Oracle® Diagnostics Pack

An Oracle Technical White Paper

November 1998

ORACLE
Enabling the Information Age™

Oracle Diagnostics Pack

OVERVIEW

Oracle Diagnostics Pack is an optional set of applications built on the Oracle® Enterprise Manager systems management platform. This comprehensive collection of tools provides performance management, including graphical monitoring, problem detection and correction, diagnostics, and long-term planning for the Oracle environment. Oracle Diagnostics Pack includes:

- *Capacity Planner* — A tool for collecting, storing, and analyzing historical performance data for Oracle, the host operating system and other third party databases, enabling customers to plan for future resource needs.
- *Performance Manager* — A real-time, Oracle performance monitor providing graphical views of an array of Oracle and host operating system performance metrics, with the ability to drill down into detailed data of performance diagnostics, such as identifying lock contention.
- *TopSessions* — A tool providing focused, performance diagnostics for Oracle8*i* sessions.
- *Advanced Events* — A set of pre-defined events built on the Oracle Event Management System, offering "lights out" performance monitoring and problem solving for distributed systems.
- *Trace* — An event-based, data collector providing end-to-end tracing of performance data for applications and databases.

This paper examines the functionality and use of Oracle Diagnostics Pack.

ENTERPRISE DATABASE SYSTEMS MANAGEMENT

Today's enterprise computing environment faces unparalleled growth and change. The movement from mainframe systems to downsized, client/server environments creates opportunities for increased flexibility and data access. But this movement also increases system management needs and costs in an environment that lacks established, centralized, management infrastructures. Distributed client/server systems, rapidly expanding applications, very large databases (VLDB), and other advancements add complexity and create the need for a new generation of system management technology.

Many client/server system management challenges have already been met with products that manage distributed hardware and network configurations. But databases and related business applications also require solutions for administering, monitoring, planning, and solving the complex problems of large-scale, client/server systems.

Emerging new products increase the manageability of client/server database configurations, from point solutions for analyzing database structure efficiencies, to large, multipurpose applications that monitor and solve a wide range of database problems.

Oracle® Enterprise Manager Product Family

Oracle8*i* addresses the need for a sophisticated database, system-management environment with the Oracle Enterprise Manager platform and product family. The Oracle Enterprise Manager product family consists of Oracle® Diagnostics Pack, Oracle® Tuning Pack, Oracle® Change Management Pack, Oracle® Management Pack for Oracle8*i*, Oracle® Management Pack for Oracle Applications and Oracle® Management Pack for SAP/R3. All of these applications are fully integrated into the Oracle Enterprise Manager console and provide a unified systems management framework for end-to-end management of the Oracle environment.

Built upon open Internet standards such as Java™, CORBA, and IIOP, these products provide the first management framework designed to support Internet computing. All Oracle Enterprise Manager applications can be accessed from anywhere using a standard Web browser. A reliable and scalable multi-administrator repository leverages the administrative staff by providing cooperative management. Using the Oracle Enterprise Manager product family, administrators and IT managers can increase productivity, deliver better services, and reduce the total cost of information systems.

The Goal: Systematic Performance Management

Improved performance relies on sophisticated tools to provide comprehensive and customizable database monitoring, planning, problem solving, and diagnostics. Oracle Enterprise Manager's functionality meets these management needs through an optional set of applications known as Oracle Diagnostics Pack.

ORACLE ENTERPRISE MANAGER DATA COLLECTION FRAMEWORK

To effectively monitor the Oracle environment, diagnose problems, or plan for future resource needs, decisions must be based on performance and usage data, both real-time and historical. Keeping an Oracle system running at peak performance requires a detailed understanding of the demands users make on the system.

Administrators can readily collect and examine performance data, such as CPU use, disk input/output (I/O), memory used, or number of users logged on. But many find it difficult to collect, store and analyze this data with minimal impact on the system. To meet these management needs, Oracle has extended the Oracle Enterprise Manager framework to include a low-overhead data collection mechanism.

