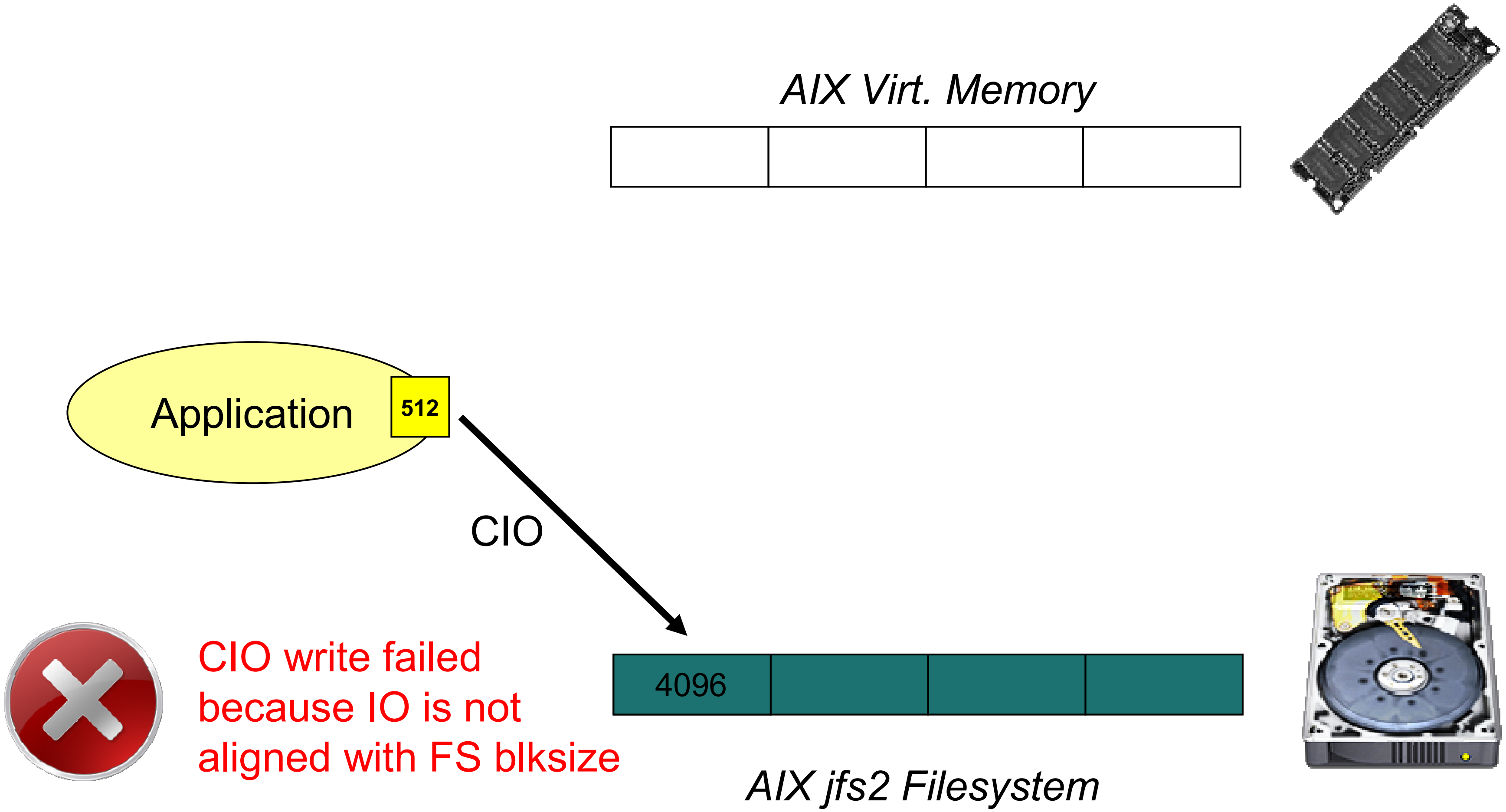
   => When you create a JFS2, use the "**mkfs –o agblksize=XXX**" Option to adapt the FS blocksize with the application needs.

Rule : IO request = n x agblksize

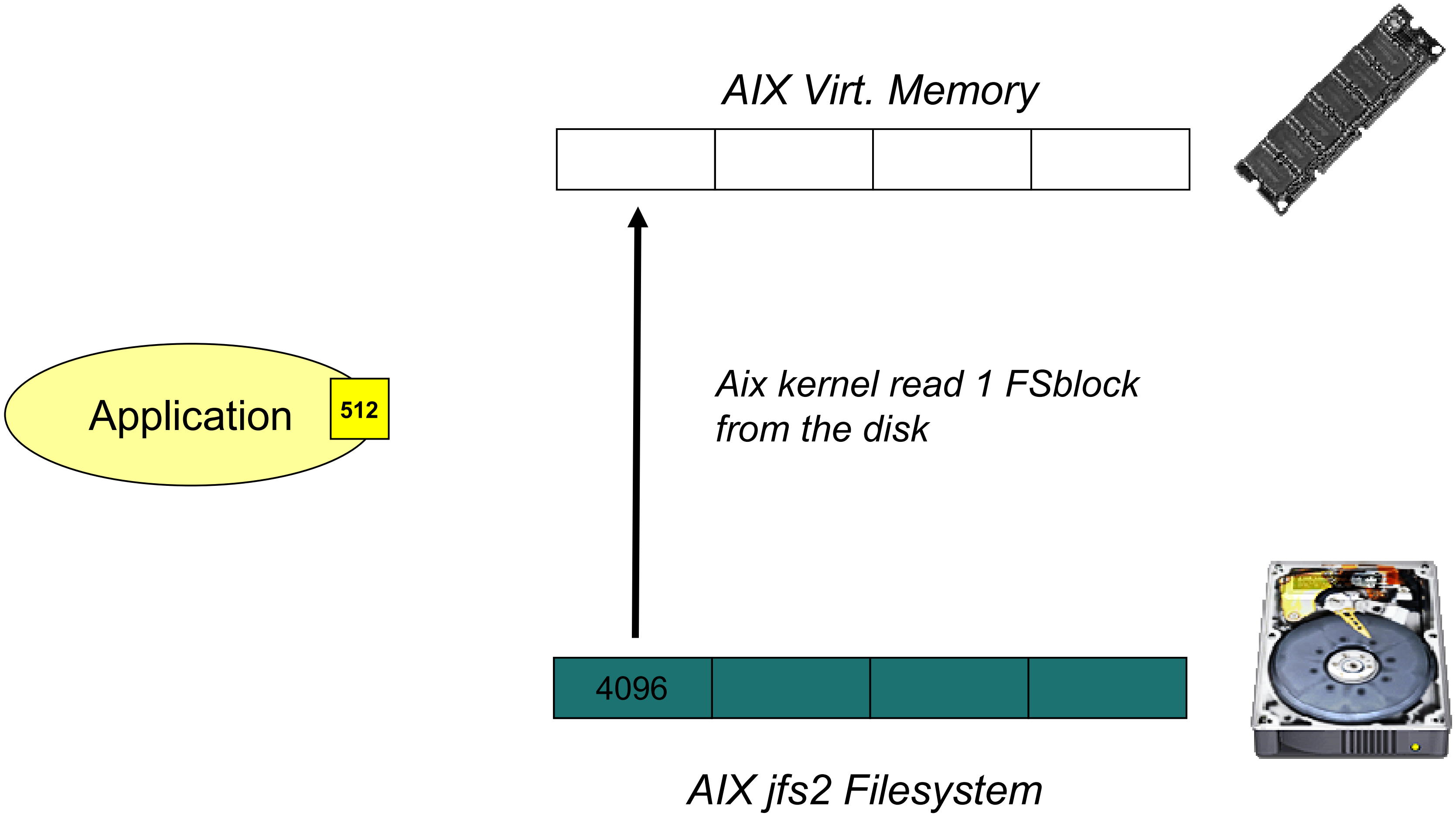
Exemples: if DB blocksize > 4k ; then jfs2 agblksize=4096

