MAXGAUGE for Oracle Web Version 5.3
PRODUCT DOCUMENTATION
MAXGAUGE for Oracle
The New MaxGauge is now Faster, Simpler, and Easier to Perceive!

With improved UX, user friendliness is always our top priority.

Monitoring is done in real-time from an integrated perspective and it provides most important & detailed information through various performance data.

**REAL TIME MONITORING**
Real-Time Monitoring of Various Performance Stats

**PERFORMANCE TREND**
Top-Down Trend Analysis of Macro & Micro Data

**SMART ANALYSIS**
Effective Performance Analysis through Performance Statistics

**POWER COMPARISON**
Multiple Performance Comparison Analysis from Various Perspectives

MaxGauge is a professional software solution for database performance management and helps to effectively manage the database system’s availability and performance. It provides pro-active monitoring techniques which help to recognize problems quickly and analyze bottlenecks in applications effectively, and stores up various performance data to provide a clear analysis report about the effects in comparison to the resource investment.
Collects All Performance Data at **Minimum Workload** for Monitoring and Storage
Collects All Performance Data through Direct Memory Access Method

When a complicated performance problem occurs, monitoring and analysis are disabled

Even with a database hang, monitoring and detailed analysis can be done
Intuitive and Easy User Interface
Database Optimization through a Convenient **TOP-DOWN Approach Method**
All Activity History Data which can be Collected are Stored in 1 Second Unit

You can analyze the past history in unit of second just as if for real-time.
REAL-TIME MONITOR

OVERVIEW
VIEW TYPE
METHOD
FRAME
MENU
ICON
TOOL
CONFIG
Optimized **Monitoring Patterns** for Database Performance Management

**SINGLE VIEW** / **TREND VIEW** / **RAC VIEW** / **EXA VIEW**

- **SINGLE VIEW**: Intensive Monitoring on a Single Database
- **TREND VIEW**: Optimized Simultaneous Monitoring of Multiple Databases
- **RAC VIEW**: Effective RAC Monitoring by Adding a Specialized Frame
- **EXA VIEW**: Optimization of Monitoring EXADATA Key Management Items
Provides a Screen Exclusively for **RAC Monitoring**

**VIEW TYPE**

**RAC VIEW**

Through the RAC View, you can easily check the **Global Lock**, **Load Balance**, and **Network** usage information.

**FEATURE**

- Network (Bandwidth, Link)
- Global Lock (Enqueue, Library Cache Pin, Library Cache Lock)
- Load Balance (CPU, Free Memory, Active Session, Physical Reads)
Provides a Screen Exclusively for EXADATA Monitoring

VIEW TYPE

EXA VIEW

The EXA VIEW provides the main stats generation rate such as Smart Scan and Storage Index which are necessary for EXADATA monitoring. It also provides the Cell Server’s Resource Usage and information regarding Disk/Flash, which not only help to serve the purpose of DB monitoring but, even providing an effective control over the EXADATA platform.

FEATURE

- Smart Scan
- Storage Index
- Disk Request, I/O
- Flash Request, I/O
- CELL Server Resource (CPU, Memory)
Drill Down Based on **Top Down Approach**

**MONITORING METHOD**

**TOP DOWN APPROACH**

Through the Top Down Approach which works through **System > Session > SQL**, you can easily identify the components that are negatively affecting the database in real-time.

**FEATURE**

- Process (Session) Connection related to Each Stat
- Session Manager Connection
- Session Detail Connection and SQL Confirmation (Kill Session, Trace)
Provides Various **Performance Stats TRENDS**

**TREND CHART**

The TREND CHART provides a chart format monitoring screen by using stats related to the database activities and delays. By default, it provides **6 Main Performance Stats for monitoring**, however you can change or expand the number of performance stats as desired depending on the system or nature of the business.

