HP Asset Manager

Software Version: 9.50
Windows® and Linux® operating systems

Asset Manager Web Monitoring for Windows x64, Oracle, MS SQL Server and Tomcat

Document Release Date: March 2015
Software Release Date: March 2015

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

Trademark Notices

Adobe® is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to: **http://h20230.www2.hp.com/selfsolve/manuals**

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to: **http://h20229.www2.hp.com/passport-registration.html**

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

Visit the HP Software Support Online web site at: **http://www.hp.com/go/hpsoftwaresupport**

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

HP Software Solutions Now accesses the HPSW Solution and Integration Portal Web site. This site enables you to explore HP Product Solutions to meet your business needs, includes a full list of Integrations between HP Products, as well as a listing of ITIL Processes. The URL for this Web site is **http://h20230.www2.hp.com/sc/solutions/index.jsp**

Contents

Introduction

In a production system, tiers are often separated into different teams.

In a typical Asset Manager (AM) system, there are 4 different teams:

- AMAPP: The Asset Manager application team is responsible for the architecture implementation, customization, and AM product maintenance tasks.
- WEB: The web team is responsible for the monitoring, maintenance, and tuning of Web Servers. This includes the control of the Web Servers in production.
- NETWRK: The network team is responsible for the maintenance of the network environment.
- DBA: The DBMS (Oracle 11 and greater) DBA team is responsible for database maintenance (backup, security, recovery) and performance tuning.

This guide provides a task list of monitoring tasks.

Some actions are described briefly in this guide. Refer to the existing guides or documentation of the third-party products or Asset Manager documentation to handle them.

AM web server

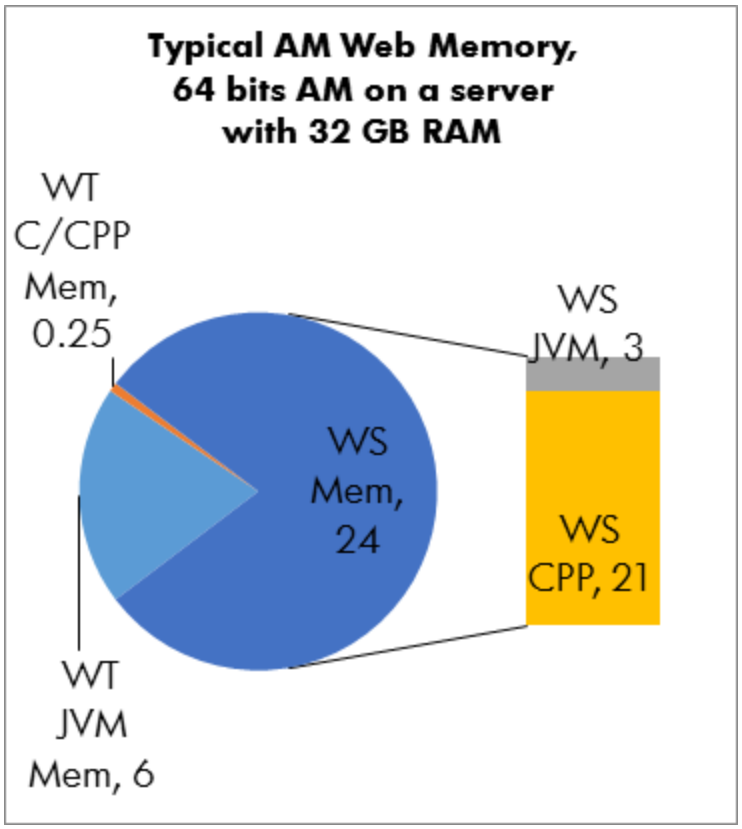
We recommend that you use the 64-bit web server in production, therefore the monitoring and settings in this guide are for 64-bit environment. Baselines provided are valid for 64-bit Tomcat AM web instances.

Baseline

Tomcat instances, JVM, aamapi*.dll run in 64-bit application mode.

Separated Tomcat instances for AM web tier and AM web services

The following figure shows an example of a baseline sizing for an AM web service (WS) and AM web server tier (WT), running on 2 separated 64-bit Tomcat instances. Numbers are given in gigabytes.



