Oracle Trace Analyzer

A tool for identifying application performance problems.

Presented by

Joe Johnson

Senior Database Administrator

Lands' End

joe.johnson@yahoo.com

Objectives

- Examine traditional tuning techniques, comparing them to Trace Analyzer
- Overview how Trace Analyzer works
- Walk through a Trace Analyzer session
- Review sample Trace Analyzer results
- Describe Trace Analyzer limitations
- Review Trace Analyzer alternatives

Traditional Application Tuning Approaches

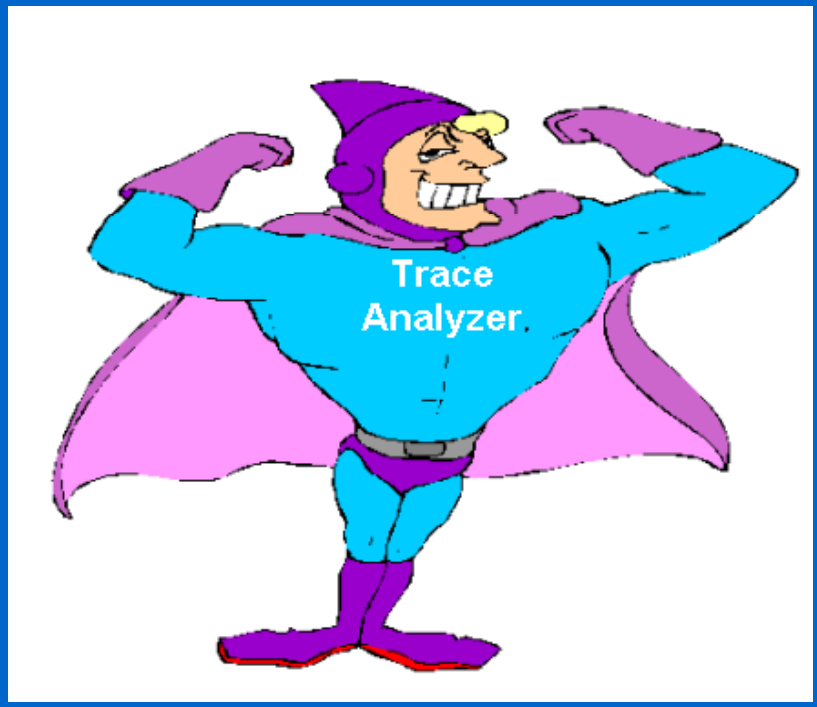
- Ratio analysis
- utlbstat/utlestat reports
- STATSPACK reports
- Checklists
- TKPROF output
- EM GUI tools

Traditional Application Tuning Drawbacks

- Unreliable
- Not reproducible
- Inadequate information
- Too much information
- Too broad in scope

Oracle Trace Analyzer

- Focused
- Reproducible
- Specific information
- Reliable



How Trace Analyzer Works

- Intended for use in Oracle Applications environments
- SQL script based
- Uses 10046 trace event
- Formats raw trace file
- Creates report
- Identifies bottlenecks

Trace Analyzer Setup

- Review Metalink Note 224270.1
- Download TRCA.zip
- Create repository tablespace
- Create repository tables

http://metalink.oracle.com

Sample Trace Analyzer Session

1. Identify session to trace
2. Activate 10046 trace for that session
3. Perform application activity
4. Deactivate 10046 trace for session
5. Confirm trace file creation

Sample Trace Analyzer Session

6. Format trace file using Trace Analyzer
7. Review results
8. Identify bottlenecks
9. Address bottlenecks
10. Regenerate trace file and confirm results

Step 1: Identify Session

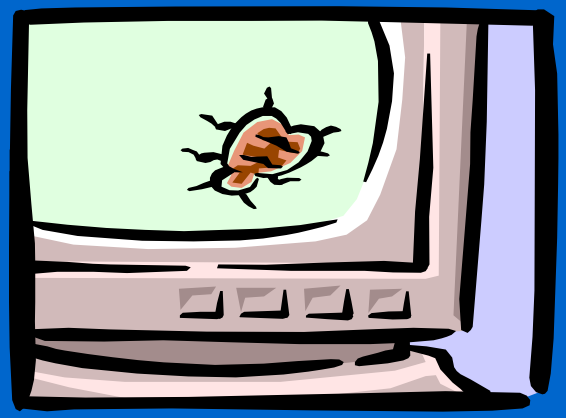
- Identify a session to trace
  - Query V$SESSION
  - Can be difficult with some applications
  - May have to trace multiple sessions
  - May have to examine several trace files