**FEATURE**

- Frame Docking
- Oracle Wait / Stat / Ratio / OS Stat Change
- Manage Graph / Bar Format Change
- Scale Up Feature Included
Intensive Monitoring of **Wait Event Status**

**WAIT EVENT**

- **Total Wait** provides Wait Time by Trend and Wait Event of the Total Wait Time, by each Instance. **Active Session Wait Class Trend** provides the number of sessions in each group of active session wait events grouped by class.

**FEATURE**

- Frame Docking
  Provides Wait Time by each Wait Event
  Graph / Bar Change Possible
  Top Session Connection
  Provides Number of Session Related to each Wait Class
Real-Time Diagnosis of SQL Elapsed Time

You can check the SQL Elapsed Time Distribution by instance. You can also check detailed information such as STAT/Plan/Bind Value of SQLs in the mouse-drag selected period.

FRAME

SQL ELASDED TIME

You can check the SQL Elapsed Time Distribution by instance. You can also check detailed information such as STAT/Plan/Bind Value of SQLs in the mouse-drag selected period.

FEATURE

Frame Docking
Provides SQL List by Elapsed Time
Provides SQL Full Text, SQL Plan, Bind Value
Provides **Comparison Analysis** Feature by **Linking with PA**

**24 HOUR TREND & COMPARISON**

You can compare the performance trends of today and yesterday or any select dates and intuitively identify the performance issues.

**FEATURE**

- Frame Docking
- Manage Stats Selection
- Manage Date Selection
- PA Performance Trend Connection
Lock / PQ Monitoring Feature

**LOCK TREE & PQ SESSION**

You can use the Lock Tree to check the relationship between lock holding sessions and wait sessions in a tree structure. You can also check information such as SQL and Dead Lock generation status and Lock Type information. Furthermore, it displays the relationship of coordinator and slave in a tree format in regards to the issued Parallel SQL, and job distribution among the slaves can be monitored.

**FRAME FEATURE**

- Frame Docking
  - Provides Lock Wait Session and Holder Session
  - Session Detail Connection (Kill Session, Trace)
  - Provides the Relationship of Coordinator and Slave
  - Provides Parallel SQL’s Elapsed Time and Wait Time
Check **Blocking Session** Information with One-Click

Not only the transaction related locks (TX/TM), but even for the lock types whose holder information cannot be known through a general method, you can use the **Blocking Session** link feature to trace the root cause. Once you connect to the Session Detail screen, you can execute session control functions (Kill Session, Trace).

**FEATURE**

- Provides **Blocking Session**
- Session Detail Connection (Kill Session, Trace)
Simultaneous PEAK Period Monitoring and Real-Time Diagnostics

REAL-TIME DIAGNOSTICS

Provides a feature which quickly diagnoses the problem in the Resource Peak period by using the Previous 3 Minute Data without connecting to PA.

FEATURE

Check previous DB Status up to 10 minutes maximum from the current time
Provides SQL Elapsed Time, Trend Chart, Active Session and Lock Tree
Move from one time point to another in unit of 3 seconds
Provides Wait Event Description Feature

**TOOL**

**EVENT DESCRIPTION**

During the monitoring process, you can use the Wait Event Description link feature to obtain information such as the Wait Event Description and solutions regarding the wait events currently being generated.

**FEATURE**

- Connect to Active Session Grid
- Wait Event Name Search Feature
- Provides the Root Cause of Wait Event Generation
- Provides Parameter Description
Various Support for **User Management**

**CONFIGURATION**

**ACCOUNT MANAGEMENT**

- Use the Account Management feature to manage the user account settings. You can set the connection IP Band, Database List and Privileges.

**FEATURE**

- Manage Account Password
- Set Connection IP Band by Account
- Manage and control instances for monitoring by Account
- Manage user privileges of executing SQL Text, SQL Bind, Session Kill and SQL Trace for each instance.
Alert Settings Feature for All Performance Stats

CONFIGURATION

ALERT MANAGEMENT

You can set alerts for all performance stats and wait events available in the oracle for an effective monitoring. You can also set alerts for Main OS Performance Stats and File System Usage, Alert Log and Table Space.