Redolog are always written in 512B block; So jfs2 agblksize ***must be 512***

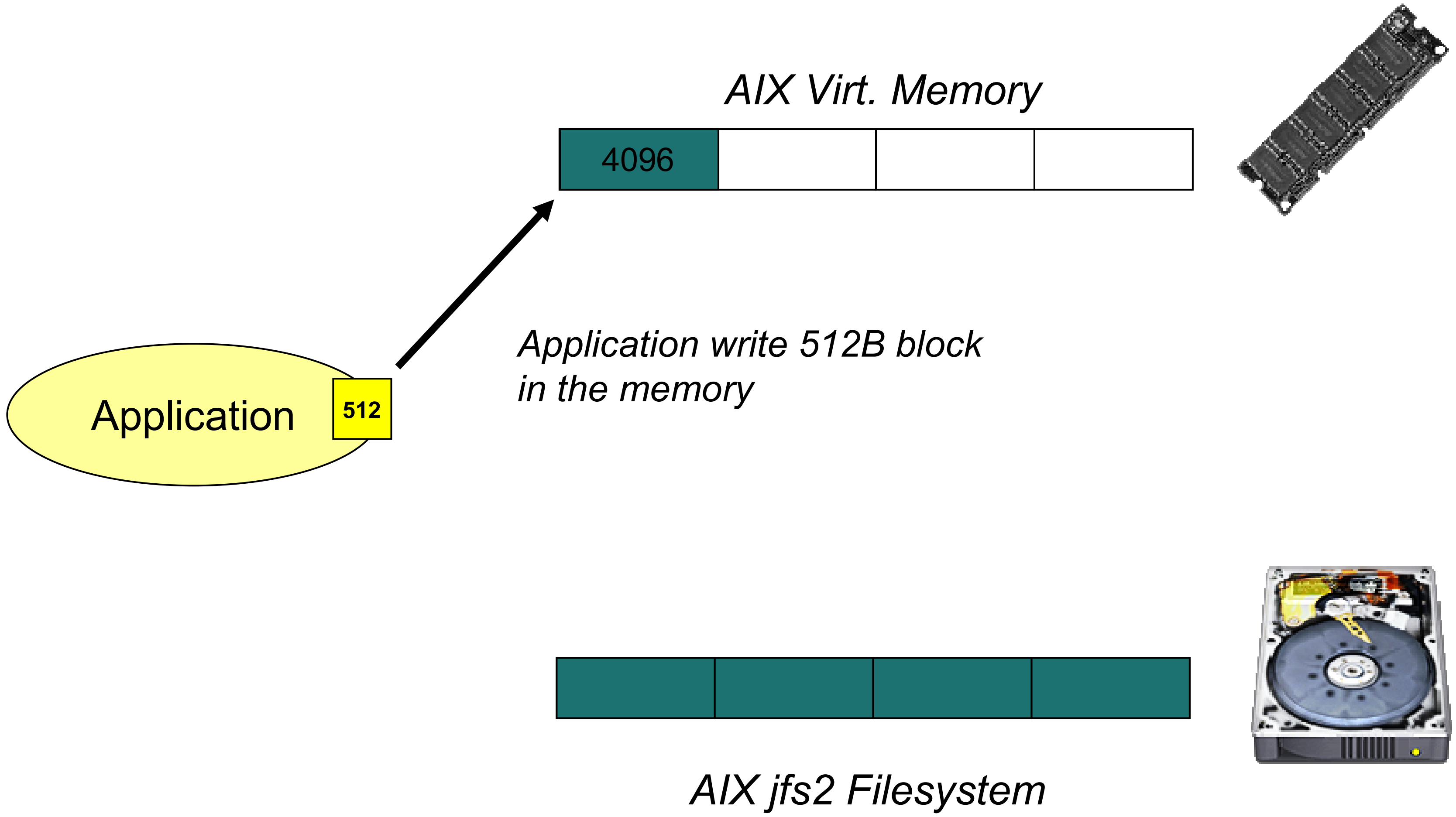
IO : Direct IO demoted – Step 1



IO : Direct IO demoted – Step 2



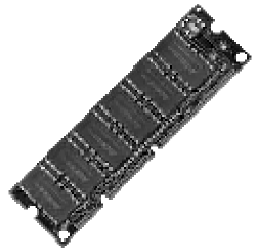
IO : Direct IO demoted – Step 3

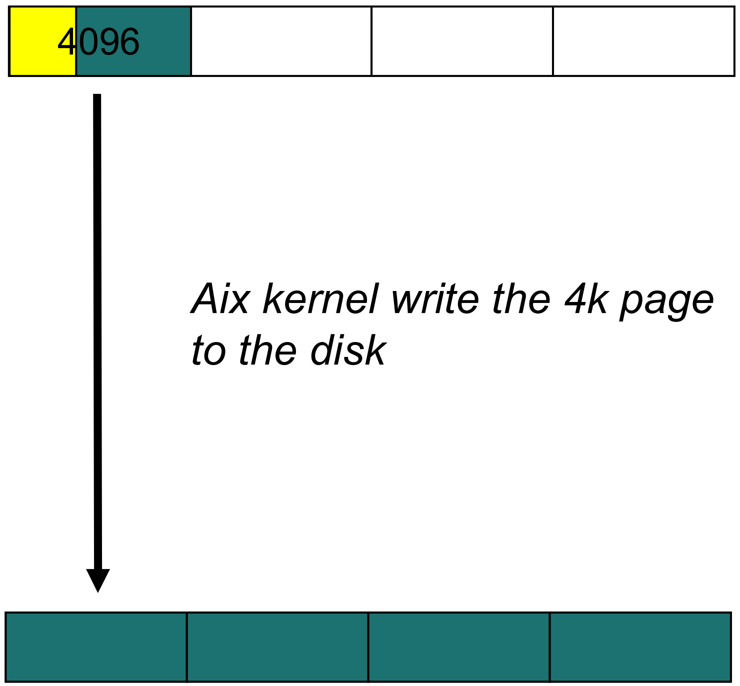


IO : Direct IO demoted – Step 4

Because of the CIO access,the 4k page is remove from memory

AIX Virt. Memory





Application

Aix kernel write the 4k page to the disk

AIX jfs2 Filesystem

To Conclued : demoted io will consume :

- more CPU time (kernel)

- more physical IO : 1 io write = 1 phys io read + 1 phys io write

IO : Direct IO demoted

- Extract from Oracle AWR *(test made in Montpellier with Oracle 10g)*

*<u>Waits on redolog (**with** demoted IO, **FS blk=4k**)</u>*

	Waits%	Time -outs	Total Wait Time (s)	Avg wait (ms)	Waits /txn
log file sync	2,229,324	0.00	62,628	**28**	1.53

*<u>Waits on redolog (**without** demoted IO, **FS blk=512**)</u>*

	Waits%	Time -outs	Total Wait Time (s)	Avg wait (ms)	Waits /txn
log file sync	494,905	0.00	1,073	2	1.00

