DBA ON AUTOPILOT WITH ORACLE DATABASE 10G

Arun Kumar R., Cingular Wireless, Dallas.

Introduction

Until Oracle 9i, database administrators used to spend a good deal of time on database monitoring, identifying problem areas and performance bottlenecks and trying to improve the database performance. These tasks were focused on System Resources, Storage management, Space Management, Application and SQL tuning, and Backup and Recovery Management. Oracle Database 10g has come up with a slew of novel changes to its architecture over the previous versions. By making the database more sophisticated and powerful, Oracle has automated many of the traditional administrative functions, making Oracle Database 10g suitable even for small businesses. This paper will focus on giving an overview of all the major automatic features of Oracle Database 10g so that the DBA can run many of administrative routines (system resources, storage management, and SQL tuning) in an "autopilot" mode.

What is Automatic Workload Repository?

Automatic Workload Repository (AWR) forms the central component of the new Oracle Database 10g manageability infrastructure. It provides services to the database to access, collect, process, and maintain performance statistics on various functionalities within the database. It is a built-in repository in every database.

During regular intervals, Oracle Database 10g takes snapshots of all vital statistics and workload data, and then stores them in the repository. This data is later used for analysis and as statistics for problem detection and self-tuning. By default the interval is 60 minutes and the data is stored for a period of seven days, after which it then gets purged. The interval and retention period can be altered. This captured data can be used for system level and user level analysis. This data is optimized to minimize overhead. In a nutshell, AWR is the basis for all self-management functionalities of the database. It helps the Oracle Database with the historical perspective on its usage enabling it to make accurate decisions quickly.

Architecture of AWR

The AWR infrastructure consists of two major components: In-Memory Statistics Collection Facility (useful for 10g components to collect statistics, stored in memory for performance reasons) and AWR Snapshots which represent the persistent portion of the facility. The memory version of the statistics is written to disk regularly by a new background process called Memory Monitor or Memory Manageability Monitor (MMON). AWR snapshots can be viewed through data dictionary views. The AWR statistics are kept in persistent storage to survive any database instance crashes and to provide historical data for baseline comparisons.

Here is a partial list of the many statistics AWR in Oracle Database 10g collects: time model statistics based on time spent by the activities, object statistics that represent the access and usage statistics of database segments, session and system statistics retained in *v$sesstat* and *v$sysstat*, and optimizer statistics for self-learning and tuning, ADDM, and Active Session History.

Active Session History

The Active Session History (ASH) contains recent session activity. The memory assigned to the ASH comes from the SGA and is fixed for the instance lifetime. The ASH works by sampling the *v$session* view every second and recording the events the active sessions are waiting for. The *v$active_session_history* view is used to access the ASH statistics. Since ASH data is of high volume, flushing all ASH data is not advisable. Therefore, the data is filtered out by writing it to disk using MMON every 30 minutes and by MMNL (Memory Monitor Light) when the buffer gets full.

AWR Snapshots

The Automatic Workload Repository (AWR) is a collection of persistent system statistics stored in the system defined Workload Repository (WR) schema. The WR schema resides in the SYSAUX tablespace and is one of the main SYSAUX components.

A snapshot is a set of performance statistics captured at a certain time in the database. It is used for computing the rate of change of a statistic. Every snapshot is identified by a snapshot sequence number (*snap_id*) that is unique within AWR. Snapshots are generated every 60 minutes by default. The snapshot *interval* parameter can be used to change this frequency.

This interval determines if snapshots of all major activities in the database are being obtained. Automatic snapshots are generated using an internal MMON task, which runs periodically. Manual snapshots can be taken by using the database control or a PL/SQL procedure.

Statistics Collection Process

Oracle Database 10g metrics can be tracked using AWR. The usage metrics are divided into two categories- Database feature usage and High Water Mark (HWM) value of certain database attributes. These metrics are helpful in determining how often a particular feature is used, and helpful in determining resource usage within the database. MMON tracks and records the database feature usage and HWM statistics on a weekly basis. The tracking is based on a sampling mechanism of the data dictionary. These statistics are recorded in AWR snapshots. The *dba_feature_usage_statistics* view and the *dba_high_water_mark_statistics* view can be queried to get these statistics. The Enterprise Manager can also be used to view the recorded statistics.

