Best Practices for Oracle Business Intelligence Discoverer Plus OLAP Version 10.1.2.2

An Oracle White Paper
January 2007

Best Practices for Oracle Business Intelligence Discoverer Plus OLAP Version 10.1.2.2

Best Practices for Oracle Business Intelligence Discoverer Plus OLAP Version 10.1.2.2

INTRODUCTION

Oracle Business Intelligence Discoverer Plus OLAP (Discoverer Plus OLAP) is a powerful Business Intelligence tool that enables ad hoc query and analysis of data stored in the Oracle Database OLAP option. An intuitive wizard allows the user to quickly create queries to get insight into their data. The results of this analysis can be published on the Internet or corporate intranet using Discoverer Viewer or by taking advantage of Discoverer's native integration with Oracle Portal. This white paper describes best practices for query creation and worksheet design to optimize performance and make the process of analysis easier for end users. Also included are sections that contain tips for optimum system configuration.

This document applies to Discoverer Plus OLAP 10g Release 2 (10.1.2.2) and is intended for end users, administrators, application implementers, and Oracle database administrators.

USING DISCOVERER PLUS OLAP

As an end user designing workbooks and worksheets in Discoverer Plus OLAP, you can use various techniques to optimize the execution of these workbooks. The following sections describe what you can do.

Designing the Organization of Workbooks and Worksheets

The following sections describe best practices to consider when designing the organization of workbooks and worksheets.

Share a New Workbook

If you plan to share a new workbook with others, follow these steps:

1. Save the new workbook in the Shared/<*username*> folder as opposed to the Users/<*username*> folder. The Shared/<*username*> folder is configured such that you can provide access to other users.
2. Change the access privileges for other users using the Manage Catalog dialog to open the Properties dialog of the workbook.
3. Ensure that saved selections or calculations that this shared workbook uses exist in the Shared/<*username*> folder. Move the objects there if they

are located somewhere else; do not copy them. Give other users the same access to these objects as for the workbook. If you cannot change the privileges, ask the Discoverer Administrator to change the privileges for you.

Note: The sharing of saved selections and calculations is a feature that is likely to be enhanced in a future release of Discoverer Plus OLAP.

Share an Existing Workbook

If you plan to share an existing workbook with others, then follow these steps:

1. Copy the workbook to the Shared/<*username*> folder, if it is not already located there.
2. Change the access privileges for other users using the Manage Catalog dialog to open the Properties dialog of the workbook.
3. Ensure that saved selections or calculations that this shared workbook uses exist in the Shared/<*username*> folder. Move the objects there if they are located somewhere else; do not copy them. Give other users the same access to these objects as for the workbook. If you cannot change the privileges, ask the Discoverer Administrator to change the privileges for you.

   Note: The sharing of saved selections and calculations is a feature that is likely to be enhanced in a future release of Discoverer Plus OLAP.

Reduce the Number of Crosstab Columns

In general, Discoverer Plus OLAP optimizes access to the multidimensional result set when there are fewer crosstab columns and more crosstab rows. Consider putting additional dimension selections on the page edge, or hiding dimensions that are not critical for your analysis. Hidden dimensions appear on the Hidden Dimensions panel along with the members of the dimensions to use in the evaluation of the query.

Keep the Number of Worksheets in a Workbook to a Minimum

Discoverer Plus OLAP workbooks are a convenient way to organize multiple worksheets. When an end user enters a worksheet for the first time during a Discoverer for OLAP session, the underlying query is executed and the corresponding RDBMS resources are cached while the workbook is open. This can affect the overall system performance if there are many concurrent users working with large workbooks that contain many worksheets. Therefore, keep the number of worksheets in a workbook to a minimum. In particular, review any saved workbooks and determine whether they contain obsolete worksheets that you can delete.

Maximize the Worksheet Window

Discoverer Plus OLAP supports viewing multiple worksheets at once within separate sub-windows. This can be very useful, but it requires that *all* underlying queries be executed when the parent workbook is opened. If the workbook contains many worksheets with long-running queries, keep the worksheet window maximized so that only the current worksheet query is executed when the workbook is reopened.

