

Performance  
Diagnosis with  
“*Custom*” ASH and  
AWR Reporting

Daniel Fink

I use the quotes around “CUSTOM” because I show more than just home grown scripts. I have also found that there are several supplied reports that people don’t know about. Many of the custom reports/statements can be easily adapted to statspack.



## Session Goals

Review lesser known ASH/AWR features and reports

Demonstrate how to combine data analysis techniques and Oracle's metrics to go beyond the supplied reports

Identify and/or Eliminate possible causes for performance problems in both reactive and proactive exercises.



# Agenda

- **ASH/AWR Data**
  - Data Structures
  - Sampling
  - Snapshots and Deltas
- **Analytical SQL**
  - Partitioning/Windowing
  - Functions
- **Data Analysis**
  - Profiling
  - Variance
  - Trending

## Case Study 2 + 4 != 6

- Overnight process suddenly started running long
- What changed?

March 31, 2011

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

Page 4 of 104\*

### Variances

1 – statement running longer

0 – statement started later, so the ETL process was running longer

2 – data being processed was about average

3 – Run time was significantly higher

4 – cpu time was about average

5 – wait time was significantly higher and most of it was in physical i/o

6 – Actual disk/memory reads was about the same

7 – current Average pio time was 50% higher than previous average

# Really, Larry?

- The use of AWR and ASH, even just the data, is restricted by license (Diagnostics Pack).
  - Always check for proper licensing before running these reports or querying the data.
- Disclaimer

March 31, 2011

www.optimaldba.com

Page 5 of 104\*

Seriously – this is one of the <dumbest> restrictions. If Oracle wants to restrict the use of ADDM or the GC interface, fine...but to restrict you from data that the instance is collecting (using your systems resources!) is quite stupid. Imagine where we would be if the Wait Event Interface was separately licensed. Realistically, the value derived from the ASH/AWR data likely exceeds the licensing cost. Especially ASH.

From Oracle Licensing manual – “Some Diagnostics Pack features are accessed by way of database server APIs and command-line interfaces:

- The DBMS\_WORKLOAD\_REPOSITORY package is part of this pack.
- The DBMS\_ADVISOR package is part of this pack if you specify ADDM as the value of the *advisor\_name* parameter, or if you specify for the value of the *task\_name* parameter any value starting with the ADDM prefix.
- The V\$ACTIVE\_SESSION\_HISTORY dynamic performance view is part of this pack.
- All data dictionary views beginning with the prefix DBA\_HIST\_ are part of this pack, along with their underlying tables.
- All data dictionary views with the prefix DBA\_ADVISOR\_ are part of this pack if queries to these views return rows with the value 'ADDM' in the ADVISOR\_NAME column or a value of 'ADDM\*' in the TASK\_NAME column or the corresponding TASK\_ID.
- The following reports found in the */rdbms/admin/* directory of the Oracle home directory are part of this pack: awrrpt.sql, awrrpti.sql, addmrtp.sql, addmrpti.sql, awrrpt.sql, awrrpti.sql, addmrtp.sql, addmrpti.sql.”



## ASH/AWR Data

- Data Structures
- Sampling
- Snapshots and Deltas



# Data Structures

- Creation

- catawr.sql

- Naming

- v\$active\_session\_history
- WRH\$ tables
  - some partitioned on dbid, snap\_id
- DBA\_HIST views



# Views

- ASH

- v\$active\_session\_history
- dba\_hist\_active\_sess\_history

- AWR

- dba\_hist\_snapshot
- dba\_hist\_system\_event
- dba\_hist\_sysstat
- dba\_hist\_sys\_time\_model
- dba\_hist\_sqlstat



# Sampling

- Active Session History
  - Circular buffer
  - Snapshot every second
- ASH AWR sample
  - 1 of 10 ASH snapshots
- ASH reports
  - Will automatically use ash/awr data



## Totals v. Deltas

- Sample
  - Current state, not cumulative
- Running Totals
  - Cumulative
  - Monotonically increasing values
  - Time value is previous value – end value
- Deltas
  - Not all structures save deltas



## DB Time Model

- What time was being spent by sessions?
  - Each sample represents a 'moment' in time
- Active
  - Working
  - Waiting
- Inactive

# Configuration

- What is my current config?

- DBA\_HIST\_WR\_CONTROL

- dbid

- snap\_interval

- retention

- topnsql

- awrinfo.sql

- Changing

- DBMS\_WORKLOAD\_REPOSITORY.MODIFY\_SNAPSHOT\_SETTINGS

March 31, 2011

www.optimaldba.com

Page 12 of 104\*

The one thing you cannot change is the snapshot level, despite what any site may tell you. At least as off 11gr2.

```

SELECT snap_interval, retention, topnsql
FROM dba_hist_wr_control;

SNAP_INTERVAL      RETENTION          TOPNSQL
-----
+00000 01:00:00.0 +00008 00:00:00.0 DEFAULT

BEGIN
  DBMS_WORKLOAD_REPOSITORY.MODIFY_SNAPSHOT_SETTINGS
    (retention => 43200, interval => 30, topnsql => 30);
END;
/
PL/SQL procedure successfully completed.

SELECT snap_interval, retention, topnsql
FROM dba_hist_wr_control;

SNAP_INTERVAL      RETENTION          TOPNSQL
-----
+00000 00:30:00.0 +00030 00:00:00.0      30

```

You can adjust the interval, retention, and captured Top SQL of snapshot generation for a specified database ID, but note that this can affect the precision of the Oracle Database diagnostic tools.

The INTERVAL setting affects how often the database automatically generates snapshots. The RETENTION setting affects how long the database stores snapshots in the workload repository. The TOPNSQL setting affects the number of Top SQL to flush for each SQL criteria (Elapsed Time, CPU Time, Parse Calls, sharable Memory, and Version Count). The value for this setting is not affected by the statistics/flush level and will override the system default behavior for the AWR SQL collection. It is possible to set the value for this setting to MAXIMUM to capture the complete set of SQL in the cursor cache, though by doing so (or by setting the value to a very high number) may lead to possible space and performance issues because there will more data to collect and store. To adjust the settings, use the MODIFY\_SNAPSHOT\_SETTINGS procedure. For example:

```
BEGIN DBMS_WORKLOAD_REPOSITORY.MODIFY_SNAPSHOT_SETTINGS(
retention => 43200, interval => 30, topnsql => 100, dbid => 3310949047); END; /
```

In this example, the retention period is specified as 43200 minutes (30 days), the interval between each snapshot is specified as 30 minutes, and the number of Top SQL to flush for each SQL criteria as 100. If NULL is specified, the existing value is preserved. The optional database identifier is 3310949047. If you do not specify a value for dbid, the local database identifier is used as the default value. You can check the current settings for your database instance with the DBA\_HIST\_WR\_CONTROL view.

# Configuration

## ■ retention

- number of minutes to retain snapshot

Time	Day	Week	Month	6 Months	Year
Minutes	1440	10080	43200	259200	525600

- Set to 0 to retain snapshots for almost forever

## ■ topnsql

- How many statements do you want to look at tuning at any one time?
- What historical information do you want to retain?
- To always track statements use `add_colored_sql` procedure



# Configuration

- AWR INFO Report
- Provides information about the contents of the Automatic Workload Repository
- `$ORACLE_HOME/rdbms/admin/awrinfo`



# AWR Snapshots Information

- (1a) SYSAUX usage - Schema breakdown (dba\_segments)
- (1b) SYSAUX occupants space usage (v\$sysaux\_occupants)
- (1c) SYSAUX usage - Unregistered Schemas
- (1d) SYSAUX usage - Unaccounted space in registered schemas
- (2) Size estimates for AWR snapshots
- (3a) Space usage by AWR components (per database)
- (3b) Space usage within AWR Components (> 500K)



## AWR Snapshots Information

- (4) Space usage by non-AWR components (> 500K)
- (5a) AWR snapshots - last 50
- (5b) AWR snapshots with errors or invalid
- (5c) AWR snapshots -- OLDEST Non-Baselined snapshots
- (6) AWR Control Settings - interval, retention
- (7a) AWR Contents - row counts for each snapshots
- (7b) AWR Contents - average row counts per snapshot
- (7c) AWR total item counts - names, text, plans



## Advisor Framework Info

- (1) Advisor Tasks - Last 50
- (2) Advisor Task - Oldest 5
- (3) Advisor Tasks With Errors - Last 50



## ASH Usage Info

- (1a) ASH histogram (past 3 days)
- (1b) ASH histogram (past 1 day)
- (2a) ASH details (past 3 days)
- (2b) ASH details (past 1 day)
- (2c) ASH sessions (Fg Vs Bg) (past 1 day across all instances in RAC)



# AWR INFO Report

- [AWR INFO Report](#)

March 31, 2011

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

Page 20 of 104\*

# Supplied Reporting

- ADDM
- AWR SQL Report
- AWR Compare Periods Report
  - aka AWR Diff Report
- Daily/weekly AWR
- `dbms_xplan.display_awr`

March 31, 2011

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

Page 21 of 104\*

Diff report is actually “Compare Periods Report”



# ADDM

- If you are licensed, then run the ADDM report with AWR reports.
- Let it do some preliminary analysis



# AWR SQL Report

- `$ORACLE_HOME/rdbms/admin/awrsqrpt.sql`
  - Typical parameters + sql\_id
- Display all information about the statement
  - Including multiple execution plans!

# WORKLOAD REPOSITORY SQL Report

## Snapshot Period Summary

DB Name	DB Id	Instance	Inst num	Release	RAC	Host
PROD2	1309734745	PROD2	1	10.2.0.4.0	NO	db-prod2

	Snap Id	Snap Time	Sessions	Cursors/Session
Begin Snap:	16124	29-Jun-10 00:00:10	240	14.0
End Snap:	16243	01-Jul-10 11:30:11	272	12.7
Elapsed:		3,570.01 (mins)		
DB Time:		56,565.06 (mins)		

## SQL Summary

SQL Id	Elapsed Time (ms)	Module	Action	SQL Text
5w5a8vw9waabh	214,202,043	httpd@app-d2 (TNS V1-V3)		SELECT /*+ first rows */ DECODE(SIGN(MIN DATE TIME - :B4), 1, '1', D...