- **<u>How to detect demoted IO :</u>**

<u>Trace command to check demoted io :</u>

\# trace –aj 59B,59C ; sleep 2 ; trcstop ; trcrpt –o directio.trcrpt

\# grep –i demoted directio.trcrpt

```
{ws1:root}/ # grep demoted /home/seb/trace.out
59B     0.007599931     0.021877     JFS2 IO dio demoted: vp = F10001003868C7F8, mode = 0001, bad = 0002, rc = 0000, rc2 = 0000
59B     0.011683593     0.013113     JFS2 IO dio demoted: vp = F10001003868C7F8, mode = 0001, bad = 0002, rc = 0000, rc2 = 0000
59B     0.015687468     0.013658     JFS2 IO dio demoted: vp = F10001003868C7F8, mode = 0001, bad = 0002, rc = 0000, rc2 = 0000
59B     0.019676449     0.013199     JFS2 IO dio demoted: vp = F10001003868C7F8, mode = 0001, bad = 0002, rc = 0000, rc2 = 0000
59B     0.023682093     0.013472     JFS2 IO dio demoted: vp = F10001003868C7F8, mode = 0001, bad = 0002, rc = 0000, rc2 = 0000
59B     0.027689625     0.015219     JFS2 IO dio demoted: vp = F10001003868C7F8, mode = 0001, bad = 0002, rc = 0000, rc2 = 0000
59B     0.031689187     0.014969     JFS2 IO dio demoted: vp = F10001003868C7F8, mode = 0001, bad = 0002, rc = 0000, rc2 = 0000
59B     0.035691494     0.015452     JFS2 IO dio demoted: vp = F10001003868C7F8, mode = 0001, bad = 0002, rc = 0000, rc2 = 0000
59B     0.039335873     0.018614     JFS2 IO dio demoted: vp = F10001003868C7F8, mode = 0001, bad = 0002, rc = 0000, rc2 = 0000
59B     0.043720031     0.013389     JFS2 IO dio demoted: vp = F10001003868C7F8, mode = 0001, bad = 0002, rc = 0000, rc2 = 0000
```

FS mount options advices

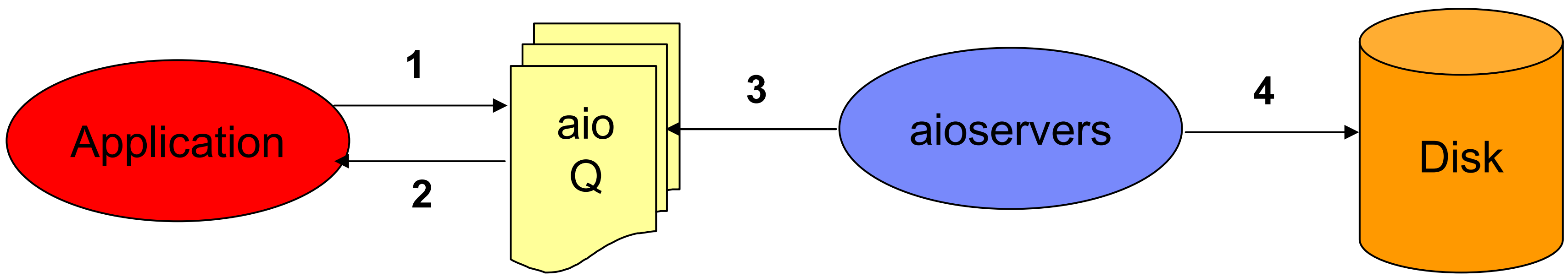
	With standard mount options	With Optimized mount options
Oracle binaries	mount –o rw ***Cached by AIX (fscache)***	mount –o noatime ***Cached by AIX (fscache)*** ***noatime reduce inode modification on read***
Oracle Datafile	mount –o rw *Cached by AIX (fscache)* *Cached by Oracle (SGA)*	mount –o noatime,cio *Cached by Oracle (SGA)*
Oracle Redolog	mount –o rw *Cached by AIX (fscache)* *Cached by Oracle (SGA)*	mount –o noatime,cio **jfs2 agblksize=512** ***(to avoid io demotion)*** *Cached by Oracle (SGA)*
Oracle Archivelog	mount –o rw *Cached by AIX (fscache)*	mount –o noatime,rbrw *Use jfs2 cache, but memory is released after read/write.*
Oracle Control files	mount –o rw ***Cached by AIX (fscache)***	mount –o noatime ***Cached by AIX (fscache)***

Agenda

- CPU
  - Power 7
- Memory
  - AIX VMM tuning
  - Active Memory Expansion
- IO
  - Storage consideration
  - AIX LVM Striping
  - Disk/Fiber Channel driver optimization
  - Virtual Disk/Fiber channel driver optimization
  - AIX mount option
  - Asynchronous IO
- NUMA Optimization
- Other Tips

IO : Asynchronous IO (AIO)

- Allows multiple requests to be sent without to have to wait until the disk subsystem has completed the physical IO.
- Utilization of asynchronous IO is strongly advised whatever the type of file-system and mount option implemented (JFS, JFS2, CIO, DIO).



➢Posix vs Legacy

Since AIX5L V5.3, two types of AIO are now available : Legacy and Posix. For the moment, the Oracle code is using the Legacy AIO servers.

IO : Asynchronous IO (AIO) tuning

- **important tuning parameters :**

check AIO configuration with :

AIX 5.X : lsattr –El aio0

AIX 6.1 & 7.1 : ioo –L | grep aio

- **maxreqs** : size of the Asynchronous queue.
- **minserver** : number of kernel proc. Aioservers to start (AIX 5L system wide).
- **maxserver** : maximum number of aioserver that can be running **per logical CPU**

➢ **Rule of thumb :**

maxservers should be = (10 * <# of disk accessed concurrently>) / # cpu

maxreqs (= a multiple of 4096) should be > 4 * #disks * queue_depth

➢ but only tests allow to set correctly minservers and maxservers

➢ **Monitoring :**

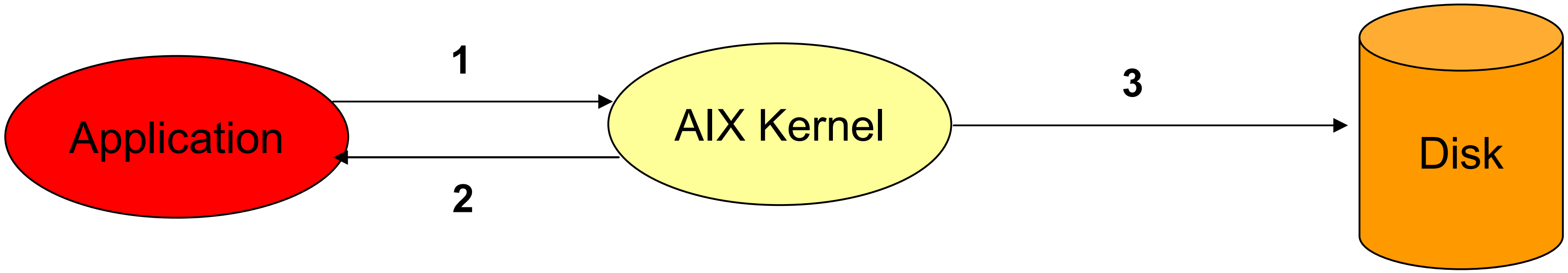
In Oracle's alert.log file, if maxservers set to low : "**Warning: lio_listio returned EAGAIN**" "**Performance degradation may be seen**"

#aio servers used can be monitored via "**ps –k | grep aio | wc –l**" , "**iostat –A**" or **nmon** (option A)

IO : Asynchronous IO (AIO) fastpath

With fsfastpath, IO are queued directly from the application into the LVM layer without any "aioservers kproc" operation.