Understand Options for Creating Saved Selections

When creating a saved selection, you can use one of two options in the Save Selection As dialog:

1. When you choose the **Save Steps** option, the saved selection is reevaluated every time it is referenced in a worksheet. This very powerful feature can affect performance if the dimension selection consists of many complex steps such as data-based exceptions, top *n* values, and so on.
2. When you choose the **Save Members** option, the saved selection is evaluated only at the time of the save operation. When this saved selection is referenced in a worksheet, it always refers to the same fixed set of dimension members. This option is less flexible (for example, dimension members added to the database later on will not be included), but it will sometimes result in faster execution times.

Use the Exit Command to Exit a Discoverer Plus OLAP Session

When shutting down a Discoverer Plus OLAP session, encourage end users to always use the **Exit** command from the File menu. The use of this command ensures that the RDBMS connection is closed in a timely manner and that shared RDBMS resources are released. Simply closing the browser window without shutting down the application will cause RDBMS resources to be held until the underlying connection times out.

Capture Diagnostic Information

If you encounter errors while using Discoverer Plus OLAP, then capture any error messages that are displayed. In addition, launch the Diagnose Connection dialog from the Help menu in order to capture details about the Discoverer Plus OLAP environment. Options are available to run Quick and Extensive diagnoses and to save the results to a file. Refer to the "How to inspect the database schema and connection" Help topic for additional information.

Designing Individual Worksheets

The following sections describe best practices to consider when designing individual worksheets.

Use Hierarchical Conditions in Place of Very Long Lists of Dimension Members

If you find yourself expanding the Members tab in the Discoverer Plus OLAP Navigator and dragging all the members of a dimension (or an entire branch of the hierarchy sub-tree), then consider using the Query Wizard to create a hierarchical condition that references the same set of dimension members. You can easily create conditions based upon:

- Parent-child relationships (children, siblings, descendents, and so on)
- Hierarchical relationships relative to a specified level
- All dimension members at a specified level
- All dimension members across all levels

Try to Create Dimension Selections with as Few Steps as Possible

The Discoverer Plus OLAP model for describing dimension selections through a series of step-by-step refinements is very powerful and easy to use. However, when designing a worksheet that will be accessed by many users over an extended period of time, take extra care to reduce the number of query steps for each dimension selection.

The following list provides examples of reducing the number of query steps:

- Collapse multiple match conditions into a single condition. For example, instead of a dimension selection that looks like this:

  Start with: Products containing *Audio*
  Then Add: Products containing *Video*
  Then Add: Products containing *Game*

  Consider creating a single match condition with a comma-delimited set of criteria:

  Start with: Products containing *Audio,Video,Game*

- Ask the database administrator to create additional attributes to characterize complex (but common) sets of dimension members. For example, if you find that you must specify multiple steps to obtain all the members of an Account dimension that represent expense items, then ask the administrator to create an Expense attribute for Account (with values of YES or NO), so that you can reference all expense items with a single match condition. Alternatively, the administrator can create a generic Item Type attribute and provide a List of Values (LOV) that consists of multiple types.

Make the Criteria as Precise as Possible

Give criteria as much precision as you can. For example, when creating a match condition, you will see slightly better performance if you specify a particular label type. Also, if you are confident of capitalization, then check the Match Case box.

Limit the Use of Null and Zero Value Settings

Discoverer Plus OLAP worksheets expose properties for hiding rows or columns that contain only null values or only null values and zeroes. Setting these properties can be very useful, but the use of this feature can be detrimental to query execution performance, particularly when the underlying data is very sparse.

Limit the Use of Manual Sort

Discoverer Plus OLAP supports precise control of how dimension members are ordered. Examples of sort order include:

- Move Z first
- Move B before A
- Drag and drop of dimension members in the Members tab

This feature is very powerful but does affect worksheet performance. If you find that you create precise sort criteria on a regular basis, then consider asking the database administrator to create additional metadata to make this easier. Options that are available to the administrator include:

- Creating alternate hierarchies with custom sort orders
- Creating new attributes or measures to be used within sort criteria

CONFIGURING DISCOVERER PLUS OLAP

As an administrator who configures Discoverer Plus OLAP, you can use various techniques to manage the application. The following sections describe what you can do.

