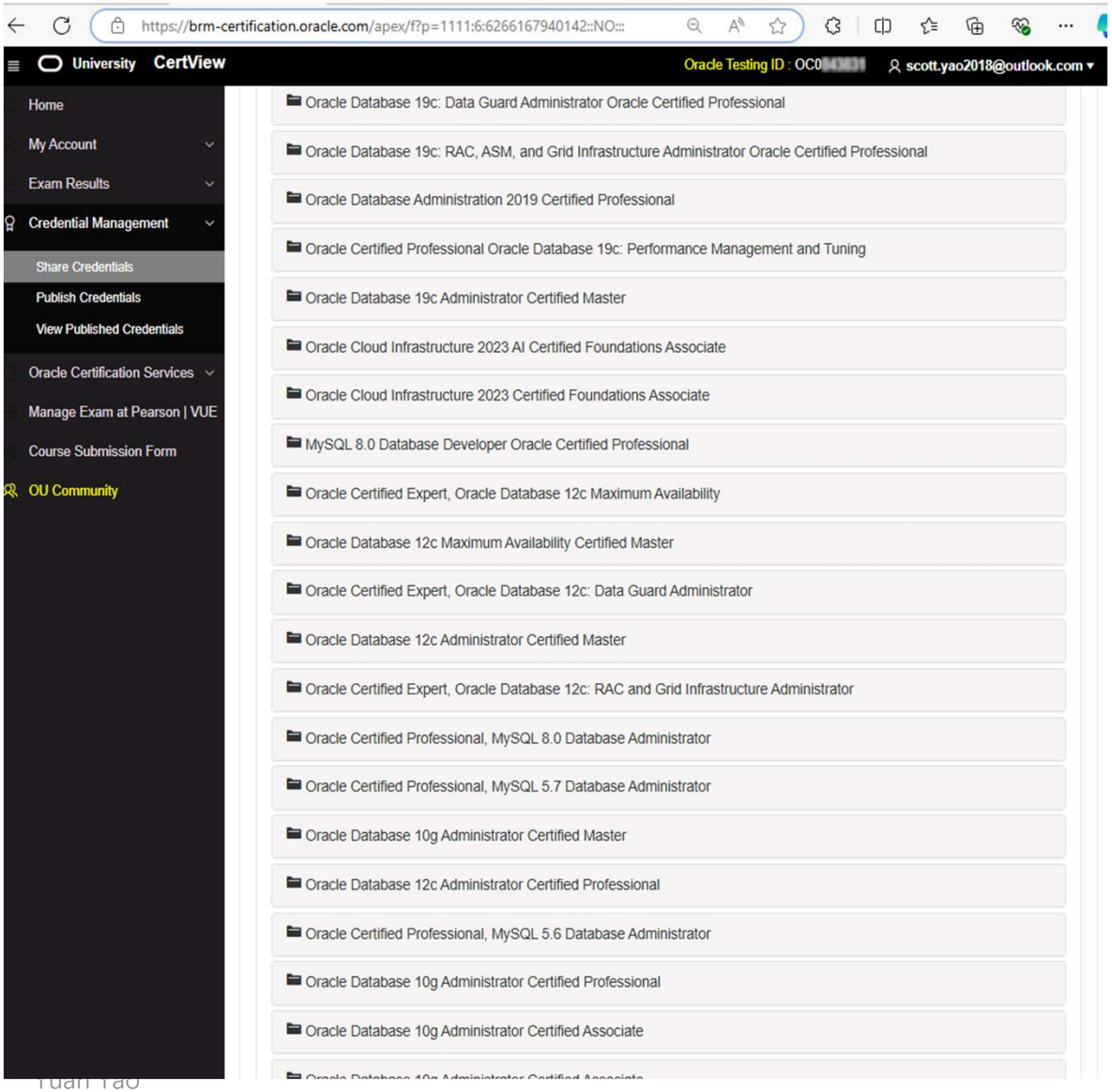
Using AWR and ASH in Tandem for Oracle Database Tuning

Yuan Yao

About Me

- Yuan Yao
- Oracle ACE with over 20 years of experience in Oracle and MySQL databases
- Former database manager at IBM
- Holder of more than 20 Oracle certifications
- Inventor of two computer patents
- Author of the book *MySQL 8.0 Operations and Optimization* (ISBN: 9787302602682)
- Served over 20,000 database clients
- Blog: byte-way.com
- Follow me on X: @YuanOracle



Agenda

1. Why do we need both?
2. Troubleshooting “enq: TX - row lock contention” waits
3. Demystifying a Sudden Performance Degradation

Why Do We Need Both?

