






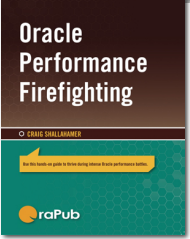
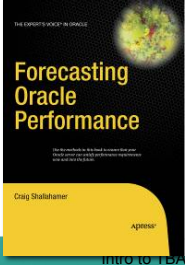
# Introduction to Time-Based Analysis: Stop the Guessing!




  
 Craig A. Shallahamer  
 OraPub, Inc.  
[craig@orapub.com](mailto:craig@orapub.com)  
 

This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.

## Who Am I?

- Studied economics, mathematics and computer science at Cal Polytechnic State University San Luis Obispo, California, USA.
- Started working with Oracle technology in 1989 as a Forms 2.3 developer on Oracle version 5.
- Soon after started performance firefighting daily.
- Co-founded both Oracle's Core Technology and System Performance Groups.
- Left Oracle to start OraPub, Inc. in 1998.
- Authored 24+ technical papers and worked in 31 countries.
- Author two books: Oracle Performance Firefighting and Forecasting Oracle Performance.
- Teaches performance analysis around the world.
- Oracle ACE Director: 
- Blogs performance research: A Wider View



(c)2013 OraPub, Inc. 



**raPub**  
focusing exclusively on Oracle systems performance analysis



Automated. Interactive.  
Oracle SE Performance Analysis

[www.storifree.com](http://www.storifree.com)  
[www.stori.orapub.com](http://www.stori.orapub.com)


### Resources

- Research Blog
- Free Tools
- Free Presentations
- Free Papers
- Books
- Consulting
- Training




(c)2013 OraPub, Inc. Intro to TBA

This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.



# Agenda

- Always remember these four things.
- Get the diagnostic data
- Building the framework
- Example

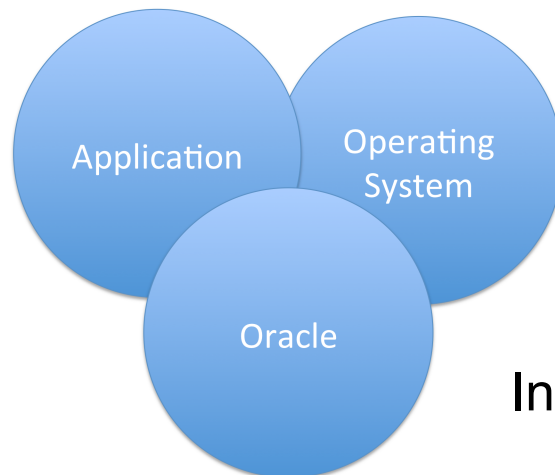


(c)2013 OraPub, Inc. Intro to TBA


Always remember  
these four things.

This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.

## Holistic Performance Analysis



Insurance



## WORKLOAD REPOSITORY report for

Top 5 Timed Events


Event	Waits	Time(s)	Avg Wait(ms)	% Total Call Time	Wait Class
CPU time		3,641		66.3	
db file sequential read	489,550	587	1	10.7	User I/O
db file scattered read	12,142	565	47	10.3	User I/O
direct path read temp	34,932	470	13	8.6	User I/O
log file parallel write	6,253	235	38	4.3	System I/O

### Main Report

- [Report Summary](#)
- [Wait Events Statistics](#)
- [SQL Statistics](#)
- [Instance Activity Statistics](#)
- [IO Stats](#)
- [Buffer Pool Statistics](#)
- [Advisory Statistics](#)
- [Wait Statistics](#)
- [Undo Statistics](#)
- [Latch Statistics](#)
- [Segment Statistics](#)
- [Dictionary Cache Statistics](#)
- [Library Cache Statistics](#)
- [Memory Statistics](#)
- [Streams Statistics](#)
- [Resource Limit Statistics](#)
- [init.ora Parameters](#)


# Quantitative

trustworthy, repeatable,  
demonstrable




(c)2013 OraPub, Inc. Intro to TBA

This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.



Suppose I need to process  
100 pieces of work. Each  
piece takes 1 second to  
process. How long will take  
take to process all 100 pieces?



(c)2013 OraPub, Inc. Intro to TBA

# Serialization is death.

(understanding elapsed time)

$$E = \frac{\text{Pieces of work} \times \text{Time to process a piece of work}}{\text{Parallelization}}$$

This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.



## Analysis Results

This Stori analysis is based upon the CIMARP database Statspack data from 04-Dec-2012 03:00:04 to 04-Dec-2012 04:00:03. Details: host udbou20, dbid 1523308526, instance 1, snap 14515 to 14525, Stori version 0.30-0.12-0.13.

### Performance Analysis Summary

Overall, processes are requesting more CPU resources than the server can provide. The CPU subsystem is certainly the bottleneck. Solutions will focus on either reducing CPU requirements or increasing CPU capacity.

The 8 core CPU subsystem is 88% utilized and this Oracle instance is consuming 84% of the database server CPU capacity. Once utilization gets into the 80% area, performance is noticeably affected.

While all the recommendations presented are technically valid, they need to be analyzed in conjunction with user priorities and comments, budgetary constraints, IT administration complexity, and application uptime requirements. Only then will the best overall initial solution be discovered.



### Recommendations

- For SQL ID 7aw349n5chnzz focus on reducing elapsed time and CPU time. (rank: .644)
- Decrease CPU requirements on the server. For example, look for non-essential...