- Better performance compare to non-fastpath
- No need to tune the min and max aioservers
- No aioservers proc. => "ps –k | grep aio | wc –l" is not relevent, use "iostat –A" instead



- **ASM :**
  - enable asynchronous IO **fastpath. :**

    AIX 5L : chdev -a fastpath=enable -l aio0 *(default since AIX 5.3)*
    AIX 6.1 : ioo –p –o aio_fastpath=1 *(default setting)*
    AIX 7.1 : ioo –p –o aio_fastpath=1 *(default setting* + *restricted tunable)*

- **FS with CIO/DIO and AIX 5.3 TL5+ :**
  - Activate **fsfastpath** (comparable to fast_path but for FS + CIO)

    AIX 5L : adding the following line in /etc/inittab: **aioo:2:once:aioo –o fsfast_path=1**
    AIX 6.1 : ioo –p –o aio_fsfastpath=1 *(default setting)*
    AIX 7.1 : ioo –p –o aio_fsfastpath=1 *(default setting* + *restricted tunable)*

IO : AIO,DIO/CIO & Oracle Parameters

➢ **How to set filesystemio_options parameter**

Possible values

ASYNCH : enables asynchronous I/O on file system files (default)
DIRECTIO : enables direct I/O on file system files (disables AIO)
SETALL : enables both asynchronous and direct I/O on file system files
NONE : disables both asynchronous and direct I/O on file system files

*Since version 10g, Oracle will open data files located on the JFS2 file system with the O_CIO (O_CIO_R with Oracle 11.2.0.2 and AIX 6.1 or Later) option if the filesystemio_options initialization parameter is set to either **directIO** or **setall**.*

Advice : set this parameter to 'ASYNCH', and let the system managed CIO via mount option (see CIO/DIO implementation advices) …

If needed, you can still re-mount an already mounted filesystem to another mount point to have it accessed with different mounting options. Example, your oracle datafiles are on a CIO mounted filesystem, you want to copy them for a cold backup and would prefer to access them with filesystem cache to backup them faster. Then just re-mount this filesystem to another mount point in "rw" mode only.

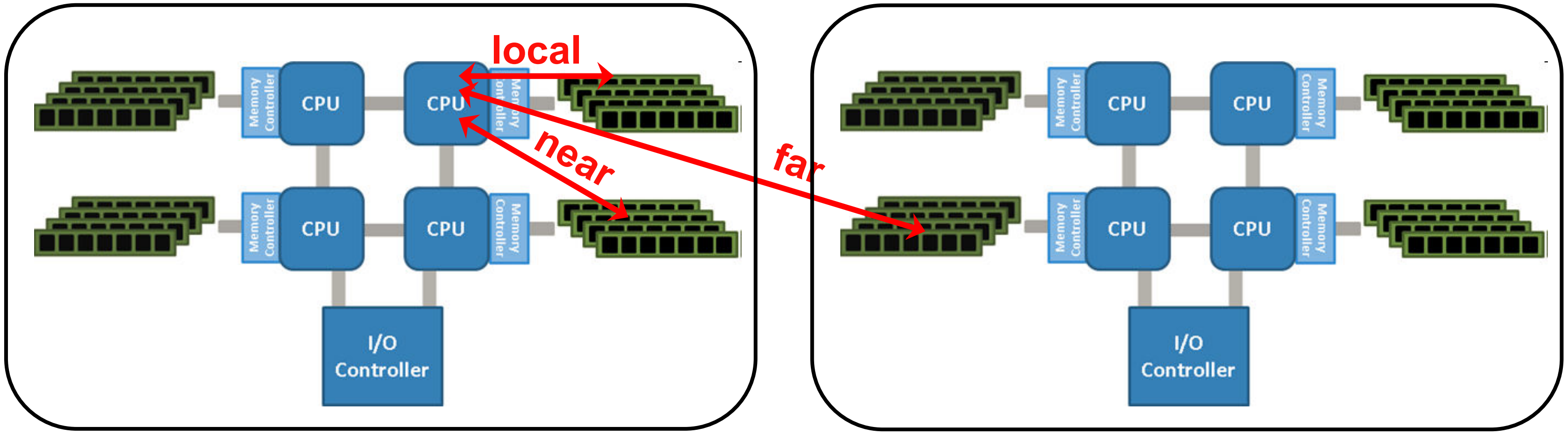
Note : set the disk_asynch_io parameter to 'true' as well

Agenda

- CPU
  - Power 7
- Memory
  - AIX VMM tuning
  - Active Memory Expansion
- IO
  - Storage consideration
  - AIX LVM Striping
  - Disk/Fiber Channel driver optimization
  - Virtual Disk/Fiber channel driver optimization
  - AIX mount option
  - Asynchronous IO
- NUMA Optimization
- Other Tips

NUMA architecture

- NUMA stands for Non Uniform Memory Access.
- It is a computer memory design used in multiprocessors, where the memory access time depends on the memory location relative to a processor.
- Under NUMA, a processor can access its own local memory faster than non-local memory, that is, memory local to another processor or memory shared between processors.



Oracle NUMA feature

Oracle DB NUMA support have been introduced since 1998 on the first NUMA systems. It provides a memory/processes models relying on specific OS features to better perform on this kind of architecture.

On AIX, the NUMA support code has been ported, **default is off** in Oracle 11g.

- _enable_NUMA_support=true is required to enable NUMA features.

- When NUMA enabled Oracle checks for AIX rset named "${ORACLE_SID}/0" at startup.

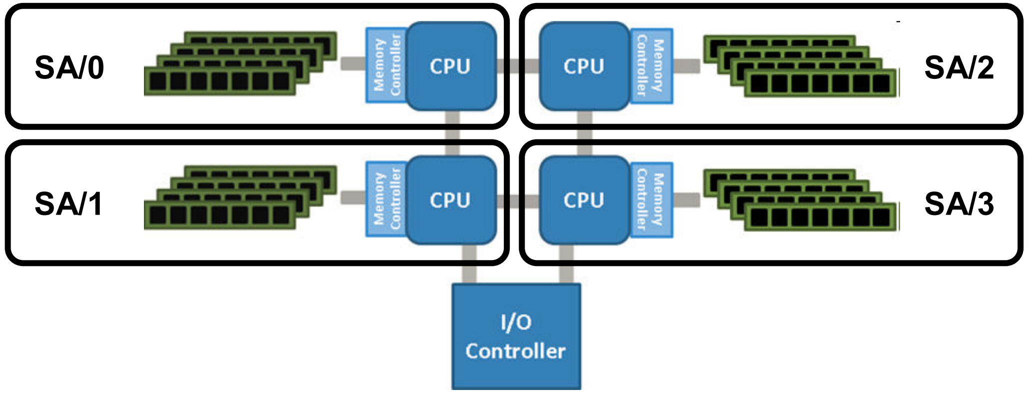
- For now, it is assumed that it will use rsets ${ORACLE_SID}/0, ${ORACLE_SID}/1, ${ORACLE_SID}/2, etc if they exist.

