

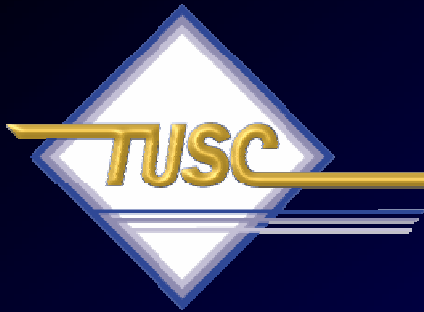
# Statspack, AWR Tuning & Block Tuning

## SOUG – 2007; Tampa, Florida



**Rich Niemiec, TUSC**

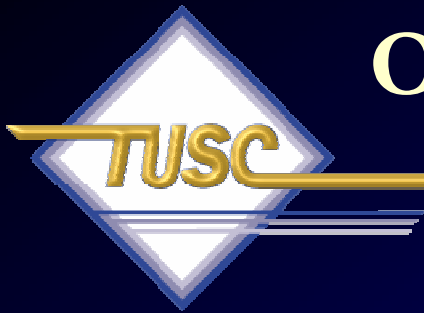
(Thanks: Jay Rossiter, John Kennedy, Julie Wong, Arsalan Farooq, Benoit Dageville, Tom Kyte, Brad Brown & Joe Trezzo)



# Audience Knowledge

- Oracle8i Experience ?
- Oracle9i Experience ?
- Oracle9i RAC Experience?
- Oracle10g Experience?
- Goals
  - Overview of Tuning Statspack/AWR
  - Focus on a few nice features of Oracle 10g
- Non-Goals
  - Learn ALL aspects of Tuning Oracle





# Overview

- Statspack, Tools & Scripts that you can still use
  - Top Waits
  - Load Profile
  - Latch Waits
  - Top SQL
  - Instance Activity
  - File I/O
- The Future OEM & ADDM
- Helpful V\$/X\$
- Summary



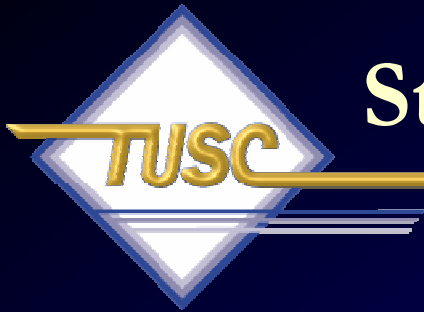


# Tuning in General

- Both an Art and a Science
- Exceptions often rule the day...Not a “one size fits all”
- Hardware & Architecture must be right for your application or difficult to succeed.
- Statspack & Enterprise Manager (also 3<sup>rd</sup> party products) are best for simple tuning and ongoing maintenance.
- V\$/X\$ are best for drilling deep into problems
- **Enterprise Manager 10g will radically change things.**

# Statspack - Still nice; Some new 10g features



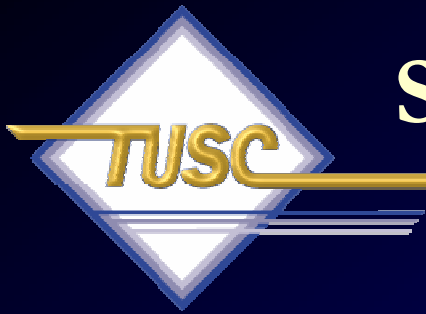


# Statspack – Check Regularly

---

1. Top 5 wait events
2. Load Profile
3. Instance Efficiency Hit Ratios
4. Wait Events / **Wait Event Histograms**
5. Latch Waits
6. Top SQL
7. Instance Activity / **Time Model Stats / O/S Stats**
8. File I/O / **File Read Histogram / Undo Stats**
9. Memory Allocation
10. Undo





## Statspack – Miscellaneous notes

- SQL> @spcreate (system/manager as SYSDBA) <1m
- SQL> alter user PERFSTAT account lock; (also unlock)
- SQL> EXECUTE STATSPACK.MODIFY\_STATSPACK\_PARAMETER –  
(i\_snap\_level=>5, i\_buffer\_gets\_th=>100000, - i\_modify\_parameter=>'true');
- SQL> execute STATSPACK.SNAP; (do this for start/end) PL/SQL  
procedure successfully completed.  
*(spauto.sql – Setup a snap on the hour every hour)*
- SQL> @ORACLE\_HOME/rdbms/admin/spreport (also sprepsql)



# Statspack – Header Information



DB Name	DB Id	Instance	Inst Num	Release	Cluster	Host
ORA92	968233682	P10	1	9.2.0.4.0	NO	RJN1

	Snap Id	Snap Time	Sessions	Curs/Sess	Comment
Begin Snap:	458	28-Nov-03 00:15:00	814	179.1	
End Snap:	505	28-Nov-03 23:45:00	816	211.4	
Elapsed:		1,410.00 (mins)			

Cache Sizes (end)

~~~~~

|                   |         |                 |        |
|-------------------|---------|-----------------|--------|
| Buffer Cache:     | 32,773M | Std Block Size: | 8K     |
| Shared Pool Size: | 2,048M  | Log Buffer:     | 1,024K |





# Statspack – Header Information



| DB Name | DB Id      | Instance | Inst Num | Release    | RAC | Host       |
|---------|------------|----------|----------|------------|-----|------------|
| ORCL    | 1050469182 | orcl     | 1        | 10.1.0.2.0 | NO  | RJNMOBILE5 |

|             | Snap Id | Snap Time          | Sessions | Curs/Sess | Comment |
|-------------|---------|--------------------|----------|-----------|---------|
| Begin Snap: | 1       | 12-Apr-04 11:36:02 | 14       | 5.4       |         |
| End Snap:   | 2       | 12-Apr-04 12:25:32 | 18       | 5.6       |         |
| Elapsed:    |         | 49.50 (mins)       |          |           |         |

Cache Sizes (end)

~~~~~

Buffer Cache:	24M	Std Block Size:	8K
Shared Pool Size:	80M	Log Buffer:	256K



# Statspack – Header Information



- Ensure that you're running for the right instance.
- Check the start/end times
- Check the cache sizes
  - Could have been changed during the run
  - Last page will tell you more on starting/ending values of initialization parameters



# Statspack – Load Profile



## Load Profile

~~~~~

|                  | Per Second   | Per Transaction |
|------------------|--------------|-----------------|
|                  | -----        | -----           |
| Redo size:       | 1,409,245.79 | 36,596.21       |
| Logical reads:   | 157,472.47   | 4,089.35        |
| Block changes:   | 4,061.85     | 105.48          |
| Physical reads:  | 5,965.05     | 154.90          |
| Physical writes: | 587.76       | 15.26           |
| User calls:      | 5,922.08     | 153.79          |
| Parses:          | 92.11        | 2.39            |
| Hard parses:     | 0.17         | 0.00            |
| Sorts:           | 93.88        | 2.44            |
| Logons:          | 0.25         | 0.01            |
| Executes:        | 5,686.76     | 147.68          |
| Transactions:    | 38.51        |                 |

|                             |      |                   |        |
|-----------------------------|------|-------------------|--------|
| % Blocks changed per Read:  | 2.58 | Recursive Call %: | 2.38   |
| Rollback per transaction %: | 1.22 | Rows per Sort:    | 114.10 |



# AWR – Load Profile



## Report Summary

### Cache Sizes

|                   | Begin   | End     |                 |         |
|-------------------|---------|---------|-----------------|---------|
| Buffer Cache:     | 10,240M | 10,240M | Std Block Size: | 8K      |
| Shared Pool Size: | 1,264M  | 1,264M  | Log Buffer:     | 36,744K |

### Load Profile

|                  | Per Second    | Per Transaction |
|------------------|---------------|-----------------|
| Redo size:       | 37,741,608.27 | 5,236,744.44    |
| Logical reads:   | 239,964.89    | 33,295.74       |
| Block changes:   | 137,275.83    | 19,047.37       |
| Physical reads:  | 1.84          | 0.25            |
| Physical writes: | 4,708.71      | 653.35          |
| User calls:      | 42.00         | 5.83            |
| Parses:          | 24.05         | 3.34            |
| Hard parses:     | 0.04          | 0.01            |
| Sorts:           | 0.34          | 0.05            |
| Logons:          | 0.71          | 0.10            |
| Executes:        | 31.85         | 4.42            |
| Transactions:    | 7.21          |                 |

|                             |       |                   |        |
|-----------------------------|-------|-------------------|--------|
| % Blocks changed per Read:  | 57.21 | Recursive Call %: | 78.22  |
| Rollback per transaction %: | 25.00 | Rows per Sort:    | 153.62 |



# Statspack – Instance Efficiency



## Instance Efficiency Percentages (Target 100%)

~~~~~

Buffer Nowait %:	99.08	Redo NoWait %:	99.86
Buffer Hit %:	96.39	In-memory Sort %:	99.95
Library Hit %:	100.00	Soft Parse %:	99.82
Execute to Parse %:	98.38	Latch Hit %:	99.64
Parse CPU to Parse Elapsed %:	85.11	% Non-Parse CPU:	99.86

# Statspack - Instance Efficiency

## Things to look for...



- Buffer NoWait % of less than 99 percent. This is ratio of hits on a request for a specific buffer where the buffer was immediately available in memory. If the ratio is low, then could be a (hot) block(s) being contended for that should be found in the Buffer Wait Section.

# Statspack - Instance Efficiency

## Things to look for...



- Buffer Hit % of less than 95 percent. This is the ratio of hits on a request for a specific buffer and the buffer was in memory instead of needing to do a physical I/O.
  - When this varies greatly one day to the next, further investigation should be done as to the cause.
  - If you have unselective indexes that are frequently accessed, it will drive your hit ratio higher, which can be misleading indication of good performance.
  - When you effectively tune your SQL and have effective indexes on your entire system, this issue is not encountered as frequently and the hit ratio is a better performance indicator.



# Statspack - Instance Efficiency

## Things to look for...



- Library Hit % of less than 95 percent. A lower library hit ratio usually indicates that SQL is being pushed out of the shared pool early (could be due to a shared pool that is too small).
  - A lower ratio could also indicate that bind variables are not used or some other issue is causing SQL not to be reused (in which case a smaller shared pool may only be a band-aid that will potentially fix a library latch problem which may result).
  - You must fix the problem (use bind variables or `CURSOR_SHARING`) and then appropriately size the shared pool. I'll discuss this further when we get to latch issues.

# Statspack - Instance Efficiency

## Things to look for...



- In-Memory Sort % of less than 95 percent in OLTP. In an OLTP system, you really don't want to do disk sorts. Setting the PGA\_AGGREGATE\_TARGET (or SORT\_AREA\_SIZE) initialization parameter effectively will eliminate this problem.
- Latch Hit % of less than 99 percent is usually a big problem. Finding the specific latch will lead you to solving this issue. More in the Latch Wait section.



# AWR - Instance Efficiency

## Instance Efficiency Percentages (Target 100%)

Buffer Nowait %:	99.31	Redo NoWait %:	99.99
Buffer Hit %:	100.00	In-memory Sort %:	100.00
Library Hit %:	99.94	Soft Parse %:	99.82
Execute to Parse %:	24.50	Latch Hit %:	94.65
Parse CPU to Parse Elapsed %:	91.87	% Non-Parse CPU:	99.96

## Shared Pool Statistics

	Begin	End
Memory Usage %:	68.02	68.20
% SQL with executions>1:	81.94	81.36
% Memory for SQL w/exec>1:	77.24	74.72

## Top 5 Timed Events

Event	Waits	Time(s)	Avg Wait(ms)	% Total Call Time	Wait Class
log buffer space	52,521	8,851	169	35.8	Configuration
CPU time		7,636		30.9	
log file sync	7,362	5,122	696	20.7	Commit
buffer busy waits	1,564,508	2,145	1	8.7	Concurrency
log file sequential read	35,171	701	20	2.8	System I/O



# What are you Waiting on?





# Statspack – Top Wait Events



## Top 5 Timed Events

~~~~~

| Event                                      | Waits       | Time (s)  | % Total<br>Ela Time |
|--------------------------------------------|-------------|-----------|---------------------|
| -----                                      | -----       | -----     | -----               |
| db file sequential read                    | 399,394,399 | 2,562,115 | 52.26               |
| CPU time                                   |             | 960,825   | 19.60               |
| buffer busy waits                          | 122,302,412 | 540,757   | 11.03               |
| PL/SQL lock timer                          | 4,077       | 243,056   | 4.96                |
| log file switch<br>(checkpoint incomplete) | 188,701     | 187,648   | 3.83                |

# Statspack - Top Wait Events

## Things to look for...



### Wait Problem

Sequential Read

Scattered Read

Free Buffer

Buffer Busy

### Potential Fix

Indicates many index reads – tune the code (especially joins); Faster I/O

Indicates many full table scans – tune the code; cache small tables; Faster I/O

Increase the DB\_CACHE\_SIZE; shorten the checkpoint; tune the code to get less dirty blocks, faster I/O, use multiple DBWR's.

Segment Header – Add freelists (if inserts) or freelist groups (esp. RAC). Use ASSM.



# Statspack - Top Wait Events

## Things to look for...



### Wait Problem

Buffer Busy

### Potential Fix

Data Block – Separate ‘hot’ data; potentially use reverse key indexes; fix queries to reduce the blocks popularity, use smaller blocks, I/O, Increase initrans and/or maxtrans (this one’s debatable) Reduce records per block.

Buffer Busy

Undo Header – Add rollback segments or increase size of segment area (auto undo)

Buffer Busy

Undo block – Commit more (not too much) Larger rollback segments/area. Try to fix the SQL.





# Statspack - Top Wait Events

## Things to look for...



### Wait Problem

### Potential Fix

Enqueue - ST

Use LMT's or pre-allocate large extents

Enqueue - HW

Pre-allocate extents above HW (high water mark.)

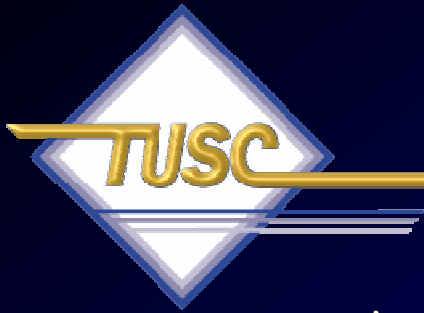
Enqueue – TX

Increase initrans and/or maxtrans (TX4) on (transaction) the table or index. Fix locking issues if TX6. Bitmap (TX4) & Duplicates in Index (TX4).

Enqueue - TM  
(trans. mgmt.)

Index foreign keys; Check application locking of tables. DML Locks.

# Why INITRANS Matter!



## Transactions Moving through Oracle: *ITL* & *Undo Blocks*

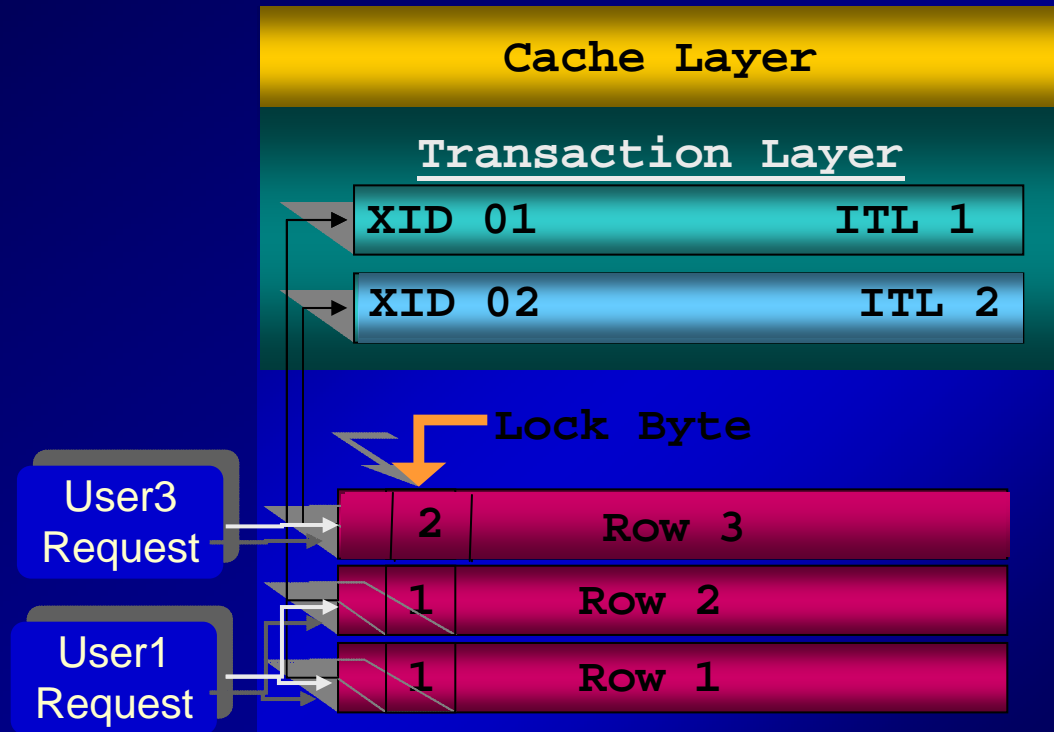




# User 1 – Updates Row# 1&2

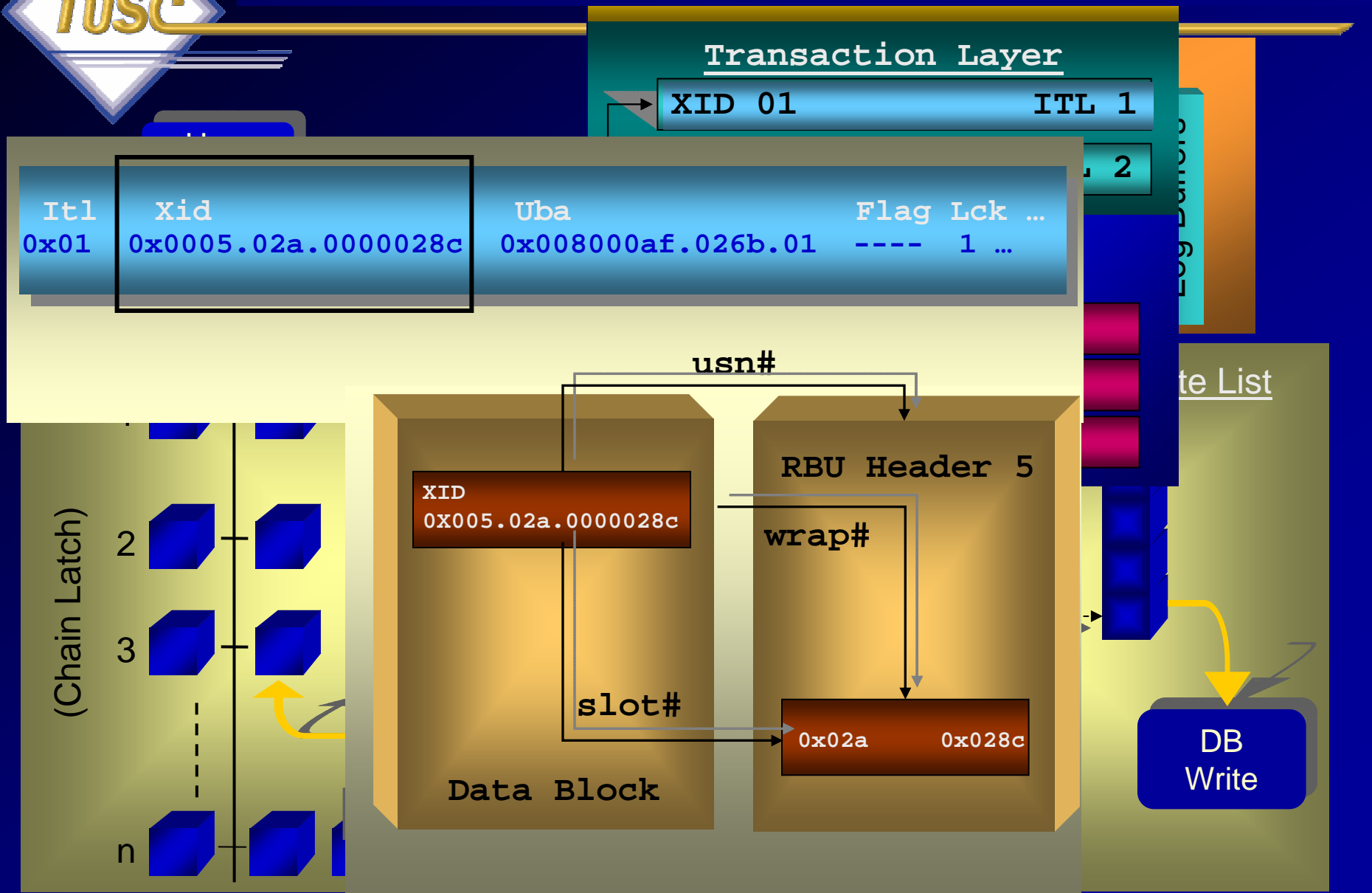
## User 2 updates Row 3

- User1 updates a row with an insert/update/delete – an ITL is opened and xid tracks it in the data block.
- The xid ties to the UNDO header block which ties to the UNDO data block for undo.
- If user2 wants to query the row, they create a clone and rollback the transaction going to the undo header and undo block.
- If user3 wants to update same row (they wait). If user 3 wants to update different row then they open a second ITL with an xid that maps to an undo header that maps to an undo block.





# Transaction Identifiers





# Block Dumps – Top Section / ITL

| Itl  | Xid                 | Uba                | Flag | Lck | Scn/Fsc             |
|------|---------------------|--------------------|------|-----|---------------------|
| 0x01 | 0x0004.010.00000fba | 0x0080003d.08b5.10 | ---- | 4   | fsc 0x009d.00000000 |
| 0x02 | 0x0004.016.00000fae | 0x008000cc.08af.34 | C--- | 0   | scn 0x0000.003deb5b |



ITL – 2 Interested  
Transaction Lists



Transaction ID  
Undo#.slot#.wrap#  
(Undo#,slot#,seq#)



UBA:  
File.block(Undo dba).sequence.record  
Undo block address where last change  
is recorded.

Rows Locked:  
4 rows deleted  
for this xid in  
this block.



# AWR – ITL Issues

## Segments by ITL Waits

- % of Capture shows % of ITL waits for each top segment compared
- with total ITL waits for all segments captured by the Snapshot

| Owner | Tablespace Name | Object Name | Subobject Name | Obj. Type       | ITL Waits | % of Capture |
|-------|-----------------|-------------|----------------|-----------------|-----------|--------------|
|       |                 |             |                | INDEX PARTITION | 126       | 32.06        |
|       |                 |             |                | INDEX PARTITION | 112       | 28.50        |
|       |                 |             |                | INDEX PARTITION | 66        | 16.79        |
|       |                 |             |                | INDEX PARTITION | 65        | 16.54        |
|       |                 |             |                | INDEX PARTITION | 12        | 3.05         |



# Statspack - Top Wait Events

## Things to look for...



### Wait Problem

### Potential Fix

Latch Free

Investigate the detail (Covered later)

Log Buffer Space

Increase the Log Buffer; Faster disks for the Redo Logs

Log File Switch

Archive destination slow or full; Add more or larger Redo Logs

Log file sync

Commit more records at a time; Faster Redo Log disks; Raw devices





# Statspack - Top Wait Events

## Things to look for...



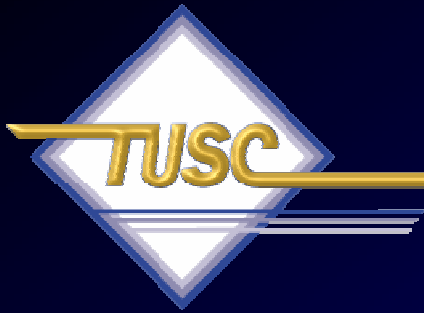
Wait Problem  
CBC Latches

Potential Fix

Cache Buffers Chains Latches – Reduce the length of the hash chain (less copies) by reducing block's popularity. Increase the latches by increasing buffers. Use Oracle SQ generator.

LRU Chain Latch

This latch protects the LRU list when a user needs the latch to scan the LRU chain for a buffer. When a dirty buffer is encountered it is linked to the LRU-W. When adding, moving, or removing a buffer this latch is needed.



# Decoding the Hot/Cold Regions (fyi only)

The Indus Script



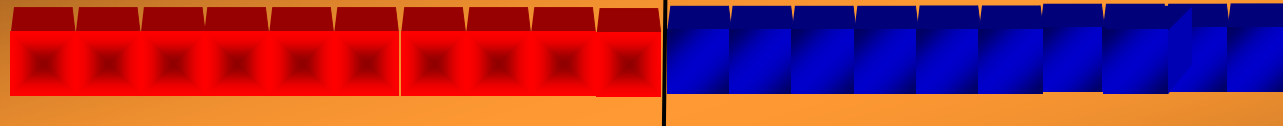


# The percent of buffers in the hot region A look at pointers in the LRU

LRU Lists

Managed as FIFO

Managed as LRU/MRU



`_db_percent_hot_default = 50`

The percent of buffers in the hot region

LRU Lists

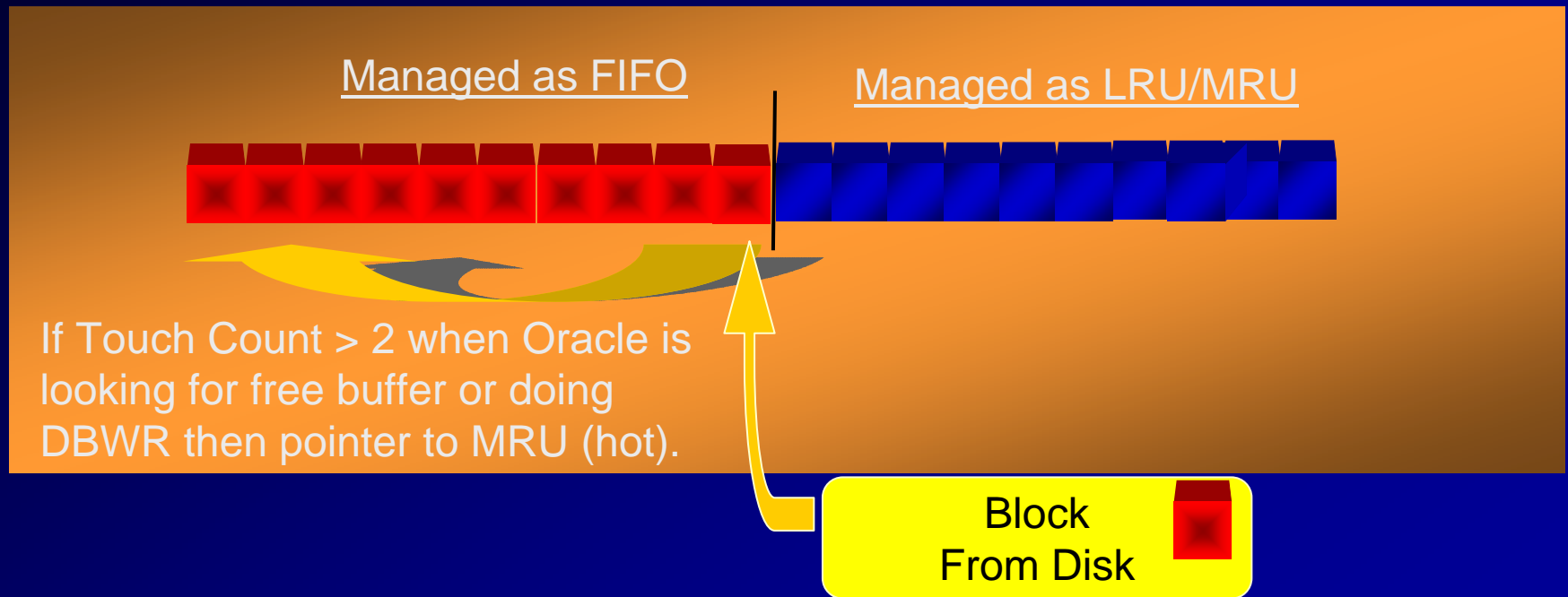


`_db_percent_hot_default = 70 (too high?)`

The percent of buffers in the hot region

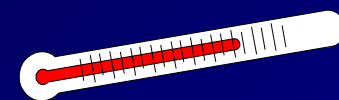


# A look at pointers in the LRU



# Altering the Hot/Cold LRU

## Really advanced tuning!

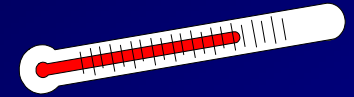


1. `_db_percent_hot_default` (50) – The percent of buffers in the hot region.
2. `_db_aging_touch_time*` (3) – Seconds that must pass to increment touch count again. (Higher - less LRU movement)
3. `_db_aging_hot_criteria` (2) – Threshold to move a buffer to the MRU (hot) end of LRU chain.
4. `_db_aging_stay_count` (0) – \*\*Touch count reset to this when moved to MRU (hot) end. Set=0 even if it was 200 previously!
5. `_db_aging_cool_count` (1) – Touch count reset to this when moved to LRU (cold) end. Set=1 even if it was 200 previously!

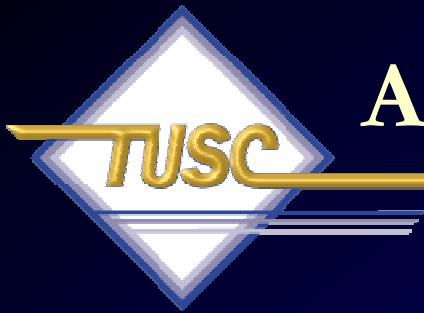
*Setting parameter 1 (above) lower, we increase hanging on to older buffers and setting it higher will cause a flush sooner. (\*Error in description) <sup>34</sup>*

# Altering the Hot/Cold LRU

## Really advanced tuning!



6. `_db_aging_freeze_cr` (FALSE) – Setting this to TRUE will make cr (consistent read) buffers too cold to keep in the cache.
  7. `_db_percent_hot_keep` (0) – Percent of keep buffers considered hot (in hot region). Old LRU algorithm!
  8. `_db_percent_hot_recycle` (0) – Percent of recycle buffers considered hot (in hot region). Old LRU algorithm!
- \*\* If `_db_aging_stay_count` => `_db_aging_hot_criteria` then touch count is set to  $\frac{1}{2}$  it's current count instead of setting it to the `_db_aging_stay_count` when moved to the hot end of LRU.*
- FTS, FFIS (multi-block) are put on the cold end of the LRU.*



## Additional LRU's / Faster!!

- LRU Main block replacement list
- LRU-W Old dirty buffers and reco/temp
- LRU-P Ping Buffer list / RAC
- LRU-XO Buffers to be written for drop/truncate
- LRU-XR Buffers to be written for reuse range
- Thread CKPT Thread Checkpoint Queue
- File CKPT File Checkpoint Queue
- Reco CKPT Reco Checkpoint
- LRU-MAIN & LRU-AUX help LRU





# Locally Managed Tablespaces- LMT



- Manage space locally using bitmaps
- Benefits
  - no tablespace fragmentation issues
  - better performance handling on large segments
  - lower contention for central resources, e.g., no ST enqueue contention
  - fewer recursive calls
- Implementation
  - specify “EXTENT MANAGEMENT LOCAL” clause during tablespace creation
  - in-place migration



# Automatic Segment Space Management (ASSM)



- Automatic intra-object space management
- Benefits –
  - Eliminates Rollback Segment Management
  - simplified administration (**no more FREELISTS, FREELIST GROUPS, PCTUSED**)
  - improved space utilization & concurrency
  - enhanced performance
- Can set undo retention time
  - If set to longest running query time, no more “Snapshot too old”!
- Implementation
  - specify “SEGMENT SPACE MANAGEMENT AUTO” clause during tablespace creation

# 10g Only – Wait Event Histogram (Breaks down the wait detail)



Wait Event Histogram DB/Inst: ORCL/orcl Snaps: 1-2  
-> ordered by event (idle events last)

Event

|                             | 0 - 1 ms | 1 - 4 ms | 4 - 8 ms | 8 - 16 ms | 16 - 32 ms | 32+ ms          |
|-----------------------------|----------|----------|----------|-----------|------------|-----------------|
| enq: HW - contention        | 243      | 74       | 73       | 60        | 55         | 197             |
| enq: TX - contention        | 6        | 3        | 0        | 0         | 0          | 61              |
| latch free                  | 657      | 2        | 0        | 0         | 0          | 0               |
| latch: In memory undo latch | 3        | 3        | 1        | 1         | 0          | 0               |
| latch: cache buffers chains | 89       | 7        | 14       | 68        | 1          | 0               |
| latch: redo allocation      | 35       | 56       | 25       | 3         | 0          | 39 <sup>0</sup> |



# 10g– Enqueues Spelled Out



Enqueue activity DB/Inst: ORCL/orcl Snaps: 1-2

-> Enqueue stats gathered prior to 10g should not be compared with 10g data

Enqueue Type (Request Reason)

| Requests                          | Succ Gets | Failed Gets | Waits | Wt Time (s) | Av Wt Time(ms) |
|-----------------------------------|-----------|-------------|-------|-------------|----------------|
| -----                             |           |             |       |             |                |
| HW-Segment High Water Mark        |           |             |       |             |                |
| 5,264                             | 5,245     | 19          | 552   | 168         | 304.96         |
| TX-Transaction                    |           |             |       |             |                |
| 1,330                             | 1,330     | 0           | 61    | 77          | 1,255.21       |
| CF-Controlfile Transaction        |           |             |       |             |                |
| 4,400                             | 4,298     | 102         | 159   | 43          | 271.14         |
| CI-Cross-Instance Call Invocation |           |             |       |             |                |
| 460                               | 460       | 0           | 4     | 1           | 157.75         |
| US-Undo Segment                   |           |             |       |             |                |
| 1,021                             | 1,021     | 0           | 7     | 0           | 60.14          |
| FB-Format Block                   |           |             |       |             |                |
| 3,655                             | 3,655     | 0           | 1     | 0           | 60.40          |
| -----                             |           |             |       |             |                |



## Statspack – Top 25



- Tuning the top 25 buffer get and top 25 physical get queries has yielded system performance gains of anywhere from 5 percent to 5000 percent.
- The SQL section of the statspack report tells you which queries to potentially tune first.
- The top 10 of your SQL statements should not be more than 10 percent of your buffer gets or disk reads.



# Statspack – Top SQL



| Buffer Gets<br>(s) Hash Value | Executions | Gets per Exec | %Total | Time(s) | Time |
|-------------------------------|------------|---------------|--------|---------|------|
|-------------------------------|------------|---------------|--------|---------|------|

|             |       |             |       |         |          |
|-------------|-------|-------------|-------|---------|----------|
| -----       | ----- | -----       | ----- | -----   | -----    |
| 627,226,570 | 117   | 5,360,910.9 | 4.7   | 9627.09 | 10367.04 |

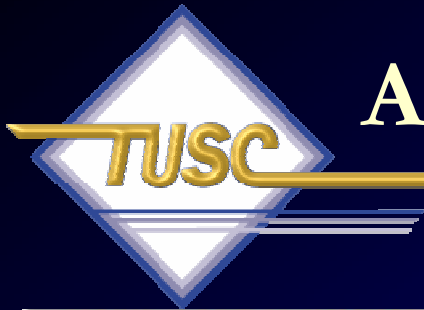
Module: JDBC Thin Client