# Tell a story.

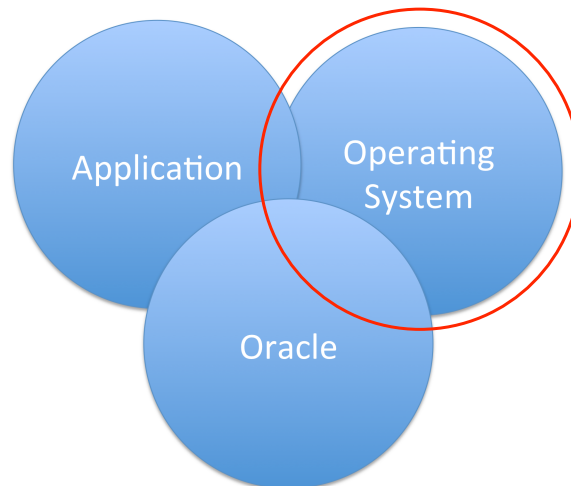
understandable

# Get the diagnostic data

## Build the framework.

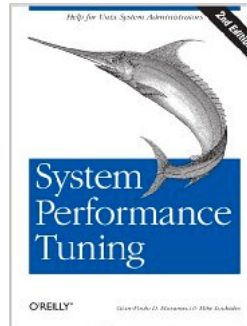
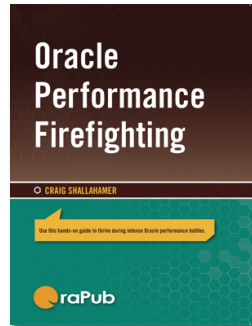
This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.

## Holistic Performance Analysis



Limiting data  
source to  
Oracle  
views.

This book changed my life!



*System Performance Tuning*

Gian-Paolo D. Musmeci  
Mike Loukides

This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.

# O/S – CPU utilization

## Operating System Statistics

Statistic	Total
AVG_BUSY_TIME	116,897
AVG_IDLE_TIME	242,244
AVG_IOWAIT_TIME	53,542
AVG_SYS_TIME	20,811
AVG_USER_TIME	95,754
BUSY_TIME	469,494
IDLE_TIME	970,850
IOWAIT_TIME	215,489
SYS_TIME	84,604
USER_TIME	384,890
LOAD	0
OS_CPU_WAIT_TIME	537,600
RSRC_MGR_CPU_WAIT_TIME	0
PHYSICAL_MEMORY_BYTES	10,737,418,240
NUM_CPUS	4
NUM_CPU_CORES	4

$$U = \frac{R}{C} = \frac{busy\_time}{busy\_time + idle\_time}$$

$$U = 469 / (469 + 970) = 0.33$$

ROT: Once utilization gets into the 80%'s, performance is noticeably impacted.

base view: v\$osstat

# OS – IO R/W Response Time

## Wait Events

- s - second
- cs - centisecond - 100th of a second
- ms - millisecond - 1000th of a second
- us - microsecond - 1000000th of a second
- ordered by wait time desc, waits desc (idle events last)

ROT: Single block synchronous reads should be than 10ms. Write requests, less than 5ms.

Event	Waits	%Time - outs	Total Wait Time (s)	Avg wait (ms)	Waits /txn
db file sequential read	489,550	0.00	587	1	252.35
db file scattered read	12,142	0.00	565	47	6.26
direct path read temp	34,932	0.00	470	13	18.01
log file parallel write	6,253	0.00	235	38	3.22
db file parallel write	4,029	0.00	176	44	2.08
direct path write temp	30,705	0.00	160	5	15.83
log buffer space	416	0.00	46	111	0.21
log file sync	1,938	0.00	38	20	1.00
db file parallel read	1,822	0.00	23	13	0.94
log file switch completion	133	0.00	14	108	0.07

base view: v\$sqlsystem\_event



This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.

# OS – Network Time

## Wait Events

- s - second
- cs - centisecond - 100th of a second
- ms - millisecond - 1000th of a second
- us - microsecond - 1000000th of a second
- ordered by wait time desc, waits desc (idle events last)

Event	Waits	%Time - outs	Total Wait Time (s)	Avg wait (ms)	Waits /txn
db file sequential read	489,550	0.00	587	1	252.35
SQL*Net more data to client	94,595	0.00	5	0	48.76
SQL*Net more data from client	63	0.00	0	5	0.03

ROT: If a "more data" event contains significant time, it is a clue calls from the client process are taking too long. Could be lots of data to transfer, network latency issues, etc. Also do "tnsping", look at network throughput, use some basic OS tools like netstat.

base view: v\$sqlsystem\_event





# OS – Memory Capacity

## Operating System Statistics

Statistic	Total
AVG_BUSY_TIME	116,897
AVG_IDLE_TIME	242,244
AVG_IOWAIT_TIME	53,542
AVG_SYS_TIME	20,811
AVG_USER_TIME	95,754
BUSY_TIME	469,494
IDLE_TIME	970,850
IOWAIT_TIME	215,489
SYS_TIME	84,604
USER_TIME	384,890
LOAD	0
OS_CPU_WAIT_TIME	537,600
RSRC_MGR_CPU_WAIT_TIME	0
PHYSICAL_MEMORY_BYTES	10,737,418,240
NUM_CPUS	4
NUM_CPU_CORES	4

If you suspect a memory issue, check with your OS Administrator to ensure Oracle is collecting and reporting correctly.

Memory Capacity =  
 $10737418240 / 1024 / 1024 / 1024 = 10 \text{ GB}$

base view: v\$osstat

This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.

