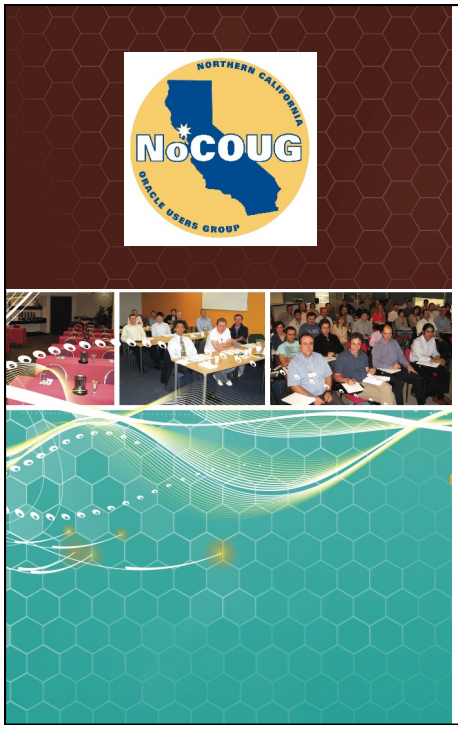


Introduction to Time-Based Analysis: Stop the Guessing!

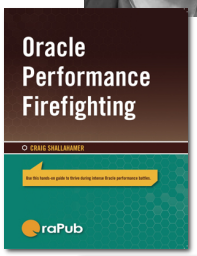
raPub

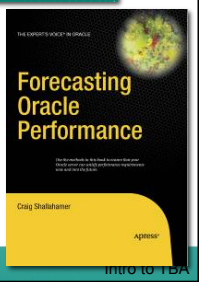
Craig A. Shallahamer
OraPub, Inc.
craig@orapub.com

LinkedIn XING

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

raPub

raPub

focusing exclusively on Oracle systems performance analysis

Automated. Interactive.
Oracle SE Performance Analysis

www.storifree.com
www.stori.orapub.com

Resources

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

raPub

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

raPub

Agenda

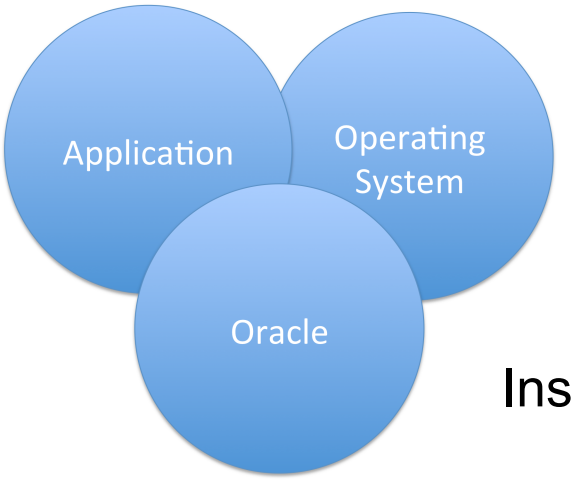
- Always remember these four things.
- Get the diagnostic data
- Building the framework
- Example

raPub

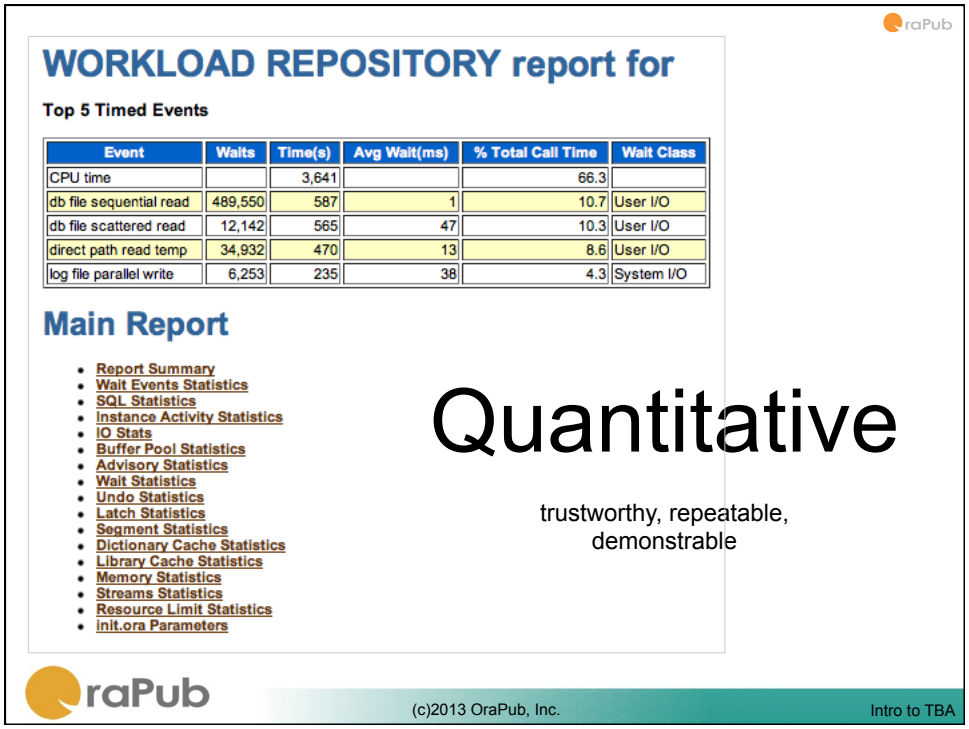
Always remember these four things.