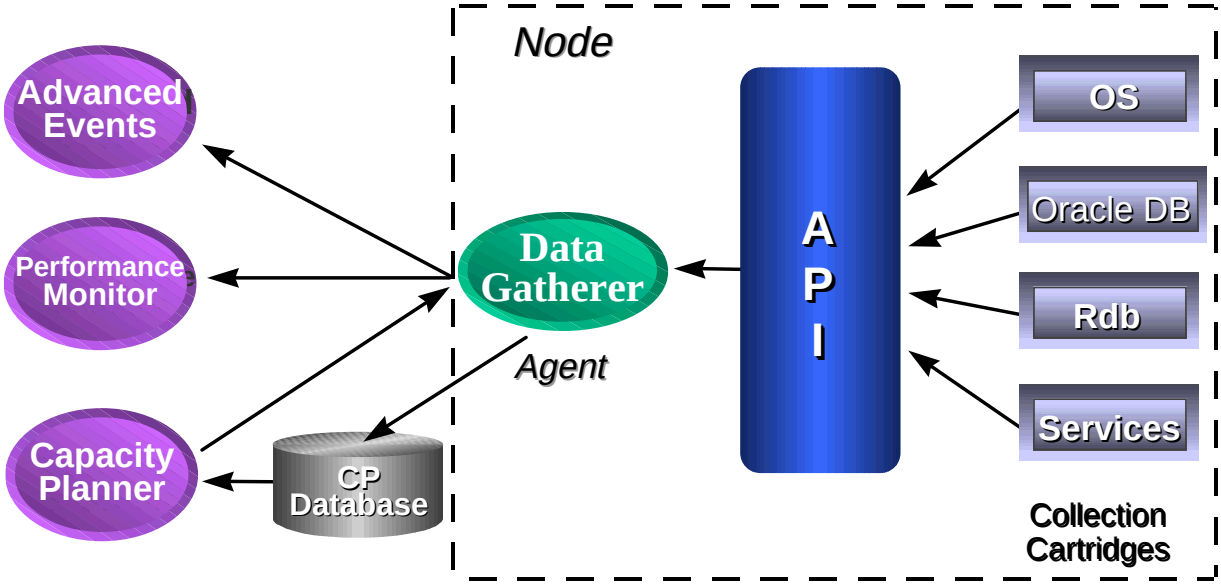

Figure 1: Oracle Enterprise Manager's Data Collection Framework is Extensibleand Has a Low Overhead

The Oracle Enterprise Manager data collection framework:

- Collects performance data sampled at a user-specified frequency.
- Automatically loads the collected data into an Oracle database at user-defined intervals.
- Provides flexible management of the collected historical data, including aggregation, rollup, and aging out of old data.
- Collects data for the operating system, Oracle8*i* server or third party databases such as Microsoft SQL Server or Informix.
- Supports user-defined performance metrics.
- Opens application program interfaces (APIs) for extending the framework to collect data for any managed service or application.

Collecting Performance Data

The data collection framework uses the Oracle data gatherer, an extension of the Oracle Enterprise Manager Intelligent Agent. Installed on each managed node, the data gatherer handles requests from client applications to collect data. Client applications using the data gatherer include Capacity Planner and Performance Manager. Each client application specifies the performance data it needs, e.g., file I/O or CPU usage, and the time interval between samples. The data gatherer then sends the requested data to each client at the proper interval.

Data Collection Options

Data collection options installed on the managed node determine the performance data available for collection. The data collection option defines the performance metrics available for a particular service or application and the access method for maximum flexibility in collecting performance metrics.

Oracle Enterprise Manager ships with collection options for the Oracle server, the host operating system,Rdb and third party databases, such as Microsoft SQL Server or Informix.

Extensible Collection Architecture

The data gatherer includes an open interface for "plugging-in" collection options to allow anyone to develop new options for collecting performance data. This extensibility greatly increases the data collection framework's ability to manage and monitor complex Oracle environments.

Loading and Storing Historical Performance Data

The data gatherer stores data in a binary file on its host, then periodically loads the data into an Oracle database. The user can specify the historical database to hold the loaded data and specify the load schedule. For example, users can collect and load all new data once a day at midnight.

The loader also automatically aggregates the data stored in the historical database. For example, if the data gatherer collects data once every 60 seconds and loads each 60-second sample, it then averages those samples into an hourly aggregate value. When 24 hours of samples have been aggregated, the data gatherer averages those samples into a daily aggregate value. This aggregation continues through granularities by week, month, and year.

The loader also ages out data based on user-defined policy. For example, data sampled every minute may be retained for one week, hourly data for one month, and daily data for six months. The loader decides which data is old and purges it based on this policy. This capability allows users to control the amount of disk space used by the historical database.

Using the Historical Data

The data collection framework allows customers to easily collect, store, and manage in-depth data about the behavior and use of their Oracle systems. This data helps administrators understand the system they are managing and forms the foundation for tools that monitor the Oracle environment, diagnose potential problems, and plan for the future.

CAPACITY PLANNER

Capacity Planner allows administrators to specify the performance data to be collected, collection interval, load schedule, and data management policy. Administrators can also use Capacity Planner to run in-depth analyses and reports, format the data into easy-to-use graphs and reports, and predict future resource needs.

Specifying Collections

The intuitive graphical interface in the Capacity Planner makes it easy to browse the performance metrics available for each node. In its first release, Capacity Planner collects data for the Oracle server and host operating system and categorizes the performance metrics to make selection easy. Administrators can select typical metrics like CPU usage or file I/O up to the most intensive "top" sessions.

