

# AWR Ambiguity: What to do when the numbers don't add up?

John Beresniewicz  
(12 yrs at Oracle, presently at Teradata Labs)

# an Oaktable inquiry...

The screenshot shows a Gmail interface with a dark theme. At the top, there's a navigation bar with 'Gmail' and several action buttons: a left arrow, a download icon, an exclamation mark, a trash can, 'Move to Inbox', a tag icon, and 'More'. On the left sidebar, there's a 'COMPOSE' button and a list of folders: 'Inbox (241)', 'Important', 'Sent Mail', 'Drafts', 'Spam', and 'Circles'. Below the folders is a profile picture of 'John' and a search icon. The main content area shows an email thread. The top part of the thread is from 'Oaktable member @yahoo.com' dated 'Apr 3'. The email body contains the text: '..picking your brain here... I'm sure you've explained this to us before... why is it that sometimes the top waits in AWR (Top 10) sum greater than 100 when "waits" for DB CPU is high?'. Below this is a response from 'John Beresniewicz <john.beresniewicz@gmail.com>' dated 'Apr 3'.

(I didn't really understand the question)

# ...can you send me the AWR?

```
Top 10 Foreground Events by Total Wait Time
~~~~~
```

Event	Waits	Total Wait Time (sec)	Wait Avg (ms)	% DB Wait time	Wait Class
db file parallel read	139,819	469.8	3.36	78.1	User I/O
DB CPU		243		40.4	
db file sequential read	201,921	52.4	0.26	8.7	User I/O
db file scattered read	221	.5	2.21	.1	User I/O
kksfbc child completion	4	.2	37.85	.0	Other
control file sequential read	249	0	0.10	.0	System I
Disk file Mirror Read	61	0	0.21	.0	User I/O
library cache load lock	19	0	0.60	.0	Concurre
cursor: pin S wait on X	4	0	1.61	.0	Concurre
Disk file operations I/O	140	0	0.03	.0	User I/O

**CPU + Wait = DB Time**

**40.4 + 86.8 = 127.2% (> 100% WTF???)**

# Things to keep in mind...

- AWR report presumes accurate instrumentation
  - this liability is unfortunate and unnecessary
- ADDM presumes accurate instrumentation
  - also unfortunate and unnecessary
- Instrumentation is not always accurate
  - this is why you need to understand it