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

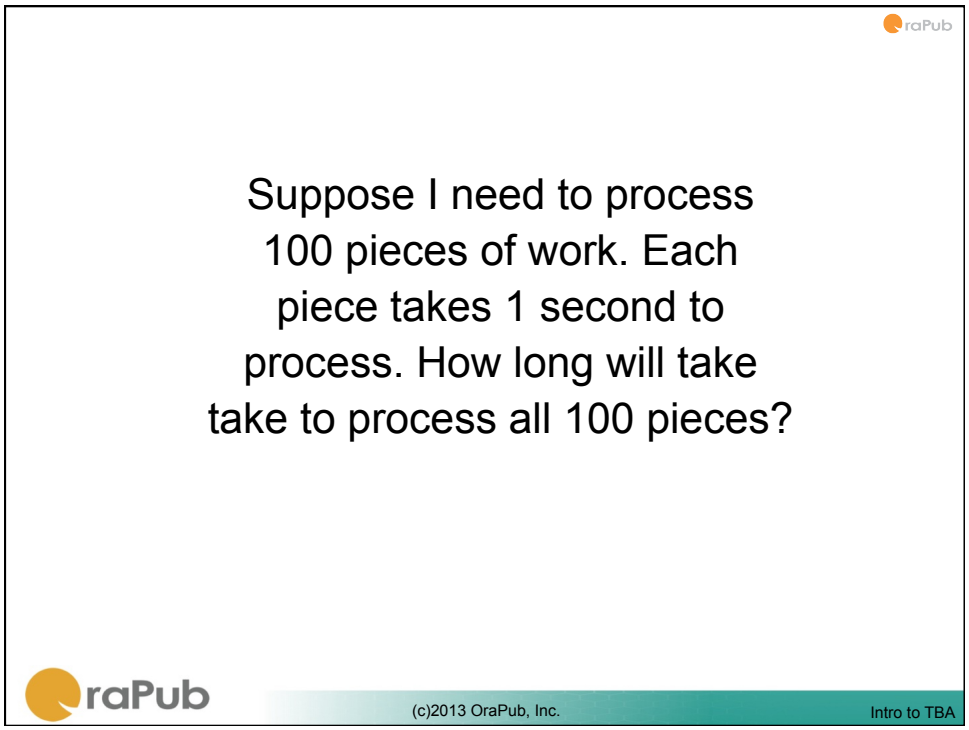
Holistic Performance Analysis



Insurance



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

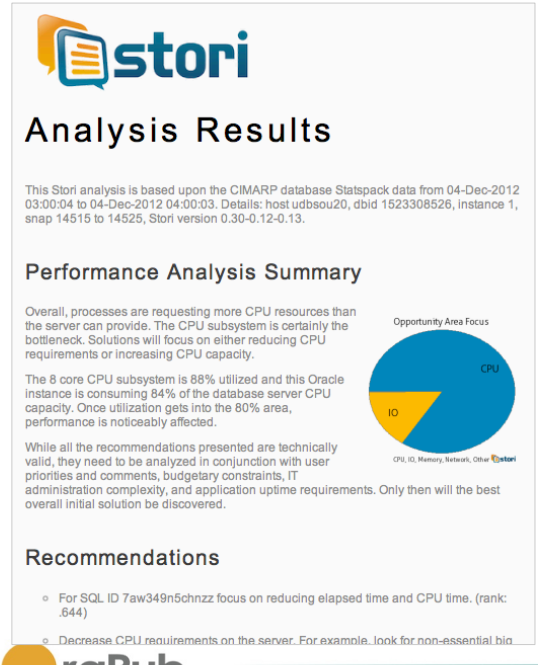


Serialization is death.

