

ActiveBase Priority

Performance improvement and Service Level stability

Table of Contents

Do I need to change my Applications or Database?	1
How do I install AB*Priority?	1
What is ActiveBase program language?	2
Does AB*Priority manages resources using 'nice' command?.....	2
Is there a possibility to create rule which will catch all queries which run on fact tables?	2
What are the resources AB*Priority consumes as it is installed on the database server?	2
What Databases does AB*Priority support?.....	2
How AB*Priority manages locks, so they are not prolonged more than required?	3
How does AB*Priority identifies and manages heavy I/O consuming sessions?.....	3
How does AB*Priority identifies and manages Parallel Query Server sessions?	3
How do you upgrade AB*Priority?	3
How is AB*Priority different from native Operating system and Oracle resource manager features and tools?	3
How can I reduce resources of sessions that run longer than 30 minutes or consume more than 1000000 I/O?	3
What does the Resource Grouping Rules action: Limit CPU Usage percentage calculates?	5
How do you compare with other resource priority solutions?.....	5

How large is the download and what does it include?

The download for the AB*Priority server software ~35MB, dependant on your OS version, and includes the installation as well as documentation and a management console installation.

How long will it take to install and configure AB*Priority on a single database?

Installation only takes about an hour. You will first install the server software (opening a tar), followed by a PC – administrator management consol.

Predefined rules will be available immediately, and you will be able to define custom rules


Do I need to change my Applications or Database?

No! AB*Priority is completely transparent to both applications and databases. NO DATABASE CONFIGURATION CHANGES ARE REQUIRED!

How do I install AB*Priority?

You create user active, and open AB*Priority tar file. It also comes with its OWN JVM installation, inside the installation tar. In order to throttle Oracle





processes, it requires that an Oracle OS user 'starts' the service. If you need to throttle OS processes, you need to 'start' AB*Priority using root. When starting AB*Priority from root user, AB*Priority is able to control OS processes, outside the domain of Oracle (for example, controlling an export OS command that consumes many resources and interfering with the database work).

For starting AB*Priority using Oracle OS user, you might need to change the group privileges of all files in directories activepriority/cfg and activepriority/log to be accessed by the Oracle OS user.

After installation, you define the Oracle databases that run on the server. When introducing a database in AB*Priority, you need to enter oracle username and password. This Oracle user needs to have read privileges into v\$session, processes, px and sesstat views.

What is ActiveBase program language?

Java 1.5.0_04, which is included within ActiveBase installation, so no special preparation is required prior installation.

Does AB*Priority manages resources using 'nice' command?

No! Nice OS commands are inaccurate with limited efficiency. We have created a Process Manager component that throttles the relevant processes with high precision.

Is there a possibility to create rule which will catch all queries which run on fact tables?

Yes AB*Priority can. We include a session identification capability based on the actual SQL statement executed (identification includes regular expression pattern matching on the SQL text).

What are the resources AB*Priority consumes as it is installed on the database server?

As it only processes ACTIVE USER SESSIONS (usually less than 10% of total number of sessions) and not analyzes all sessions, collecting statistics only every 15 seconds (can be changed by customer), overhead is negligible - about 1% CPU load.

What Databases does AB*Priority support?

AB*Priority is currently available for Oracle8 – 11g running on UNIX (HP, Sun or IBM AIX) or Linux.

How AB*Priority manages locks, so they are not prolonged more than required?

AB*Priority applies resource management (i.e. suspend/resume processes) on sessions only when the process is in user mode. As latches are applied in Kernel mode and released in the same call, process will never be suspended while holding a latch.

How does AB*Priority identifies and manages heavy I/O consuming sessions?

AB*Priority includes an identification criteria based on session statistics. You can define session consumption thresholds and rate limits to any one of the 300 session statistics provided by Oracle.

How does AB*Priority identifies and manages Parallel Query Server sessions?