# First questions to answer:

- Elapsed time of report?
  - performance analysis is always (ultimately) about time
- Version of Oracle DB?
  - bug lookup, report contents, available data
- System CPU-bound?
  - known instrumentation issues under CPU-stress

# Begin at the beginning...

WORKLOAD REPOSITORY report for

DB Name	DB Id	Instance	Inst Num	Startup Time	Release	RAC
DB12C	1329819247	db12cn1	1	04-Apr-15 06:19	12.1.0.2.0	NO

Host Name	Platform	CPUs	Cores	Sockets	Memory (GB)
oral.dssdhop.lab	Linux x86 64-bit	72	36	2	252.17

	Snap Id	Snap Time	Sessions	Curs/Sess
Begin Snap:	410	04-Apr-15 06:22:16	61	.8
End Snap:	411	04-Apr-15 06:24:18	56	.7
<b>Elapsed:</b>		<b>2.02 (mins)</b>		
<b>DB Time:</b>		<b>10.03 (mins)</b>		

DB version? 12.1.0.2

Elapsed time 2 minutes (120 secs)

CPU bound? NO (36 >> 5 ; Cores >> AAS)

# Next questions to answer:

- What is DB Time over interval?
- What is DB CPU over interval?
- What is (expected) Wait Time over interval?
  - $\text{DB Time} - \text{DB CPU} = \text{Wait Time (expected)}$
- Model:  $\text{DB Time} = \text{CPU Time} + \text{Wait Time}$

# DB Time is gold

- Session (foreground) time spent in DB calls
  - measured by Oracle
- Session DB Time has clear instrumentation points:
  - [**call entry**:start timer]...[**call exit**:stop timer]
- System DB Time = SUM(Session DB Time)
- We trust DB Time accuracy implicitly



# DB CPU is also gold *(except on AIX)*

- Foreground CPU actually used during reporting period
  - does not include run-queue time
- Measured by OS, collected by Oracle
  - independent instrumentation
- **We have high confidence in DB CPU accuracy**
  - NOTE: AIX CPU utilization reporting is seriously amiss for hyper-threaded cores

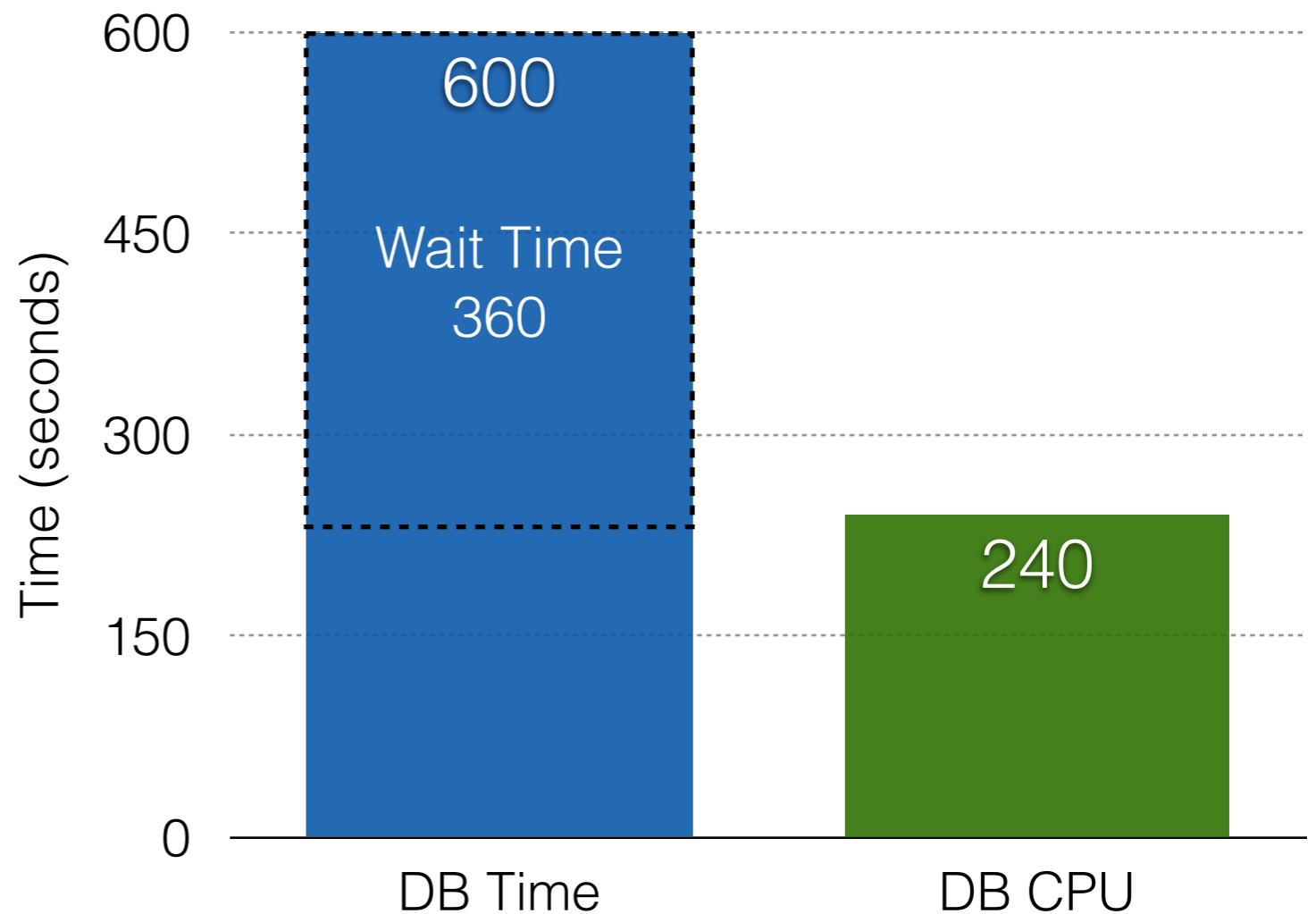
# Load Profile has the gold...

Load Profile	Per Second	Per Transaction	Per Exec	Per Call
~~~~~	-----	-----	-----	-----
DB Time (s) :	5.0	150.4	0.01	3.13
DB CPU (s) :	2.0	60.8	0.00	1.27
Background CPU (s) :	0.0	0.4	0.00	0.00

DB Time = 600 secs  
( 5 \* 120 )

DB CPU = 240 secs  
( 2 \* 120 )

