SRDC - EBS Applications Performance (Doc ID 1920979.1)

APPLIES TO:

Oracle EBS Applications Performance - Version 11.5.10.2 to 12.2 [Release 11.5.10 to 12.2]
Information in this document applies to any platform.

MAIN CONTENT

Applies To

Product:

- Oracle EBS Applications Performance - Version 11.5.10.2 to 12.2

Privileges Required:

- Database Administrator - access to the Database is required in order to perform this collection.

System Impact:

- None - following the collection instructions given will have no impact on the system.

What is being collected and why?

The process outlined belo ll guide the user through collecting diagnostic data, which will help Support narrow down the cause of your problem.

EBS Applications Performance Data Collection Items

I. Required Configuration: Requirements before initiating a tracing activity

II. Database Level Trace: Instance wide performance issues

III. Concurrent Program Trace: Report overview of where time is spent (formatting, retrieving data, etc)

IV. PL/SQL Profiler Report from Forms: Tracing activities of a specific form

V. PL/SQL Profiler Report from Self Service: Tracing activities from OA Fr

I. Create a TKPROF with an Explain Plan: Create a readable trace file

VII. Display Cursor Report: Runtime execution plan of any cursor loaded in the cursor cache via sql id

VIII. SQL Monitor Report: 11g only, actual executions and row counts for each sql plan line

IX. PL/SQL Hierarchical Profiler Report: Time analysis of server side PL/SQL steps

Safe Harbor Statement: Oracle will use this information to help diagnose the cause of the issue. In many cases this set of information will be sufficient for diagnosis - in some cases additional specific diagnostic data may be required at a later stage.

Action Plan

Be sure to review Note 1985929.1: Additional Applications Performance Tr leshooting Tips to be Used with 'SRDC - EBS Applications Performance (Doc ID 1920979.1)' for other steps to be used in conjunction with this note.

I. Required Configuration
The following steps must be performed by the DBA on the database server before collecting trace files.

1. Set TIMED_STATISTICS to TRUE
For performance issues, make sure TIMED_STATISTICS is turned on, before attempting to generate the trace.

Set the following in the init.ora file:

- TIMED_STATISTICS=TRUE

OR in SQL*Plus:

- ALTER SYSTEM SET TIMED_STATISTICS=TRUE;

2. Set the location of the trace output.
Set the following in the init.ora file:

- USER_DUMP_DEST = <preferred directory for the trace output>

3. Create the PLAN_TABLE to hold the output of the explain plan. Run the SQL script called UTLXPLAN.SQL to create this in the apps schema.
This script is usually in $ORACLE_HOME/rdbms/admin.

4. If the init.ora file has been updated, you must shut down and restart the database before the changes will take effect.

II. Database Level Trace
Before performing the following steps, please review section **I. Required Configuration** above.

This will enable trace on all processes that are running on the instance. This step should be performed by the DBA.

1. Enable SQL Trace with Waits:
SQL> alter system set events '10046 trace name context forever, level 8';

2. Perform the steps to reproduce the issue.

3. Disable SQL Trace:
SQL> alter system set events '10046 trace name context off';

4. Follow steps outlined in section **VI. Create a TKPROF with an Explain Plan** to produce a readable tr le with an explain plan. Upload the output file.

5. Provide the last 400 lines of the database alert.log file. Steps to identify the location of the database alert log file are outlined in note 'How to Find the alert.log File (11g and Later) (Doc ID 438148.1)'. Issue the following command and upload the alert_400.log:
$ tail -n 400 alert.log > alert_400.log

6. Review note 'bde_chk_cbo.sql - EBS initialization parameters - Healthcheck (Doc ID 174605.1)'. Upload the output of the script. This will help the Oracle Support Engineer verify that the database init.ora is intact.

7. Run the utility outlined in note 'Utility /Script To Check The Techstack Component Versions (Forms, Http Server, JDK, Fr tabase, etc) (Doc ID 601736.1)'. Upload the output of the utility.

III. Concurrent Program Trace
Before performing the following steps, please review section **I. Required Configuration** above.