(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.



Tell a story.

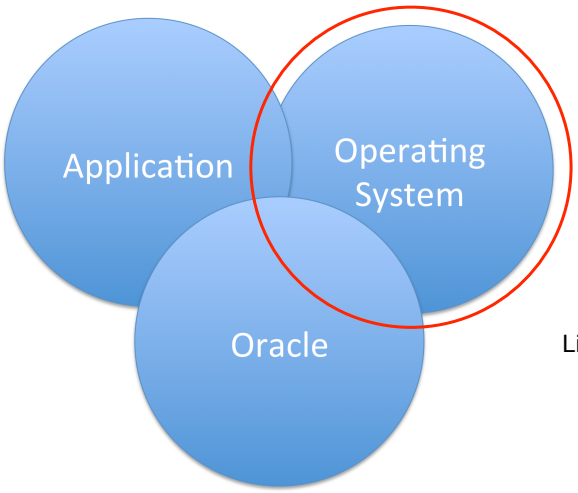
understandable

Get the diagnostic data

Build the framework.

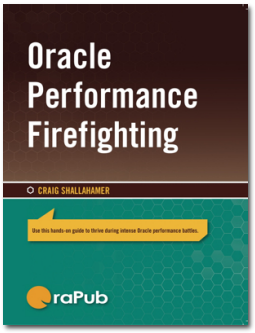
This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.

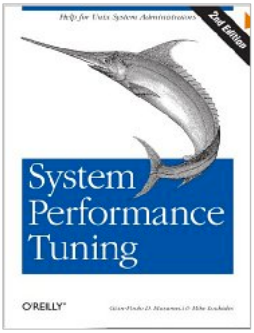
Holistic Performance Analysis



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$system_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$system_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 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

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	

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	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	138
	PL/SQL	4.53	3.37	0.03	0.02	0	6	145	145

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?

IO Subsystem Summary

IO is r 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 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 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.

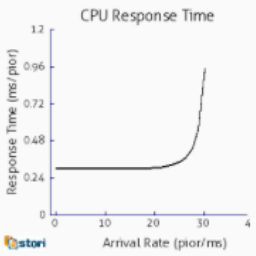
This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.

CPU Subsystem Summary

The CPU subsystem neck. The average CPU utilization of the 12 core server is 16%.

Refering the CPU Response Time plot, the database server is operating at the point where the arrival rate (the horizontal axis) is at 5.07 pior/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.



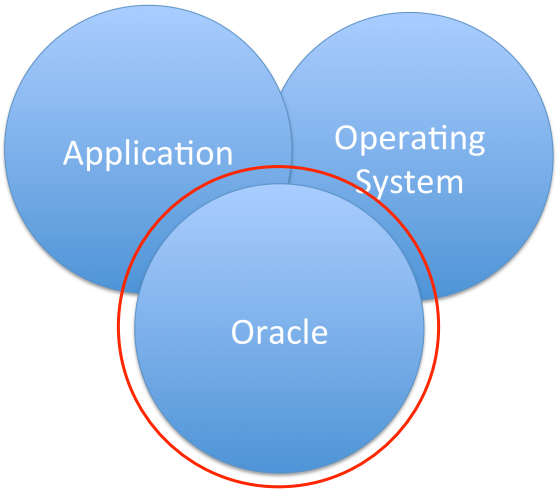
The answer is...

