Top 10 Ways To Totally Mess Up An AWR Analysis

https://www.orapub.com/ppts

Craig Shallahamer
craig@orapub.com

ORACLE ACE Director

ORAPUB

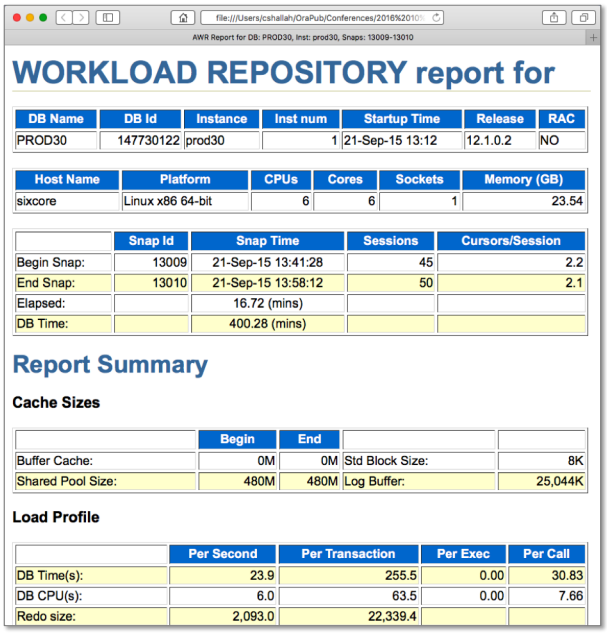
REGISTER NOW AT COLLABORATE.IOUG.ORG

ORAPUB

For the most updated version of this presentation, go to:

www.orapub.com/ppts

ORAPUB



AWR Report for DB: PROD30, Inst: prod30, Snaps: 13009-13010

WORKLOAD REPOSITORY report for

DB Name	DB Id	Instance	Inst num	Startup Time	Release	RAC
PROD30	147730122	prod30	1	21-Sep-15 13:12	12.1.0.2	NO

Host Name	Platform	CPUs	Cores	Sockets	Memory (GB)
sixcore	Linux x86 64-bit	6	6	1	23.54

	Snap Id	Snap Time	Sessions	Cursors/Session
Begin Snap:	13009	21-Sep-15 13:41:28	45	2.2
End Snap:	13010	21-Sep-15 13:58:12	50	2.1
Elapsed:		16.72 (mins)		
DB Time:		400.28 (mins)		

Report Summary

Cache Sizes

	Begin	End		
Buffer Cache:	0M	0M	Std Block Size:	8K
Shared Pool Size:	480M	480M	Log Buffer:	25,044K

Load Profile

	Per Second	Per Transaction	Per Exec	Per Call
DB Time(s):	23.9	255.5	0.00	30.83
DB CPU(s):	6.0	63.5	0.00	7.66
Redo size:	2,093.0	22,339.4		

ORAPUB

ORAPUB

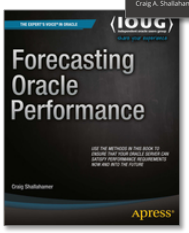
ORAPUB

By doing hundreds of time based analysis, I've noticed there are a few key ways many **DBAs invalidate their AWR analysis.**

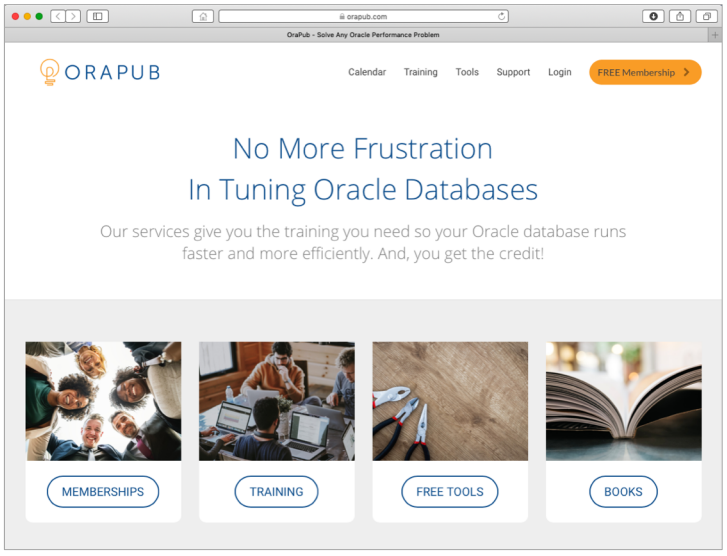
About Me...

ORAPUB

- Long time Oracle DBA
- Specialize in Oracle Database performance and performance engineering
- Performance researcher
- Blogger: A Wider View About Oracle Performance Tuning
- Author: Oracle Performance Firefighting and Forecasting Oracle Performance.
- Conference speaker
- Teacher and mentor
- Oracle ACE Director
- IOUG DBA Track Manager



ORAPUB



OraPub works with Oracle DBAs empowering them to beat bots, AI, machine learning and autonomous anything.

ORAPUB

It's all about time

because that's what users care about

ORAPUB

CPU Consumption Exclusion

1/10

ORAPUB

Who gets the major award?

Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Wait Avg(ms)	% DB time	Wait Class
DB CPU		47.7K		82.4	
cell smart table scan	1,440,243	4575	3.18	7.9	User I/O
cell single block physical read	3,741,750	2692.8	0.72	4.7	User I/O

Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Wait Avg(ms)	% D
DB CPU		47.7K		
cell smart table scan	1,440,243	4575	3.18	
cell single block physical read	3,741,750	2692.8	0.72	
enq: KO - fast object checkpoint	5,746	1045.8	182.01	
log file sync	462,766	385.3	0.83	
reliable message	3,306	346	104.65	
flashback log file sync	1,077	143.5	133.22	
cell multiblock physical read	296,720	139.9	0.47	
direct path write	949	102.1	107.60	
cursor: pin S wait on X	19	62.2	3274.42	

ORAPUB

CPU Consumption Exclusion

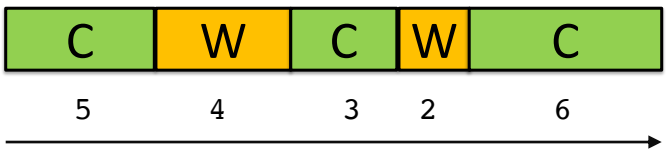
1/10

Elapsed Time Confusion

2/10

Oracle Time. What is it?

Database Time = CPU Time + Non-Idle Wait Time



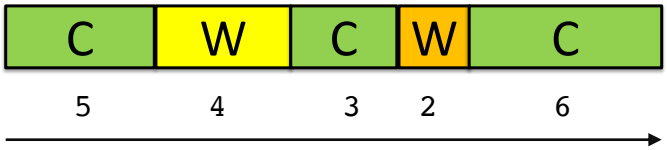
T I M E		W O R K
CPU	= 14	
Non-Idle Wait	= 6	300K LIOs
Elapsed = DB time	= 20	
Idle Wait	= 0	
Experience	= 20	

Single Serial Session

Oracle Time. What is it?