Preparing a system for Oracle NUMA Optimization

The test is done on a POWER7 machine with the following CPU and memory distribution (dedicated LPAR). It has 4 domains with 8 CPU and >27GB each. If the lssrad output shows unevenly distributed domains, fix the problem before proceeding.

- *Listing SRAD (Affinity Domain)*

```
# lssrad -va
REF1    SRAD        MEM         CPU
0
           0    27932.94       0-31
           1    31285.00       32-63
1
           2    29701.00       64-95
           3    29701.00       96-127
```



- We will set up 4 rsets, namely SA/0, SA/1, SA/2, and SA/3, one for each domain.

```
# mkrset -c 0-31 -m 0 SA/0
# mkrset -c 32-63 -m 0 SA/1
# mkrset -c 64-95 -m 0 SA/2
# mkrset -c 96-127 -m 0 SA/3
```

- Required Oracle User Capabilities

```
# lsuser -a capabilities oracle
oracle capabilities=CAP_NUMA_ATTACH,CAP_BYPASS_RAC_VMM,CAP_PROPAGATE
```

- Before starting the DB, let’s set vmo options to cause process private memory to be local.

```
# vmo -o memplace_data=1 -o memplace_stack=1
```

Oracle shared segments differences

- The following messages are found in the *alert log*. It finds the 4 rsets and treats them as NUMA domains.
  LICENSE_MAX_USERS = 0
  SYS auditing is disabled
  NUMA system found and support enabled (4 domains - 32,32,32,32)
  Starting up Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - 64bit Production

- The shared memory segments. There are total of 7, one of which owned by ASM. The SA instance has 6 shared memory segments instead of 1.

ipcs -ma|grep oracle

Without NUMA optimization

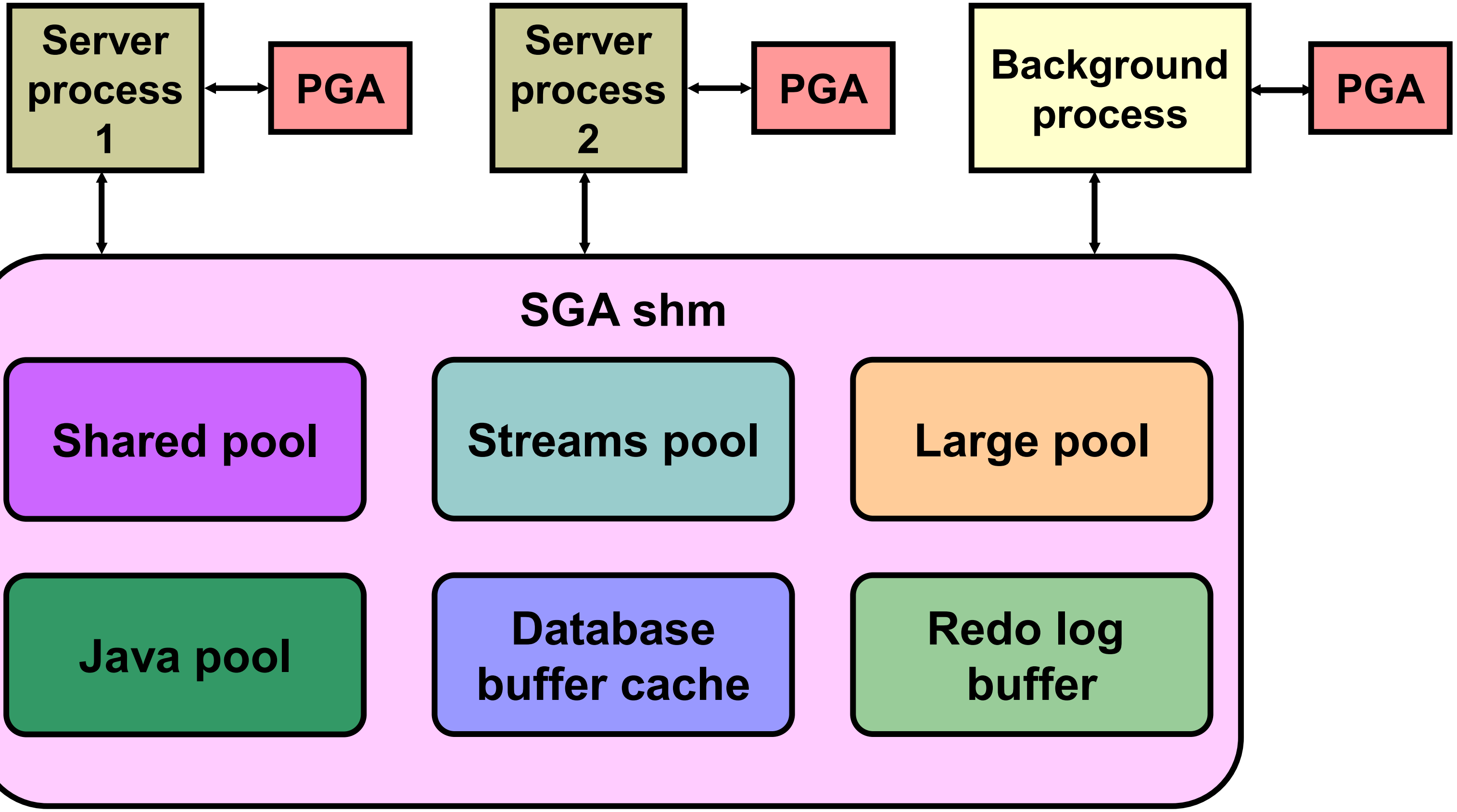
```
– m   2097156 0x8c524c30 --rw-rw----  oracle  dba  oracle  dba  31 285220864 13369718 23987126 23:15:57 23:15:57 12:41:22
– m 159383709 0x3b5a2ebc --rw-rw----  oracle  dba  oracle  dba  59 54089760768 17105120 23331318 23:16:13 23:16:13 23:15:45
```