Step 1: Identify Session

```
SQL> select sid, serial#, username, osuser, machine, program
  2  from v$session
  3  where username = 'CEM';
 SID  SERIAL# USERNAME OSUSER   MACHINE  PROGRAM
---- -------- -------- -------- -------- ---------------
   7    35786 CEM      oracle   srvr040  JDBC Thin Client
  19    41959 CEM      oracle   srvr040  JDBC Thin Client
  43    63784 CEM      oracle   srvr040  JDBC Thin Client
...
```

Step 2: Activate Trace

- Modify system settings (if needed)
  - alter system set timed_statistics = true;
  - alter system set max_dump_file_size = 2147483647;
- Activate 10046 trace event
  - Version 8.1.7 +
    - dbms_system.set_ev
    - alter session
  - Version 8.1.6
    - oradebug
    - See Metalink Note 1058210.6



Step 2: Activate Trace

For another user session

sid	serial#	trace event	trace level

```
SQL> exec sys.dbms_system.set_ev(7, 35786,10046, 8,'')
```

Trace levels:

0 – no tracing

1 – Least verbose, bind variables are displayed by name

4 – Level 1 detail, with actual bind variable values displayed

8 – Level 1 detail, plus wait events whose elapsed time > CPU time

12 – Most verbose, combines level 1, 4, and 8 information

Step 2: Activate Trace

For your own session

```
SQL> alter session set events '10046 trace name context forever, level 8';
```

Step 3: Perform Actions

- Using one of the traced sessions:
  - Do the application activities that are performing poorly
  - Allow the activities to run to completion

Step 4: Deactivate Trace

- Turn off 10046 trace event
- Modify system settings (if needed)
  - alter system set timed_statistics = false;
  - alter system set max_dump_file_size = 5242880;

Step 4: Deactivate Trace

For another user session

```
SQL> exec sys.dbms_system.set_ev(7, 35786,10046, 0,'')
```

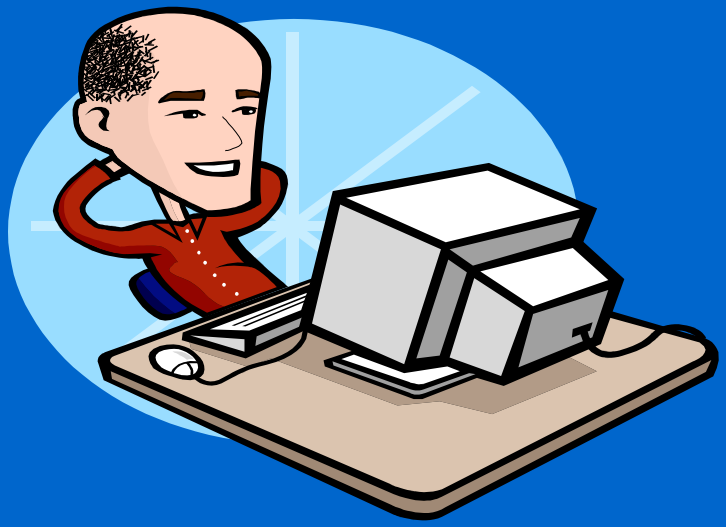
Step 4: Deactivate Trace

For your own session

```
SQL> alter session set events '10046 trace name context off';
```

Step 5: Confirm Trace File

- Confirm creation of trace file
  - Created in user_dump_dest directory
  - Identify by timestamp
  - Identify by process ID



Step 5: Confirm Trace File

```
SQL> select p.spid, s.sid, s.serial#, s.username
  2  from v$session s, v$process p
  3  where s.paddr=p.addr
  4  and s.sid = 7;

SPID                SID    SERIAL# USERNAME
--------- ---------- ---------- --------
213152                7      35786 CEM
```

Step 5: Confirm Trace File

```
$ ls -lat /u01/app/oracle/admin/PROD/udump/*213152*

-rw-rw----   1 oracle dba  2924 Jan 10 13:51 ora_213152_prod.trc
```