[Back to Top](#)

## SQL ID: 5w5a8vw9waabh

- 1st Capture and Last Capture Snap IDs refer to Snapshot IDs within the snapshot range
- `SELECT /*+ first rows */ DECODE(SIGN(MIN DATE TIME - :B4), 1, '1', DECO...`

#	Plan Hash Value	Total Elapsed Time(ms)	Executions	1st Capture Snap ID	Last Capture Snap ID
1	182079744	213,539,880	54	16223	16243
2	1150740400	662,163	12	16132	16176

## Plan 1(PHV: 182079744)

- [Plan Statistics](#)
- [Execution Plan](#)

[Back to Top](#)

### Plan Statistics

- % Total DB Time is the Elapsed Time of the SQL statement divided into the Total Database Time multiplied by 100

Stat Name	Statement Total	Per Execution	% Snap Total
Elapsed Time (ms)	213,539,880	3,954,442.22	6.29
CPU Time (ms)	14,448,843	267,571.17	1.02
Executions	54		
Buffer Gets	73,291,064	1,357,241.93	0.75
Disk Reads	50,209,879	929,812.57	17.08
Parse Calls	53	0.98	0.00
Rows	25	0.46	
User I/O Wait Time (ms)	201,571,019		
Cluster Wait Time (ms)	0		
Application Wait Time (ms)	0		
Concurrency Wait Time (ms)	813		
Invalidations	4		
Version Count	64		
Sharable Mem(KB)	1,401		

## Execution Plan

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	Pstart	Pstop
0	SELECT STATEMENT				692K(100)			
1	HASH GROUP BY		3938	169K	692K (7)	02:18:27		
2	HASH JOIN		3938	169K	692K (7)	02:18:27		
3	VIEW		11	88	4 (0)	00:00:01		
4	CONNECT BY WITH FILTERING							
5	TABLE ACCESS BY INDEX ROWID	RT	1	29	2 (0)	00:00:01		
6	INDEX UNIQUE SCAN	RT_PK	1		1 (0)	00:00:01		
7	NESTED LOOPS							
8	CONNECT BY PUMP							
9	TABLE ACCESS BY INDEX ROWID	RT	11	143	4 (0)	00:00:01		
10	INDEX RANGE SCAN	RT_IX1	7		1 (0)	00:00:01		
11	PARTITION RANGE ALL		121K	4278K	692K (7)	02:18:27	1	90
12	TABLE ACCESS FULL	MCST	121K	4278K	692K (7)	02:18:27	1	90

# Plan 2(PHV: 1150740400)

- [Plan Statistics](#)
- [Execution Plan](#)

[Back to Top](#)

## Plan Statistics

- % Total DB Time is the Elapsed Time of the SQL statement divided into the Total Database Time multiplied by 100

Stat Name	Statement Total	Per Execution	% Snap Total
Elapsed Time (ms)	662,163	55,180.22	0.02
CPU Time (ms)	116,168	9,680.68	0.01
Executions	12		
Buffer Gets	707,362	58,946.83	0.01
Disk Reads	236,485	19,707.08	0.08
Parse Calls	10	0.83	0.00
Rows	51	4.25	
User I/O Wait Time (ms)	594,438		
Cluster Wait Time (ms)	0		
Application Wait Time (ms)	0		
Concurrency Wait Time (ms)	0		
Invalidations	0		
Version Count	12		
Sharable Mem(KB)	208		

## Execution Plan

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	Pstart	Pstop
0	SELECT STATEMENT				201 (100)			
1	HASH GROUP BY		1479	65076	201 (1)	00:00:03		
2	TABLE ACCESS BY GLOBAL INDEX ROWID	MCST	164	5904	155 (0)	00:00:02	ROW L	ROW L
3	NESTED LOOPS		1479	65076	200 (1)	00:00:03		
4	VIEW		9	72	3 (0)	00:00:01		
5	CONNECT BY WITH FILTERING							
6	TABLE ACCESS BY INDEX ROWID	RT	1	30	2 (0)	00:00:01		
7	INDEX UNIQUE SCAN	RT_PK	1		1 (0)	00:00:01		
8	NESTED LOOPS							
9	CONNECT BY PUMP							
10	TABLE ACCESS BY INDEX ROWID	RT	9	126	3 (0)	00:00:01		
11	INDEX RANGE SCAN	RT_IX1	7		1 (0)	00:00:01		
12	INDEX RANGE SCAN	MCSIX3	164		3 (0)	00:00:01		

## Full SQL Text

SQL Id	SQL Text
5w5a8w9waabh	<pre>SELECT /*+ first_rows */ DECODE(SIGN(MIN_DATE_TIME - :B4), 1, '1', DECODE(SIGN(MIN_DATE_TIME - :B3), 1, '2', DECODE(SIGN(MIN_DATE_TIME - :B2), 1, '3', '4')) DT_RANGE, NVL(COUNT(T1.MC_ID), 0) MC_IDS, NVL(SUM(T1.MAX_COUNT), 0) MAX_COUNT, NVL(SUM(T1.BOX_COUNT), 0) BOX_COUNT, NVL(SUM(T1.BULK_COUNT), 0) BULK_COUNT FROM ( SELECT R_ID FROM RT START WITH R_ID = :B1 CONNECT BY PRIOR R_ID = PARENT_R_ID ) T2, MCST T1 WHERE T2.R_ID = T1.R_ID AND T1.MS_ID = 21 AND T1.MIN_DATE_TIME &gt;= :B5 GROUP BY DECODE(SIGN(MIN_DATE_TIME - :B4), 1, '1', DECODE(SIGN(MIN_DATE_TIME - :B3), 1, '2', DECODE(SIGN(MIN_DATE_TIME - :B2), 1, '3', '4')))</pre>



## AWR Compare Periods Report

- `$ORACLE_HOME/rdbms/admin/awrddrpt.sql`
  - 2 sets of begin/end snaps
- “Green Report”
  - No need to manually run/print two AWR reports

## WORKLOAD REPOSITORY COMPARE PERIOD REPORT

Snapshot Set	DB Name	DB Id	Instance	Inst num	Release	Cluster	Host
First (1st)	PROD2	1309734745	PROD2	1	10.2.0.4.0	NO	db-prod2
Second (2nd)	PROD2	1309734745	PROD2	1	10.2.0.4.0	NO	db-prod2

Snapshot Set	Begin Snap Id	Begin Snap Time	End Snap Id	End Snap Time	Elapsed Time (min)	DB Time (min)	Avg Active Users
1st	17499	18-Jul-10 00:00:54	17595	19-Jul-10 00:00:08	1,439.24	9,442.01	6.56
2nd	17883	22-Jul-10 00:00:01	17979	23-Jul-10 00:00:29	1,440.45	36,968.56	25.69

### Configuration Comparison

	1st	2nd	%Diff
Buffer Cache:	15,664M	15,664M	0.00
Std Block Size:	8K	8K	0.00
Shared Pool Size:	608M	608M	0.00
Log Buffer:	14,328K	14,328K	0.00
SGA Target:	0	0	0.00
PGA Aggregate Target:	3,736M	3,736M	0.00
Undo Management:	AUTO	AUTO	

**Load Profile**

	1st Per Sec	2nd Per Sec	%Diff	1st Per Txn	2nd Per Txn	%Diff
Redo size:	853,063.63	1,078,509.50	26.43	7,184.44	5,237.24	-27.10
Logical reads:	23,899.82	50,224.05	110.14	201.28	243.89	21.17
Block changes:	3,835.72	4,966.54	29.48	32.30	32.30	0.00
Physical reads:	572.08	1,253.87	119.18	4.82	6.09	26.35
Physical writes:	336.70	170.16	-49.46	2.84	0.83	-70.77
User calls:	363.47	644.10	77.21	3.06	3.13	2.29
Parses:	241.57	234.25	-3.03	2.03	1.14	-43.84
Hard parses:	0.02	0.10	400.00	0.00	0.00	0.00
Sorts:	135.83	236.77	74.31	1.14	1.15	0.88
Logons:	0.93	1.21	30.11	0.01	0.01	0.00
Executes:	1,549.35	2,348.24	51.56	13.05	11.40	-12.64
Transactions:	118.74	205.93	73.43			
				1st	2nd	Diff
% Blocks changed per Read:				16.05	9.89	-6.16
Recursive Call %:				84.41	81.40	-3.02
Rollback per transaction %:				0.18	0.17	-0.02
Rows per Sort:				125.24	28.31	-96.93
Avg Length of Calls (sec):				0.02	0.02	0.00

**Top 5 Timed Events**

1st					2nd				
Event	Waits	Time(s)	Percent Total DB Time	Wait Class	Event	Waits	Time(s)	Percent Total DB Time	Wait Class
CPU time		315,827.3	55.75		CPU time		930,402.6	41.95	
log file sync	10,005,664	183,844.3	32.45	Commit	*log file switch completion	485,376	470,322.5	21.20	Configuration
*PX Deq Credit: send blkd	840,048	40,456.5	7.14	Other	log file sync	17,556,582	313,636.7	14.14	Commit
*db file parallel write	2,198,031	24,620.7	4.35	System I/O	*enq: UL - contention	1,164,912	232,328.2	10.47	Application
db file sequential read	9,162,303	22,934.5	4.05	User I/O	db file sequential read	27,869,366	125,500.0	5.66	User I/O
-enq: UL - contention	418,633	17,611.5	3.11	Application	-db file parallel write	3,879,623	70,528.6	3.18	System I/O
-log file switch completion	3,052	616.4	.11	Configuration	-PX Deq Credit: send blkd	333	0.2	.00	Other

## Wait Events

- Ordered by 'Diff' column of 'DB time %' descending (idle events last)