FEATURE

- Manage Threshold Settings by Instance
- Provides Two Types of Alert Level (Warning, Critical)
- Manage Check Cycle
- SMS Connection
SMS Connection Feature for the Preset Alerts

**CONFIGURATION**

**SMS MANAGEMENT**

Provides a feature for connecting with SMS service by using the preset alert information. You can manage the hours open for SMS reception and the hours SMS is not accepted, and select whether to receive SMS or not by user group.

**FEATURE**

Manage the settings for SMS Notification Period
Configure to set the SMS Notification Days via Weekly Schedule Setup Feature
PERFORMANCE ANALYZER

OVERVIEW
SQL ANALYSIS
TREND ANALYSIS
VISUALIZATION
COMPARISON
CAPACITY
RAC-EXA
Database’s 6 Various Types of Analysis Method Presented

SQL ANALYSIS / TREND ANALYSIS / DATA VISUALIZATION / POWER COMPARISON / CAPACITY PLANNING / RAC-EXA

Various Performance Analysis of SQL

Detailed Performance and Problem Analysis

Analysis with Visualization Technique Applied on Data

Performance Comparison Analysis of Various Perspectives

Database Capacity Trends Analysis

Optimized Performance Analysis on EXADATA
SQL ANALYSIS

**TOP-N ANALYSIS**

Provides analysis data by Schema / Program / Module / Machine / OS user / SQL Top-N through trends analysis feature.

By getting connected with the Long-Term Trend feature, you can do a long term trends analysis.