# OS – Oracle Memory Requirements

## Memory Statistics

- [Process Memory Summary](#)
- [SGA Memory Summary](#)
- [SGA breakdown difference](#)

Mem Req = 3.3 GB

[Back to Top](#)

## Process Memory Summary

- B: Begin snap E: End snap
- All rows below contain absolute values (i.e. not diffed over the interval)
- Max Alloc is Maximum PGA Allocation size at snapshot time
- Hist Max Alloc is the Historical Max Allocation for still-connected processes
- ordered by Begin/End snapshot, Alloc (MB) desc

Category	Alloc (MB)	Used (MB)	Avg Alloc (MB)	Std Dev Alloc (MB)	Max Alloc (MB)	Hist Max Alloc (MB)	Num Proc	Num Alloc
B Other	232.42		1.59	5.91	69	72	146	146
SQL	82.23	77.68	0.59	6.27	74	79	140	139
Freeable	40.88	0.00	0.55	0.41	2		74	74
PL/SQL	4.50	3.40	0.03	0.02	0		6	146
E Other	275.08		1.90	6.98	69	72	145	145
Freeable	44.13	0.00	0.59	0.41	2		75	75
SQL	21.71	17.00	0.16	1.14	13		88	139
PL/SQL	4.53	3.37	0.03	0.02	0		6	145

## SGA Memory Summary

SGA regions	Begin Size (Bytes)	End Size (By
Database Buffers	2,415,919,104	
Fixed Size	2,076,176	
Redo Buffers	14,700,544	
Variable Size	721,420,784	

Add both SGA and PGA.

PGA: Use max "Alloc (MB) from either Begin or End".

base views:  
 v\$sgastat,  
 v\$pgastat

## OS – Memory Utilization

$$U = \frac{R}{C} = \frac{3.3}{10.0} = 0.33 = 33\%$$

Remember this 33% is only for THIS instance and does not include any other memory.

ROT: If memory utilization is greater than 75%, performance can be significantly impacted. It can affect CPU and IO performance... really bad.

This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.

## What is the OS bottleneck?

## Is there is a bottleneck?

raPub

### IO Subsystem Summary

IO is [redacted] from an Operating System perspective. From an Oracle mb/sec perspective, Oracle processes are somewhat likely waiting too much for IO resources. For this Oracle instance, IO activity averaged 42.03 mb/sec (R:40.12 W:1.91) and 208.22 IOPS (R:68.26 W:139.96).

As mentioned above, IO read activity is significantly more intense than IO write activity. To give you an idea of the IO subsystem read response time, when an Oracle process makes a non buffered physical read (ref:dpr) request, it takes an average of 12.3 ms to complete.

A general IO read time rule of thumb threshold is 10 ms. Since the Oracle process average read time is higher than our rule of thumb, we have two IO subsystem focused solutions: 1) Reduce IO read times by either using faster IO devices or increasing the number of IO devices. 2) Reduce the number of IO requests that result in physical disk activity. An example of this is to increase the IO subsystem read cache or change the RAID configuration to reduce the IO request to physical disk read ratio. Consult your IO team for the best solution path.

### Memory Subsystem Summary

The memory subsystem [redacted] bottleneck. The Oracle caches and process global area are consuming 49% (15.5 GB) of the database server memory (31.5 GB). The operating system is paging memory to disk at an average rate of 1.9 KB/sec.

### Network Subsystem Summary

The network subsystem [redacted] bottleneck. Oracle specific network traffic averaged 4.8 mb/sec and Oracle processes were not likely to be waiting for network resources.

Where did Stori get the data?  
Which area provides opportunity?  
This is analysis is NOT based on the previous AWR data.

stori

raPub

(c)2013 OraPub, Inc.

Intro to TBA

This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.

raPub

### CPU Subsystem Summary

The CPU subsystem [redacted] bottleneck. The average CPU utilization of the 12 core server is 16%.

Referring the CPU Response Time plot, the database server is operating at the point where the arrival rate (the horizontal axis) is at 5.07 prior/ms. Notice where the arrival rate intersects the response time curve. As you can see, the system is operating far from being deep into the elbow of the response time curve.

The CPU subsystem appears to be fine. Be on the lookout for opportunities to use available CPU resources to improve performance. For example, consider increasing application focused parallelization.

CPU Response Time

Arrival Rate (prior/ms)	Response Time (ms/prior)
0	0.24
10	0.24
20	0.24
30	0.24
35	0.48
40	0.96
45	1.2

stori

raPub

(c)2013 OraPub, Inc.

Intro to TBA

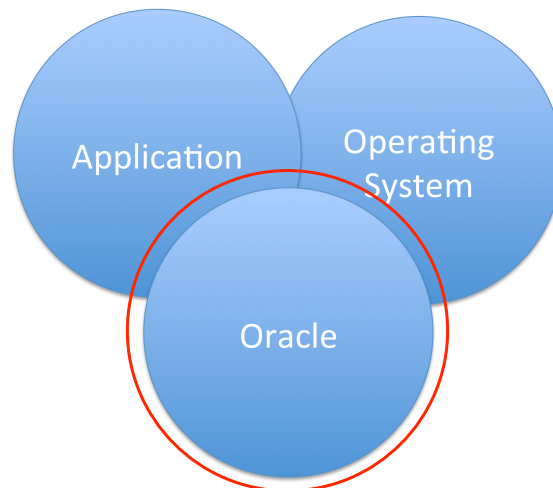
# The answer is...

## Operating System Performance Analysis

After analyzing the operating system CPU, IO, network, and memory subsystems, there is a somewhat serious IO situation. Each subsystem is analyzed below in more detail.

This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.

# Holistic Performance Analysis



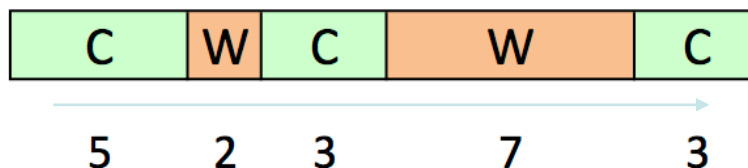
# Oracle **T**ime **B**ased **A**nalysis

- Modern Oracle performance analysis is about meeting user and business requirements, in large part by reducing *time*.
- *Time* to “run” a SQL statement, a batch job, a module, or business function.
- Batch jobs: Focus on reducing duration.
- OLTP: Focus on reducing elapsed time.
- **Unit of Work Time Based Analysis** focuses on the time it takes to process a single unit of work and allows the unification of an Oracle analysis with Operations Research.

This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.