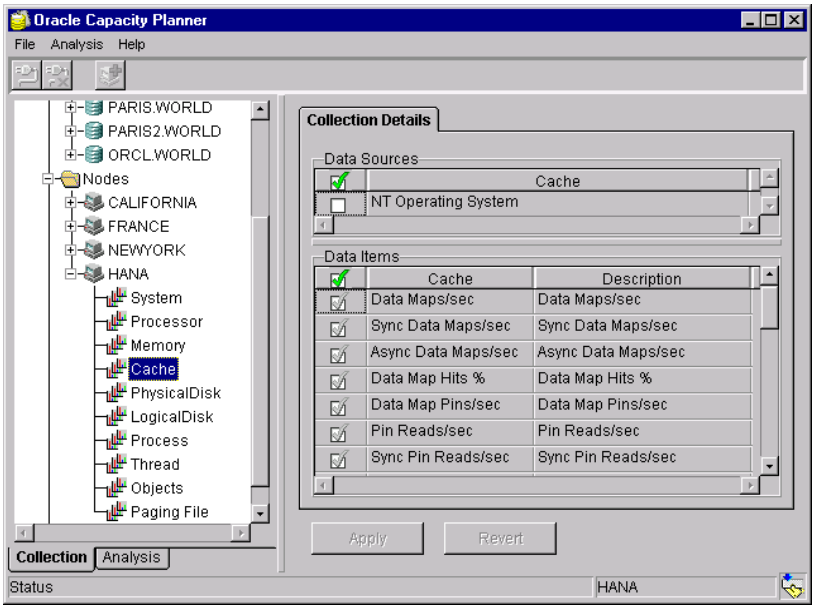

Figure 2: Capacity Planner Collects a Wide Variety of Performance and System data for Both the Server and the Operating System

Loading and Data Management

Capacity Planner sets up the load and data management policy, using the loader page in the figure below. A similar dialog box is used to specify how long to retain each type of data.

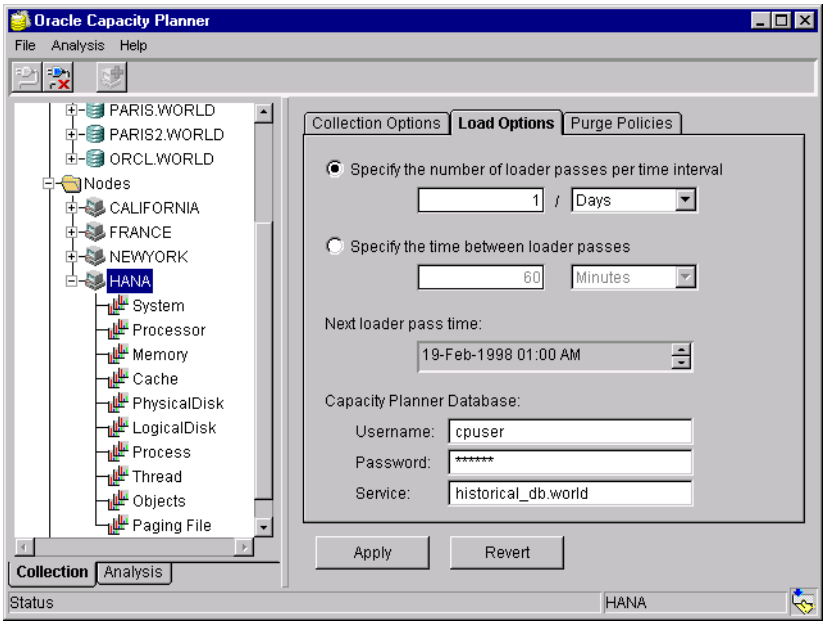

Figure 3: Capacity Planner Specifies the Schedule for Loading Historical Data to Best Fit Customer Needs

Browsing Historical Data and Creating Graphical Reports

Capacity Planner allows administrators to connect to the historical database and view performance metrics for each node and service. Administrators can select the specific data they want to view by browsing a tree listing or navigator, then Capacity Planner displays the data in easy-to-read graphs or tables. These graphs and tables help build in-depth reports to chronicle system performance over time.

Planning for the Future

Capacity Planner helps interpret data and extrapolate performance metrics. For example, administrators can extrapolate to a particular value (e.g., when will my disk be 75 percent full?) or a point in time (e.g., how full will my disk be on May 1?). Capacity Planner plots extrapolation analyses on graphs, making them easy to interpret and add to reports.

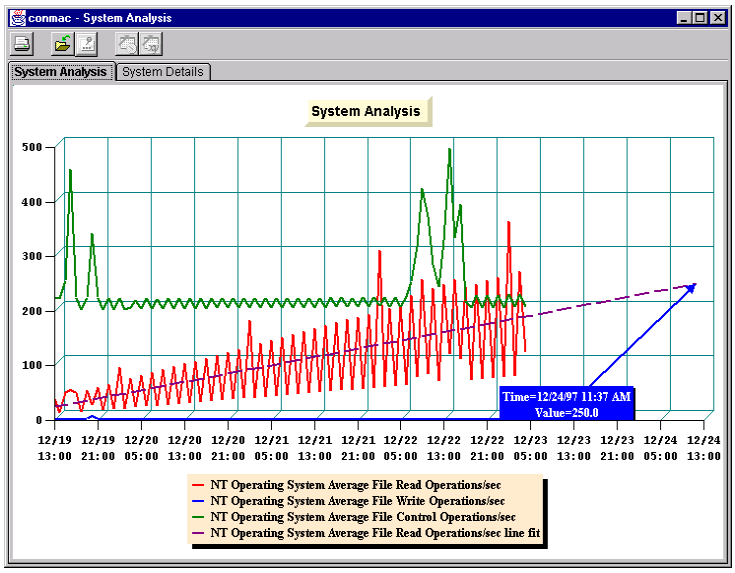

Figure 4: Predicting When File Reads Will Reach a Specific Value

PERFORMANCE MANAGER

A complex, large production Oracle database requires monitoring in several areas to sustain high system performance. Performance depends on many factors, including effective use of memory, minimizing disk I/O, and avoiding resource contention. Performance Manager captures, computes, and presents performance data to enable administrators to focus on key metrics and balance all these factors.