Step 6: Format Trace File

- Run TRCANLZR.sql script
- Pass in trace file name
- Use UDUMP directory

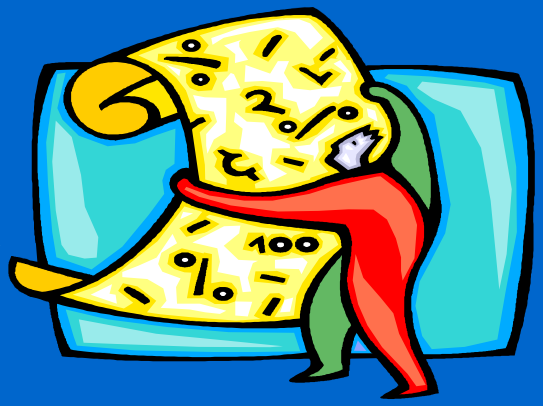
Step 6: Format Trace File

```
SQL> start TRCANLZR.sql UDUMP ora_213152_prod.trc;
```

The trace file formatting process can be a resource intensive operation!

Step 7: Review Trace File

- Created in same directory as TRCANLZR.sql script
- TRCANLZR_ prefixes the formatted file name
- The .LOG suffix is added to the formatted trace file name



Step 7: Review Trace File

```
$ ls -lat /u01/app/oracle/admin/PROD/udump/*213152*

-rw-rw----   1 oracle dba  2924 Jan 10 13:51 ora_213152_prod.trc
-rw-rw----   1 oracle dba  2924 Jan 10 13:51 TRCANLZR_ora_213152_prod.LOG
```

Step 8: Identify Bottlenecks

- Trace Analyzer report summarizes most impactful SQL
  - CPU utilization
  - Elapsed time
  - Non-idle waits
  - Idle waits
- These statements should be examined first

Step 8: Identify Bottlenecks

TOP SQL (SUMMARY OF CPU, ELAPSED AND WAITS PER TOP EXPENSIVE CURSOR)

cursor id	user id	command type	count	cpu	top	elapsed	top	non-idle waits	top	idle waits	top
127...	23..	select.....	27	94.20	1	93.88	1	0.01		0.03	
87....	23..	select.....	3	2.48	2	2.46		0.06		0.00	
159...	23..	select.....	3	2.48	3	2.50	5	0.14		0.00	
64....	23..	select.....	3	2.47	4	2.51	3	0.16		0.00	
243...	23..	select.....	3	2.46	5	2.50		0.11		0.00	
11....	23..	insert.....	419	0.50		2.60	2	2.12	1	0.99	4
263...	23..	select.....	4	2.41		2.50	4	0.10		0.21	
156...	23..	select.....	3	0.00		0.62		0.61	2	0.00	

...

Step 8: Identify Bottlenecks

```
CURSOR_ID:127  LENGTH:256  ADDRESS:3e40ba80
HASH_VALUE:3390289568 OPTIMIZER_GOAL:CHOOSE  USER_ID:23 (CEM)
SELECT Attachment_id
FROM attachments
WHERE AttType = 2 AND Attachment_id NOT IN (SELECT Attachment_id
FROM mess_attach) AND  Attachment_id NOT IN  (SELECT Attachment_id
FROM templ_attach)
AND Attachment_id NOT IN  (SELECT Attachment_id FROM draft_attach)
```

call	count	cpu	elapsed	disk	query	current	rows	misses
Parse	9	0.00	0.01	0	0	0	0	1
Execute	9	0.00	0.00	0	0	0	0	0
Fetch	9	94.20	93.87	0	751344	36	1	0
total	27	94.20	93.88	0	751344	36	1	1

Step 8: Identify Bottlenecks

```
|          Rows Row Source Operation
| ------------ ---------------------------------------------------
|             1 FILTER
|         39665 .TABLE ACCESS FULL ATTACHMENTS
|         39656 .INDEX FULL SCAN PK_MSG_ATTACH
|            55 .INDEX FULL SCAN PK_TMP_ATTACH
|            55 .INDEX FULL SCAN PK_DRFT_ATTACH
Explain Plan
-----------------------------------------------------------
...6 SELECT STATEMENT  (COST=4 CARD=1 BYTES=7 )
...5 .FILTER
...1 ..TABLE ACCESS (FULL) OF 'CEM.ATTACHMENTS'  (COST=4 CARD=1 BYTES=7 )
...2 ..INDEX (FULL SCAN) OF 'CEM.PK_MSG_ATTACH' (UNIQUE)  (COST=26 CARD=1 BYTES=4 )
...3 ..INDEX (FULL SCAN) OF 'CEM.PK_TMP_ATTACH' (UNIQUE)  (COST=26 CARD=1 BYTES=8 )
...4 ..INDEX (FULL SCAN) OF 'CEM.PK_DRFT_ATTACH' (UNIQUE)  (COST=26 CARD=2 BYTES=8 )
```