```
SELECT * FROM (select d1.tablespace_name, d1.owner, d1.segment_t  
ype, d1.segment_name, d1.header_file, d1.extents, d1.bytes, d1.b  
locks, d1.max_extents , d1.next_extent from sys.dba_segments d1  
where d1.segment_type != 'CACHE' and tablespace_name not in (s  
elect distinct tablespace_name from sys.dba_rollback_segs) orde
```

|             |         |         |     |       |          |
|-------------|---------|---------|-----|-------|----------|
| 409,240,446 | 175,418 | 2,332.9 | 3.1 | ##### | 59430.83 |
|-------------|---------|---------|-----|-------|----------|

Module: ? @sap10ci (TNS V1-V3)

```
SELECT "TABNAME" , "VARKEY" , "DATALN" , "VARDATA" FROM "KAPOL"  
WHERE "TABNAME" = :A0 AND "VARKEY" LIKE :A1 ORDER BY "TABNAME" ,  
"VARKEY"
```



# AWR – Top SQL

## SQL ordered by Gets

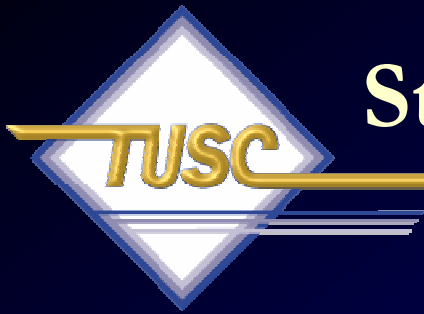
- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- Total Buffer Gets: 225,112,503
- Captured SQL account for 99.9% of Total

| Buffer Gets | Executions | Gets per Exec | % Total | CPU Time (s) | Elapsed Time (s) | SQL Id | SQL Module | SQL Text |
|-------------|------------|---------------|---------|--------------|------------------|--------|------------|----------|
| 224,907,873 | 1,680      | 133,873.73    | 99.91   | 7568.17      | 19515.02         |        |            |          |
| 31,779      | 6,881      | 4.62          | 0.01    | 2.61         | 2.61             |        |            |          |
| 21,515      | 1,688      | 12.75         | 0.01    | 6.18         | 14.19            |        |            |          |
| 19,827      | 1,688      | 11.75         | 0.01    | 2.71         | 10.71            |        |            |          |
| 15,186      | 1,685      | 9.01          | 0.01    | 4.94         | 11.82            |        |            |          |
| 13,501      | 1,685      | 8.01          | 0.01    | 2.84         | 9.72             |        |            |          |
| 7,867       | 1          | 7,867.00      | 0.00    | 7.56         | 23.36            |        |            |          |
| 4,783       | 1          | 4,783.00      | 0.00    | 0.76         | 1.09             |        |            |          |
| 3,906       | 651        | 6.00          | 0.00    | 0.53         | 0.53             |        |            |          |
| 2,640       | 646        | 4.09          | 0.00    | 0.62         | 1.21             |        |            |          |

[Back to SQL Statistics](#)

[Back to Top](#)

## SQL ordered by Reads



# Statspack - Latch Waits



Latch Free – Latches are low-level queueing mechanisms (they're accurately referred to as mutually exclusion mechanisms) used to protect shared memory structures in the System Global Area (SGA).

- Latches are like locks on memory that are very quickly obtained and released.
- Latches are used to prevent concurrent access to a shared memory structure.
- If the latch is not available, a latch free miss is recorded.





# Statspack - Latch Waits



## Latch Free –

- Most latch problems are related to:
  - The failure to use bind variables (library cache latch)
  - Redo generation issues (redo allocation latch)
  - Buffer cache contention issues (cache buffers lru chain)
  - Hot blocks in the buffer cache (cache buffers chains).
- There are also latch waits related to bugs; check MetaLink for bug reports if you suspect this is the case ([oracle.com/support](http://oracle.com/support)).
- When latch miss ratios are greater than 0.5 percent, you should investigate the issue.



# Statspack - Latch Waits



Latch Activity for DB: ORA9I Instance: ora9i Snaps: 1 -2

| Latch                   | Get<br>Requests | Pct<br>Get<br>Miss | Avg<br>Slps<br>/Miss | Wait<br>Time<br>(s) | NoWait<br>Requests | Pct<br>NoWait<br>Miss |
|-------------------------|-----------------|--------------------|----------------------|---------------------|--------------------|-----------------------|
| -----                   | -----           | -----              | -----                | -----               | -----              | -----                 |
| KCL freelist latch      | 4,924           | 0.0                |                      |                     | 0                  |                       |
| cache buffer handles    | 968,992         | 0.0                | 0.0                  |                     | 0                  |                       |
| cache buffers chains    | 761,708,539     | 0.0                | 0.4                  |                     | 21,519,841         | 0.0                   |
| cache buffers lru chain | 8,111,269       | 0.1                | 0.8                  |                     | 19,834,466         | 0.1                   |
| library cache           | 67,602,665      | 2.2                | 2.0                  |                     | 213,590            | 0.8                   |
| redo allocation         | 12,446,986      | 0.2                | 0.0                  |                     | 0                  |                       |
| redo copy               | 320             | 0.0                |                      |                     | 10,335,430         | 0.1                   |
| user lock               | 1,973           | 0.3                | 1.2                  |                     | 0                  |                       |



# Statspack - Latch Waits



Latch Sleep breakdown for DB:

-> ordered by misses desc

| Latch Name              | Get Requests | Misses    | Sleeps    | Spin & Sleeps 1->4                    |
|-------------------------|--------------|-----------|-----------|---------------------------------------|
| -----                   | -----        | -----     | -----     | -----                                 |
| library cache           | 67,602,665   | 1,474,032 | 2,935,368 | 199143/28003<br>6/582413/412<br>440/0 |
| cache buffers chains    | 761,708,539  | 192,942   | 83,559    | 110054/82239<br>/628/21/0             |
| redo allocation         | 12,446,986   | 25,444    | 1,135     | 24310/1133/1<br>/0/0                  |
| cache buffers lru chain | 8,111,269    | 6,285     | 4,933     | 1378/4881/26<br>/0/0                  |
| process allocation      | 177          | 7         | 7         | 0/7/0/0/0                             |
| -----                   | -----        | -----     | -----     | -----                                 |

- Note that 10g only has “Spin & Sleeps 1-3+”



# Statspack - Latch Waits



Latches that are willing to wait try to acquire a latch. If none are available, it will spin and then request the latch again. It will continue to do this up to the `_SPIN_COUNT` initialization parameter (note that spinning costs CPU).

- If it can't get a latch after spinning up to the `_SPIN_COUNT`, it will go to sleep. It will wake up after one centisecond (one hundredth of a second). It will do this twice.
- It will then start this process again, spinning up to the `_SPIN_COUNT` and then sleeping for twice as long (two centiseconds). After doing this again it will double again. So the pattern is 1,1,2,2,4,4 etc. It will do this until it gets the latch
- Every time the latch sleeps, it will create a latch sleep wait. An example of a “willing to wait” latch is a library cache latch.



## Statspack - Latch Waits



- Some latches are “not willing to wait.” This type of latch does not wait for the latch to become available.
- They immediately time out and retry to obtain the latch.
  - A **redo copy latch** is an **example** of a “not willing to wait” latch.
  - A not willing to wait latch will generate information for the `immediate_gets` and the `immediate_misses` columns of the `V$LATCH` view and also in the statspack report.
  - The hit ratio for these latches should also approach 99% and the misses should never fall below 1 percent misses.



## Statspack - Latch Waits



- *Gets* - The number of times a willing to wait request for a latch was requested and it was available.
- *Misses* - The number of times a willing to wait request for latch was initially requested but was not available.
- *Sleeps* - The number of a willing to wait request for a latch failed over and over until the spin count was exceeded and the process went to sleep. The number of sleeps may be higher than the misses. Processes may sleep multiple times before obtaining the latch.
- *NoWait Misses* - The number of times immediate (not willing to wait) request for a latch was unsuccessful.



# Statspack - Latch Waits

## Things to look for...



### Latch Problem

Library Cache

Shared Pool

Redo allocation

Redo copy

Row cache objects

### Potential Fix

Use bind variables; adjust the  
shared\_pool\_size

Use bind variables; adjust the  
shared\_pool\_size

Minimize redo generation and  
avoid unnecessary commits

Increase the  
\_log\_simultaneous\_copies

Increase the Shared Pool



## Use Bind Variables / Latch Issues

- If you don't use bind variables in your application code you usually end up with latch contention with the *shared\_pool* and *library cache* latches.
- This latch wait time can be reduced by changing just the top couple of executed statements that were using literal SQL instead of bind variables.
- Oracle 8i Release 2 (8.1.6) has an auto-conversion of literals into bind variables... **Oracle9i extends this slightly...**

**CURSOR\_SHARING=FORCE**  
(Default is EXACT; 8iR2)





## Cursor Sharing - 8.1.6+

If v\$sqlarea looks like this:

```
select empno from rich778 where empno =451572
```

```
select empno from rich778 where empno =451573
```

```
select empno from rich778 where empno =451574
```

```
select empno from rich778 where empno =451575
```

```
select empno from rich778 where empno =451576
```

Use cursor sharing=force (sqlarea goes to this):

```
select empno from rich778 where empno =:SYS_B_0
```



# Statspack – Instance Activity Terminology...



| <u>Statistic</u>      | <u>Description</u>                                                                                                                                                                        |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Session Logical Reads | All reads cached in memory. Includes both consistent gets and also the db block gets.                                                                                                     |
| Consistent Gets       | These are the reads of a block that are in the cache. They are NOT to be confused with consistent read (cr) version of a block in the buffer cache (usually the current version is read). |
| Db block gets         | These are block gotten to be changed. MUST be the CURRENT block and not a cr block.                                                                                                       |
| Db block changes      | These are the db block gets (above) that were actually changed.                                                                                                                           |
| Physical Reads        | Blocks not read from the cache. Either from disk, disk cache or O/S cache; there are also physical reads direct which bypass cache using Parallel Query (not in hit ratios).              |



# Statspack – Instance Activity



| Statistic                        | Total          | per Second | per Trans |
|----------------------------------|----------------|------------|-----------|
| -----                            | -----          | -----      | -----     |
| branch node splits               | 7,162          | 0.1        | 0.0       |
| consistent gets                  | 12,931,850,777 | 152,858.8  | 3,969.5   |
| current blocks converted for CR  | 75,709         | 0.9        | 0.0       |
| db block changes                 | 343,632,442    | 4,061.9    | 105.5     |
| db block gets                    | 390,323,754    | 4,613.8    | 119.8     |
| hot buffers moved to head of LRU | 197,262,394    | 2,331.7    | 60.6      |
| leaf node 90-10 splits           | 26,429         | 0.3        | 0.0       |
| leaf node splits                 | 840,436        | 9.9        | 0.3       |
| logons cumulative                | 21,369         | 0.3        | 0.0       |
| physical reads                   | 504,643,275    | 5,965.1    | 154.9     |
| physical writes                  | 49,724,268     | 587.8      | 15.3      |
| session logical reads            | 13,322,170,917 | 157,472.5  | 4,089.4   |
| sorts (disk)                     | 4,132          | 0.1        | 0.0       |
| sorts (memory)                   | 7,938,085      | 93.8       | 2.4       |
| sorts (rows)                     | 906,207,041    | 10,711.7   | 278.2     |
| table fetch continued row        | 25,506,365     | 301.5      | 7.8       |
| table scans (long tables)        | 111            | 0.0        | 0.0       |
| table scans (short tables)       | 1,543,085      | 18.2       | 55<br>0.5 |



# Statspack – Time Model Stats



Time Model System Stats DB/Inst: ORCL/orcl Snaps: 1-2

-> Total Time in Database calls 5560.5s (or 5560475336us)

| Statistic                           | Time (s) | % of DB Time       |
|-------------------------------------|----------|--------------------|
| -----                               | -----    | -----              |
| DB CPU                              | 119.1    | 2.1                |
| DB time                             | 5,560.5  |                    |
| Java execution elapsed time         | 0.1      | .0                 |
| PL/SQL compilation elapsed time     | 0.4      | .0                 |
| PL/SQL execution elapsed time       | 0.2      | .0                 |
| background cpu time                 | 16.4     | .3                 |
| background elapsed time             | 2,097.6  | 37.7               |
| connection management call elapsed  | 1.2      | .0                 |
| failed parse elapsed time           | 0.2      | .0                 |
| hard parse (sharing criteria) elaps | 0.0      | .0                 |
| hard parse elapsed time             | 13.3     | .2                 |
| parse time elapsed                  | 14.1     | .3                 |
| sql execute elapsed time            | 5,556.3  | 99.9 <sup>56</sup> |



# Statspack – Time Model Stats



## Time Model Statistics

- Total time in database user-calls (DB Time): 24728.7s
- Statistics including the word "background" measure background process time, and so do not contribute to the DB time statistic
- Ordered by % or DB time desc, Statistic name

| Statistic Name                             | Time (s)  | % of DB Time |
|--------------------------------------------|-----------|--------------|
| sql execute elapsed time                   | 19,563.36 | 79.11        |
| DB CPU                                     | 7,635.77  | 30.88        |
| PL/SQL execution elapsed time              | 9.04      | 0.04         |
| connection management call elapsed time    | 5.37      | 0.02         |
| parse time elapsed                         | 3.94      | 0.02         |
| hard parse elapsed time                    | 0.37      | 0.00         |
| hard parse (sharing criteria) elapsed time | 0.12      | 0.00         |
| sequence load elapsed time                 | 0.01      | 0.00         |
| hard parse (bind mismatch) elapsed time    | 0.00      | 0.00         |
| repeated bind elapsed time                 | 0.00      | 0.00         |
| DB time                                    | 24,728.66 |              |
| background elapsed time                    | 2,773.25  |              |
| background cpu time                        | 1,360.81  |              |

[Back to Wait Events Statistics](#)

[Back to Top](#)

## Wait Class



# Statspack – O/S Statistics



OS Statistics DB/Inst: ORCL/orcl Snaps: 1-2

| Statistic      | Total      | per Second |
|----------------|------------|------------|
| -----          | -----      | -----      |
| AVG_IN_BYTES   | 37,371,904 | 12,583     |
| AVG_OUT_BYTES  | 1,179,648  | 397        |
| IN_BYTES       | 37,371,904 | 12,583     |
| OUT_BYTES      | 1,179,648  | 397        |
| AVG_BUSY_TICKS | 24,326     |            |
| AVG_IDLE_TICKS | 272,598    |            |
| AVG_SYS_TICKS  | 9,483      |            |
| AVG_USER_TICKS | 14,843     |            |
| BUSY_TICKS     | 24,326     |            |
| IDLE_TICKS     | 272,598    |            |
| SYS_TICKS      | 9,483      |            |
| USER_TICKS     | 14,843     |            |
| NUM_CPUS       | 1          |            |



# Statspack – Undo Stats



Undo Segment Stats DB/Inst: ORCL/orcl Snaps: 1-2

-> ordered by Time desc

| End Time     | Undo<br>Blocks | Num<br>Trans | Max Qry<br>Len (s) | Max Tx<br>Concy | Snap<br>TooOld | OutOf<br>Space | uS/uR/uU/<br>eS/eR/eU |
|--------------|----------------|--------------|--------------------|-----------------|----------------|----------------|-----------------------|
| 12-Apr 12:19 | 31,664         | 243          | 818                | 5               | 0              | 0              | 0/0/0/32/28<br>0/0    |
| 12-Apr 12:09 | 27,248         | 357          | 210                | 5               | 0              | 0              | 0/0/0/55/57<br>6/0    |
| 12-Apr 11:59 | 16,403         | 276          | 171                | 5               | 0              | 0              | 0/0/0/28/15<br>2/0    |
| 12-Apr 11:49 | 3,602          | 327          | 0                  | 5               | 0              | 0              | 0/0/0/8/16/<br>0      |
| 12-Apr 11:39 | 469            | 2,946        | 0                  | 4               | 0              | 0              | 0/0/0/0/0/0           |



# Statspack – File I/O



## Tablespace

| Tablespace | Av<br>Reads | Av<br>Reads/s | Av<br>Rd(ms) | Av<br>Blks/Rd | Writes    | Av<br>Writes/s | Buffer<br>Waits | Av Buf<br>Wt(ms) |
|------------|-------------|---------------|--------------|---------------|-----------|----------------|-----------------|------------------|
| PSAPSTABI  | 14,441,749  | 171           | 7.9          | 1.0           | 521,275   | 6              | 1,234,608       | 6.2              |
| PSAPVBAPD  | 13,639,443  | 161           | 6.2          | 1.7           | 10,057    | 0              | 2,672,470       | 4.2              |
| PSAPEDII   | 11,992,418  | 142           | 5.3          | 1.0           | 83,757    | 1              | 4,115,714       | 4.4              |
| PSAPEDID   | 10,617,042  | 125           | 8.1          | 1.0           | 64,866    | 1              | 3,728,009       | 6.4              |
| PSAPROLL   | 998,328     | 12            | 13.2         | 1.0           | 8,321,252 | 98             | 285,060         | 65.7             |

- Reads should be below 14ms





# AWR – File I/O

## File IO Stats

- ordered by Tablespace, File

| Tablespace | Filename | Reads | Av Reads/s | Av Rd (ms) | Av Blks/Rd | Writes | Av Writes/s | Buffer Waits | Av Buf Wt (ms) |
|------------|----------|-------|------------|------------|------------|--------|-------------|--------------|----------------|
|            |          | 7     | 0          | 1.43       | 1.00       | 480    | 1           | 93           | 19.68          |
|            |          | 6     | 0          | 0.00       | 1.00       | 34     | 0           | 34           | 16.76          |
|            |          | 3     | 0          | 0.00       | 1.00       | 3      | 0           | 0            | 0.00           |
|            | .dbf     | 7     | 0          | 5.71       | 1.00       | 4      | 0           | 0            | 0.00           |
|            | .dbf     | 7     | 0          | 0.00       | 1.00       | 4      | 0           | 0            | 0.00           |
|            | .dbf     | 7     | 0          | 0.00       | 1.00       | 4      | 0           | 0            | 0.00           |
|            | .dbf     | 7     | 0          | 0.00       | 1.00       | 4      | 0           | 0            | 0.00           |
|            | .dbf     | 7     | 0          | 0.00       | 1.00       | 4      | 0           | 0            | 0.00           |
|            | .dbf     | 7     | 0          | 2.86       | 1.00       | 4      | 0           | 0            | 0.00           |
|            | .dbf     | 7     | 0          | 0.00       | 1.00       | 4      | 0           | 0            | 0.00           |
|            | .dbf     | 7     | 0          | 1.43       | 1.00       | 4      | 0           | 0            | 0.00           |
|            | .dbf     | 7     | 0          | 2.86       | 1.00       | 4      | 0           | 0            | 0.00           |
|            | .dbf     | 7     | 0          | 0.00       | 1.00       | 4      | 0           | 0            | 0.00           |
|            | .dbf     | 7     | 0          | 0.00       | 1.00       | 4      | 0           | 0            | 0.00           |
|            | .dbf     | 7     | 0          | 0.00       | 1.00       | 4      | 0           | 0            | 0.00           |
|            | .dbf     | 8     | 0          | 0.00       | 1.00       | 3      | 0           | 0            | 0.00           |
|            | .dbf     | 8     | 0          | 0.00       | 1.00       | 3      | 0           | 0            | 0.00           |
|            | .dbf     | 8     | 0          | 0.00       | 1.00       | 3      | 0           | 0            | 0.00           |
|            | .dbf     | 8     | 0          | 0.00       | 1.00       | 3      | 0           | 0            | 0.00           |
|            | .dbf     | 8     | 0          | 0.00       | 1.00       | 3      | 0           | 0            | 0.00           |



# Statspack – File Read Histogram



File Read Histogram Stats DB/Inst: ORCL/orcl Snaps: 1-2

->Number of single block reads in each time range

->ordered by Tablespace, File

| Tablespace |                                     | Filename  |            |        |  |
|------------|-------------------------------------|-----------|------------|--------|--|
| -----      |                                     | -----     |            |        |  |
| 0 - 4 ms   | 4 - 8 ms                            | 8 - 16 ms | 16 - 32 ms | 32+ ms |  |
| -----      |                                     | -----     |            |        |  |
| USERS      | C:\ORACLE\ORADATA\ORCL\USERS01.DBF  |           |            |        |  |
| 560        | 2,225                               | 1,452     | 4,089      | 8,373  |  |
|            |                                     |           |            |        |  |
| SYSAUX     | C:\ORACLE\ORADATA\ORCL\SYSAUX01.DBF |           |            |        |  |
| 123        | 40                                  | 242       | 357        | 405    |  |
|            |                                     |           |            |        |  |
| SYSTEM     | C:\ORACLE\ORADATA\ORCL\SYSTEM01.DBF |           |            |        |  |
| 100        | 51                                  | 223       | 207        | 362    |  |



# Statspack – Check Regularly



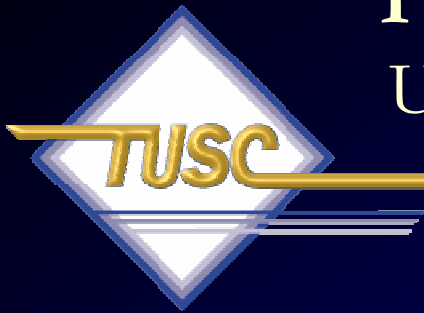
1. Top 5 wait events
2. Load Profile
3. Instance Efficiency Hit Ratios
4. Wait Events / **Wait Event Histograms**
5. Latch Waits
6. Top SQL
7. Instance Activity / **Time Model Stats / O/S Stats**
8. File I/O / **File Read Histogram / Undo Stats**
9. Memory Allocation
10. Undo



## Tuning the RAC Cluster Interconnect

### Using STATSPACK Reports (FYI Only)

- The STATSPACK report shows statistics ONLY for the node or instance on which it was run.
- Run `statspack.snap` procedure and `spreport.sql` script on each node you want to monitor to compare to other instances.
- Single-instance tuning should be performed before attempting to tune the processes that communicate via the cluster interconnect.



# Tuning the RAC Cluster Interconnect Using STATSPACK Reports (FYI Only)

## Top 5 Timed Events

Top 5 Timed Events

~~~~~

Event

Waits

Time (s)

% Elapsed  
Time

global cache cr request

820

154

72.50

CPU time

54

25.34

global cache null to x

478

1

.52

control file sequential read

600

1

.52

control file parallel write

141

1

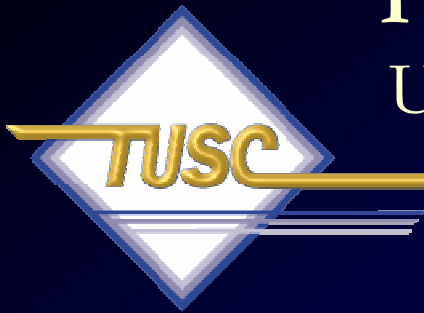
.28

-----

CPU time (processing  
time) should be the  
predominant event

Exceeds CPU time,  
therefore needs  
investigation.

- Transfer times excessive from other instances in the cluster to this instance.
- Could be due to network problems or buffer cache sizing issues.