The pre-configured graphical monitor in Performance Manager monitors applications in real time, captures and displays in real-time, and can be recorded for replay. The monitor can be customized to display information in two- or three-dimensional graphical views, such as tables, line, bar, cube, and pie charts. Administrators can also extend the system by defining charts for database scripts or displaying information from Oracle tables, or customize the rate of monitoring.

Using the Data Collection Framework

Both Capacity Planner and Performance Manager work within the Oracle Enterprise Manager data collection framework. But the data gatherer sends collected data directly to Performance Manager in real-time. This framework allows users to monitor performance metrics for both the Oracle8*i* database, host operating system, and third party databases. Administrators can also define custom performance metrics or add extensions to the framework. Both will be automatically displayed in Performance Manager. For example, when a new data collection option is installed, that data is immediately available in Performance Manager monitors.

Monitoring Memory Performance

Performance Manager tracks real-time memory performance, allowing administrators to tune and correct poor memory usage, such as under-allocation of system memory. For example, the Parse Ratio Chart measures an application's success at finding available, parsed SQL in the database's library cache buffer. A low hit rate may indicate that the Shared Pool Memory Allocation is insufficient.

Performance Manager's Buffer Cache and Library Cache Hit Ratio Charts graphically point out memory allocation problems, such as the percentage of failed attempts by a business transaction to find the desired data in memory. Performance charts can also be linked, allowing the user to drill down in a logical progression of analysis. For example, if the Library Cache Hit Ratio detects a problem, the DBA can drill down to the Library Cache Details Chart to investigate further. The Details Chart contains statistics, such as the number of reloads required to process statements in the SQL area. Excessive reloads can indicate the shared pool size should be increased.

Other memory-monitoring charts include the Data Dictionary Cache Hit Ratio, Memory Allocated, and the Sort Hit Ratio.

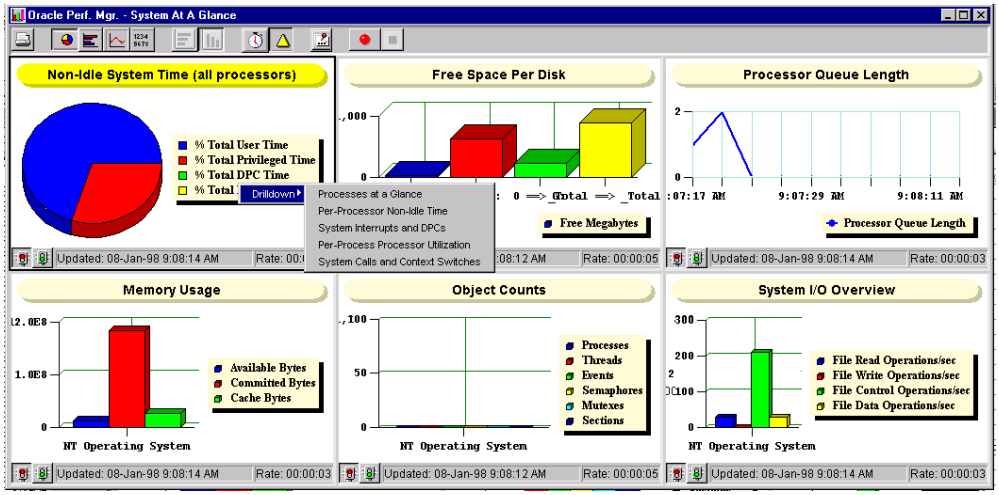

Figure 5: *Viewing Overall System Performance and Drilling Down on a Problem Area in Performance Manager*

Monitoring I/O

The DBA must also focus on minimizing I/O operations. A disk I/O accesses data more slowly from storage than from memory buffers. Proper memory management helps, but the DBA must also effectively use disk space for database objects, and tune database operations involved with I/O management, such as the number of database-writer processes and use of sort-direct writes.

To help analyze database I/O activity, Performance Manager graphically displays I/O performance metrics. For example, the File I/O Rate Chart depicts the number of physical reads and writes per second for the database files. If more details are required, the DBA can drill down to the File I/O Rate Details Chart for statistics on the physical reads and writes for individual files. This pinpoints the files generating the most I/O activity, allowing the DBA to tune database-sort operations and re-size the sort cache.

Further investigation is possible by examining file I/O counts per database session. This drill down enters the TopSessions application, which automatically displays the database sessions generating the highest file I/O activity.