Event	% DB Time			# Waits/sec (Elapsed Time)			Avg Wait Time (ms)		
	1st	2nd	Diff	1st	2nd	%Diff	1st	2nd	%Diff
log file switch completion	0.11	21.20	21.09	0.04	5.62	13,950.00	201.98	968.99	379.75
log file sync	32.45	14.14	-18.31	115.87	203.14	75.32	18.37	17.86	-2.78
enq: UL - contention	3.11	10.47	7.37	4.85	13.48	177.94	42.07	199.44	374.07
PX Deq Credit: send bldd	7.14	0.00	-7.14	9.73	0.00	-100.00	48.16	0.75	-98.44
buffer busy waits	0.26	4.56	4.30	5.38	13.66	153.90	3.19	85.73	2,587.46
log file parallel write	3.15	1.25	-1.90	85.68	97.85	14.20	2.41	3.27	35.68
read by other session	0.02	1.74	1.72	0.03	26.90	89,566.67	45.89	16.64	-63.74
db file sequential read	4.05	5.66	1.61	106.10	322.46	203.92	2.50	4.50	80.00
db file parallel write	4.35	3.18	-1.17	25.45	44.89	76.39	11.20	18.18	62.32
LGWR wait on LNS	2.21	1.26	-0.95	85.69	97.87	14.21	1.69	3.31	95.86
LGWR-LNS wait on channel	2.14	1.24	-0.90	119.25	162.10	35.93	1.18	1.96	66.10
direct path write temp	0.83	0.01	-0.82	2.42	0.07	-97.11	22.57	46.62	106.56
LNS wait on SENDREQ	1.92	1.16	-0.76	85.69	97.86	14.20	1.47	3.04	106.80
control file sequential read	0.87	0.26	-0.61	1,050.46	1,128.76	7.45	0.05	0.06	20.00
enq: CF - contention	0.07	0.64	0.56	0.01	0.07	600.00	378.77	2,305.82	508.77
log buffer space	0.35	0.00	-0.34	0.03	0.00	-100.00	797.98	392.59	-50.80
log file sequential read	0.36	0.15	-0.21	344.14	374.87	8.93	0.07	0.10	42.86
control file parallel write	0.26	0.05	-0.21	1.40	0.87	-37.86	12.15	16.03	31.93
direct path read	0.17	0.03	-0.14	4.51	3.77	-16.41	2.47	1.72	-30.36

## Time Model Statistics

- Ordered by 'Diff' column of '% DB Time' descending (DB Time statistic first, background statistics last)

Statistic Name	% DB Time			Time (seconds)		Time per Trans (seconds)		
	1st	2nd	Diff	1st	2nd	1st	2nd	%Diff
DB time	100.00	100.00	0.00	566,520.56	2,218,113.61	0.06	0.12	100.00
sql execute elapsed time	59.95	82.73	22.78	339,629.89	1,834,938.88	0.03	0.10	233.33
DB CPU	55.75	41.95	-13.80	315,827.26	930,402.61	0.03	0.05	66.67
PL/SQL execution elapsed time	5.34	11.58	6.24	30,231.94	256,826.16	0.00	0.01	100.00
RMAN cpu time (backup/restore)	0.34	0.00	-0.34	1,928.57	0.00	0.00	0.00	0.00
inbound PL/SQL rpc elapsed time	0.13	0.00	-0.13	764.21	0.00	0.00	0.00	0.00
repeated bind elapsed time	0.26	0.14	-0.13	1,490.12	3,029.39	0.00	0.00	0.00
hard parse elapsed time	0.03	0.12	0.08	181.86	2,570.54	0.00	0.00	0.00
connection management call elapsed time	0.14	0.06	-0.08	779.25	1,240.02	0.00	0.00	0.00
parse time elapsed	0.28	0.20	-0.08	1,580.09	4,394.16	0.00	0.00	0.00
failed parse elapsed time	0.00	0.08	0.08	0.21	1,703.99	0.00	0.00	0.00
sequence load elapsed time	0.04	0.06	0.02	223.99	1,280.04	0.00	0.00	0.00
Java execution elapsed time	0.01	0.00	-0.00	32.08	42.36	0.00	0.00	0.00
hard parse (sharing criteria) elapsed time	0.01	0.01	0.00	38.27	230.64	0.00	0.00	0.00
hard parse (bind mismatch) elapsed time	0.00	0.00	0.00	0.25	6.14	0.00	0.00	0.00
PL/SQL compilation elapsed time	0.00	0.00	0.00	7.29	32.97	0.00	0.00	0.00
failed parse (out of shared memory) elapsed t	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
background elapsed time	15.20	9.66	-5.54	86,133.26	214,247.84	0.01	0.01	0.00
background cpu time	4.38	1.52	-2.86	24,832.92	33,719.09	0.00	0.00	0.00

## Operating System Statistics

- Ordered by 'Statistic Name'

Statistic Name	Value		per Second (DB Time)			per Second (Elapsed Time)			per Trans		
	1st	2nd	1st	2nd	%Diff	1st	2nd	%Diff	1st	2nd	%Diff
AVG_BUSY_TIME	1,118,389	3,093,995	2	1	-29.44	13	36	176.45	0	0	54.55
AVG_IDLE_TIME	7,512,846	5,544,191	13	2	-81.15	87	64	-26.26	1	0	-57.53
AVG_IOWAIT_TIME	0	0	0	0	0.00	0	0	0.00	0	0	0.00
AVG_SYS_TIME	304,439	463,827	1	0	-61.11	4	5	52.12	0	0	0.00
AVG_USER_TIME	811,081	2,627,294	1	1	-17.48	9	30	223.75	0	0	87.50
BUSY_TIME	35,879,473	99,101,105	63	45	-29.45	415	1,147	175.97	3	6	59.14
IDLE_TIME	240,502,867	177,507,604	425	80	-81.15	2,785	2,054	-26.26	23	10	-57.50
IOWAIT_TIME	0	0	0	0	0.00	0	0	0.00	0	0	0.00
LOAD	-2	1	-0	0	0.00	-0	0	0.00	-0	0	0.00
OS_CPU_WAIT_TIME	562,400	1,397,800	1	1	-36.36	7	16	148.39	0	0	60.00
PHYSICAL_MEMORY_BYTES	0	0	0	0	0.00	0	0	0.00	0	0	0.00
RSRC_MGR_CPU_WAIT_TIME	0	0	0	0	0.00	0	0	0.00	0	0	0.00
SYS_TIME	9,834,602	14,934,543	17	7	-61.23	114	173	51.73	1	1	-12.50
USER_TIME	26,044,871	84,166,562	46	38	-17.45	302	974	222.88	3	5	86.22
VM_IN_BYTES	17,719,296	40,099,840	31	18	-42.20	205	464	126.12	2	2	30.06
VM_OUT_BYTES	0	0	0	0	0.00	0	0	0.00	0	0	0.00

## Top 10 SQL Comparison by Execution Time

- Ordered by 'Diff' column of 'Exec Time % of DB Time' descending.
- 'N/A' indicates no data was captured for the statement in the period
- 'Multiple Plans' column indicates whether more than one plan exists for the statement in the two periods
- Total SQL Execution as a % DB Time First: 59.95%, Second: 82.73%

SQL Id	Exec Time % of DB Time			Exec Time (ms) / Exec		#Exec/sec (DB Time)		CPU Time (ms) / Exec		Physical Reads / Exec		#Rows Processed / Exec		Multiple Plans	SQL Text
	1st	2nd	Diff	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd		
53sn2vrvzvnah	0.10	19.65	19.55	28	10,738	0.04	0.02	17	9,256	0.29	121.12	114.45	114.16	Yes	SELECT t1.mai...
q3ixzp0t4xf4m	12.47	20.85	8.37	12	46	10.36	4.57	7	10	0.35	0.42	1.00	1.00	Yes	begin download.update_ca...
94upmqzvqvmc	0.10	7.94	7.84	7	344	0.14	0.23	0	1	0.04	0.04	1.00	1.00	Yes	UPDATE MBT SET ATTEMPT_...
f5pc12stmwqu	0.48	8.24	7.76	5	337	0.87	0.24	2	2	0.10	0.10	1.00	1.00	Yes	begin download.mark_retr...
7abwsvh6v8c4q	4.05		-4.05	22,932,123		0.00		12,248,072		6,453,162.00		1,884,314.00		Yes	MERGE INTO a_owm.m_campa...
9mkaxfc3uqkzy		3.80	3.80		15,115		0.00		14,681		34.70		18.01	Yes	select t1.raw_sending_ip, t1...
3fnq3qbh9xhqh	0.98	4.47	3.50	2	17	5.33	2.66	1	1	0.14	0.14	1.00	1.00	Yes	INSERT INTO MMR...
gydhqdzs8crx	1.80	5.03	3.23	3	19	5.36	2.67	2	3	0.15	0.16	1.00	1.00	Yes	begin repcheck.upda...
9qw4vtwqkqm5	1.38	4.23	2.85	2	13	6.66	3.30	1	1	0.12	0.13	1.00	1.00	Yes	UPDATE MMT SET INSERT_D...
2cmsqqrqjbst8	2.42		-2.42	13,721,215		0.00		11,476,643		7,154,565.00		161,795.00		Yes	MERGE INTO a_owm.m_campa...
9c9zpc1dnavuq	2.38		-2.38	13,503,268		0.00		10,060,547		3,476,870.00		0.00		Yes	MERGE INTO a_owm.m_campa...
3p2znad6ng6gv	0.09	2.28	2.19	48	3,538	0.02	0.01	40	49	0.45	0.39	0.99	0.87	Yes	begin download.get_retri...
12f0vxpkpgs00	0.10	2.28	2.18	48	3,500	0.02	0.01	40	49	0.45	0.39	37.22	32.77	Yes	SELECT T1.MBID, T1.EMA...
68m495d75p9xb	2.04		-2.04	11,534,691		0.00		8,189,919		10,343,147.00		0.00		Yes	DECLARE v_date2 date ...



## Daily AWR Report

- Look at supplied scripts
- Set parameters
- Run!
- Beware of instance restarts

```
DEFINE awr_rpt_type = 'html'
DEFINE awr_rpt_opt = 8
SET LINESIZE 1500 PAGESIZE 0 TRIMSPOOL ON FEEDBACK OFF VERIFY OFF
SELECT      d.dbid dbid
           ,   d.name db_name
           ,   i.instance_number instid
FROM        v$database d,
           v$instance i;
SELECT      MIN(snap_id) bsnap
FROM        dba_hist_snapshot
WHERE       TRUNC(end_interval_time) = TRUNC(sysdate - 1);
SELECT      MIN(snap_id) esnap
FROM        dba_hist_snapshot
WHERE       TRUNC(end_interval_time) = TRUNC(sysdate);
SELECT      output awr_rpt
FROM        TABLE(dbms_workload_repository.awr_report_html(&dbid,
                  &instid,&bsnap, &esnap, &awr_rpt_opt))

SPOOL      awr_daily.html
/
SPOOL OFF
```

## dbms\_xplan.display\_awr

- Like dbms\_xplan.display\_cursor
  - But it can read what is saved in the repository
- Parameters
  - sql\_id
  - plan\_hash\_value
  - db\_id
  - format

```
SELECT    plan_table_output
FROM      TABLE(dbms_xplan.display_awr('b4am5xf5w8ab8',
                                         null, null, 'ALLSTATS LAST'))
```