# Tuning the RAC Cluster Interconnect Using STATSPACK Reports (FYI Only)

- Network changes were made
- An index was added
- STATSPACK report now looks like this:

Top 5 Timed Events

~~~~~

Event

Waits

Time (s)

% Total

Ela Time

-----

CPU time

global cache null to x

enqueue

global cache busy

DFS lock handle

1,655

46

104

38

99

28

8

7

2

64.87

18.43

5.12

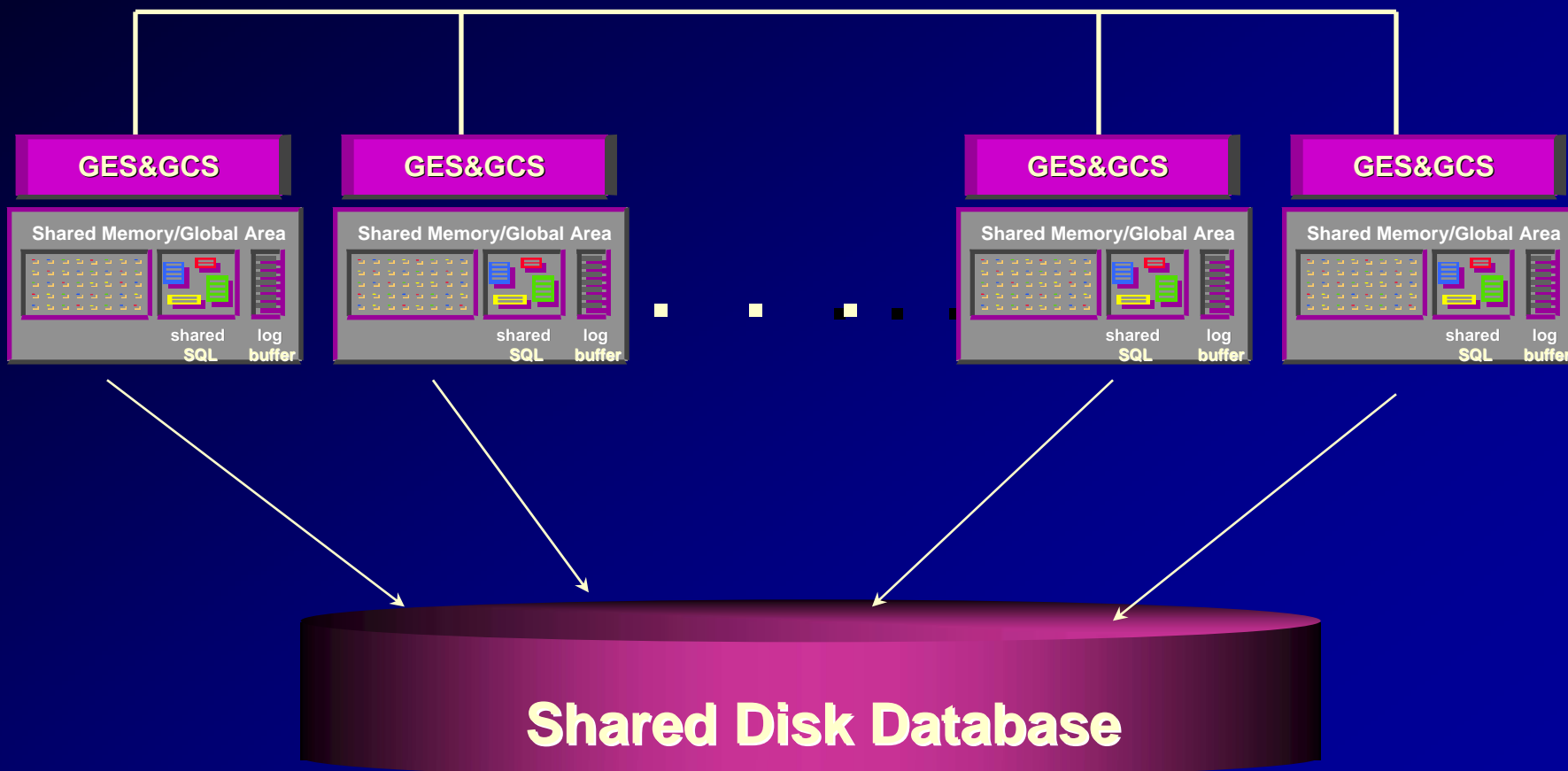
4.73

1.64

CPU time is now the  
predominant event

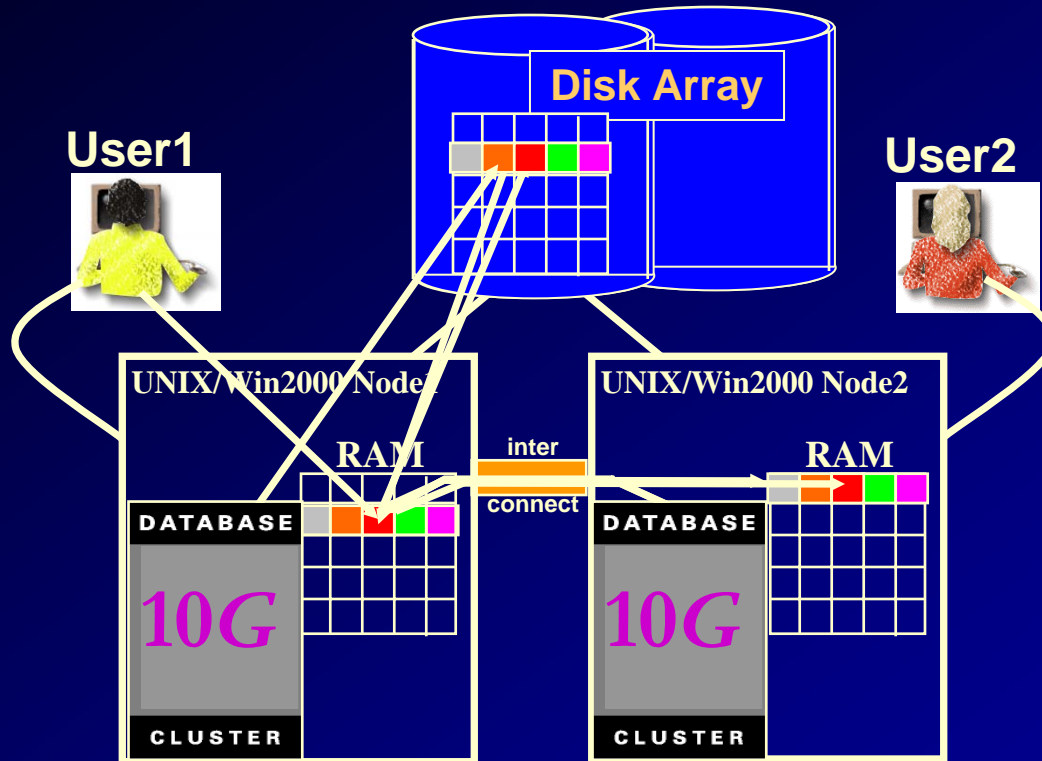


# RAC Architecture Shared Data Model





# Real Applications Clusters - Cache Fusion



1. User1 queries data
2. User2 queries same data - via interconnect with no disc I/O
3. User1 updates a row of data and commits
4. User2 wants to update same block of data – 10g keeps data concurrency via interconnect





# Tuning the RAC Cluster Interconnect Using STATSPACK Reports (FYI)

## – Workload characteristics for this instance:

Cluster Statistics for DB: DB2 Instance: INST1

Global Cache Service - Workload Characteristics

-----

|                                                 |      |      |
|-------------------------------------------------|------|------|
| Ave global cache get time (ms):                 | 8.2  | 3.1  |
| Ave global cache convert time (ms):             | 16.5 | 3.2  |
| Ave build time for CR block (ms):               | 1.5  | 0.2  |
| Ave flush time for CR block (ms):               | 6.0  | 0.0  |
| Ave send time for CR block (ms):                | 0.9  | 1.0  |
| Ave time to process CR block request (ms):      | 8.5  | 1.3  |
| Ave receive time for CR block (ms):             | 18.3 | 17.2 |
| Ave pin time for current block (ms):            | 13.7 | 0.2  |
| Ave flush time for current block (ms):          | 3.9  | 0.0  |
| Ave send time for current block (ms):           | 0.8  | 0.9  |
| Ave time to process current block request (ms): | 18.4 | 1.1  |
| Ave receive time for current block (ms):        | 17.4 | 3.1  |
| Global cache hit ratio:                         | 2.5  | 1.7  |
| Ratio of current block defers:                  | 0.2  | 0.0  |
| % of messages sent for buffer gets:             | 2.2  | 1.4  |
| % of remote buffer gets:                        | 1.6  | 1.1  |
| Ratio of I/O for coherence:                     | 2.9  | 8.7  |
| Ratio of local vs remote work:                  | 0.5  | 0.6  |
| Ratio of fusion vs physical writes:             | 0.0  | 0.0  |

Solved after network  
and index changes.



## Tuning the RAC Cluster Interconnect Using STATSPACK Reports (FYI Only)

- Global Enqueue Services (GES) control the inter-instance locks in Oracle 9i RAC.
- The STATSPACK report contains a special section for these statistics.

### Global Enqueue Service Statistics

-----

|                                                    |     |
|----------------------------------------------------|-----|
| Ave global lock get time (ms):                     | 0.9 |
| Ave global lock convert time (ms):                 | 1.3 |
| Ratio of global lock gets vs global lock releases: | 1.1 |



## Tuning the RAC Cluster Interconnect Using STATSPACK Reports (FYI Only)

- Guidelines for GES Statistics:
  - All times should be  $< 15\text{ms}$
  - Ratio of global lock gets vs global lock releases should be near 1.0
- High values could indicate possible network or memory problems
- Could also be caused by application locking issues
- May need to review the enqueue section of STATSPACK report for further analysis.

Complete Presentation by Oracle's Rich Niemiec's at:  
<http://www.oracleracsig.org>



# Tuning the RAC Cluster Interconnect Using AWR Reports (FYI Only)

## Global Cache Load Profile

|                                | Per Second | Per Transaction |
|--------------------------------|------------|-----------------|
| Global Cache blocks received:  | 0.38       | 0.05            |
| Global Cache blocks served:    | 0.26       | 0.04            |
| GCS/GES messages received:     | 766.83     | 106.40          |
| GCS/GES messages sent:         | 1,278.25   | 177.36          |
| DBWR Fusion writes:            | 0.01       | 0.00            |
| Estd Interconnect traffic (KB) | 404.57     |                 |

## Global Cache Efficiency Percentages (Target local+remote 100%)

|                                 |        |
|---------------------------------|--------|
| Buffer access - local cache %:  | 100.00 |
| Buffer access - remote cache %: | 0.00   |
| Buffer access - disk %:         | 0.00   |

## Global Cache and Enqueue Services - Workload Characteristics

|                                                   |     |
|---------------------------------------------------|-----|
| Avg global enqueue get time (ms):                 | 1.9 |
| Avg global cache cr block receive time (ms):      | 1.8 |
| Avg global cache current block receive time (ms): | 1.9 |
| Avg global cache cr block build time (ms):        | 0.0 |
| Avg global cache cr block send time (ms):         | 0.2 |
| Global cache log flushes for cr blocks served %:  | 0.0 |
| Avg global cache cr block flush time (ms):        |     |
| Avg global cache current block pin time (ms):     | 0.1 |



# Tuning the RAC Cluster Interconnect Using AWR Reports (FYI Only)

SQL ordered by Cluster Wait Time

| Cluster Wait Time (s) | CWT % of Elapsed Time | Elapsed Time(s) | CPU Time (s) | Executions | SQL Id | SQL Module | SQL Text |
|-----------------------|-----------------------|-----------------|--------------|------------|--------|------------|----------|
| 75.00                 | 0.38                  | 19,515.02       | 7,568.17     | 1,680      |        |            |          |
| 0.85                  | 3.63                  | 23.36           | 7.56         | 1          |        |            |          |
| 0.15                  | 13.69                 | 1.09            | 0.76         | 1          |        |            |          |
| 0.09                  | 31.73                 | 0.27            | 0.08         | 46         |        |            |          |
| 0.06                  | 12.31                 | 0.46            | 0.30         | 1          |        |            |          |
| 0.03                  | 0.20                  | 14.19           | 6.18         | 1,688      |        |            |          |
| 0.03                  | 0.26                  | 10.71           | 2.71         | 1,688      |        |            |          |
| 0.03                  | 83.42                 | 0.03            | 0.00         | 1          |        |            |          |
| 0.02                  | 36.30                 | 0.07            | 0.06         | 4          |        |            |          |
| 0.02                  | 43.85                 | 0.06            | 0.01         | 1          |        |            |          |
| 0.02                  | 1.77                  | 1.12            | 0.54         | 10         |        |            |          |
| 0.02                  | 10.23                 | 0.15            | 0.13         | 1          |        |            |          |
| 0.02                  | 3.76                  | 0.40            | 0.02         | 1          |        |            |          |
| 0.01                  | 12.17                 | 0.11            | 0.09         | 1          |        |            |          |
| 0.01                  | 33.26                 | 0.03            | 0.02         | 1          |        |            |          |
| 0.01                  | 8.39                  | 0.09            | 0.09         | 177        |        |            |          |
| 0.01                  | 11.26                 | 0.02            | 0.02         | 1          |        |            |          |

Done

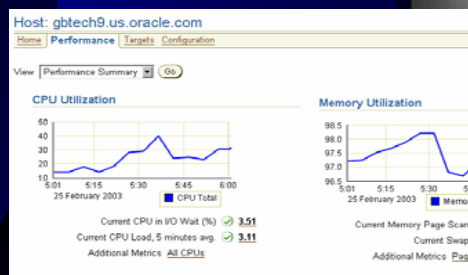
Internet



# The Future...

## Enterprise Manager 10g for the Grid

### Host and Hardware



### Database

#### State

Active Sessions 19

SQL Response Time (%) 83.87 (compared to baseline)

Bad SQL 11

Top SQL Report 238

Duplicate SQL 738

Latest Alert Log Entry No ORA- errors

### Oracle9iAS

Application Server: ias902.dlsun1641.us.oracle.com

View | Top Applications by Average Servlet/JSP Processing Time

| Name   | ORA/J Instance | Total Processing Time (seconds) | Average Servlet/JSP Time (seconds) | Servlet/JSP Requests Processed | Servlet Process Time (sec) |
|--------|----------------|---------------------------------|------------------------------------|--------------------------------|----------------------------|
| hrapp  | home           | 167.20                          | 12.09                              | 11                             | 1                          |
| defaut | home           | 562.77                          | 0.17                               | 3 235                          | 5                          |

### Network and Load Balancer

#### Alerts

| Metric              | Transaction        | Severity |
|---------------------|--------------------|----------|
| Packets Dropped (%) | mail.us.oracle.com | X        |
| Status              | mail.us.oracle.com | X        |

### Administration Monitoring Provisioning Security

Enterprise Manager

### Applications

### Storage

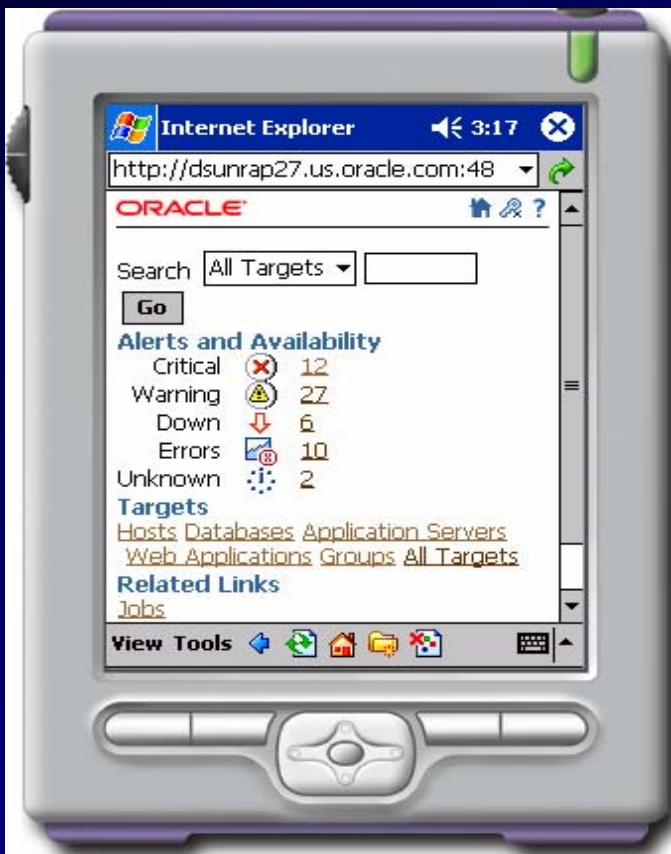
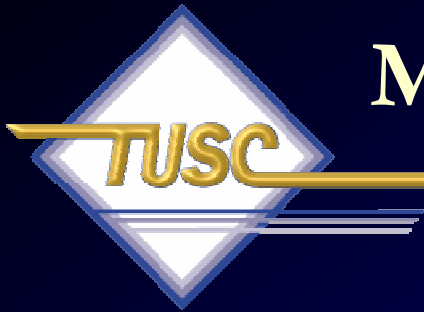
#### Qtrees (ordered by Used (%))

| Status | Name         | Volume | Total GB | Used GB | Used (%) |
|--------|--------------|--------|----------|---------|----------|
| X      | slot3        | eb04   | 60.0     | 58.82   | 98.03    |
| X      | ndw_top      | app04  | 250.0    | 231.48  | 92.59    |
| X      | local_backup | back04 | 250.0    | 219.68  | 87.87    |
| X      | qum_top      | app04  | 350.0    | 298.05  | 85.16    |
| X      | slot1        | eb04   | 60.0     | 49.51   | 82.50    |
| X      | slot2        | eb04   | 60.0     | 47.92   | 79.87    |
| X      | slot4        | eb04   | 60.0     | 47.66   | 79.43    |
| X      | andbackup    | back04 | 100.0    | 82.67   | 82.67    |
| X      | app901sun    | app04  | 50.0     | 26.3    | 52.61    |



# The Future ....

## Manage end to end



Web Services

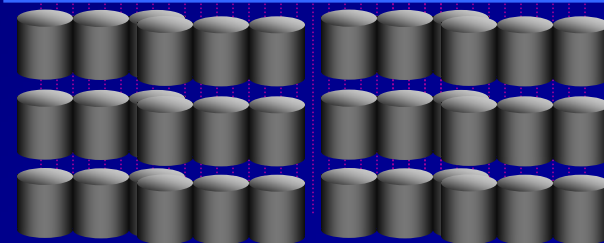
Service Framework

Processor Virtualization



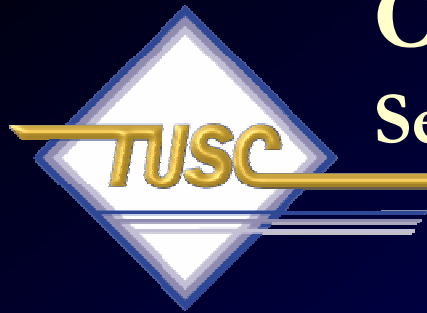