Monitoring Contention — Rollback Segments, Redo Log Files, and Database Locks

The Performance Manager also detects problems related to the top three suspects in resource contention:

- *Rollback segments* — Database contention occurs when there is an insufficient number of rollback segments or if rollback segments are not properly sized or distributed over devices. The Performance Manager's Rollback Nowait Hit Ratio Chart detects the incident of transactions waiting for rollback segment headers, a key indicator of contention.
- *Redo log files* — Redo log files protect against the loss of data in case the system fails. Performance Manager monitors redo log performance in several ways. The Redo Allocation Hit Ratio Chart displays percentages of redo-allocation hits and misses. In multi-threaded server environments, Performance Manager's Shared Server Chart detects contention for dispatcher and shared-server processes by displaying queuing activity.
- *Database locks* — Database locks protect data integrity, but sometimes cause performance bottlenecks. Performance Manager can identify and eliminate troublesome locking situations. The graphical displays include details such as the locking user, lock type, object locked, and mode held and requested. After careful investigation, locked sessions can be easily terminated by the Performance Manager's "kill session" feature.

A Flexible Monitor

Performance Manager also provides Database Instance and Overview monitoring views. The Database Instance charts afford a comprehensive display of instance-usage statistics, including system statistics, tablespace status, and database-user statistics.

The Overview Charts provide a neat, composite view of many Oracle performance characteristics, all visible in a single window. The default set of Overview Charts includes instance-usage activity, such as number of logged-on users, active users, and users waiting. It also includes performance metrics such as buffer hits, system I/O rates, and rollback-segment contention. The Overview Charts can easily monitor database performance at a glance. A customized overview can also be created by defining a combination of monitor charts for display in a single window, including pre-defined and user-defined charts.

User-defined charts can be created for virtually any data in the database. Performance Manager provides dialog boxes for entering the SQL to retrieve the data, defining operations to be performed on the data, and selecting the type of chart best suited to graphically display the data.

Reacting to Performance Problems

Performance Manager corrects problems by using Oracle Enterprise Manager. For example, buffer size parameters can be reset using Oracle Enterprise Manager Instance Manager application. Storage parameters can be reset to resolve I/O or contention problems using Oracle Enterprise Manager Storage Manager application. Oracle Enterprise Manager Tablespace Manager application can analyze problems further and reorganize data storage.

More detailed diagnostics data can be obtained with Oracle Diagnostics Pack's TopSessions and Trace. A comprehensive tuning analysis can also be conducted by Oracle Tuning Pack's Expert™, an automated performance-tuning application, which produces recommendations and scripts for improving performance.

TopSessions

Performance Manager's TopSessions goes beyond general monitoring to provide a focused view of database activity by database session. TopSessions extracts and analyzes sample dynamic Oracle performance data by session, automatically determining the top Oracle users based on selection criteria, such as memory, CPU usage, or file I/O activity.

TopSessions provides two views of session data: an Overview and a Session Details view. The application starts with an Overview of the top 10 sessions connected to the database instance, with a default sort based on session PGA memory usage. The Overview displays data such as session ID, node, application, user name, last session command executed, and the status of the session (idle, active, blocked, or killed). Displays can be customized by changing the number of monitored sessions or the type of statistical filtering and sorting.

The Session Details display allows DBAs to drill down into a particular session to see detailed displays of general-session information, session statistics, cursors, and locks. The Session Details General Page adds information such as identifiers for the schema, SQL, datafiles, rows, and blocks. The Statistics Page displays detailed performance statistics for the session captured from the V$SESSTAT view.

The Cursors page provides information on the session's shared cursors, including SQL Statements and Explain Plans. Administrators have the option of displaying the session's currently executing SQL statements, or displaying all the session's former or future SQL statements. The Session Details Locks Page displays information about the database locks held or requested by the session.

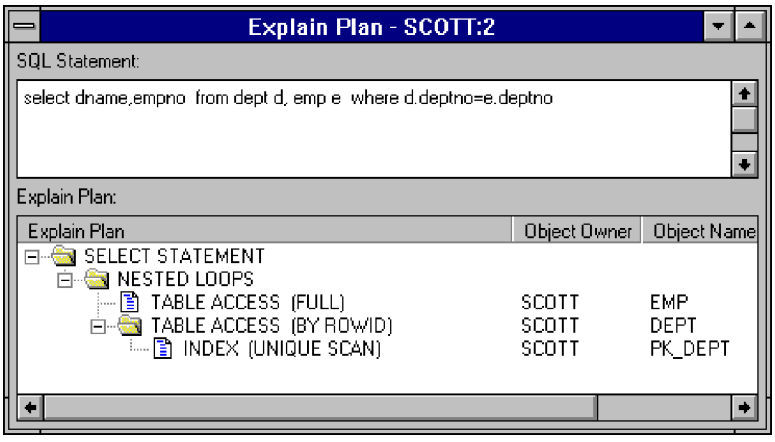

Figure 6: Using TopSessions to View the Explain Plan for a Problem Session

More than one TopSessions display can be opened to monitor multiple instances. The information displayed is static until refreshed. The default refresh cycle is set for 10 seconds, but the user can determine the frequency of the refresh and whether it is manual or automatic.

Combining TopSessions with Performance Manager produces a tremendous amount of practical information for managing the Oracle database. For example, the user can drill down from Performance Manager's File I/O Rate Details Chart into TopSessions to display and investigate the details of the top I/O generating sessions. If the user suspects problems with high disk sort volumes, then SQL statistics for the sorts to disk metric for the session can be examined. Or the open cursors for the session can be investigated to examine the SQL statements and explain the plans executed by the session to identify SQL tuning opportunities.