## Elapsed time from an Oracle time perspective.

Elapsed Time = CPU Time + Wait Time



**CPU** = 11

**Wait** = 9

-----  
**E** = 20

Single Session

# It's a lot about instrumentation.

Instrumentation is the branch of science that deals with measurement and control in order to increase efficiency.

Oracle's kernel code instrumentation allows us to better understand where Oracle processes spend their time.

We could instrument our workday by keeping records of where we spent our time. The result would be a time card.  
We can ask Oracle for its "time card."

This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.

## Source code instrumentation.

- Instrumented source code has time-gathering functions placed into the source code enabling an understanding of where time is spent.
- It can enable fast and accurate diagnosis.
- Yes, there is overhead but without instrumentation you have a limited understanding of where the performance problems reside.
- Sampling is another option; less disruptive but cannot provide completely accurate information and sometimes cannot provide the detail needed.
- There are different levels/types of instrumentation.

# There is actually a “family” of core wait event views.

- **v\$event\_name**. Lists all Oracle wait events including their Oracle assigned classification.
- **v\$system\_event**. High-level view of all wait events.
- **v\$session\_event**. A high-level *session* specific view of wait events. Time is updated when a new event is waited upon.
- **v\$session\_wait** and **v\$session (10g+)**. A low-level session specific view of wait events. Time may be updated only every three seconds.
- **v\$event\_histogram (10g+)**. Storing wait times occurrences by event, a better understanding of wait times is possible. Waits times are skewed from their average.

This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.

# Oracle wait time over an interval.

## Wait Events

- s - second
- cs - centisecond - 100th of a second
- ms - millisecond - 1000th of a second
- us - microsecond - 1000000th of a second
- ordered by wait time desc, waits desc (idle events last)

Event	Waits	%Time -outs	Total Wait Time (s)	Avg wait (ms)	Waits /txn
db file sequential read	489,550	0.00	587	1	252.35
db file scattered read	12,142	0.00	565	47	6.26
direct path read temp	34,932	0.00	470	13	18.01
log file parallel write	6,253	0.00	235	38	3.22
db file parallel write	4,029	0.00	176	44	2.08
direct path write temp	30,705	0.00	180	5	15.83
log buffer space	416	0.00	46	111	0.21
log file sync	1,938	0.00	38	20	1.00

NI Wait time = 587 + 565 + 470 + 235 + 176 + 180 + other = 1848 sec

```
base view:
wait: v$system_event
cpu : v$sys_time_model
```

# Oracle process CPU consumption

## Time Model Statistics

- Total time in database user-calls (DB Time): 5488.7s
- Statistics including the word "background" measure background
- Ordered by % or DB time desc, Statistic name

Total CPU =  
3641 + 84 = 3725

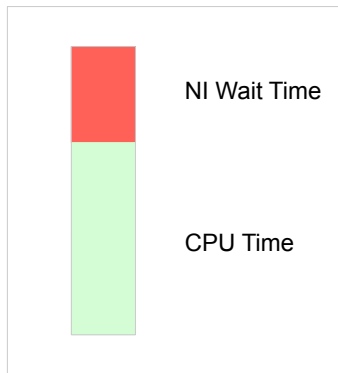
Statistic Name	Time (s)	% of DB Time
sql execute elapsed time	5,385.61	98.12
DB CPU	3,641.47	66.34
parse time elapsed	139.84	2.55
hard parse elapsed time	132.46	2.41
PL/SQL execution elapsed time	12.18	0.22
PL/SQL compilation elapsed time	9.35	0.17
connection management call elapsed time	9.21	0.17
hard parse (sharing criteria) elapsed time	5.65	0.10
sequence load elapsed time	1.19	0.02
failed parse elapsed time	0.76	0.01
repeated bind elapsed time	0.46	0.01
hard parse (bind mismatch) elapsed time	0.31	0.01
DB time	5,488.72	
background elapsed time	534.09	
background cpu time	84.49	

base view:  
v\$sys\_time\_model



This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.

# Database time



“DB time” does NOT include background process CPU consumption, though all OS processes compete for CPU resources. You decide.

NI Wait time = 587 + 565 + 470 + 235 + 176 + 180 + other = 1848 sec  
 CPU time = 3641 sec = DB CPU = Oracle foreground process CPU time  
 Big Bar Time = DB time = 5489 sec = NI Wait time + CPU time





# Don't start the story here.



How can I help you? get oracle wait time by event top 6

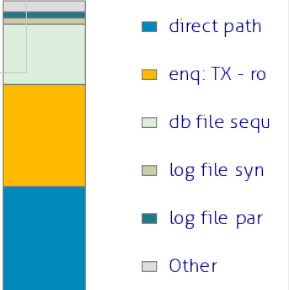
```

direct path read,2530.741517
enq: TX - row lock contention,2432.46223
db file sequential read,1445.357118
log file sync,168.80679
log file parallel write,154.141394
read by other session,85.438134
    
```


How can I help you? get chart bigbar oracle events

work/1376145486\_oracle\_event\_x\_BB.png

### Oracle Top 5 Wait Events



- direct path
- enq: TX - ro
- db file sequ
- log file syn
- log file par
- Other




(c)2013 OraPub, Inc.

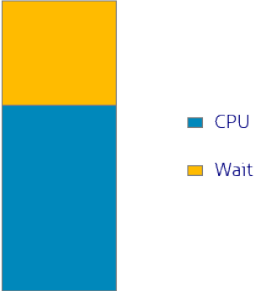
Intro to TBA

This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.