Start with AWR to spot trends, then use ASH to pinpoint root causes (macro-to-micro approach)

Oracle ACE

Troubleshooting “enq: TX - row lock contention” waits

Yuan Yao

Spotting the Event in Top Events

Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Avg Wait	% DB time	Wait Class
db file parallel read	316,269	22.8K	71.95ms	93.5	User I/O
DB CPU		1859		7.6	
direct path read	353,746	606.7	1.72ms	2.5	User I/O
enq: TX - row lock contention	24	135.9	5664.17ms	.6	Application
db file sequential read	146,541	55.1	375.93us	.2	User I/O
read by other session	1,411	23.6	16.72ms	.1	User I/O
SQL*Net message to client	1,175,576	.8	657.53ns	.0	Network
log file sync	334	.6	1.89ms	.0	Commit
library cache: mutex X	69	.3	4.52ms	.0	Concurrency
row cache mutex	24	.2	7.14ms	.0	Concurrency

24 Row Lock Waits on a Table

Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Avg Wait	% DB time	Wait Class
db file parallel read	316,269	22.8K	71.95ms	93.5	User I/O
DB CPU		1859		7.6	
direct path read	353,746	606.7	1.72ms	2.5	User I/O
enq: TX - row lock contention	24	135.9	5664.17ms	.6	Application
db file sequential read	146,541	55.1	375.93us	.2	User I/O
read by other session	1,411	23.6	16.72ms	.1	User I/O
SQL*Net message to client	1,175,576	.8	657.53ns	.0	Network
log file sync	334	.6	1.89ms	.0	Commit
library cache: mutex X	69	.3	4.52ms	.0	Concurrency
row cache mutex	24	.2	7.14ms	.0	Concurrency

Segments by Row Lock Waits

- % of Capture shows % of row lock waits for each top segment compared
- with total row lock waits for all segments captured by the Snapshot
- When ** MISSING ** occurs, some of the object attributes may not be available

Owner	Tablespace Name	Object Name	Subobject Name	Obj. Type	Obj#	Dataobj#	Row Lock Waits	% of Capture
YUAN	YUAN	WAREHOUSE		TABLE	73275	73275	24	100.00

A Suspicious 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 99.8% of Total DB Time (s): 24,346
- Captured PL/SQL account for 98.9% of Total DB Time (s): 24,346

$$154.88x(1-8.41\%)=142$$

Elapsed Time (s)	Executions	Elapsed Time per Exec (s)	%Total	%CPU	%IO	SQL Id	SQL Module	SQL Text
8,484.71	12	707.06	34.85	10.18	95.83	29qp10usqkqh0	Sales Rep Query	SELECT TT.ORDER_TOTAL, TT.SALE...
8,482.73	19	446.46	34.84	10.18	95.85	absqp4380420m		BEGIN :1 := orderentry.SalesRe...
8,054.68	18	447.48	33.08	5.54	98.04	awhuaysw8s09b		BEGIN :1 := orderentry.Warehou...
8,054.67	12	671.22	33.08	5.54	98.04	gkxxkqhxubh1a	Warehouse Orders Query	SELECT ORDER_MODE, ORDERS.WARE...
7,484.61	18	415.81	30.74	5.77	97.97	fsu2krpxk3wz7	JDBC Thin Client	BEGIN :1 := orderentry.Warehou...
7,484.59	16	467.79	30.74	5.77	97.97	56pwkjspvmg3h	Warehouse Activity Query	WITH STAGE1 AS (SELECT /*+ mat...
154.88	587,069	0.00	0.64	8.40	0.01	6zw2mftgcgm4c	New Order	UPDATE warehouse SET W_NAME = ...
31.56	220	0.14	0.13	9.16	93.29	3n4tadqgd9b9r	JDBC Thin Client	BEGIN :1 := orderentry.neworde...
26.53	103	0.26	0.11	98.02	1.27	fhf8upax5cxsz		BEGIN sys.dbms_auto_report_int...
24.62	103	0.24	0.10	98.35	0.86	0w26sk6t6gq98		SELECT XMLTYPE(DBMS_REPORT.GET...