AUTOMATED MONITORING AND PROBLEM DETECTION

Total performance monitoring must include the ability to monitor remote systems. Oracle Enterprise Manager provides an Internet computing architecture to manage remote systems from the Oracle Enterprise Manager console. The Intelligent Agent can monitor services, manage database operations on remote systems, and launch other programs or operations on remote nodes.

The Event Management System uses the Intelligent Agent to autonomously monitor and react to problems detected on managed servers. Oracle Diagnostics Pack includes a collection of over 45 pre-defined Advanced Event Tests that run on Event Management System and can be launched from the console.

Database Performance Event Monitoring

Oracle Diagnostics Pack's Advanced Events monitor key database performance metrics. Pre-defined event tests include:

- *I/O monitoring* — This covers disk I/O rates and Net8™ I/O rates. Users can specify the I/O rate threshold and receive a warning when the metric is exceeded.
- *Memory-structure performance monitoring* — This covers hit rates for the library cache, data dictionary, and database buffers.
- *User program response time* — This monitors any statistic captured by V$SYSTAT, the dynamic performance table. Users can define a specific numeric threshold, or a threshold representing a maximum delta value between two consecutive samples for a certain V$SYSTAT statistic.

  For example, the V$SYSTAT table provides a running total of user rollbacks — an expensive database operation. Users can set an alert point to allow a maximum value of user rollbacks, or monitor the growth in user rollbacks by measuring the delta value of user-rollback metric samples.

Space Management and Resource Usage Event Monitoring

Advanced Events can be used to monitor the status and performance of Oracle storage structures. Users can set a threshold for the maximum allowable number of extents for a specific segment or set an alert to occur when contiguous free space is reaching a hazardous limit. Users can monitor opportunities for index rebuilding or manage resources more effectively by setting thresholds for maximum users, processes, locks, and datafile size.

Autonomous Node Monitoring

Advanced Event Tests for node monitoring can detect problems with excessive CPU utilization, excessive CPU load or paging, and disk capacity problems.

User Defined Event Test Thresholds and Monitoring Intervals

Advanced Event Tests can be customized with two types of user-defined thresholds: event test threshold and the occurrence threshold. Exceeding event test thresholds can be set at a warning level or an alert level, then tracked against a user-defined, occurrence threshold. Users are notified when the occurrence threshold for the event is reached. Users can also specify how often the Event Management System checks for the existence of an exceeded, event test threshold. For example, users can establish an event test for monitoring Data Dictionary Cache performance, and ask for a warning after three occurrences of the "data dictionary" ratio exceed 10 and 15 percent, respectively.

Advanced Events

Advanced Event Tests can be grouped together to form "Events" — reusable, logical configurations of pre-defined event tests. For example, administrators can define an Event for "SGA Monitoring," and include events such as Library Cache Hit %, Buffer Cache Hit %, and the Data Dictionary Cache Miss %. Once the Event is created, it can be registered, or launched on any database services managed by Oracle Enterprise Manager.

Multifaceted Event Notification

Registered Events can be configured to notify administrators via pager, e-mail, a visual alert to the console or by creating a Simple Network Management Protocol (SNMP) trap. A SNMP trap notifies a third party, or an SNMP-based console, such as HP Openview. Multiple administrators can register for event notification with the console.

Automated Problem Correction

Advanced Events can also be configured to automatically correct the problem event. Using a "Fixit Job," a predetermined action will automatically occur when an event test level is reached. An administrator creates Fixit Jobs using the Oracle Enterprise Manager Job System. For example, a Fixit Job can be created to increase the size of a datafile, then take corrective action in response to an alert situation for theDatafile Limit event test.

ORACLE TRACE

Performance monitoring applications collect data based on sampling techniques. The data is collected at pre-set intervals regardless of the system activity being monitored. By applying statistical analysis, the collected data can represent the real activity with a reasonable degree of accuracy.

This type of data collection provides an efficient, statistically valid mechanism for many performance monitoring applications. Performance Manager and TopSessions use this technique by periodically collecting data from the Oracle dynamic performance tables, increasing monitoring depth by shortening the sample refresh cycle.

For applications that require precise data, Oracle Diagnostics Pack's Trace collects data for each and every key event occurrence in the monitored application. Trace provides an entire census of event data, rather than a sample. This allows performance problems detected through sampling techniques to be pinpointed to specific occurrences.

Trace provides an open API to monitor events in any software product. For example, specific events have been identified in the Oracle Server, such as the SQL statement parse, execute, and fetch. Each of these SQL events have been instrumented with Trace API calls. These calls are activated when the event occurs during a

scheduled Trace collection. An "event" can also be an application transaction, such as a "deposit" in a banking application. Any product can be programmed with Trace API calls for event-based data collection.

Information collected by Oracle Diagnostics Pack's Trace can include:

- System resources used by events, such as CPU time, memory usage and pagefaults.
- Elapsed time and frequency of events.
- Application-specific data for events, such as user name, transaction type and dollar amount.
- Tracking events occurring within other events, such as all SQL statements executed within a single logical transaction.
- Tracking events related across products, such as a transaction that spans an application and a database server.