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



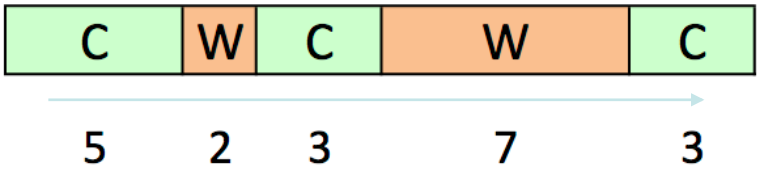
Oracle **T**ime **B**ased **A**nalysis

- 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.
- **U**nit of **W**ork **T**ime **B**ased **A**nalysis 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	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

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

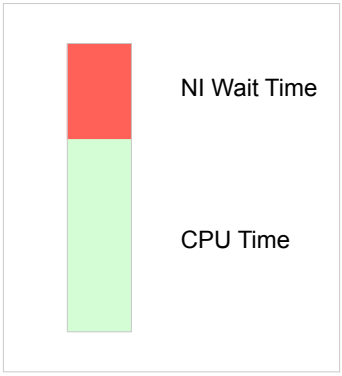
Statistic Name	Time (s)	% of DB Time
sql execute elapsed time	5,385.61	98.12
DB CPU	3,641.47	66.34
parse time elapsed	139.84	2.55
hard parse elapsed time	132.46	2.41
PL/SQL execution elapsed time	12.18	0.22
PL/SQL compilation elapsed time	9.35	0.17
connection management call elapsed time	9.21	0.17
hard 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	

Total CPU =
3641 + 84 = 3725

```
base view:
v$sys_time_model
```


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

Database time

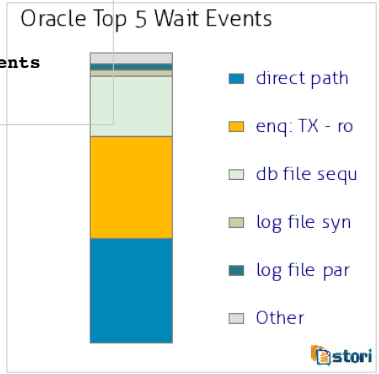


"DB time" does NOT include background process CPU consumption, though all OS processes compete for CPU resources. You decide.

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

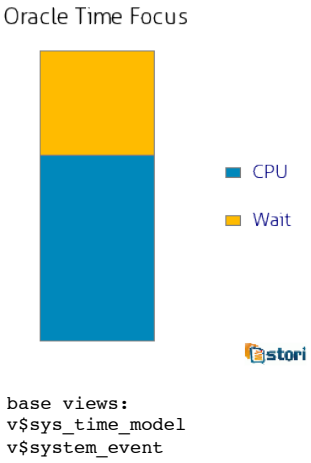
Don’t start the story here.

```
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
```



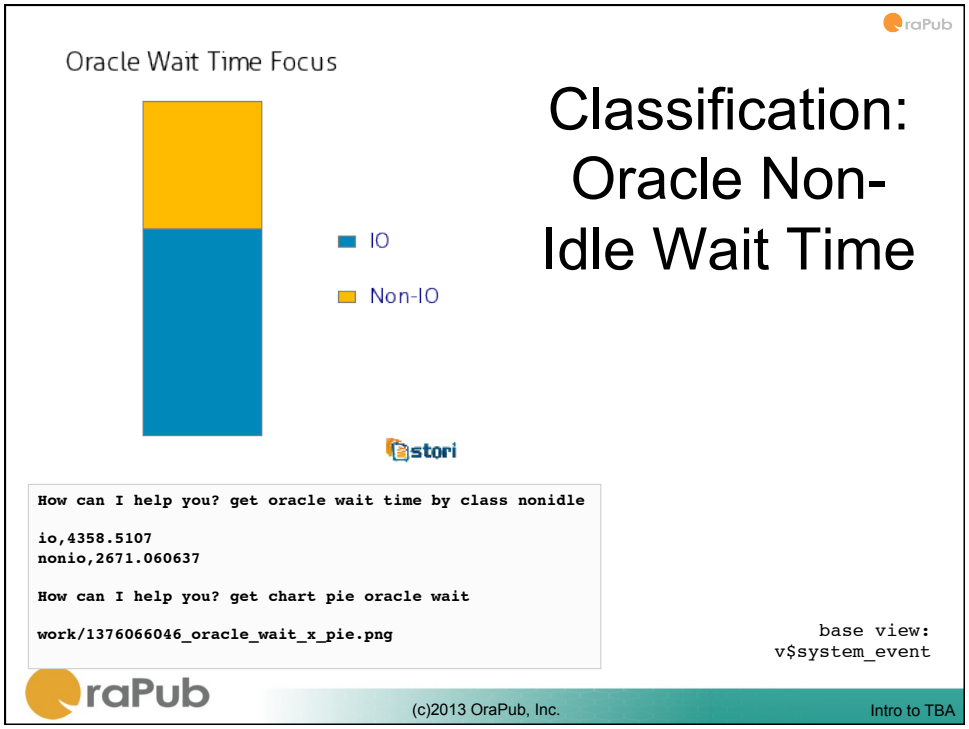
This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.