PLAN\_TABLE\_OUTPUT

-----  
 SQL\_ID b4am5xf5w8ab8  
 -----

```

WITH job_hist AS ( SELECT      emp.employee_id      ,
jobs.job_title      , job_hist.start_date      ,
job_hist.end_date      , dept.department_name      , loc.city
      , loc.state_province FROM      hr.employees emp      ,
hr.job_history job_hist      , hr.departments dept      ,
hr.locations loc      , hr.jobs jobs WHERE
job_hist.employee_id = emp.employee_id AND jobs.job_id =
job_hist.job_id AND job_hist.department_id = dept.department_id
AND loc.location_id = dept.location_id ) SELECT
dept.department_name      , loc.city      , loc.state_province
      , emp.first_name      , emp.last_name      , emp.hire_date
      , emp.salary      , jobs.job_title      , mgr.first_name
      , mgr.last_name      , job_hist.job_title      ,
job_hist.start_date      , job_hist.end_date      ,
job_hist.department_name      , job_hist.city      ,
job_hist.state_province FROM      hr.employees emp      , hr.jobs
jobs      , hr.departments dept      , hr.locations loc      ,
hr.employees mgr      , job_hist job_hist WHERE emp.job_id =
jobs.job_id AND emp.department_id = dept.department_id AND
loc.location_id = dept.location_id AND emp.manager_id =
mgr.employee_id (+) AND job_hist.employee_id (+) =
emp.employee_id ORDER BY dept.department_name      , loc.city
      , emp.last_name      , job_hist.start_date
  
```

Plan hash value: 3058872810

Id	Operation	Name	E-Rows
0	SELECT STATEMENT		
1	SORT ORDER BY		106
2	NESTED LOOPS OUTER		106
3	HASH JOIN		106
4	TABLE ACCESS FULL	JOBS	19
5	HASH JOIN RIGHT OUTER		106
6	VIEW	index\$_join\$_010	107
7	HASH JOIN		
8	INDEX FAST FULL SCAN	EMP_NAME_IX	107
9	INDEX FAST FULL SCAN	EMP_EMP_ID_PK	107
10	HASH JOIN		106
11	MERGE JOIN		27
12	TABLE ACCESS BY INDEX ROWID	DEPARTMENTS	27
13	INDEX FULL SCAN	DEPT_LOCATION_IX	27
14	SORT JOIN		23
15	TABLE ACCESS FULL	LOCATIONS	23
16	TABLE ACCESS FULL	EMPLOYEES	107
17	VIEW PUSHED PREDICATE		1
18	NESTED LOOPS		
19	NESTED LOOPS		1
20	NESTED LOOPS		1
21	NESTED LOOPS		1
22	TABLE ACCESS BY INDEX ROWID	JOB_HISTORY	1
23	INDEX RANGE SCAN	JHIST_EMPLOYEE_IX	1
24	TABLE ACCESS BY INDEX ROWID	JOBS	1
25	INDEX UNIQUE SCAN	JOB_ID_PK	1
26	TABLE ACCESS BY INDEX ROWID	DEPARTMENTS	1
27	INDEX UNIQUE SCAN	DEPT_ID_PK	1
28	INDEX UNIQUE SCAN	LOC_ID_PK	1
29	TABLE ACCESS BY INDEX ROWID	LOCATIONS	1



# Custom AWR Report

- Available on [www.optimaldba.com](http://www.optimaldba.com)
- Works on 10g
  - Probably 11g as well
- Generates much smaller, more focused report
- Can handle instance restarts
  - Running Totals
  - SQL Deltas

```

*** Databases in AWR Repository ***

DB Name      DBID      Version    RAC
-----
PROD2*       580662250  10.2.0.4.0 RAC

Database Name <PROD2> :
*** Instances for PROD2 in AWR Repository ***

Inst Instance
ID  Name      Start Time                Host Name
-----
1   PROD21    24-OCT-10 07.43.14.000 AM    prod21
2   PROD22    24-OCT-10 07.44.08.000 AM    prod22
3*  PROD23    12-DEC-10 08.23.22.000 AM    prod23

Instance ID <PROD23> :

Current Snapshot Interval for PROD2 ==> 30 minutes
Current Snapshot Retention for PROD2 ==> 30 days

*** Snapshots in AWR repository for Database PROD2 Instance PROD23
*** Weekly by hour (intermediate snapshots not listed)
      Sunday      Monday      Tuesday      Wednesday      Thursday      Friday      Saturday
      03/06/11    03/07/11    03/08/11    03/09/11    03/10/11    03/11/11    03/12/11
-----
00:00      8008      8056      8104      8152      8200      8248      8296
01:00      8010      8058      8106      8154      8202      8250      8298
02:00      8012      8060      8108      8156      8204      8252      8300
03:00      8014      8062      8110      8158      8206      8254      8302
21:00      8050      8098      8146      8194      8242      8290      8338
22:00      8052      8100      8148      8196      8244      8292      8340
23:00      8054      8102      8150      8198      8246      8294      8342

Enter Beginning Snapshot ID : 8104
Enter Ending Snapshot ID   : 8150

```

\*\*\* Report Times \*\*\*

	Minutes	Seconds
Elapsed Time	1,380.00	82,798.00
Database Time	37,851.00	2,271,032.00

\*\*\* Load Profile \*\*\*

Statistic	Value	Per Second	Per Trans
transactions	6,470,084	78.14	
user calls	12,157,572	146.83	1.88
recursive calls	611,425,504	7,384.54	94.50
execute count	239,956,540	2,898.10	37.09
parse count (total)	32,415,343	391.50	5.01
parse count (hard)	10,450	.13	.00
session logical reads	16,745,106,038	202,240.47	2,588.08
physical reads	329,014,060	3,973.70	50.85
physical writes	111,903,106	1,351.52	17.30
db block changes	2,578,033,089	31,136.42	398.45
redo size	579,098,231,648	6,994,108.94	89,503.97
sorts	255,186,556	3,082.04	39.44

\*\*\* Operating System Statistics \*\*\*

\*\*\* O/S Configuration (end) \*\*\*

Statistic	Value
NUM_CPUS	16
PHYSICAL_MEMORY_BYTES	135,334,133,760

\*\*\* O/S Time \*\*\*

Statistic	Centiseconds	Seconds	Seconds Per Second
BUSY_TIME	20,164,233	201,642.33	2.44
SYS_TIME	1,735,081	17,350.81	.21
USER_TIME	17,415,927	174,159.27	2.10
IDLE_TIME	112,286,671	1,122,866.71	13.56

\*\*\* Time Model Statistics \*\*\*

Statistic	Microseconds	Seconds	DB Time
DB time	2,271,032,048,614	2,271,032.05	
DB CPU	196,477,768,657	196,477.77	8.65%
sql execute elapsed time	1,316,371,092,645	1,316,371.09	57.96%
parse time elapsed	1,117,269,463	1,117.27	.05%
hard parse elapsed time	549,044,036	549.04	.02%
PL/SQL execution elapsed time	5,517,228,742	5,517.23	.24%
background elapsed time	163,454,727,598	163,454.73	
background cpu time	17,525,464,011	17,525.46	

\*\*\* Top 10 Timed Events \*\*\*

Event	Seconds	Pct	Avg Time (micro)	Event Count
db file sequential read	961,032.42	41.12	6.90	139,231,272
wait for a undo record	808,979.60	34.61	97.55	8,292,963
CPU time	196,477.77	8.41		
direct path read	67,260.49	2.88	30.84	2,181,199
db file parallel write	33,992.54	1.45	1.39	24,379,850
read by other session	30,710.20	1.31	5.18	5,927,546
db file scattered read	28,718.91	1.23	29.34	978,957
enq: PS - contention	28,024.74	1.20	35.96	779,272
db file parallel read	20,878.76	0.89	15.10	1,382,826
log file parallel write	20,700.37	0.89	2.50	8,286,800

\*\*\* SQL Ranking Overview \*\*\*

SQL ID	Elapsed Time (us)		CPU Time (us)	
08dm6k7xttqhv	545,309,814,033	(1)	74,090,152,025	(1)
fvd82k4u7Eyum	167,295,825,378	(2)	10,681,719,513	(7)
3704g5cnc01rv	134,786,583,969	(3)	11,434,635,916	(6)
1h13dnwjwz5hr	43,058,196,593	(12)	11,837,766,009	(5)
2ddqg88ha6rtq	80,891,592,572	(4)	7,977,199,812	(11)
1dmj1xyblj2tb	41,138,492,873	(13)	15,910,793,890	(4)
a2bjzvnymdf9j*	40,813,524,017	(14)	7,332,733,762	(13)
2dy0wxzsvy3sf	51,744,669,933	(9)	3,821,009,117	(21)
fp9ksqx3x5x0v	65,412,353,478	(5)	2,140,160,517	(29)
flb8d04adg5j3	9,698,411,890	(32)	1,390,837,955	(39)
0a9r76gfjcvpz	26,892,829,092	(17)	24,853,210,285	(2)

Logical I/O	Physical I/O		Executions	Rows	
8,019,818,250	(1)	42,522,949	(1)	306,119	306,119
591,811,559	(11)	14,866,380	(7)	9,486,877	4,634,116
634,635,797	(9)	11,759,435	(10)	10,386,274	2,384,084
866,029,560	(6)	4,202,091	(21)	278	278
115,990,626	(28)	40,779,521	(2)	0	0
2,183,102,137	(4)	2,407,631	(30)	16,787,833	146,662
934,939,045	(5)	3,269,278	(25)	16,789,911	6,605,527
37,474,438	(44)	15,569,294	(5)	0	0
41,502,338	(42)	10,578,566	(11)	0	0
17,705,670	(58)	16,059,979	(3)	6	2,268
5,739,665,984	(2)	43,287	(139)	44,425	44,425

```

*** SQL Executions ***
08dm6k7xttpqhv      BEGIN LOADER_API.Insert_Records( :1,:2,:3,:4,:5,:6,:7,:8,:9,:10,
Plan Hash          ==> 0
                    Executions      ==> 306,119
                    Ela Time(s)     ==> 545,309.814033
                    Ela per Exec    ==> 1.781365
                    Logical I/O     ==> 8,019,818,250
                    Physical I/O    ==> 42,522,949
Rows               ==> 306,119      Rows per Exec ==> 1
CPU Time(us)      ==> 74,090.152025 Wait Time(us) ==> 471,219.662008
CPU per Exec      ==> 0.242031     Wait per Exec ==> 1.539335
LIO per Exec      ==> 26,198      LIO per Row  ==> 26,198
PIO per Exec      ==> 139         PIO per Row  ==> 139

a2bjzvnymdf9j      SELECT NVL (SC_SESSION_ID, 0), NVL (PLAY_TRANSACTION_ID, 0)
Plan Hash          ==> 2281194519
                    Executions      ==> 11,251,902
                    Ela Time(s)     ==> 38,768.015323
                    Ela per Exec    ==> 0.003445
                    Logical I/O     ==> 915,912,716
                    Physical I/O    ==> 3,092,972
Rows               ==> 4,559,168   Rows per Exec ==> 0
CPU Time(us)      ==> 6,446.896198 Wait Time(us) ==> 32,321.119125
CPU per Exec      ==> 0.000573     Wait per Exec ==> 0.002873
LIO per Exec      ==> 81          LIO per Row  ==> 201
PIO per Exec      ==> 0           PIO per Row  ==> 1

Plan Hash          ==> 3700437993
                    Executions      ==> 5,517,957
                    Ela Time(s)     ==> 2,020.368272
                    Ela per Exec    ==> 0.000366
                    Logical I/O     ==> 18,873,452
                    Physical I/O    ==> 174,254
Rows               ==> 2,037,607   Rows per Exec ==> 0
CPU Time(us)      ==> 884.119810   Wait Time(us) ==> 1,136.248462
CPU per Exec      ==> 0.000160     Wait per Exec ==> 0.000206
LIO per Exec      ==> 3           LIO per Row  ==> 9
PIO per Exec      ==> 0           PIO per Row  ==> 0