A Suspicious 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 99.8% of Total DB Time (s): 24,346
- Captured PL/SQL account for 98.9% of Total DB Time (s): 24,346

154.88x(1-8.41%)=142

Elapsed Time (s)	Executions	Elapsed Time per Exec (s)	%Total	%CPU	%IO	SQL Id	SQL Module	SQL Text
8,484.71	12	707.06	34.85	10.18	95.83	29qp10usqkqh0	Sales Rep Query	SELECT TT.ORDER_TOTAL, TT.SALE...
8,482.73	19	446.46	34.84	10.18	95.85	absqp4380420m		BEGIN :1 := orderentry.SalesRe...
8,054.68	18	447.48	33.08	5.54	98.04	awhuaysw8s09b		BEGIN :1 := orderentry.Warehou...
8,054.67	12	671.22	33.08	5.54	98.04	qkxxkqhxubh1a	Warehouse Orders Query	SELECT ORDER_MODE, ORDERS.WARE...
7,484.61	18	415.81	30.74	5.77	97.97	fsu2krpxk3wz7	JDBC Thin Client	BEGIN :1 := orderentry.Warehou...
7,484.59	16	467.79	30.74	5.77	97.97	56pwkjspvmg3h	Warehouse Activity Query	WITH STAGE1 AS (SELECT /*+ mat...
154.88	587,069	0.00	0.64	8.40	0.01	6zw2mftqcgm4c	New Order	UPDATE warehouse SET W_NAME = ...
31.56	220	0.14	0.13	9.16	93.29	3n4tadqgd9b9r	JDBC Thin Client	BEGIN :1 := orderentry.neworde...
26.53	103	0.26	0.11	98.02	1.27	fhf8upax5cxsz		BEGIN sys.dbms_auto_report_int...
24.62	103	0.24	0.10	98.35	0.86	0w26sk6t6gq98		SELECT XMLTYPE(DBMS_REPORT.GET...

Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Avg Wait	% DB time	Wait Class
db file parallel read	316,269	22.8K	71.95ms	93.5	User I/O
DB CPU		1859		7.6	
direct path read	353,746	606.7	1.72ms	2.5	User I/O
enq: TX - row lock contention	24	135.9	5664.17ms	.6	Application
db file sequential read	146,541	55.1	375.93us	.2	User I/O
read by other session	1,411	23.6	16.72ms	.1	User I/O
SQL*Net message to client	1,175,576	.8	657.53ns	.0	Network
log file sync	334	.6	1.89ms	.0	Commit
library cache: mutex X	69	.3	4.52ms	.0	Concurrency
row cache mutex	24	.2	7.14ms	.0	Concurrency

Fruitless Executions

SQL ordered by Executions

- %CPU - CPU Time as a percentage of Elapsed Time
- %IO - User I/O Time as a percentage of Elapsed Time
- Total Executions: 1,195,251
- Captured SQL account for 99.4% of Total

Executions	Rows Processed	Rows per Exec	Elapsed Time (s)	%CPU	%IO	SQL Id	SQL Module	SQL
587,069	587,069	1.00	17.31	65.9	2.9	3bmjk5pxjjuhp	New Order	SELECT W_ID, W_NA
587,069	- 587,061	= 8 1.00	154.88	8.4	0	6zw2mftgcgm4c	New Order	UPDATE warehouse
3,447	48,272	14.00	4.88	13	89.8	0y1prvxqc2ra9	Browse Products	SELECT PRODUCTS
2,568	11,500	4.48	20.19	7.8	95.2	c13sma6rkr27c	New Order	SELECT PRODUCTS
731	660	0.90	0.93	19.9	94.3	8z3542ffmp562	New Order	SELECT QUANTITY_
682	0	0.00	0.05	100	0	bxpcry2tpc217		select /*+ opt_param(
682	0	0.00	1.83	99	0	f7b069b8zkhvu		SELECT to_number(v
660	660	1.00	3.49	5.3	95.6	f7rxuxzt64k87	New Order	INSERT INTO ORDE
510	510	1.00	1.14	3.8	95.6	5ckxyqfvu60pj	New Order	SELECT CUSTOMER
498	396	0.80	0.71	100	0	fg4skqcja2cyj		SELECT EXTRACTV