ORAPUB

Database Time = CPU Time + Non-Idle Wait Time



T I M E		W O R K
CPU	= 14	
Non-Idle Wait	= 2	300K LIOs
Elapsed = DB time	= 16	
Idle Wait	= 4	
Experience	= 20	

Single Serial Session

ORAPUB

Elapsed Time May Not Equal Wall Time

$$\text{Wall Time} = \frac{\text{Elapsed Time}}{\text{Parallelism}}$$

ORAPUB

WORKLOAD REPOSITORY report for

DB Name	DB Id	Instance	Inst num	Startup Time	Release	RAC
BOX54	708114003	box54b	1	04-Feb-18 03:02	12.1.0.2.0	YES

Host Name	Platform	CPUs	Cores	Sockets	Memory (GB)
sumertime001.westchck.edu	Linux x86 64-bit	72	36	2	755.34

	Snap Id	Snap Time	Sessions	Cursors/Session	Instances
Begin Snap:	26871	07-Aug-18 08:00:06	2194	7.6	2
End Snap:	26872	07-Aug-18 09:00:18	2251	8.0	2
Elapsed:		60.20 (mins)			
DB Time:		963.42 (mins)			

This "Elapsed Time" is

NOT

Oracle Time-Based "Elapsed Time"

ORAPUB

SQL Statistics

Yes!

- SQL ordered by Elapsed Time
- SQL ordered by CPU Time
- SQL ordered by User I/O Wait Time
- SQL ordered by Gets
- SQL ordered by Reads
- SQL ordered by Physical Reads (UnOptimized)
- SQL ordered by Executions
- SQL ordered by Parse Calls
- SQL ordered by Sharable Memory
- SQL ordered by Version Count
- SQL ordered by Cluster Wait Time
- Complete List of SQL Text

ORAPUB

SQL Statistics

- SQL ordered by Elapsed Time
- SQL ordered by CPU Time
- SQL ordered by User I/O Wait Time
- SQL ordered by Gets
- SQL ordered by Reads
- SQL ordered by Physical Reads (UnOptimized)
- SQL ordered by Executions
- SQL ordered by Parse Calls
- SQL ordered by Sharable Memory
- SQL ordered by Version Count
- SQL ordered by Cluster Wait Time
- Complete List of SQL Text

Back to Top

Yes!

SQL ordered by Elapsed Time

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- % Total DB Time is the Elapsed Time of the SQL statement divided into the Total Database Time multiplied by 100
- %Total - Elapsed Time as a percentage of Total DB time
- %CPU - CPU Time as a percentage of Elapsed Time
- %IO - User I/O Time as a percentage of Elapsed Time
- Captured SQL account for 86.3% of Total DB Time (s): 57,805
- Captured PL/SQL account for 27.9% of Total DB Time (s): 57,805

Elapsed Time (s)	Executions	Elapsed Time per Exec (s)	%Total	%CPU	%IO	SQL Id	SQL Module	SQL Text
13,474.01	17,600	0.77	23.31	96.79	1.21	62r2rym6qb9x2	JDBC Thin Client	begin :1 := GRAIN_SOUP_CUST_FI...
12,385.62	3,201	3.87	21.43	99.71	0.03	dhan4yksznnw4	JDBC Thin Client	SELECT DISTINCT HDOR.HDOR_ID A...
7,983.11	9,479	0.84	13.81	98.74	0.77	462qgd858h389	JDBC Thin Client	SELECT listagg(SOAP.SBSB_CK, ...
4,870.69	482	10.11	8.43	89.72	9.99	gzdmst9p1cf75	JDBC Thin Client	SELECT DISTINCT RSCL.* , TRIM(...
4,800.13	15,082	0.32	8.30	2.89	76.13	gdzc93ujgmctp	JDBC Thin Client	SELECT TRANSACTION_ID, CREATED...
2,929.27	172	17.03	5.07	99.34	0.01	99sw69njw3a24	JDBC Thin Client	SELECT PAYEE_TAX_ID, PAYEE_CHK...
2,137.05	1,375	1.55	3.70	98.00	0.25	6qk55fvzqmckr	JDBC Thin Client	SELECT listagg(SOAP.SBSB_CK, ...
1,642.74	4,269	0.38	2.84	96.48	0.48	2963tngg936vk	JDBC Thin Client	SELECT listagg(SOAP.SBSB_CK, ...
1,534.36	426	3.60	2.65	99.28	0.05	dd5778vkt0dyy	JDBC Thin Client	SELECT DISTINCT RSCL.* , TRIM(...

ORAPUB

SQL ordered by CPU Time

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- %Total - CPU Time as a percentage of Total DB CPU
- %CPU - CPU Time as a percentage of Elapsed Time
- %IO - User I/O Time as a percentage of Elapsed Time
- Captured SQL account for 88.0% of Total CPU Time (s): 47,652
- Captured PL/SQL account for 31.9% of Total CPU Time (s): 47,652

Yes!

CPU Time (s)	Executions	CPU per Exec (s)	%Total	Elapsed Time (s)	%CPU	%IO	SQL Id	SQL Module	SQL Text
13,041.31	17,600	0.74	27.37	13,474.01	96.79	1.21	62r2rym6qb9x2	JDBC Thin Client	begin :1 := GRAIN_SOUP_CUST_FI...
12,349.30	3,201	3.86	25.92	12,385.62	99.71	0.03	dh		...R_ID A...
7,882.44	9,479	0.83	16.54	7,983.11	98.74	0.77	46		..., ...
4,369.79	482	9.07	9.17	4,870.69	89.72	9.99	gz		...IM(...
2,910.07	172	16.92	6.11	2,929.27	99.34	0.01	99		...E_CHK...
2,094.20	1,375	1.52	4.39	2,137.05	98.00	0.25	6q		..., ...
1,584.98	4,269	0.37	3.33	1,642.74	96.48	0.48	29		..., ...

f Elapsed Time
al CPU Time (s): 47,652
Total CPU Time (s): 47,652

%Total	Elapsed Time (s)	%CPU	%IO	
27.37	13,474.01	96.79	1.21	62r2
25.92	12,385.62	99.71	0.03	dhar
16.54	7,983.11	98.74	0.77	462c
9.17	4,870.69	89.72	9.99	gzdr
6.11	2,929.27	99.34	0.01	99sv
4.39	2,137.05	98.00	0.25	6qk5
3.33	1,642.74	96.48	0.48	2963