OPEN API AND DATA COLLECTION SERVICES

Trace provides a simple, open API used to identify application events, activate event collections and pass event data to the Trace collection services. The collection services consist of runtime libraries linked to an application using the API and a set of control files to track the status of instrumented application processes and runtime collections.

Applications do not have to be instrumented in order to use Trace. Most Trace users collect data for pre-instrumented applications, and use the collected data in another monitoring or analysis tool.

Trace efficiently buffers and passes event data to collection files, where it can be formatted to Oracle tables for access by performance tools, report writers, and user scripts. Trace data collected for the Oracle server can also be easily accessed through the new Trace Data Viewer.

Oracle Server Event Instrumentation

The Oracle server uses the Trace API to trace and collect data on key database events, including: logical and physical database transactions; SQL events such as the parse, execute, and fetch; and the complete set of Oracle database waits. Event data includes resource usage, such as I/Os and CPU usage, and a wealth of database statistics specifically for the database event.

Trace Manager

Oracle Enterprise Manager includes a management application to control Trace collections on Oracle Enterprise Manager nodes. Oracle Diagnostics Pack's Trace Manager runs on the Oracle Enterprise Manager console and uses the Intelligent Agent and Job System for remote discovery of Trace-enabled products. Trace Manager searches Oracle directories on target Oracle Enterprise Manager nodes for Trace product definition files. These files contain the descriptions of events and data for products instrumented with the Trace API and can include Oracle products and third-party and end-user applications. These products report to the Trace Manager application, and are updated to users for Trace collection scheduling.

For improved focus and efficiency, Trace collections can be filtered for specific database users. For example, all SQL activity associated with a particular user can be collected over a specific time period. Or any of hundreds of types of server wait events, such as all waits associated with database file I/Os, can be filtered for collection to allow more effective troubleshooting.

Trace makes collected data accessible through a data formatting process, by converting the collection files to relational format and loading them into Oracle tables. The Trace application creates these tables dynamically, as required. If a new product event is being formatted, a new table will be created automatically.

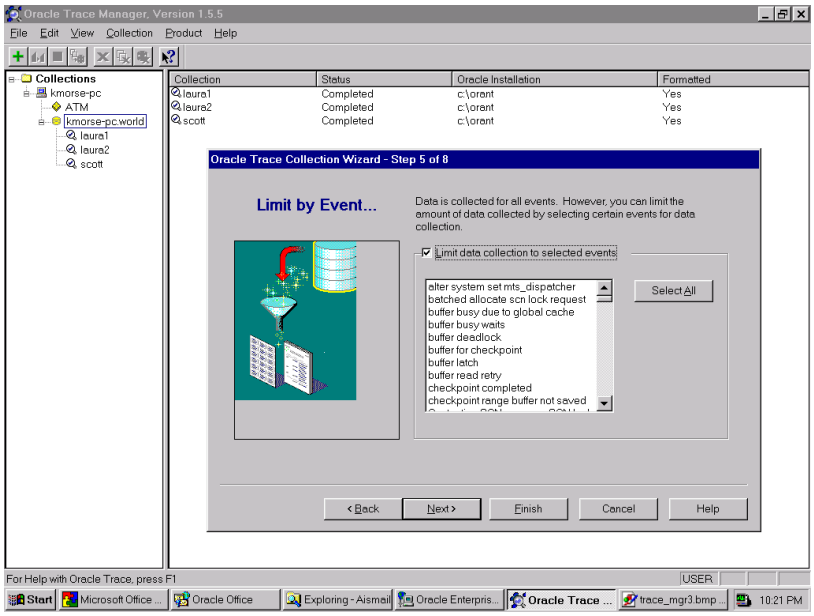

Figure 7: Specifying the Parameters for a Trace Collection

Trace Manager uses the advanced scheduling features of the Oracle Enterprise Manager Job System. Trace collections can be scheduled to occur at flexible intervals and duration on remote nodes.

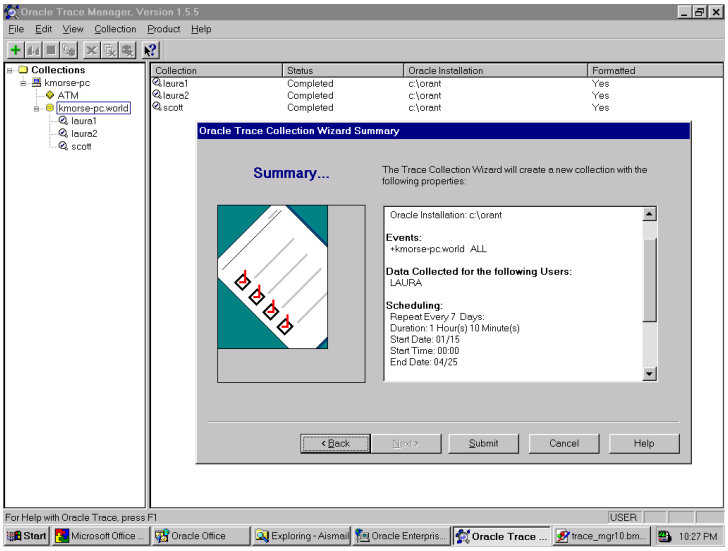

Figure 8: Using the Collection Wizard to Schedule a Trace Collection

Customizable Event Sets

Subsets of a Trace-enabled product's events can be grouped into "event sets," when run in the advanced "administrators mode." This enables administrators to schedule a focused, more manageable collection. For example, the Oracle8*i* server product definition files contain an event set for an Oracle Tuning Pack's Expert workload collection, a subset that captures SQL workload details. This data can then be formatted and loaded into Expert for analysis and tuning.