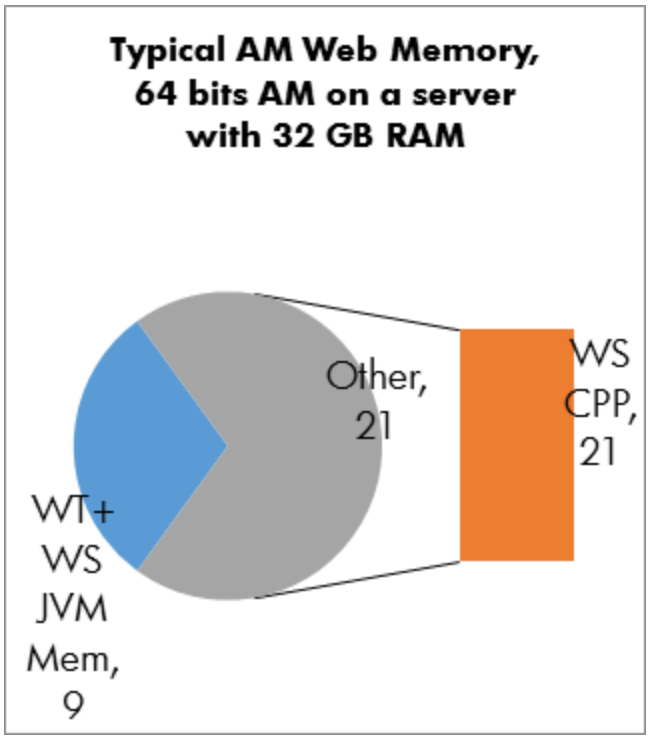
- AM web tier
  - Dedicated Tomcat instance
  - Xmx=6G

  - Almost no non-Java memory required (some for the web server itself)

- AM web service
  - Dedicated Tomcat instance
  - Xmx=3G
  - >21 GB or more available for the C/CPP caches and database connection pool
  - Some RAM for the OS and processes

Shared Tomcat instance for AM web tier and AM web services

The following figure shows an example of a baseline sizing for an AM web service and AM web server tier, running on the same 64-bit Tomcat instance. Numbers are given in gigabytes.



- Web tier and web service running on the same Tomcat instance (diagnosis is a little easier).
  - Xmx=9G
  - >21 GB or more available for the C/CPP caches and database connection pool
  - Some RAM for the OS and processes

Web tier example: Bench machine has 2 Intel Quad-core Xeon E5462 @ 2.80 GHz CPU; 24 GB RAM

```
JVM 1.6.0_24: JAVA_OPTS=-Xms4096m -Xmx4096M -XX:+UseParNewGC
-XX:+UseConcMarkSweepGC -XX:+UseTLAB -XX:SurvivorRatio=6 -XX:NewSize=2048m -XX:MaxN
ewSize=2048m -server -Dsun.lang.ClassLoader.allowArraySyntax=true
```

Tomcat 5.5 settings:

```
<Connector port="8080" redirectPort="8443" maxThreads="550" minSpareThreads="100" m
axSpareThreads="300” acceptCount="575" connectionTimeout="180000” -enableLookups="f
alse“ debug="0"  disableUploadTimeout="true" maxKeepAliveRequests="1000" keepAliveT
imeout="180000" />
```

Recommended setup and tuning

We recommend that you:

- Split the web service and web tier into 2 web application instances.
- Use dedicated web instances (no other applications on the same web application server instance).
- Use the tagged version of Asset Manager web (not Head).
- Make sure that the package.property file of the web tier is configured properly to use the tagged version on the web service.
- Make sure that the package.property is configured to pre-load the tagged version at startup.
- Combining several tagged versions on the same web service is possible, but it is not recommended at all (memory size due to caches).

Recommended settings of the connection pool

Connection pool settings control the number of connections to the Asset Manager database that are kept opened in the Asset Manager web service memory pool. Setting these parameters in a medium or large production environment is mandatory.