ORAPUB

Elapsed Time Confusion

2/10

ORAPUB

Focusing on the insignificant

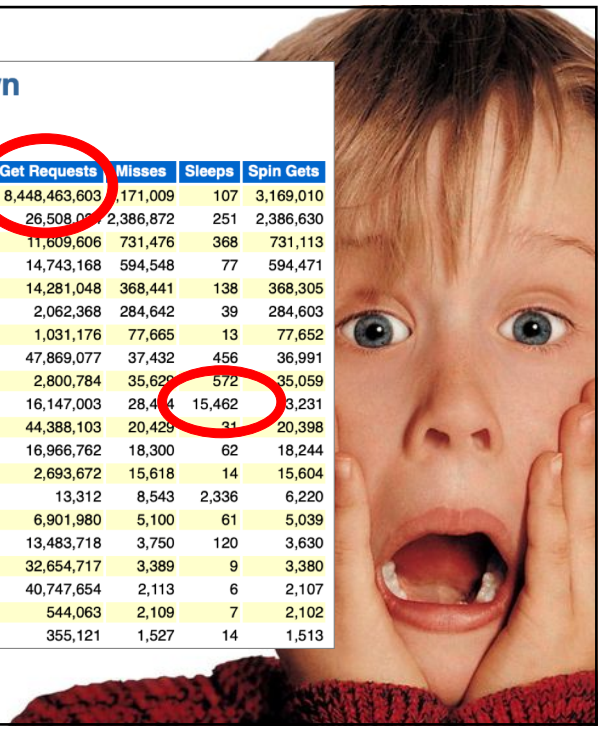
(looking for something interesting)

3/10

Latch Sleep Breakdown

- ordered by misses desc

Latch Name	Get Requests	Misses	Sleeps	Spin Gets
cache buffers chains	8,448,463,603	,171,009	107	3,169,010
messages	26,508,0	2,386,872	251	2,386,630
active checkpoint queue latch	11,609,606	731,476	368	731,113
enqueue hash chains	14,743,168	594,548	77	594,471
redo writing	14,281,048	368,441	138	368,305
LGWR NS Write	2,062,368	284,642	39	284,603
readredo stats and histogram	1,031,176	77,665	13	77,652
ges resource hash list	47,869,077	37,432	456	36,991
redo allocation	2,800,784	35,62	572	35,059
gc element	16,147,003	28,4 4	15,462	3,231
row cache objects	44,388,103	20,42	31	20,398
cache buffers lru chain	16,966,762	18,300	62	18,244
flashback sync request	2,693,672	15,618	14	15,604
space background task latch	13,312	8,543	2,336	6,220
ges cached resource lists	6,901,980	5,100	61	5,039
shared pool	13,483,718	3,750	120	3,630
ges enqueue table freelist	32,654,717	3,389	9	3,380
ASM map operation hash table	40,747,654	2,113	6	2,107
lgwr LWN SCN	544,063	2,109	7	2,102
KJC message pool free list	355,121	1,527	14	1,513



Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Wait Avg(ms)	% DB time	Wait Class
DB CPU		1988.6		41.8	
direct path read temp	315,169	1583.9	5.03	33.3	User I/O
db file sequential read	87,178	944.5	10.83	19.9	User I/O
direct path write temp	10,568	92	8.71	1.9	User I/O
db file scattered read	6,792	58.8	8.66	1.2	User I/O
log file switch (checkpoint incomplete)	64	31.7	494.96	.7	Configuratio
read by other session	3,872	17.1	4.41	.4	User I/O
direct path read	986	14	14.23	.3	User I/O
log file switch completion	32	3.2	98.91	.1	Configuratio
db file parallel read	39	.3	7.42	.0	User I/O

ORAPUB

Focusing on the insignificant

(looking for something interesting)

3/10

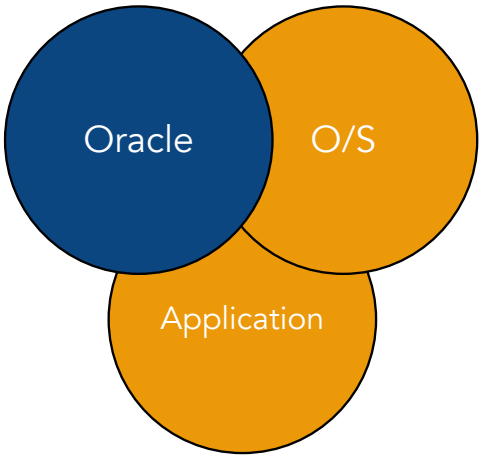
ORAPUB

I'm A DBA NOT An OS Admin

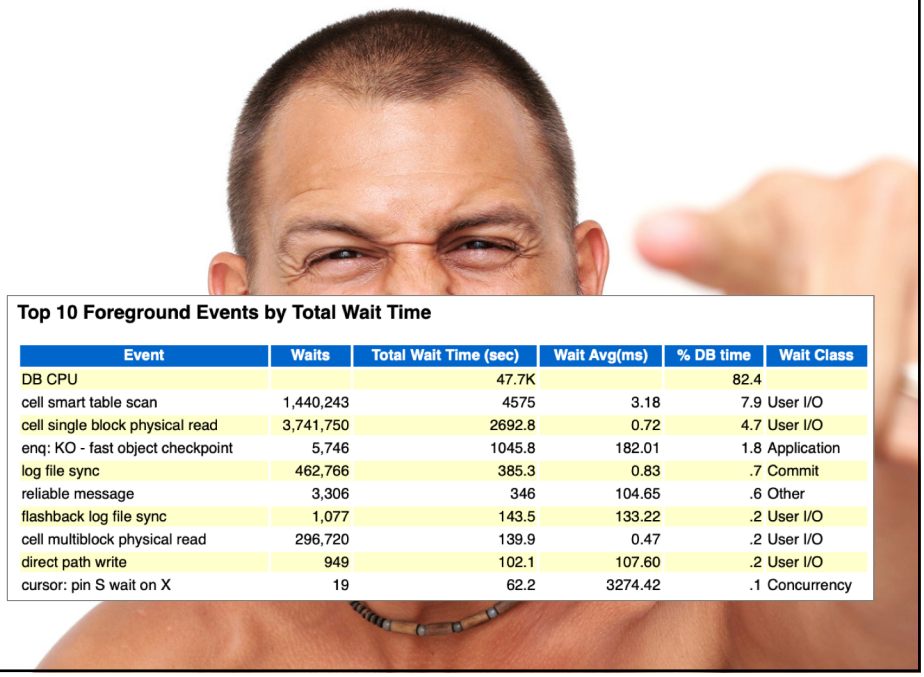
4/10

ORAPUB

It's more than just Oracle.