# Start the story at a high and natural level.



### Oracle Time Focus



- CPU
- Wait

base views:  
v\$sys\_time\_model  
v\$system\_event

How can I help you? get chart bigbar oracle

work/1376085158\_oracle\_x\_BB.png

How can I help you? get stats systemmodel db%cpu

```

db cpu, 12689550897
    
```

How can I help you? get stats systemmodel background%

```

background elapsed time, 316661099
background cpu time, 75689503
    
```

How can I help you? get oracle cpu consumed


```

12689.551
    
```

How can I help you? get oracle wait time by class nonidle

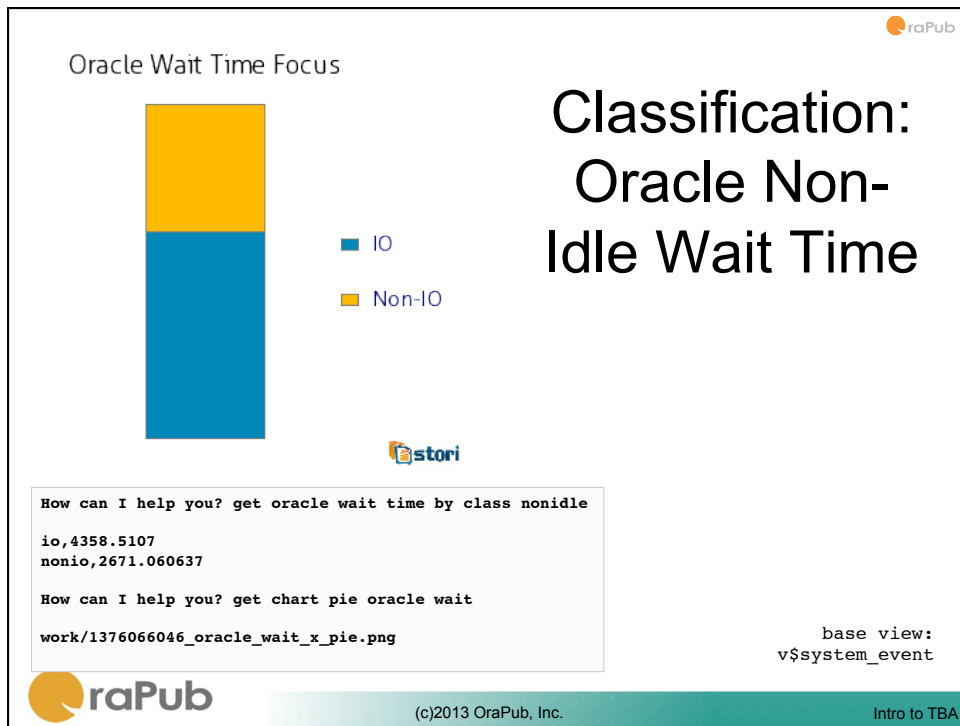
```

io,4358.5107
nonio,2671.060637
    
```

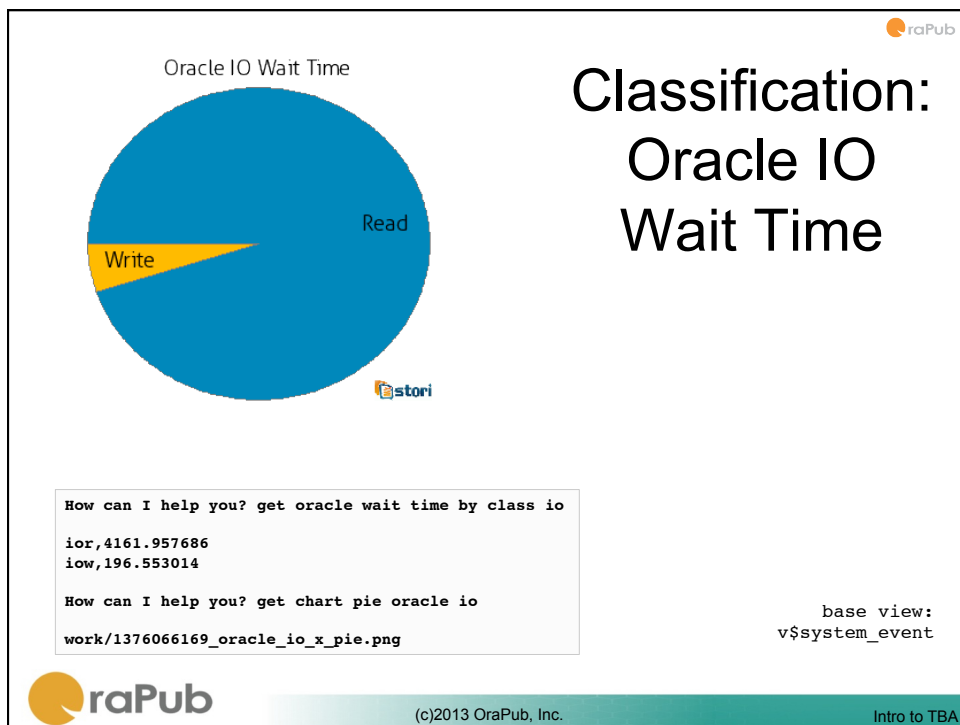


(c)2013 OraPub, Inc.

Intro to TBA



This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.



raPub

# Classification: Oracle Read IO Wait Time

How can I help you? get oracle wait time by event class ior top 3

```


direct path read,2530.741517
db file sequential read,1445.357118
read by other session,85.438134
        
```

How can I help you? get oracle wait time by event name db%seq%read

```

db file sequential read,1445.357118,8.609004
        
```

base view:  
v\$system\_event


 (c)2013 OraPub, Inc. Intro to TBA

This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.