```



## AWR Data Repository

- Data from multiple systems
- Cannot run standard reports
  - sql scripts use current db/instance
  - Use dbms\_workload\_repository



## Transporting AWR data

- Available in 10.2
- Extract
  - awrextr.sql
- Load
  - awrload.sql
  - Can import 10.2 data into 11g
  - Don't include .dmp when prompted for file

```
@?/rdbms/admin/awrextr.sql
~~~~~
AWR EXTRACT
~~~~~
~~~~~
~ This script will extract the AWR data for a range of snapshots ~
~ into a dump file. The script will prompt users for the ~
~ following information: ~
~ (1) database id ~
~ (2) snapshot range to extract ~
~ (3) name of directory object ~
~ (4) name of dump file ~
~~~~~

Databases in this Workload Repository schema
~~~~~

  DB Id      DB Name      Host
-----
* 3036635614 PROD          prod_db

The default database id is the local one: '3036635614'. To use this
database id, press <return> to continue, otherwise enter an alternative.

Enter value for dbid:

Using 3036635614 for Database ID
```

Specify the number of days of snapshots to choose from  
~~~~~  
Entering the number of days (n) will result in the most recent  
(n) days of snapshots being listed. Pressing <return> without  
specifying a number lists all completed snapshots.

Enter value for num\_days: 7

Listing the last 7 days of Completed Snapshots

| DB Name | Snap Id | Snap Started      |
|---------|---------|-------------------|
| PROD    | 31700   | 29 Aug 2010 00:00 |
|         | 31701   | 29 Aug 2010 01:00 |
| ...     |         |                   |
|         | 31859   | 04 Sep 2010 15:00 |
|         | 31860   | 04 Sep 2010 16:00 |

Specify the Begin and End Snapshot Ids  
~~~~~

Enter value for begin\_snap: 31844  
Begin Snapshot Id specified: 31844

Enter value for end\_snap: 31860  
End Snapshot Id specified: 31860

Specify the Directory Name

~~~~~

| Directory Name | Directory Path                            |
|----------------|-------------------------------------------|
| DATA_PUMP_DIR  | /opt/oracle/product/10.2.0/db_1/rdbms/log |
| PUMP_WORK_DIR  | /u4/app/oracle/product/10.2.0/work        |
| WORK_DIR       | /home/oracle                              |

Choose a Directory Name from the above list (case-sensitive).

Enter value for directory\_name: WORK\_DIR

Using the dump directory: WORK\_DIR

Specify the Name of the Extract Dump File

~~~~~

The prefix for the default dump file name is awrdat\_31844\_31860.  
To use this name, press <return> to continue, otherwise enter  
an alternative.

Enter value for file\_name:

Using the dump file prefix: awrdat\_31844\_31860

```
~~~~~  
The AWR extract dump file will be located  
in the following directory/file:  
  /home/oracle  
  awrdat_31844_31860.dmp  
~~~~~
```

```
*** AWR Extract Started ...
```

```
This operation will take a few moments. The  
progress of the AWR extract operation can be  
monitored in the following directory/file:  
  /home/oracle  
  awrdat_31844_31860.log
```

```
End of AWR Extract  
PROD1>
```

```
@?/rdbms/admin/awrload
~~~~~
AWR LOAD
~~~~~
~ This script will load the AWR data from a dump file. The ~
~ script will prompt users for the following information: ~
~ (1) name of directory object ~
~ (2) name of dump file ~
~ (3) staging schema name to load AWR data into ~
~~~~~

Specify the Directory Name
~~~~~

Directory Name          Directory Path
-----
DATA_PUMP_DIR          C:\oracle\admin\t500dwf\dpdump\
IDR_DIR                c:\oracle\diag\rdbms\t500dwf\t500dwf\ir

Choose a Directory Name from the list above (case-sensitive).

Enter value for directory_name: DATA_PUMP_DIR

Using the dump directory: DATA_PUMP_DIR
```

Specify the Name of the Dump File to Load

~~~~~

Please specify the prefix of the dump file (.dmp) to load:

Enter value for file\_name: awrdat\_31672\_31860

Loading from the file name: awrdat\_31672\_31860.dmp

Staging Schema to Load AWR Snapshot Data

~~~~~

The next step is to create the staging schema

where the AWR snapshot data will be loaded.

After loading the data into the staging schema,

the data will be transferred into the AWR tables

in the SYS schema.

The default staging schema name is AWR\_STAGE.

To use this name, press <return> to continue, otherwise enter  
an alternative.

Enter value for schema\_name:

Using the staging schema name: AWR\_STAGE

Choose the Default tablespace for the AWR\_STAGE user

~~~~~

Choose the AWR\_STAGE users's default tablespace. This is the tablespace in which the AWR data will be staged.

| TABLESPACE_NAME | CONTENTS  | DEFAULT TABLESPACE |
|-----------------|-----------|--------------------|
| -----           | -----     | -----              |
| EXAMPLE         | PERMANENT |                    |
| SYSAUX          | PERMANENT | *                  |
| USERS           | PERMANENT |                    |

Pressing <return> will result in the recommended default tablespace (identified by \*) being used.

Enter value for default\_tablespace:

Using tablespace SYSAUX as the default tablespace for the AWR\_STAGE

Choose the Temporary tablespace for the AWR\_STAGE user

~~~~~

Choose the AWR\_STAGE user's temporary tablespace.

TABLESPACE_NAME	CONTENTS	DEFAULT TEMP TABLESPACE
-----	-----	-----
TEMP	TEMPORARY	*

Pressing <return> will result in the database's default temporary tablespace (identified by \*) being used.

Enter value for temporary\_tablespace:

Using tablespace TEMP as the temporary tablespace for AWR\_STAGE

```
... Creating AWR_STAGE user

|
| ~~~~~
| Loading the AWR data from the following
| directory/file:
|   C:\oracle\admin\t500dwf\dpdump\
|   awrdat_31672_31860.dmp
| ~~~~~
|
| *** AWR Load Started ...
|
| This operation will take a few moments. The
| progress of the AWR load operation can be
| monitored in the following directory/file:
|   C:\oracle\admin\t500dwf\dpdump\
|   awrdat_31672_31860.log
|
| ... Dropping AWR_STAGE user
| End of AWR Load
|
| select distinct dbid from dba_hist_snapshot;
| 999617550
| 3036635614
```



# Errors!

- Incorrect file name
  - Don't include .dmp when prompted for file
- Importing awr data into it's own repository

```
Specify the Name of the Dump File to Load
~~~~~
Please specify the prefix of the dump file (.dmp) to load:

Enter value for file_name: AWRDAT_250_261.DMP

Loading from the file name: AWRDAT_250_261.DMP.dmp
<<error>>
DBMS_DATAPUMP.ADD_FILE(dump file)
ORA-39001: invalid argument value
Exception encountered in AWR_LOAD
begin
*
ERROR at line 1:
ORA-20115: datapump import encountered error:
ORA-39001: invalid argument value
ORA-39000: bad dump file specification
ORA-31640: unable to open dump file
  "C:\oracle\admin\t500dwf\dpdump\AWRDAT_250_261.DMP.dmp" for
  read
ORA-27041: unable to open file
OSD-04002: unable to open file
O/S-Error: (OS 2) The system cannot find the file specified.
ORA-06512: at "SYS.DBMS_SWRF_INTERNAL", line 1713
ORA-06512: at line 3
```

```
Specify the Name of the Dump File to Load
~~~~~
Please specify the prefix of the dump file (.dmp) to load:

Enter value for file_name: AWRDAT_250_261

Loading from the file name: AWRDAT_250_261.dmp
<<error>>
begin
*
ERROR at line 1:
ORA-20105: unable to move AWR data to SYS
ORA-06512: at "SYS.DBMS_SWRF_INTERNAL", line 2709
ORA-20107: not allowed to move AWR data for local dbid
ORA-06512: at line 3
```



# Custom Reporting

- Querying AWR/ASH
- Trend Analysis
- Data Visualization

\*Some application to statspack



## Customizing ASH/AWR Reporting

- Data stored in documented tables
  - Trending, profiling, alerting possible with custom scripting
- Build a corporate dba repository
  - Keep little online in production
  - Long term trending/forecasting
  - Will require custom scripting
- Find your System's Top 5

# Views

## ■ ASH

- v\$active\_session\_history
- dba\_hist\_active\_sess\_history

## ■ AWR

- dba\_hist\_snapshot
- dba\_hist\_system\_event
- dba\_hist\_sysstat
- dba\_hist\_sys\_time\_model



# Data Analysis

- Profiling
- Variance
- Trending



# Profiling

- Profile
  - Normal
  - Baseline
- Key to identifying variance
  - Without knowing 'normal' how do you know what is 'abnormal'
- Daily/Weekly AWR/ASH



# Variance

- Varying performance = frustrated users
- Compare to baselines/profiles
- Supplied Reports
- Case studies



# Querying AWR/ASH

- Analytical SQL
- ASH
  - buffer
  - snapshots
- AWR
  - dba\_hist% views
  - Tables partitioned on DBID and SNAP\_ID



# Functions

- Documented in SQL Reference and Data Warehousing Guide
- Some functions have the same name as non-analytical functions (avg)
  - Always identified by the OVER clause
- LAG – previous row value
- AVG – sliding average



# Partitioning

- *Not to be confused with table/index partitioning*
- A set of rows grouped by defined value or values
  - Default partition is entire result set
  - Functions are applied relative to the partition
  - “DAY HH24”



# Windowing

- Range of rows relative to the current row
  - rows between unbounded preceding and 1 preceding
- Used to
  - calculate running total
  - calculate relative average
- Default is all rows in result set
  - Don't forget partitioning!



# LAG

LAG (<expression>) OVER  
(PARTITION BY <expression>  
ORDER BY <expression>)

- Previous row value
- Running totals
  - current\_value – LAG()
- What's the right PARTITION?



# AVG

AVG(<expression>) OVER  
(PARTITION BY <expression>  
ORDER BY <expression>  
ROWS UNBOUNDED PRECEDING)

- 'Sliding' Average
- Will not cross partition boundary



## Filter limitation

- Cannot be used to filter records
- WHERE (filter predicate) is applied BEFORE functions are processed
- Then...how can I create a Top 5 Query?
  - With WITH\*
  - \*no this is not a typo...



## Just SQL

- Analytical sql not always required
- Use the data
- Take the meaningful metrics and use them



## Active Sessions By Day

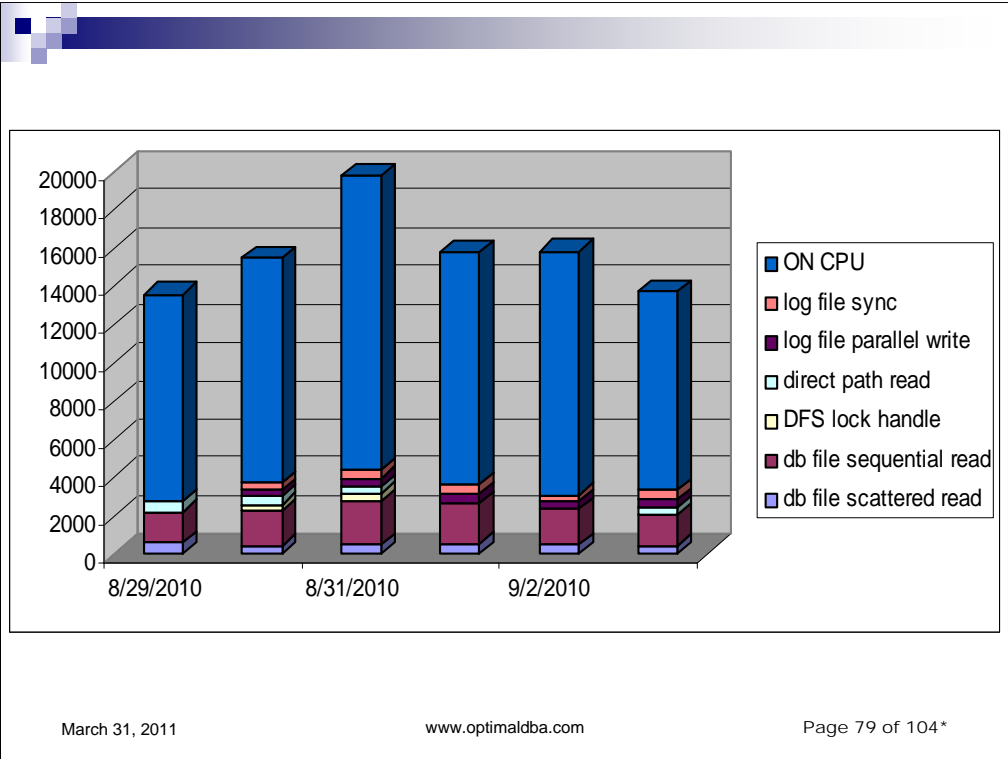
- Total active sessions by day
  - Actually, total database activity time!
- Trending
  - Busy days
  - Increasing use
  - Spikes/Variations

```

SELECT      TRUNC(d.sample_time) sample_day
           , NVL(d.event, d.session_state) event
           , COUNT(1) active_sessions
FROM        dba_hist_active_sess_history d
WHERE       d.sample_time BETWEEN '28-AUG-10'
           AND '03-SEP-10'
GROUP BY   TRUNC(d.sample_time)
           , NVL(d.event, d.session_state)
HAVING     COUNT(1) > 300
ORDER BY   sample_day, active_sessions desc;

```

SAMPLE_DA	EVENT	ACTIVE_SESSIONS
-----	-----	-----
28-AUG-10	ON CPU	7570
28-AUG-10	db file sequential read	1238
28-AUG-10	RMAN backup & recovery I/O	404
28-AUG-10	db file scattered read	381
29-AUG-10	ON CPU	10854
29-AUG-10	db file sequential read	1655
29-AUG-10	db file scattered read	569
29-AUG-10	direct path read	499





# Active Sessions By Hour

- Total active sessions by hour
  - Actually, total database activity time!
- Trending
  - Busy times
  - Increasing use/slow response?
  - Spikes/Variations
  - Times for resource intensive maintenance

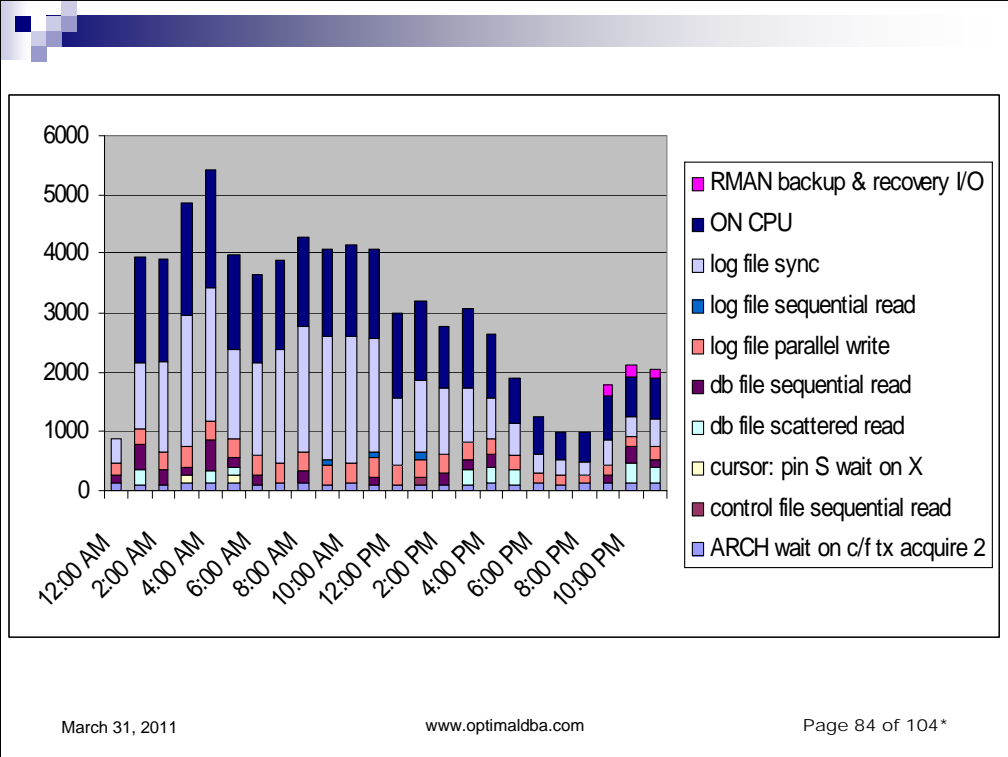
```

WITH act_sess_count AS
( SELECT      TO_DATE(TO_CHAR(d.sample_time,'YYYYMMDD hh24'),
                    'YYYYMMDD hh24') sample_hour
      ,      NVL(d.event, d.session_state) event
      ,      COUNT(1) active_sessions
FROM      dba_hist_active_sess_history d
GROUP BY  TO_DATE(TO_CHAR(d.sample_time,'YYYYMMDD hh24'),
                    'YYYYMMDD hh24'),
          NVL(d.event, d.session_state)
),
act_sess_avg AS
( SELECT      sample_hour
      ,      event
      ,      active_sessions
      ,      ROUND(AVG(active_sessions) OVER
                    (PARTITION BY
                     TO_CHAR(sample_hour,'DY hh24'),
                     event
                     ORDER BY sample_hour
                     ROWS BETWEEN UNBOUNDED PRECEDING
                     AND 1 PRECEDING)) active_sessions_daily_avg
FROM      act_sess_count
)
---continued on next slide

```

```
---continued from previous slide
SELECT      TO_CHAR(sample_hour, 'MM/DD hh24:mi')
            time_sample
            ,   event
            ,   active_sessions
            ,   active_sessions_daily_avg
            ,   ROUND(((active_sessions -
                        active_sessions_daily_avg)/
                        active_sessions_daily_avg) *100) var_pct
FROM        act_sess_avg
ORDER BY    time_sample, event;
```

Time	Event	Active Sessions	Avg Sessions	Variance
09/01 10:00	ARCH wait on c/f tx acquire 2	119	133	-11
	ON CPU	1529	1197	28
	control file sequential read	88	80	10
	db file sequential read	90	364	-75
	log file parallel write	324	315	3
	log file sequential read	87	70	24
	log file sync	2166	2480	-13
09/01 11:00	ARCH wait on c/f tx acquire 2	104	127	-18
	ON CPU	1509	1095	38
	control file sequential read	88	87	1
	db file sequential read	136	289	-53
	log file parallel write	328	312	5
	log file sequential read	100	67	49
	log file sync	1896	1816	4





# Average Event Duration

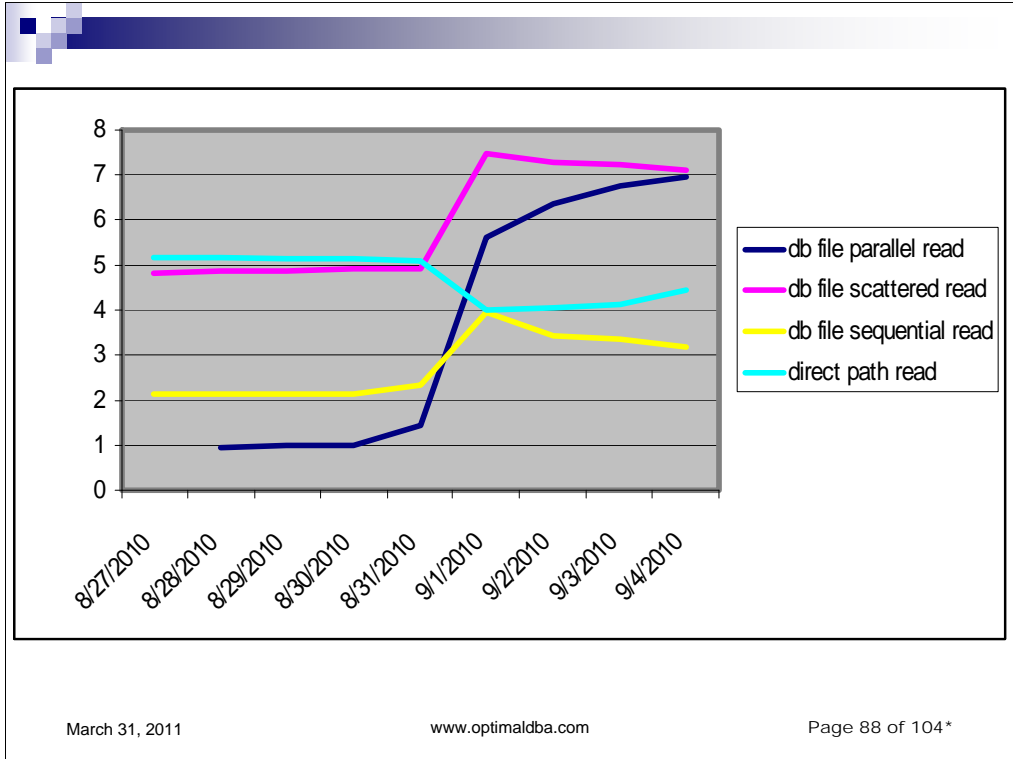
- Average Daily duration
  - Physical i/o
- Trending
  - Spikes/Variations
  - Changes causing changes