**FEATURE**  Top SQL Performance Analysis / Change Target Output Criteria / Sort by Items

Provides SQL Text, Stat information / Provides Execution Plan for each SQL

### DETAIL VIEW

<table>
<thead>
<tr>
<th>Schema</th>
<th>Program</th>
<th>Module</th>
<th>SQL Text</th>
<th>SQL ID</th>
<th>SQL Plan Hash</th>
<th>Executions</th>
<th>Percent Time (%)</th>
<th>CPU Time (%)</th>
<th>Elapsed Time (Sec)</th>
<th>CPU Time (Sec)</th>
<th>Wait Time (Sec)</th>
<th>Logical Reads</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEST</td>
<td>subplus00042A23</td>
<td>SQLPlus</td>
<td>update &quot;<em>&quot; English &quot;</em>&quot; test, lock set begin=1</td>
<td>1000threads20k</td>
<td>3530019330</td>
<td>21,885</td>
<td>23.5%</td>
<td>0.7%</td>
<td>525,208</td>
<td>61</td>
<td>525,146</td>
<td></td>
</tr>
<tr>
<td>TEST</td>
<td>subplus00042A23</td>
<td>SQLPlus</td>
<td>update &quot;<em>&quot; English &quot;</em>&quot; test, lock set end=4</td>
<td>1000threads20k</td>
<td>3530019330</td>
<td>20,449</td>
<td>21.2%</td>
<td>0.7%</td>
<td>679,743</td>
<td>66</td>
<td>624,730</td>
<td></td>
</tr>
<tr>
<td>TEST</td>
<td>subplus00042A23</td>
<td>SQLPlus</td>
<td>update &quot;<em>&quot; English &quot;</em>&quot; test, lock set begin=5</td>
<td>1000threads20k</td>
<td>3530019330</td>
<td>19,647</td>
<td>20.0%</td>
<td>0.6%</td>
<td>775,677</td>
<td>67</td>
<td>783,619</td>
<td></td>
</tr>
<tr>
<td>TEST</td>
<td>subplus00042A23</td>
<td>SQLPlus</td>
<td>update &quot;<em>&quot; English &quot;</em>&quot; test, lock set end=4</td>
<td>1000threads20k</td>
<td>3530019330</td>
<td>19,642</td>
<td>19.3%</td>
<td>0.6%</td>
<td>775,745</td>
<td>66</td>
<td>778,677</td>
<td></td>
</tr>
<tr>
<td>TEST</td>
<td>subplus00042A23</td>
<td>SQLPlus</td>
<td>update procedure test, proc compile</td>
<td>1000threads20k</td>
<td>3530019330</td>
<td>0</td>
<td>15,679</td>
<td>13.2%</td>
<td>0.0%</td>
<td>518,367</td>
<td>716</td>
<td>518,152</td>
</tr>
<tr>
<td>TEST</td>
<td>subplus00042A23</td>
<td>SQLPlus</td>
<td>update &quot;<em>&quot; English &quot;</em>&quot; test, lock set begin=5</td>
<td>1000threads20k</td>
<td>3530019330</td>
<td>0</td>
<td>6,676</td>
<td>2.1%</td>
<td>2.7%</td>
<td>82,626</td>
<td>338</td>
<td>82,588</td>
</tr>
<tr>
<td>TEST</td>
<td>subplus00042A23</td>
<td>SQLPlus</td>
<td>update &quot;<em>&quot; English &quot;</em>&quot; test, lock set end=4</td>
<td>1000threads20k</td>
<td>3530019330</td>
<td>0</td>
<td>6,676</td>
<td>2.1%</td>
<td>2.7%</td>
<td>82,626</td>
<td>338</td>
<td>82,588</td>
</tr>
<tr>
<td>TEST</td>
<td>subplus00042A23</td>
<td>SQLPlus</td>
<td>update &quot;<em>&quot; English &quot;</em>&quot; test, lock set begin=5</td>
<td>1000threads20k</td>
<td>3530019330</td>
<td>0</td>
<td>6,676</td>
<td>2.1%</td>
<td>2.7%</td>
<td>82,626</td>
<td>338</td>
<td>82,588</td>
</tr>
<tr>
<td>TEST</td>
<td>subplus00042A23</td>
<td>SQLPlus</td>
<td>update &quot;<em>&quot; English &quot;</em>&quot; test, lock set end=4</td>
<td>1000threads20k</td>
<td>3530019330</td>
<td>0</td>
<td>6,676</td>
<td>2.1%</td>
<td>2.7%</td>
<td>82,626</td>
<td>338</td>
<td>82,588</td>
</tr>
<tr>
<td>TEST</td>
<td>subplus00042A23</td>
<td>SQLPlus</td>
<td>update &quot;<em>&quot; English &quot;</em>&quot; test, lock set begin=5</td>
<td>1000threads20k</td>
<td>3530019330</td>
<td>0</td>
<td>6,676</td>
<td>2.1%</td>
<td>2.7%</td>
<td>82,626</td>
<td>338</td>
<td>82,588</td>
</tr>
<tr>
<td>TEST</td>
<td>subplus00042A23</td>
<td>SQLPlus</td>
<td>update &quot;<em>&quot; English &quot;</em>&quot; test, lock set end=4</td>
<td>1000threads20k</td>
<td>3530019330</td>
<td>0</td>
<td>6,676</td>
<td>2.1%</td>
<td>2.7%</td>
<td>82,626</td>
<td>338</td>
<td>82,588</td>
</tr>
</tbody>
</table>

---

© Copyrights 2001~2017 EXEM CO., LTD. All Rights Reserved.

27
Intuitive Display of SQL Elapsed Time and Distribution for Analysis

**SUMMARY TREND**
The SQL elapsed time is provided according to wait class. The analysis period is grouped into categories so you can do a top-down method analysis.