Run Custom Code at Startup

Discoverer Plus OLAP allows you to run PL/SQL code at the start of the application. For example, you can run custom PL/SQL at startup to log the number of times that the application is started, to pre-attach an analytic workspace, to turn on SQL Trace, and so on.

To register a custom action, you need to create a procedure called D4O_AUTOGO in the D4OSYS schema.

User Logging Example: These steps describe how to use the D4O_AUTOGO program to log the number of times that the application is started:

1. Create the D4O_LOG table definition, which will contain the login entries:

```
CREATE TABLE D4O_LOG (USERNAME VARCHAR(32), WHEN TIMESTAMP);
```

2. Create the D4O_AUTOGO procedure:

```
CREATE OR REPLACE PROCEDURE D4O_AUTOGO(user IN varchar) AS

BEGIN
   INSERT INTO D4O_LOG VALUES(user, sysdate);
   COMMIT;
END;
```

3. Query recent user sessions:

```
SELECT * FROM D4O_LOG ORDER BY WHEN DESC;
```

Note: You can also execute OLAP DML commands using the PL/SQL package DBMS_AW.EXECUTE command.

Pre-configure Connections

If you associate the OracleBI Discoverer installation with an Oracle Application Server (OracleAS) Infrastructure instance, then users can save their own connections. This allows end users to start Discoverer Plus OLAP without having to specify the connection information and their user IDs. For security reasons, they must specify their passwords.

If OracleAS Single Sign-On (SSO) is enabled, then Discoverer end users can be authenticated by OracleAS Single Sign-On and can select a connection without confirming their passwords.

For information on associating a Discoverer installation with an OracleAS Infrastructure, see Chapter 2, “About Oracle Business Intelligence installations and OracleAS Infrastructures” in the *Oracle Business Intelligence Discoverer Configuration Guide*.

For more information on Connections and SSO, see Chapter 4, “Managing OracleBI Discoverer connections” in the *Oracle Business Intelligence Discoverer Configuration Guide*.

Use URL Parameters to Start Discoverer Plus OLAP

You can use URL parameters to start Discoverer Plus OLAP. For example, if you use URL parameters to specify that the workbook and worksheet be opened by default along with preconfigured connections, then users can create Web pages with links that immediately launch Discoverer Plus OLAP *and* automatically open specified worksheets.

The following sample URL launches Discoverer Plus OLAP and after the user enters an appropriate password, automatically opens a workbook named Finance and a worksheet named Profit:

```
http://discoolap1.us.oracle.com/discoverer/plus?workbookname=
   Users/<user>/Finance&sheet=Profit
```

For a list of the Discoverer Plus OLAP URL parameters, see the "URL parameters for the Discoverer Plus OLAP Servlet" topic in the *Oracle Business Intelligence Discoverer Configuration Guid*e.

Change the Discoverer Plus OLAP Look and Feel

You can change the look and feel of Discoverer Plus OLAP. There are four look and feel options from which you can choose:

- plastic (default)
- system (which gives a more native operating system-like look and feel)
- browser
- oracle

To change the look and feel, use the Oracle Application Server Control or the URL parameter lookandfeelname. For information on using the Oracle Application Server Control or the URL parameter lookandfeelname to change the look and feel, see the *Oracle Business Intelligence Discoverer Configuration Guid*e.

Understand How to Change Settings in the `configuration.xml` File

You can adjust a small set of configuration options by directly editing the `configuration.xml` file in the `<`*`IAS_HOME`*`>/discoverer/config` directory on the host machine where Discoverer is installed. If it does not already exist, then you must add a new `<D4O>` element (just before the `</plus>` tag) and include one or more of these configuration options as attributes within that single `<D4O>` element.