Fruitless Executions

SQL ordered by Executions

- %CPU - CPU Time as a percentage of Elapsed Time
- %IO - User I/O Time as a percentage of Elapsed Time
- Total Executions: 1,195,251
- Captured SQL account for 99.4% of Total

Executions	Rows Processed	Rows per Exec	Elapsed Time (s)	%CPU	%IO	SQL Id	SQL Module	SQL Text
587,069	587,069	1.00	17.31	65.9	2.9	3bmjk5pxjjuhp	New Order	SELECT W_ID, W_NAME, W_YTD FRO...
587,069	- 587,061 = 8	1.00	154.88	8.4	0	6zw2mftgcgm4c	New Order	UPDATE warehouse SET W_NAME = ...
3,447	48,272	14.00	4.88	13	89.8	0y1prvxqc2ra9	Browse Products	SELECT PRODUCTS.PRODUCT_ID, PR...
2,568	11,500	4.48	20.19	7.8	95.2	c13sma6rkr27c	New Order	SELECT PRODUCTS.PRODUCT_ID, PR...
731	660	0.90	0.93	19.9	94.3	8z3542ffmp562	New Order	SELECT QUANTITY_ON_HAND FROM P...
682	0	0.00	0.05	100	0	bxpcry2tpc217		select /*+ opt_param('parallel...
682	0	0.00	1.83	99	0	f7b069b8zkhvu		SELECT to_number(value) FROM s...
			3.49	5.3	95.6	f7rxuxzt64k87	New Order	INSERT INTO ORDER_ITEMS (ORDE...
			1.14	3.8	95.6	5ckxyqfvu60pj	New Order	SELECT CUSTOMER_ID, CUST_FIRST...
			0.71	100	0	fg4skqcja2cyj		SELECT EXTRACTVALUE(VALUE(D), ...

Key Instance Activity Stats

- Ordered by statistic name

Statistic	Total	per Second	per Trans
db block changes	55,429	30.62	110.20
execute count	1,195,251	660.25	2,376.24
logons cumulative	16	0.01	0.03
opened cursors cumulative	1,197,474	661.48	2,380.66
parse count (total)	1,182,568	653.25	2,351.03
parse time elapsed	809	0.45	1.61
physical reads	84,677,085	46,775.24	168,344.11
physical writes	5,909	3.26	11.75
redo size	7,959,320	4,396.69	15,823.70
session cursor cache hits	1,191,459	658.16	2,368.71
session logical reads	94,223,044	52,048.39	187,322.16
user calls	1,763,380	974.08	3,505.73
user commits	495	0.27	0.98
user rollbacks	8	0.00	0.02
workarea executions - optimal	8,714	4.81	17.32

enqueue conversions	5,612	3.10	11.16
enqueue deadlocks	8	0.00	0.02
enqueue releases	232,025	128.17	461.28
enqueue requests	232,026	128.17	461.28

Enqueue Activity

- only enqueues with requests are shown
- Enqueue stats gathered prior to 10g should not be compared with 10g data
- ordered by Wait Time desc, Waits desc, Requests desc

Enqueue Type (Request Reason)	Requests	Succ Gets	Failed Gets	Waits	Wt Time (s)	Av Wt Time(ms)
TX-Transaction (row lock contention)	24	- 16 = 8	0	24	138	5,732.96

Pinpointing the Locked Rows

```
SELECT ob.owner,
       ob.object_name,
       dbms_rowid.rowid_create(
          rowid_type => 1,
          object_number => ob.data_object_id,
          relative_fno => sh.current_file#,
          block_number => sh.current_block#,
          row_number => sh.current_row#
       ) AS row_id,
       COUNT(*) AS occurrence_count
  FROM dba_hist_active_sess_history sh
  JOIN dba_objects ob ON sh.current_obj# =
ob.object_id
 WHERE sh.snap_id = 2267
   AND sh.event = 'enq: TX - row lock
contention'
 GROUP BY ob.owner,
          ob.object_name,
          dbms_rowid.rowid_create(
             rowid_type => 1,
             object_number =>
ob.data_object_id,
             relative_fno => sh.current_file#,
             block_number =>
sh.current_block#,
             row_number => sh.current_row#
          )
 ORDER BY occurrence_count DESC;
```