**ELAPSED TIME SCATTER**
The SQL elapsed time is displayed in a scatter chart which allows you to intuitively analyze the data. If you drag a select time, the corresponding time period’s SQL information is displayed.
Long Term Trend Analysis Using **1 Day Average Value** for SQL or Top-N

**SQL ANALYSIS**

**LONG TERM ANALYSIS**

**FEATURE**

- Long Term Trends Analysis / Grouping by Various Conditions
- Individual SQL’s Performance History Analysis

**TOP ANALYSIS**

Provides a long term trend analysis feature for Top-N. The Top-N criteria are Schema/ Program/ Module/ SQL.

**1SQL ANALYSIS**

Provides a long term trends analysis feature for a specific SQL. You can check the daily execution count and elapsed time, I/O, SQL text, and bind information.
SQL Execution Pattern Analysis **By OBJECT**

**SQL ANALYSIS**

**OBJECT ANALYSIS – ACCESS STATISTICS**

You can analyze the execution information of SQL accessing the select object. You can check Access Operation, Index Scan Operation and Count information, and when you select a specific SQL, it displays the execution plan information.

**FEATURE**  Specific Object Analysis Details / Provides Access Count / Provides Performance of SQL in Access

**DETAIL VIEW**

![Execution Plan](image)
**DML Pattern Analysis** of Table

**SQL ANALYSIS**

**OBJECT ANALYSIS – CRUD ANALYSIS**

You can analyze the job frequency of table by date/time using the CRUD (Create/Read/Update/Delete) feature. Using this feature, you can easily identify the table's time at which the DML is frequent.

**FEATURE**

- CRUD Analysis of a Specific Segment
- Top-Down Analysis
- SQL Type and Performance Analysis

**DETAIL VIEW**

| EMPLOYEES | 08:00:00-08:55:59 | # of SQL | LINUX11204: 6 |

<table>
<thead>
<tr>
<th>Instance</th>
<th>User Name</th>
<th>Program</th>
<th>Module</th>
<th>Action</th>
<th>SQL Text</th>
<th>SQL ID</th>
<th>SQL Plan Hash</th>
<th>Executions</th>
<th>Elapsed Time (Sec)</th>
<th>CPU Time (Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINUX11204</td>
<td>HR</td>
<td>create@DEVQG2...</td>
<td>SQL+Plus</td>
<td>select /*+ FULL4 PARALLEL(x,5) */ count...</td>
<td>8q2646bm3n3b5</td>
<td>203079695</td>
<td>18</td>
<td>68.70</td>
<td>53.36</td>
<td></td>
</tr>
<tr>
<td>LINUX11204</td>
<td>HR</td>
<td>create@DEVQG2...</td>
<td>SQL+Plus</td>
<td>select /*+ FULL4 PARALLEL(x,5) */ count...</td>
<td>8q2646bm3n3b5</td>
<td>203079695</td>
<td>18</td>
<td>20.33</td>
<td>7.70</td>
<td></td>
</tr>
<tr>
<td>LINUX11204</td>
<td>HR</td>
<td>create@DEVQG2...</td>
<td>SQL+Plus</td>
<td>select /*+ FULL4 PARALLEL(x,5) */ count...</td>
<td>8q2646bm3n3b5</td>
<td>203079695</td>
<td>18</td>
<td>13.66</td>
<td>1.70</td>
<td></td>
</tr>
<tr>
<td>LINUX11204</td>
<td>HR</td>
<td>create@DEVQG2...</td>
<td>SQL+Plus</td>
<td>select /*+ FULL4 PARALLEL(x,5) */ count...</td>
<td>8q2646bm3n3b5</td>
<td>203079695</td>
<td>14</td>
<td>9.90</td>
<td>4.60</td>
<td></td>
</tr>
<tr>
<td>LINUX11204</td>
<td>HR</td>
<td>create@DEVQG2...</td>
<td>SQL+Plus</td>
<td>select /*+ FULL4 PARALLEL(x,5) */ count...</td>
<td>8q2646bm3n3b5</td>
<td>203079695</td>
<td>18</td>
<td>9.25</td>
<td>1.95</td>
<td></td>
</tr>
<tr>
<td>LINUX11204</td>
<td>HR</td>
<td>create@DEVQG2...</td>
<td>SQL+Plus</td>
<td>select /*+ FULL4 PARALLEL(x,5) */ count...</td>
<td>8q2646bm3n3b5</td>
<td>203079695</td>
<td>18</td>
<td>6.35</td>
<td>1.60</td>
<td></td>
</tr>
</tbody>
</table>
View SQL Plan Change History

