



AB*Performance Suite Operation Concepts

ActiveBase Server Version 3

email: support@active-base.com

Visit us: www.active-base.com

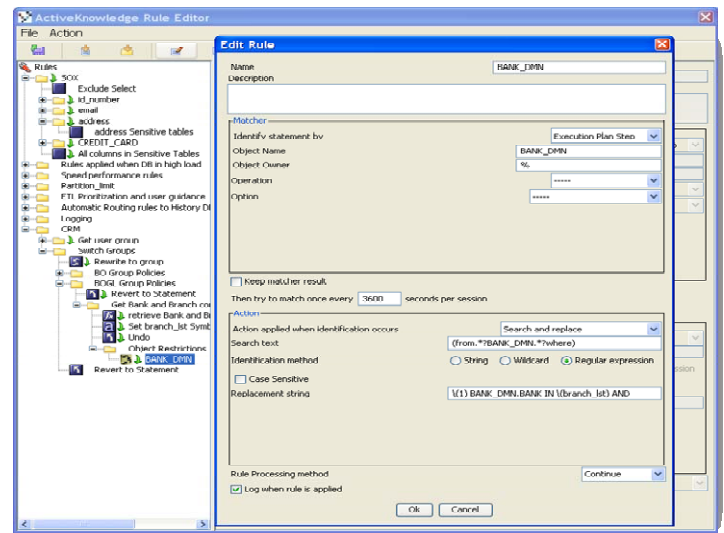
1 Overview

AB*Performance Suite manages and optimizes incoming SQL request traffic and manages database server resources according to business priorities.

It automatically applies ad-hoc and predefined rules for:

1. Managing CPU/IO resources assigned to active database sessions – **optimizing resource allocation to SQL statements currently running in the database.**
2. Improving and managing both SQL statements traveling between clients/applications and the databases - **optimizing incoming SQL statements before reaching the database.**

AB*Performance Suite includes Knowledge Pack rules. Knowledge Packs are XML files that include rule examples and best practice for various business applications (e.g., rules for managing and optimizing Business Objects or Amdocs Billing app).



Management Console: flexible and powerful Rule Tree

Knowledge Packs include solutions to common performance issues experienced by IT (e.g., changing index range scan to partition range based on number of partitions, blocking never-ending requests, removing unneeded functions. conversions on 'Where' clause conditions etc.).

Rules are administered using a single centralized management GUI.

Export / Import utility, rule propagation between environments, rule grouping into folders, rule simulation and various identification methods enable administrators to solve any challenge.

1.1 Rules managing SQL statements currently running in the database (active sessions)

- AB*Performance suite rules ensure that database server resources are allocated to transactions according to business priorities.
- It monitors Oracle active sessions and their underlined server processes.
- AB*Performance suite applies ad-hoc and predefined rules that manage the database server resources (CPU and I/O) assigned to Oracle sessions.

- Rules apply actions based on various **identification** criteria, including:
 1. Identifying SQL statement text patterns (syntax matching and regular expression)
 2. Active session information (e.g., module or program name)
 3. Time of day
 4. Overall server CPU and I/O load
 5. Instance load and session load
 6. Session resource load
 7. Grouping sessions (e.g., analysts resource load > 30% of database server resources)

- AB*Performance suite rule **actions** on Oracle active sessions include:
 1. Reduce Oracle session resources by X% CPU and I/O
 2. Ensure that all active sessions within a group consume less than Y% CPU and I/O

- AB*Performance suite priority module is installed on the database server for mapping the different Oracle active sessions to the Operating System processes, and when instructed by rule actions, allocates resources by issuing kernel commands to the processes accordingly.

- These kernel calls manage the activity span of the processes, thus directly affecting the CPU and I/O resources used by the process.

- By reducing or expanding the process time allocation (process activity span change), It effectively reduces the database server resource utilization, freeing substantial CPU and I/O capacity to perform more important business processes. For example, in a call center environment, if the administrator reduces resources of a reporting session by 50%, the session will run two times slower, freeing these processing resources 50% of the time, to the call center on-line requests.