Server Generated Alerts

Oracle Database 10g has features that help the DBA reduce the time spent on database monitoring tasks by automatically notifying the DBA of performance or resources allocation issues and suggesting remedial actions. A *server-generated alert* is a notification message from the server of an impending problem. Whenever a database metric does not match expected values or thresholds levels on different metrics are reached, an alert will be issued. This notification will contain the error/alert condition, and may contain suggestions for correcting the problem. Alerts are also generated when the problem condition has been cleared.

MMON helps by regularly scheduling monitoring actions. The main difference between OEM alerts and server generated alerts is that with server generated alerts the metrics computation and threshold validations are performed by Oracle Database 10g and not by the OEM agent. In Oracle Database 10g, the SGA can be access directly and the MMON wakes up every minute to compute the metric values. For those metrics with thresholds, MMON verifies these values and generates alerts as needed. Alerts based on threshold levels can be triggered at warning levels and critical levels. These levels can be internally set, customer defined, or customer altered from preset values.

There are two kinds of server generated alerts, threshold alerts and non-threshold alerts. Most alerts are configured by setting two threshold values on database metrics, a warning threshold and a critical threshold. These threshold alerts are also known as stateful alerts, which are automatically cleared upon fixing the alert condition. Stateful alerts are stored in *dba_outstanding_alerts* and are moved to *dba_alert_history* when cleared.

An alert message will be sent to the SYS owned predefined persistence queue called *alert_que*. Oracle Enterprise Manager (OEM) reads this queue to provide notifications about outstanding alerts, and may suggest corrective actions. The alerts are displayed on the Enterprise Manager console. OEM can be customized to send these messages to pager or email addresses. If the alert cannot be written to the queue, a message about the alert will be written to the Oracle Database alert log.

Reviewing Metrics and Thresholds

Oracle defines Metrics as a set of statistics for certain system attributes. These statistics are calculated and stored by the Automatic Workload Repository (AWR). These results are displayed in OEM through the *All Metrics* page under *Related Links* on the *Database Home* page.

Responding to Alerts

Whenever you receive an alert, follow the recommendations it provides. Or you can run ADDM or another advisor as needed to get a more detailed diagnostics of system or object behavior. You can also opt to run a corrective script to run on receiving an alert as mentioned in Managing Metric Thresholds section.We will need to clear the alerts as well as a part of alert management.

Clearing of Alerts

Oracle clears most of the alerts automatically when the cause of the alert disappears. But, there are certain alerts that need to be acknowledged by the DBA, and alerts that require corrective measures. After taking the necessary corrective actions, acknowledge the alert by clearing it or purging it. Clearing an alert sends it to the Alert History, which is viewable from the home page of OEM under Related Links; whereas, purging an alert removes it from the Alert History.

To clear an alert, choose the *Alerts* link in the Home page. Click the *Alert link* and the Alert Log page appears. Select the alert to be cleared and click *Clear*. You can click *Purge* to purge the alert. You can also *Clear Every Open Alert* or *Purge Every Alert* using these buttons, though it is not recommended as a good practice.

Advisory Framework

Oracle Database 10g has server components called advisors to provide feedback about resource utilization and performance. The important advisors are listed below, though we will limit our discussion to system, space, and SQL tuning advisors.

- Automatic Database Diagnostic Monitor (ADDM) - ADDM does a top-down analysis of the database, identifies problems and potential issues, and gives recommendations for fixing these problems. ADDM can invoke other advisors.
- SQL Tuning Advisor - provides tuning advice for SQL statements.
- SQL Access Advisor - provides advice on database schema issues and determines optimal data access paths.
- SGA Advisor - is responsible for tuning SGA size depending on the pattern of access for various SGA components.
- PGA Advisor - gives detailed statistics for the work areas and provides recommendations on optimal PGA usage based on workload characteristics.
- Buffer Cache Advisor - predicts buffer cache hit rates for different buffer cache sizes.
- Library Cache Advisor - predicts the cursor cache hit rate for different library cache sizes.
- Segment Advisor - tracks object space issues and analyze growth trends.
- Undo Advisor - recommends parameter values and amount of additional space needed for flashback support for a specified time.

