What Is Oracle DB Time, DB CPU, Wall Time and Non-Idle Wait Time

Posted on 05-Aug-2014 by Craig Shallahamer, craig@orapub.com

If you are into tuning Oracle Database systems, you care about time. And if you care about time, then you need to understand the most important time parameters: what they are, their differences, how they relate to each other and how to use them in your performance tuning work.

The key Oracle Database time parameters are elapsed time, database time (DB Time), non-idle wait time and server process CPU consumption (DB CPU) time.

This first post is pretty basic, yet core fundamental stuff. So in the following two posts I'll introduce elapsed time, add parallelism into the mix and revisit wall time. What initially seems simple can some take very interesting twists!

You probably know that I am all about quantitative Oracle performance analysis. I research, write, teach, and speak about it. I even have an OraPub Online Institute seminar about how to tune your Oracle Database systems from a standard AWR or Statspack report using an Oracle Time Based Analysis (OTBA) framework.

So let's get started!

Wall Time & Run Time

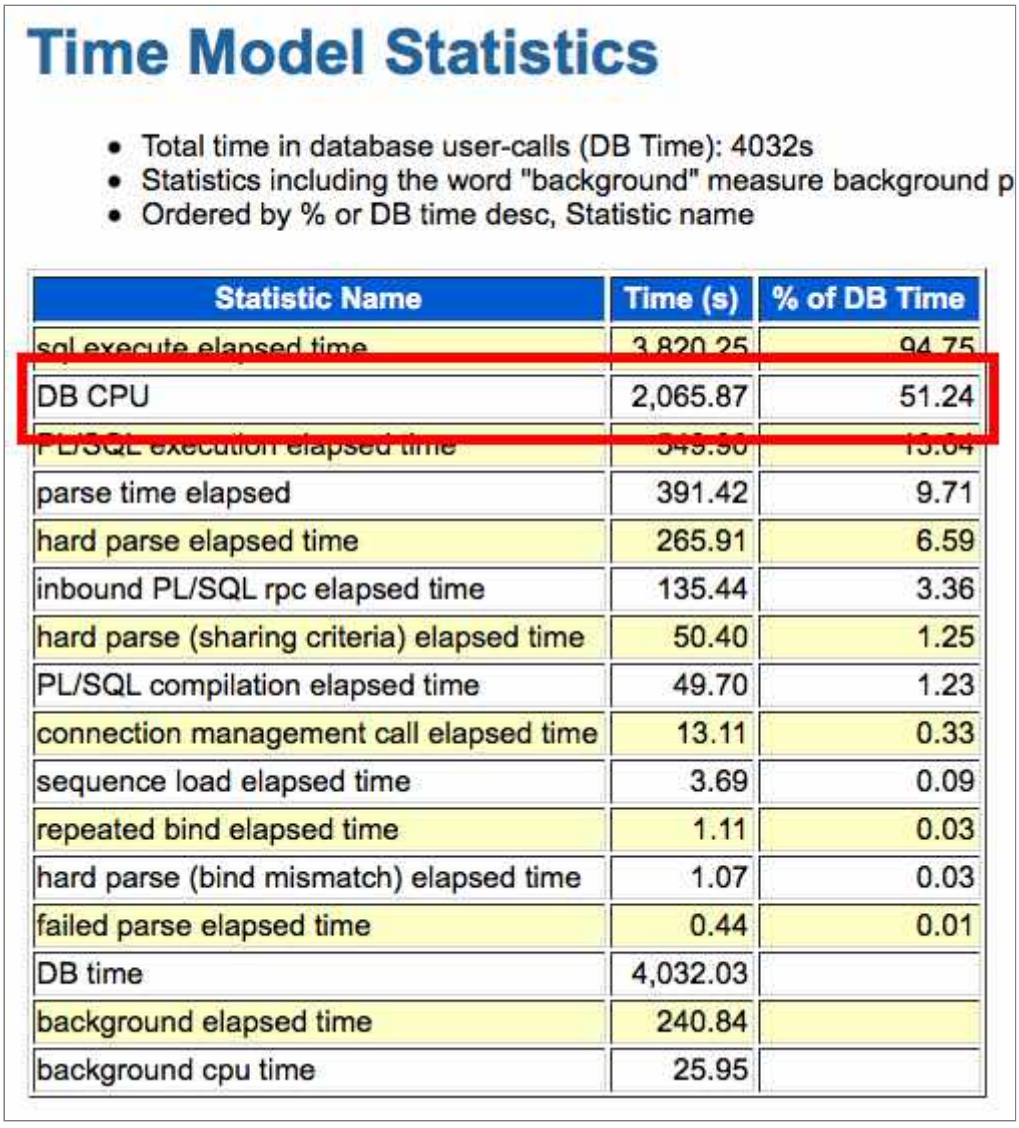
I'll start with Wall Time because that is close (hopefully) to what a user experiences. In fact, if there is no time gap between the Oracle Database and the user, then we can do a little math and figure out what the users are, on average, experiencing. I'll get back to wall time in the next post, where I include elapsed time and parallelism into the equation.