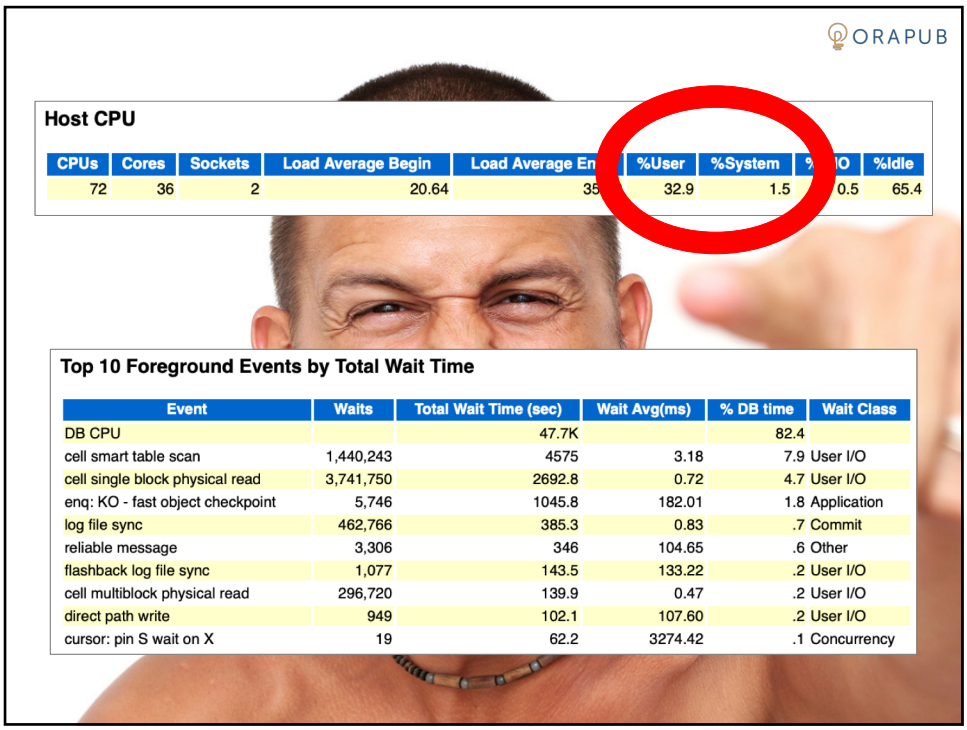
ORAPUB



Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Wait Avg(ms)	% DB time	Wait Class
DB CPU		47.7K		82.4	
cell smart table scan	1,440,243	4575	3.18	7.9	User I/O
cell single block physical read	3,741,750	2692.8	0.72	4.7	User I/O
enq: KO - fast object checkpoint	5,746	1045.8	182.01	1.8	Application
log file sync	462,766	385.3	0.83	.7	Commit
reliable message	3,306	346	104.65	.6	Other
flashback log file sync	1,077	143.5	133.22	.2	User I/O
cell multiblock physical read	296,720	139.9	0.47	.2	User I/O
direct path write	949	102.1	107.60	.2	User I/O
cursor: pin S wait on X	19	62.2	3274.42	.1	Concurrency

ORAPUB



Host CPU

CPUs	Cores	Sockets	Load Average Begin	Load Average En	%User	%System	%	IO	%Idle
72	36	2	20.64	35	32.9	1.5		0.5	65.4

Top 10 Foreground Events by Total Wait Time

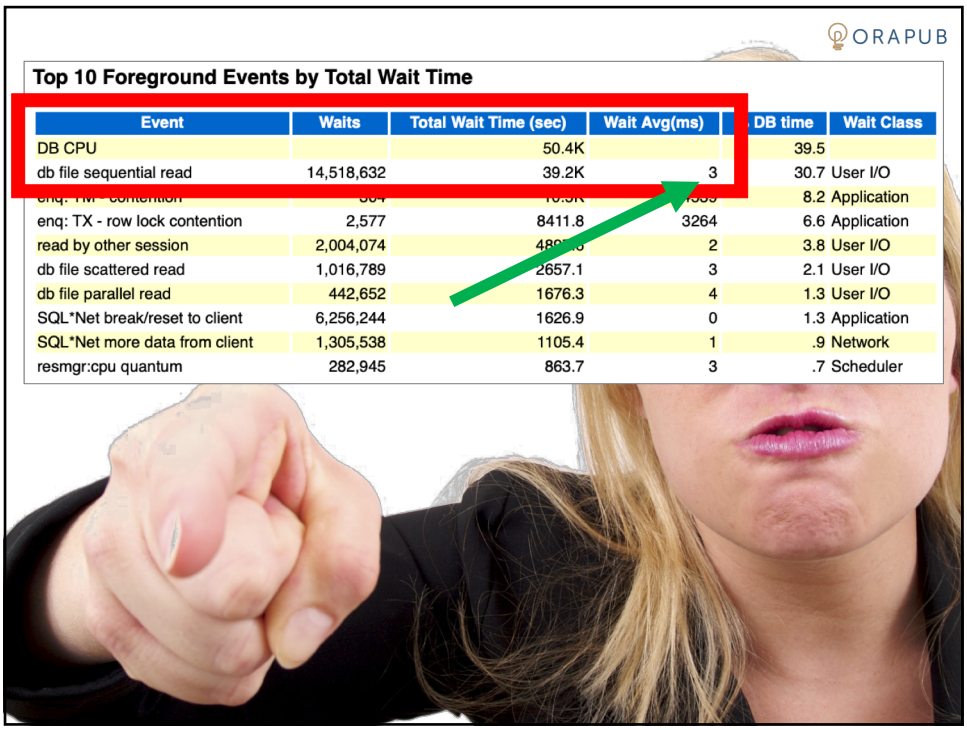
Event	Waits	Total Wait Time (sec)	Wait Avg(ms)	% DB time	Wait Class
DB CPU		47.7K		82.4	
cell smart table scan	1,440,243	4575	3.18	7.9	User I/O
cell single block physical read	3,741,750	2692.8	0.72	4.7	User I/O
enq: KO - fast object checkpoint	5,746	1045.8	182.01	1.8	Application
log file sync	462,766	385.3	0.83	.7	Commit
reliable message	3,306	346	104.65	.6	Other
flashback log file sync	1,077	143.5	133.22	.2	User I/O
cell multiblock physical read	296,720	139.9	0.47	.2	User I/O
direct path write	949	102.1	107.60	.2	User I/O
cursor: pin S wait on X	19	62.2	3274.42	.1	Concurrency

ORAPUB

Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Wa
DB CPU		50.4K	
db file sequential read	14,518,632	39.2K	
enq: TX - row lock contention	2,577	8411.8	
read by other session	2,004,074	4897.6	
db file scattered read	1,016,789	2657.1	
db file parallel read	442,652	1676.3	
SQL*Net break/reset to client	6,256,244	1626.	
SQL*Net more data from client	1,305,538	1105	
resmgr:cpu quantum	282,945	863	

ORAPUB



ORAPUB

I'm A DBA
NOT
An OS Admin

4/10

ORAPUB

When an AWR report analysis is the wrong analysis strategy

5/10

ORAPUB

Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Wait Avg(ms)	% DB time	Wait Class
DB CPU		18.3K		25.0	
db file sequential read	595,835	18K	30	24.7	User I/O
log file sync	26,296	13.9K	530	19.1	Commit
SQL*Net message from dblink	3,230	7780.4	2409	10.7	Network
TCP Socket (KGAS)	122,399	7631.2	62	10.5	Network
Disk file operations I/O	76,981	3641.5	47	5.0	User I/O
direct path read	167,212	2316.2	14	3.2	User I/O
read by other session	26,883	1112.4	41	1.5	User I/O
buffer busy waits	3,499	711	203	1.0	Concurrency
direct path write temp	4,389	593.1	135	.8	User I/O

By design,
an AWR report focuses on
an Oracle instance or cluster.


