EXAM OBJECTIVE: MONITOR AND MAINTAIN THE AWR

OVERVIEW OF COMMON MANAGEABILITY FEATURE OF ORACLE 10G

In earlier versions of Oracle, a major part of a DBA responsibility revolved around determining solutions to various issues about the health of the database. In Oracle 10g the Oracle Database performs database manageability tasks that automatically inform the DBA of any performance or resource allocation problems. This is popularly known as the Manageability infrastructure of Oracle 10g.

The main components of this common manageability infrastructure are:

- **The Automatic Workload Repository (AWR)** – This is central element of the manageability infrastructure. This component installs with the Oracle database, and is responsible for collecting, maintaining, processing and utilizing performance statistics and metrics for problem detection and self-tuning purposes.
- **Server-Based Advisors** – A number of new advisors have been designed around the manageability infrastructure to provide valuable recommendations for specific problems. The Automatic Database Diagnostic Monitor (ADDM), identifies bottlenecks in the database, and additionally makes recommendations on the options that are available to fix them.
- **Automatic Routine Administration tasks** – This is implemented via the Scheduler. Certain routine or repetitive tasks can be delegated to the Scheduler, in order to ensure the good shape of the Oracle database. Once such routine task could be analyzing optimizer statistics.
- **Server-Generated Alerts –** In this regard, a DBA can configure alerts that trigger during certain problem areas, such as out-of-space errors for tablespaces. When a problem is detected, the Oracle database sends you an alert message with possible remedial actions.

THE AUTOMATIC WORKLOAD REPOSITORY

In earlier versions of Oracle, historical statistical data could be manually obtained by means of Statspack. However, it was the database administrators responsibility to know how frequently statistical data should be captured. It provides the ability to capture statistics with minimal user intervention as well as detect and provide advice based on history of performance data in the system.

IN-MEMORY COLLECTION

- The automatic workload repository consists of an in-memory collection facility. Statistics are stored and are accessible by means of fixed views. They are referred to as base statistics. Base statistics include object statistics which determine both access and usage statistics of database segments, time statistics

that indicate how much time was spent for a particular activity, system and session statistics, oracle optimizer statistics and ADDM active session history which represents the history of recent sessions activity.

WORKLOAD REPOSITORY

- The AWR is also made up a Workload repository, which is persistent and accessible by means of data dictionary views.
- They can be used to generate baseline comparisons. Historical statistics are useful to analyze problems that occur intermittently as well as identify a trend.
- The workload repository is created in a system-defined **WR** schema.
- The WR schema resides in the new **SYSAUX tablespace** and is owned by the user SYS.
- A snapshot is a set of performance statistics captured by the workload repository at a certain time. By default snapshots are captured automatically every 60 minutes. This interval can be modified if necessary. Snapshots are uniquely identified by an identifier (SNAP_ID). A DBA can also take snapshots manually by invoking a PL/SQL procedure, however this practice is not highly recommended.
- In order to control the set of statistics captured you can set the STATISTICS_LEVEL initialization parameter. If the parameter is set to a value of BASIC, then all self-tuning capabilities are turned off, implying that AWR statistics and metrics are tuned off. If set to the recommended default of TYPICAL, only part of the statistics are captured. A value of ALL indicates all possible statistics are captured.

  The memory statistics are transferred to disk on a regular basis by a new background process called the **MMON**. To create space for capturing new statistics, old snapshots are purged automatically by MMON every night. You can also control the amount of historical statistics retained in the Workload repository by setting a retention period. In general, snapshots are removed in the order they were added.

WR SNAPSHOT CAPTURE AND PURGING POLICY

- To configure the Retention period and interval for statistics collection you can use the **DBMS_WORKLOAD_REPOSITORY .MODIFY_SNAPSHOT_SETTINGS packaged procedure.**
- The RETENTION parameter, indicating how long the statistics should be retaine can be set. The default being 7 days. Automatic purging can be turned off by setting it to a value of 0, and AWR maintains a minimum of one day of history. Deletions of snapshots occur if the SYSAUX tablespace runs out of space.

- Automatic snapshot generation is configured by setting the INTERVAL parameter of **DBMS_WORKLOAD_REPOSITORY. MODIFY_SNAPSHOT_SETTINGS packaged procedure**. The value cannot be less than 10 minutes. The default value is 60 minutes. Setting it to a value of 0, turns off automatic capturing of snapshots.

DBMS_WORKLOAD_REPOSITORY.MODIFY_SNAPSHOT_SETTINGS (retention NUMBER, interval NUMBER).

BASE STATISTICS AND METRICS

Base statistics is raw data collected by the database. Metrics are secondary statistics derived from base statistics. Most metrics track the rates of change of activities in the Oracle database. For e.g. the average physical reads in the system in the last 60 minutes is a metric. Internally the MMON process updates metric data from the corresponding base statistics.