All of the database advisors listed above have certain attributes in common. An advisor can be launched in one of two modes depending on how much time is available for completing the advisory task. Either Limited Mode or Comprehensive Mode is the mode in which an advisor should be invoked. Some advisors support both modes.

When an advisor is launched to look at a problem; the more statistics it looks at, the more in-depth analysis it performs and the better the quality of advice it generates. For a long running analysis, the DBA may want to limit running time of the advisor to get a smaller time frame or Limited period. Limited means a relatively shallow analysis; while Comprehensive means a thorough, in-depth analysis taking a much longer time to complete.

A typical tuning session consists of the following tasks:

- Create an Advisor Task (*dbms_advisor.create_task*) - An advisor task is an executable data area in the advisor repository that manages the tuning efforts.
- Adjust the Appropriate Task Parameters (*dbms_advsior.set_task_parameter*) - Parameters are set in the main advisor task, which controls its behavior. Typical parameters are *target_objects*, *time_window*, and *time_limit*.
- Perform Analysis (*dbms_advisor.execute_task*) - Task execution is a synchronous process. Until the execution is completed or interrupted by a user, the control is retained by the database.
- Review the Results (*dbms_advisor.create_task_report*) - The results of the analysis can be reviewed using the above procedure or built-in views. The DBA can accept, reject or ignore the recommendations. If the recommendation is rejected, it is advisable to re-run the analysis using the rejected recommendation as advice for the next analysis operation.

ADDM Performance Monitoring

ADDM in Oracle Database 10g provides proactive and reactive features for monitoring instead of the tedious tuning process in earlier versions. Proactive monitoring is done by ADDM and Server Generated Alerts.

Statistical information is automatically captured from the SGA and stored inside the workload repository in the form of snapshots in sixty (60) minute intervals. These snapshots are then written to disk and are similar to STATSPACK snapshots, except that they are more detailed.

The ADDM also initiates the MMON process to run automatically on every database instance to detect problems proactively. Every time a snapshot is taken, the ADDM triggers an analysis of the period corresponding to the last two snapshots. This helps the ADDM to proactively monitor the instance and detect bottlenecks before they become catastrophic. The analysis

results are stored inside the workload repository. These results are accessible through the OEM console. The ADDM can also be invoked manually from the OEM - Advisory Central page to perform analysis across any two snapshots.

ADDM can be invoked from the OEM screen or manually invoked using the *runad* PL/SQL procedure and by the *$ORACLE_HOME/rdbms/admin/addmrpt.sql* procedure. This procedure has four arguments: *db_id*, *snap1*, *snap2*, and *task_nm*. The *db_id* is the database identifier, *snap1* and *snap2* are the beginning and ending snapshot identifiers, and *task_nm* is the name of the task used to invoke the advisor.

Using OEM to Track Problems Detected by ADDM

ADDM proactively examines the data periodically captured in the AWR and performs analysis to determine any issues. ADDM also recommends solutions and expected benefits from implementing those solutions. The Oracle Enterprise Manager (OEM) homepage enables monitoring of the general health of the database and all functions performed by ADDM.

Automatic Shared Memory Management (ASMM)

The commonly tuned system global area (SGA) components are the database buffer cache, the shared pool, the large pool, and the Java pool. The ASMM enables Oracle Database 10g to automatically determine the appropriate values of these components within the limits of the total SGA. In Oracle Database 10g, the DBA can simply mention the total amount of SGA memory available to an instance using the *sga_target* initialization parameter. Oracle database will automatically distribute this memory among various sub-components to ensure the most effective memory utilization.