Adjust the OLAP Buffer Size

By default, data is fetched from the RDBMS into the Discoverer Plus OLAP client in increments of 1000 records. If the majority of worksheets in the installation are very large (for example, greater than 100 columns) and if end users complain about the resulting performance (particularly during scrolling), then consider adjusting the default OLAP buffer size.

Use the OlapBufferSize attribute to specify a new default OLAP buffer size. For example, to change the default OLAP buffer size to 10,000 records, add the following to the `<D4O>` element:

```
<D4O OlapBufferSize="10000" />
```

In order to activate the setting for existing workbooks (that is, existing before the OlapBufferSize setting was added), you must open and save all workbooks again to enable the setting.

Increasing the default OLAP buffer size can improve scrolling performance within large worksheets, but at the expense of degrading performance when the workbook is first opened and the additional data is fetched from the RDBMS. Therefore, experiment with different settings to find the right balance for your installation.

Change the Behavior of Alphabetical Sorting

By default, sort criteria are evaluated relative to the implicit order as specified by the hierarchy of the selection. For example, when sorting Products alphabetically from A to Z, the default behavior sorts all siblings alphabetically in ascending order, but retains them under their hierarchical parent.

Many installations that work extensively with value-based hierarchies (that is, parent-child hierarchies without named levels) prefer alphabetical sorting to be evaluated uniformly across the selection results, without regard to the implicit hierarchy order. If that is the case, then change the default behavior of alphabetical sorting for selections on value-based hierarchies with the AlphabeticalSortingBetweenLevels attribute. Valid values for this attribute are "false" (the default) or "true". For example, to change the default behavior so that alphabetical sorting for selections is evaluated without regard to the implicit value-based hierarchy order, add the following to the `<D4O>` element:

```
        <D4O AlphabeticalSortingBetweenLevels="true" />
```

Important: To change the alphabetical sorting behavior of an existing workbook, you must delete and re-create the corresponding Sort Criteria, then resave the workbook.

Select the Available Formats Copied to the Windows Clipboard

By default, data copied from a Discoverer Plus OLAP worksheet is placed on the Windows Clipboard in both Excel (HTML) format (which includes all font and style information) and in tab-delimited format (which does not include any font or style information). If end users require only one of these formats, then you can restrict the available formats copied to the Windows Clipboard with the CopyDataFlavor attribute. Valid values for this attribute are "both" (the default), "excel", or "text". For example, to change the default behavior so that worksheet data is copied to the Windows Clipboard only as tab-delimited text, add the following to the `<D4O>` element:

```
        <D4O CopyDataFlavor="text" />
```

An added advantage of restricting the available formats that are copied to the Windows Clipboard is improved performance of copy and paste operations.

Configure the Java-Client Memory Setting

Both Discoverer Plus Relational and Discoverer Plus OLAP run as Java applets in a browser. The maximum amount of memory that these programs are allowed to use is controlled by the Java runtime environment that is installed for the browser. This environment is generally either the Sun Java Plug-in or the Oracle JInitiator Plug-in.

Both the Sun Java Plug-in and the Oracle JInitiator Plug-in allocate a maximum of 64Mb of memory to an individual applet. In some cases, this amount of memory is insufficient, especially when using larger workbooks, or if running large print or export jobs. You might see an "Insufficient memory" error message or very slow performance.

To prevent insufficient memory errors, increase the amount of memory that the applet can use. Never increase this amount beyond 75% of the physical memory on the machine. The following examples describe how to increase the memory to 256Mb.

Example 1: Changing the memory setting for the Sun Java Plug-in to 256 Mb. The steps in this procedure can vary depending on the Sun Java version:

1. Launch the Windows Control Panel by selecting Start, Settings, Control Panel.
2. Launch the Sun Java Plug-in control from within Windows Control Panel.
3. On the Java tab, click the Java application runtime settings and type the following in the Java runtime parameters column:

   ```
   -Xmx256m
   ```

Example 2: Changing the memory setting for the Oracle JInitiator Plug-in to 256 Mb

1. Launch the Windows Control Panel by selecting Start, Settings, Control Panel.
2. Launch the Oracle JInitiator Plug-in control from within Windows Control Panel.
3. On the Basic tab, type the following in the Java runtime parameters field:

   ```
   -Xmx256m
   ```

Identify Discoverer Plus OLAP Database Sessions

If you (or your Oracle database administrator) want to monitor the Discoverer Plus OLAP database sessions using, for example, SQL*Plus, you can identify the database sessions by filtering on the session property "client_info" to display only the Discoverer Plus OLAP sessions. All Discoverer Plus OLAP client sessions have "Discoverer Plus OLAP" as the value for this property.