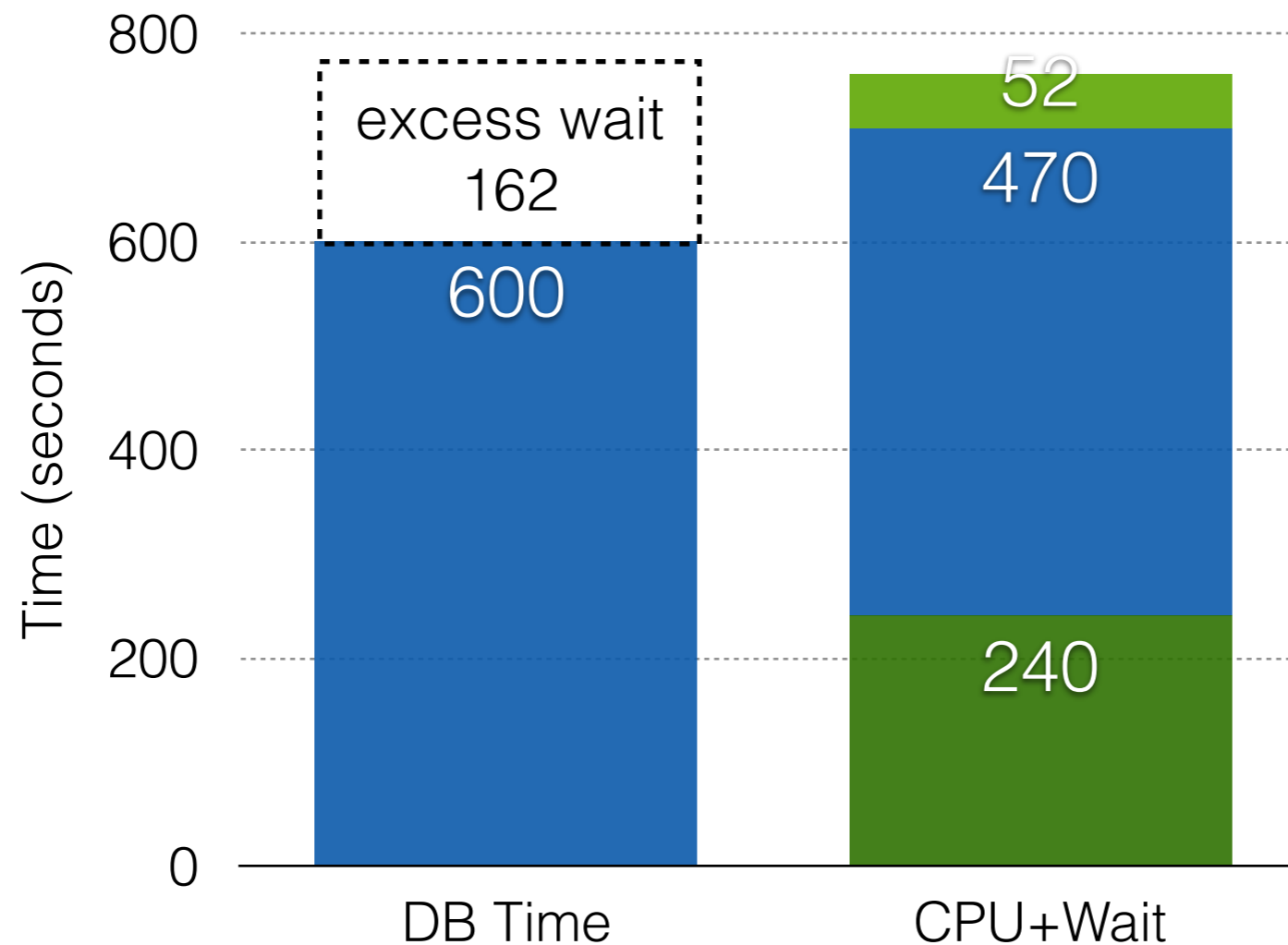
Wait Time = 360 secs  
( 600 - 240 )