Trace Data Viewer

A Trace server collection typically generates a large volume of valuable information that can be used to troubleshoot and tune the Oracle server. But since large amounts of data can be difficult to process and sort, Oracle Diagnostics Pack includes a new application for easier access to data from Trace collections. The Trace Data Viewer handles the complex task of processing a large Trace collection, extracting and aggregating key server performance metrics.

The Trace Data Viewer automatically locates and presents Trace collections for processing and presentation. For example, an administrator can select a Trace collection and choose to process SQL metrics and/or server wait events metrics. The Trace Data Viewer then runs through all of the Trace formatter tables, extracting, processing and aggregating performance data, then displays this data in a comprehensive set of Trace "data views." Trace organizes the data views into performance metric categories such as I/O, Elapsed Time, Row Processing, and CPU usage.

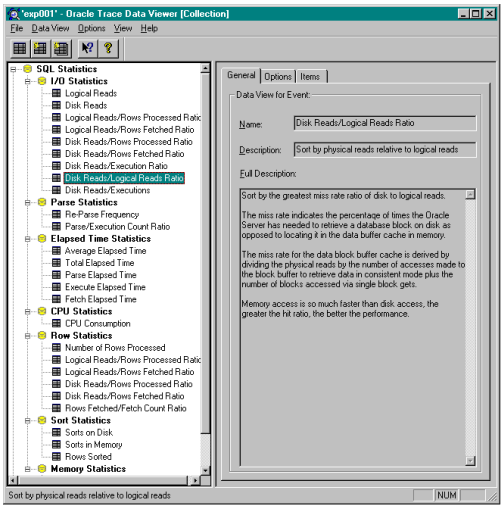

Figure 9: Browsing Trace Statistics in Trace Data Viewer

Each of these categories includes a set of pre-defined data views including: Average Elapsed Time per SQL statement, Logical Reads to Rows Processed Ratio, and Rows Fetched to Fetch Count Ratio. These metrics are calculated for every SQL statement that occurred during the Trace collection, so administrators have absolute assurance that no important SQL was missed. The SQL having the greatest impact — as measured by elapsed time, CPU usage, I/Os, or other metrics — rises to the top of the collection for troubleshooting and tuning.

Once the SQL is presented through the selected data view, administrators can examine some important details, similar to those found in the TKPROF utility. Administrators see a complete breakdown of the SQL processing operations for each SQL statement, including details such as the number of parses, executes and fetches, resources used by each operation, elapsed time per operation, and library cache misses.

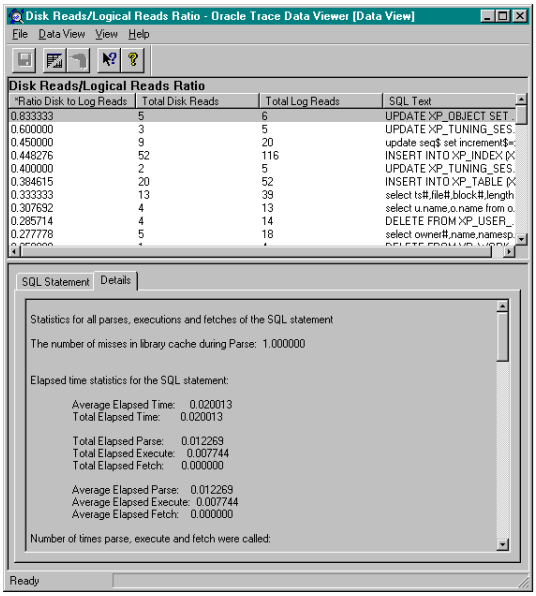

Figure 10: Examining Specific Performance Statistics in Trace Data Viewer

Trace Data Viewer allows users to customize any of the pre-defined views through a "create-like" feature. This feature can be used to add additional data fields to the view, change the selection and sorting criteria, or drill down to examine more details.

High impact SQL statements can be copied from the Trace Data Viewer to Oracle Tuning Pack's SQL Analyze application for tuning.

QUANTIFIABLE GAINS

The applications in Oracle Diagnostics Pack provide the administrator with a set of sophisticated tools to manage Oracle performance, including monitoring, "lights out" problem solving, diagnostics, historical data collection, and planning. By using the tools in Oracle Diagnostics Pack, the Oracle administrator can make significant gains in productivity, manageability, and database performance.

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

Worldwide Inquiries:
+1.650.506.7000
Fax +1.650.506.7200
http://www.oracle.com/