raPub

# Classification: Oracle Non-IO Wait Time

Oracle Non-IO Wait Time



- concur
- other
- serial
- network

base view:  
v\$system\_event


How can I help you? get oracle wait time by event class nonio top 3

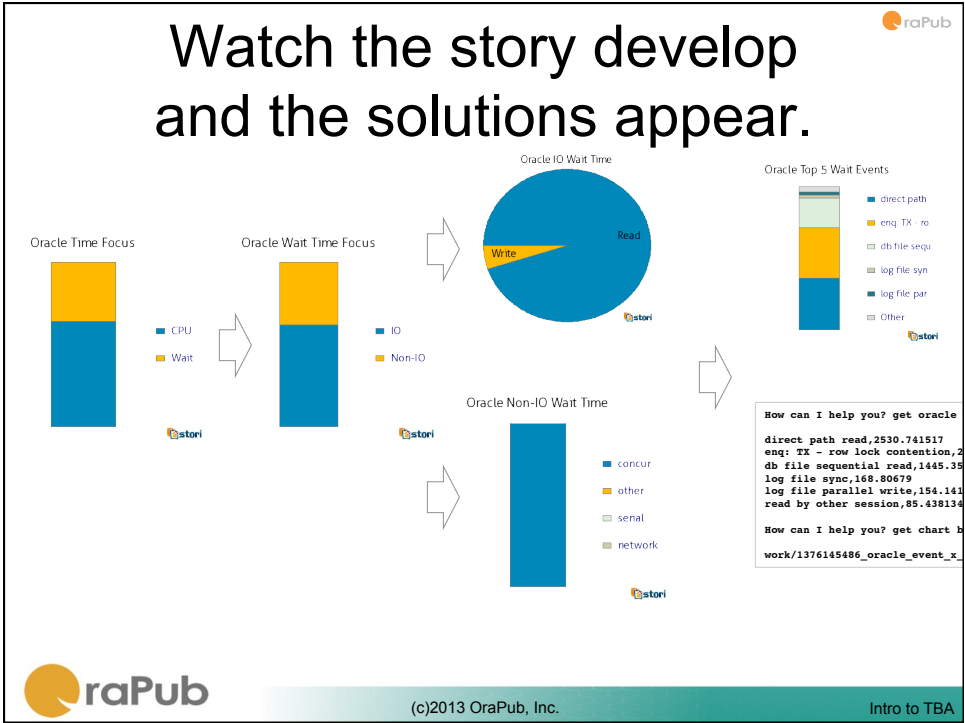
```

eng: TX - row lock contention,2432.46223
log file sync,168.80679
eng: TX - index contention,38.995883
        
```

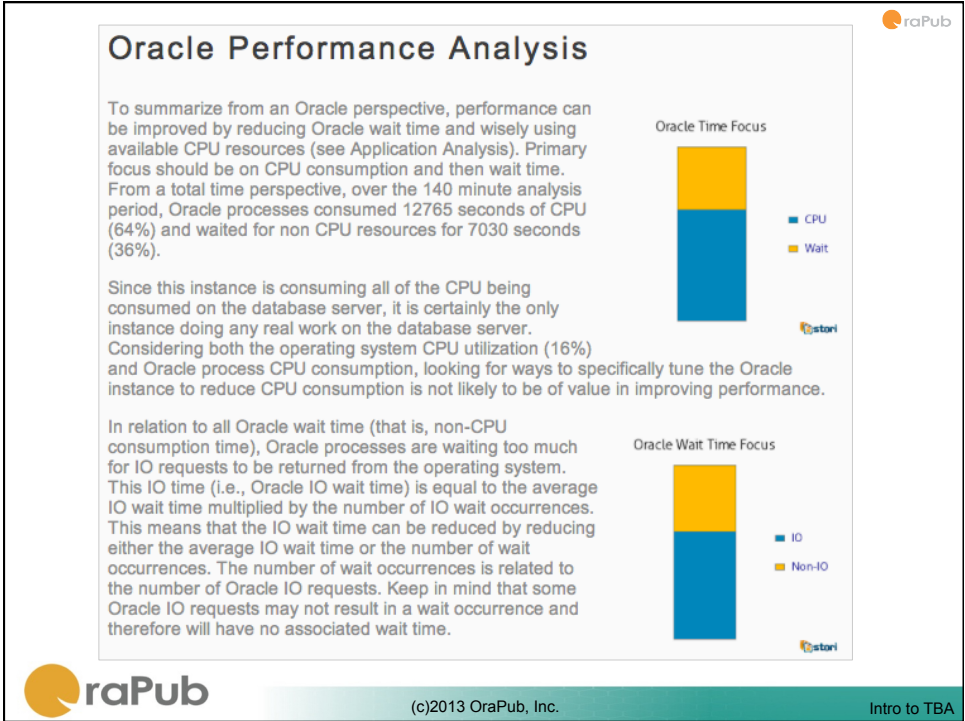
How can I help you? get chart bigbar oracle nonio

work/1376066704\_oracle\_nonio\_x\_BB.png

 (c)2013 OraPub, Inc. Intro to TBA



This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.



# Firefighting Diagnostic Template

	Total (s)	Avg (ms)
<b>Total Rt (sec)</b>	51427.600	
<b>Total CPU Time</b>	16881.600	
<b>Srvr Proc</b>		16749.800
<b>BG Proc</b>		131.800
<b>Total Non-Idle Wait Time</b>	34546.000	
<b>IO</b>	0.000	0.000
<b>Read</b>		
1 na	0.000	0.000
2 na	0.000	0.000
3 na	0.000	0.000
<b>Write</b>		
1 na	0.000	0.000
2 na	0.000	0.000
3 na	0.000	0.000
<b>Other</b>	34546.000	
1 latch: cbc	31764.000	591.000
2 latch free	1851.000	13.000
3 latch: cb handles	931.000	13.000

Go to OraPub.com and search for "ff diag"

This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.