Step 8: Identify Bottlenecks

OWNER.TABLE_NAME ...owner.index_name	num rows	blocks	sample	last analyzed date
CEM.ATTACHMENTS.....	4180	39	209	2004-11-06 07:46:00
.....cem.pk_attachment...	Missing index statistics			
CEM.DRAFT_ATTACH.....	220	2	11	2004-11-06 07:46:01
.....cem.pk_drft_attach..	Missing index statistics			
CEM.MESS_ATTACH.....	4160	31	208	2004-11-06 07:46:17
.....cem.pk_msg_attach...	Missing index statistics			
CEM.TEMPL_ATTACH.....	0	0	0	2004-11-06 07:46:24
.....cem.pk_tmp_attach...	Missing index statistics			

Step 8: Identify Bottlenecks

```
INDEX_NAME (UNIQUENESS) [indexed columns]
-----------------------------------------------
PK_ATTACHMENT (UNIQUE) [attachment_id]
PK_DRFT_ATTACH (UNIQUE) [messagekey attachment_id]
PK_MSG_ATTACH (UNIQUE) [messagekey attachment_id]
PK_TMP_ATTACH (UNIQUE) [templatekey attachment_id]
```

Step 8: Identify Bottlenecks

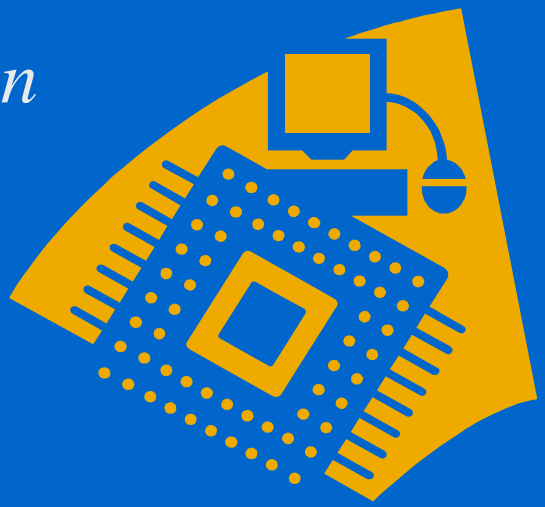
Event waited on	Times Waited	Count Zero Time	Max. Wait	Total Waited	Blocks Accessed
latch free (066 cache buffers chains)..	1	0	0.01	0.01	
SQL*Net message from client (idle).....	36	33	0.01	0.03	
SQL*Net message to client (idle).....	36	36	0.00	0.00	
total.....	73	69	0.01	0.04	0
non-idle waits.....	1	0	0.01	0.01	0
idle waits.....	72	69	0.01	0.03	

Step 8: Identify Bottlenecks

- Most Common Bottlenecks
  - Too many logical I/Os
  - Slow physical I/O
  - Latch and lock contention
  - Oracle Net

Step 8: Identify Bottlenecks

- Too many logical I/Os
  - Results in high DBBC hit ratio
  - Shows up in Trace Analyzer report as high CPU cursors and cursors with *cache buffers chains latch* and *cache buffers LRU chain* waits
  - Usually caused by bad SQL
  - Stresses CPU



Step 8: Identify Bottlenecks

- Slow physical I/O
  - Caused by fewer, larger drives instead of more, smaller drives
  - Inadequate controller caches
  - Shows up in Trace Analyzer report as cursors with *free buffer waits*, *buffer busy waits*, and *log buffer space waits*.

Step 8: Identify Bottlenecks

- Latch and lock contention
  - Shows up in Trace Analyzer Report as cursors with *cache buffers chains latch* waits, *library cache latch* waits, or *cache buffers LRU chain*
  - Locking issues can lead to perceived performance problems

Step 8: Identify Bottlenecks

- Oracle Net
  - Shows up in Trace Analyzer Report as cursors with *SQL*Net message from client (idle)* waits
  - Shows up in Trace Analyzer Report as cursors with *SQL*Net message to client (idle)* waits
  - A certain number of these are normal, excessive waits for these events should be investigated
  - Connection poolers will cause these waits