```

WITH snap_range AS
( SELECT      MIN(snap_id) bsnap
  ,          MIN(end_interval_time) btime
  ,          MAX(snap_id) esnap
  ,          MAX(end_interval_time) etime
  FROM      dba_hist_snapshot
), event_delta AS
( SELECT      hse.snap_id,      hse.event_name
  ,          (LAG(hse.total_waits) OVER
              (PARTITION BY hse.event_name
               ORDER BY hse.snap_id)) -
              hse.total_waits total_waits
  ,          (LAG(hse.time_waited_micro) OVER
              (PARTITION BY hse.event_name
               ORDER BY hse.snap_id)) -
              hse.time_waited_micro time_waited_micro
  FROM      dba_hist_system_event hse
  ,          snap_range sr
  WHERE     hse.event_name IN ('db file parallel read`
                              , 'direct path read`
                              , 'db file sequential read`
                              , 'db file scattered read`)
  AND      hse.snap_id BETWEEN sr.bsnap AND sr.esnap
)
---continued on next slide

```

```
---continued from previous slide
SELECT      TRUNC(s.end_interval_time) snap_time
            ,      ed.event_name
            ,      ROUND(AVG((ed.time_waited_micro /
            ed.total_waits)/1000),2) avg_wait_millis
FROM        event_delta ed
            ,      dba_hist_snapshot s
WHERE       ed.total_waits > 0
            AND    ed.snap_id = s.snap_id
GROUP BY    TRUNC(s.end_interval_time), ed.event_name
ORDER BY    TRUNC(s.end_interval_time), ed.event_name;
```





# Space Forecasting

- **DBA\_HIST\_TBSPC\_SPACE\_USAGE**
  - `tablespace_id = dba_hist_tablespace_stats.ts#`
  - Units are blocks
- How many days until I run out?
- What about 30 day (or more) space?

TSNAME	MAX_MEGS	USED_MEGS	FREE_MEGS	AVG_MEGS	FREE_DAYS	DAY30_MEGS
METRICS_DATA	21480	21235	245	156	2	25918
METRICS_INDEX	19036	18362	674	132	6	22316
PLANMGR_DATA	50	0	50	0	50	0
MPD01	8329	6791	1538	27	58	7598
MPX01	5215	4275	940	16	59	4761
DATA_MISC	4844	1297	3547	27	131	2110
INDEX_MISC	2527	624	1903	12	155	993
SYSTEM	2048	975	1073	0	1073	975
SYSAUX	9392	3137	6255	5	1159	3299
USERS	2048	19	2029	0	2029	19

```

WITH    day_last_snap AS
( SELECT      s.dbid
            , MAX(s.snap_id) last_snap_id
            , TRUNC(s.begin_interval_time) snap_day
      FROM    dba_hist_snapshot s
      GROUP BY s.dbid, TRUNC(s.begin_interval_time)
    ),
usedspace_deltas AS
( SELECT      s.last_snap_id
            , s.snap_day
            , t.tablespace_id
            , tn.tsname
            , ROUND(t.tablespace_maxsize * 8192 / 1048576)
              tablespace_maxsize
            , ROUND(t.tablespace_usedsize * 8192 / 1048576)
              tablespace_usedsize
            , ROUND(( t.tablespace_usedsize - LAG (t.tablespace_usedsize)
              OVER (PARTITION BY t.tablespace_id
                    ORDER BY s.last_snap_id)) * 8192 / 1048576) day_space_used
      FROM    dba_hist_tbsp_space_usage t
            , day_last_snap s
            , dba_hist_tablespace_stat tn
      WHERE   t.dbid = 1637742403
            AND s.dbid = t.dbid
            AND s.last_snap_id = t.snap_id
            AND tn.dbid = t.dbid
            AND tn.snap_id = t.snap_id
            AND tn.ts# = t.tablespace_id),

```