ipcs -ma|grep oracle

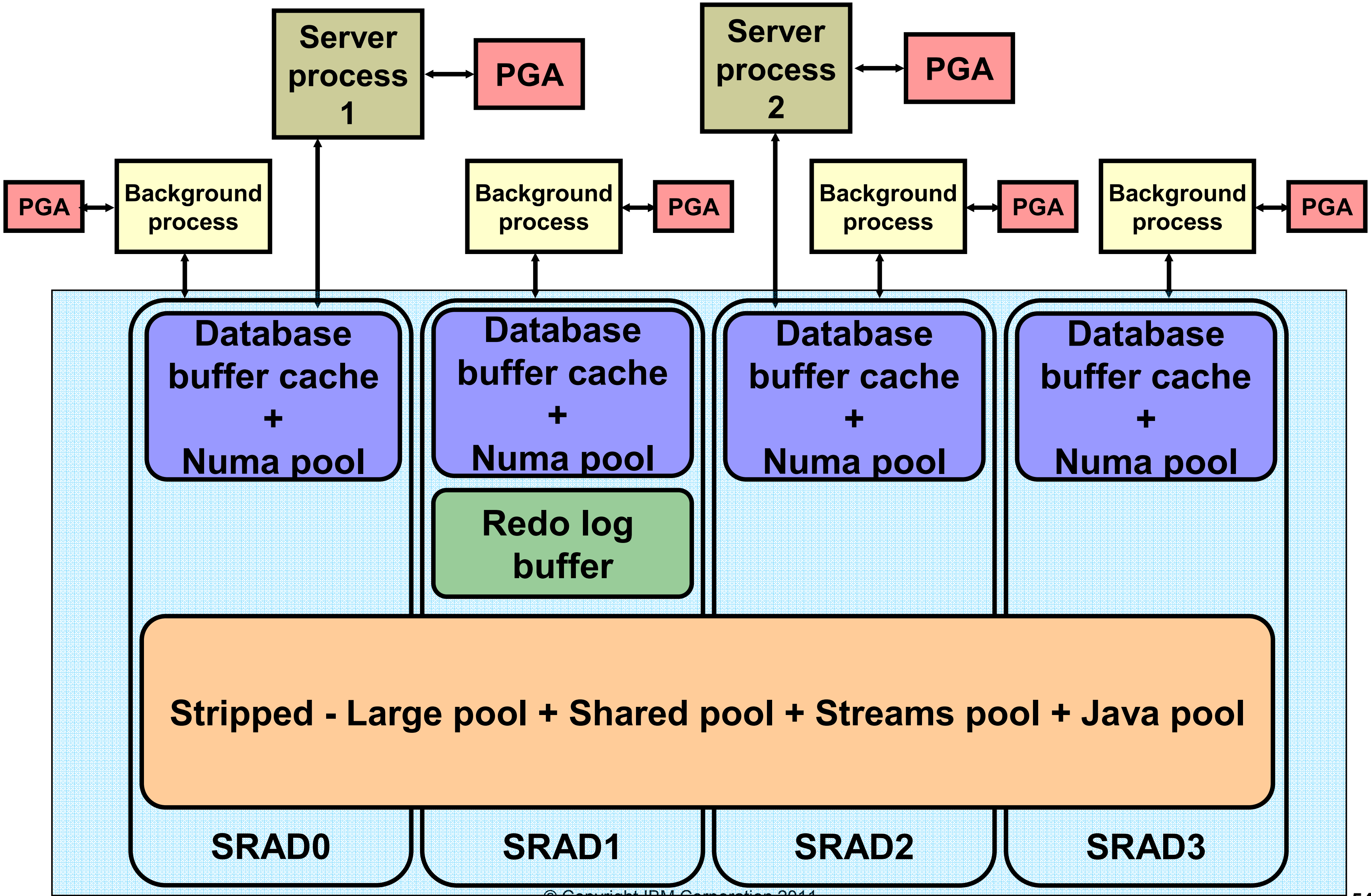
With NUMA optimization

```
– m   2097156 0x8c524c30 --rw-rw----  oracle  dba  oracle  dba  29 285220864 13369718 7405688 23:27:32 23:32:38 12:41:22
– m 702545926  00000000 --rw-rw----  oracle  dba  oracle  dba  59 2952790016 23987134 23920648 23:32:42 23:32:42 23:27:21
– m 549453831  00000000 --rw-rw----  oracle  dba  oracle  dba  59 2952790016 23987134 23920648 23:32:42 23:32:42 23:27:21
– m 365953095 0x3b5a2ebc --rw-rw----  oracle  dba  oracle  dba  59     20480 23987134 23920648 23:32:42 23:32:42 23:27:21
– m 1055916188  00000000 --rw-rw----  oracle  dba  oracle  dba  59 3087007744 23987134 23920648 23:32:42 23:32:42 23:27:21
– m 161480861  00000000 --rw-rw----  oracle  dba  oracle  dba  59 42144366592 23987134 23920648 23:32:42 23:32:42 23:27:21
– m 333447326  00000000 --rw-rw----  oracle  dba  oracle  dba  59 2952790016 23987134 23920648 23:32:42 23:32:42 23:27:21
```

Standard Oracle Memory Structures



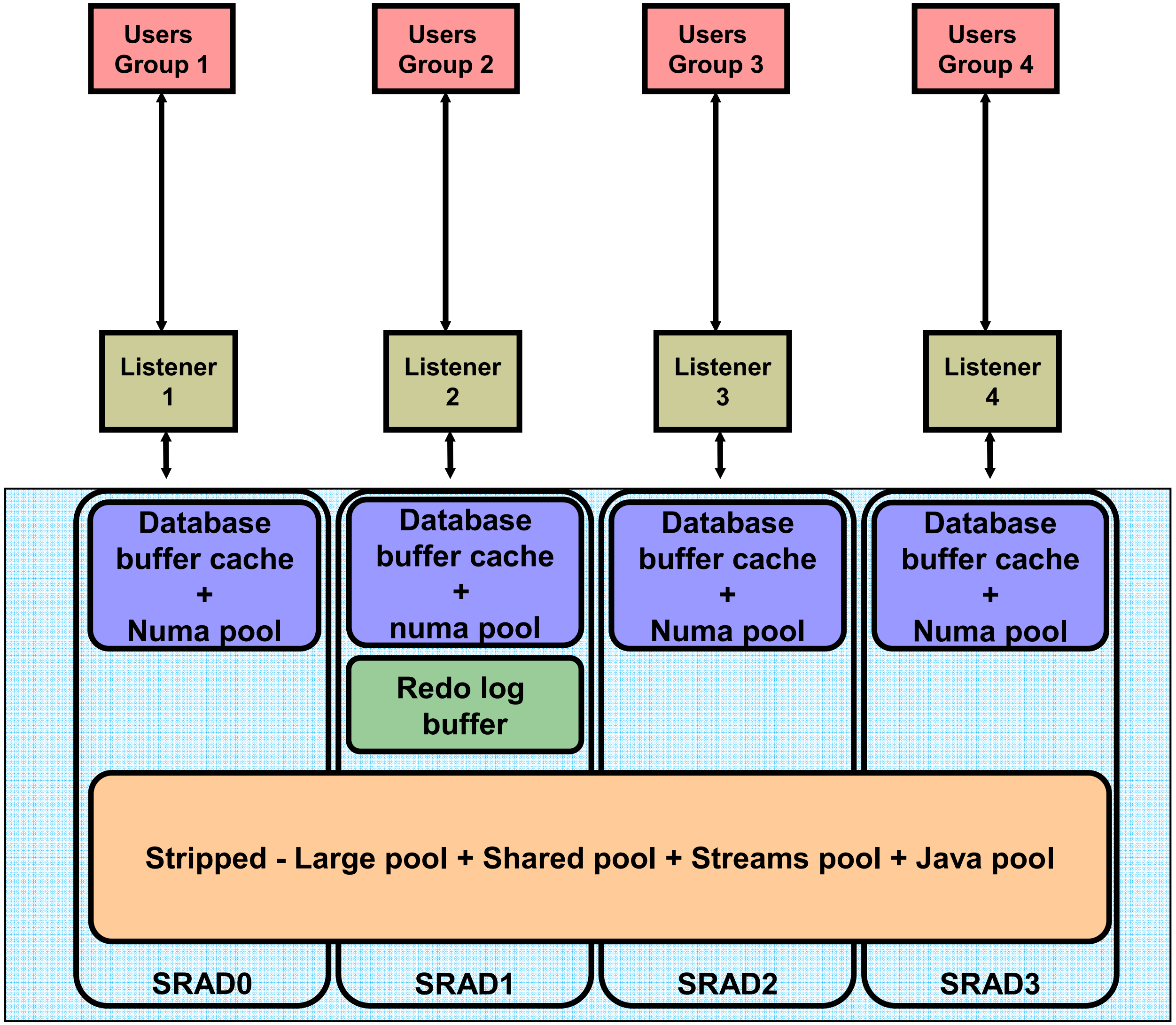
NUMA Oracle Memory Structures



Affinitizing User Connections

- If Oracle shadow processes are allowed to migrate across domains, the benefit of NUMA-enabling Oracle will be lost. Therefore, arrangements need to be made to affinitize the user connections.
- For network connections, multiple listeners can be arranged with each listener affinitized to a different domain. The Oracle shadow processes are children of the individual listeners and inherit the affinity from the listener.
- For local connections, the client process can be affinitized to the desired domain/rset. These connections do not go through any listener, and the shadows are children of the individual clients and inherit the affinity from the client.