Step 9: Address Bottlenecks

```
SELECT Attachment_id
FROM attachments
WHERE AttType = 2 AND Attachment_id NOT IN (SELECT Attachment_id
FROM mess_attach) AND  Attachment_id NOT IN  (SELECT Attachment_id
FROM templ_attach)
AND Attachment_id NOT IN  (SELECT Attachment_id FROM draft_attach)

Explain Plan
----------------------------------------------------------
...6 SELECT STATEMENT  (COST=4 CARD=1 BYTES=7 )
...5 .FILTER
...1 ..TABLE ACCESS (FULL) OF 'CEM.ATTACHMENTS'  (COST=4 CARD=1 BYTES=7 )
...2 ..INDEX (FULL SCAN) OF 'CEM.PK_MSG_ATTACH' (UNIQUE)  (COST=26 CARD=1 BYTES=4 )
...3 ..INDEX (FULL SCAN) OF 'CEM.PK_TMP_ATTACH' (UNIQUE)  (COST=26 CARD=1 BYTES=8 )
...4 ..INDEX (FULL SCAN) OF 'CEM.PK_DRFT_ATTACH' (UNIQUE)  (COST=26 CARD=2 BYTES=8 )
----------------------------------------------------------
```

Step 9: Address Bottlenecks

```
create index CEM.LE_ATTTYPE_ATTID_IDX
on CEM.ATTACHMENTS (ATTACHMENT_ID, ATTYPE)
tablespace indx03;

exec sys.dbms_stats.gather_index_stats(ownname=>'CEM',indname =>'LE_ATTTYPE_ATTID_IDX');
```

Step 10: Regenerate Trace File

- Turn on 10046 trace event
- Perform application actions again
- Compare results to first trace file
- Start working on next bottleneck

Step 10: Regenerate Trace File

TOP SQL (SUMMARY OF CPU, ELAPSED AND WAITS PER TOP EXPENSIVE CURSOR)

cursor id	user id	command type	count	cpu top		elapsed top		non-idle waits top		idle waits top	
4.....	23..	select.....	3	2.80	1	3.30	1	0.30	1	0.01	
3.....	23..	select.....	26	0.90	2	1.07	2	0.23	2	0.12	1
5.....	23..	select.....	3	0.02	3	0.02	3	0.00		0.00	
7.....	23..	insert.....	3	0.02	4	0.00		0.00		0.01	5
1.....	23..	set transaction.....	22	0.01	5	0.01	4	0.00	3	0.02	2
9.....	23..	select.....	3	0.00		0.01	5	0.00		0.00	
6.....	23..	set transaction.....	14	0.00		0.00		0.00	4	0.01	3
16....	23..	select.....	3	0.00		0.00		0.00	5	0.00	
10....	23..	select.....	3	0.00		0.00		0.00		0.01	4

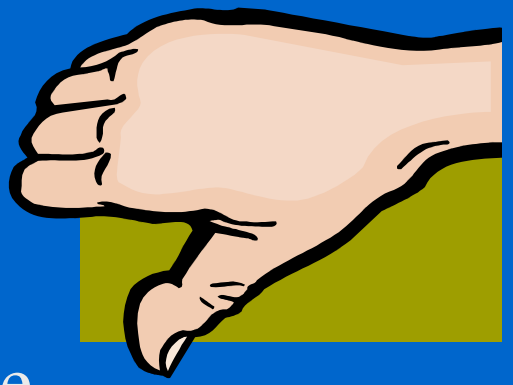
Step 10: Regenerate Trace File

```
CURSOR_ID:4  LENGTH:615  ADDRESS:3fcf18b8  HASH_VALUE:274392568  OPTIMIZER_GOAL:CHOOSE  USER_ID:23 (CEM)
SELECT mess_info.MessageKey,mess_info.M_FromRaw,mess_info.FromEaten,mess_type.TimeDone,
mess_info.M_Subject,mess_info.Source,mess_parsed.TrackingNumber,mess_parsed.TrackRefs,
mess_info.FromRefs,mess_parsed.CommentTag,mess_type.StatusFlags,mess_type.Type,mess_info.M_Reply_To,
mess_parsed.AccessMode,mess_type.Owner,mess_parsed.PksPriority,mess_type.PksOwner_id
FROM mess_info,mess_type,mess_parsed
WHERE mess_info.MessageKey=mess_type.MessageKey AND mess_type.MessageKey=mess_parsed.MessageKey
AND (  Lower(mess_info.M_FromRaw) LIKE Lower('%asmith@acme.com%') )
AND mess_type.Type <> 6 AND mess_type.Type <> 200
```

call	count	cpu	elapsed	disk	query	current	rows	misses
Parse	1	0.00	0.01	0	0	0	0	1
Execute	1	0.00	0.00	0	0	0	0	0
Fetch	1	2.80	3.29	2887	8616	17	2	0
total	3	2.80	3.30	2887	8616	17	2	1