**SQL ANALYSIS**

**OBJECT ANALYSIS – PLAN CHANGE HISTORY**

Provides trends of the number of SQLs whose execution plan has changed by date. If you use the Plan Diff feature, you can compare 2 execution plans on one screen.

**FEATURE**  SQL Plan Analysis Details / Performance Improvement Confirmation / Performance History Comparison by Plan

---

### DETAIL VIEW

<table>
<thead>
<tr>
<th>Time</th>
<th>Sql Plan Hash</th>
<th>Executions</th>
<th>Elapsed Time/exec (Sec)</th>
<th>CPU Time/exec (Sec)</th>
<th>Logical Reads/exec (block)</th>
<th>Physical Reads/exec (block)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2016-12-26 23:50:00</td>
<td>908878949</td>
<td>194</td>
<td>1.400</td>
<td>0.041</td>
<td>235</td>
<td>111</td>
</tr>
<tr>
<td>2016-12-27 13:30:00</td>
<td>3300108386</td>
<td>1</td>
<td>75.500</td>
<td>2.300</td>
<td>52,633</td>
<td>16,991</td>
</tr>
</tbody>
</table>

**Prev [908878949]**

<table>
<thead>
<tr>
<th>ID</th>
<th>PLAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SELECT STATEMENT</td>
</tr>
<tr>
<td>1</td>
<td>* SORT (ORDER BY)</td>
</tr>
<tr>
<td>2</td>
<td>*** HASH (GROUP BY)</td>
</tr>
<tr>
<td>3</td>
<td>***** HASH JOIN (OUTER)</td>
</tr>
</tbody>
</table>

**Current [3300108386]**

<table>
<thead>
<tr>
<th>ID</th>
<th>PLAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SELECT STATEMENT</td>
</tr>
<tr>
<td>1</td>
<td>* SORT (ORDER BY)</td>
</tr>
<tr>
<td>2</td>
<td>*** HASH (GROUP BY)</td>
</tr>
<tr>
<td>3</td>
<td>***** HASH JOIN (OUTER)</td>
</tr>
</tbody>
</table>
Quick Check on the Root Cause of Problem in the Peak Period with One-Click

TREND ANALYSIS

1-DAY SUMMARY VIEW

FEATURE
Hourly Performance Trends Analysis / Provides Hourly Event information
Top-Down Analysis / Root Cause SQL Analysis

PERFORMANCE
Provides a feature which allows you to easily identify the peak period by providing hourly average trends graph of main performance stats and wait class.

BOTTLENECK
Provides information of SQLs which waited on the Wait Event, Provides a feature by which you can trace the root cause of the problem through the process of Wait Class -> Wait Event -> SQL Level.
Detailed Analysis of Peak Period with **Main Stat and Wait Class**

**PERFORMANCE TREND**

Provides a feature which allows you to easily identify the peak period by providing hourly average trends graph of main stats and wait class. You can retrieve information about the sessions and SQLs which are causing the overload in regards to the peak period by using Time Slice (Detailed Analysis Feature).

**FEATURE**

- Trends Analysis by Dates
- Change Main Stats
- Top-Down Analysis
- Provides Session Info in Unit of Second
- 1 Minute Analysis Link
Session and SQL Search by Various Conditions

**SESSION & SQL LIST**

**FEATURE**
- Utilize Various Search Conditions / Session Detail Link
- SQL Detail Link / Long Term Analysis

**SESSION LIST**
You can analyze the collected sessions and review the session connection info in unit of second, status, SQL execution information, and wait event.