- Rule processing starts with applying Session rules and ends by applying Resource Grouping rules.
 1. Session rules are applied on all active sessions, managing individual session.
 2. Resource grouping rules managing group of sessions. For example, session rules group all business objects sessions in an ERP environment as analysts. Resource grouping rules limit their CPU and I/O resources to 20%.

After another 15 seconds a new rule processing cycle commences.

1.1.1 Session Rules

Session rules apply actions on specific active sessions. Every 15 seconds (cycle time), The rule engine retrieves Oracle active session information and resource

consumption, instance and database server resource condition from the operating system.

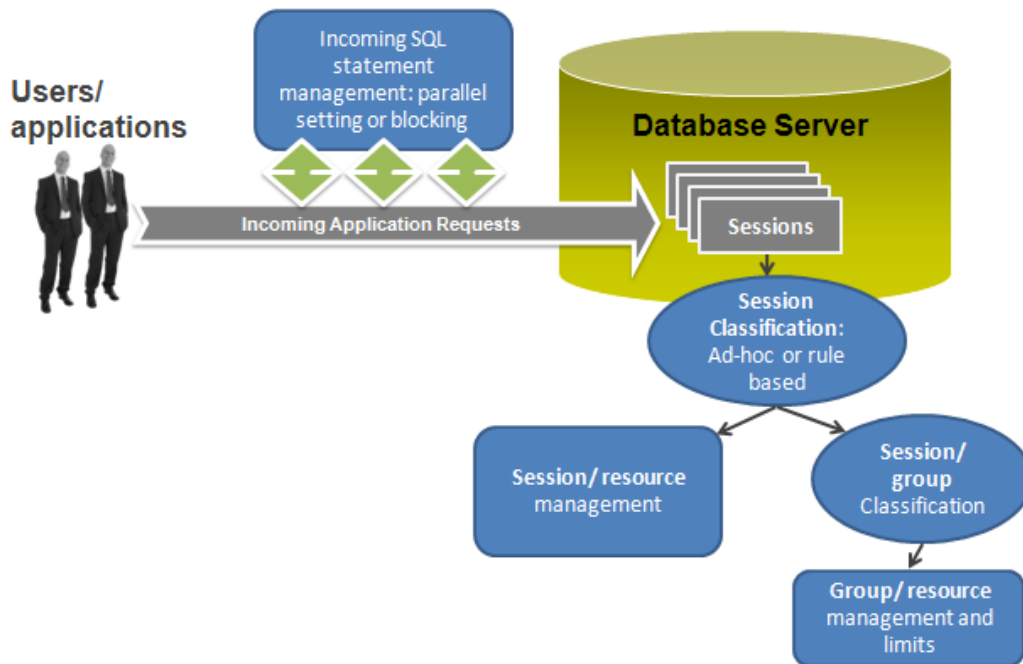
It validates the rule identification matchers on the active sessions and applies the rule actions appropriately.

Session rules also classify active sessions into groups, used as a classification to the Resource Grouping rules.

1.1.2 Resource Group Rules

After applying Session Rules, the rule engine continues to process the Resource Group rules. The Resource Group rules identify sessions based on a grouping criteria set in the Session Rules – to apply resource management at the group level. For example, In an ERP application, when database server CPU load exceeds 85%, both batch and reporting processes do not consume more than 25% of all database server resources.

Server Resource management >> process flow



1.2 Rules for managing incoming SQL statements before reaching the database

- AB*Performance suite includes a powerful software module that acts as a SQL*Net proxy. The AB*Performance proxy works on the SQL*Net protocol layer, identifying SQL statement protocol packets flowing from the clients/applications to the databases through AB*Performance, applying predefined rules in real-time.