Start the story at a high and natural level.

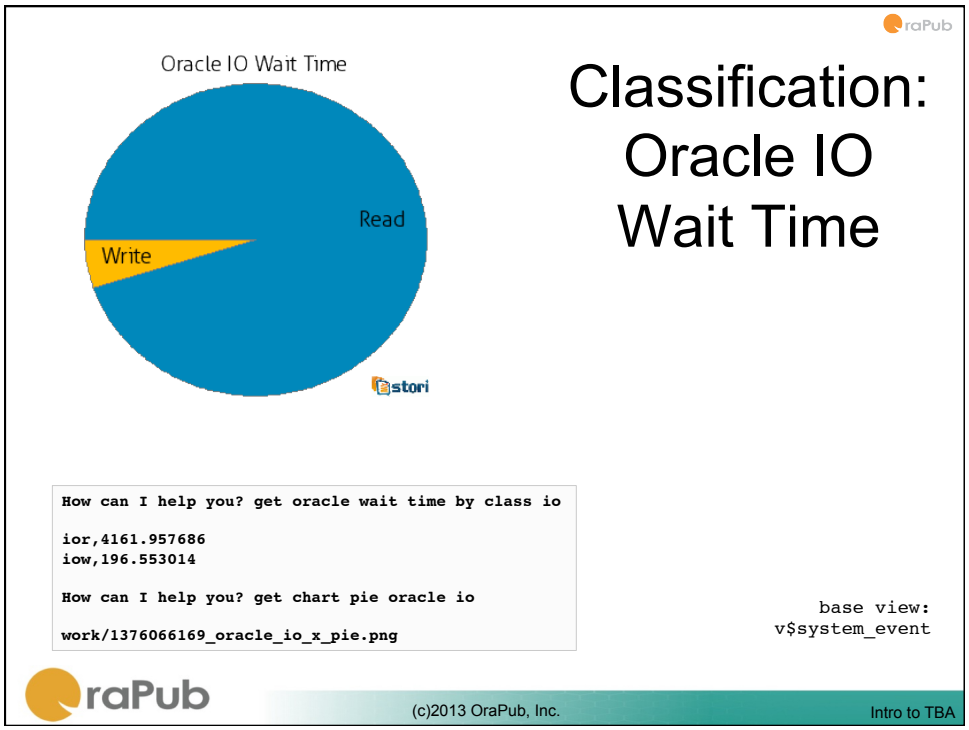


base views:
v$sys_time_model
v$system_event

```
How can I help you? get chart bigbar oracle

work/1376085158_oracle_x_BB.png

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

db cpu, 12689550897

How can I help you? get stats systimemodel 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
```