ORAPUB
AWR report says CPU and physical read IO is the heart of the issue.
Users say performance is slow every Friday afternoon.


ORAPUB
AWR report says CPU and physical read IO is the heart of the issue.
A few users said their screen "just froze" this morning at 10:30.

ORAPUB

Strategy? AWR or ASH or either

Characteristic	AWR	ASH
Repeatable?	Yes - Expected	No - Random
Problem Scope	Instance or RAC	Anything
Number of users	Must drive the "current"	Can be a single session, many sessions or all sessions
Need session details	No – Not possible	Yes – Session level samples
Need Wait Times	Yes – Average and more	No – Except 20+ sec

ORAPUB

When an AWR report analysis is the wrong analysis strategy

5/10

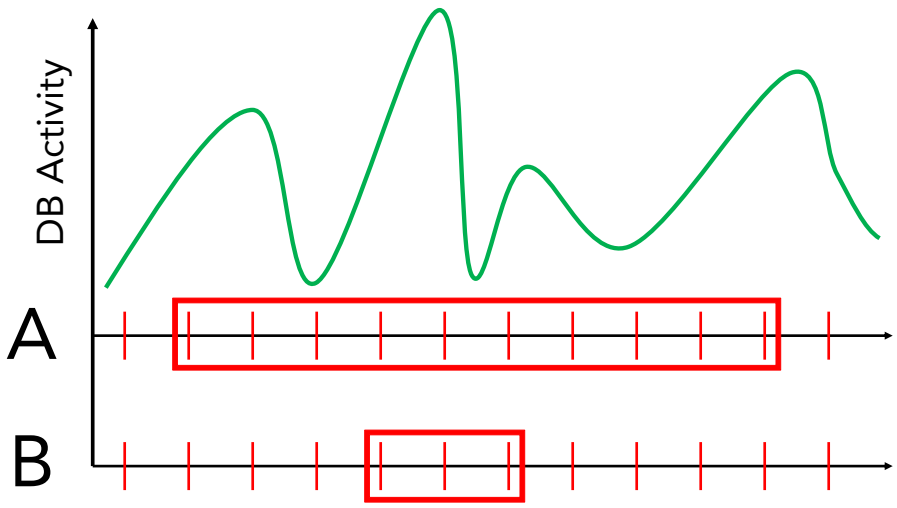
ORAPUB

Too BIG or SMALL snapshot interval

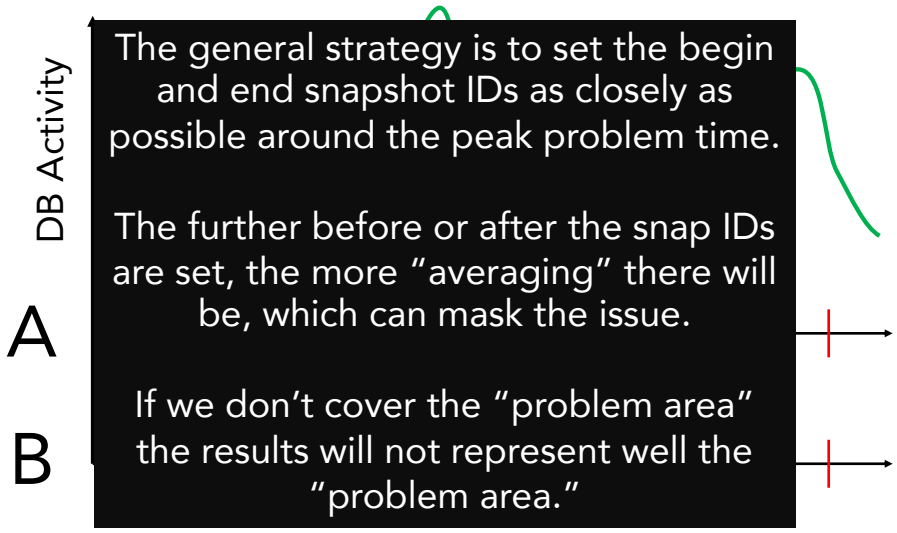
6/10

ORAPUB

Which snapshot interval?



Which snapshot interval?

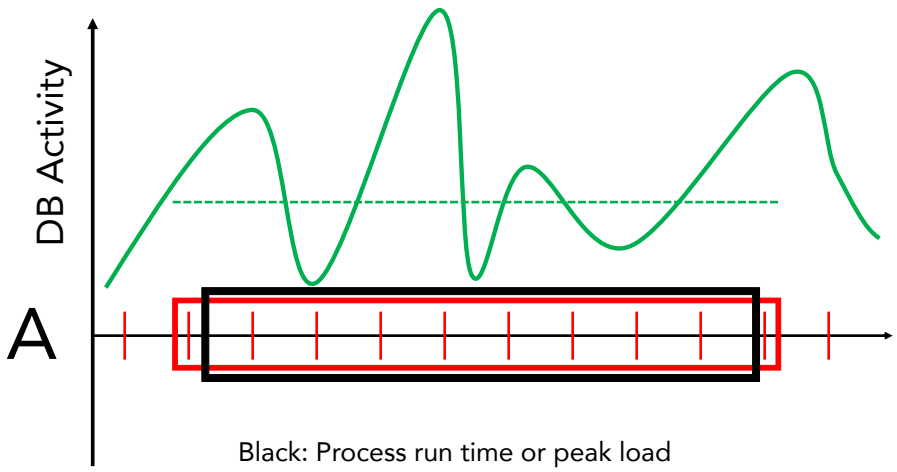


The general strategy is to set the begin and end snapshot IDs as closely as possible around the peak problem time.

The further before or after the snap IDs are set, the more "averaging" there will be, which can mask the issue.

If we don't cover the "problem area" the results will not represent well the "problem area."