Some of the data to be collected is outlined in the '*Steps to enable trace for Concurrent Requests: Request Level*' section of note 1121043.1: 'Collecting Diagnostic Data for Performance Issues in Oracle E-Business Suite (Doc ID 1121043.1)'. Additional data may be collected if relevant. The minimum data required may be collected by following these steps:

1. Log in to the environment from which you want to trace.
2. Set the value for the profile option Concurrent: Allow Debugging to "Yes" at user level.
3. Choose the appropriate responsibility and concurrent program to be executed.
4. Click on the "Debug Options" button.
5. Enable tracing by selecting the "SQL Trace" Check box, and choose the tr level from the DropDown list. Confirm your selection by clicking the "OK" button.
6. Submit the concurrent program.
7. If there are more than one concurrent requests to be traced, perform Steps 1-6 for each individual concurrent request.
8. Write down the request_id of your concurrent program job.
9. Once the concurrent request completes, retrieve the raw tr file using the request_id (fr tep 7) or the tracefile_identifier (set by default to the userid).
10. Follow steps outlined in section **VI. Create a TKPROF with an Explain Plan** to produce a readable trace file with an explain plan. Upload the output file.
11. Review note 'bde_chk_cbo.sql - EBS initialization parameters - Healthcheck (Doc ID 174605.1)'. Upload the output of the script. This will help the Oracle Support Engineer verify that the database init.ora is intact.
12. Review note 'Concurrent Processing - CP Analyzer for E-Business Suite (Doc ID 1411723.1)'. This is not required, but highly recommended as it provides Oracle Support Engineers a good view of the Concurrent Manager Sub System.
13. Run the utility outlined in note 'Utility /Script To Check The Techstack Component Versions (Forms, Http Server, JDK, Fr tabase, etc) (Doc ID 601736.1)'. Upload the output of the utility.

IV. PL/SQL Profiler Report from Forms

Before performing the following steps, please review section **I. Required Configuration** above.

Toggle trace on/off on the form to tr lication functions.

1. Sign off and sign on again before proceeding further. Some form queries may be cached and may not be executed on subsequent visits to the form. Starting a "fresh" session will ensure that you do not run into that.
2. Set the profile option “Utilities:Diagnostics“ (internal name DIAGNOSTICS) user-level value to “Yes”.
3. Choose the relevant responsibility for the form and navigate to the form that will be traced.
4. Enable SQL Trace using the menu option : Help > Diagnostics > Trace. Check the “PL/SQL Profiling” box.
5. Execute the required actions on the Form.
6. When complete, disable SQL Tr lp > Diagnostics > Trace > Uncheck the “PL/SQL Profiling” box.
7. If the "PL/SQL Profiler Report" concurrent request is not started automatically then do the following:

   A. Identify the run id for the actions traced above using the SQL:
   SQL> SELECT runid, related_run, TO_CHAR(run_date, 'DD-MON-YYYY HH24:MI:SS'), run_comment, run_comment1 FROM plsql_profiler_runs;

   B. Submit a request for concurrent program “PL/SQL Profiler Report” from the “System Administrator” responsibility.
   Enter the parameters Run ID and Purge Profiler Data = “No” (default is “Yes“).

8. When the "PL/SQL Profiler Report" concurrent request is complete, click on the “View Output” button to see the PL/SQL Profiler report. Upload the report to the Service Request.
9. Run the utility outlined in note 'Utility /Script To Check The Techstack Component Versions (Forms, Http Server, JDK, Fr tabase, etc) (Doc ID 601736.1)'. Upload the output of the utility.

V. PL/SQL Profiler Report from Self Service

Before performing the following steps, please review section **I. Required Configuration** above.

This is also sometimes referred to as OAF (Or le Applications Framework) after the technology framework used on these applications. To obtain a PL/SQL Profiler report for Self Service (OAF) pages:

1. Log in to the environment that will be traced.
2. Set the profile option “FND: Diagnostics“ (internal name FND_DIAGNOSTICS) user-level value to “Yes”.
3. Choose the relevant responsibility for the HTML-based application.

4. To enable SQL Trace click the “Diagnostics” icon at the top right of the page > Set Trace Level > Go > Enable PL/SQL Profiler > Save.
5. Select “Home” and then execute the required actions.
6. When complete, disable the tr licking on the “Diagnostics” icon > Set Trace Level > Go > Disable PL/SQL Profiler > Save
7. The “Trace Ids” will be provided on the left hand side of the screen. This is relevant for SQL Traces (not PL/SQL Profiler)
8. Exit the HTML Application and log out.
9. If the "PL/SQL Profiler Report" concurrent request is not started automatically then do the following:

A. Identify the run id for the actions traced above using the SQL:

```
SQL> SELECT runid, related_run, TO_CHAR(run_date, 'DD-MON-YYYY HH24:MI:SS'),
run_comment, run_comment1 FROM plsql_profiler_runs;
```

B. Submit a request for concurrent program “PL/SQL Profiler Report” from the “System Administrator” responsibility.
Enter the parameters Run ID and Purge Profiler Data = “No” (default is “Yes“).