**SQL LIST**
You can analyze a specific SQL’s execution information. It is saved every 10 minutes in a summary, and you can analyze the overall SQL performance and the session connection information.
**Session, SQL Search** of Main Stat and Wait Class Using 1 Day Average Value

**TREND ANALYSIS**

**LONG TERM TREND**

<table>
<thead>
<tr>
<th>FEATURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Long Term Trend Analysis / Provides Average and Maximum Values by Stat</td>
</tr>
<tr>
<td>Provides Wait Event Trend</td>
</tr>
</tbody>
</table>

**STATISTICS TREND**

It provides the main performance stat’s average value and maximum value by date and by time, and the long term trend analysis is possible.

**CLASS TREND**

Provides long term trends of wait time by Wait Event (Class) and wait events generated; and a specific wait event’s wait time by date.
Provides Main **Parameters and Modification History**

**TREND ANALYSIS**

**PARAMETER HISTORY**

FEATURE: Provides Parameter information / Provides Parameter Modification History

**PRE CHECK**

Provides a screen which displays the required parameters classified according to its attributes. You can also check each parameter’s description and setting values which can be helpful for the parameter based analysis.

**MODIFICATION HISTORY**

Provides parameter modification history information. You can check the modified date, and before and after modification setting values.
**Data Visualization** Technique Which Allows You to Identify the Hotspot At a Glance

**DATAPATH VIEW**
Bidirectional analysis of data size of Instance / Schema / Program / SQL regarding multiple databases (or multiple dates for a single database) can be done, and the UI allows you to understand it intuitively.

**HOTSPOT VIEW**
Provides a feature which allows you to quickly and easily identify the Hotspot based on CPU(%) for multiple databases (or multiple dates for a single database).
Top-SQL Rank Comparison Analysis through the Top SQL Comparison

POWER COMPARISON

TOP SQL COMPARISON

Provides a feature for comparing the Top SQLs of the standard against the Top SQLs of the comparing dates. By comparing the SQL ranking by date, you can use this as a basic resource for monitoring SQLs that are causing problem or the influx of new jobs. Long term trend analysis can be done by linking with the Long-Term Trend feature.

FEATURE SQL 1:M, N:M Comparison / Chagme Output Criteria / Provides Diff Values by Date / Provides SQL, Bind, Plan Information
Module & Program Comparison Analysis

**POWER COMPARISON**

**CLIENT COMPARISON**

**FEATURE**  Comparison Module & Program / Provides Diff Values in Reference to the Base Date

**MODULE COMPARISON**

In the event of a specific module’s performance degradation, this feature allows you to compare against the previous execution history.

**PROGRAM COMPARISON**

In the event of a specific program’s performance degradation, this feature allows you to compare against the previous execution history.
Peak Time **Comparison Analysis**

**POWER COMPARISON**

**PEAK TIME COMPARISON**

In the event performance degradation period is generated, this feature allows you to compare the corresponding time point of previous dates.