Affinitizing User Connections



A Simple Performance Test

- Four Oracle users each having it’s own schema and tables are defined. The 4 schemas are identical except the name.
- Each user connection performs some query using random numbers as keys and repeats the operation until the end of the test.
- The DB cache is big enough to hold the entirety of all the 4 schemas. therefore, it is an in-memory test.
- All test cases are the same, except domain-attachment control. Each test runs a total of 256 connections, 64 of each oracle user.

Relative Performance

	Case 0	Case 1	Case 2
NUMA config	No	Yes	Yes
Connection affinity	No	RoundRobin*	Partitioned**
Relative performance	100%	112%	144%

* RoundRobin = 16 connections of each oracle user run in the each domain;

** Partitioned = 64 connections of 1 oracle user run in each domain.

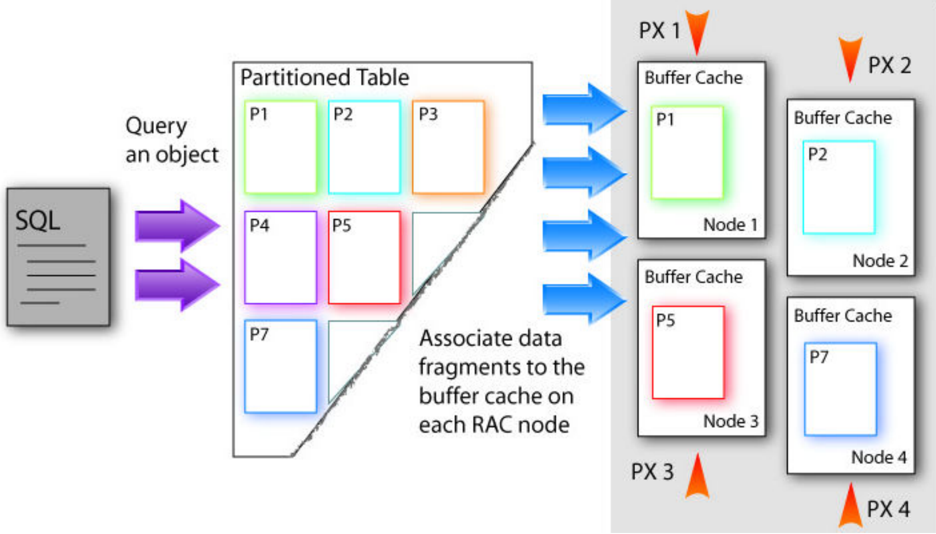
the relative performance shown applies only to this individual test, and can vary widely with different workloads.

Agenda

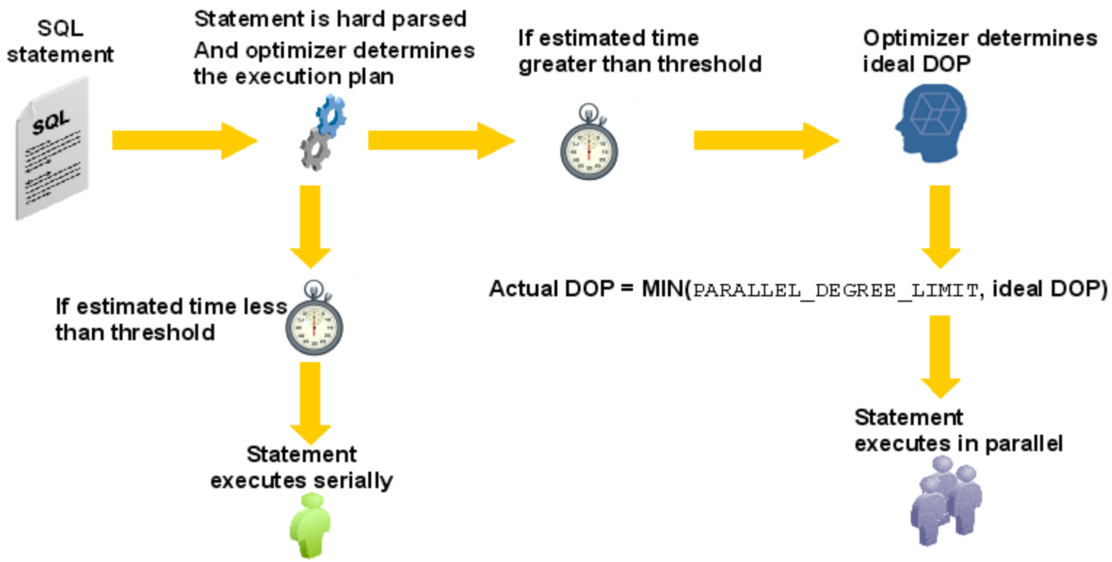
- CPU
  - Power 7
- Memory
  - AIX VMM tuning
  - Active Memory Expansion
- IO
  - Storage consideration
  - AIX LVM Striping
  - Disk/Fiber Channel driver optimization
  - Virtual Disk/Fiber channel driver optimization
  - AIX mount option
  - Asynchronous IO
- NUMA Optimization
- Other Tips

In-Memory Parallel Execution

- When using parallelism, we recommend using In-Memory Parallel Execution.



- In Memory Parallel Execution a subpart of SGA



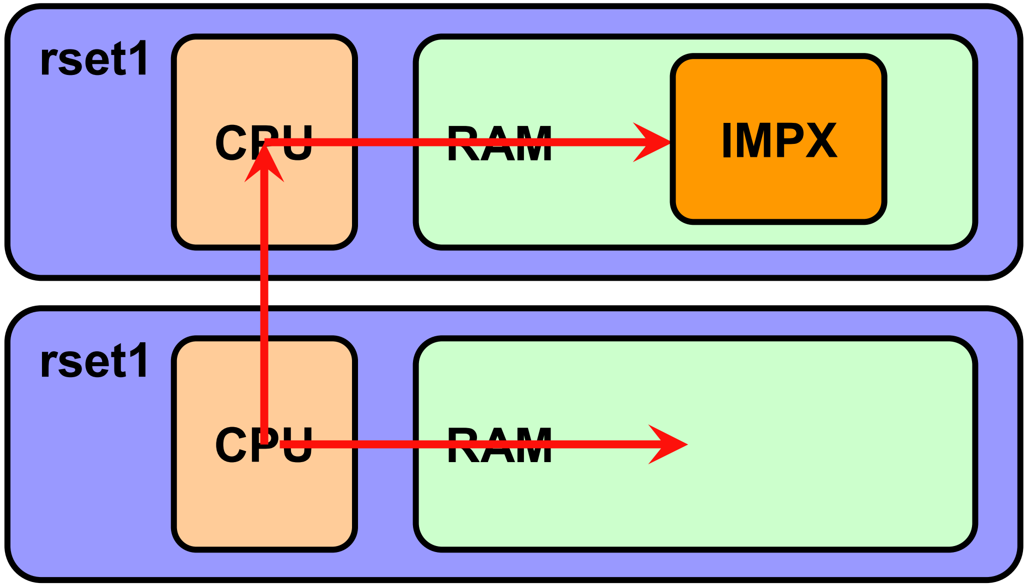
In-Memory Parallel Execution

- Oracle Configuration
  - Version 11gR2
  - parallel_degree_policy = auto in spfile
  - optimizer_feature_enable at the exact version number of Oracle Engine
  - Calibrate IO through DBMS_RESOURCE_MANAGER.CALIBRATE_IO when there is no activity on the database.
  - Update Statistics