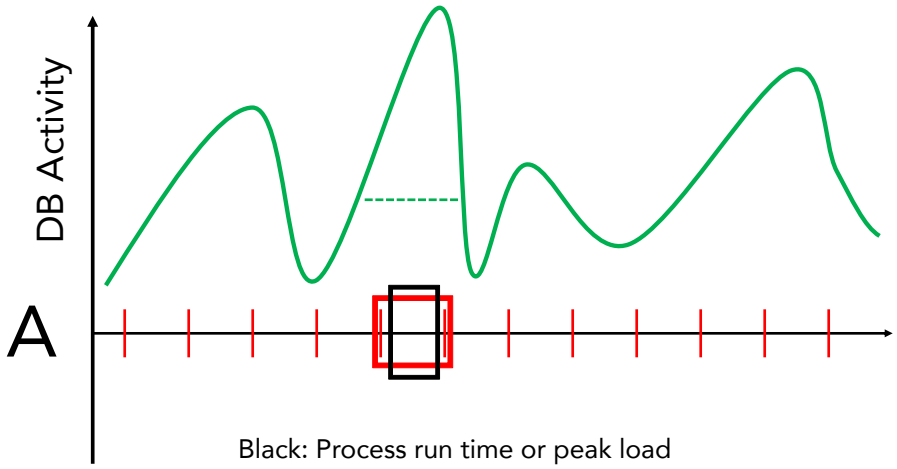
Good or bad?



Black: Process run time or peak load
Red : Snapshot begin and end times

ORAPUB

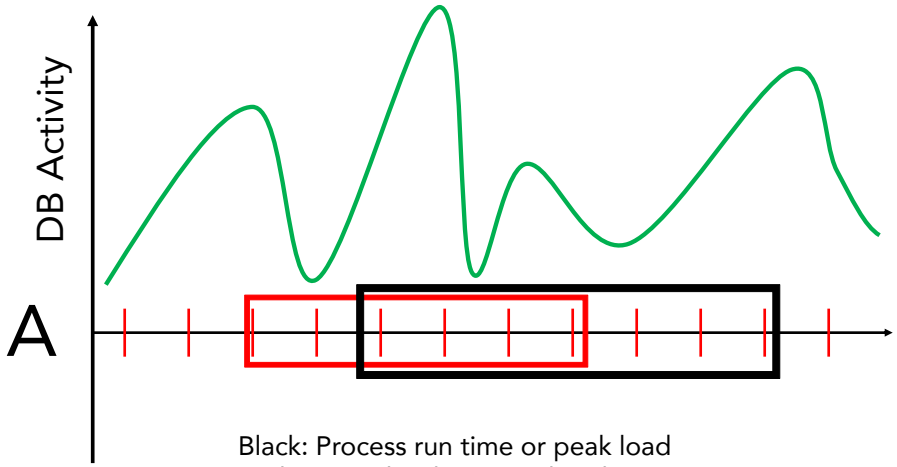
Good or bad?



Black: Process run time or peak load
Red : Snapshot begin and end times

ORAPUB

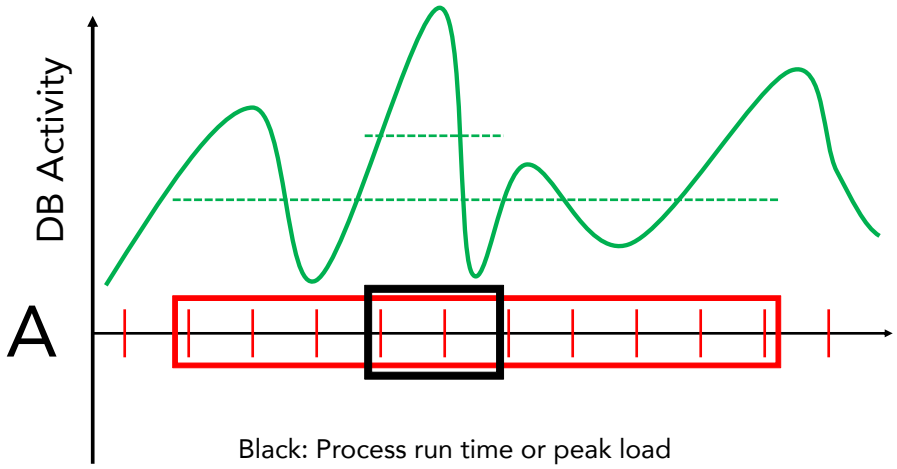
Good or bad?



Black: Process run time or peak load
Red : Snapshot begin and end times

ORAPUB

Good or bad?



Black: Process run time or peak load
Red : Snapshot begin and end times

ORAPUB

Too

BIG or SMALL

snapshot interval

6/10

ORAPUB

Not Defining Top SQL

7/10

ORAPUB

Just look at my "TOP SQL" monitor!

ORAPUB

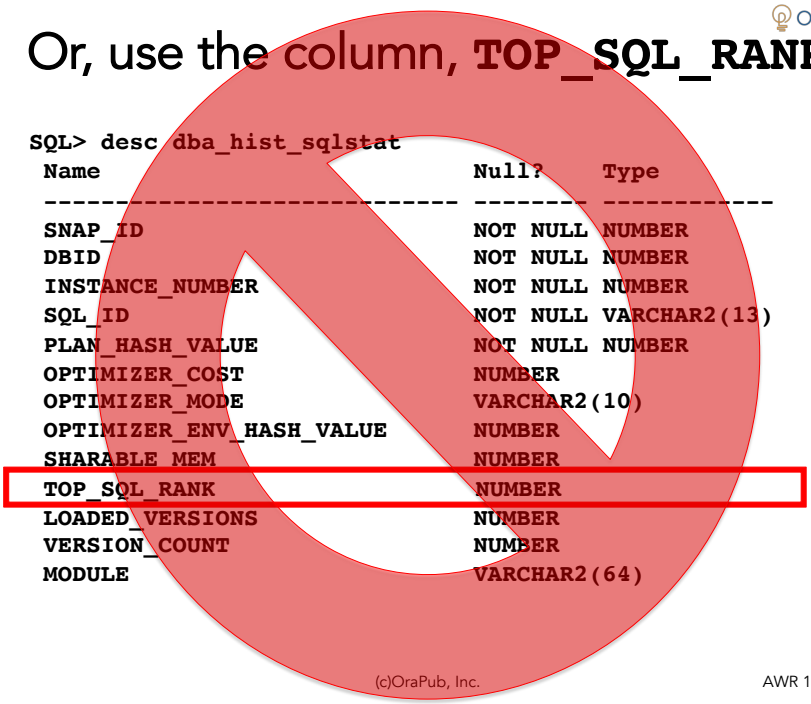
Or, use the column, **TOP_SQL_RANK**

```
SQL> desc dba_hist_sqlstat
 Name                          Null?    Type
 ----------------------------- -------- ------------
 SNAP_ID                       NOT NULL NUMBER
 DBID                          NOT NULL NUMBER
 INSTANCE_NUMBER               NOT NULL NUMBER
 SQL_ID                        NOT NULL VARCHAR2(13)
 PLAN_HASH_VALUE               NOT NULL NUMBER
 OPTIMIZER_COST                NUMBER
 OPTIMIZER_MODE                VARCHAR2(10)
 OPTIMIZER_ENV_HASH_VALUE      NUMBER
 SHARABLE_MEM                  NUMBER
 TOP_SQL_RANK                  NUMBER
 LOADED_VERSIONS               NUMBER
 VERSION_COUNT                 NUMBER
 MODULE                        VARCHAR2(64)
```