DB CPU

DB CPU is Oracle server/foreground/shadow process CPU consumption. Each Oracle server process gathers its own CPU consumption using the time and/or getrusage C function system call. So unless there is a major screw-up by either the operating system or the Oracle kernel developers, the time will be good... very good. The name DB CPU is taken from the actual statistic name, which is found in both v$sess_time_model and v$sys_time_model.

If you look at any AWR or Statspack report in the "Time Model" section, you will see DB CPU. The value shown will be all server process CPU consumption within the reporting snapshot interval, converted to seconds. (The raw statistic is stored in microseconds.)

Below is an example Time Model Statistics screen shot from a standard AWR report. I've highlighted DB CPU.



Time Model Statistics

- Total time in database user-calls (DB Time): 4032s
- Statistics including the word "background" measure background p
- Ordered by % or DB time desc, Statistic name

Statistic Name	Time (s)	% of DB Time
sql execute elapsed time	3,820.25	94.75
DB CPU	2,065.87	51.24
PL/SQL execution elapsed time	549.90	13.64
parse time elapsed	391.42	9.71
hard parse elapsed time	265.91	6.59
inbound PL/SQL rpc elapsed time	135.44	3.36
hard parse (sharing criteria) elapsed time	50.40	1.25
PL/SQL compilation elapsed time	49.70	1.23
connection management call elapsed time	13.11	0.33
sequence load elapsed time	3.69	0.09
repeated bind elapsed time	1.11	0.03
hard parse (bind mismatch) elapsed time	1.07	0.03
failed parse elapsed time	0.44	0.01
DB time	4,032.03	
background elapsed time	240.84	
background cpu time	25.95	

If you run one of my OraPub System Monitor (OSM) time related tools like ttpctx.sql or rtpctx.sql you see a CPU time statistic. That contains both the DB CPU (i.e., server process) and "background process cpu" statistics. Here's an example.

```
SQL> @ttpctx.sql
Remember: This report must be run twice so both the initial and
final values are available. If no output, press ENTER about 11 times.

Database: prod35                                              31-JUL-14 12:09pm
Report:   ttpctx.sql           OSM by OraPub, Inc.                Page         1
                      Total Time Activity (142 sec interval)

                                                    Avg Time        Time     Wait
Time Component                          % TT    % WT Waited (ms)       (sec) Count(k)
----------------------------------- ------- ------- ----------- ---------- --------
CPU consumption: Oracle SP + BG procs   95.95    0.00       0.000    347.212        0
PX Deq: Slave Session Stats              1.45   35.74       0.113      5.240       47
```

```
library cache: mutex X                0.58   14.26       0.136       2.090       15
PX Deq: Slave Join Frag               0.43   10.57       0.067       1.550       23
PX Deq: Signal ACK EXT                0.29    7.16       0.045       1.050       23
control file parallel write           0.28    7.03      20.600       1.030        0
PX qref latch                         0.27    6.75       0.012       0.990       85
latch free                            0.20    4.91       0.090       0.720        8
log file parallel write               0.16    4.02      12.826       0.590        0
```

Non-Idle Wait Time

When an Oracle process can not consume CPU, it will pause. As an Oracle DBA, we know this as wait time. Sometimes a process waits and it's not a performance problem, so we call this Idle Wait Time. Oracle background processes typically have lots of idle wait time. However, when a user is waiting for sometime to complete and way down deep their Oracle server process is waiting to get perhaps a lock or latch, this is Non-Idle Wait Time. Obviously, when tuning Oracle we care a lot about non-idle wait time.