```

---continued from previous slide
usedspace_metrics AS
( SELECT      ud.last_snap_id
      , ud.tablespace_id
      , ud.tsname
      , ud.tablespace_maxsize
      , ud.tablespace_usedsize
      , ud.day_space_used
      , AVG(ud.day_space_used) OVER (PARTITION BY ud.tsname
      ORDER BY ud.last_snap_id ROWS UNBOUNDED PRECEDING)
      avg_day_space_used
FROM      usedspace_deltas ud
WHERE     ud.day_space_used IS NOT NULL
),
SELECT      um.tsname
      , um.tablespace_maxsize ts_max_megs
      , um.tablespace_usedsize ts_used_megs
      , um.tablespace_maxsize - um.tablespace_usedsize ts_free_megs
      , ROUND(um.avg_day_space_used) avg_daily_used_megs
      , CEIL((um.tablespace_maxsize - um.tablespace_usedsize) /
      DECODE(um.avg_day_space_used,0,1,um.avg_day_space_used))
      free_days
      , um.tablespace_usedsize + (um.avg_day_space_used*30) day30_megs
      , um.tablespace_usedsize + (um.avg_day_space_used*60) day60_megs
      , um.tablespace_usedsize + (um.avg_day_space_used*90) day90_megs
FROM      usedspace_metrics um
WHERE     um.last_snap_id = 22200
ORDER BY  free_days ASC

```



# Visualizing Data

- Easy to spot patterns

- 10g

- Data -> spreadsheet
    - Pivot table
    - Stacked chart

- 11g

- Pivot query?
    - Stacked chart

Microsoft Excel - active\_sessions.xls

File Edit View Insert Format Tools Data Window Help Adobe PDF

B74 PX Deq Credit: send blkd

	A	B	C	D	E
1	Time	Event	Active Sessions		
2	7/11/2010 0:00	log file sync	27232		
3	7/11/2010 0:00	ON CPU	13994		
4	7/11/2010 0:00	Streams AQ: enqueue blocked due to flow	1620		
5	7/11/2010 0:00	enq: UL - contention	1290		
6	7/11/2010 0:00	db file sequential read	1090		
7	7/11/2010 0:00	log buffer space	545		
8	7/11/2010 0:00	buffer busy waits	323		
9	7/11/2010 0:00	PX Deq Credit: send blkd	144		
10	7/11/2010 0:00	db file scattered read	144		
11	7/11/2010 0:00	enq: TX - index contention	112		
12	7/12/2010 0:00	log file sync	30767		
13	7/12/2010 0:00	ON CPU	20579		
14	7/12/2010 0:00	enq: UL - contention	2074		
15	7/12/2010 0:00	db file sequential read	1482		
16	7/12/2010 0:00	log buffer space	262		
17	7/12/2010 0:00	buffer busy waits	241		
18	7/12/2010 0:00	log file switch completion	113		
19	7/12/2010 0:00	db file scattered read	108		

Microsoft Excel - active\_sessions.xls

File Edit View Insert Format Tools Data Window Help Adobe PDF

B2 log file sync

	A		D	E
1	Time	Event		
2	7/11/2010 0:00	log file sync		
3	7/11/2010 0:00	ON CPU		
4	7/11/2010 0:00	Streams AQ: enqueue		
5	7/11/2010 0:00	enq: UL - contention		
6	7/11/2010 0:00	db file sequential read		
7	7/11/2010 0:00	log buffer space		
8	7/11/2010 0:00	buffer busy waits		
9	7/11/2010 0:00	PX Deq Credit: send bl		
10	7/11/2010 0:00	db file scattered read		
11	7/11/2010 0:00	enq: TX - index content		
12	7/12/2010 0:00	log file sync		
13	7/12/2010 0:00	ON CPU	20579	
14	7/12/2010 0:00	enq: UL - contention	2074	
15	7/12/2010 0:00	db file sequential read	1482	
16	7/12/2010 0:00	log buffer space	262	
17	7/12/2010 0:00	buffer busy waits	241	
18	7/12/2010 0:00	log file switch completion	113	
19	7/12/2010 0:00	db file scattered read	108	

Sort...  
Filter  
Form...  
Subtotals...  
Validation...  
Table...  
Text to Columns...  
Consolidate...  
Group and Outline  
PivotTable and PivotChart Report...  
Import External Data  
List  
XML  
Refresh Data

Microsoft Excel - active\_sessions.xls

File Edit View Insert Format Tools Data Window Help Adobe PDF

B2 log file sync

	A	B	C	D
1	Time	Event	Active Sessions	
2	7/11/2010 0:00	log file sync	27232	
3	7/11/2010 0:00	ON CPU	13994	
4	7/11/2010 0:00	Streams AQ: enqueue blocked due to flow	1620	
5	7/11/2010 0:00	enq: UL - contention	1290	
6	7/11/2010 0:00	db file sequential read	1090	
7	7/11/2010 0:00			
8	7/11/2010 0:00			
9	7/11/2010 0:00			
10	7/11/2010 0:00			
11	7/11/2010 0:00			
12	7/12/2010 0:00			
13	7/12/2010 0:00			
14	7/12/2010 0:00			
15	7/12/2010 0:00			
16	7/12/2010 0:00			
17	7/12/2010 0:00			
18	7/12/2010 0:00			
19	7/12/2010 0:00			
20	7/13/2010 0:00			
21	7/13/2010 0:00			
22	7/13/2010 0:00			
23	7/13/2010 0:00			

**PivotTable and PivotChart Wizard - Step 1 of 3**

Where is the data that you want to analyze?

- Microsoft Office Excel list or database
- External data source
- Multiple consolidation ranges
- Another PivotTable report or PivotChart report

What kind of report do you want to create?

- PivotTable
- PivotChart report (with PivotTable report)

Cancel < Back Next > Finish

Microsoft Excel - active\_sessions.xls

File Edit View Insert Format Tools Data Window Help Adobe PDF

A1 log file sync

	A	B	C	D	E
1	Time	Event	Active Sessions		
2	7/11/2010 0:00	log file sync	27232		
3	7/11/2010 0:00	ON CPU	13994		
4	7/11/2010 0:00	Streams AQ: enqueue blocked due to flow	1620		
5	7/11/2010 0:00	enq: UL - contention	1290		
6	7/11/2010 0:00	db file sequential read	1090		
7	7/11/2010 0:00	log buffer space	545		
8	7/11/2010 0:00	buffer busy waits	323		
9	7/11/2010 0:00	PX Deq Credit: send blkd	144		
10	7/11/2010 0:00	db file scattered read	144		
11	7/11/2010 0:00	enq: TX - index contention	112		
12	7/12/2010 0:00	log file sync	20767		
13	7/12/2010 0:00	ON CPU			
14	7/12/2010 0:00	enq: UL - contention			
15	7/12/2010 0:00	db file sequential read			
16	7/12/2010 0:00	log buffer space			
17	7/12/2010 0:00	buffer busy waits			
18	7/12/2010 0:00	log file switch completion	113		
19	7/12/2010 0:00	db file scattered read	108		

PivotTable and PivotChart Wizard - Step 2 of 3

Where is the data that you want to use?

Range:  Browse...

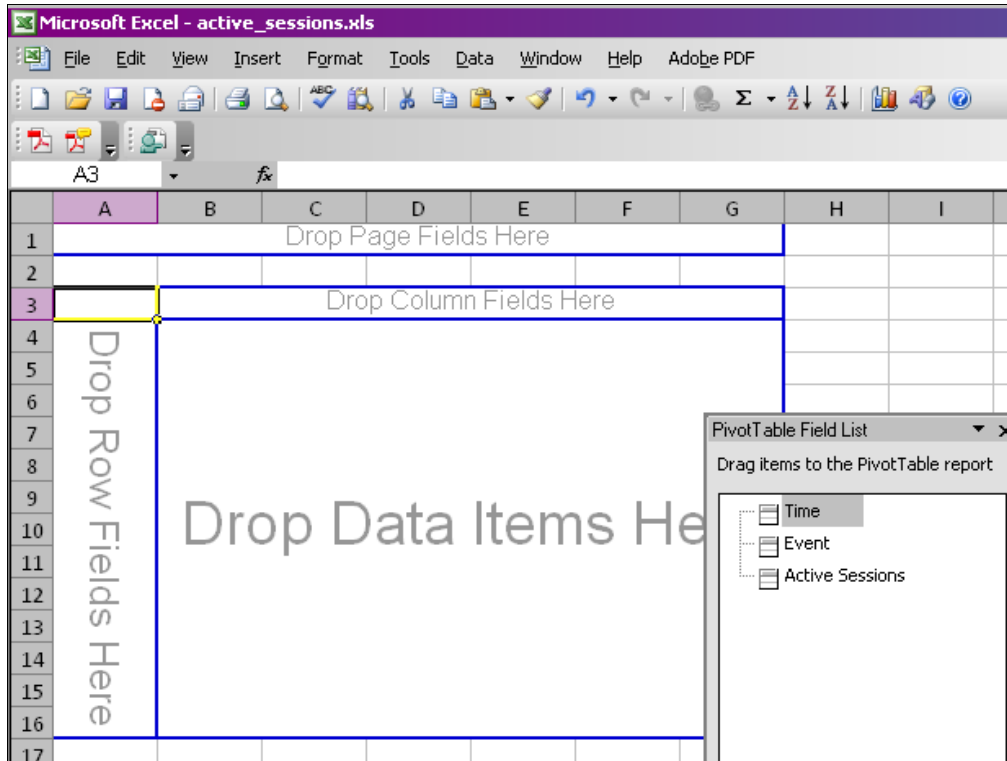
Cancel < Back Next > Finish

Microsoft Excel - active\_sessions.xls

File Edit View Insert Format Tools Data Window Help Adobe PDF

B2 log file sync

	A	B	C	D	E
1	Time	Event	Active Sessions		
2	7/11/2010 0:00	log file sync	27232		
3	7/11/2010 0:00	ON CPU	13994		
4	7/11/2010 0:00	Streams AQ: enqueue blocked due to flow	1620		
5	7/11/2010 0:00	enq: UL - contention	1290		
6	7/11/2010 0:00	db file sequential read	1090		
7	7/11/2010 0:00	log buffer space	545		
8	7/11/2010 0:00	buffer busy waits	323		
9	7/11/2010 0:00	DB Dba Credits cond blkd	144		
10	7/11/2010	PivotTable and PivotChart Wizard - Step 3 of 3			
11	7/11/2010	Where do you want to put the PivotTable report?			
12	7/12/2010	<input checked="" type="radio"/> New worksheet <input type="radio"/> Existing worksheet			
13	7/12/2010	<input type="text"/>			
14	7/12/2010	Click Finish to create your PivotTable report.			
15	7/12/2010	<input type="button" value="Layout..."/> <input type="button" value="Options..."/> <input type="button" value="Cancel"/> <input type="button" value=" &lt; Back"/> <input type="button" value=" Next &gt;"/> <input type="button" value=" Finish"/>			
16	7/12/2010				
17	7/12/2010				
18	7/12/2010				
19	7/12/2010				
20	7/13/2010				



Microsoft Excel - active\_sessions.xls

File Edit View Insert Format Tools Data Window Help Adobe PDF

A4 fx Time

	A	B	C
1			
2			
3	Sum of Active Sessions	Event	
4	Time	buffer busy waits	db file scattered read
5	7/11/2010 0:00		323
6	7/12/2010 0:00		241
7	7/13/2010 0:00		
8	7/14/2010 0:00		
9	7/15/2010 0:00		
10	7/16/2010 0:00		
11	7/17/2010 0:00		
12	Grand Total		
13			
14			
15			
16			
17			
18			
19			
20			
21			

PivotTable Field List

Drag items to the PivotTable report

- Time
- Event
- Active Sessions

Add To Row Area

Microsoft Excel - active\_sessions.xls

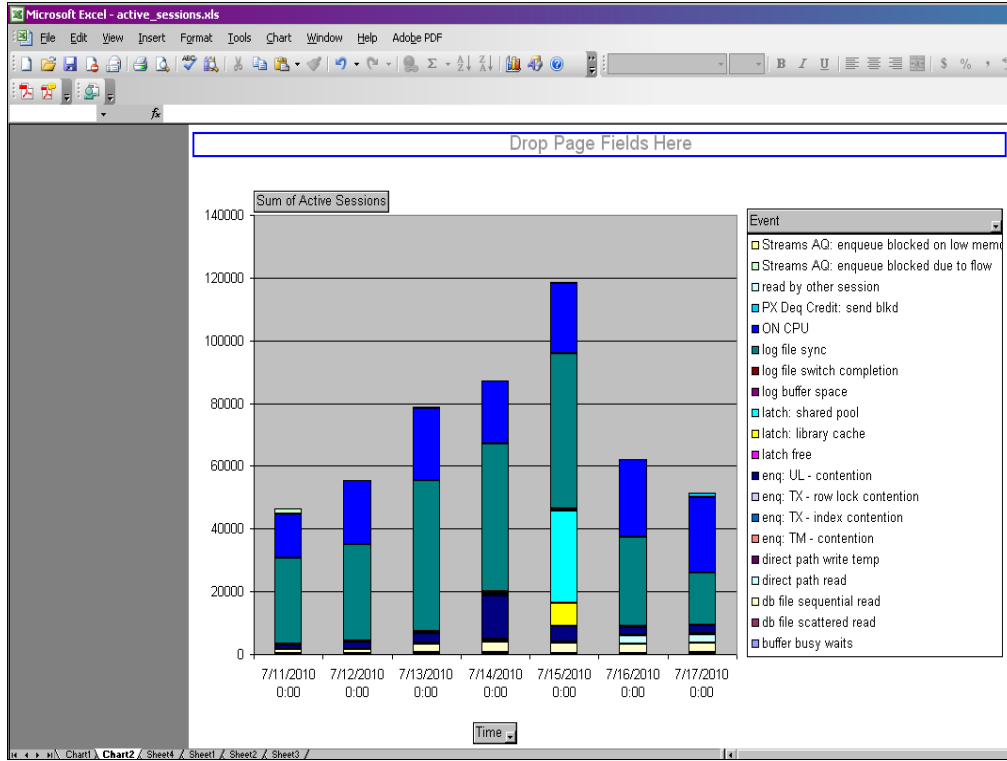
File Edit View Insert Format Tools Data Window Help Adobe PDF

Calibri

Chart Wizard

A4 Time

	A	B	C
1			
2			
3	Sum of Active Sessions	Event	
4	Time	buffer busy waits	db file scattered read
5	7/11/2010 0:00		144
6	7/12/2010 0:00		108
7	7/13/2010 0:00		219
8	7/14/2010 0:00		317
9	7/15/2010 0:00		164
10	7/16/2010 0:00		195
11	7/17/2010 0:00		436
12	Grand Total	2212	1583
13			
14			
15			
16			
17			
18			





## Customizing ASH/AWR Reporting

- Data stored in documented tables
  - Trending, profiling, alerting possible with custom scripting
- Build a corporate dba repository
  - Keep little online in production
  - Long term trending/forecasting
  - Will require custom scripting
- Find the Metrics that Matter



# Questions?

- [daniel.fink@optimaldba.com](mailto:daniel.fink@optimaldba.com)
- [www.optimaldba.com](http://www.optimaldba.com)
  - Sorry...it's really obvious I'm not a web developer!