ORAPUB

Or, use the column, **TOP_SQL_RANK**

SQL> desc dba_hist_sqlstat
 Name Null? Type
 ----------------------------- -------- ------------
 SNAP_ID NOT NULL NUMBER
 DBID NOT NULL NUMBER
 INSTANCE_NUMBER NOT NULL NUMBER
 SQL_ID NOT NULL VARCHAR2(13)
 PLAN_HASH_VALUE NOT NULL NUMBER
 OPTIMIZER_COST NUMBER
 OPTIMIZER_MODE VARCHAR2(10)
 OPTIMIZER_ENV_HASH_VALUE NUMBER
 SHARABLE_MEM NUMBER
 TOP_SQL_RANK NUMBER
 LOADED_VERSIONS NUMBER
 VERSION_COUNT NUMBER
 MODULE VARCHAR2(64)



ORAPUB

There is no AWR `TOP_SQL_RANK` column.

So, what does "top SQL" mean anyway?

ORAPUB

Let your time-based analysis, the user experience and the OS situation drive and define the "top" SQL.

Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Wait Avg(ms)	% DB time	Wait Class
DB CPU		47.7K		82.4	
cell smart table scan	1,440,243	4575	3.18	7.9	User I/O
cell single block physical read	3,741,750	2692.8	0.72	4.7	User I/O
enq: KO - fast object checkpoint	5,746	1045.8	182.01	1.8	Application
log file sync	462,766	385.3	0.83	.7	Commit
reliable message	3,306	346	104.65	.6	Other
flashback log file sync	1,077	143.5	133.22	.2	User I/O
cell multiblock physical read	296,720	139.9	0.47	.2	User I/O
direct path write	949	102.1	107.60	.2	User I/O
cursor: pin S wait on X	19	62.2	3274.42	.1	Concurrency

How should we define the "top SQL"?

ORAPUB

DB Time is dominated by CPU consumption. So, CPU can be one way to define the "top SQL."

SQL ordered by CPU Time

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- %Total - CPU Time as a percentage of Total DB CPU
- %CPU - CPU Time as a percentage of Elapsed Time
- %IO - User I/O Time as a percentage of Elapsed Time
- Captured SQL account for 88.0% of Total CPU Time (s): 47,652
- Captured PL/SQL account for 31.9% of Total CPU Time (s): 47,652

CPU Time (s)	Executions	CPU per Exec (s)	%Total	Elapsed Time (s)	%CPU	%IO	SQL Id	SQL Module	SQL Text
13,041.31	17,600	0.74	27.37	13,474.01	96.79	1.21	62r2rym6qb9x2	JDBC Thin Client	begin :1 := GRAIN_SOUP_CU
12,349.30	3,201	3.86	25.92	12,385.62	99.71	0.03	dhan4yksznnw4	JDBC Thin Client	SELECT DISTINCT HDOR.HI
7,882.44	9,479	0.83	16.54	7,983.11	98.74	0.77	462qgd858h389	JDBC Thin Client	SELECT listagg(SOAP.SBSB
4,369.79	482	9.07	9.17	4,870.69	89.72	9.99	gzdmst9p1cf75	JDBC Thin Client	SELECT DISTINCT RSCL.* ,
2,910.07	172	16.92	6.11	2,929.27	99.34	0.01	99sw69njw3a24	JDBC Thin Client	SELECT PAYEE_TAX_ID, PA

ORAPUB

Could the top CPU SQL also be the top IO read and/or top Elapsed Time SQL?

SQL ord
- Resource
- %Total - C
- %CPU - C
- %IO - Use
- Captured
- Captured

CPU Time (s)	E								SQL Text
13,041.31									gin :1 := GRAIN_SOUP_CU
12,349.30	3,201	3.86	25.92	12,385.62	99.71	0.03	dhan4yksznnw4	JDBC Thin Client	SELECT DISTINCT HDOR.HI
7,882.44	9,479	0.83	16.54	7,983.11	98.74	0.77	462qgd858h389	JDBC Thin Client	SELECT listagg(SOAP.SBSB
4,369.79	482	9.07	9.17	4,870.69	89.72	9.99	gzdmst9p1cf75	JDBC Thin Client	SELECT DISTINCT RSCL.* ,
2,910.07	172	16.92	6.11	2,929.27	99.34	0.01	99sw69njw3a24	JDBC Thin Client	SELECT PAYEE_TAX_ID, PA

ORAPUB

The top CPU SQL could also be the top elapsed time SQL and the top buffer get SQL.

SQL ordered by Elapsed Time

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- % Total DB Time is the Elapsed Time of the SQL statement divided into the Total Database Time multiplied by 100
- %Total - Elapsed Time as a percentage of Total DB time
- %CPU - CPU Time as a percentage of Elapsed Time
- %IO - User I/O Time as a percentage of Elapsed Time
- Captured SQL account for 86.3% of Total DB Time (s): 57,805
- Captured PL/SQL account for 27.9% of Total DB Time (s): 57,805

Elapsed Time (s)	Executions	Elapsed Time per Exec (s)	%Total	%CPU	%IO	SQL Id	SQL Module	SQL Text
13,474.01	17,600	0.77	23.31	96.79	1.21	62r2rym6qb9x2	JDBC Thin Client	begin :1 := GRAIN_SOUP_C
12,385.62	3,201	3.87	21.43	99.71	0.03	dhan4yksznnw4	JDBC Thin Client	SELECT DISTINCT HDOR.H
7,983.11	9,479	0.84	13.81	98.74	0.77	462qgd858h389	JDBC Thin Client	SELECT listagg(SOAP.SBSB
4,870.69	482	10.11	8.43	89.72	9.99	gzdmst9p1cf75	JDBC Thin Client	SELECT DISTINCT RSCL.* ,
4,800.13	15,082	0.32	8.30	2.89	76.13	gdzc93ujgmctp	JDBC Thin Client	SELECT TRANSACTION_ID,
2,929.27	172	17.03	5.07	99.34	0.01	99sw69njw3a24	JDBC Thin Client	SELECT PAYEE_TAX_ID, PA

ORAPUB

ORAPUB

Not Defining Top SQL

7/10

ORAPUB

Not Using Parallelism To Your Advantage

8/10

Q: Which full bucket will empty first?

Q: Which query will complete first?

ORAPUB

Make sure your IO subsystem can absorb the increased IO rate, but for a shorter period of time.

Yes PQ

ORAPUB

Which SQL will likely benefit from increased parallelism?