Below is a simple query showing wait event classifications. In this system there are 119 Idle wait events, so all the rest would be classified as non-idle wait events.

```
SQL> l
  1* select wait_class,count(*) from v$event_name group by wait_class
SQL> /

WAIT_CLASS             COUNT(*)
-------------------- ----------
Concurrency                  34
User I/O                     51
System I/O                   34
Administrative               57
Other                      1123
Scheduler                    10
Configuration                26
Cluster                      57
Application                  17
Queueing                      9
Idle                        119
Network                      28
Commit                        4

13 rows selected.
```

Oracle uses a variety of methods to determine wait time. I have a number of postings and educational content available about this. You'll see them if you do an OraPub or blog search for "time".

When working with non-idle wait time, remember the 80/20 rule. Most of the wait time we care about will be contained with in the largest ("top") two to four wait events. Don't waste YOUR time focusing on the 20%.

Here's an example. In the screen shot below, while not shown the total wait time is 1966 seconds.

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

If you add up the displayed "top" four wait events, their combined wait time is 1857. This is about 95% of all the non-idle wait time. This is a good example demonstrating that most of the wait time is found in the top two to four events.

My OSM toolkit has many wait time related tools. Most start with "sw" for "session wait" but the both ttpctx.sql or rtpctx.sql will contain the non-idle wait time and also CPU consumption. This is a good time to transition into DB Time.

DB Time

DB Time is a time model statistic that is the sum of Oracle process CPU consumption and non-idle wait time. When optimizing Oracle systems we typically focus on reducing "time", though many times database work is also part of the equation. This "time" is essentially DB Time, though sometimes I take control over what I consider idle wait time.

The name DB Time comes from the actual statistic name in both v$sess_time_model and v$sys_time_model.

If you look at any AWR or Statspack report in the "Time Model" section, you will see DB Time.

Time Model Statistics

- Total time in database user-calls (DB Time): 4032s
- Statistics including the word "background" measure background DB time statistic
- Ordered by % or DB time desc, Statistic name

Statistic Name	Time (s)	% of DB Time
sql execute elapsed time	3,820.25	94.75
DB CPU	2,065.87	51.24
PL/SQL execution elapsed time	549.90	13.64
parse time elapsed	391.42	9.71
hard parse elapsed time	265.91	6.59
inbound PL/SQL rpc elapsed time	135.44	3.36
hard parse (sharing criteria) elapsed time	50.40	1.25
PL/SQL compilation elapsed time	49.70	1.23
connection management call elapsed time	13.11	0.33
sequence load elapsed time	3.69	0.09
repeated bind elapsed time	1.11	0.03
hard parse (bind mismatch) elapsed time	1.07	0.03
failed parse elapsed time	0.44	0.01
DB time	4,032.03	
background elapsed time	240.64	
background cpu time	25.95	

The DB time value is technically all server process CPU consumption plus the non-idle wait time within the reporting snapshot interval, converted to seconds. (The raw statistic is stored in microseconds.) Surprisingly, Oracle does not include "background cpu time" in the DB Time statistic. There are both good and not so good reasons the background CPU time is not include, but that's a topic for another posting.

A Little Math

We have enough detail to relate DB Time, DB CPU and non-idle wait time together... using a little math.

DB Time = DB CPU + non_idle_wait_time

And of course,

non_idle_wait_time = DB Time - DB CPU

This is important, because there is no single statistic that shows all the non-idle wait time. This must be derived. Shown above is one way to derive the non-idle wait time. Take a look at the AWR report snippet below.

In the Non-Idle Wait Time section above, I stated that the total non-idle wait time was 1966 seconds. I derived this from the Time Model screen shown above. I simply did:

```
non_idle_wait_time = DB Time - DB CPU
           1966.16 = 4032.03 - 2065.87
```

Coming Up Next

I wanted to keep this post short, which means I left out the more interesting topics. So in the next post I'll merge into the picture elapsed time along with parallelism and revisit wall time. Then in the third post (that's my guess at this point), I'll actually demonstrate this in two different systems.

In future posts I will also discuss why sometimes DB Time is greater than CPU time plus the non-idle wait time. The deeper you go, the more interesting it gets!

Thanks for reading,

Craig.

If you have any questions or comments, feel free to email me directly at craig at orapub.com.