To see all the Discoverer Plus OLAP sessions, enter the following SQL command:

```
SELECT SID FROM SYS.V_$SESSION WHERE CLIENT_INFO ='DISCOVERER PLUS
OLAP';
```

Configure Database Connection Timeout

End users might leave their sessions open too long or might close Discoverer Plus OLAP with the “x” button in the upper-right corner of the window instead of choosing Exit from the File menu. (See the section “Use the Exit Command to Exit a Discoverer Plus OLAP Session.”) To improve performance under these conditions, use a database setting to force inactive sessions to time out automatically.

The following steps describe how to set the database time-out period from SQL*Plus. Alternatively you can use the Database Resource Manager in Database Control.

1. Enable resource limits dynamically with the ALTER SYSTEM statement and the RESOURCE_LIMIT initialization parameter.

   ```
   SQL>CONNECT SYS AS SYSDBA;
   SQL>ALTER SYSTEM SET RESOURCE_LIMIT=TRUE SCOPE=BOTH;
   ```

2. Create a profile using the CREATE PROFILE statement. Include the IDLE_TIME resource parameter in the CREATE PROFILE or ALTER PROFILE SQL statements to specify the permitted periods of continuous inactive time during a database session, expressed in minutes. When the inactivity of the session exceeds the IDLE_TIME limit, the session is rolled back. Long-running queries and other operations are not subject to this limit.

   The following example code defines the database time-out limit as 120 minutes.

   ```
   SQL>CREATE PROFILE d4ouser1_profile LIMIT IDLE_TIME 120;
   ```

3. Assign the profile to the Discoverer Plus OLAP user with the CREATE USER or ALTER USER statement.

   ```
   SQL> ALTER USER d4ouser1 PROFILE d4ouser1_profile;
   ```

If the user tries to access the session after the IDLE_TIME limit is exceeded, then an error such as the following one is displayed for the user and in the error stack:

> ORA-02396: exceeded maximum idle time, please connect again

MANAGING THE DISCOVERER CATALOG

As an administrator, you can use various techniques to manage the Discoverer Catalog (a very powerful and flexible content store for workbooks, calculations, and saved selections). The following sections describe what you can do.

Use the Discoverer Administration Tools from the Command Line

Discoverer Plus OLAP includes a command-line utility for performing administrative functions. This utility is called d4o.jar and it resides in the directory `<IAS_HOME>\sysman\webapps\emd\WEB-INF\lib`.

The syntax of this utility is as follows:

```
java –classpath d4o.jar oracle.dss.d4o.administration.D4OCommand –h
<host> –po <port> –sid <sid> –p <d4osys-password> <subcommand>
```

where:

- *<host>* is the IP address or host name of the machine on which Discoverer Plus OLAP is installed.
- *<port>* is the port number on which Discoverer Plus OLAP is installed.
- *<sid>* is the SID of the database on which the Discover Catalog is running.
- *<d4osys-password>* is the database password for the D4OSYS database account. D4OSYS contains the Discoverer Catalog schema.
- *<subcommand>* is the D4OCommand to execute. Some of these commands are listed here:
    - users: Lists the authorized Discoverer users.
    - version: Displayed the Discoverer Catalog version number.
    - allUsers: Lists all the users in the database.
    - export -f <*filename*>: Exports the contents of the Discoverer Catalog to the specified file.
    - import -f <*filename*>: Imports the contents from the specified file.
    - authorize -u <*username*> — Authorizes the specified database user.
    - install: Installs the Discoverer Catalog.
    - deinstall: Removes the Discoverer Catalog.