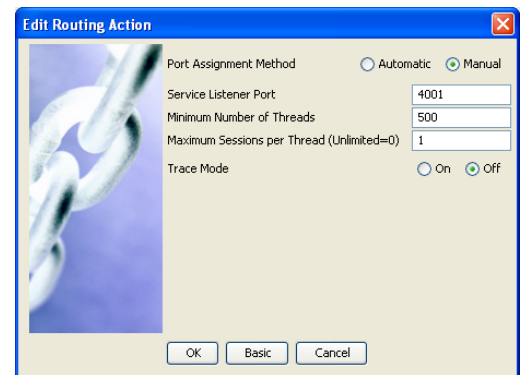
- It is implemented by simply redirecting selectively clients/applications to connect to an ActiveBase Listener ports (by changing TNSNAME.ORA or JDBC property file). Using the ActiveBase Listener port enables administrators to apply SQL statement optimizations transparently, without touching applications or databases.
- AB*Performance suite connection switching functions, routes only selected applications/modules connection requests through ActiveBase Listener, where the rest of the applications completely bypass ActiveBase proxy.
- It applies predefined rules on incoming SQL statements.
- Rules apply actions based on various **identification** criteria, including:
 1. Identifying SQL patterns (syntax matching and regular expression),
 2. 'From' clause objects included in the SQL
 3. 'Where' clause conditions – identifying number of days requested on a column in transaction table (for determining if Nested loop hint will be applied or an index range scan hint)
 4. Partial explain plan pattern matching or single execution plan step,
 5. Number of partitions to be scanned,
 6. Time of day
 7. Oracle cost.
- AB*Performance suite rule **actions** on incoming SQL requests include:
 1. SQL Rewrite
 2. Add hint
 3. Block
 4. Search-and-replace
 5. Define symbol, offload to replication or to history database
- Rule examples:
 1. Rule for adding Oracle hint for applying full scan instead of index range scan, when transaction 'Where' clause condition exceeds 10 days.
 2. Rule for rewriting unneeded decode and format functions on 'Where' clause conditions that massively degrade Oracle performance.

3. Rule for blocking SQL requests that will never complete based on partial explain plan identification. AB*Performance suite issues an ORA-900 error code with a custom error text message returned to the application defined in the rule.

2 ActiveBase SQL traffic routing scalability

ActiveBase includes several features that guarantee the support for OLTP applications with throughputs of as high as 50,000 SQL statements per second through the ActiveBase Listener.

1. Selective routing based on application/module/application server/client host name, OS user or program name list
2. ActiveBase server can be installed on the database server to minimize network hops (Unix/Linux/Windows OS)
3. ActiveBase is built on a multi-threaded server software, with a powerful configuration, managing both the number of threads created upon service initiation, and the number of sessions opened per thread to minimize any possible session delays.
4. ActiveBase supports multiple listener port configurations, to divide incoming application traffic to several different listener ports and to several different ActiveBase servers.
5. Low latency – Packet average transit time in ActiveBase Performance is 100 microseconds.
6. Fast Rule engine – several patented innovations have been included in the rule engine in order to be able to support both high-load OLTPs, as well as large scale BI applications. Rules falls into one of three categories:
 - a. Text based, which takes ~20 Microsecond per parsed statement (20/1,000,000 of a second).
 - b. Parser based, which takes ~1 millisecond per parsed statement
 - c. Database based (explain plan or PL/SQL), which takes ~50-100 milliseconds per parsed statement. While usually not harmful on BI environments these should be used more carefully in heavy OLTP environments.



- Rule tree folders minimize rule overhead on incoming traffic by first applying quick text based identification (e.g., identifying SQL statements with large tables and problematic 'Where' clause condition), followed by parser or database based rules, matched to a small percentage of overall SQL statements.
7. A rule tree has been included, that enables ActiveBase to bypass all fast and efficient SQL statements with no delay, while only the long and inefficient SQL statement are caught and manipulated by the rules – accelerating them by x10-1000 times.
 8. ActiveBase software resource consumption is around 1% of the server CPU, with no I/O overhead.

3 Hardware Requirements

- Processor speed: 1GHz or higher (Either as single processor or combined multi processor speed)
- 2 GB of available hard-disk space (600M are required for statement logging purposes)
- Minimum of 1 GB RAM

4 Supported Operating Systems

ActiveBase server can be installed on the following operating systems (32 or 64 bit):

- Sun Solaris 8 or higher
- HP-UX 11 or higher
- AIX 5.2 and higher
- Linux Redhat 7.2 or higher

The AB*Performance suite GUI Management Console (for administrating the AB*Priority) can be installed on Microsoft Windows NT/2000 (server or professional)

ActiveBase software is built on Java technology, and such, is installed with a dedicated Java Virtual Machine (JVM), and does not depend on pre-existing Java installations on the server. ActiveBase is released with JVM 1.5.0_04.