Trace Analyzer Limitations

- Only works with version 8.1.6 and above
- No support for Shared Server environments
- Adds some overhead while *generating* trace file
- Can add significant overhead while *formatting* trace file
- Repository tables have to be built in application schema



Things To Remember

- Collect traces in good times and bad
- Save trace files for future reference
- Use traces files during development

Other 10046 Trace File Utilities

Free Utilities:

- AppsDBA Resource Profiler (Perl based)
  - http://home.comcast.net/~arivenes/utilities_resource.htm
- Simple Profilier (requires HTML DB)
  - http://www.niall.litchfield.dial.pipex.com/SimpleProfiler/SimpleProfiler.html
- Analyzer
  - http://www.oraperf.com/

Commercial Utility:

- Hotsos Profiler
  - http://www.hotsos.com

Other Tuning Utilities

- Oracle 9i EM Top SQL, Oracle Expert, SQL Analyze, Top Sessions, and Index Tuning Wizard
- Oracle 10g Database Control
- Quest Central SQL Tuning Lab
- BMC SQL Explorer for Oracle
- Veritas i^3 for Oracle

References

White Papers

- *Oracle 9.2 Event 10046 Segment-level Statistics* (Rivenes, 2003). See http://www.orapub.com/cgi/exodus.cgi?p1=sub&p2=abs148.

- *Oracle System Performance Analysis Using Oracle Event 10046* (Millsap and Holt, 2002). See http://www.nyoug.org/hotsos_perf.pdf.

- *Why a 99%+ Database Buffer Cache Hit Ratio is Not OK* (Cary Millsap, 2001). See http://www.hotsos.com/e-library/abstract.php?id=6.

- *Yet Another Performance Profiling Method* (Kolk et. al., 1996). See http://oraperf.com/whitepapers.html.

References

Books

- *Expert Oracle One-on-one,* Tom Kyte (Wrox).
- *Optimizing Oracle Performance*, Millsap and Holt (O'Reilly).
- *Oracle8i Internal Services for Waits, Latches, Locks, and Memory,* Steve Adams (O'Reilly).

Presentations

- *OraPerf.com: Real Life Performance Data,* Oracle Open World Session #1493 (Kolk, 2004). See https://www.openworld2004.com/published/1493/1493_kolk.ppt

Thanks!

Contact information:

Joe Johnson

Senior Database Administrator

Lands' End

joe.johnson@yahoo.com

Case Study #2

- Application was having sporadic performance problems
- Interface to use is an Excel spreadsheet
- Created 10046 Trace files
- Two statements had the highest wait times

Case Study #2

```
WAITS FOR ALL NON-RECURSIVE STATEMENTS FOR USER 24 (SRCADMIN)

INSERT INTO TempTableNames(TempTableName) VALUES('Crosssum401374460')
Event                                  Times        Count        Max.     Total   Blocks
waited on                              Waited   Zero Time        Wait    Waited Accessed
---------------------------------- --------- --------- --------- -------- --------
log file sync.....................         1         0      0.00     0.00
SQL*Net message from client (idle).         5         0     51.96    51.97
SQL*Net message to client (idle)...         5         0      0.00     0.00
---------------------------------- --------- --------- --------- -------- --------
total.............................        11         0     51.96    51.97        0
```

Case Study #2

WAITS FOR ALL NON-RECURSIVE STATEMENTS FOR USER 24 (SRCADMIN)

INSERT INTO TempTableNames(TempTableName) VALUES('Crosssum389180482')

Event waited on	Times Waited	Count Zero Time	Max. Wait	Total Waited	Blocks Accessed
log file sync.....	1	0	0.00	0.00	
SQL*Net message from client (idle).	5	0	50.92	50.93	
SQL*Net message to client (idle)...	5	0	0.00	0.00	
total.....	11	0	50.92	50.93	0

Case Study #2

- Performance issue was in the Oracle Net layer
- Oracle Names was being used for database name resolution
- Changing to a local tnsnames.ora instead resolved the `SQL*Net message from client` waits