When the automatic SGA memory management feature is enabled in the database, the sizes of different components are flexible to resize to adapt to the needs of the workload without additional intervention. The database automatically distributes the SGA among the various components as required, allowing the system to maximize the consumption of all available memory. With manual configuration, it is possible that compiled SQL statements frequently age out of the shared pool due to inadequate size. With automatic SGA management, the internal tuning algorithm monitors the workload performance and increases the shared pool as needed to reduce the number of parses.

The ASMM feature uses a new Memory Manager (MMAN) background process. MMAN coordinates the sizing of the memory components and acts as a memory broker. The MMAN keeps track of all memory components and pending resize operations.

Based on workload information from AWR, the ASMM can capture statistics periodically (60 minutes) in the background, use different memory advisories, and move memory to where it is most needed (large pool or buffer cache). For example in a long running OLTP day job, a larger buffer cache is needed; while for a data warehousing (DSS) batch job at night, more memory is needed for the large pool. The ASMM moves the memory to buffer cache or large pool as needed.

When *sga_target* is set to a non-zero value to enable ASMM, the auto tuned SGA parameters are set to zero values. These components are then automatically sized by ASMM. Whenever an auto tuned component is resized to a larger value, the memory is taken from another auto-tuned component to supplement the change, while the manually tuned components are left untouched.

Using Memory Advisor through OEM

The Memory Advisor can be used only when the automatic memory tuning is disabled. This will help to tune the size of memory structures. The Memory Advisor has three advisors that give recommendations: Shared pool in SGA, Buffer Cache in SGA, and PGA.

Automatic Storage Management

Automatic Storage Management (ASM) is a new database service for efficient management of disk drives with 24/7 availability. It helps the DBA from potentially managing thousands of database files across multiple database instances by creating disk groups. The disk groups are comprised of disks and the files that reside on them. ASM will not eliminate any existing database functionalities with file systems or raw devices, and Oracle Managed Files (OMF) as in previous versions. With it, the DBA only needs to manage a smaller number of disk groups. ASM also serves as a cluster file system for RAC configurations.

Adding ASM will not eliminate any existing database functionality. New files can be created as ASM files, while old files are administered the traditional way. In a nutshell, we can have a mixture of ASM files; Oracle managed files and manually managed files all at the same time. Existing files can be migrated to ASM if needed. For smaller database installations, ASM is not recommended as a quick fix for space management issues.

ASM is responsible for file management and prevents accidental file deletion by eliminating the file system interface. It provides raw disk I/O performance for all files, striping them across multiple storage arrays. It reduces the cost of managing storage with a clustered volume manager and integrated file system functionality. ASM adds the reliability features found in LVM (Logical Volume Managers) such as mirroring protection and eliminating the purchase of third party products. Similarly in a RAC environment, ASM eliminates the need for a Cluster LVM or Cluster File System (CFS).

ASM also offers the benefits of mirroring and striping. An advantage of ASM over conventional VMS is the file based storage reliability policy than the volume based. Hence, the same disk group can have a combination of files protected by mirroring or no protection. ASM does not manage binaries, alert logs, trace files, or password files.

Automated SQL Tuning Features

SQL Tuning Advisor and SQLAccess Advisor are the actual tuning tools which are supported with services rendered by AWR and ADDM.

SQL Tuning Advisor provides tuning advice for SQL statements with out modifying any statement. For complex applications and large databases, SQLAccess Advisor comes in very handy. SQLAccess Advisor is a tuning tool that provides advice on indexes, materialized views and materialized view logs for a given work load. It also provides advice on database schema issues and determines optimal data access paths.

For both the tuning tools, the user has to create a task, run the Advisor, generate the recommendations and implement them. The user can accept or reject the recommendations given by these tools. These Advisor tools are available through Oracle Enterprise Manager 10g (OEM) and from SQL command prompt.

Automatic SQL tuning in Oracle Database 10g is offered by the improved *Query Optimizer*. The improvised features are available to the user through the *SQL Tuning Advisor*.