WORKLOAD REPOSITORY STRUCTURE

The Workload Repository (WR) schema consists or metadata tables and statistical tables. The metadata tables are used to control, process and describe the WR tables. It gives Oracle an idea as to when to perform snapshots and what to capture to disk. The historical statistic tables store information in the database for future reference.

All names in the WR tables are prefixed with the **WR*x*$.** The *x* specifies the kind of table, and can take the character M or H. A table prefixed with WRM$, represents a table that stores metadata and a table prefixed with the WRH$ stores historical data and snapshots.

Data dictionary views can also be used to query the workload repository data. A data dictionary view that is related to historical data may have a prefix like **DBA_HIST_.**

GENERATING BASELINES

A baseline is a means of labeling sets of snapshot data for important periods. It is defined on a pair of snapshots. A base-line is identified by either a user-supplied name or a system-generated identifier. A baseline may be created using the **DBMS_WORKLOAD_REPOSITORY.CREATE_BASELINE** packaged procedure.
Snapshots that belong to baselines are retained until the baseline is dropped.

Procedure Name	Description of Procedure
CREATE_SNAPSHOT	To manually create a snapshot immediately

DROP_SNAPSHOT_RANGE	To drop a range of snapshots
CREATE_BASELINE	To create a single baseline
DROP_BASELINE	To drop a single baseline
MODIFY_SNAPSHOT_SETTINGS	To modify a snapshot settings

AWR REPORTS

- You can generate reports based on statistics stored in the Workload repository. The report generation interface is provided in the form of **awrrpt.sql**.

EXAM OBJECTIVE: USING THE ACTIVE SESSION HISTORY (ASH)

Session activity is recorded in another new area in memory known as the Active Session History.

The memory for the ASH comes from the System Global Area (SGA) and is fixed for the lifetime of the instance. The ASH memory cannot exceed a maximum bound of five percent of the shared pool size.

The view **V$SESSION** has been improved in Oracle 10g. The **data is polled every second** and only the active sessions are polled. The contents of the ASH are particularly important since the Workload Repository **snapshots are taken every hour** indicating the last snapshot could have been taken nearly an hour ago. The contents of the ASH are filtered and transferred into workload repository by the **MMON** every 60 minutes and by the MMNL process whenever the buffer is full. Every second the Oracle Database looks at the currently running active sessions, and records the events these sessions are waiting for. Using this sampling technique the Oracle database internal structures are directly accessed. Data that is sampled include:

- SQL_ID : New hash value to uniquely identify SQL statements
- SID
- Client ID, Service ID
- Program, Module, Action
- Object, File, Block
- Wait Event number, actual wait time (if waiting)

ASH statistics are available through the V$ACTIVE_SESSION_HISTORY fixed view.

EXAM OBJECTIVES: MONITOR AND MANAGE SERVER-GENERATED ALERTS

Server-generated alerts are the capability of the Oracle database to automatically detect alarming situations that may arise in the database. If an alarming situation occurs, the DBA can be alerted with suggestions for remedial actions.

An alert may be configured to identify some internally determined or user-defined problem situation. An alert can be contain information about the object such as a tablespace, a description of the problem, a brief advice on remedial action, the name of an advisor that can provide more detailed recommendations and the status or level of severity.

An alert may be queued in a predefined queue called **ALERT_QUE** that is owned by the SYS user. The Enterprise Manager may be used to notify the DBA about outstanding alerts. This may be done by means of email or pager. Server-generated alerts are always displayed on the EM console.

There are two kinds of server-generated alerts:

- Threshold
- Non-threshold alerts.

Most of the alerts in the Oracle database are configured by setting two threshold values, namely **Warning** threshold and **Critical** threshold. In the Oracle database there are about 161 metrics for which you can define thresholds. Some of them being User Commits Per Sec, CPU Time Per call, SQL service Response Time, Physical Reads Per Second. A threshold alert is automatically cleared when the DBA has responded to the alert.

A threshold alert is displayed in the **DBA_OUTSTANDING_ALERTS** view and when cleared is sent to the **DBA_ALERT_HISTORY** view.

Server generated errors may also correspond to database events and are known as non-threshold errors. These errors go directly to the history table. Examples of some situations that result in alerts are:

- Snapshot too old errors
- Resumable session suspended
- Recovery Area Low on Free space errors

Certain alerts are always enabled in the database and may be called out-of-the-box alerts. To mention a few:

- Tablespace usage
- Snapshot too old
- Resumable session suspended

- Recovery Area Low on Free space errors

The SERVER-ALERT USAGE

As a DBA if you want the Oracle database to notify you of alert situations, you can use the model given below:

- Using the EM console or PL/SQL procedure set the threshold limits for the server alert metrics.
- Identify a means to be notified, either by Email address, blackout period or pager).
- When the alert is generated, the alert displays in the EM console. EM sends out a notification to the administrators.
- You may use the suggestions provided to correct the alert.