Data Management

Storage Virtualization



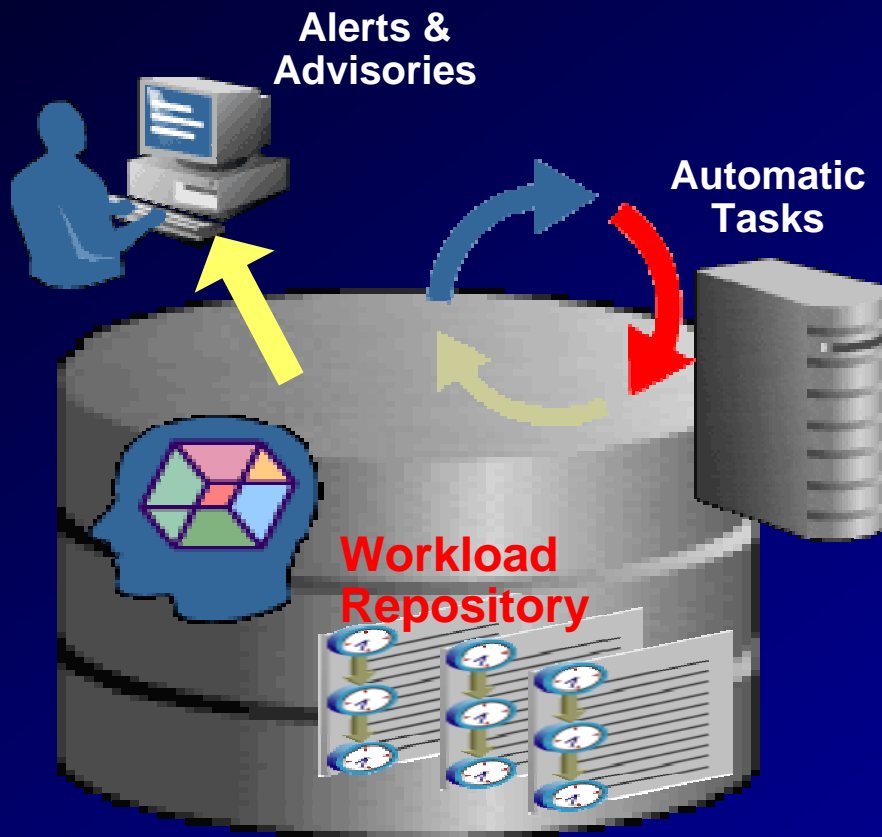
Server  
Pool

Storage  
Pool



# Oracle 10<sup>g</sup> Database

## Self-Managing Intelligent Infrastructure



- Aware
  - Self-monitoring
- Proactive
  - Automatic tasks
  - Proactive alerts
- Intelligent
  - Self-diagnosing
  - Self-tuning





# Monitor All Targets

Monitor  
Targets...  
are they  
up?

Target  
Alerts

Oracle Enterprise Manager (SYSMAN) - Oracle Enterprise Manager Console Homepage - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites Media Print Window Help

Address http://stdlinux01.us.oracle.com:7777/em/console/home

Go Links

ORACLE Enterprise Manager 10g  
Grid Control

Home Targets Deployments Alerts Jobs Management System

Page Refreshed Feb 10, 2004 8:11:14 PM

View All Targets

Status

Total Monitored Targets 34

All Targets Availability

3%  
97%

Unknown(1)  
Up(33)

All Targets Alerts

Critical 1  
Warning 1  
Errors 2

All Targets Jobs

Problem Executions (last 7 days) 0  
Suspended Executions (last 7 days) 0

Target Search

Search All Targets Go

Critical Patch Advisories

Patch Advisories 0  
Affected Oracle Homes 0

Deployments Summary

View Database Installations

| Database Installations         | Targets | Installations | Interim Patches Applied |
|--------------------------------|---------|---------------|-------------------------|
| Oracle Database 10g 10.1.0.2.0 | 1       | 1             | No                      |
| Oracle9i 9.0.1.5.0             | 2       | 2             | No                      |

Resource Center

[Documentation](#)  
[Release Notes](#)  
[Support](#)  
[Oracle Technology Network](#)

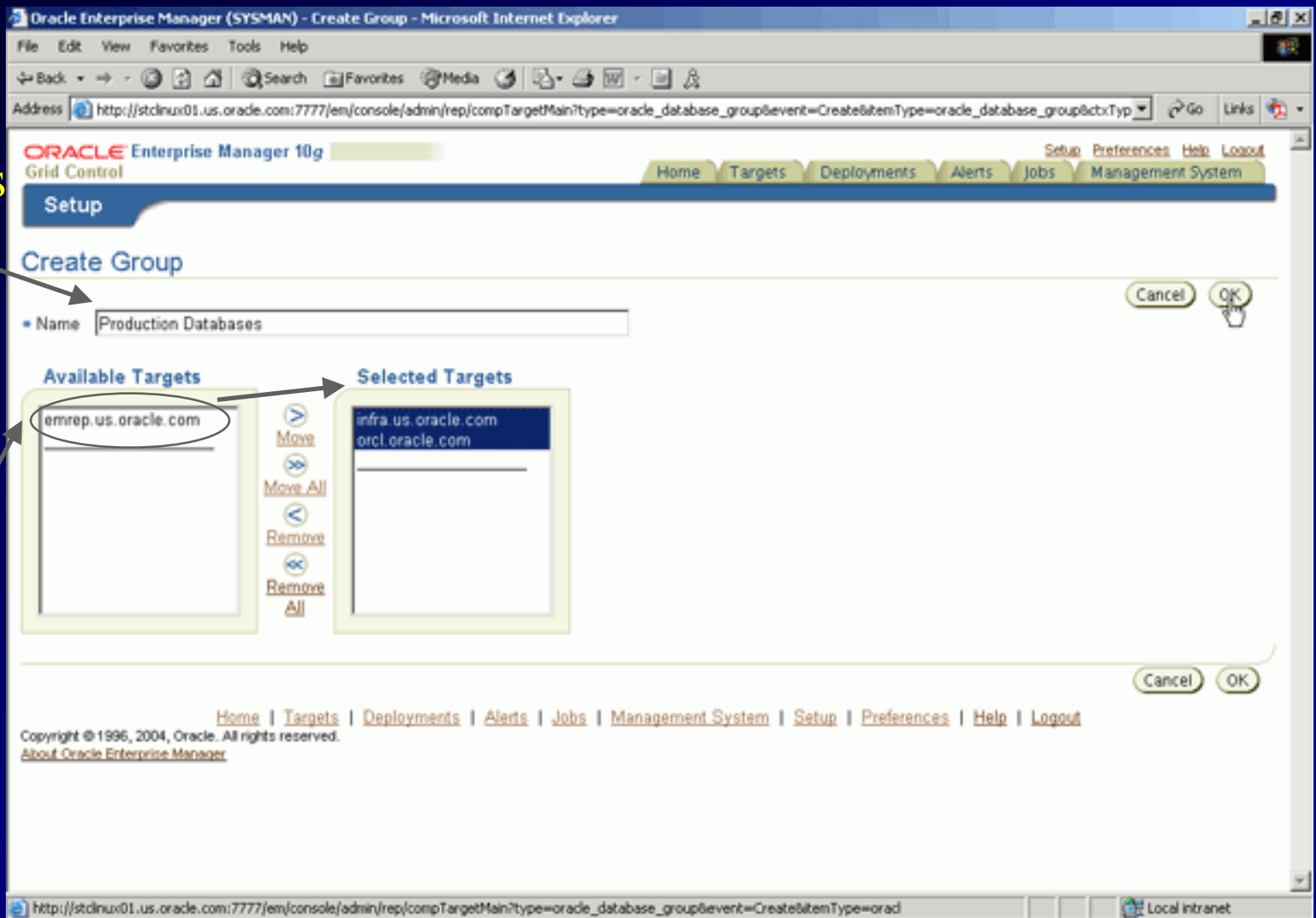
Local intranet

# Database Performance

TUSC

Create a  
Prod DB's  
Group

Pick  
Items for  
the Group

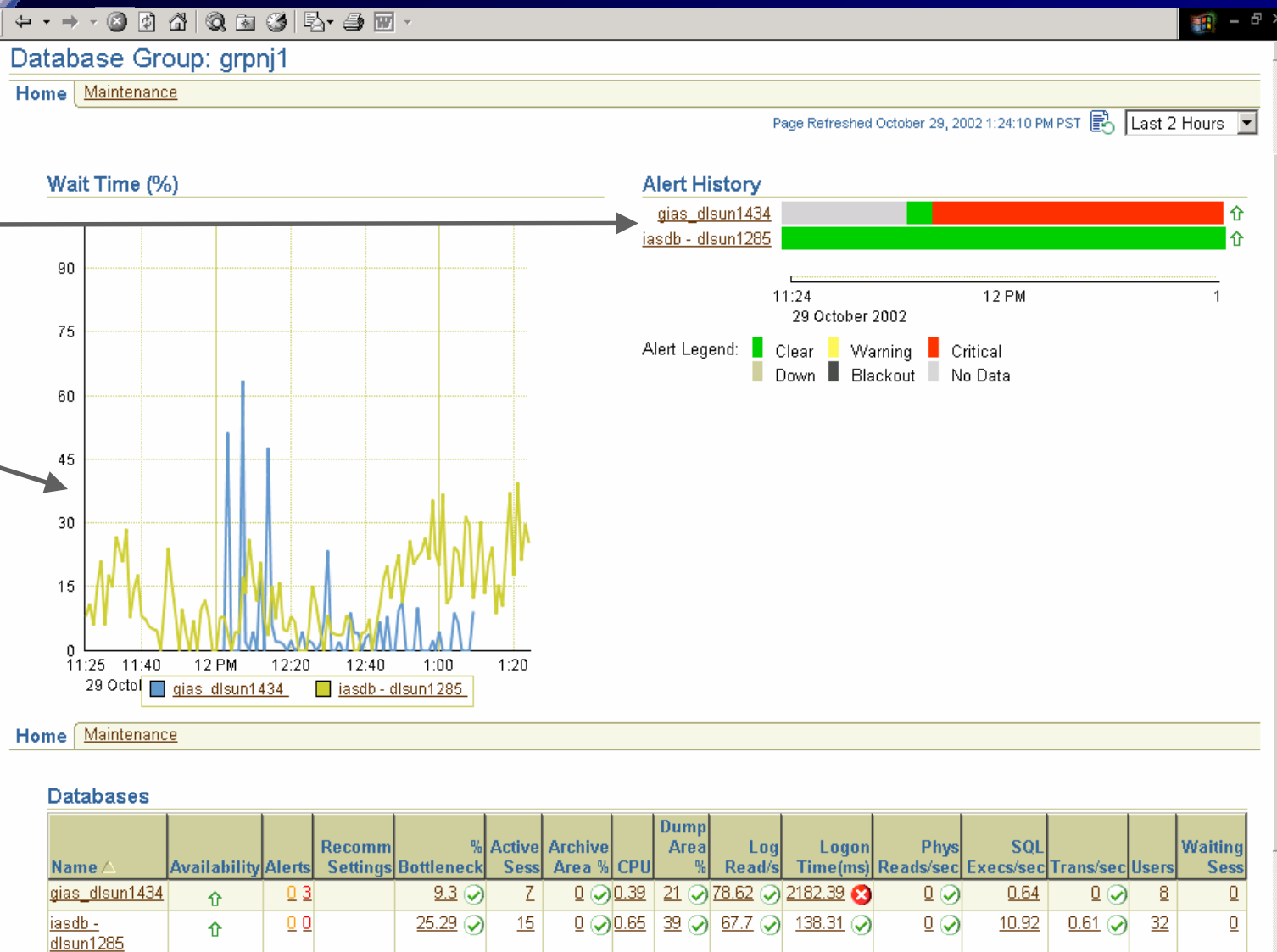




# Database Performance

Alert  
Issues

Major  
Waits





# Database Performance

Monitor  
Database

We have a  
CPU  
issue!

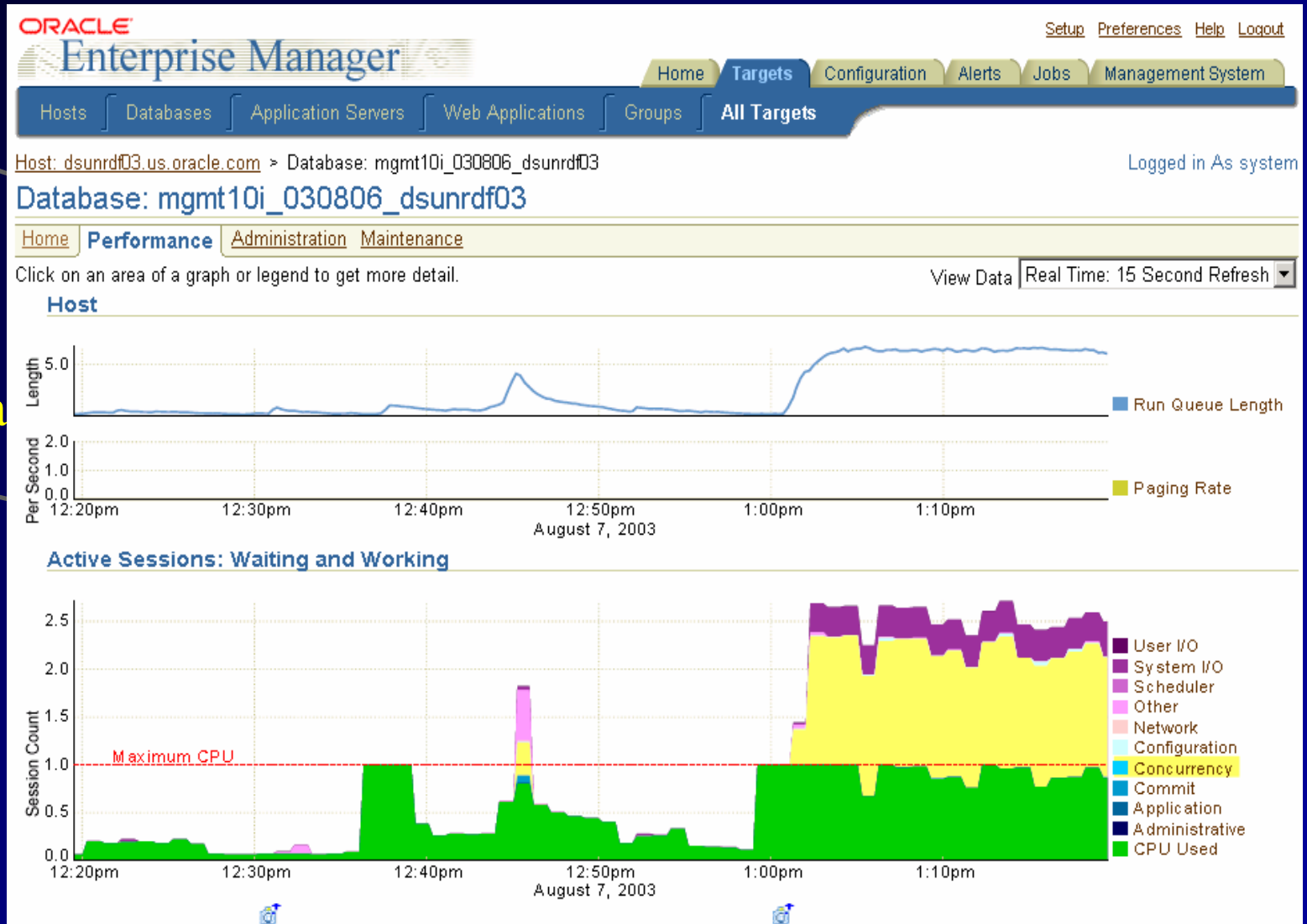




# Database Performance

Monitor  
Perform.

We have a  
Concurr.  
issue!

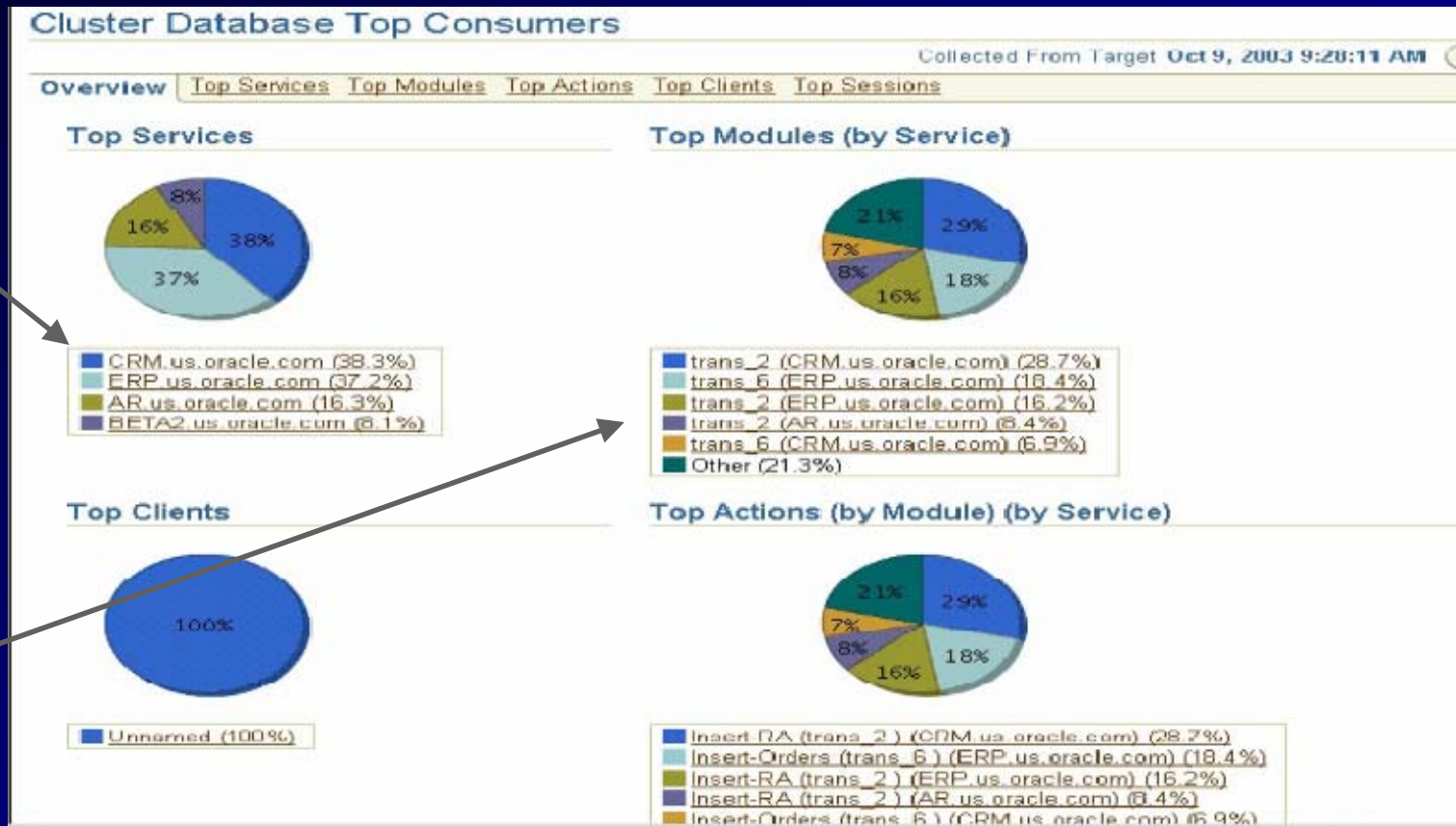




# Grid Services - Automatic Workload Management

Top Services

Top Modules



- Complete Presentation by Oracle's Erik Peterson at:
- <http://www.oracleracsig.org>



# 10g RAC Enhancements

- GRID Control
  - Allows for RAC instance startup, shutdown
  - Allows for RAC instance creation
  - Allows for resource reallocation based on SLAs
  - Allows for automatic provisioning when used with RAC, ASM and Linux



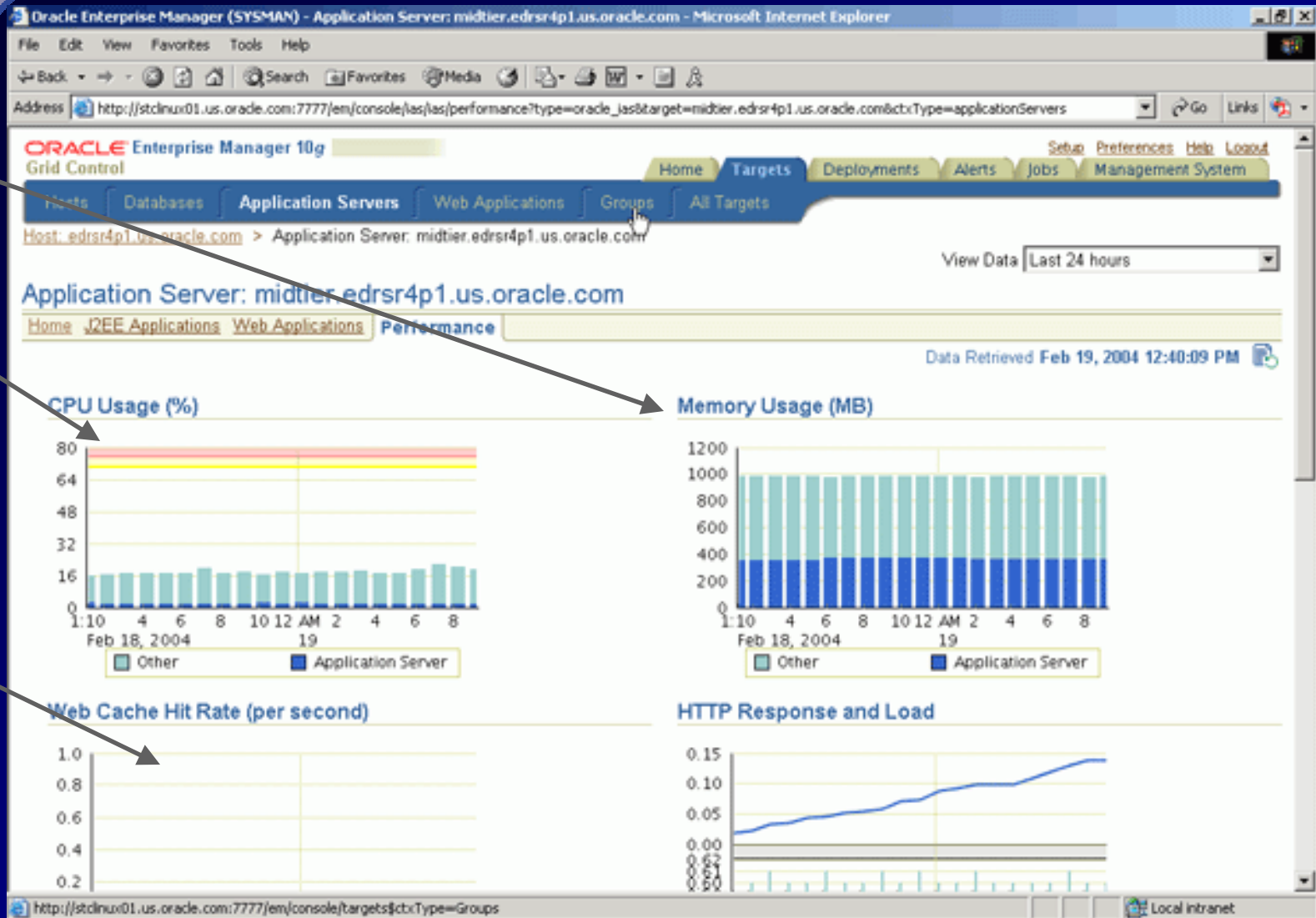


# App. Server Performance

Memory

CPU

Web  
Cache  
Hits

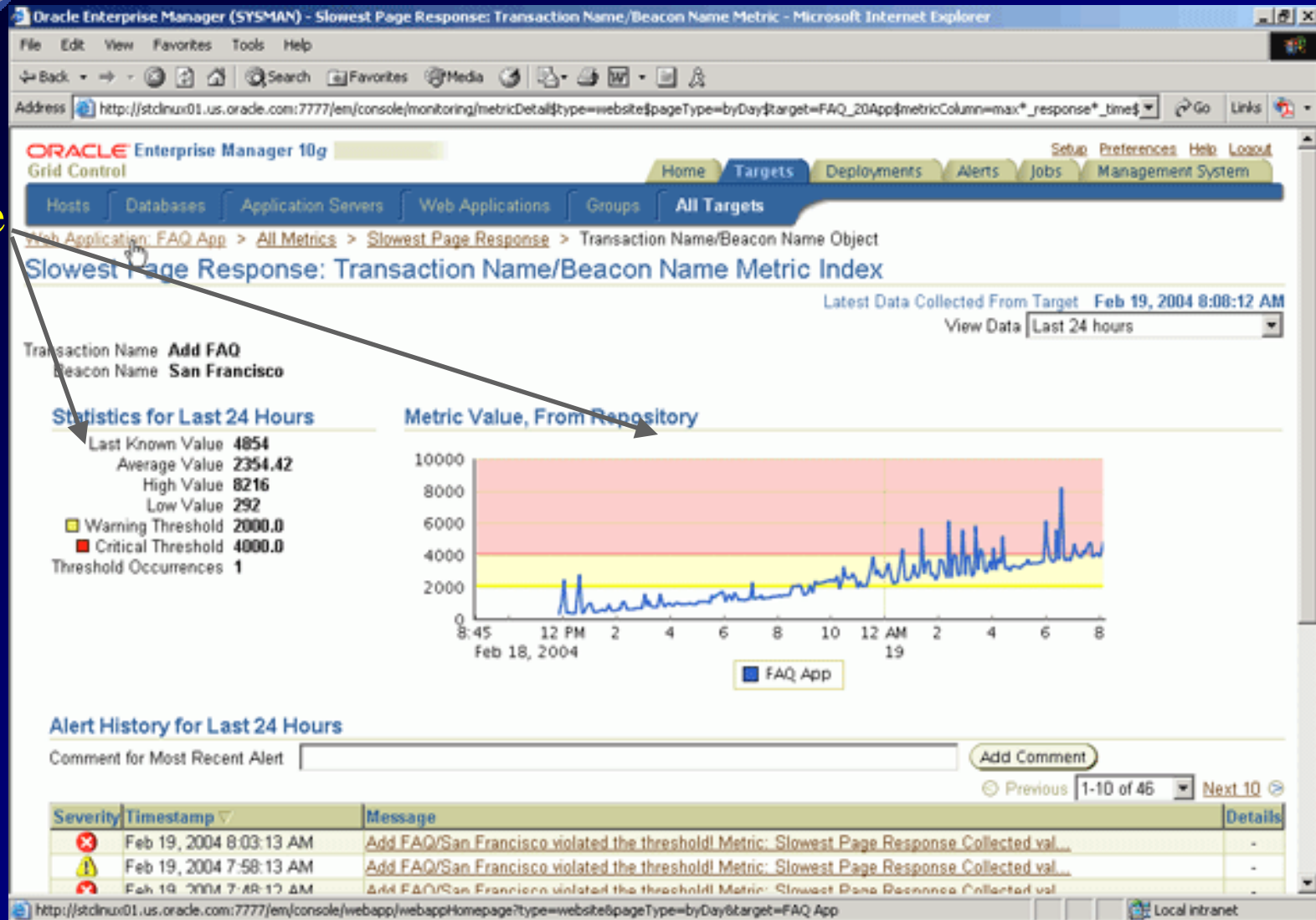






# View the Web Application

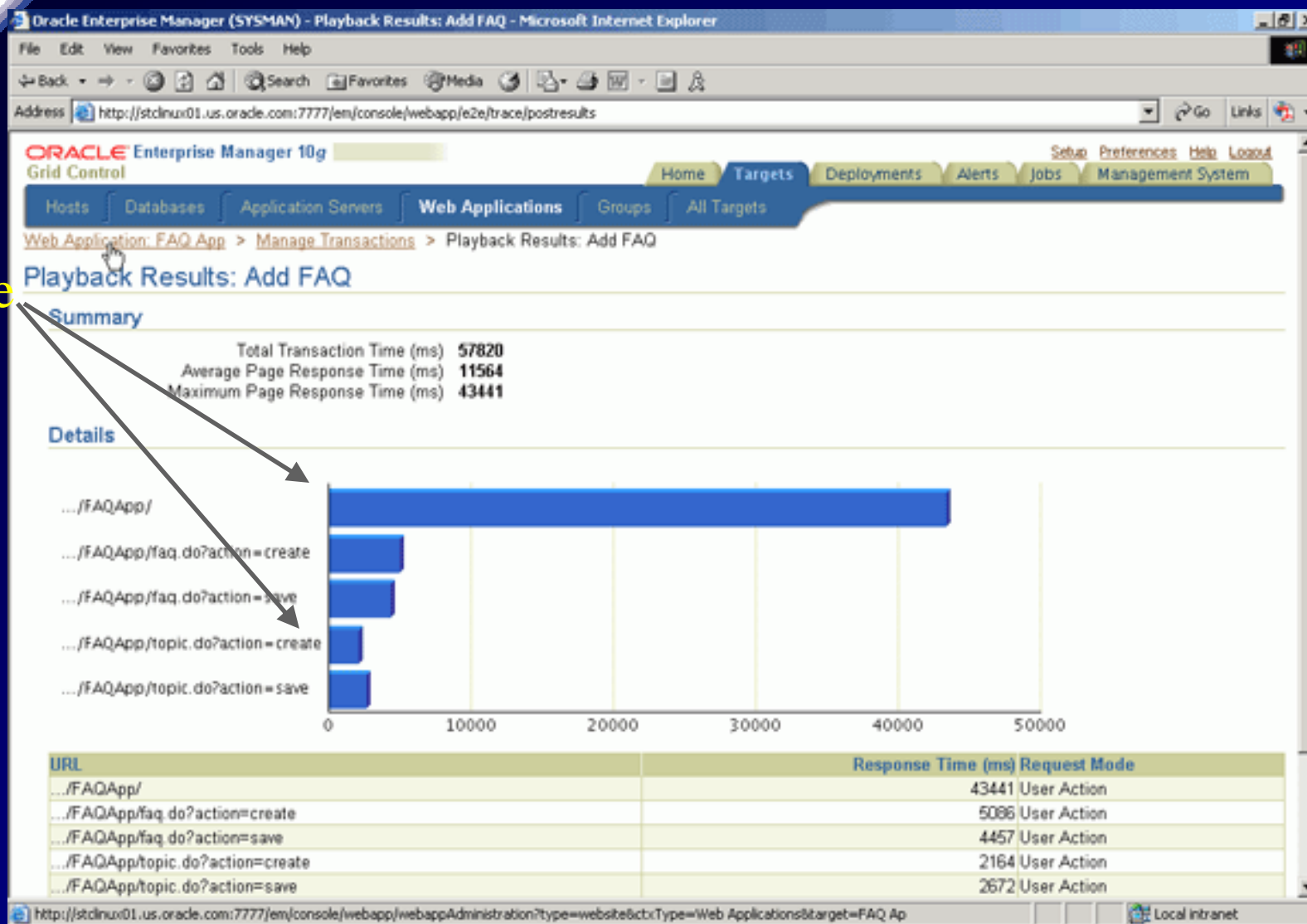
Page  
Response  
Time  
Detail

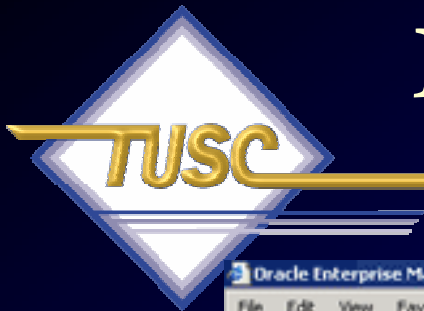




# URL Response Times

Web  
Page  
Response  
Time all  
URL's

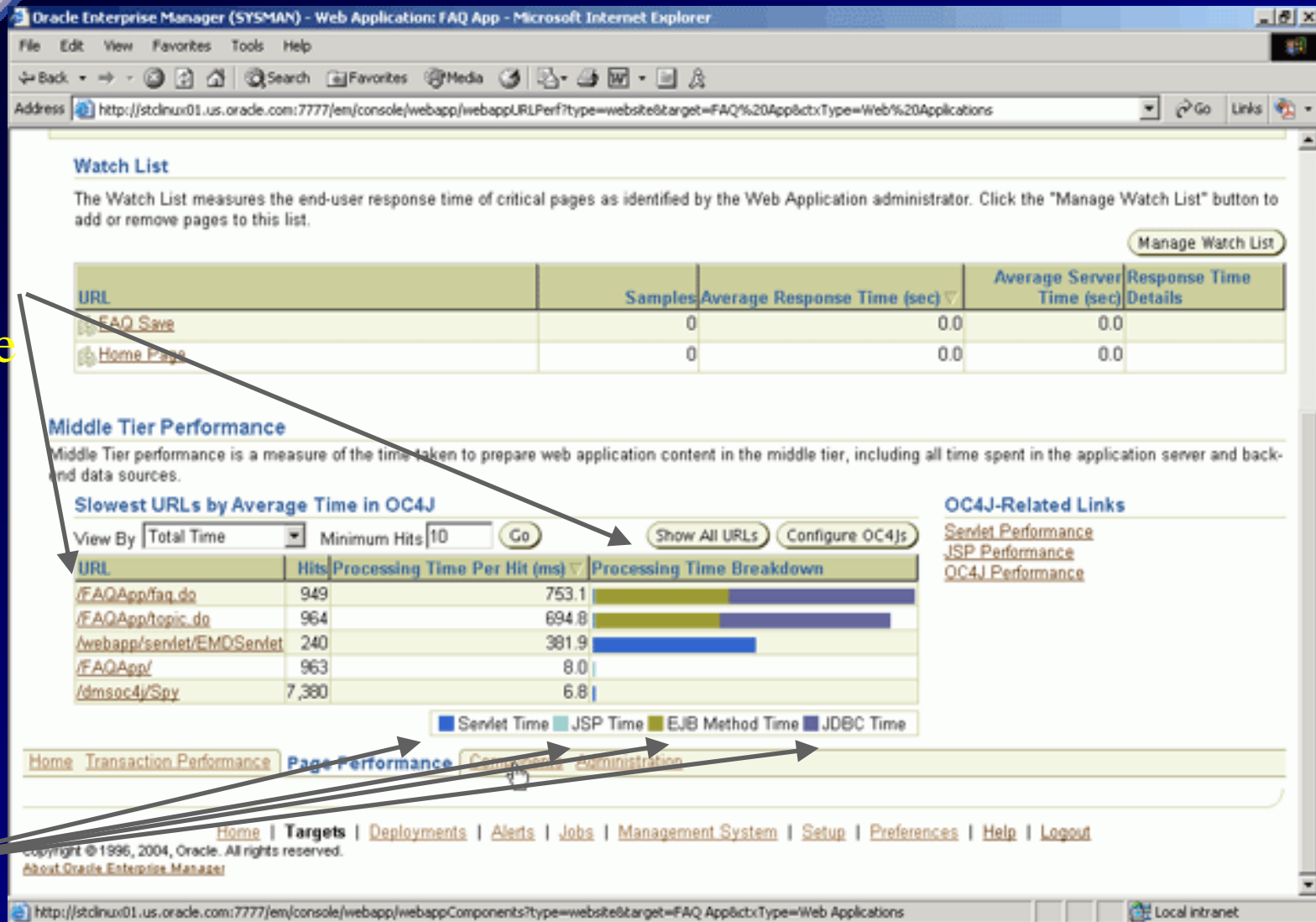




# Middle Tier Performance

Web  
Page  
Detail  
Response  
Time all  
URL's

Splits  
Time  
into  
Parts





# Host Performance

Side by  
Side  
Compare

Oracle Enterprise Manager 10g  
Grid Control

Home Targets Deployments Alerts Jobs Management System

Comparison Results Summary

First Host **edrsr4p1.us.oracle.com**  
Date Feb 18, 2004 12:51:27 PM

Second Host **edrsr11p1.us.oracle.com**  
Date Feb 19, 2004 2:04:56 AM

Hardware & Operating System

| Comparison Result | edrsr4p1.us.oracle.com                                              | edrsr11p1.us.oracle.com                                             |
|-------------------|---------------------------------------------------------------------|---------------------------------------------------------------------|
| Different         | i686, 1 CPUs, 1005.94140625 MB Memory                               | i686, 1 CPUs, 1005.94140625 MB Memory                               |
| Different         | Red Hat Enterprise Linux AS release 3 (Taroon) 2.4.21 4.EL (32-bit) | Red Hat Enterprise Linux AS release 3 (Taroon) 2.4.21 4.EL (32-bit) |

Oracle Software

Compare Products

| Product                                 | edrsr4p1.us.oracle.com | edrsr11p1.us.oracle.com |
|-----------------------------------------|------------------------|-------------------------|
| Additional Management Agent 10.1.0.2.0  | ✓                      | ✓                       |
| Oracle Application Server 10g 9.0.4.0.0 | ✓                      |                         |
| Oracle Database 10g 10.1.0.2.0          |                        | ✓                       |
| OracleAS Infrastructure 10g 9.0.4.0.0   | ✓                      |                         |

OS-Registered Software

Previous 1-10 of 686 Next 10

| Product         | Vendor        | edrsr4p1.us.oracle.com | edrsr11p1.us.oracle.com |
|-----------------|---------------|------------------------|-------------------------|
| 4Suite 0.11.1   | Red Hat, Inc. | ✓                      | ✓                       |
| a2ps 4.13b      | Red Hat, Inc. | ✓                      | ✓                       |
| acl 2.2.3       | Red Hat, Inc. | ✓                      | ✓                       |
| alhamiet 1.0.37 | Red Hat, Inc. | ✓                      | ✓                       |

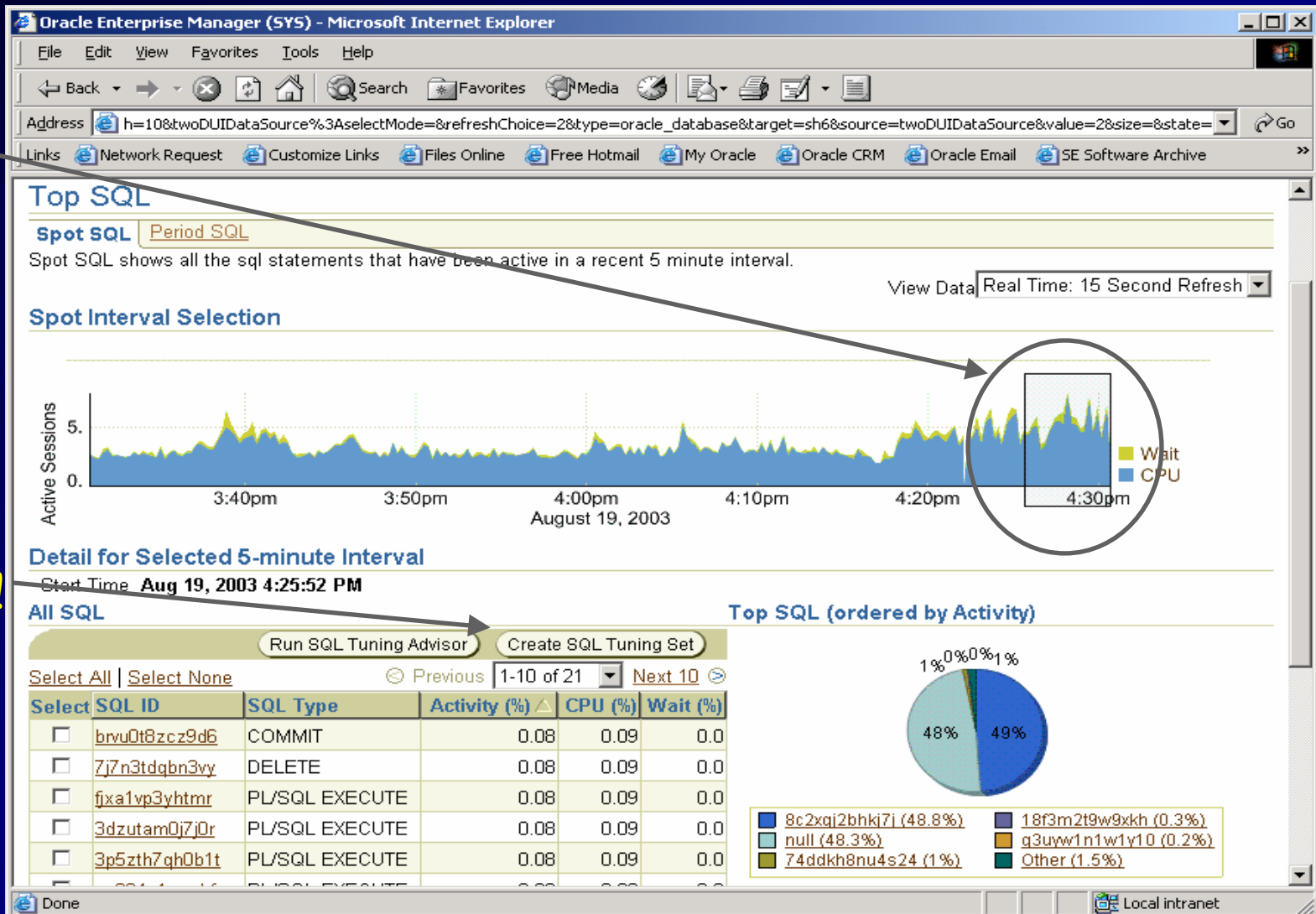
http://stdlinux01.us.oracle.com:7777/em/console/targets



# ADDM SQL Tuning Advisor

Drill into  
Top SQL  
for worst  
time  
period

Get Help!





# Using SQL Tuning Sets

TUSC

Tuning  
Results  
for this  
job

Suggests  
using a  
profile  
For this  
SQL

Oracle Enterprise Manager (SYSMAN) - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Links rap23

Address http://dsunrap23.us.oracle.com:7777/em/console/database/instance/sqltune?task\_id=395&event=view&advisoryCentralURL=/em/cons

Hosts Databases Application Servers Web Applications Groups All Targets

Host: as01.us.oracle.com > Database: as01\_db > Advisor Central > SQL Tuning Results: TASK\_00006

### SQL Tuning Results

|             |                         |              |                         |
|-------------|-------------------------|--------------|-------------------------|
| Task name   | TASK_00006              | Task status  | COMPLETED               |
| Tuning mode | COMPREHENSIVE           | Time limit   | 1800                    |
| SQL ID      | 8c2xqj2bhkj7j           | Running time | 4 seconds               |
| Started at  | Jun 13, 2003 7:33:42 PM | Completed at | Jun 13, 2003 7:33:46 PM |

#### Overview of recommendations

Select a statement and... [View Recommendations](#)

Previous 1-1 of 1 Next

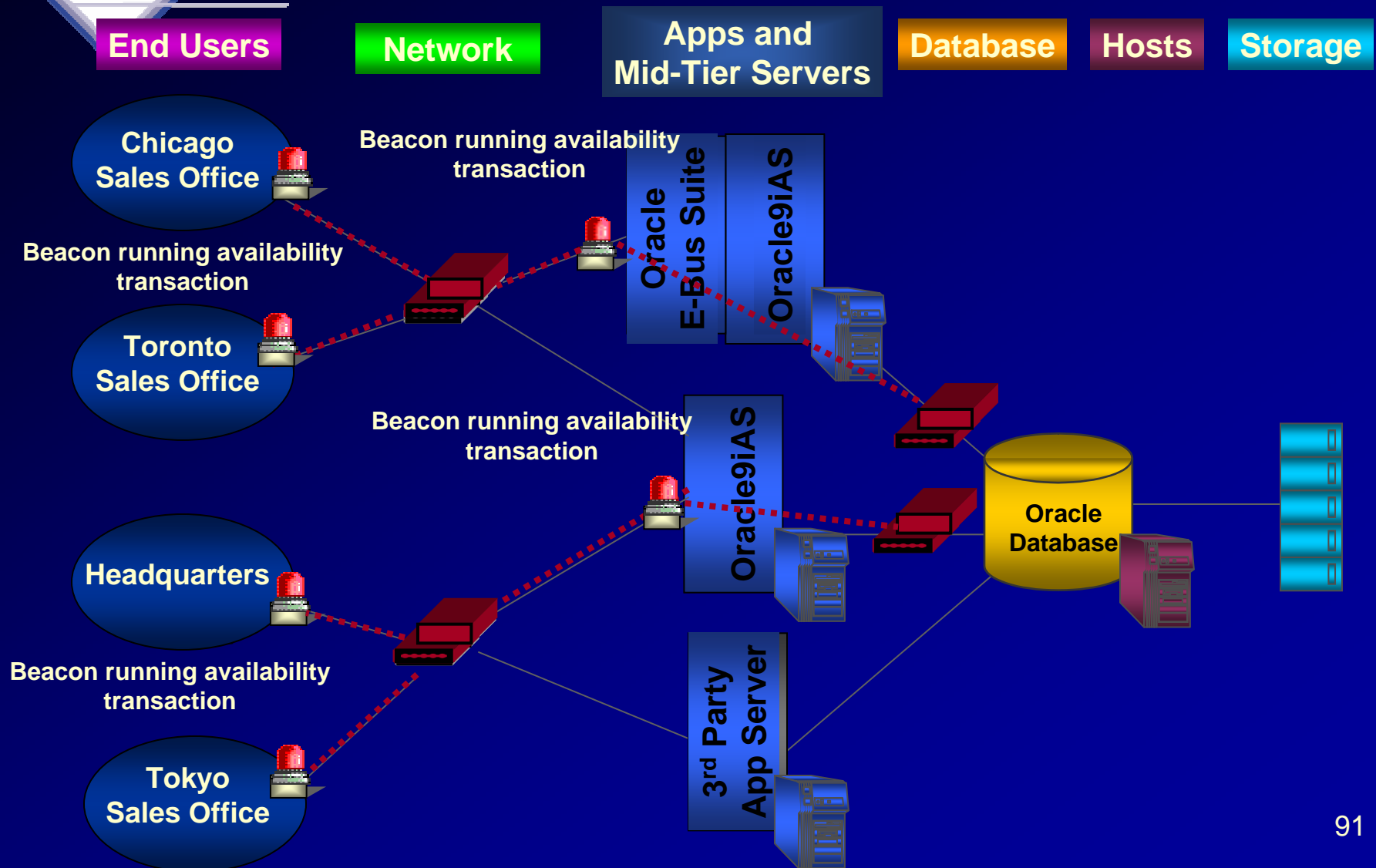
| Select                           | Parsing Schema | SQL Text                                                                 | Statistics | SQL Profile | Index | Rewrite | Misc | Error |
|----------------------------------|----------------|--------------------------------------------------------------------------|------------|-------------|-------|---------|------|-------|
| <input checked="" type="radio"/> |                | select time_id, QUANTITY_SOLD, AMOUNT_SOLD from sales s, customers c ... |            | ✓           |       |         |      |       |

Home Targets Configuration Alerts Jobs Management Setup Preferences Help Logout

Local intranet

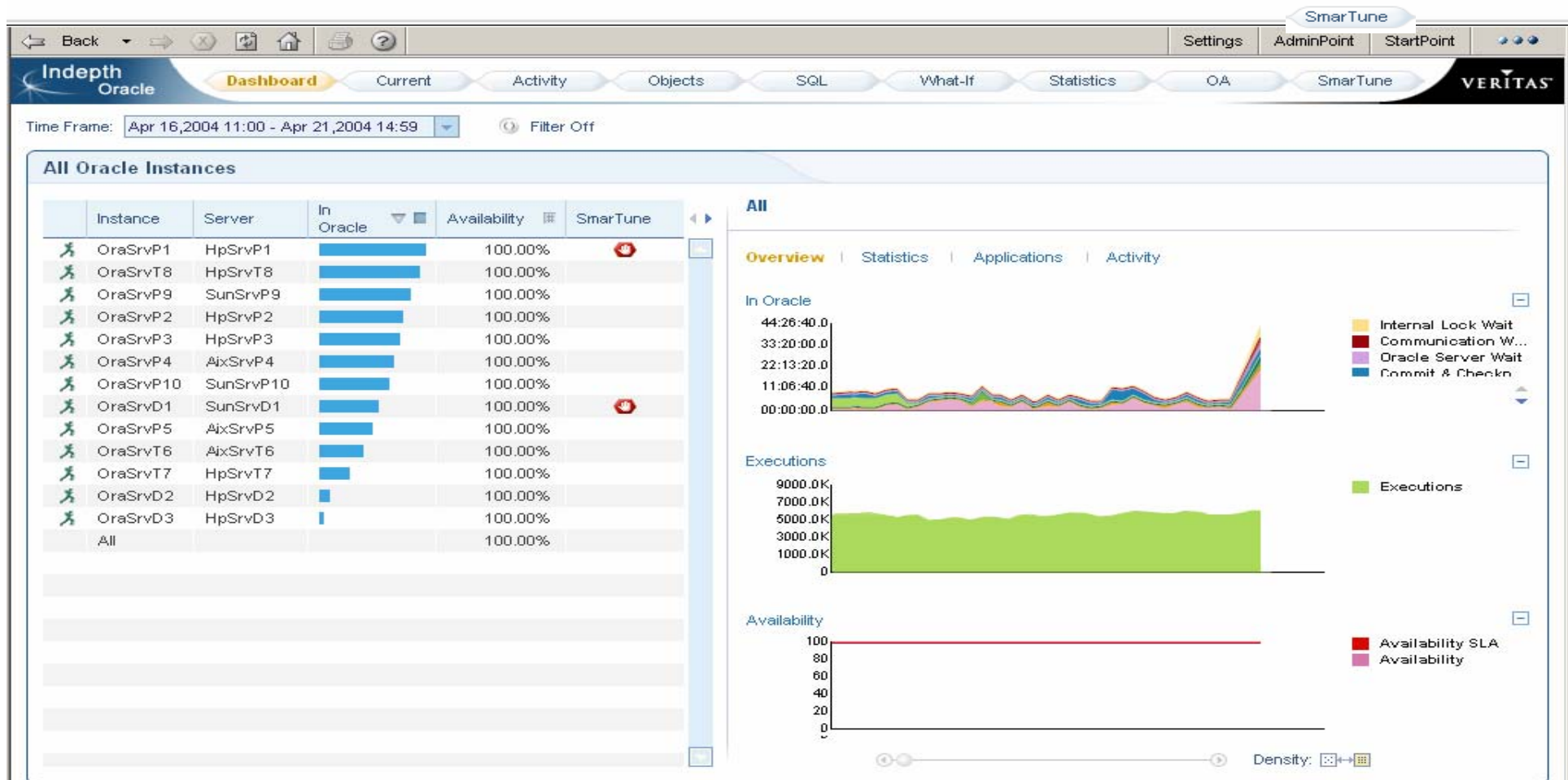


# Availability Monitoring Topology





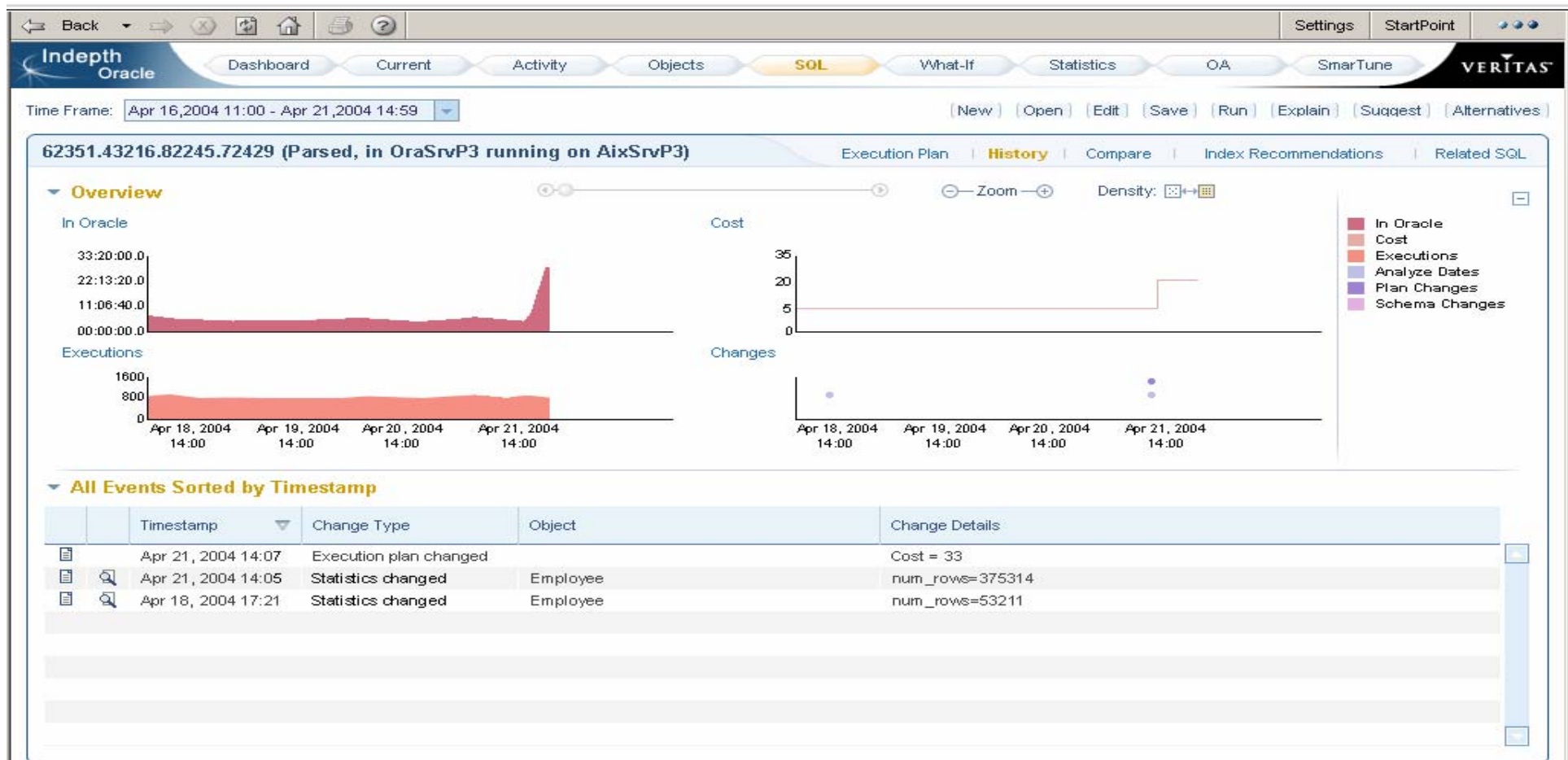
# 3<sup>rd</sup> party – Indepth Oracle







# Investigate Reaction To Changes





# Identify Solutions

Back

Settings

AdminPoint

StartPoint

Indepth Oracle

Dashboard

Current

Activity

Objects

SOL

What-If

Statistics

OA

SmartTune

VERITAS

New

Open

Edit

Save

Run

Explain

Suggest

Alternatives

62351.43216.82245.72429 (Parsed, in OraSrvP3 running on AixSrvP3)

Execution Plan

History

Compare

Index Recommendations

Related SQL

| Intensity | Table / Index   | Recommendation     | Columns     |
|-----------|-----------------|--------------------|-------------|
|           | SIGAL.DEPARTMEN | Create index(Norma | MGRNO       |
|           | SIGAL.EMPLOYEE  | Create index(Norma | EMPNO,BONUS |
|           | SIGAL.PROJECT   | Analyze table      |             |
|           | SIGAL.EMPLOYEE  | Analyze table      |             |
|           | SIGAL.DEPARTMEN | Analyze table      |             |

Create index(Normal) QA.DEPARTMENT

Highlights

Columns

DDL

Background

Original Statement:

Recommendation:

Table / Index:

Columns:

Created:

62351.43216.82245.72429

Create index(Normal)

SIGAL.DEPARTMENT

MGRNO

Apr 21, 2004 15:10

More Information

For background information about indexes and the role the play in performance tuning, switch to the

Quick Links

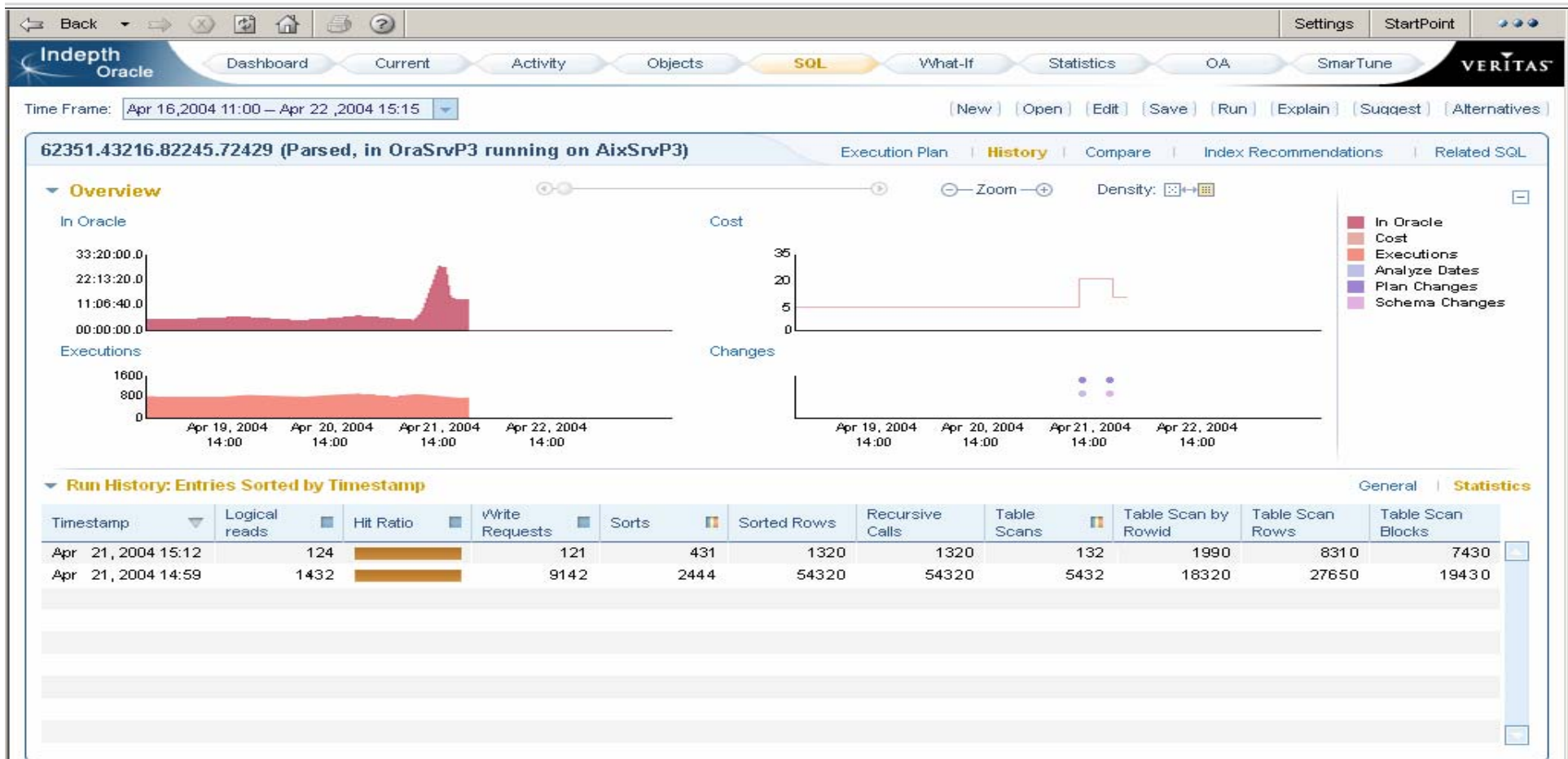
To examine the original statement's execution plan, click [here](#).

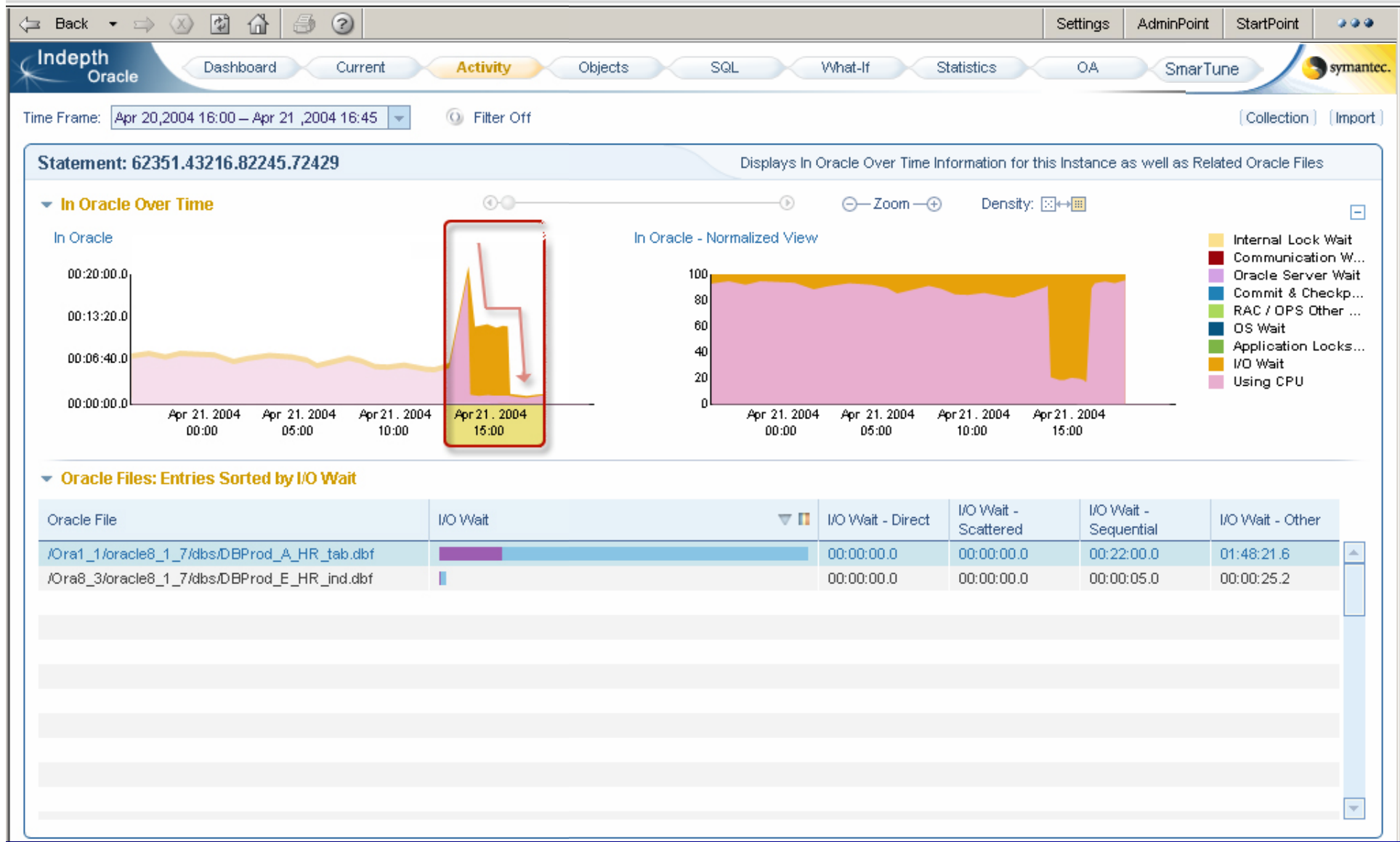
To examine the influence of this index definition on all the SQL statements in the system, click [here](#).

To examine the influence all the index recommendations on all the SQL statements in the system, click [here](#).



# Verify Solution







# Rank Findings

Back

Settings

AdminPoint

StartPoint

Indepth Oracle

Dashboard

Current

Activity

Objects

SQL

What-If

Statistics

SmarTune

symantec.

Time Frame: Jun 06,2004 12:00 - Jun 07,2004 12:59 Instance: OraSrvP1(HpSrvP1)

Findings

Changes | Instance

| Finding                                 | Rank | Potential Gain (%) | Occurrence |
|-----------------------------------------|------|--------------------|------------|
| Shared pool too small or not configured |      | 18.15%             | 307        |
| SQL*Net overhead due to application e   |      | 9.08%              | 302        |
| Insufficient shared server processes    |      | 7.43%              | 263        |
| Too many small transactions             |      | 5.78%              | 413        |
| Extent management contention for obje   |      | 5.45%              | 11         |
| Block ITL contention or unique index ke |      | 5.23%              | 12         |
| Hot block                               |      | 4.24%              | 1          |
| Too many hard parsed statements         |      | 3.54%              | 413        |