OWNER	OBJECT_NAME	ROW_ID	OCCURRENCE_COUNT
YUAN	WAREHOUSE	AAAR47AAAAAANMhAAA	4
YUAN	WAREHOUSE	AAAR47AAAAABCAZAAA	1
YUAN	WAREHOUSE	AAAR47AAAAAANMgAAA	1
YUAN	WAREHOUSE	AAAR47AAAAAANMuAAA	1
YUAN	WAREHOUSE	AAAR47AAAAABCAPAAA	1
YUAN	WAREHOUSE	AAAR47AAAAABCAtAAA	1

6 rows selected.

Pinpointing the Locked Rows

```
SQL> select * from yuan.warehouse where rowid='AAAR47AAAAAANMhAAA';

      W_ID      W_YTD      W_TAX W_NAME     W_STREET_1           W_STREET_2     W_CITY     W_STATE W_ZIP
---------- ---------- ---------- ---------- -------------------- -------------- ---------- ------- ----------
        28  528875085       .14 3RJSfEol67 2uKUS55GUQi1CYjwr    lvCRszzpQH     vOwNeAGG5ub cl      449311111
```

Demystifying a Sudden Performance Degradation

Yuan Yao

User Complained a Sudden Performance Degradation in Oracle

The Standout Troublemaker

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 100.4% of Total DB Time (s): 1,261
- Captured PL/SQL account for 100.6% of Total DB Time (s): 1,261

Elapsed Time (s)	Executions	Elapsed Time per Exec (s)	%Total	%CPU	%IO	SQL Id	SQL Module	SQL Text
1,252.91	19	65.94	99.35	99.56	39.06	g81cbrq5yamf5	New Order	SELECT ADDRESS_ID, CUSTOMER_I
1,191.94	18	66.22	94.51	99.56	38.80	3n4tadqgd9b9r	JDBC Thin Client	BEGIN :1 := orderentry.neworde...
67.19	1	67.19	5.33	99.54	39.07	ajjqqrmacwv34		BEGIN :1 := orderentry.browsea...
5.36	13	0.41	0.43	99.53	0.24	fhf8upax5cxsz		BEGIN sys.dbms_auto_report_int...
5.18	13	0.40	0.41	99.54	0.13	0w26sk6t6gq98		SELECT XMLTYPE(DBMS_REPORT.G
4.84	13	0.37	0.38	99.62	0.14	0za9fv0j1vgkk		WITH MONITOR_DATA AS (SELECT *.
3.34	1	3.34	0.26	98.69	0.68	3ga4fz007nx5y	sqlplus@dell.scutech (TNS V1-V3)	BEGIN DBMS_WORKLOAD_REPOSIT
0.50	1	0.50	0.04	99.61	0.00	5n48v6nam7jg2		insert into wrh$_instance_reco...
0.49	4	0.12	0.04	99.27	0.00	c9umxngkc3byq		select sql_id, sql_exec_id, db...
0.45	13	0.03	0.04	99.77	0.00	atwuyuvqkf27w		SELECT /*+ OPT_PARAM('_fix_con...

Two Execution Plans

```
Plan hash value: 1286489376

-------------------------------------------------------------------------------
| Id  | Operation          | Name      | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |           |       |       |  2681K(100)|          |
|*  1 |  COUNT STOPKEY     |           |       |       |            |          |
|*  2 |   TABLE ACCESS FULL| ADDRESSES |     2 |   154 |  2681K  (1)| 00:01:45 |
-------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter(ROWNUM<:B1)
   2 - filter("CUSTOMER_ID"=:B2)

SQL_ID g81cbrq5yamf5
--------------------
SELECT ADDRESS_ID, CUSTOMER_ID, DATE_CREATED, HOUSE_NO_OR_NAME,
STREET_NAME, TOWN, COUNTY, COUNTRY, POST_CODE, ZIP_CODE FROM ADDRESSES
WHERE CUSTOMER_ID = :B2 AND ROWNUM < :B1

Plan hash value: 2480532011

-----------------------------------------------------------------------------------------------
| Id  | Operation                             | Name            | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                      |                 |       |       |     6 (100)|          |
|*  1 |  COUNT STOPKEY                        |                 |       |       |            |          |
|   2 |   TABLE ACCESS BY INDEX ROWID BATCHED | ADDRESSES       |     2 |   154 |     6   (0)| 00:00:01 |
|*  3 |    INDEX RANGE SCAN                   | ADDRESS_CUST_IX |     2 |       |     4   (0)| 00:00:01 |
-----------------------------------------------------------------------------------------------
```