# Oracle TBA Template

```

OraPub Oracle Response Time Analysis Template:
CPU cores: __, Dur: __m, __s CPU capacity: __s
Workload
- uc/s : __/s redo/s : __mb/s
- exe/s : __/s blks chng/s : __/s
- lio/s : __k/s pio/s : __k/s
- IOPS R: __ IOPS W : __
- IO R: __mb/s IO W : __mb/s
ORT Total: __s
-- CPU Total: __s __% of RT (oracle util%: __)
-- Srvr Proc: __s __% of CPU Total
-- BG Proc: __s __% of CPU Total
-- Non-Idle Wait Time Total: __s __% of ORT Total
-- IO: __s __% of NI Wait Time Total
-- R: __s __% of IO
-- #1: __s __ms
-- #2: __s __ms
-- #3: __s __ms
-- W: __s __% of IO
-- #1: __s __ms
-- #2: __s __ms
-- #3: __s __ms
-- OTHER: __s __% of NI Wait Time Total
-- #1: __s
-- #2: __s
-- #3: __s
    
```

Moving people to thinking in a pure response time like way...

[http://filebank.orapub.com/perf\\_stats/ORTA\\_Template\\_2e.pdf](http://filebank.orapub.com/perf_stats/ORTA_Template_2e.pdf)

raPub

# Holistic Performance Analysis

Limiting data source to Oracle views.

raPub

(c)2013 OraPub, Inc. Intro to TBA

This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.

raPub

## Think and Ask...

### What type of SQL are we looking for?

Situation:

- High consuming CPU SQL ... want to tune (oltp) and parallelize (DW, oltp)
- Available CPU, so parallelize long run SQL.
- High consuming IO read SQL ... want to reduce IO read consumption.
- Row level locking SQL ... DML that changes one or more rows.

- CPU/LIO/Buffer Gets
- PIO-R
- DML
- Sorting
- Query
- Parsing
- Long running
- ...

base views:  
v\$sql (Statspack),  
v\$sqlstats

raPub

(c)2013 OraPub, Inc. Intro to TBA

## WORKLOAD REPOSITORY report for

### SQL Statistics

- [SQL ordered by Elapsed Time](#)
- [SQL ordered by CPU Time](#)
- [SQL ordered by Gets](#)
- [SQL ordered by Reads](#)
- [SQL ordered by Executions](#)
- [SQL ordered by Parse Calls](#)
- [SQL ordered by Sharable Memory](#)
- [SQL ordered by Version Count](#)
- [Complete List of SQL Text](#)

[Back to Top](#)

Once you know the type of SQL you're looking for, Statspack and AWR will like have categorized the SQL for you, making it easy to find the SQL that warrants your attention.

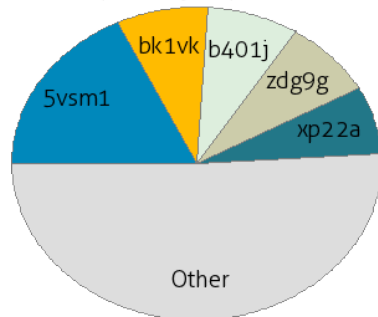
base views:  
v\$sql (Statspack),  
v\$sqlstats



This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.

## Find top elapsed time SQL.

Top SQL By Elapsed Time



Last five SQL ID characters.

Goal?  
Reduce elapsed time through reducing resource consumption and/or parallelizing.

```
How can I help you? get sql id by elapsed_time_s top 3 value
8pq06s4k5vsm1 (2819.236835) bgx1d4x6bk1vk (1325.145248) 5p8xmhxb401j (1323.546347)

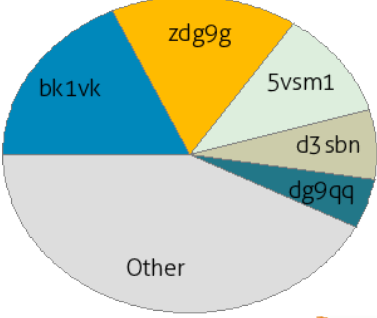
How can I help you? get chart pie appl sql id by elapsed_time_s
work/1376068709_appl_sql_elapsed_pie.png
```




# Find top CPU time SQL.

raPub

Top SQL By CPU



Last five SQL ID characters. 

How can I help you? get sql id by cpu\_time\_s top 3 value


bgx1d4x6bk1vk (1312.502491) dygn0jztzd9g (1178.448831) 8pq06s4k5vsm1 (813.668307)

How can I help you? get chart pie appl sql id by cpu\_time\_s

work/1376069057\_appl\_sql\_cpu\_pie.png

Goal:  
Reduce elapsed time by CPU consumption (serial) and/or by **increasing** CPU bust by parallelizing.

base views: v\$sql (Statspack), v\$sqlstats



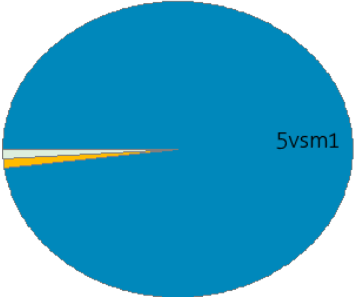
(c)2013 OraPub, Inc. Intro to TBA


This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.

# Find top IO Read time SQL.

raPub

Top SQL By Phys Blocks Read



Last five SQL ID characters. 

How can I help you? get sql id by disk\_reads top 3 value


8pq06s4k5vsm1 (40231964) 86uzc3dybkg58 (438512) 97hwq2fjz8xk4 (305120)

How can I help you? get chart pie appl sql id by disk\_reads

work/1376069422\_appl\_sql\_pior\_pie.png

Goal:  
Reduce elapsed time by reducing IO reads (serial) and/or by increasing CPU consumption bust by parallelizing.

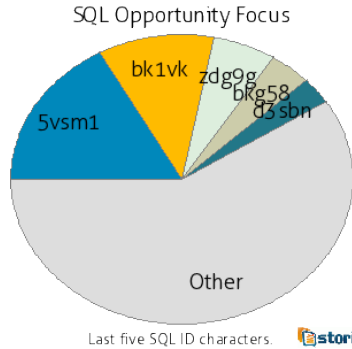
base views: v\$sql (Statspack), v\$sqlstats



(c)2013 OraPub, Inc. Intro to TBA



# Now it's time to prioritize.



It's important to factor in uptime time requirements, confidence in meeting the stated goal, time constraints, budget constraints, etc.

```

How can I help you? get top sql 3 value
8pq06s4k5vsm1 (2.333)  bgx1d4x6bk1vk (1.5)  dygn0jztzdg9g (.75)

How can I help you? get chart pie appl sql
work/1376069574_appl_sql_opr_pie.png
    
```



This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.