SQL Tuning Advisor

SQL Tuning Advisor provides automatic tuning advice for SQL statements. It takes one or more SQL statements as input and invokes the automatic tuning optimizer to perform SQL tuning with out actually modifying any statement. The output is a series of advice or recommendations along with the rationale behind each recommendation and its expected benefits. These recommendations will prompt the user to collect statistics on the affected objects, create new indexes, restructure the statements, or creation of new profiles. The user can accept the recommendations or reject them.

For tuning multiple SQL statements, a *SQL Tuning Set* (*STS*) is created. An STS is nothing but a database object to store SQL statements with their execution context. STS can be created from command line or the new and improvised Oracle Enterprise Manager 10g. SQL Tuning Advisor gets its inputs from a variety of sources including ADDM, top SQL statements from AWR, Cursor Cache, SQL Tuning Set and user inputs.

SQL Tuning Sets can be handled through OEM or managed with *DBMS_SQLTUNE* package procedures. The SQL Tuning Set APIs allow you to create a new STS (*DBMS_SQLTUNE.CREATE_SQLSET*), load the STS (*DBMS_SQLTUNE.LOAD_SQLSET*), select the STS and review contents (*DBMS_SQLTUNE.SELECT_SQLSET*), update if needed (*DBMS_SQLTUNE.UPDATE_SQLSET*), create a tuning task with STS as input, and drop it when finished (*DBMS_SQLTUNE.DELETE_SQLSET*). For using these APIs, the developer needs ADMINISTER ANY SQL TUNING SET system privilege.

Deploying SQL Tuning Advisor

We can control the scope and duration of a tuning task done by SQL Tuning Advisor. The scope of the task can be set to *limited* or *comprehensive*. With the limited option, the SQL Tuning Advisor produces recommendations based on analysis of statistics, access paths, and SQL structure. SQL Profile recommendations are not generated. When the comprehensive option is chosen, the SQL Tuning Advisor does everything under limited scope and SQL profiling. In the comprehensive option, the user can also set the time limit of a tuning task, which has a default value of 30 minutes.

SQL Tuning Advisor provides advice based on optimizing the execution plan, the rationale for the new plan, estimated performance benefit, and commands for the changes. As explained before, the users have to make a choice on whether to accept or reject the recommendations to optimize the SQL statements.

To access the SQL Tuning Advisor through the OEM, go to Advisor Central- SQL Tuning Advisor link. The SQL Tuning Advisor can also be invoked through the DBMS_SQLTUNE package by anyone with DBA role and ADVISOR privilege. The following is a list of DBA views to review information gathered for tuning SQL statements. (The DBA role has ADVISOR privileges by default. The developer will need DBA privileges to access these views)

SQLAccess Advisor

When you have an application with complex queries on large sets of data*, SQLAccess Advisor* comes in very handy. For improving the performance, it will recommend a combination of indexes, materialized views, and materialized view logs. These structures can result in significant performance improvements, though they require a considerable amount of time and space to create and maintain.

The SQLAccess Advisor can be run from OEM using the SQLAccess Advisor Wizard or by invoking the *DBMS_ADVISOR* package. Using the SQLAccess Advisor, we can tune SQL statements, manage workloads, recommend materialized views and indexes, mark, update and remove recommendations, and manage materialized views.

SQL Tuning with SQLAccess Advisor

To use the SQLAccess Advisor for tuning, we have to perform four steps -create a task, define the workload, generate the recommendations, and implement them.

A task has all the information relating to recommendation process and its results. To automatically create a task, use the wizard in OEM or the *DBMS_ADVISOR.QUICK_TUNE*. For those who prefer hands-on approach, use *DBMS_ADVISOR.CREATE_TASK* procedure.