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



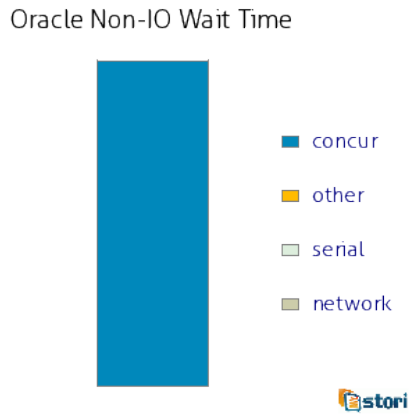
Classification: Oracle Read IO Wait Time

```
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

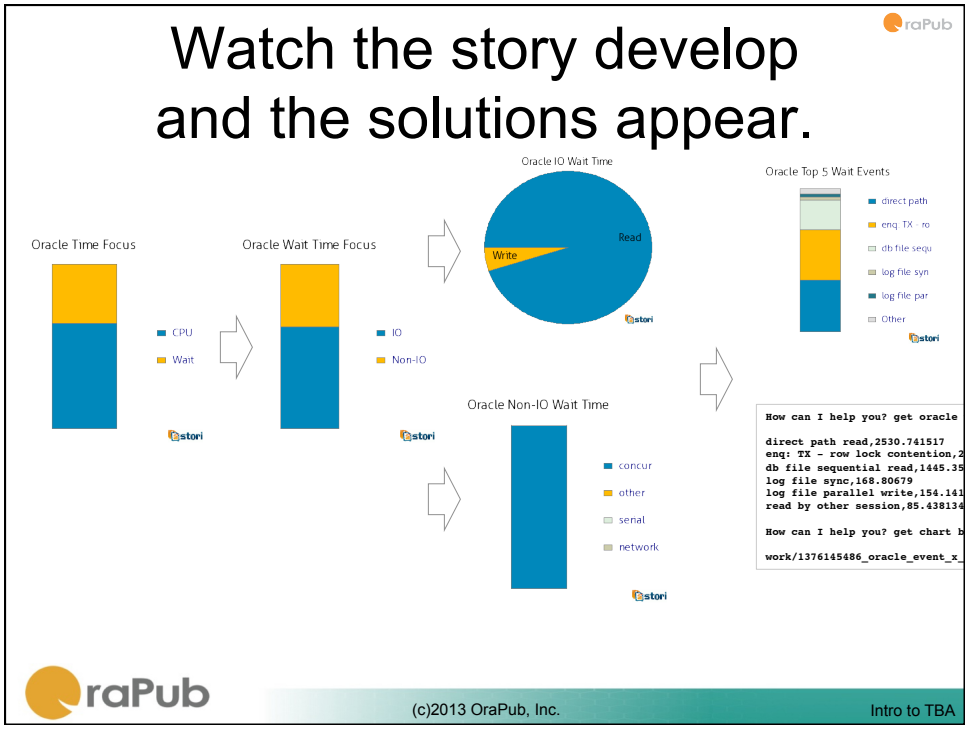
This presentation was given by Craig Shallahamer at the NoCOUG conference on 15-AUG-2013.

Classification: Oracle Non-IO Wait Time

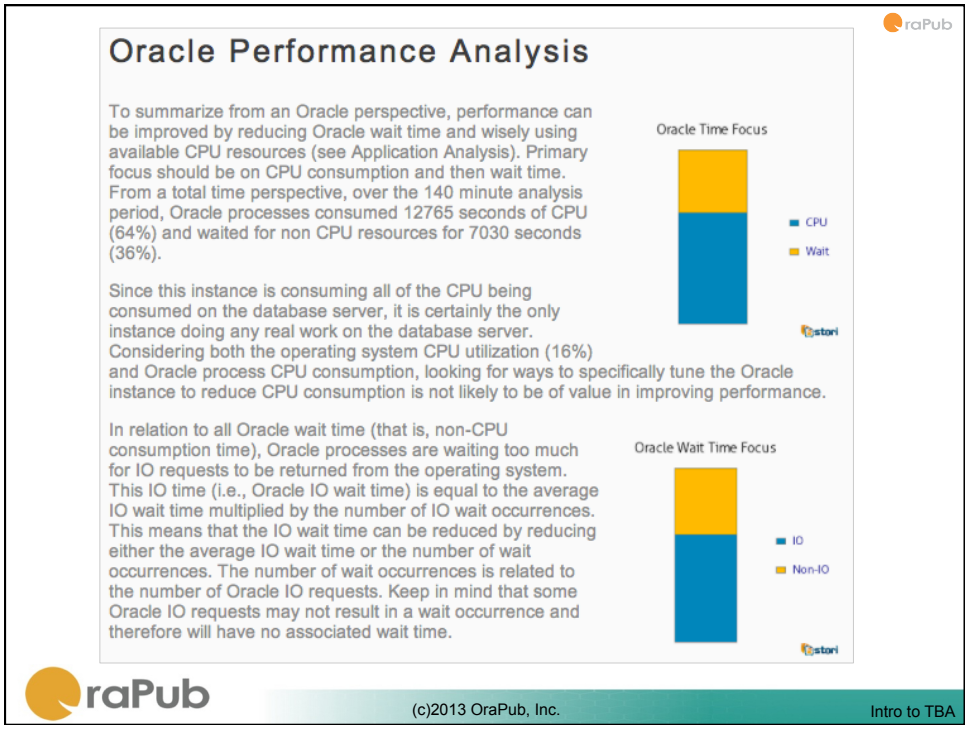