Obtain a list of all available commands by entering the following command:

```
java –classpath d4o.jar oracle.dss.d4o.administration.D4OCommand
```

Ensure that Users Exist in the Target Environment Prior to Importing Discoverer Catalogs

When you import catalog contents from one Discoverer Catalog to another, all path names of the exported objects (workbooks, calculations, and saved selections) are preserved in the catalog. Because most objects are stored in the Users/<*username*> folder or the Shared/<*username*> folder, it is important that these folders exist. Therefore, to facilitate a successful import, ensure that all users exist in the target environment prior to the import.

For any users that do not exist in the target environment, you must create them with the database administration tools and authorize them using the Discoverer administration tools.

Use the BI Beans Administration Tool to Perform Fine-Grained Import and Exports of Workbooks

You might want to move only a small number of workbooks from one Discoverer Catalog to another. To do so, use the Oracle Business Intelligence Beans (BI Beans) command-line utilities, which allow more fine-grained export and import options. When you export a workbook, all objects on which it depends, such as saved selections and calculations, are also exported.

The command to export a single workbook is as follows

```
java -classpath d4o.jar oracle.dss.persistence.administration.Export
-h <host> -po <port> -sid <sid> -j thin -c jdbc –u  D4OSYS -p
<d4osys-password> -lu D4OSYS export.xml Users/<user>/myworkbook
```

The command to import a single workbook is as follows:

```
java -classpath d4o.jar oracle.dss.persistence.administration.Import
-h <host> -po <port> -sid <sid> -j thin -c jdbc –u D4OSYS -p
<d4osys-password> -lu D4OSYS export.xml Users/<user>/myworkbook
```

SETTING UP THE ANALYTIC WORKSPACE

As an application implementer who is setting up a standard form analytic workspace that is accessed by end-user tools, you can use various techniques to optimize analytic workspace performance. The following sections describe what you can do.

Assign Attributes to the Hierarchies to Which They Apply

If you add new dimension attributes, then ensure that they are assigned *only* to the hierarchy or hierarchies to which they apply. To do so, in Analytic Workspace Manager, use the Apply Attributes To area toward the bottom of the General tab when creating or modifying a dimension attribute. Alternatively, you can use the Oracle OLAP Analytic Workspace Java API.

Make Formula Equations Performant

If you modify standard form measure formula equations, then ensure that they are as performant as possible.

OLAP Applied Engineering is available to review any modified formula equations for good practice. Contact your account team to request a review by OLAP Applied Engineering.

Implement Dimension Value Orderings Using an Alternate Hierarchy or a Dimension Attribute

You can implement dimension value orderings that are used frequently in end-user reports using an alternate hierarchy with that ordering or by using a dimension attribute of type text. To establish these structures, use Analytic Workspace Manager or the Analytic Workspace Java API.

For more information about Analytic Workspace Manager or the Analytic Workspace Java API, see *Oracle OLAP Application Developer's Guide*.

Implement Dimension Value Selections Using Attributes Defined in the Analytic Workspace

You can implement dimension value selections for Discoverer Plus OLAP users using attributes that are defined in the analytic workspace. Implement these attributes as analytic workspace relations; check the Index box on the Create Attribute screen in Analytic Workspace Manager or perform the equivalent operation using the Analytic Workspace Java API.

TUNING TIPS FOR THE RDBMS WITH THE OLAP OPTION

As an Oracle DBA tuning the RDBMS, you can use various techniques to optimize performance.

Note: Database tuning is a complex task. The suggestions presented in this section are fairly general in nature. You can contact an Oracle OLAP tuning expert through your account team who can discuss specific issues or who can perform a thorough analysis of your Oracle database system.

For any discussions with an Oracle OLAP tuning expert, have detailed hardware specifications on CPU, memory, disk I/O, and a list of the current database parameter settings. Also create an Automatic Database Diagnostic Management Report, as described in the next section.

Know Your System

If the database instance for Discoverer Plus OLAP is not on a dedicated machine, then you should establish a baseline resource utilization record for your system to determine resource availability for Oracle OLAP.