The Workload consists of one or more SQL statements and various statistics and attributes for each statement. The SQLAccess Advisor can be used without a workload; thereby generating and using a hypothetical workload based on the dimensions defined in your schema. For best results, a workload must be provided in the form of a SQL Tuning Set, a user supplied table or imported from SQL Cache. The recommendation process (types and naming conventions) and workload customization (duration and filtering) are controlled by SQLAccess Advisor parameters. To set these parameters, use *SET_TASK_PARAMETER* and *SET_SQLWKLD_PARAMETER,* which are valid for the life span of the task or workload object. Then the task is linked to the workload using *DBMS_ADVISOR.ADD_SQLWKLD_REF*.
Once a task exists and linked to a workload, use the *DBMS_ADVISOR.EXECUTE_TASK* procedure to generate the recommendations, which are in-turn stored in the SQLAccess Advisor repository. To view the recommendations, use the catalog views DBA_ ADVISOR_RECOMMENDATIONS, USER_ADVISOR_RECOMMENDATIONS or get a script using *DBMS_ADVISOR.GET_TASK_SCRIPT* procedure. OEM can also be used to get recommendations.

All these recommendations are stored in the SQLAccess Advisor repository, which is a part of the Oracle database dictionary. This repository has a lot of benefits like being managed by the server, support of historical data etc.

Implementing SQLAccess Advisor Recommendations

When the SQLAccess advisor is run from the OEM or by using *DBMS_ADVISOR* packages, it creates a set of recommendations and lets the user decide whether to implement all of its recommendations, or some or none. In OEM, this is achieved by selecting the required recommendations on the screen. Once you have decided to implement the recommendations, you are given a choice on how to implement the recommendations. For do-it-yourself DBAs, the SQLAccess Advisor will generate a SQL script to be used any time. The DBA can also edit the SQL, change naming conventions used etc. in the script. Or you can schedule a job in the OEM to execute the script and implement the recommendations.

SQLAccess Advisor is generally used in large database/data warehouse environments. The advantage of using SQLAccess Advisor in such environments is that the DBA can tune parts of the database using selective workloads. It is a quick and easy to use tool, which helps to build a pattern of usage over a time frame. Also the DBA gets to decide and select the recommendations to be implemented. SQLAccess Advisor is a quick to use tool to be run on any database to identify parts of the system which needs tuning, which would have been left out by ordinary tuning procedures.

Alerts Management Using OEM

Alerts can be managed easily from the Oracle Enterprise Manager console. Oracle has vastly improved the Enterprise Manager Console to manage all aspects of the hardware and software environment, including the Grid control. Oracle Enterprise Manager can be invoked from a Web browser, from client installation or from the database server itself. The Database Control is installed with every Oracle Database 10g system and can be used to monitor and administer single or multiple database instances.

The Enterprise Manager framework monitors the entire Oracle environment and provides detailed system monitoring for timely detection and notification of problems. It makes use of metrics and thresholds for each monitored parameter and uses alerts to provide information. Metrics are units of measurement defined to assess the health of the system under monitoring. Every target has a set of predefined metrics with thresholds associated with them. Thresholds are the boundary values against which the metric values are compared. Once a value reaches its threshold, an alert is generated. Alerts are also generated when a significant change is noted by clearing of a previous alert, change in availability of a monitored service, or when a specific error condition or database action occurs.

The Enterprise Manager provides aggregate information of all monitored targets and collects aggregate performance and availability data over time. Metrics can be compared to determine trends in performance across various subsets of data. Using OEM, you can schedule different monitoring jobs to be run on the database or have the database send you an alert for the specified events.

Wrapping it up

In this paper, we have seen a quick overview of the automatic database administration features of Oracle Database 10g. Detailed explanation of all these features is given in Oracle Documentations, related Oracle Database 10g books, and Rampant TechPress's *Easy Oracle Automation*. By judicious utilization of all the automatic tuning and management features of Oracle Database 10g, a DBA can drastically reduce the time spent on routine database maintenance tasks and make the databases run on 'autopilot'.

About the author

Dr. Arun Kumar R., Systems Architect for Enterprise Data Services at Cingular Wireless, has a decade of architecture experience in Oracle technologies and multiple relational database systems. He also serves as the Associate Editor for SELECT Journal (IOUG) and a columnist for DBTA magazine. He has published papers in many technical journals and spoken at Oracle technology conferences and international seminars. He is the author of "Easy Oracle Automation" from Rampant TechPress. Arun can be reached at aruniou g@dbatrends.com.