| Extent management contention during s   |      | 2.27%              | 415        |
| Inefficient cursor caching              |      | 2.21%              | 2          |

Shared pool too small or not configured properly

Highlights | Advice | Shared pool waits and statistics | Background

Your application is waiting 18.15% of the time for shared pool-related resources, 10.70% out of which are due to reloads.

More Information

To learn more about the situation and what can be done, click the [Advice tab](#).  
For background information about the shared pool and its role in performance tuning, click the [Background tab](#).

Quick Links

To examine the size of your shared pool, open the [Objects workspace](#).  
To examine how your application is waiting for shared pool-resources and how this is related to parsing, click the [Parse Waits and Statistics tab](#).



# Provide Expert Advice

Back

Settings

AdminPoint

StartPoint

Indepth Oracle

Dashboard

Current

Activity

Objects

SQL

What-If

Statistics

SmarTune

symantec.

Time Frame: Jun 06,2004 12:00 - Jun 07,2004 12:59 Instance: OraSrvP1(HpSrvP1)

Findings

Changes | Instance

| Finding                                 | Rank | Potential Gain (%) | Occurrence |
|-----------------------------------------|------|--------------------|------------|
| Shared pool too small or not configured |      | 18.15%             | 307        |
| SQL*Net overhead due to application e   |      | 9.08%              | 302        |
| Insufficient shared server processes    |      | 7.43%              | 263        |
| Too many small transactions             |      | 5.78%              | 413        |
| Extent management contention for obje   |      | 5.45%              | 11         |
| Block ITL contention or unique index ke |      | 5.23%              | 12         |
| Hot block                               |      | 4.24%              | 1          |
| Too many hard parsed statements         |      | 3.54%              | 413        |
| Extent management contention during s   |      | 2.27%              | 415        |
| Inefficient cursor caching              |      | 2.21%              | 2          |

Shared pool too small or not configured properly

Highlights | Advice | Shared pool waits and statistics | Background

To improve performance, choose the approach that best fits your needs. Following is a list of possible solutions:

- Increase your shared pool size by doing one of the following:
  - In Oracle 9i and higher, run the command `alter system set shared_pool_size=XXXXXX;`
  - Increase the shared\_pool\_size init parameter.
- Pin heavily used PL/SQL packages by running `dbms_shared_pool.keep('xxxxxx');`

**Note:** If you increase the total size of your SGA, make sure not to exceed the available physical memory of the Oracle server, to avoid paging.

Consider also one of the following actions:

- Make sure objects are not analyzed too frequently, unless there is a significant change in their content.
- Avoid using synonyms. Instead, do one of the following:
  - Use fully qualified names, such as APPS.TAB1, in your SQL statements.
  - Run the following command to make your application change from the current schema to the schema that owns all objects: `alter session set current_schema=APPS`

**See Also**

To examine the size of your shared pool, open the [Objects workspace](#).

To examine how your application is waiting for shared pool resources and how this is related to parsing, click the [Parse Waits and Statistics tab](#).





# Block Dumps

| Dec | Hx | Oct | Char                               | Dec | Hx | Oct | Html  | Chr          | Dec | Hx | Oct | Html  | Chr      | Dec | Hx | Oct | Html   | Chr        |
|-----|----|-----|------------------------------------|-----|----|-----|-------|--------------|-----|----|-----|-------|----------|-----|----|-----|--------|------------|
| 0   | 0  | 000 | <b>NUL</b> (null)                  | 32  | 20 | 040 | &#32; | <b>Space</b> | 64  | 40 | 100 | &#64; | <b>@</b> | 96  | 60 | 140 | &#96;  | <b>`</b>   |
| 1   | 1  | 001 | <b>SOH</b> (start of heading)      | 33  | 21 | 041 | &#33; | <b>!</b>     | 65  | 41 | 101 | &#65; | <b>A</b> | 97  | 61 | 141 | &#97;  | <b>a</b>   |
| 2   | 2  | 002 | <b>STX</b> (start of text)         | 34  | 22 | 042 | &#34; | <b>"</b>     | 66  | 42 | 102 | &#66; | <b>B</b> | 98  | 62 | 142 | &#98;  | <b>b</b>   |
| 3   | 3  | 003 | <b>ETX</b> (end of text)           | 35  | 23 | 043 | &#35; | <b>#</b>     | 67  | 43 | 103 | &#67; | <b>C</b> | 99  | 63 | 143 | &#99;  | <b>c</b>   |
| 4   | 4  | 004 | <b>EOT</b> (end of transmission)   | 36  | 24 | 044 | &#36; | <b>\$</b>    | 68  | 44 | 104 | &#68; | <b>D</b> | 100 | 64 | 144 | &#100; | <b>d</b>   |
| 5   | 5  | 005 | <b>ENQ</b> (enquiry)               | 37  | 25 | 045 | &#37; | <b>%</b>     | 69  | 45 | 105 | &#69; | <b>E</b> | 101 | 65 | 145 | &#101; | <b>e</b>   |
| 6   | 6  | 006 | <b>ACK</b> (acknowledge)           | 38  | 26 | 046 | &#38; | <b>&amp;</b> | 70  | 46 | 106 | &#70; | <b>F</b> | 102 | 66 | 146 | &#102; | <b>f</b>   |
| 7   | 7  | 007 | <b>BEL</b> (bell)                  | 39  | 27 | 047 | &#39; | <b>'</b>     | 71  | 47 | 107 | &#71; | <b>G</b> | 103 | 67 | 147 | &#103; | <b>g</b>   |
| 8   | 8  | 010 | <b>BS</b> (backspace)              | 40  | 28 | 050 | &#40; | <b>(</b>     | 72  | 48 | 110 | &#72; | <b>H</b> | 104 | 68 | 150 | &#104; | <b>h</b>   |
| 9   | 9  | 011 | <b>TAB</b> (horizontal tab)        | 41  | 29 | 051 | &#41; | <b>)</b>     | 73  | 49 | 111 | &#73; | <b>I</b> | 105 | 69 | 151 | &#105; | <b>i</b>   |
| 10  | A  | 012 | <b>LF</b> (NL line feed, new line) | 42  | 2A | 052 | &#42; | <b>*</b>     | 74  | 4A | 112 | &#74; | <b>J</b> | 106 | 6A | 152 | &#106; | <b>j</b>   |
| 11  | B  | 013 | <b>VT</b> (vertical tab)           | 43  | 2B | 053 | &#43; | <b>+</b>     | 75  | 4B | 113 | &#75; | <b>K</b> | 107 | 6B | 153 | &#107; | <b>k</b>   |
| 12  | C  | 014 | <b>FF</b> (NP form feed, new page) | 44  | 2C | 054 | &#44; | <b>,</b>     | 76  | 4C | 114 | &#76; | <b>L</b> | 108 | 6C | 154 | &#108; | <b>l</b>   |
| 13  | D  | 015 | <b>CR</b> (carriage return)        | 45  | 2D | 055 | &#45; | <b>-</b>     | 77  | 4D | 115 | &#77; | <b>M</b> | 109 | 6D | 155 | &#109; | <b>m</b>   |
| 14  | E  | 016 | <b>SO</b> (shift out)              | 46  | 2E | 056 | &#46; | <b>.</b>     | 78  | 4E | 116 | &#78; | <b>N</b> | 110 | 6E | 156 | &#110; | <b>n</b>   |
| 15  | F  | 017 | <b>SI</b> (shift in)               | 47  | 2F | 057 | &#47; | <b>/</b>     | 79  | 4F | 117 | &#79; | <b>O</b> | 111 | 6F | 157 | &#111; | <b>o</b>   |
| 16  | 10 | 020 | <b>DLE</b> (data link escape)      | 48  | 30 | 060 | &#48; | <b>0</b>     | 80  | 50 | 120 | &#80; | <b>P</b> | 112 | 70 | 160 | &#112; | <b>p</b>   |
| 17  | 11 | 021 | <b>DC1</b> (device control 1)      | 49  | 31 | 061 | &#49; | <b>1</b>     | 81  | 51 | 121 | &#81; | <b>Q</b> | 113 | 71 | 161 | &#113; | <b>q</b>   |
| 18  | 12 | 022 | <b>DC2</b> (device control 2)      | 50  | 32 | 062 | &#50; | <b>2</b>     | 82  | 52 | 122 | &#82; | <b>R</b> | 114 | 72 | 162 | &#114; | <b>r</b>   |
| 19  | 13 | 023 | <b>DC3</b> (device control 3)      | 51  | 33 | 063 | &#51; | <b>3</b>     | 83  | 53 | 123 | &#83; | <b>S</b> | 115 | 73 | 163 | &#115; | <b>s</b>   |
| 20  | 14 | 024 | <b>DC4</b> (device control 4)      | 52  | 34 | 064 | &#52; | <b>4</b>     | 84  | 54 | 124 | &#84; | <b>T</b> | 116 | 74 | 164 | &#116; | <b>t</b>   |
| 21  | 15 | 025 | <b>NAK</b> (negative acknowledge)  | 53  | 35 | 065 | &#53; | <b>5</b>     | 85  | 55 | 125 | &#85; | <b>U</b> | 117 | 75 | 165 | &#117; | <b>u</b>   |
| 22  | 16 | 026 | <b>SYN</b> (synchronous idle)      | 54  | 36 | 066 | &#54; | <b>6</b>     | 86  | 56 | 126 | &#86; | <b>V</b> | 118 | 76 | 166 | &#118; | <b>v</b>   |
| 23  | 17 | 027 | <b>ETB</b> (end of trans. block)   | 55  | 37 | 067 | &#55; | <b>7</b>     | 87  | 57 | 127 | &#87; | <b>W</b> | 119 | 77 | 167 | &#119; | <b>w</b>   |
| 24  | 18 | 030 | <b>CAN</b> (cancel)                | 56  | 38 | 070 | &#56; | <b>8</b>     | 88  | 58 | 130 | &#88; | <b>X</b> | 120 | 78 | 170 | &#120; | <b>x</b>   |
| 25  | 19 | 031 | <b>EM</b> (end of medium)          | 57  | 39 | 071 | &#57; | <b>9</b>     | 89  | 59 | 131 | &#89; | <b>Y</b> | 121 | 79 | 171 | &#121; | <b>y</b>   |
| 26  | 1A | 032 | <b>SUB</b> (substitute)            | 58  | 3A | 072 | &#58; | <b>:</b>     | 90  | 5A | 132 | &#90; | <b>Z</b> | 122 | 7A | 172 | &#122; | <b>z</b>   |
| 27  | 1B | 033 | <b>ESC</b> (escape)                | 59  | 3B | 073 | &#59; | <b>;</b>     | 91  | 5B | 133 | &#91; | <b>[</b> | 123 | 7B | 173 | &#123; | <b>{</b>   |
| 28  | 1C | 034 | <b>FS</b> (file separator)         | 60  | 3C | 074 | &#60; | <b>&lt;</b>  | 92  | 5C | 134 | &#92; | <b>\</b> | 124 | 7C | 174 | &#124; | <b> </b>   |
| 29  | 1D | 035 | <b>GS</b> (group separator)        | 61  | 3D | 075 | &#61; | <b>=</b>     | 93  | 5D | 135 | &#93; | <b>]</b> | 125 | 7D | 175 | &#125; | <b>}</b>   |
| 30  | 1E | 036 | <b>RS</b> (record separator)       | 62  | 3E | 076 | &#62; | <b>&gt;</b>  | 94  | 5E | 136 | &#94; | <b>^</b> | 126 | 7E | 176 | &#126; | <b>~</b>   |
| 31  | 1F | 037 | <b>US</b> (unit separator)         | 63  | 3F | 077 | &#63; | <b>?</b>     | 95  | 5F | 137 | &#95; | <b>_</b> | 127 | 7F | 177 | &#127; | <b>DEL</b> |

|   | A | E | I | O | U | A2 | O2 | U2 |
|---|---|---|---|---|---|----|----|----|
|   | Ɔ | Ɔ | Ɔ | Ɔ | Ɔ | Ɔ  |    |    |
| D | Ɔ | Ɔ | Ɔ |   | Ɔ |    |    |    |
| J | Ɔ | Ɔ |   |   | Ɔ |    |    |    |
| K | Ɔ | Ɔ | Ɔ | Ɔ | Ɔ |    |    |    |
| M | Ɔ | Ɔ | Ɔ |   |   |    |    |    |
| N | Ɔ | Ɔ | Ɔ |   | Ɔ |    |    |    |
| P | Ɔ |   | Ɔ | Ɔ | Ɔ | Ɔ  |    | Ɔ  |
| Q | Ɔ | Ɔ | Ɔ |   |   |    |    |    |
| R | Ɔ | Ɔ | Ɔ | Ɔ | Ɔ | Ɔ  |    |    |
| S | Ɔ | Ɔ | Ɔ |   | Ɔ |    |    |    |
| T | Ɔ | Ɔ | Ɔ | Ɔ | Ɔ | Ɔ  |    |    |
| W | Ɔ |   | Ɔ |   |   |    |    |    |
| Z | Ɔ | Ɔ |   | Ɔ | Ɔ |    |    |    |

Other symbols

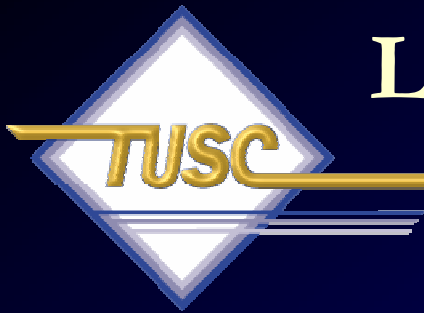


au (?) pi (?) pa (?) twe (?)

Unclassified symbols



Source: [www.LookupTables.com](http://www.LookupTables.com)

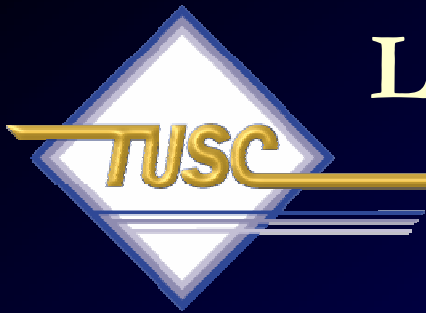


## Last Resort - Block Dumps