The STATISTIC_LEVEL initialization parameter should be set to TYPICAL or ALL for this functionality.

SETTING THRESHOLD VALUES USING PL/SQL

The new **DBMS_SERVER_ALERT** package is used to set up thresholds on metrics.

PROCEDURE NAME	Description
SET_THRESHOLD	Used to define threshold settings for a given metric
GET_THRESHOLD	Used to read settings for a given metric

Consider the example below:

```
SQL> BEGIN
       DBMS_SERVER_ALERT.set_threshold (
       DBMS_SERVER_ALERT.tablespace_pct_full,
       DBMS_SERVER_ALERT.operator_ge, 80,
       DBMS_SERVER_ALERT.operator_ge, 95, 1, 1, ORCL,
       DBMS_SERVER_ALERT.object_type_tablespace, 'USERS');
     END;
```

You can use the NULL value to return to the database-wide default values.

The parameters are explained in the order they appear in the example:

- The internal name of the metric (tablespace_pct_full)
- The operator for comparing the actual value with the threshold value.
- The warning threshold value (80)
- The operator for comparing the actual value with the threshold value
- The critical threshold value (95)

- The observation period which defines for how long the actual behaviour of the system must deviate from the threshold value before the alert is issued. (1 hour).
- The consecutive occurrences which define how many observation periods the metric value should violate the threshold values before the alert is issued (1 observation period)
- The name of the instance for which the threshold is set (ORCL).
- The type of metric (object_type_tablespace)
- The name of the object for which the thresholds are set (USERS).

NEW VIEWS IN ORACLE 10g

The MMON background process is responsible for computing metric values regularly and retaining them in memory for one hour.

The in-memory metric values can be viewed through the **V$SYSMETRIC** and **V$SYSMETRIC_HISTORY** view. The views provide system-level statistics.

For service level metrics you can view the **V$SERVICEMETRIC** and **V$SERVICEMETRIC_HISTORY** view. To display the above mentioned metrics from disk you can query the **DBA_HIST_*** views.

The views that provide information about server alerts are:

- **DBA_OUTSTANDING_ALERTS –** outstanding alerts
- **DBA_ALERT_HISTORY** – time-limited history of alerts that are no longer outstanding.
- **DBA_THRESHOLDS –** threshold settings defined for the instance.
- **V$ALERT_TYPES** – information of each alert.

USING EM TO VIEW ALERTS

Alerts can be displayed on the Database Home Page, under the Diagnostic Summary section. To change the metric thresholds you can **Select Database Home Page -> Related Links region -> Manage Metrics->Edit Thresholds button ->Edit thresholds page.**

EXAM OBJECTIVE: EXPLAIN THE AUTOMATED TASKS FEATURE

We now concentrate on the next important component of the new manageability infrastructure. This is to do with automated tasks. The **Scheduler** is used by Oracle 10g database to perform certain routine or repetitive tasks that should be performed for the overall health of the database. One such task could be the collection of optimizer statistics. The Scheduler is discussed in detail as part of another exam objective.

THE DBCA and AUTOMATED TASKS

The Database Configuration Assistant (DBCA) has a page that lets you configure the automation of backup jobs.

As part of automatic routine administration tasks we now look at how automatic optimizer statistic collection is done.

EXAM OBJECTIVE: DESCRIBE THE ADVISORY FRAMEWORK

Advisors are server components that provide relevant feedback about resource utilization and performance with respect to specific database components.

The advisors use a common data source known as the **Automatic workload repository (AWR).** A list of available advisors is given below:

1. **Automatic Database Diagnostic Monitor (ADDM)**
   This advisor monitors the overall health of the database. Uses a top-down approach starting with symptoms and coming down to the root causes. It also comes up with recommendations for fixing the problems. This advisor is a central component to the advisory framework and can call upon other advisors.

2. **SQL Tuning Advisor**
   Provides SQL statement tuning advice.

3. **SQL Access Advisor**
   Deals with schema issues and determines optimal ways to access data.

4. **Memory Advisor**
   An advisor for system memory and optimizes the memory for an instance. Used when automatic memory tuning is not enabled.

   a. PGA Advisor – Gives detailed advice and recommendations on work areas along with optimal usage of PGA memory based on workload characteristics.
   b. SGA Advisor – tunes and provides recommendations for the SGA, depending on the access to various components in the SGA.
   c. Buffer Cache Advisor – Predicts cache hit rates for buffer access for different sizes of the buffer cache.
   d. Library Cache Advisor – Predicts the cursor cache hit rate for he library cache for different sizes.