The Missing Index

```
SQL>  SELECT index_name, index_type, uniqueness
  2  FROM user_indexes
  3* WHERE table_name = 'ADDRESSES';

   INDEX_NAME    INDEX_TYPE    UNIQUENESS
_______________ _____________ _____________
ADDRESS_PK     NORMAL/REV    UNIQUE
```

A Timeline of the SQL Execution

```
SQL> define top_sql_id='g81cbrq5yamf5'
SQL>
SQL> SELECT b.snap_id,
  2         TO_CHAR(b.end_interval_time, 'HH24:MI') AS snap_time,
  3         a.plan_hash_value,
  4         TRUNC(a.elapsed_time_delta / 1000000 / NULLIF(a.executions_delta, 0), 5) AS avg_elapsed_second
  5  FROM dba_hist_sqlstat a,
  6       dba_hist_snapshot b
  7  WHERE sql_id = '&top_sql_id'
  8    AND a.snap_id (+) = b.snap_id
  9    AND b.begin_interval_time > TRUNC(SYSDATE)
 10* ORDER BY a.snap_id;

   SNAP_ID    SNAP_TIME    PLAN_HASH_VALUE    AVG_ELAPSED_SECOND
_________ ____________ _________________ ____________________
      2932 06:00              2480532011                0.00034
      2933 06:30              2480532011                0.00121
      2934 07:00              2480532011                0.00124
      2935 07:30              2480532011                0.00124
      2936 08:00              2480532011                0.00119
      2937 08:30              2480532011                0.00112
      2938 09:00              2480532011                0.00114
      2939 09:30              2480532011                0.00115
      2940 10:00              2480532011                 0.0011
      2941 10:30              1286489376              104.99351
      2942 11:00              1286489376               63.29527
      2943 11:30              1286489376               64.50286
      2944 12:00              1286489376               63.36064
      2945 12:30              1286489376               63.42192
      2946 13:00              1286489376               64.47289
      2947 13:30              1286489376               64.52588

16 rows selected.
```

Pinpoint the Time

```
SQL>     select
  2          sql_id,
  3          sql_child_number,
  4          to_char(sample_time, 'HH24:MI:SS') track_time,
  5          sql_plan_hash_value curr_sql_plan
  6      from
  7          dba_hist_active_sess_history
  8      where
  9          snap_id=2941
 10          and sql_id = '&top_sql_id'
 11* order by sample_time;
```

```
g81cbrq5yamf5                 0 10:11:20          2480532011
g81cbrq5yamf5                 0 10:11:51          2480532011
g81cbrq5yamf5                 0 10:12:22          2480532011
g81cbrq5yamf5                 0 10:12:32          2480532011
g81cbrq5yamf5                 0 10:14:25          2480532011
g81cbrq5yamf5                 1 10:15:26          1286489376
g81cbrq5yamf5                 1 10:15:26          1286489376
g81cbrq5yamf5                 1 10:15:26          1286489376
g81cbrq5yamf5                 1 10:15:26          1286489376
g81cbrq5yamf5                 1 10:15:26          1286489376
g81cbrq5yamf5                 1 10:15:26          1286489376
g81cbrq5yamf5                 1 10:15:26          1286489376
g81cbrq5yamf5                 1 10:15:26          1286489376
g81cbrq5yamf5                 1 10:15:26          1286489376
g81cbrq5yamf5                 1 10:15:26          1286489376

SQL_ID        SQL_CHILD_NUMBER TRACK_TIME    CURR_SQL_PLAN
------------- ---------------- ------------- -------------
g81cbrq5yamf5                 1 10:15:26          1286489376
```