SQL ordered by Elapsed Time

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- % Total DB Time is the Elapsed Time of the SQL statement divided into the Total Database Time multiplied by 100
- %Total - Elapsed Time as a percentage of Total DB time
- %CPU - CPU Time as a percentage of Elapsed Time
- %IO - User I/O Time as a percentage of Elapsed Time
- Captured SQL account for 86.3% of Total DB Time (s): 57,805
- Captured PL/SQL account for 27.9% of Total DB Time (s): 57,805

Elapsed Time (s)	Executions	Elapsed Time per Exec (s)	%Total	%CPU	%IO	SQL Id	SQL Module	SQL Text
13,474.01	17,600	0.77	23.31	96.79	1.21	62r2rym6qb9x2	JDBC Thin Client	begin :1 := GRAIN_SOUP_C
12,385.62	3,201	3.87	21.43	99.71	0.03	dhan4yksznnw4	JDBC Thin Client	SELECT DISTINCT HDOR.H
7,983.11	9,479	0.84	13.81	98.74	0.77	462qgd858h389	JDBC Thin Client	SELECT listagg(SOAP.SBSB
4,870.69	482	10.11	8.43	89.72	9.99	gzdmst9p1cf75	JDBC Thin Client	SELECT DISTINCT RSCL.* ,
4,800.13	15,082	0.32	8.30	2.89	76.13	gdzc93ujgmctp	JDBC Thin Client	SELECT TRANSACTION_ID,
2,929.27	172	17.03	5.07	99.34	0.01	99sw69njw3a24	JDBC Thin Client	SELECT PAYEE_TAX_ID, PA

ORAPUB

Not Using Parallelism To Your Advantage

8/10

ORAPUB

Avoiding The User Community

9/10

ORAPUB

Yes, users can be scary.
But they also provide lots of information.
Some helpful. Some not helpful.

ORAPUB

Yes, users can be scary.
But they also provide lots of information.
Some helpful. Some not helpful.

ORAPUB

Avoiding The User Community

9/10

ORAPUB

Blindly Trust Oracle Time Data

10/10

Oracle time problem

ORAPUB

48.4K

Time Model Statistics

- DB Time represents total time in user calls
- DB CPU represents CPU time of foreground processes
- Total CPU Time represents foreground and background processes
- Statistics including the word "background" measure background
- Ordered by % of DB time in descending order, followed by

Statistic Name	Time (s)	% of DB Time
sql execute elapsed time	82,924.74	96
DB CPU	48,443.49	56
parse time elapsed	15,672.18	18
hard parse elapsed time	15,534.96	18
hard parse (sharing criteria) elapsed time	255.99	0
PL/SQL execution elapsed time	45.66	0
connection management call elapsed time	35.38	0
sequence load elapsed time	24.94	0.03
PL/SQL compilation elapsed time	3.33	0.00
failed parse elapsed time	1.48	0.00
repeated bind elapsed time	0.75	0.00
hard parse (bind mismatch) elapsed time	0.31	0.00
DB time	86,270.46	
background elapsed time	431.18	
background cpu time	121.45	
total CPU time	48,564.94	

86.3K

Foreground Wait Class

14.2K

- s - second, ms - millisecond - 1000th of a second
- ordered by wait time desc, waits desc
- %Timeouts: value of 0 indicates value was < .5%. Value of null is reported as 0
- Captured Time accounts for 72.5% of Total DB time 86,270.46 (
- Total FG Wait Time: 14,131.32 (s) DB CPU time: 48,443.49 (s)

Wait Class	Waits	%Time -outs	Total Wait Time (s)	Avg wait (ms)	%DB time
DB CPU			48,443		56.15
User I/O	5,906,084	0	9,260	1.57	10.73
Application	14,203	0	3,739	263.26	4.33
Other	47,314	0	791	16.71	0.92
Concurrency	51,365	0	183	3.57	0.21
Commit	80,982	0	94	1.16	0.11
System I/O	72,921	0	57	0.79	0.07
Network	7,225,651	0	6	0.00	0.01
Configuration	31	6	1	18.62	0.00

```
DB CPU              =  48.4K
Non-Idle Wait       =  14.2K
-------------------------
Derived DB time     =  62.6K
Reported DB time    =  86.3K
```

?

Oracle time problem

ORAPUB

5.3K

Time Model Statistics

- DB Time represents total time in user calls
- DB CPU represents CPU time of foreground processes
- Total CPU Time represents foreground and background process
- Statistics including the word "background" measure background statistic
- Ordered by % of DB time in descending order, followed by Statis

Statistic Name	Time (s)	% of DB Time
sql execute elapsed time	11,621.32	64.62
DB CPU	5,258.74	29.24
parse time elapsed	187.56	1.04
hard parse elapsed time	159.21	0.89
PL/SQL execution elapsed time	89.30	0.50
PL/SQL compilation elapsed time	13.95	0.08
connection management call elapsed time	1.20	0.01
repeated bind elapsed time	0.83	0.00
sequence load elapsed time	0.04	0.00
hard parse (sharing criteria) elapsed time	0.02	
failed parse elapsed time	0.00	
DB time	17,985.44	
background elapsed time	842.50	
background cpu time	333.28	
total CPU time	5,592.02	

18.0K

Wait Classes by Total Wait Time

Wait Class	Waits	Total Wait Time (sec)	A
~~DB CPU~~		~~5,259~~	
Network	69,188,140	4,351	
User I/O	1,910,950	2,518	
System I/O	304,232	574	
Concurrency	384,021	203	
Other	1,023,467	73	
Cluster	422,409	36	
Application	4,346	3	
Commit	3,539	2	
Configuration	486	2	
Scheduler	26	0	

7.8K

```
DB CPU              =  5.3K
Non-Idle Wait       =  7.8K
-------------------------
Derived DB time     = 13.1K
Reported DB time    = 18.0K
```

?

ORAPUB

Do the math. If it's doesn't match…

- Use ASH. Because ASH is sample based and not time based, it avoids many of the time related problems.
- Use Operating System statistics (dba_hist_osstat) to validate DB CPU.
- When the OS is bottlenecked, strange times can be recorded. Just querying and recording the AWR data, can take awhile, impacting the data quality.

ORAPUB

Blindly Trust Oracle Time Data

10/10

ORAPUB

Download ALL my presentations:
www.orapub.com/ppts

.

www.orapub.com/free-webinars

There are FREE
AWR webinars.

OraPub LVCs Are Different

You learn to master the topic at a deeper, higher confidence and more practical level compared to any other teaching method I offer.

This occurs because the class is spread out over multiple weeks, each session is 2 hours with a day in between, you do homework/activation on your real systems and I personally work with you through the entire class.

Details & Registration: https://www.orapub.com/lvc
Event Calendar: https://www.orapub.com/events

Thank You!

Top 10 Ways To Totally Mess Up An AWR Analysis

https://www.orapub.com/ppts

Craig Shallahamer
craig@orapub.com

ORACLE ACE Director

ORAPUB

REGISTER NOW AT COLLABORATE.IOUG.ORG

ORAPUB

For the most updated version of this presentation, go to:

www.orapub.com/ppts