```
select file_id, block_id, blocks  
from dba_extents  
where segment_name = 'EMP'  
and owner = 'SCOTT';
```

| FILE_ID | BLOCK_ID | BLOCKS |
|---------|----------|--------|
| -----   | -----    | -----  |
| 1       | 50465    | 3      |





## Last Resort - Block Dumps

```
ALTER SYSTEM DUMP DATAFILE 5 BLOCK 50465
```

```
/
```

```
ALTER SYSTEM DUMP DATAFILE 5 BLOCK 50466
```

```
/
```

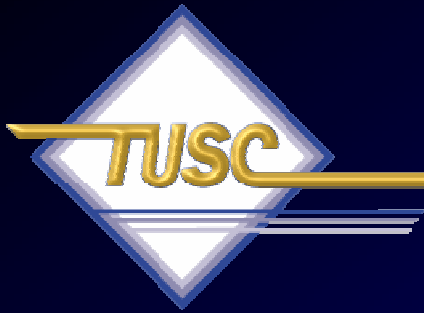
```
ALTER SYSTEM DUMP DATAFILE 5 BLOCK 50467
```

```
/
```

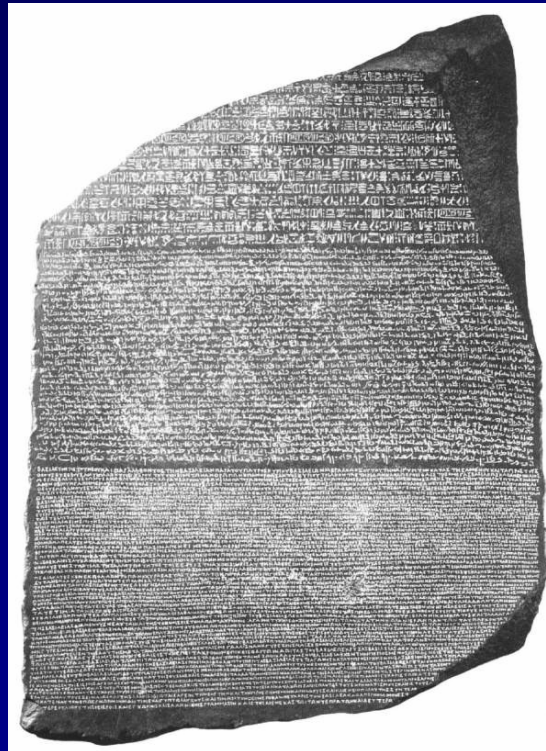
Or...

```
ALTER SYSTEM DUMP DATAFILE 5 BLOCK MIN  
50465 BLOCK MAX 50467;
```

*(Puts output in user\_dump\_dest)*



# Block Dump: Data Section





# Last Resort - Block Dumps

```
select *  
from emp  
where ename = 'MILLER';
```

| <i>EMPNO</i> | <i>ENAME</i> | <i>JOB</i> | <i>MGR</i> | <i>HIREDATE</i> | <i>SAL</i> | <i>COMM</i> | <i>DEPTNO</i> |
|--------------|--------------|------------|------------|-----------------|------------|-------------|---------------|
| 7934         | MILLER       | CLERK      | 7782       | 23-JAN-82       | 1300       |             | 10            |



## Block Dumps – output from udump

tab 0, row 13, @0x1b0b

tl: 39 fb: --H-FL-- lb: 0x0 cc: 8

col 0: [ 3] c2 50 23

col 1: [ 6] 4d 49 4c 4c 45 52

col 2: [ 5] 43 4c 45 52 4b

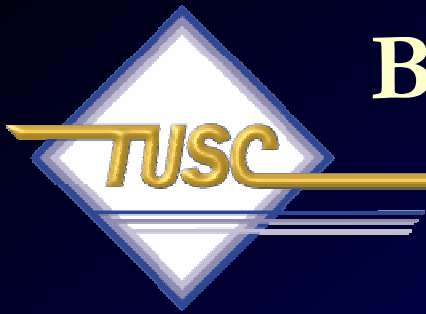
col 3: [ 3] c2 4e 53

col 4: [ 7] 77 b6 01 17 01 01 01

col 5: [ 2] c2 0e

col 6: \*NULL\*

col 7: [ 2] c1 0b...



## Block Dumps – Data Section

DUMP OUTPUT – EMPNO:

col 0: [ 3] c2 50 23

Hex to Decimal:

Col0 = EMPNO = 7934

50 (Hex) = 80 (Decimal) – 1 = 79

23 (Hex) = 35 (Decimal) – 1 = 34

c2: Number in the thousands (c2 is exponent)



## Block Dumps – Data Section

DUMP OUTPUT - ENAME:

col 1: [ 6] 4d 49 4c 4c 45 52

Hex to Character:

Col1 = ENAME = MILLER

4d (Hex) = M (Character)

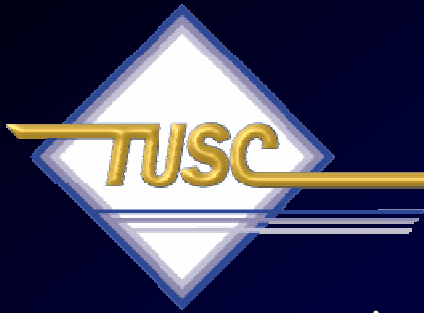
49 (Hex) = I (Character)

4c (Hex) = L (Character)

4c (Hex) = L (Character)

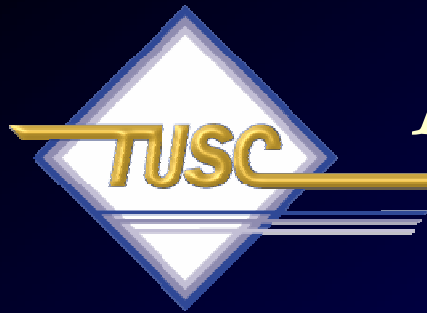
45 (Hex) = E (Character)

52 (Hex) = R (Character)



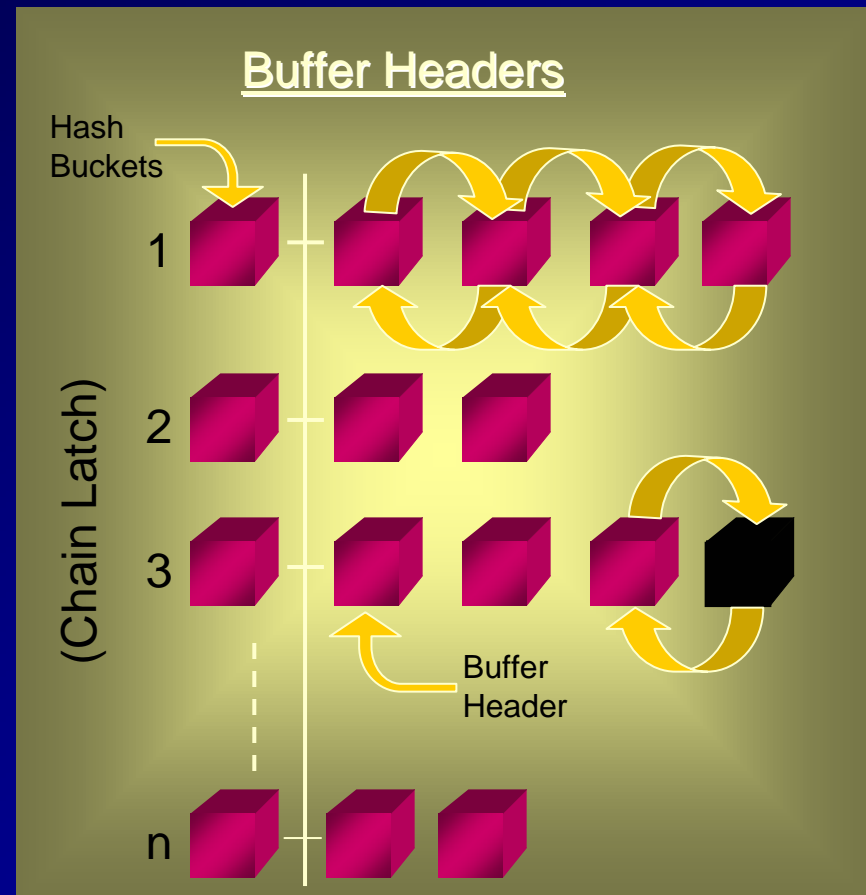
# Transactions Moving through Oracle: *ITL* & *Undo Blocks*



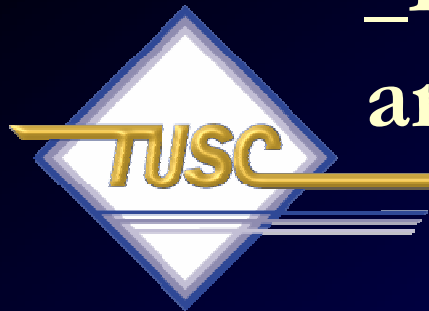


# Working with Hash Buckets And Buffer Headers (not buffers)

- Users asks for a specific data block address.
- This is hashed with a hashing algorithm and placed in the hash bucket that it hashes to.
- It walks the hash chain using the cache buffers chain latch to find the block that it needs (curr or cr).
- There can be many versions of each block







# `_DB_BLOCK_HASH_BUCKETS` and hashing data block addresses

Example: `_DB_BLOCK_HASH_BUCKETS`

*(shouldn't have to change this in Oracle9i or 10g)*

- Buffer hash table (x\$bh) has all buffer headers for all db\_block buffers.
- Buffer header ties to memory base address of the buffer.
- Buckets usually set to  $\text{Prime}(2 * \text{db\_block\_buffers})$
- A prime number is often used to avoid hashing anomalies
- Objects dba (class) is hashed to a hash bucket on the hash chain
- Get enough hash buckets (`_db_block_hash_buckets`)
- Blocks assigned to a hash bucket and onto the hash chain
- Could have multiple blocks hashed to same chain (if both hot-issues)
- Can have multiple versions of a block on same chain
- When block is replaced (based on LRU chain) new block comes in and could be (probably will be) hashed to a different hash chain.



## Query all buffer headers (state):

*col status for a6*

*select state,*

*decode(state, 0, 'FREE',*

*1, 'XCUR',*

*2, 'SCUR',*

*3, 'CR',*

*4, 'READ',*

*5, 'MREC',*

*6, 'IREC',*

*7, 'WRITE',*

*8, 'PIN') status, count(\*)*

*/\* not currently is use \*/*

*/\* held exclusive by this instance \*/*

*/\* held shared by this instance \*/*

*/\* only valid for consistent read \*/*

*/\* is being read from disk \*/*

*/\* in media recovery mode \*/*

*/\* in instance(crash) recovery mode \*/*

*/\* being written \*/*

*/\* pinned \*/*

*from x\$bh*

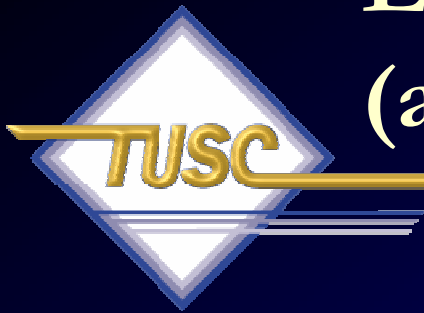
*group by state;*

*STATE STATUS COUNT(\*)*

*-----*

*1 XCUR 2001*

*3 CR 3*

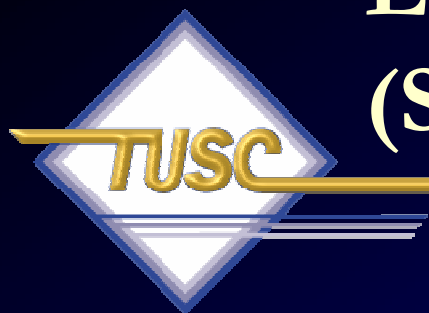


# EMP1 is Block#: 56650 (all rows are in this block)

```
select rowid,empno,
       dbms_rowid.rowid_relative_fno(rowid) fileno,
       dbms_rowid.rowid_block_number(rowid) blockno,
       dbms_rowid.rowid_row_number(rowid) rowno, rownum,
       rpad(to_char(dbms_rowid.rowid_block_number(rowid), 'FM0xxxxxxxx') || '.' ||
            to_char(dbms_rowid.rowid_row_number(rowid), 'FM0xxx') ||
            to_char(dbms_rowid.rowid_relative_fno(rowid), 'FM0xxx'), 18) myrid
from emp1;
```

| ROWID                                    | EMPNO | FILENO | BLOCKNO | ROWNO | ROWNUM |
|------------------------------------------|-------|--------|---------|-------|--------|
| -----                                    | ----- | -----  | -----   | ----- | -----  |
| MYRID                                    |       |        |         |       |        |
| -----                                    |       |        |         |       |        |
| AAAM4cAABAAAN1KAAA<br>0000dd4a.0000.0001 | 7369  | 1      | 56650   | 0     | 1      |
| AAAM4cAABAAAN1KAAB<br>0000dd4a.0001.0001 | 7499  | 1      | 56650   | 1     | 2      |
| ...                                      |       |        |         |       |        |
| ...                                      |       |        |         |       |        |
| AAAM4cAABAAAN1KAAN<br>0000dd4a.000d.0001 | 7934  | 1      | 56650   | 13    | 14     |

14 rows selected.



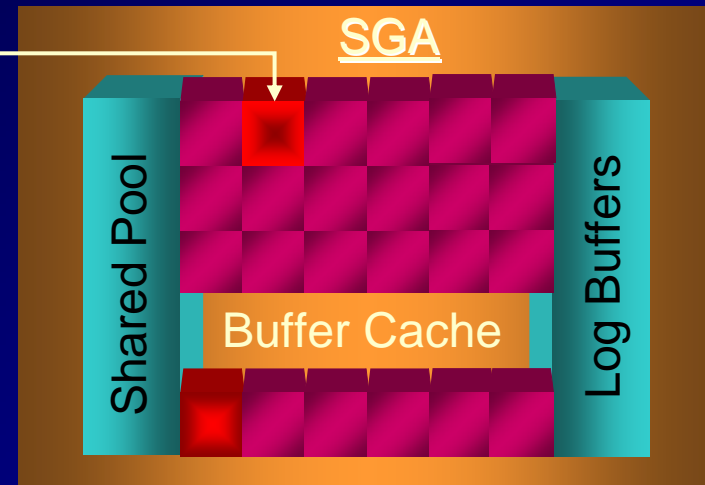
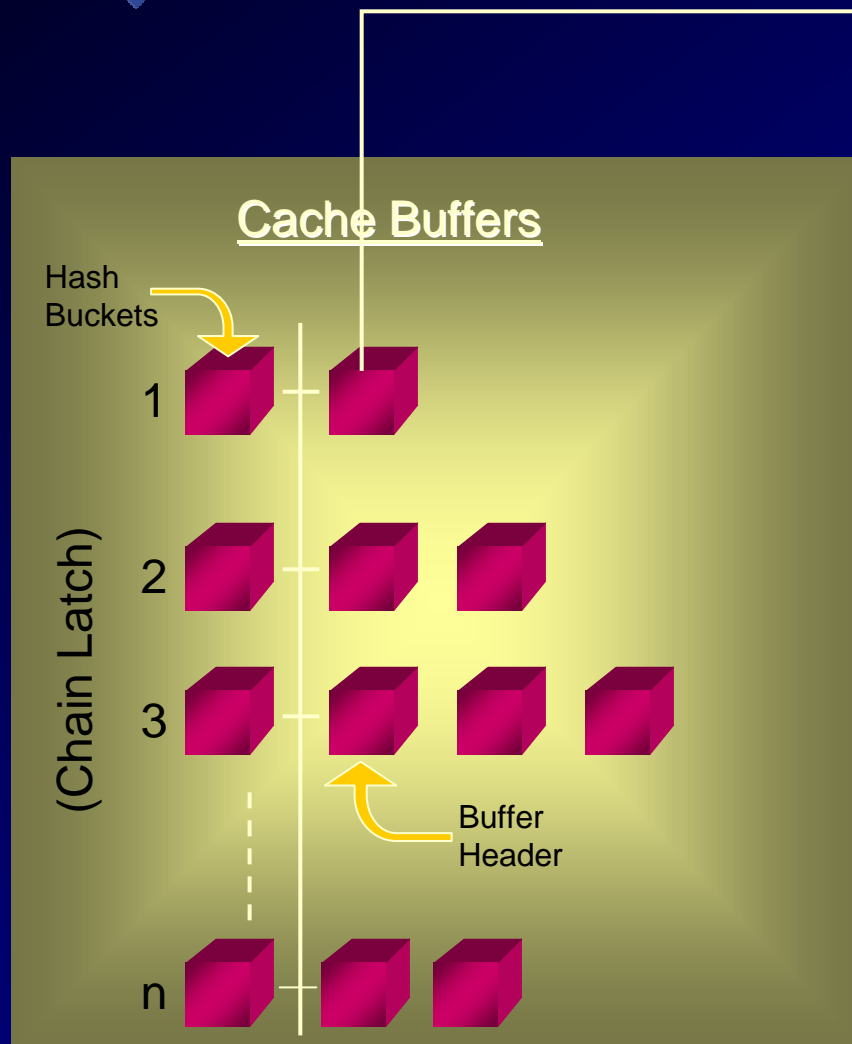
# Let's watch the EMP1 buffer header (So far it's clean and only 1 copy)

```

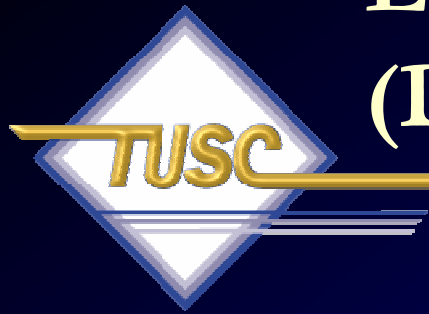
select    lrba_seq, state, dbarfil, dbablk, tch, flag, hscn_bas, cr_scn_bas,
          decode(bitand(flag,1), 0, 'N', 'Y') dirty,    /* Dirty bit */
          decode(bitand(flag,16), 0, 'N', 'Y') temp,    /* temporary bit */
          decode(bitand(flag,1536), 0, 'N', 'Y') ping, /* ping (to shared or null) bit */
          decode(bitand(flag,16384), 0, 'N', 'Y') stale, /* stale bit */
          decode(bitand(flag,65536), 0, 'N', 'Y') direct, /* direct access bit */
          decode(bitand(flag,1048576), 0, 'N', 'Y') new /* new bit */
from      x$bh
where     dbablk = 56650
order by dbablk;

```

| LRBA_SEQ   | STATE       | DBARFIL | DBABLK | TCH   | FLAG     | HSCN_BAS   |
|------------|-------------|---------|--------|-------|----------|------------|
| -----      | -----       | -----   | -----  | ----- | -----    | -----      |
| CR_SCN_BAS | D T P S D N |         |        |       |          |            |
| -----      | - - - - -   |         |        |       |          |            |
| 0          | 1           | 1       | 56650  | 0     | 35659776 | 4294967295 |
| 0          | N N N N N N |         |        |       |          |            |



*Only ONE block  
on the Hash  
Chain!*



## Let's watch the EMP1 buffer header (Delete a row)

*delete from emp1  
where comm = 0;*

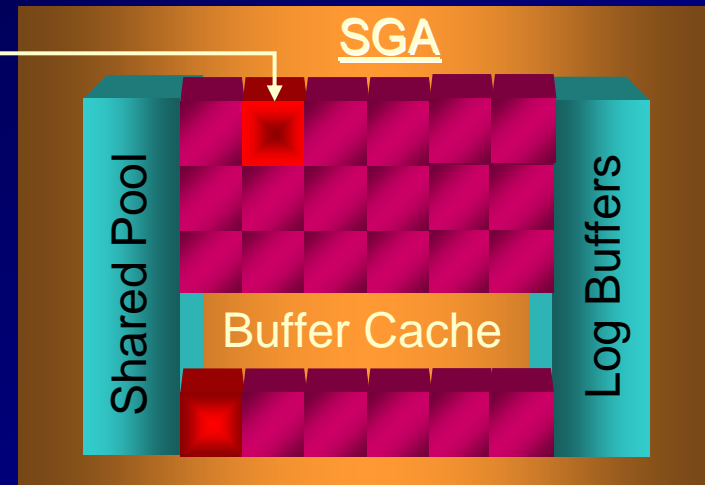
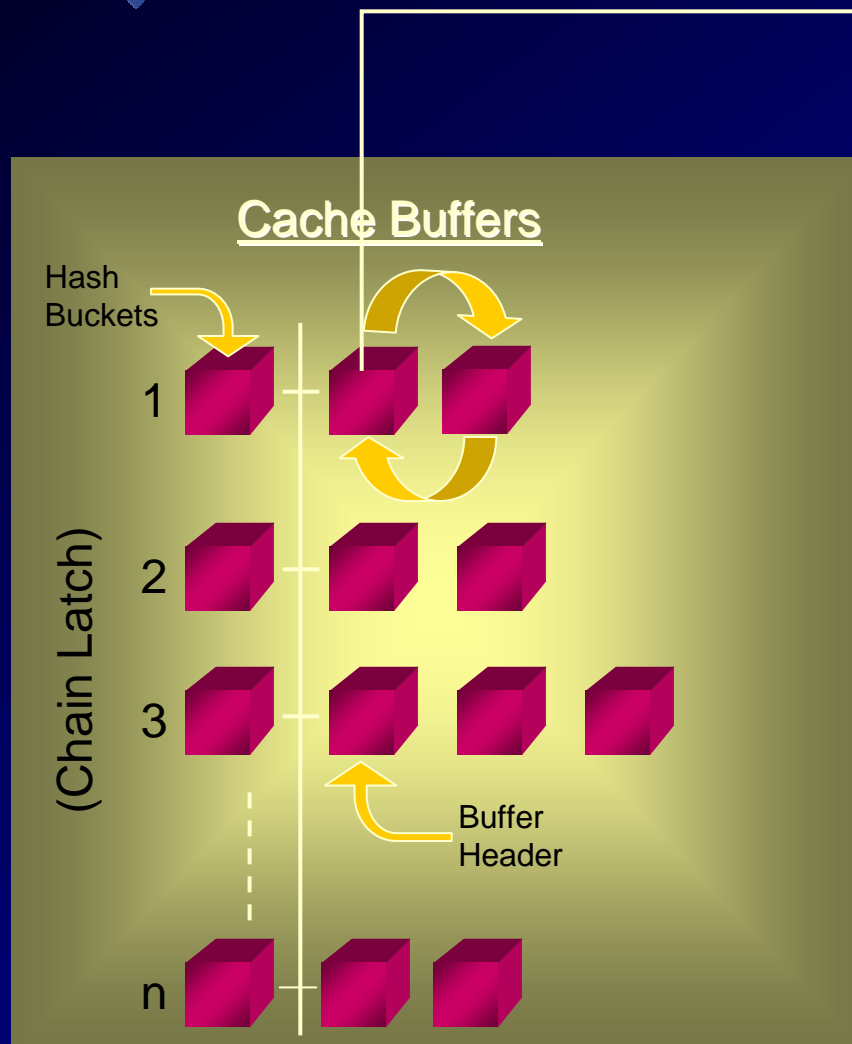
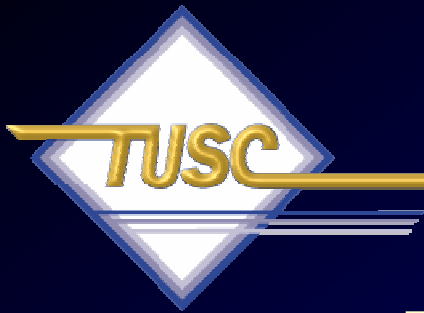
*one row deleted.*



# Let's watch the EMP1 buffer header (Make some changes 2 copies)

```
select  lrba_seq, state, dbarfil, dbablk, tch, flag, hscn_bas, cr_scn_bas,
        decode(bitand(flag,1), 0, 'N', 'Y') dirty,    /* Dirty bit */
        decode(bitand(flag,16), 0, 'N', 'Y') temp,   /* temporary bit */
        decode(bitand(flag,1536), 0, 'N', 'Y') ping, /* ping (to shared or null) bit */
        decode(bitand(flag,16384), 0, 'N', 'Y') stale, /* stale bit */
        decode(bitand(flag,65536), 0, 'N', 'Y') direct, /* direct access bit */
        decode(bitand(flag,1048576), 0, 'N', 'Y') new /* new bit */
from    x$bh
where   dbablk = 56650
order by dbablk;
```

| LRBA_SEQ   | STATE |       |       |       |       |       | DBARFIL | DBABLK | TCH   | FLAG   | HSCN_BAS   |
|------------|-------|-------|-------|-------|-------|-------|---------|--------|-------|--------|------------|
| -----      | ----- | ----- | ----- | ----- | ----- | ----- | -----   | -----  | ----- | -----  | -----      |
| CR_SCN_BAS | D     | T     | P     | S     | D     | N     |         |        |       |        |            |
| -----      | -     | -     | -     | -     | -     | -     |         |        |       |        |            |
| 0          |       |       |       |       | 1     |       | 1       | 56650  | 1     | 8200   | 4294967295 |
| 0          | N     | N     | N     | N     | N     | N     |         |        |       |        |            |
| 0          |       |       |       |       | 3     |       | 1       | 56650  | 2     | 524288 | 0 115      |
| 4347881    | N     | N     | N     | N     | N     | N     |         |        |       |        |            |



*Hash Chain is now TWO! One is a CR and the other is Current.*





## V\$Transaction now has our record (created when transactions have undo)