# Adding measured wait times...

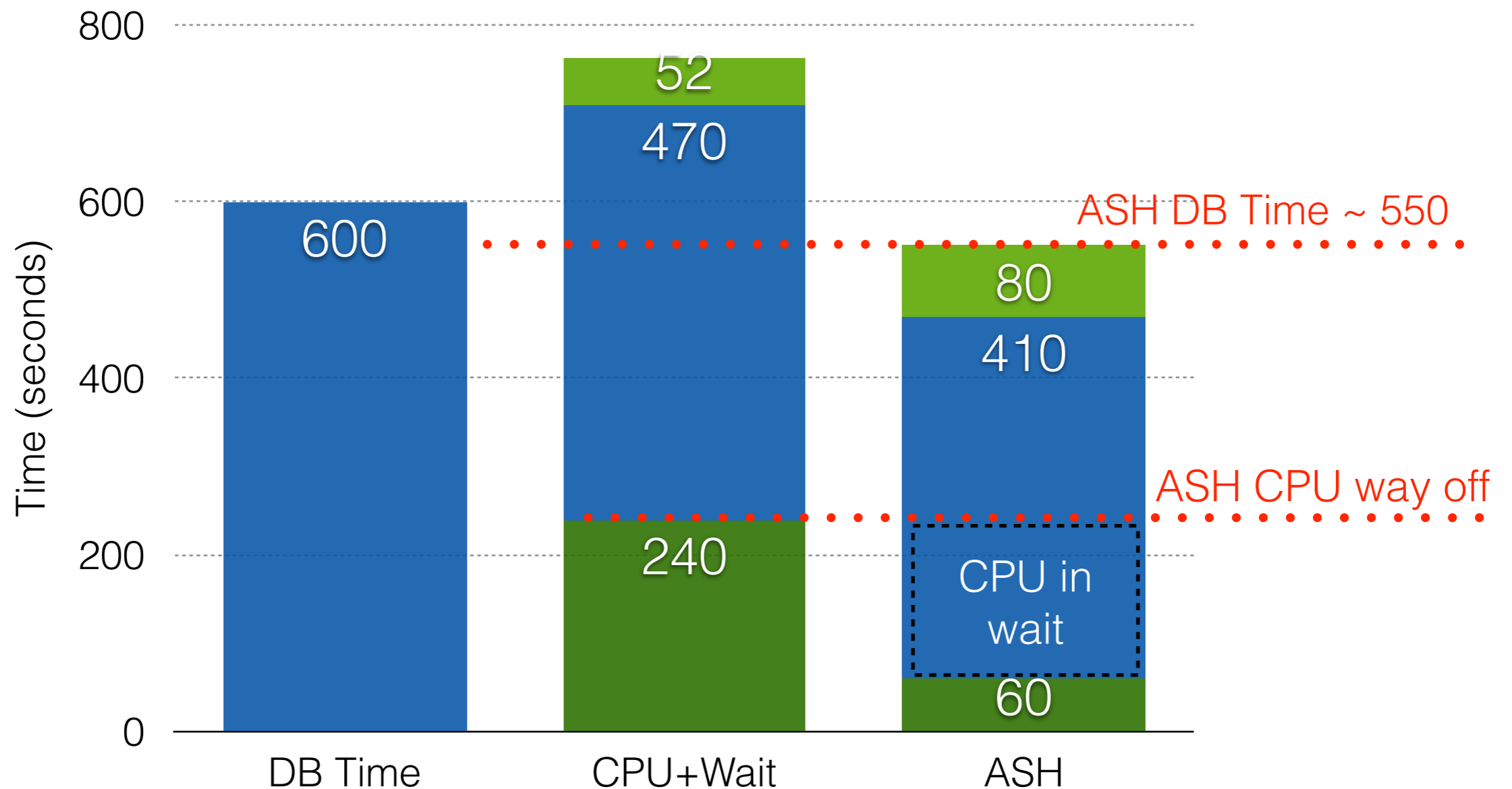
Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Wait Avg (ms)	% DB Wait time	Wait Class
db file parallel read	139,819	469.8	3.36	78.1	User I/O
db file sequential read	201,921	52.4	0.26	8.7	User I/O



# What does ASH say?

Slot Time (Duration)	Slot Count	Event	Event Count	Event %
06:22:16	(2.0 min)	55 db file parallel read	41	74.55
		db file sequential read	8	14.55
		CPU + Wait for CPU	6	10.91



# ASH CPU is copper

- ASH shows “ON CPU” when:
  - session is in a database call and NOT in active wait
- Sessions in Waits that use CPU will never show up in ASH as “ON CPU”
- ASH always conforms to the model:
  - every sample is either ON CPU or WAITING
  - thus (estimated) ASH DB Time = ASH CPU + ASH Wait

# What is our conclusion?

Event	Waits	Total Wait Time (sec)	Wait Avg (ms)	% DB Wait time	Wait Class
db file parallel read	139,819	469.8	3.36	78.1	User I/O

- “db file parallel read” is consuming significant CPU
  - $162 / 470 \sim 34\%$  of the “wait” is actually CPU
  - $162 * 1000 / 139,819 \sim 1.16$  ms/wait
- Is this a bug?
  - Maybe or maybe not, depends on who you ask
  - It does compromise the model (and AWR and ADDM)

# Instrumentation issues and symptoms

<i>Symptom</i>	<i>Possible issue</i>
DB CPU >> ASH CPU (and significant wait time)	CPU used within wait (this was the issue here)
ASH CPU >> DB CPU	System CPU-bound (ASH includes run-queue)
DB Time >> DB CPU + Wait	Un-instrumented wait (in call, not in wait, not on CPU)
DB Time >> ASH DB Time	1. Double-counted DB Time 2. ASH dropped samples

# Some concluding advice

- Don't believe the unbelievable
- Trust DB Time and DB CPU the most
- Be wary of ASH CPU and DB Wait times
- Always get ASH Report with AWR Report
- Don't ponder details if the big picture is not clear