To analyze database resource use either for a baseline determination or for measurement during Oracle OLAP usage, perform the following steps:

- During peak load and at reasonable time intervals, take workload snapshots to establish resource utilization records for your system.

  To take a snapshot, use this command:

  ```
  exec dbms_workload_repository.create_snapshot();
  ```

  To identify all the Automatic Workload Repository (AWR) Snapshots that have been taken, use this script:

  ```
  set lines 110 pages 1000

  col SNAP heading "Snap"
  col BEGIN heading "Begin"
  col END heading "End"
  col LVL format 9999 heading "Level"
  ```

```
select snap_id snap, to_char(begin_interval_time, 'DD-MON-YYYY
HH24:MI:SS') as begin,
to_char(end_interval_time, 'DD-MON-YYYY HH24:MI:SS') as end,
snap_level lvl
from dba_hist_snapshot where end_interval_time > systimestamp-
5/24;
```

Example: AWR Snapshots that occurred over the last 5 hours

```
      Snap Begin                End                  Level
---------- -------------------- -------------------- -----
       508 08-AUG-2006 11:00:59 08-AUG-2006 12:00:12     1
       509 08-AUG-2006 12:00:12 08-AUG-2006 13:00:25     1
       510 08-AUG-2006 13:00:25 08-AUG-2006 14:00:38     1
       511 08-AUG-2006 14:00:38 08-AUG-2006 15:00:50     1
       507 08-AUG-2006 10:00:46 08-AUG-2006 11:00:59     1
```

- Run reports.

  To run the Automatic Workload Repository Report, use this command

  ```
  @ <path>\rdbms\admin\awrrpt.sql
  ```

  To run the Automatic Database Diagnostic Management Report, use this command:

  ```
  @ <path>\rdbms\admin\addmrpt.sql
  ```

The snapshots and reports will help you understand what resources are available and provide information on techniques to reach your performance goal with Oracle OLAP.

Ensure that the hardware on which the RDBMS is installed is balanced so that no one aspect causes performance bottlenecks. Automatic Database Diagnostic Monitor (ADDM) is a very useful tool for this task. Faster CPUs, increased system memory, and faster and better-balanced I/O all contribute to better performance when kept in balance.

Of the various choices for disk data access, Automatic Storage Management (ASM) is the recommended choice as it provides good performance with convenient storage management.

For more information on ADDM, see the article "A Closer Look at ADDM," which you can find here: www.oracle.com/technology/oramag/oracle/04may/o34tech_talking.html

Increase Size of Redo Logs

Check the alert log for the frequency of switching to output redo logs. If this switching occurs more frequently than approximately every 10 minutes, then increase the size of the redo logs. A size of 500Mb to 1Gb generally works well.

Set the SGA_TARGET, PGA_AGGREGATE_TARGET, DB_CACHE_SIZE, and OLAP_PAGE_POOL_SIZE Parameters to Recommended Values

Set the PGA_AGGREGATE_TARGET and SGA_TARGET parameters to 35% of available memory each as a starting point. Check the ADDM output periodically to see if these settings should be changed. By default, the components of the SGA (System Global Area) are controlled automatically, which is recommended.

The PGA_AGGREGATE_TARGET setting indirectly controls two important memory areas used by Oracle OLAP:

- The database cache

  Set the DB_CACHE_SIZE to 0 (the default) to engage automatic memory allocation.

- The Oracle OLAP page pool
  - Set the OLAP_PAGE_POOL_SIZE parameter to 0 (the default), which engages the automatic and dynamic sizing of the Oracle OLAP page pool.
  - Set the _OLAP_PAGE_POOL_HI parameter, which controls the maximum amount of OLAP page pool allocated to any one user, to its default of 50. This allows a user to get up to 50% of the PGA_AGGREGATE_TARGET value. For large user communities, it may be desirable to set this parameter to a lower value.

You can learn the following information for each user who accesses analytic workspaces:

- The amount of PGA (Program Global Area) that is in use and that is the maximum size.
- The size of the Oracle OLAP page pool.
- The hit/miss ratio for Oracle OLAP pages.