```
How can I help you? get oracle wait time by event class nonio top 3

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

How can I help you? get chart bigbar oracle nonio

work/1376066704_oracle_nonio_x_BB.png
```

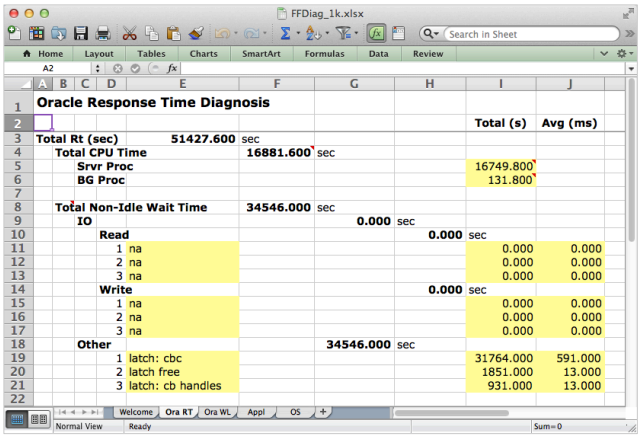
base view:
v$system_event



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



Firefighting Diagnostic Template



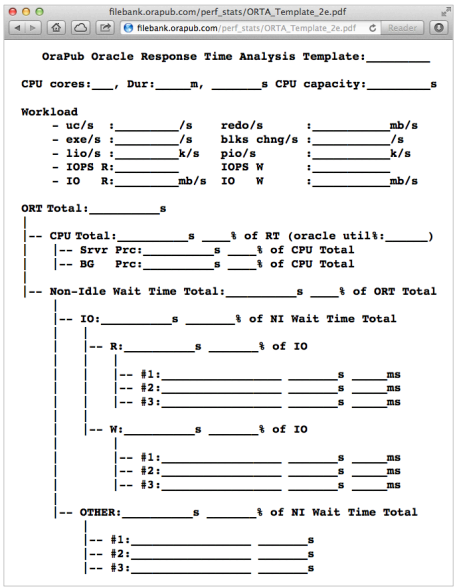
FFDiag_1k.xlsx

1	Oracle Response Time Diagnosis							
2							Total (s)	Avg (ms)
3	Total Rt (sec)		51427.600	sec				
4		Total CPU Time		16881.600	sec			
5			Srvr Proc				16749.800	
6			BG Proc				131.800	
7								
8		Total Non-Idle Wait Time		34546.000	sec			
9			IO			0.000	sec	
10				Read			0.000	sec
11					1 na		0.000	0.000
12					2 na		0.000	0.000
13					3 na		0.000	0.000
14				Write			0.000	sec
15					1 na		0.000	0.000
16					2 na		0.000	0.000
17					3 na		0.000	0.000
18			Other			34546.000	sec	
19					1 latch: cbc		31764.000	591.000
20					2 latch free		1851.000	13.000
21					3 latch: cb handles		931.000	13.000
22								

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



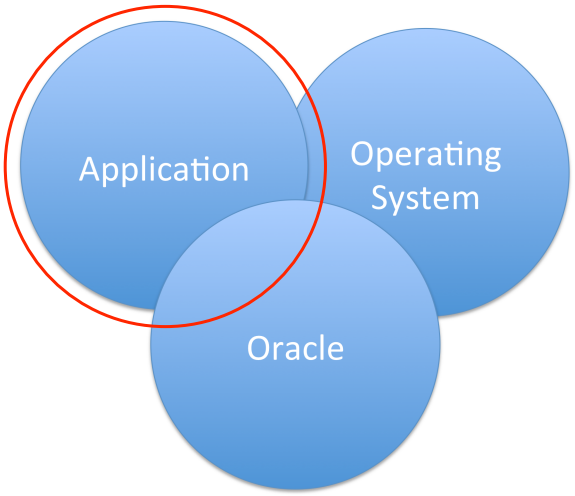
filebank.orapub.com/perf_stats/ORTA_Template_2e.pdf

```
   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 Prc:_________s ____% of CPU Total
|   |-- BG   Prc:_________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

Holistic Performance Analysis



Limiting data source to Oracle views.