```
SELECT t.addr, t.xidusn USN, t.xidslot SLOT, t.xidsqn SQL, t.status,  
       t.used_ublk UBLK, t.used_urec UREC, t.log_io LOG,  
       t.phy_io PHY, t.cr_get, t.cr_change CR_CHA  
FROM   v$transaction t, v$session s  
WHERE  t.addr = s.taddr;
```

| ADDR     | USN | SLOT | SQL    | STATUS | UBLK |
|----------|-----|------|--------|--------|------|
| -----    |     |      |        |        |      |
| UREC     | LOG | PHY  | CR_GET | CR_CHA |      |
| -----    |     |      |        |        |      |
| 69E50E5C | 5   | 42   | 652    | ACTIVE | 1    |
| 1        | 3   | 0    | 3      | 0      |      |

USN is the Undo Segment Number (rollback segment ID)

SLOT is the slot number in the rollback segment's transaction table.

SQL (Wrap) is the sequence number for the transaction.

USN+SLOT+SQL are the three values that uniquely identifies a transaction <sup>117</sup>XID



# Dump the block



## Dump the block

| <i>Itl</i> | <i>Xid</i>          | <i>Uba</i>         | <i>Flag</i> | <i>Lck</i> | <i>Scn/Fsc</i>      |
|------------|---------------------|--------------------|-------------|------------|---------------------|
| 0x01       | 0x0005.02a.0000028c | 0x008000af.02b6.01 | ----        | 1          | fsc 0x0029.00000000 |
| 0x02       | 0x0004.016.00000fae | 0x008000cc.08af.34 | C---        | 0          | scn 0x0000.003deb5b |

ITL – 2 Interested Transaction Lists

Committed Transaction

The row I deleted is still locked; fsc is 0x29 = 41 bytes

### Transaction ID

Undo 5 = 5 (decimal)  
Slot 2a = 42 (decimal)  
SEQ 28C = 652

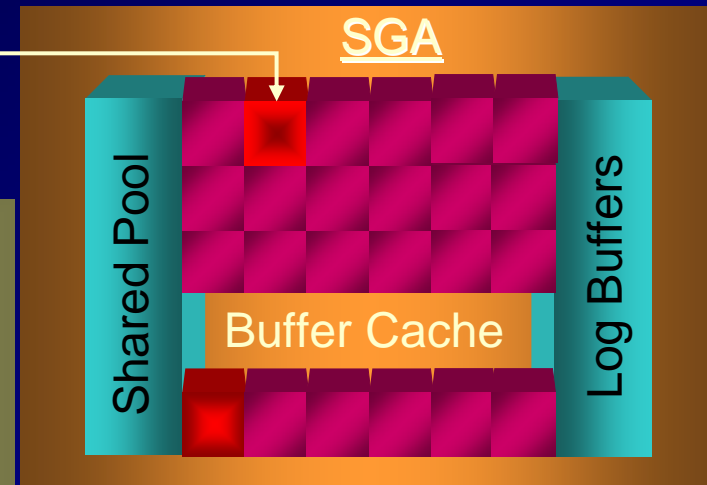
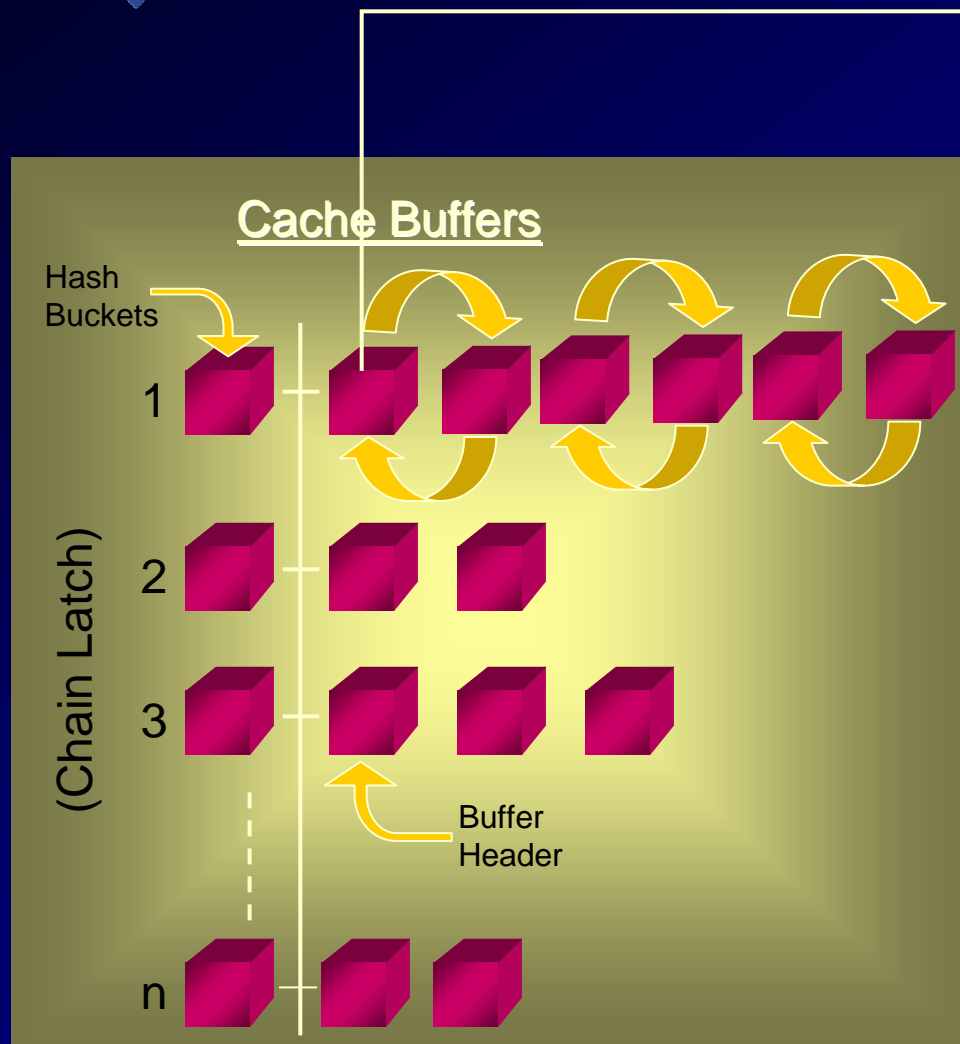
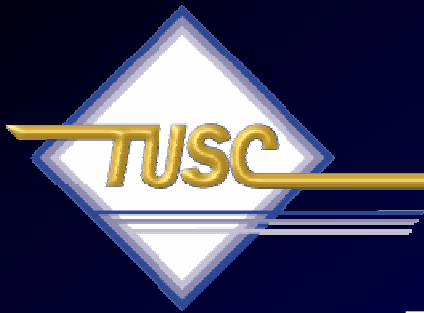
### UBA:

File.block.sequence.record  
Undo block address where last change is recorded.

# Insert in 3 other sessions & drive x\$bh up to the max of 6 versions of block



| <i>LRBA_SEQ</i>   | <i>STATE</i> |          |          |          |          |          | <i>DBARFIL</i> | <i>DBABLK</i> | <i>TCH</i> | <i>FLAG</i> | <i>HSCN_BAS</i> |
|-------------------|--------------|----------|----------|----------|----------|----------|----------------|---------------|------------|-------------|-----------------|
| -----             | -----        |          |          |          |          |          | -----          | -----         | -----      | -----       | -----           |
| <i>CR_SCN_BAS</i> | <i>D</i>     | <i>T</i> | <i>P</i> | <i>S</i> | <i>D</i> | <i>N</i> |                |               |            |             |                 |
| -----             | -            | -        | -        | -        | -        | -        |                |               |            |             |                 |
| 0                 |              |          |          |          | 3        |          | 1              | 56650         | 1          | 524416      | 0               |
| 4350120           | N            | N        | N        | N        | N        | N        |                |               |            |             |                 |
| 0                 |              |          |          |          | 3        |          | 1              | 56650         | 1          | 524416      | 0               |
| 4350105           | N            | N        | N        | N        | N        | N        |                |               |            |             |                 |
| 365               |              |          |          |          | 1        |          | 1              | 56650         | 7          | 33562633    | 4350121         |
| 0                 | Y            | N        | N        | N        | N        | N        |                |               |            |             |                 |
| 0                 |              |          |          |          | 3        |          | 1              | 56650         | 1          | 524416      | 0               |
| 4350103           | N            | N        | N        | N        | N        | N        |                |               |            |             |                 |
| 0                 |              |          |          |          | 3        |          | 1              | 56650         | 1          | 524416      | 0               |
| 4350089           | N            | N        | N        | N        | N        | N        |                |               |            |             |                 |
| 0                 |              |          |          |          | 3        |          | 1              | 56650         | 1          | 524288      | 0               |
| 4350087           | N            | N        | N        | N        | N        | N        |                |               |            |             |                 |



*Hash Chain is  
now SIX long!  
Five CR and the  
one Current.*



# Why only 6 versions of a Block?

```
select  a.ksppinm, b.kspstvl, b.kspstdf, a.kspdesc
from    x$kspci a, x$kspcv b
where   a.indx = b.indx
and     substr(ksppinm,1,1) = '_'
and     ksppinm like '%&1%'
order by ksppinm;
```

**KSPPINM**

-----

**KSPSTVL**

-----

**KSPSTDF**

-----

**KSPDESC**

-----

**\_db\_block\_max\_cr\_dba**

**6**

**TRUE**

**Maximum Allowed Number of CR buffers per dba**



# What happens after we roll everything back – x\$bh Still an LRBA:

| <i>LRBA_SEQ</i>   | <i>STATE</i> |          |          |          |          |          |   | <i>DBARFIL</i> | <i>DBABLK</i> | <i>TCH</i> | <i>FLAG</i> | <i>HSCN_BAS</i> |
|-------------------|--------------|----------|----------|----------|----------|----------|---|----------------|---------------|------------|-------------|-----------------|
| -----             | -----        |          |          |          |          |          |   | -----          | -----         | -----      | -----       | -----           |
| <i>CR_SCN_BAS</i> | <i>D</i>     | <i>T</i> | <i>P</i> | <i>S</i> | <i>D</i> | <i>N</i> |   |                |               |            |             |                 |
| -----             | -----        | -----    | -----    | -----    | -----    | -----    |   |                |               |            |             |                 |
| 0                 |              |          |          |          | 3        |          | 1 | 56650          | 1             | 524416     | 0           |                 |
| 4350120           | N            | N        | N        | N        | N        | N        |   |                |               |            |             |                 |
| 0                 |              |          |          |          | 3        |          | 1 | 56650          | 1             | 524416     | 0           |                 |
| 4350105           | N            | N        | N        | N        | N        | N        |   |                |               |            |             |                 |
| 365               |              |          |          |          | 1        |          | 1 | 56650          | 11            | 35659777   | 4350702     |                 |
| 0                 | Y            | N        | N        | N        | N        | N        |   |                |               |            |             |                 |
| 0                 |              |          |          |          | 3        |          | 1 | 56650          | 1             | 524416     | 0           |                 |
| 4350103           | N            | N        | N        | N        | N        | N        |   |                |               |            |             |                 |
| 0                 |              |          |          |          | 3        |          | 1 | 56650          | 1             | 524416     | 0           |                 |
| 4350089           | N            | N        | N        | N        | N        | N        |   |                |               |            |             |                 |
| 0                 |              |          |          |          | 3        |          | 1 | 56650          | 1             | 524288     | 0           |                 |
| 4350087           | N            | N        | N        | N        | N        | N        |   |                |               |            |             |                 |

6 rows selected.



# Let's check V\$TRANSACTION & match it up to ITL (no need to dump)

```
select  xidusn, xidslot, xidsqn, ubafil, ubablk, ubasqn, ubarec
from    v$transaction t, v$session s
where   t.ses_addr = s.saddr;
```

| XIDUSN | XIDSLOT | XIDSQN | UBAFIL | UBABLK | UBASQN | UBAREC |
|--------|---------|--------|--------|--------|--------|--------|
| 4      | 42      | 4863   | 2      | 851    | 2718   | 8      |
| 5      | 14      | 667    | 2      | 1458   | 713    | 25     |

4.42.4863 = 4.2a.12ff

5.14.667 = 5.e.29b

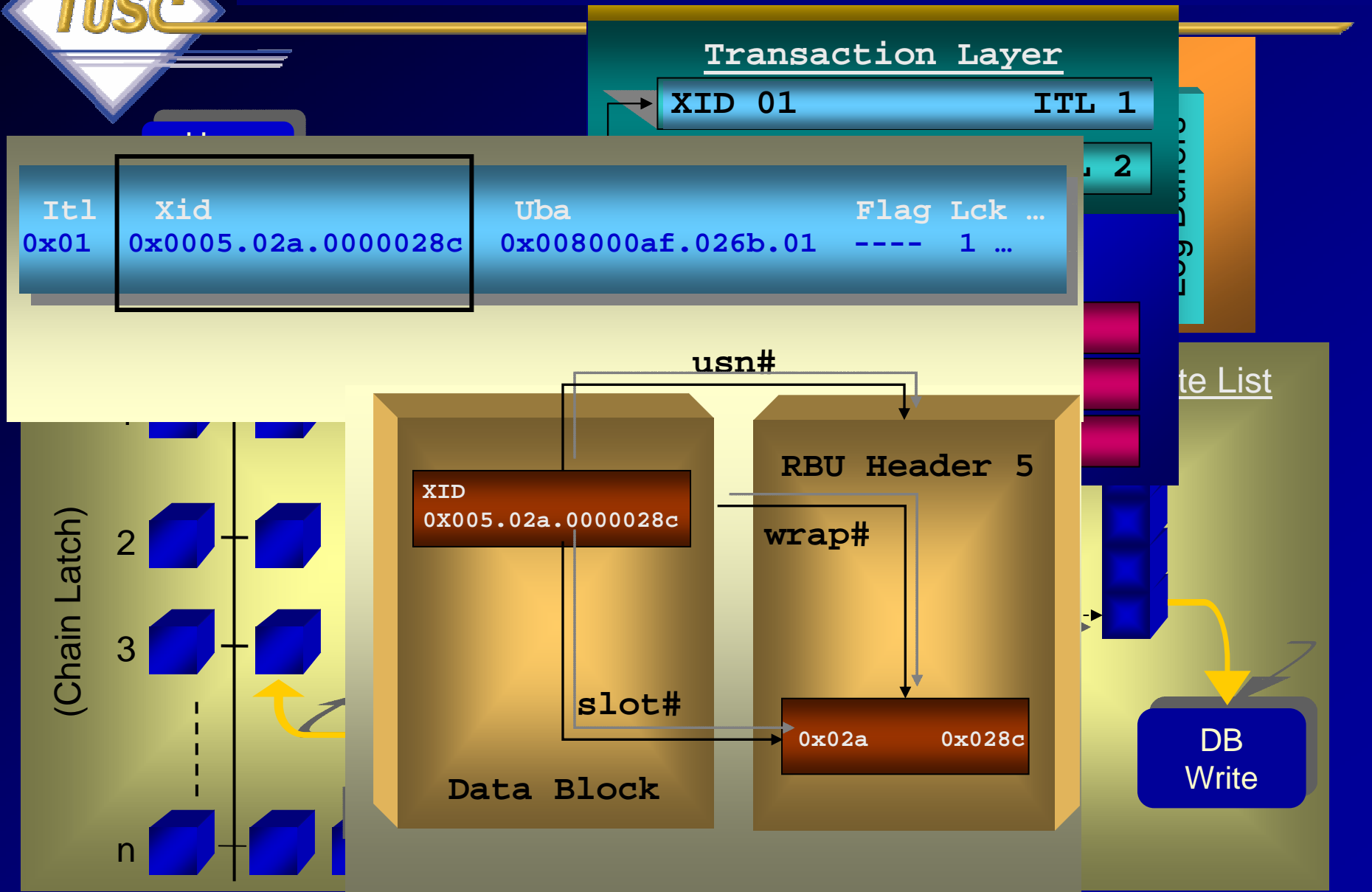
2.851.2718.8 = 800353.a9e.8

2.1458.713.25 = 8005b2.2c9.19

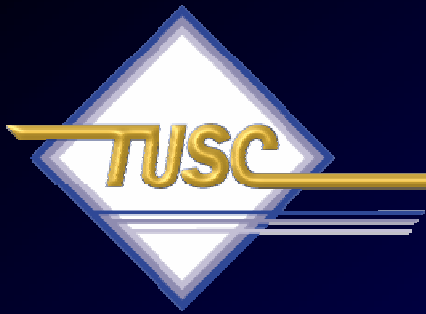
| Itl  | <del>Kid</del>      | Uba                | Flag | Lck | Scn/Fsc             |
|------|---------------------|--------------------|------|-----|---------------------|
| 0x02 | 0x0004.02a.000012ff | 0x00800353.0a9e.08 | ---- | 2   | fsc 0x0003.00000000 |
| 0x03 | 0x0005.00e.0000029b | 0x008005b2.02c9.19 | ---- | 14  | fsc 0x0000.00000000 |



# Transaction Identifiers

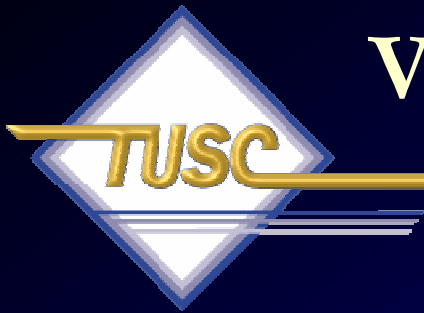




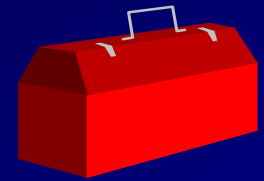


# Helpful V\$/X\$ Queries (FYI Only)





## V\$ Views over the years



| <u>Version</u> | <u>V\$ Views</u> | <u>X\$ Tables</u> |
|----------------|------------------|-------------------|
| 6              | 23               | ? (35)            |
| 7              | 72               | 126               |
| 8.0            | 132              | 200               |
| 8.1            | 185              | 271               |
| 9.0            | 227              | 352               |
| 9.2            | 259              | 394               |
| 10.1.02        | 340 (+31%)       | 543 (+38%) 126    |



# Listing of V\$ Views

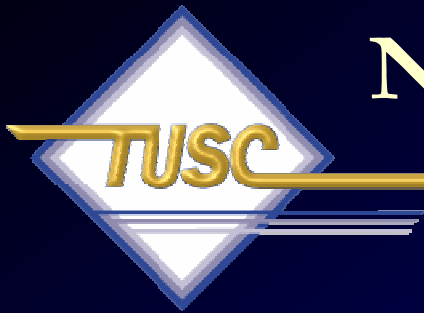


```
select      name
from        v$fixed_table
where       name like 'GV%'
order by    name;
```

*NAME*

-----

*GV\$ACCESS*  
*GV\$ACTIVE\_INSTANCES*  
*GV\$ACTIVE\_SESS\_POOL\_MTH*  
*GV\$AQ1*  
*GV\$ARCHIVE...*



## Need GV\$ - Instance ID



```
select  inst_id, (1 - (sum(decode(name, 'physical reads',value,0)) /  
                    (sum(decode(name, 'db block gets',value,0)) +  
                      sum(decode(name, 'consistent gets',value,0))))) * 100 "Hit Ratio"  
from    v$sysstat;
```

| INST_ID | Hit Ratio |
|---------|-----------|
|---------|-----------|

|       |       |
|-------|-------|
| ----- | ----- |
|-------|-------|

|   |            |
|---|------------|
| 1 | 90.5817699 |
|---|------------|



## Need GV\$ - Instance ID



```
select  inst_id, (1 - (sum(decode(name, 'physical reads',value,0)) /  
                    (sum(decode(name, 'db block gets',value,0)) +  
                      sum(decode(name, 'consistent gets',value,0)))) * 100 "Hit Ratio"  
from    gv$sysstat  
group by inst_id;
```

| INST_ID | Hit Ratio |
|---------|-----------|
|---------|-----------|

|       |       |
|-------|-------|
| ----- | ----- |
|-------|-------|

|   |            |
|---|------------|
| 1 | 90.5817699 |
|---|------------|

|   |            |
|---|------------|
| 2 | 96.2034537 |
|---|------------|



## X\$ used to create V\$



```
select  *  
from    v$fixed_view_definition  
where   view_name = 'GV$INDEXED_FIXED_COLUMN';
```

VIEW NAME

*gv\$indexed\_fixed\_column*

VIEW DEFINITION

```
select c.inst_id, kqftanam,  
        kqfcoidx, kqfconam, kqfcoipo  
from    X$kqfco c, X$kqftat  
where   t.indx = c.kqfcotab  
and     kqfcoidx != 0
```



# Listing of X\$ Tables



```
select      name
from        v$fixed_table
where       name like 'X%'
order by    name;
```

*NAME*

-----  
*X\$ACTIVECKPT*

*X\$BH*

*X\$BUFQM*

*X\$CKPTBUF*

*X\$CLASS\_STAT...*



# Listing of X\$ Indexes

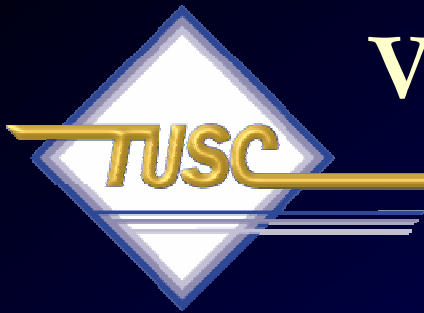
*(419 in 10g; 326 in 9i)*



```
select      table_name, index_number, column_name
from        gv$indexed_fixed_column
order by    table_name, index_number, column_name,
            column_position;
```

| <i>TABLE_NAME</i>    | <i>INDEX_NUMBER</i> | <i>COLUMN_NAME</i> |
|----------------------|---------------------|--------------------|
| <i>X\$CLASS_STAT</i> | <i>1</i>            | <i>ADDR</i>        |
| <i>X\$CLASS_STAT</i> | <i>2</i>            | <i>INDX</i>        |
| <i>X\$DUAL</i>       | <i>1</i>            | <i>ADDR</i>        |
| <i>X\$DUAL</i>       | <i>2</i>            | <i>INDX ...</i>    |





# V\$ - System Information



*SQL> select \* from v\$version;*

*BANNER*

-----  
*Oracle Database 10g Enterprise Edition Release 10.1.0.2.0 – Prod*

*PL/SQL Release 10.1.0.2.0 –Production*

*CORE 10.1.0.2.0      Production*

*TNS for 32-bit Windows: Version 10.1.0.2.0 – Production*

*NLSRTL Version 10.1.0.2.0 – Production*



## V\$ - System Information



*select \**

*from v\$option;*

*PARAMETER*

*VALUE*

-----  
*Partitioning*

-----  
*TRUE*

*Objects*

*TRUE*

*Real Application Clusters*

*FALSE*

*Advanced Replication*

*TRUE*

*Bit-Mapped Indexes*

*TRUE*

...

# Great V\$ - V\$SESSION\_WAIT (waiting right now)



```
select event, sum(decode(wait_time,0,1,0)) "Waiting Now",  
       sum(decode(wait_time,0,0,1)) "Previous Waits",  
       count(*) "Total"  
from v$session_wait  
group by event  
order by count(*);
```

*WAIT\_TIME = 0 means that it's waiting*

*WAIT\_TIME > 0 means that it previously waited this many ms*

# Great V\$ - V\$SESSION\_WAIT



| <i>EVENT</i>                       | <i>Waiting Now</i> | <i>Previous Waits</i> | <i>Total</i> |
|------------------------------------|--------------------|-----------------------|--------------|
| -----                              | -----              | -----                 | -----        |
| <i>db file sequential read</i>     | <i>0</i>           | <i>1</i>              | <i>1</i>     |
| <i>db file scattered read</i>      | <i>2</i>           | <i>0</i>              | <i>2</i>     |
| <i>latch free</i>                  | <i>0</i>           | <i>1</i>              | <i>1</i>     |
| <i>enqueue</i>                     | <i>2</i>           | <i>0</i>              | <i>2</i>     |
| <i>SQL*Net message from client</i> | <i>0</i>           | <i>254</i>            | <i>480</i>   |
| <i>...</i>                         |                    |                       |              |



# V\$SESSION\_WAIT

## Finding Current Specific waits



### Buffer Busy Waits or Write Complete Waits Events:

```
SELECT      /*+ ordered */ sid, event, owner, segment_name,  
            segment_type,p1,p2,p3  
FROM        v$session_wait sw, dba_extents de  
WHERE       de.file_id = sw.p1  
AND         sw.p2 between de.block_id  
            and de.block_id+de.blocks - 1  
AND         (event = 'buffer busy waits'  
            OR event = 'write complete waits')  
AND         p1 IS NOT null  
ORDER BY   event,sid;
```



# V\$EVENT\_NAME

## Finding P1, P2, P3



col name for a20

col p1 for a10

col p2 for a10

col p3 for a10

```
select event#,name,parameter1 p1,parameter2 p2,parameter3 p3
from v$event_name
```

where name in ('buffer busy waits', 'write complete waits')

| EVENT# | NAME                 | P1    | P2     | P3    |
|--------|----------------------|-------|--------|-------|
| -----  | -----                | ----- | -----  | ----- |
| 143    | write complete waits | file# | block# |       |
| 145    | buffer busy waits    | file# | block# | id    |



# V\$ - V\$SESSION\_WAIT\_HISTORY

## (Last 10 waits for session)



Buffer Busy Waits or Write Complete Waits Events:

```
SELECT      /*+ ordered */ sid, event, owner, segment_name,
            segment_type,p1,p2,p3
FROM        v$session_wait_history sw, dba_extents de
WHERE       de.file_id = sw.p1
AND         sw.p2 between de.block_id
            and de.block_id+de.blocks - 1
AND         (event = 'buffer busy waits'
            OR event = 'write complete waits')
AND         p1 IS NOT null
ORDER BY   event,sid;
```



# Great V\$ - V\$SESSION\_EVENT (waiting since the session started)



```
select sid, event, total_waits,  
       time_waited, event_id  
from v$session_event  
where time_waited > 0  
order by time_waited;
```

| SID | EVENT                     | TOTAL_WAITS | TIME_WAITED |
|-----|---------------------------|-------------|-------------|
| 159 | process startup           | 2           | 1           |
| 167 | latch: redo allocation    | 4           | 1           |
| 168 | log buffer space          | 2           | 3           |
| 166 | control file single write | 5           | 4           |

...





# V\$ - V\$SESSION\_WAIT\_CLASS (session waits by WAIT CLASS)



```
select wait_class, total_waits  
from v$system_wait_class;
```

| <i>SID</i> | <i>WAIT_CLASS</i> | <i>TOTAL_WAITS</i> |
|------------|-------------------|--------------------|
| -----      |                   |                    |
| 147        | Idle              | 6582               |
| 150        | Idle              | 138                |
| 153        | Application       | 10                 |
| 153        | Administrative    | 1                  |
| 153        | Idle              | 47                 |
| 153        | Network           | 48                 |
| 153        | User I/O          | 4                  |
| 153        | System I/O        | 33                 |



# Great V\$ - V\$SYSTEM\_EVENT (waits since the instance started)



```
select sid, event, total_waits,  
       time_waited, event_id  
from v$system_event  
where time_waited > 0  
order by time_waited;
```

| EVENT                       | TOTAL_WAITS | TIME_WAITED |
|-----------------------------|-------------|-------------|
| latch: session allocation   | 5644        | 1           |
| latch: redo allocation      | 4           | 1           |
| latch: cache buffers chains | 4           | 3           |
| enq: TX - index contention  | 1           | 3           |
| direct path write temp      | 57          | 6           |
| row cache lock              | 1           | 7 ... 142   |



# V\$ - V\$SYSTEM\_WAIT\_CLASS (system waits by WAIT CLASS)



```
select wait_class, total_waits  
from v$system_wait_class;
```

| <i>WAIT_CLASS</i>     | <i>TOTAL_WAITS</i> |
|-----------------------|--------------------|
| <i>-----</i>          |                    |
| <i>Other</i>          | <i>545472</i>      |
| <i>Application</i>    | <i>105</i>         |
| <i>Configuration</i>  | <i>92</i>          |
| <i>Administrative</i> | <i>1</i>           |
| <i>Concurrency</i>    | <i>30</i>          |
| <i>Commit</i>         | <i>793</i>         |
| <i>Idle</i>           | <i>186169</i>      |
| <i>Network</i>        | <i>385</i>         |
| <i>User I/O</i>       | <i>16017</i>       |
| <i>System I/O</i>     | <i>25478</i>       |



# Great V\$ - V\$OPEN\_CURSOR (Added sql\_id in 10g)



```
select sid, count(*)  
from v$open_cursor  
group by sid  
having count(*) > 300  
order by count(*) desc;
```

| <i>SID</i> | <i>COUNT(*)</i> |
|------------|-----------------|
|------------|-----------------|

|           |            |
|-----------|------------|
| <i>11</i> | <i>450</i> |
|-----------|------------|

|          |            |
|----------|------------|
| <i>8</i> | <i>320</i> |
|----------|------------|

|          |            |
|----------|------------|
| <i>9</i> | <i>301</i> |
|----------|------------|

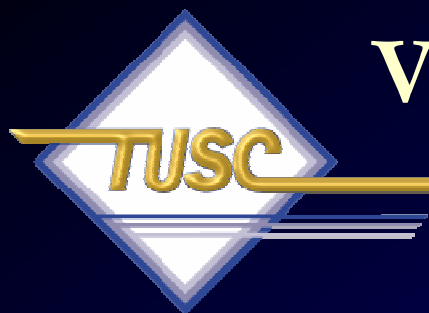
```
select sid, sql_id, count(*)  
from v$open_cursor  
group by sid, sql_id  
order by count(*);
```

| <i>SID</i> | <i>COUNT(*)</i> | <i>SQL_ID</i> |
|------------|-----------------|---------------|
|------------|-----------------|---------------|

|           |          |                      |
|-----------|----------|----------------------|
| <i>11</i> | <i>2</i> | <i>9tdqgnq7gd5cf</i> |
|-----------|----------|----------------------|

|           |          |                      |
|-----------|----------|----------------------|
| <i>11</i> | <i>1</i> | <i>a1xgtxssv5rrp</i> |
|-----------|----------|----------------------|

|           |          |                      |
|-----------|----------|----------------------|
| <i>11</i> | <i>1</i> | <i>attwbvtYu8x5c</i> |
|-----------|----------|----------------------|



## V\$ - Top 10 as % of All



```
select sum(pct_bufgets) percent
from (select rank() over ( order by buffer_gets desc )      as rank_bufgets,
      to_char(100 * ratio_to_report(buffer_gets) over      (), '999.99')
      pct_bufgets
from v$sqlarea )
where rank_bufgets < 11;
```

PERCENT

-----

97.07



# Great V\$ - V\$SQLAREA or V\$SQL



User1:

*SQL> conn system/manager*

*SQL> create table emp as select \* from scott.emp;*

*SQL> lock table emp in exclusive mode;*

User 2:

*SQL> conn system/manager*

*SQL> lock table emp in exclusive mode;*

User 3:

*SQL> connect scott/tiger*

*SQL> lock table emp in*



## Great V\$ - V\$SQLAREA or V\$SQL

```
select sql_text, users_executing, executions, users_opening  
from v$sqlarea  
where sql_text like 'lock table emp%';
```

| <i>SQL_TEXT</i>                         | <i>USERS_EXEC</i> | <i>EXEC</i> | <i>USERS_OPEN</i> |
|-----------------------------------------|-------------------|-------------|-------------------|
| <i>lock table emp in exclusive mode</i> | <i>1</i>          | <i>2</i>    | <i>3</i>          |



## Great V\$ - V\$SQLAREA or V\$SQL

```
select sql_text, users_executing, executions, users_opening  
from v$sql  
where sql_text like 'lock table emp%';
```

| <i>SQL_TEXT</i>                         | <i>USERS_EXEC</i> | <i>EXEC</i> | <i>USERS_OPEN</i> |
|-----------------------------------------|-------------------|-------------|-------------------|
| -----                                   | -----             | -----       | -----             |
| <i>lock table emp in exclusive mode</i> | <i>1</i>          | <i>1</i>    | <i>2</i>          |
| <i>lock table emp in exclusive mode</i> | <i>0</i>          | <i>1</i>    | <i>1</i>          |





## V\$ - What Users are doing...