5. **Segment Advisor**

Responsible for space issues regarding a database object. It can analyze growth trends.

6. **Undo Advisor** – Advices and recommends values for undo retention and size of the undo tablespace based on current workload and workload history.

An advisor may be invoked during the MMON performing its monitoring actions or in a wizard in some of the DBA tools. Certain server alerts may also recommend the DBA to view a certain Advisor for additional information.

ATTRIBUTES OF ADVISORS:

- Can operate in **LIMITED** and **COMPREHENSIVE** Modes: Limited displays a relatively shallow analysis whereas a comprehensive mode displays a more in-depth analysis.
- Time limit: Specifies a maximum amount of time that the advisor can run.
- Interruptible: Specifies if an advisor provides partial results upon being interrupted.
- User Directive: Specifies whether the advisor can take user directives. These could influence the recommendations provided.

INVOKING THE ADVISORS FROM EM CONSOLE

Select the Database Control Home Page -> Related Links -> Advisor Central Link -> Advisor Central Page.

From the advisor central page you can list all the advisor tasks that were registered in the workload repository.

THE DBMS_ADVISOR PACKAGE

This is a package that is new to Oracle 10g. It consists of various procedures and constants that are used by advisor modules. To execute any of the procedures you must be granted the ADVISOR privilege.

PROCEDURE	DESCRIPTION
CREATE_TASK	Creates a new task in the repository
DELETE_TASK	Deletes a task from the repository
EXECUTE_TASK	Initiates execution of the task
INTERRUPT_TASK	Suspends a task that is currently executing
GET_TASK_REPORT	Text report about recommendations
RESUME_TASK	Resumes a suspended task
UPDATE_TASK_ATTRIBUTES	Updates task attributes

SET_TASK_PARAMETER	Modifies a task parameter
MARK_RECOMMENDATION	Marks one or more recommendations as accepted, rejected or ignored
GET_TASK_SCRIPT	Creates a script of all the recommendations that are accepted.

CONFIGURING A TYPICAL TUNING SESSION USING PL/SQL

In a typical tuning session you would execute the following steps:

1. Create an advisor task. – This is an executable data area in the advisor repository that manages the task. You can do this by executing the **DBMS_ADVISOR.CREATE_TASK** procedure.

2. Adjust the task parameters – Parameters can be set to control the behavior of the task. Parameters may include TARGET_OBJECTS, TIME_WINDOW, TIME_LIMIT etc. The TIME_WINDOW specifies the time period that the Advisor considers for its analysis. You can perform this step by executing the **DBMS_ADVISOR.SET_TASK_PARAMETER** packaged procedure.

3. Perform an analysis – This step initiates the execution of the task. A task may be interrupted and then resumed if necessary. The **DBMS_ADVISOR.EXECUTE_TASK** packaged procedure may be invoked during this step.

4. Review the results – You can generate a report consisting of the various recommendations Oracle provides. You can do this by executing the **DBMS_ADVISOR.CREATE_TASK_REPORT** packaged procedure.

Given below is an example of manually invoking the ADDM advisor to analyze the system between snapshots 60 and 66.

```
SQL> VARIABLE TNAME VARCHAR2(50)
DECLARE
        TID             NUMBER;
BEGIN
        DBMS_ADVISOR.CREATE_TASK('ADDM',TID,:TNAME);
        DBMS_ADVISOR.SET_TASK_PARAMETER(:TNAME,'START_SNAP
SHOT',60);
DBMS_ADVISOR.SET_TASK_PARAMETER(:TNAME,'END_SNAPSHOT',66
);
DBMS_ADVISOR.EXECUTE_TASK(:TNAME);
END;
```

IMPORTANT DATA DICTIONARY VIEWS

VIEW NAME	DESCRIPTION
DBMS_ADVISOR_DEFINITIONS	Properties of the advisors
DBMS_ADVISOR_TASKS	Global information about the task
DBMS_ADVISOR_LOG	Task current status information
DBMS_ADVISOR_PARAMETERS	Task's parameters
DBMS_ADVISOR_COMMANDS	Commands associated with actions
DBMS_ADVISOR_OBJECTS	Objects referenced by tasks
DBMS_ADVISOR_FINDINGS	Findings discovered by the advisor
DBMS_ADVISOR_RECOMMENDATIONS	Task's recommendations
DBMS_ADVISOR_ACTIONS	Actions associated with recommendations
DBMS_ADVISOR_RATIONALE	Rationales for the recommendations
DBA_ADVISOR_USAGE	Usage information for each advisor