- Other Oracle configuration
  - parallel_mintime_threshold (default 30s)
  - Parallel_min_servers
  - Parallel_max_servers
  - Parallel_degree_limit (default cpu)

In-Memory Parallel Execution and NUMA

- Some benchmarks showed performance improvement with In-Memory PX when deactivating NUMA, up to x5 for some queries
- Hypothesis (still under investigation)
  - In-Memory PX uses a subpart of SGA which cannot be split and then is in only one rset.
  - Either loose time when CPU and RAM are not aligned
  - Or loose time when IMPX memory is moved from one rset to the other one



Various Findings

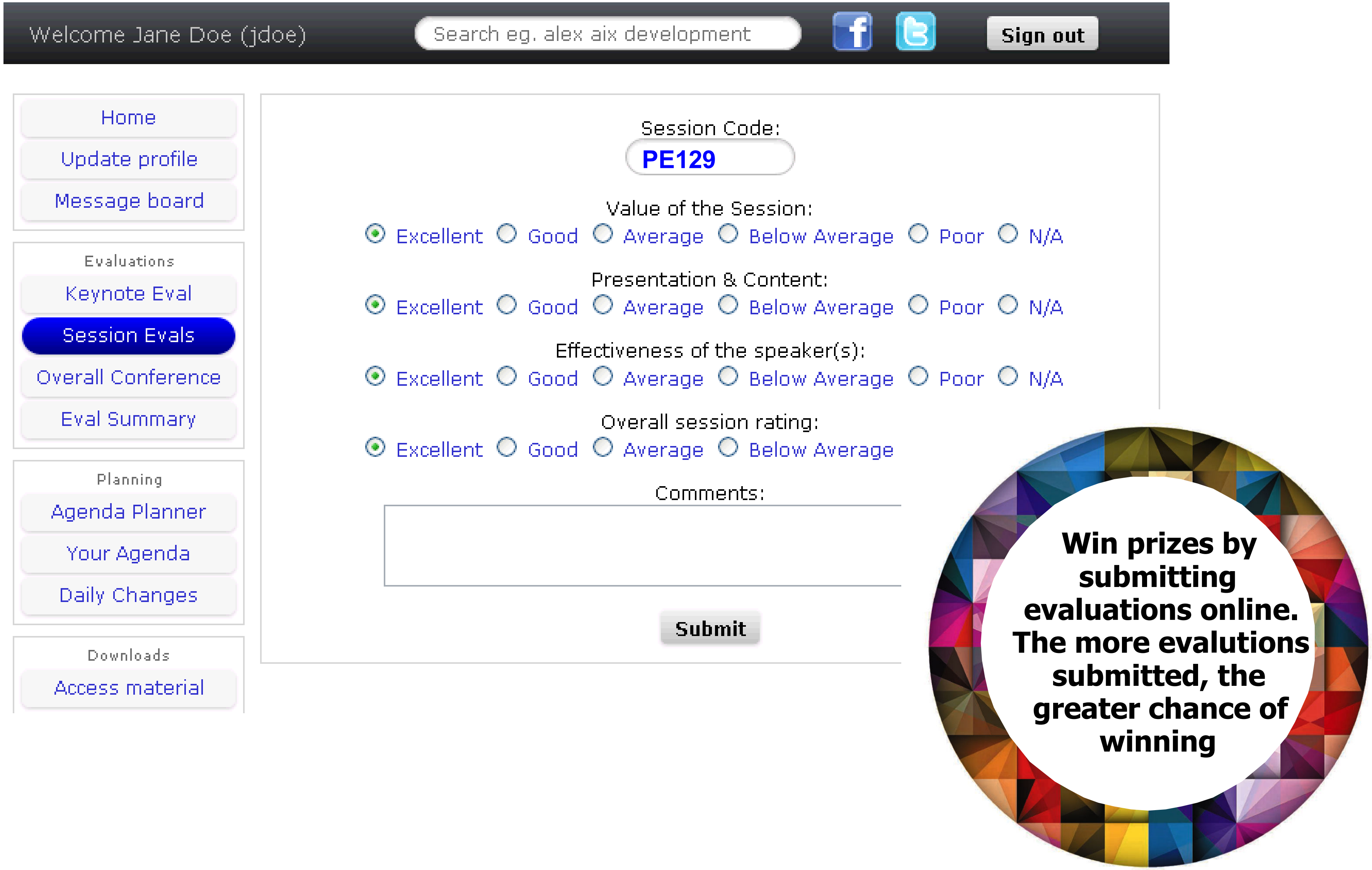
- Slow DLPAR and SMTCTL operations when DB Console is running
  - Example without DB console :
    - Add 1 core : 3.4s
    - Remove 1 core : 2.3s
  - Elapsed time with DB console running
    - Add 1 core : 2min 15.8s
    - Remove 1 core : 2min 15.1s

- Slow access to time operations (such as sysdate) in Oracle when using Olson TZ on AIX 6.1
  - Workaround is to set TZ using POSIX values
  - Example
    - Olson: TZ=Europe/Berlin
    - POSIX: TZ=MET-1MST-2,M3.5.0/02:00,M10.5.0/03:00

- Database performance progressively degrades over time until the instance is restarted.
  - Issue is exposed by a change in Rdbms 11.2.0.3
  - Triggered by large number of connections + Terabyte segments
  - Fixed in AIX 7.1 TL1 SP5
  - Workaround for ealier versions : disable Terabyte segment
    - “vmo -r -o shm_1tb_unsh_enable=0” + reboot

Session Evaluations

IBM Power Systems Technical University Dublin 2012



Power Benchmark & Proof of Concept

Our customer benchmark center is the place to validate the proposed IBM solution in a simulated production environment or to focus on specific IBM Power / AIX Technologies

hands on

- Standard benchmarks
  - Dedicated Infrastructure
  - Dedicated technical support
- Light Benchmarks
  - Mutualized infrastructure
  - Second level support
- Proof of Technology workshops
  - On-demand education sessions

IBM Montpellier

Products and Solutions Support Center

Request a benchmark :
http://d27db001.rchland.ibm.com/b_dir/bcdcweb.nsf/request?OpenForm#new

Questions ?

ronan.bourlier@fr.ibm.com

loic.fura@fr.ibm.com

Thank You

IBM Power Systems Technical University Dublin 2012

धन्यवाद
Hindi

Cám ơn
Vietnam

ขอบคุณ
Thai

Dankie
Afrikaans

Спасибо
Russian

多謝
Traditional Chinese

Gracias
Spanish

Siyabonga
Zulu

شكرا
Arabic

Danke
German

Obrigado
Brazilian Portuguese

Grazie
Italian

Merci
French

多谢
Simplified Chinese

நன்றி
Tamil Tamil

ありがとうございました
Japanese

감사합니다
Korean

Dziekuje
Polish

Tak
Danish / Norwegian

Tack
Swedish