10. When the "PL/SQL Profiler Report" concurrent request is complete, click on the “View Output” button to see the PL/SQL Profiler report. Upload the report to the Service Request.
11. Run the utility outlined in note 'Utility /Script To Check The Techstack Component Versions (Forms, Http Server, JDK, Framework, Database, etc) (Doc ID 601736.1)'. Upload the output of the utility.
12. Upload the $CONTEXT_FILE to the Service Request. This will reflect configurations like the number of JVMs and the Java Heap Size which is key to troubleshooting Application Framework performance issues.
13. If the issue appears to be instance wide, consider following steps outlined in section **II. Database Level Trace** and providing the output as well.

VI. Create a TKPROF with an Explain Plan

The TKPROF program converts Oracle trace files into a more readable form. Perform the following steps:

1. Find the trace directory by locating the user_dump_dest. Log into SQL*Plus as the apps user and issue the following command:

```
SQL> select value from V$PARAMETER where name = 'user_dump_dest';
```

2. Find the trace file for your process. Navigate the directory located in step 1 and identify any file(s) with a .trc extension. You may use this command:

```
$ ls -ltr
```

3. Run tkprof command with an explain plan. Navigate to a directory in which you have write privileges and issue the following command:

```
$ tkprof <full path to trace file> <output file name> explain=<apps username/apps password>
```

VII. Display Cursor Report

The report can be produced as follows:

Obtain the SQL ID via SQL:

```
set pages 0
set lines 300
set LONG 10000
set LONGCHUNKSIZE 10000
spool <report_name>.txt
SELECT * FROM TABLE(dbms_xplan.display_cursor('<sql_id>', NULL, 'ALL +ALLSTATS'));
spool off;
```

If querying v$SQL, use the following query:

```
SQL> select sql_id, hash_value, SUBSTR(sql_text,1, 80)
  from v$sql
  where sql_fulltext LIKE '%<part of SQL text>%';
```

Alternatively, the sql_text column can be used to search on the first 1000 characters. The UPPER function could also be used on column sql_fulltext or sql_text if the exact case of some of the SQL is unknown. See My Oracle Support Document "How to Determine the SQL_ID for a SQL Statement (Document 1627387.1)".

VIII. SQL Monitor Report (DB release 11g and higher)
Please produce a SQL Monitor report for SQL ID <sql_id>. The report can be produced by running the following.
NOTE: HTML format is preferred.

```
set trimspool on
set trim on
set pages 0
set long 10000000
set longchunksize 10000000
set linesize 200
set termout off
spool sql_monitor_for_<sql_id>.htm
variable my_rept CLOB;
BEGIN
   :my_rept := dbms_sqltune.report_sql_monitor(sql_id => '<sql_id>', report_level => 'ALL', type => 'HTML');
END;
/
print :my_rept
spool off;
set termout on
```

SQL Monitor reports are also available from Oracle Enterprise Manager.

Obtain the SQL ID above

If querying v$SQL, use the following query:

```
SELECT sql_id, hash_value, SUBSTR(sql_text,1, 80)
FROM v$sql
WHERE sql_fulltext LIKE '%<part of SQL text>%';
```

Alternatively, the sql_text column can be used to search on the first 1000 characters. The UPPER function could also be used on column sql_fulltext or sql_text if the exact case of some of the SQL is unknown. See My Oracle Support Document "How to Determine the SQL_ID for a SQL Statement (Document 1627387.1)".

IX. PL/SQL Hierarchical Profiler Report
To enable the PL/SQL Hierarchical profiler for an E-Business Suite session using the Profile Option method:

1. Log in to the environment that will be traced.
2. If the PL/SQL Hierarchical profiler is being enabled for the current (logged in) user account then navigate to the Profile > Personal window, otherwise navigate to the Profile > System window.
3. Query the profile “Initialization SQL Statement - Custom”(internal name FND_INIT_SQL).
4. Enter the following as the profile value and save.

```
BEGIN dbms_hprof.start_profiling('<profiling directory>','<raw_output_filename>'); END;
```
Note: If there is existi the profile then the above should be merged with the existi

. Log in (again) to the application using the user account being traced. This is necessary because the Initialization SQL Statement is only executed at login.
6. Execute the actions to be profiled.
7. Exit the application and log out.
8. Navigate to the profile form again, remove the SQL above (ensuring any original SQL r) from the “Initialization SQL Statement - Custom” profile and save.
9. Obtain the raw output file from the file location specified in dbms_hprof.start_profiling above.
10. Generate PL/SQL hierarchical profiler HTML reports from the raw output file using:

```
plshprof -output <html_root_filename> <profiler_output_filename>
  Where
      <html_root_filename> is the name of the root HTML file to be created.
      <profiler_output_filename> is the name of a raw profiler output file.
```

11. This will create a set of HTML r fr TML file. Please provide all these files.

Didn't find what you are looking for?