```
select  a.sid, a.username, s.sql_text
from    v$session a, v$sqltext s
where   a.sql_address = s.address
and     a.sql_hash_value = s.hash_value
order  by a.username, a.sid, s.piece;
```

| <i>SID</i> | <i>USERNAME</i> | <i>SQL_TEXT</i> |
|------------|-----------------|-----------------|
|------------|-----------------|-----------------|

|           |                   |                                             |
|-----------|-------------------|---------------------------------------------|
| <i>11</i> | <i>PLSQL_USER</i> | <i>update s_employee set salary = 10000</i> |
| <i>9</i>  | <i>SYS</i>        | <i>select a.sid, a.username, s.sql_text</i> |
| <i>9</i>  | <i>SYS</i>        | <i>from v\$session a, v\$sqltext</i>        |
| <i>9</i>  | <i>SYS</i>        | <i>where a.sql_address = s.address</i>      |

*(...partial output listing)*



# Great V\$ - V\$SEGMENT\_STATISTICS



```
select object_name, statistic_name, value  
from v$segment_statistics  
where value > 100000  
order by value;
```

| <i>OBJECT_NAME</i> | <i>STATISTIC_NAME</i>   | <i>VALUE</i>  |
|--------------------|-------------------------|---------------|
| <hr/>              |                         |               |
| <i>ORDERS</i>      | <i>space allocated</i>  | <i>96551</i>  |
| <i>ORDERS</i>      | <i>space allocated</i>  | <i>134181</i> |
| <i>ORDERS</i>      | <i>logical reads</i>    | <i>140976</i> |
| <i>ORDER_LINES</i> | <i>db block changes</i> | <i>183600</i> |



# AWR – Segments by Buffer Busy Waits



## Segments by Buffer Busy Waits

- % of Capture shows % of Buffer Busy Waits for each top segment compared
- with total Buffer Busy Waits for all segments captured by the Snapshot

| Owner | Tablespace Name | Object Name | Subobject Name | Obj. Type       | Buffer Busy Waits | % of Capture |
|-------|-----------------|-------------|----------------|-----------------|-------------------|--------------|
|       |                 |             |                | TABLE PARTITION | 1,243,890         | 89.24        |
|       |                 |             |                | INDEX PARTITION | 39,258            | 2.82         |
|       |                 |             |                | INDEX PARTITION | 33,780            | 2.42         |
|       |                 |             |                | INDEX PARTITION | 28,563            | 2.05         |
|       |                 |             |                | INDEX PARTITION | 26,637            | 1.91         |

[Back to Segment Statistics](#)

[Back to Top](#)

## Segments by Global Cache Buffer Busy

- % of Capture shows % of GC Buffer Busy for each top segment compared
- with GC Buffer Busy for all segments captured by the Snapshot

| Owner | Tablespace Name | Object Name | Subobject Name | Obj. Type | GC Buffer Busy | % of Capture |
|-------|-----------------|-------------|----------------|-----------|----------------|--------------|
| SYS   | SYSTEM          | UNDOS       |                | TABLE     | 21             | 100.00       |

[Back to Segment Statistics](#)



# AWR – Segments by Logical Reads

## Segments by Logical Reads

- Total Logical Reads: 225,112,503
- Captured Segments account for 90.9% of Total

| Owner | Tablespace Name | Object Name | Subobject Name | Obj. Type       | Logical Reads | %Total |
|-------|-----------------|-------------|----------------|-----------------|---------------|--------|
|       |                 |             |                | INDEX PARTITION | 59,714,336    | 26.53  |
|       |                 |             |                | INDEX PARTITION | 48,936,144    | 21.74  |
|       |                 |             |                | INDEX PARTITION | 40,319,312    | 17.91  |
|       |                 |             |                | INDEX PARTITION | 23,612,368    | 10.49  |
|       |                 |             |                | INDEX PARTITION | 20,381,072    | 9.05   |

[Back to Segment Statistics](#)

[Back to Top](#)

## Segments by Physical Reads

- Total Physical Reads: 1,722
- Captured Segments account for 41.1% of Total

| Owner | Tablespace Name | Object Name | Subobject Name | Obj. Type       | Physical Reads | %Total |
|-------|-----------------|-------------|----------------|-----------------|----------------|--------|
|       |                 |             |                | INDEX PARTITION | 275            | 15.97  |
|       |                 |             |                | INDEX PARTITION | 164            | 9.52   |
|       |                 |             |                | INDEX PARTITION | 97             | 5.63   |



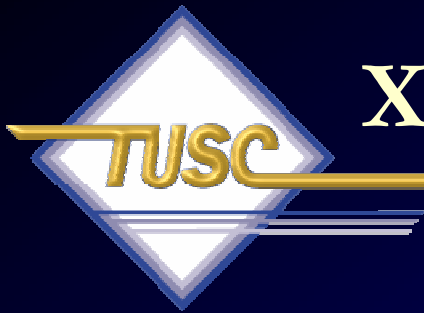
## X\$ - % of Current Redo Filled



```
SELECT  ROUND((cpodr_bno/lesiz),2)*100 || '%'
        PctCurLogFull
FROM    X$kcccp a, X$kccle b
WHERE   a.cpodr_seq = b.leseq;
```

PCTCURLOGFULL

-----  
97%



## X\$ - % of Current Redo Filled



alter system switch logfile;  
System altered.

```
SELECT      ROUND((cpodr_bno/lesiz),2)*100 || '%'
            PctCurLogFull
FROM        X$kcncp a, X$kcncle b
WHERE       a.cpodr_seq = b.leseq;

PCTCURLOGFULL
```

-----  
0%



# V\$ - V\$FILE\_HISTOGRAM



*select \* from v\$file\_histogram;*

| <i>FILE#</i> | <i>SINGLEBLKRDTIM_MILLI</i> | <i>SINGLEBLKRDS</i> |
|--------------|-----------------------------|---------------------|
| -----        | -----                       | -----               |
| 1            | 1                           | 900                 |
| 1            | 2                           | 120                 |
| 1            | 4                           | 131                 |
| 1            | 8                           | 359                 |
| 1            | 16                          | 1072                |
| 1            | 32                          | 1752                |
| ...          |                             |                     |
| 2            | 1                           | 3                   |
| 2            | 2                           | 1                   |
| 2            | 4                           | 2                   |
| 2            | 8                           | 17                  |

Oracle Enterprise Manager (SYSMAN) - Databases - Microsoft Internet Explorer

Oracle Enterprise Manager (SYSMAN) - Cluster Database: ioug - Microsoft Internet Explorer

Oracle Enterprise Manager (SYSMAN) - Cluster Database: ioug - Microsoft Internet Explorer

Oracle Enterprise Manager (SYSMAN) - Database Instance: ioug\_ioug1 - Microsoft Internet Explorer

Oracle Enterprise Manager (SYSMAN) - Database Instance: ioug\_ioug1 - Microsoft Internet Explorer

Oracle Enterprise Manager (SYSMAN) - Automatic Workload Repository - Microsoft Internet Explorer

Oracle Enterprise Manager (SYSMAN) - Snapshots - Microsoft Internet Explorer

Oracle Enterprise Manager (SYSMAN) - Snapshot Details - Microsoft Internet Explorer

Oracle Enterprise Manager (SYSMAN) - Snapshot Details - Microsoft Internet Explorer

Oracle Enterprise Manager (SYSMAN) - Snapshot Details - Microsoft Internet Explorer

Oracle Enterprise Manager (SYSMAN) - Snapshot Details - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Refresh Mail Print W

Address http://atlmind8.us.oracle.com:7777/em/console/database/instance/swrfSnapshotDetails?target=ioug\_ioug1&type=oracle\_database&startID=25&endID=26&event=viewRe Go Links

ORACLE Enterprise Manager 10g

Grid Control

Home Targets Deployments Alerts Policies Jobs Reports

All Targets | Hosts | Databases | Application Servers | Web Applications | Services | Systems | Groups

Cluster: IOUG > Cluster Database: ioug > Database Instance: ioug\_ioug1 > Automatic Workload Repository > Snapshots > Logged in As SYSTEM

Snapshot Details

Details Report

View ADDM Run

Save to File

## WORKLOAD REPOSITORY report for

| DB Name | DB Id     | Instance | Inst num | Release    | RAC | Host     |
|---------|-----------|----------|----------|------------|-----|----------|
| IOUG    | 325326670 | ioug1    | 1        | 10.2.0.2.0 | YES | atlmind1 |

|             | Snap Id | Snap Time          | Sessions | Cursors/Session |
|-------------|---------|--------------------|----------|-----------------|
| Begin Snap: | 25      | 23-Apr-06 11:30:34 | 61       | 3.6             |
| End Snap:   | 26      | 23-Apr-06 11:40:34 | 77       | 3.7             |
| Elapsed:    |         | 10.01 (mins)       |          |                 |
| DB Time:    |         | 23.14 (mins)       |          |                 |

## Report Summary

### Cache Sizes

|                   | Begin | End  |                 |         |
|-------------------|-------|------|-----------------|---------|
| Buffer Cache:     | 848M  | 832M | Std Block Size: | 8K      |
| Shared Pool Size: | 272M  | 268M | Log Buffer:     | 14,360K |

### Load Profile

Done

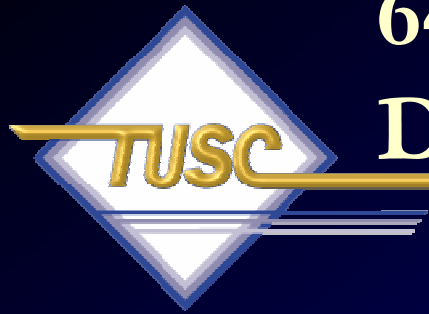
Internet





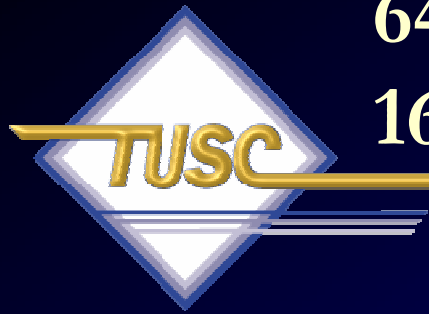
# If Time Permits... the Future!





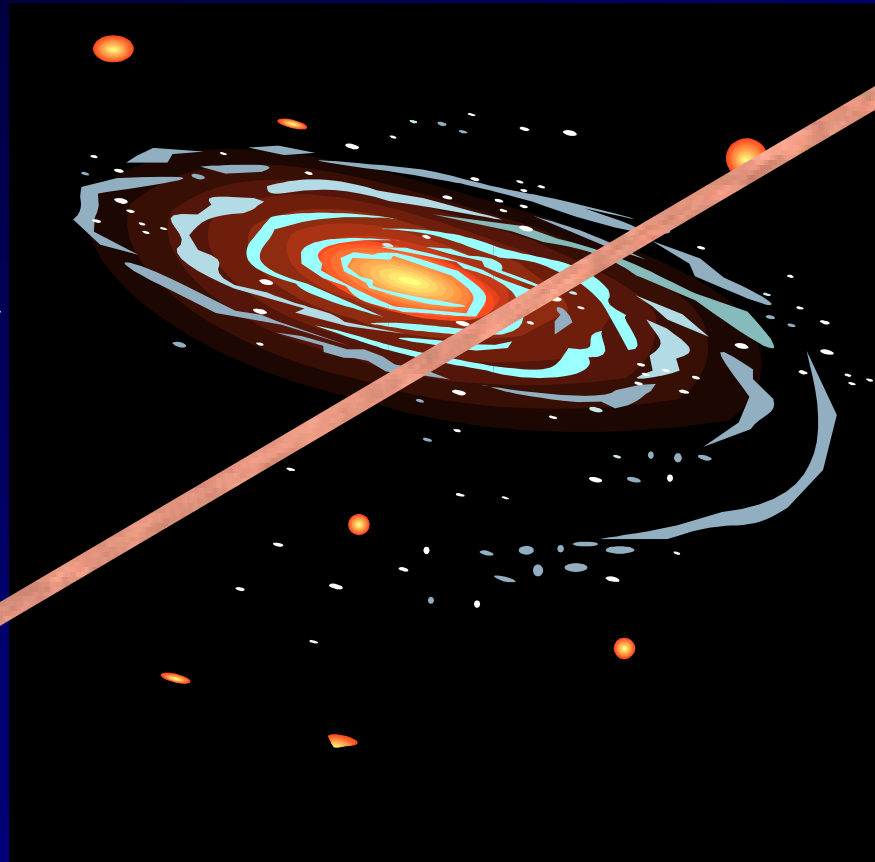
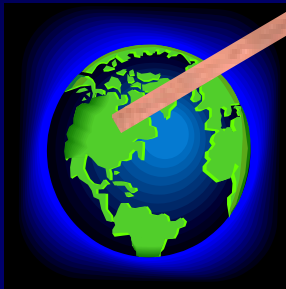
## 64-Bit advancement of Directly addressable memory

|                                                                                                                                                                                                                                                | <u>Address Direct</u>      | <u>Indirect/Extended</u> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|--------------------------|
| • <u>4 Bit:</u>                                                                                                                                                                                                                                | 16                         | (640)                    |
| • <u>8 Bit:</u>                                                                                                                                                                                                                                | 256                        | (65,536)                 |
| • <u>16 Bit:</u>                                                                                                                                                                                                                               | 65,536                     | (1,048,576)              |
| • <u>32 Bit:</u>                                                                                                                                                                                                                               | 4,294,967,296              |                          |
| • <u>64 Bit:</u>                                                                                                                                                                                                                               | 18,446,744,073,709,551,616 |                          |
| • When the hardware physically implements the theoretical possibilities of 64-Bit, things will dramatically change....<br>...moving from 32 bit to 64 bit will be like moving from 4 bit to 32 bit or like moving from 1971 to 2000 overnight. |                            |                          |



# 64bit allows Directly Addressing 16 Exabytes of Memory

Stack single  
sheets (2K worth  
of text on each)  
about 4.8B miles  
high to get 16E!!



You could stack documents from the  
Earth so high they would pass Pluto!



# Human Brain vs. Computer



## Brain

Individual maintains a ~250M capacity

Stores/processes ~ 2G in a lifetime

With over 6.5 Billion people in the world, total memory of world is 1.5P, total processed is 13 Exabytes



## Computer

Laptops ~256M in memory capacity

Stores/processes 40G in its lifetime, then buy new one

1.5+ Petabytes can be accessed and managed by one computer with today's technology (internet is 8P)

# Human Brain vs. Computer



## Brain

10T-1P of operations per second based on retina/neuron synapse speed (Moravec).

Based on Brain power in Watts consumption, the brain performs 100T – 100P operations per second.



## Computer

Fastest computer is IBM at 70.2 Teraflops (Blue / Gene L)

Computer is about equal on the low end of the brain and could be 1000 times slower on the high end of brain estimates.

# Human Brain vs. Computer



## Brain

Problem: We forget a lot of what we see and learn.

Solution: Drugs that improve memory or technology that plugs into the brain.

Problem: Information overload

Solution: Use the computer to aggregate things and use the brain for final decisions.



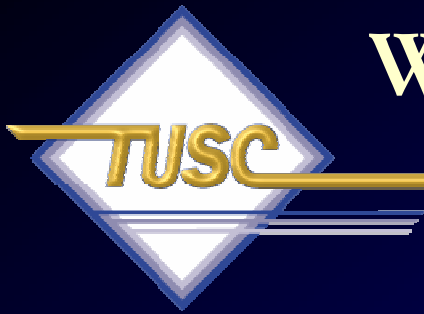
## Computer

Problem: Data Density limits or Superparamagnetic effect (SPE).

Solution: Electron Microscope writing on stainless steel in a vacuum.

Problem: More memory gets farther from the CPU & Limits to CPU speed

Solution: Chip multiplexing & multi-core CPUs



## What we covered:

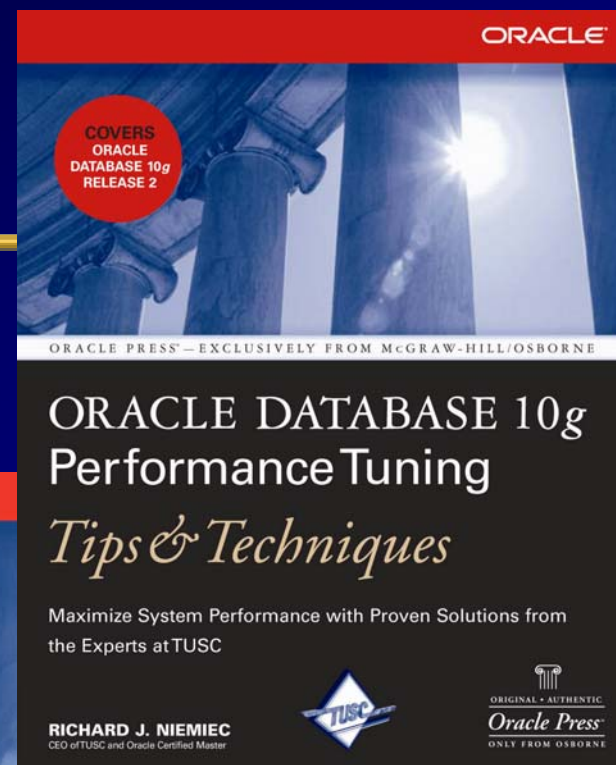
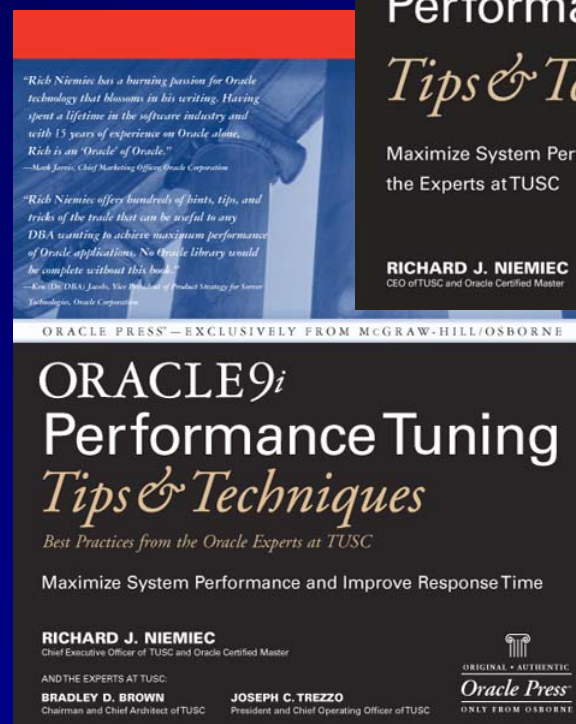
- Statspack, Tools & Scripts that you can still use
  - Top Waits
  - Load Profile
  - Latch Waits
  - Top SQL
  - Instance Activity
  - File I/O
- The Future OEM & ADDM
- Helpful V\$/X\$
- Summary





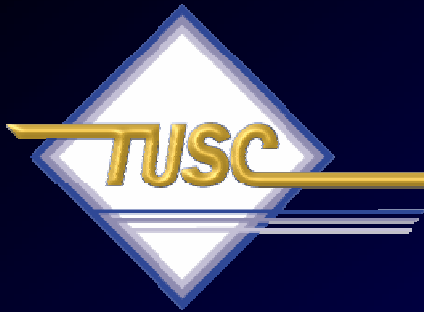
## For More Information

- [www.tusc.com](http://www.tusc.com)
- *Oracle9i Performance Tuning Tips & Techniques; Richard J. Niemiec; Oracle Press (May 2003)*
- *Oracle 10g Tuning (Early 2007)*



*"If you are going through hell, keep going" - Churchill*





*“The strength of the team is each individual member...the strength of each member is the team.”*

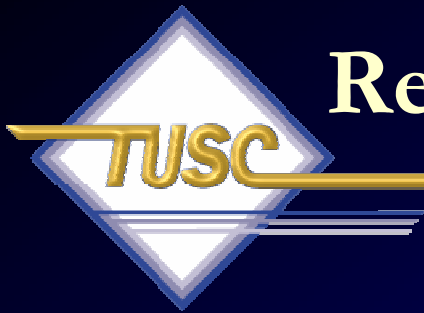
*--Phil Jackson*

*The Oracle User Groups are part of your team!*



## References

- *Oracle9i Performance Tuning Tips & Techniques*, Rich Niemiec
- *The Self-managing Database: Automatic Performance Diagnosis*; Karl Dias & Mark Ramacher, Oracle Corporation
- *EM Grid Control 10g*; otn.oracle.com, Oracle Corporation
- *Oracle Enterprise Manager 10g: Making the Grid a Reality*; Jay Rossiter, Oracle Corporation
- *The Self-Managing Database: Guided Application and SQL Tuning*; Benoit Dageville, Oracle Corporation
- *The New Enterprise Manager: End to End Performance Management of Oracle*; Julie Wong & Arsalan Farooq, Oracle Corporation
- *Enterprise Manager : Scalable Oracle Management*; John Kennedy, Oracle Corporation



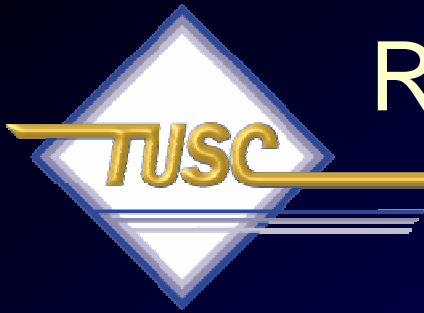
## References

- *Oracle Database 10g Performance Overview*; Hervé Lejeune, Oracle Corporation
- *Oracle 10g*; Penny Avril, Oracle Corporation
- *Forrester Reports, Inc., TechStrategy Research*, April 2002, Organic IT
- *Internals of Real Application Cluster*, Madhu Tumma, Credit Suisse First Boston
- *Oracle9i RAC; Real Application Clusters Configuration and Internals*, Mike Ault & Madhu Tumma
- *Oracle Tuning Presentation*, Oracle Corporation



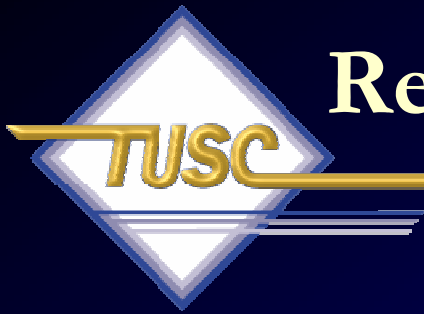
## References

- [www.tusc.com](http://www.tusc.com), [www.oracle.com](http://www.oracle.com), [www.ixora.com](http://www.ixora.com), [www.laoug.org](http://www.laoug.org), [www.ioug.org](http://www.ioug.org), [technet.oracle.com](http://technet.oracle.com)
- *Oracle PL/SQL Tips and Techniques*, Joseph P. Trezzo; Oracle Press
- *Oracle9i Web Development*, Bradley D. Brown; Oracle Press
- *Special thanks to Steve Adams, Mike Ault, Brad Brown, Don Burleson, Kevin Gilpin, Herve Lejeune, Randy Swanson and Joe Trezzo.*
- Landauer 1986. T.K. Landauer; "How much do people remember? Some estimates of the quantity of information in long-term memory," *Cognitive Science* ,10 (4) pp. 477-493.
- Ralph Merkle, *Energy limits to the computational power of the human brain*



# References

- Oracle Database 10g Automated Features , Mike Ault, TUSC
- Oracle Database 10g New Features, Mike Ault, Daniel Liu, Madhu Tumma, Rampant Technical Press, 2003, [www.rampant.cc](http://www.rampant.cc)
- Oracle Database 10g - The World's First Self-Managing, Grid-Ready Database Arrives, Kelli Wiseth, Oracle Technology Network, 2003, [otn.oracle.com](http://otn.oracle.com)



## References

- Oracle 10g; Penny Avril, Principal Database Product Manager, Server Technologies, Oracle Corporation
- To Infinity and Beyond, Brad Brown, TUSC
- Forrester Reports, Inc., TechStrategy Research, April 2002, Organic IT
- *Internals of Real Application Cluster*, Madhu Tumma, Credit Suisse First Boston
- *Oracle9i RAC; Real Application Clusters Configuration and Internals*, Mike Ault & Madhu Tumma
- *Oracle9i Performance Tuning Tips & Techniques*, Richard J. Niemiec



## References

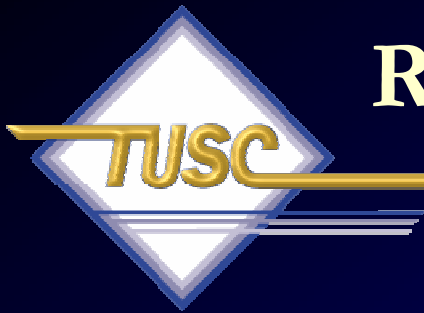
- Oracle 10g documentation
- Oracle 9i RAC class & instructor's comments
- Oracle 9i Concepts manual
- <http://geocities.com/pulliamrick/>
- *Tips for Tuning Oracle9i RAC on Linux*, Kurt Engeleiter, Van Okamura, Oracle
- *Leveraging Oracle9i RAC on Intel-based servers to build an "Adaptive Architecture"*, Stephen White, Cap Gemini Ernst & Young, Dr Don Mowbray, Oracle, Werner Schueler, Intel



## References

- *Running YOUR Applications on Real Application Clusters (RAC); RAC Deployment Best Practices*, Kirk McGowan, Oracle Corporation
- *The Present, The Future but not Science Fiction; Real Application Clusters Development*, Angelo Pruscino, Oracle
- *Building the Adaptive Enterprise; Adaptive Architecture and Oracle*, Malcolm Carnegie, Cap Gemini Ernst & Young
- *Internals of Real Application Cluster*, Madhu Tamma, Credit Suisse First Boston
- *Creating Business Prosperity in a Challenging Environment*, Jeff Henley





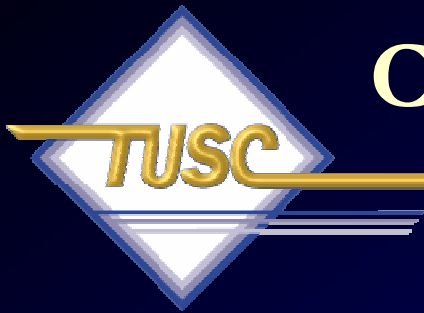
## References

- *Real Application Clusters, Real Customers Real Results*, Erik Peterson, Technical Manager, RAC, Oracle Corp.
- *Deploying a Highly Manageable Oracle9i Real Applications Database*, Bill Kehoe, Oracle
- *Getting the most out of your database*, Andy Mendelsohn, SVP Server Technologies, Oracle Corporation
- *Oracle9iAS Clusters: Solutions for Scalability and Availability*, Chet Fryjoff, Product Manager, Oracle Corporation
- *Oracle RAC and Linux in the real enterprise*, Mark Clark, Director, Merrill Lynch Europe PLC, Global Database Technologies



# TUSC Services

- Oracle Technical Solutions
  - Full-Life Cycle Development Projects
  - Enterprise Architecture
  - Database Services
- Oracle Application Solutions
  - Oracle Applications Implementations/Upgrades
  - Oracle Applications Tuning
- Managed Services
  - 24x7x365 Remote Monitoring & Management
  - Functional & Technical Support
- Training & Mentoring
- Oracle Authorized Reseller



# Copyright Information

- Neither TUSC nor the author guarantee this document to be error-free. Please provide comments/questions to [rich@tusc.com](mailto:rich@tusc.com).
- TUSC © 2007. This document cannot be reproduced without expressed written consent from an officer of TUSC, but the DOUG may reproduce or copy this for conference use.

## Contact Information

Rich Niemiec: [rich@tusc.com](mailto:rich@tusc.com)

[www.tusc.com](http://www.tusc.com)