## Application Performance Analysis

Combining the operating system and Oracle system performance analysis, the application analysis focus will center squarely on identifying SQL statements which directly contribute to the performance problems.

Oracle processes spend more time consuming CPU (12765 secs) compared to waiting for non CPU resources (7030 secs). This information combined with what is occurring at the operating system level and in Oracle will guide us in our search for the key application SQL.

### CPU Consumption

While Oracle processes are, relatively speaking, consuming more CPU compared to waiting for non CPU resources, some CPU resources are available.

This presents us with an opportunity!

We want to creatively use available CPU resources. Look for long running CPU intensive SQL statements or a CPU intensive batch job running as a single serial process. They will likely have a low execution count along with a high elapsed time. By modifying the application to parallelize the SQL or batch job, its duration will decrease because multiple processes will run concurrently. It is like having a single large shopping list but having three people find specific items instead of one person finding all the items. This approach will consume unused CPU resources while reducing run times.

For example, the top CPU consuming SQL statement (SQL\_ID=bgx1d4x6bk1vk) consumes 10% of all Oracle CPU resources (SQL, and anything else). Also, the elapsed time for longest running SQL statement (SQL\_ID=8pq06s4k5vsm1) is 10% of all cumulative SQL elapsed times. To gain a better understanding of how SQL statements and application modules compare, look below at the Top SQL Matrix and the Top Module Matrix respectively.

Slow running CPU intensive OLTP SQL that can not be parallelized will need to be tuned. A SQL statement requiring four million logical IOs with each logical IO taking only 1 ms will still run in 4 seconds.

### IO Consumption

The operating system is not able to meet the application IO read requirements. This presents us with an opportunity! To help reduce the requirements it is important to identify the top IO read SQL and then reduce either their execution rate or IO read requirements per execution.

For example, the top physical read consuming SQL statement (SQL\_ID=8pq06s4k5vsm1) consumes 94% of all physical blocks read.

Slow running IO intensive OLTP SQL that can not be parallelized will need to be tuned. A SQL statement requiring two million physical IOs with each physical IO taking only 2 ms will still run in 4 seconds.



Turn the Application focused analysis into an understandable story.



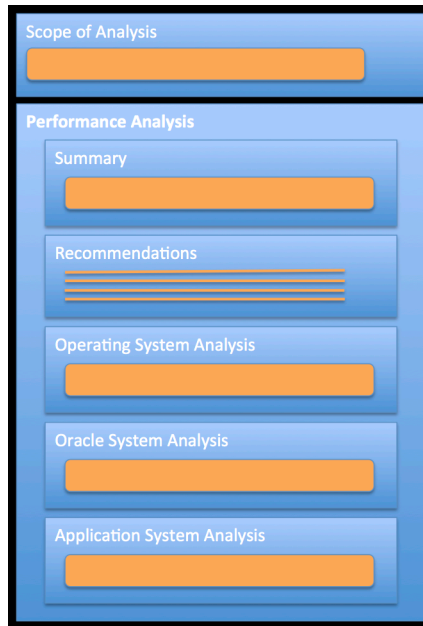
# Make your recommendations.

## Recommendations

- Look for opportunities to parallelize CPU intensive processes, which will increase the CPU workload on the server and decrease process duration. (rank: .419)
- Reduce IO read times by either using faster IO devices or increasing the number of IO devices. (rank: .235)
- Reduce the number of IO requests that result in physical disk activity. An example of this is to increase the IO subsystem read cache or change the RAID configuration to reduce the IO request to physical disk read ratio. (rank: .235)
- For SQL ID 8pq06s4k5vsm1 focus on reducing elapsed time and physical reads. (rank: .142)
- For module CustomizedStartup@CustomizedStartup focus on reducing elapsed time by reducing IO activity and by appropriately increasing CPU consumption. (rank: .137)
- To reduce time related to non buffered physical reads, if blocks are likely to be read multiple times, encourage Oracle to buffer the blocks reducing IO requests. Even if the average wait time is 1 ms, there is overhead related to finding space in the buffer cahce, requesting the blocks from the IO subsystem, and then placing them into the buffer cache. Carefully consider increasing the buffer cache, caching popular application tables, and implementing Oracle compression features. (rank: .128)

This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.

## Create your deliverable.



Notice the “3-Circle Analysis” structure is built directly into the deliverable.

This makes telling the performance story much easier.

# Review Question.

## Always remember what four things?

This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.

## Want to dig deeper?

- **Presentations:** OraPub search, "time"
- **Craig's Blog** – A Wider View
- **Training** from OraPub
  - Oracle Performance Firefighting (I)
  - Adv Oracle Performance Analysis (II)
- **Tools** at [www.orapub.com](http://www.orapub.com)
  - OSM Toolkit. OP search, "osm"
  - Firefighting Diagnostic Template. OP search "ff diag"
- **Stori.** Interactive. Automated. Find and solve Oracle performance problems.
- **Books**
  - Oracle Performance Firefighting. "FF453" \$10 discount
  - Forecasting Oracle Performance.



White Plains, NY  
Sep 30 – Oct 4

Costa Mesa, CA  
December 9-13

raPub

# Q&A

## Thank you for attending.

More questions?  
Contact Craig at  
[craig@orapub.com](mailto:craig@orapub.com) - [www.orapub.com](http://www.orapub.com)

Get StoriFree at  
<http://storifree.com>

raPub

Last twelve module characters. 

(c)2013 OraPub, Inc. Intro to TBA

This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.