To learn this information, use the following script:

```
set lines 110 pages 500

col usn format a24 heading "User (SID,SERIAL#)"
col pga_used format 9,990.9 heading "PGA Use|MB"
col pga_max format 9,990.9 heading "PGA Max|MB"
col olap_pp format 9,990.9 heading "OLAP|Pg Pool|MB"
col aw_cnt format 999 heading "AW|CNT"
col olap_hrate format 99.9 heading "OLAP|Hit|Rate"

bre on REPORT;

comp avg lab Average of aw_cnt on REPORT;
comp avg lab Average of olap_hrate on REPORT;
comp avg lab Average of pga_used on REPORT;
comp avg lab Average of pga_max on REPORT;
comp avg lab Average of olap_pp   on REPORT;
```

```
select vs.username||' ('||vs.sid||','||vs.serial#||')' usn,
     round(sum(pga_used_mem)/1024/1024,1) pga_used,
     round(sum(pga_max_mem)/1024/1024,1) pga_max,
     round(sum(pool_size)/1024/1024,1) olap_pp,
     round(100*(sum(pool_hits)/(sum(pool_hits)+sum(pool_misses))),1)
     olap_hrate,
     count(aw_number) aw_cnt
from v$process vp, v$session vs, v$aw_calc va, v$aw_olap vo
where va.session_id=vs.sid and vo.session_id=va.session_id and addr =
paddr
group by vs.username, vs.sid, vs.serial#
order by vs.username, vs.sid, vs.serial#;
```

Example: The following report shows the PGA (in use and maximum), Page Pool size, and Pages Hit/Miss Ratio numbers for the SCOTT and ADAM users. The "OLAP AW CNT" column contains the number of analytic workspaces that each user attached.

```
                                                          OLAP OLAP
                            PGA Use   PGA Max   PG Pool   Hit    AW
User (SID,SERIAL#)               MB        MB        MB  Rate   CNT
------------------------   --------  -------- --------   ----- ----
SCOTT (140,62)                283.1     562.3    18.5    99.8     3
ADAM (138,24)                  56.5      77.3    26.6    99.3     3

                           --------  -------- --------   ----- ----

Average                       169.8     319.8    22.6    99.6     3
```

Set Parameters for Analytic Workspace Build Jobs

Single process analytic workspace builds can use parallel updating processes when the analytic workspace is updated with the UPDATE command in the OLAP DML. This can increase the speed of the updating process significantly.

To turn on parallel update for analytic workspaces, use the following commands:

```
alter session set "_olap_parallel_update_threshold"=1000 scope=spfile;
```

```
alter session set "_olap_parallel_update_small_threshold"=1000
        scope=spfile;
```

If an analytic workspace build uses concurrent processes, then do not use parallel updating processes.

To turn off parallel update for analytic workspaces, use the following commands:

```
alter system set "_olap_parallel_update_threshold"=2147483647
   scope=spfile;

alter system set "_olap_parallel_update_small_threshold"=2147483647
   scope=spfile;
```

REFERENCES

To learn more about Discoverer Plus OLAP, consult the following resources:

- Discoverer Documentation on the Oracle Technology Network

  You can find this documentation here:
  http://www.oracle.com/technology/documentation/discoverer.html

- Discoverer Plus OLAP help system

  Click the Help button or press F1 in any Discoverer Plus OLAP dialog to display context sensitive help, or choose Help | Help Topics to see a list of the topics in the help system.

  To find a topic in the help system, use the:

  - Contents tab to see a structured list of the topics in the help system
  - Index tab to search the help system using pre-defined keywords and phrases
  - Search tab to search the help system for keywords and phrases that you define

- Oracle By Example (OBE) Series

  You can find this series here:
  http://www.oracle.com/technology/obe/obe_bi/bi_10122/index.html

- Oracle BI Samples

  You can find these samples here:
  http://www.oracle.com/technology/products/bi/samples/index.html

Best Practices for Oracle Business Intelligence Discoverer Plus OLAP Version 10.1.2.2
January 2007

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com