**FEATURE** Provides Peak Time by Date / Provides SQL Diff Values by Date / SQL Detail Link

---

**DETAIL VIEW**

<table>
<thead>
<tr>
<th>Schema</th>
<th>Program</th>
<th>Module</th>
<th>SQL ID</th>
<th>Impact(%)</th>
<th>Date</th>
<th>Plan Hash Value</th>
<th>Executions</th>
<th>Elapsed Time (Sec)</th>
<th>CPU Time (Sec)</th>
<th>Wait Time (Sec)</th>
<th>Logical Reads (block)</th>
<th>Physical Reads (block)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEST</td>
<td>SQL*Plus</td>
<td>9bpbwib0znkmzu2z6</td>
<td></td>
<td>162.5%</td>
<td>01.02</td>
<td>0</td>
<td>548</td>
<td>20,790</td>
<td>27</td>
<td>20,763</td>
<td>15,907</td>
<td>62</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01.01</td>
<td>0</td>
<td>402</td>
<td>13,464</td>
<td>18</td>
<td>13,385</td>
<td>12,682</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>diff (01.02-01.01)</td>
<td>56</td>
<td>7,186</td>
<td>9</td>
<td>7,178</td>
<td>3,225</td>
<td>-20</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01.02 (per exec)</td>
<td>0</td>
<td>1</td>
<td>2,086,099</td>
<td>2.766</td>
<td>2,083,393</td>
<td>1,500</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01.01 (per exec)</td>
<td>0</td>
<td>1</td>
<td>1,514,999</td>
<td>2.036</td>
<td>1,512,963</td>
<td>1,405</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>diff (01.02-01.01)</td>
<td>0</td>
<td>571,100</td>
<td>0.670</td>
<td>570,420</td>
<td>195</td>
<td>-4</td>
</tr>
<tr>
<td>TEST</td>
<td>SQL*Plus</td>
<td>5wsoi0dwq08qws</td>
<td></td>
<td>46.3%</td>
<td>01.02</td>
<td>39330/19320</td>
<td>726</td>
<td>36,962</td>
<td>3</td>
<td>36,959</td>
<td>299</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01.01</td>
<td>39330/19330</td>
<td>716</td>
<td>28,859</td>
<td>1</td>
<td>28,858</td>
<td>374</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>diff (01.02-01.01)</td>
<td>20</td>
<td>2,163</td>
<td>2</td>
<td>2,101</td>
<td>-74</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01.02 (per exec)</td>
<td>39330/19330</td>
<td>1</td>
<td>2,316,002</td>
<td>0.234</td>
<td>2,315,768</td>
<td>22</td>
<td>0</td>
</tr>
</tbody>
</table>
Intuitive Performance Comparison Through Time Matching

**POWER COMPARISON**

**TRENDS COMPARISON**

This feature allows the user to select a desired instance's specific date for comparison to analyze the performance stats. This feature can be used to compare and analyze among the different nodes in a RAC environment, or by instance dates on which the same task is repeated.

**FEATURE** Multiple Instances Trends Comparison / Multiple Dates Simultaneous Comparison Analysis / Session Info in Unit of Second

**DETAIL VIEW**
Usage Trends Useful for Database Capacity Planning

**TABLESPACE TREND**
Provides trends of tablespace usage and free space. It also provides information about the Top-N segment within the tablespace in which the usage increase ratio is high.

**SEGMENT TREND**
Provides information about the daily incremental values and the size of particular segment (TABLE, INDEX, PARTITION).
Provides EXADATA Analysis Feature

<table>
<thead>
<tr>
<th>RAC-EXA ANALYSIS</th>
<th>FEATURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provides Cell Server Performance Info / Provides Smart Scan Usage</td>
<td>Provides Flash Cache Usage</td>
</tr>
</tbody>
</table>

PERFORMANCE TREND
The cell server and the DB performance stats are configured into one screen which allows you to easily identify and analyze the EXADATA overload period.

FLASH CACHE
Displays EXADATA’s Flash Cache Hit/Miss Count grouped by time and object unit.
**DB Performance Report** by Date/Period for a Report Preparation

---

**ANALYSIS REPORT**

**FEATUE**
- Provides Excel, HTML Report / Select Stats and Top-N
- Provides Long Term Reports

---

**DAILY & LONG TERM REPORT**
Provides a feature for printing a performance report of all performance stats collected and Top-N Summary by date/period. It can also be downloaded in Excel and HTML format.

---

**HOTSPOT REPORT**
You can check the hourly trends of a specific date’s CPU (%) usage, Active User Session, and Wait Event (Class). It can be downloaded in Excel or HTML format.
Thank you