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

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
```

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

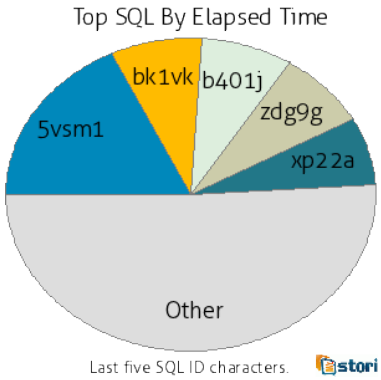
Back to Top

Once you know the type of 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.

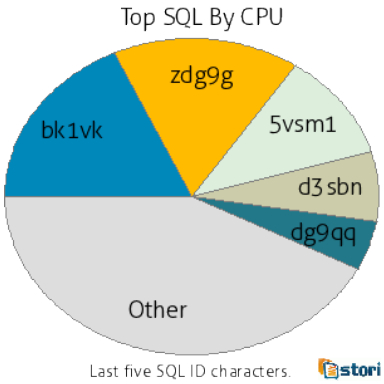
Find top elapsed time SQL.



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) 5p8xmhxhb401j (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.



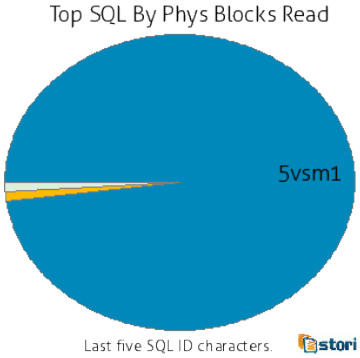
Goal:
Reduce elapsed time by CPU consumption (serial) and/or by **increasing** CPU bust by parallelizing.

```
How can I help you? get sql id by cpu_time_s top 3 value

bgx1d4x6bk1vk (1312.502491) dygn0jztzdg9g (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
```

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

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

Find top IO Read time SQL.

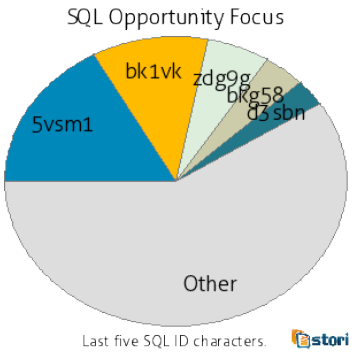


Goal:
Reduce elapsed time by reducing IO reads (serial) and/or by increasing CPU consumption bust by parallelizing.

```
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
```

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

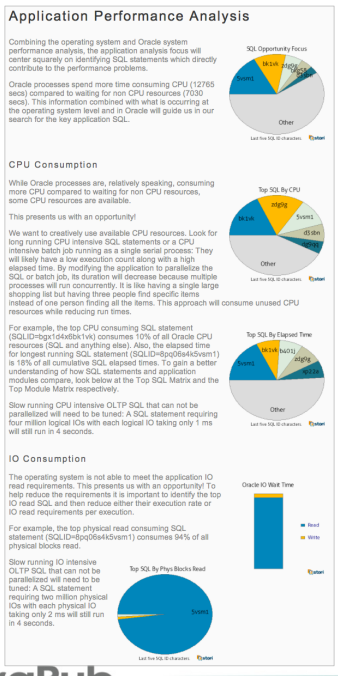
Now it's time to prioritize.



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.



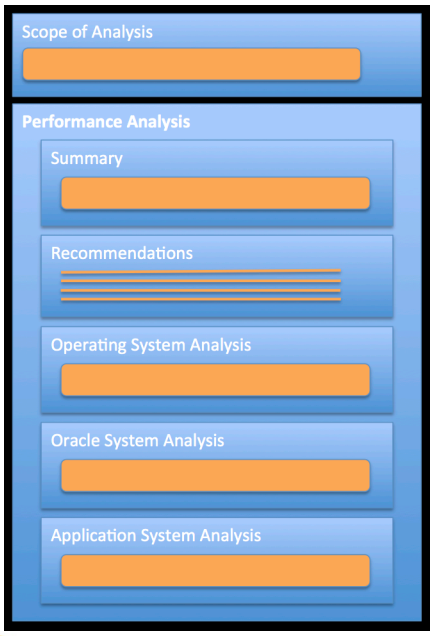
Turn the Application focused analysis into an understandable story.

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 mutiple times, encourage Oracle to buffer the blocks reducing IO requests. Even if the aveage 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
  - 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.

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