AB*Priority identifies both the query coordinator and the slave processes and manages them as a whole.

When reducing a PQ session, AB*Priority will reduce all slaves accordingly.

How do you upgrade AB*Priority?

Simply stop the server process (using `./stop` command within the `activepriority` directory), download the tar file into active directory and untar it. Upgrading does not delete your existing installation configurations. You also will need to uninstall ((control panel -> install/uninstall programs->ActiveBase),) and reinstall the Windows client to be able to connect and administer the new version.

How is AB*Priority different from native Operating system and Oracle resource manager features and tools?

See detailed comparison analysis document. Basic differences are ORM does not manage OS processes, OS priority tools do not see the database – we see both, controlling multiple instances and VM's on a consolidated machine with a powerful dashboard.

Important note: having AB*Priority, Oracle categorizes the wait events generated by AB*Priority as type Oracle Resource Manager wait events.

How can I reduce resources of sessions that run longer than 30 minutes or consume more than 1000000 I/O?

Yes, use the “session statistics” identification to identify long running sessions based on:

In case you are CPU bounded: CPU used by this session

CPU used by this session	Amount of CPU time (in 10s of milliseconds) used by a session from the time a user call starts until it ends. If a user call completes within 10 milliseconds, the start and end user-call time are the same for purposes of this statistics, and 0 milliseconds are added.
--------------------------	---

I/O bounded: Physical read IO requests

physical read IO requests	Number of read requests for application activity (mainly buffer cache and direct load operation) which read one or more database blocks per request. This is a subset of "physical read total IO requests" statistic.
---------------------------	---

Or: Session Connect Time:

session connect time	The connect time for the session in 10s of milliseconds. This value is useful only in V\$SESSTAT. It is the wall clock time since the logon to this session occurred.
----------------------	---

For full description of all session statistics, go to:

http://download.oracle.com/docs/cd/B19306_01/server.102/b14237/stats002.htm#i375475

What does the Resource Grouping Rules action: Limit CPU Usage percentage calculates?

The CPU measure calculates the CPU used by this session as percentage of the total number of CPU Cores in the server.

In Oracle10 and higher, the session CPU is retrieved from Oracle time model statistics see more information from Oracle site:

http://download.oracle.com/docs/cd/B19306_01/server.102/b14237/dynviews_2087.htm#REFRN30340

For example, in a server with 100 CPU cores, limiting crystal-report group of sessions to less than 4% means that the rule will initiate a proportional resource reduction on crystal sessions only when the total CPU consumed by all crystal report session exceeds 4 CPU cores.

The time statistics model is updated at least every five seconds.

In Oracle9, there is no time model available, thus CPU is measured from the operating system level. This measurement is less accurate (+-7%) than the time model in later Oracle versions.

How do you compare with other resource priority solutions?

For a comparison with Oracle Resource Manager please see ComparisonOraclevsABPriority data sheet.

Comparing with OS command:

1. A major advantage of a resource management solution is the ability to divide resource between usage during peak-loads (e.g., analysts on the ERP production will not consume more than 10% resources during peak load) – this cannot be done in OS commands as they are not accurate and cannot be used from grouping different sessions together.
2. No Oracle session visibility: in order to reduce a resource allocated to a long running report or never-ending job you need to identify the process_id manually, and then issue the priority setting command to every related process. In addition, Oracle enables parallel execution – meaning there can be multiple processes related to a job or a report, making the manual management using OS commands impossible.
3. AB*Priority enables preventive actions – rules that limit resources ONLY when jobs or user requests are running more than a normal time or consume a normal amount of resources, preventing resource peaks and service downtime. This resource management cannot be implemented in OS commands.

- 4. OS command priority setting does not provide an operational dashboard as well as rule management and auditing for analysis.

www.active-base.com

400-00101-301 | 02/09 | © 2009 ActiveBase, Ltd. All rights reserved. All other third-party trademarks are the property of their respective owners.