Parameters are detailed in depth in the HP Asset Manager Administration documentation, chapter **Controlling access to the database**. Check also the release notes to get the latest changes to these parameters. They are located in the `[Option]` group with `/Advanced/<parameter>=<Value>` in the `aamapi<version>.ini` file (`aamapi95.ini` for Asset Manager version 9.50).

Huge impact on performance

- **CnxPoolIdleSize** is the number of the connections that are left opened. Any new connections allocated above **CnxPoolIdleSize** is temporary. They will be allocated to perform the call, and then will be de-allocated.

  For example, if a new request comes to the web service when **CnxPoolIdleSize** is 50, **CnxPoolMaxSize** is bigger than 51 and all connections are currently busy on the web service, then:

a. A new connection will be allocated to perform the web service call.

b. After the elementary web service call (for example, get a list, get a detail) is over, a connection will be de-allocated to come down to **CnxPoolIdleSize**. The connection that is de-allocated can be either an older connection (if there are some that are not busy), or the newly created connection (if all other connections are still busy).

c. The impact on performance is **HUGE** because allocating and de-allocating a connection are processes demanding CPU resource.

- **CnxPoolMaxSize** is the parameter used to limit the maximum number of connections allocated in the connection pool. It should always be equal to or bigger than **CnxPoolIdleSize** and **CnxPoolMinSize**. Setting a large value to this parameter can have effects on memory and should not be taken as a baseline of the number of connections in the connection pool. When you consider a larger value to this parameter, you should also consider setting **CnxPoolMinSize** to a larger value.

Medium impact on performance

- **CnxPoolLifeTimeMinutes** is the time period (in minutes) that sets the maximum lifetime of a connection in the pool. By default, it is 90 minutes. This parameter enables memory refresh for connections. In some specific cases, a connection can hold a large amount of memory. Refreshing connections resets the memory allocated by those large connections. Notice that this parameter is global to an Asset Manager web service, it cannot be controlled individually depending on the profile used by the connection.

Low impact on performance

- **CnxPoolMinSize** is the minimal number of connections that Asset Manager web server should open at startup (slower startup, but better scalability). This parameter is not very important.

- **CnxPoolMemory** can be discarded in 64-bit Asset Manager web service systems. No memory protection mechanism is set on 64-bit web services. Total amount of memory allocated can be mostly controlled by the **CnxPoolMaxSize** and **CnxPoolLifeTimeMinutes** parameters.

DBMS

Oracle

There is a Tuning Guide of Asset Manager that provides the baseline of an Oracle instance settings.

Oracle tuning mostly happens in each implementation depending on the list configuration, access restrictions, custom wizards, scripts, queries, and profiles. Asset Manager is not a self-tuning application, therefore, DBA monitoring is required.

Most of the time, the tuning is done by adding some multi-column indices in the database, or/and redefining the queries and access restriction or list configuration customized at implementation.

For advanced tuning such as required sorts on a large linked table (different from the main table), Asset Manager has the capability to add a denormalized field (copy the value of a field on an external table into the main table) to solve issues, with a performance slowdown on updates and inserts.

OPTIMIZER_MODE

At the beginning, before going in depth in the Oracle tuning, carefully choose the OPTIMIZER_MODE. Most often fractions of the dataset are retrieved in Asset Manager. Therefore, make sure that the OPTIMIZER_MODE is set to FIRST_ROWS_1, FIRST_ROWS_10, FIRST_ROWS_100, or FIRST_ROWS_1000.

The second tuning parameter is the index cost factor (optimizer_index_cost_adj) that often needs to be tuned for Asset Manager on Oracle. It can be set in the Oracle instance or for the Asset Manager database only.

The following is an example of a trigger that performs this on an Asset Manager schema (DBA).

```
CREATE OR REPLACE TRIGGER AMSCHEMA.AM_POSTCONNECTSQL AFTER
LOGON ON AMSCHEMA.SCHEMA declare str1 varchar2 (1000);
Begin
str1 := 'alter session set optimizer_mode=first_rows_10';
execute immediate (str1);
str1 := 'alter session set optimizer_index_cost_adj=25';
execute immediate (str1);
End;
/
```

If aliases are set and new Oracle logon names are created for AM users (to restrict rights on tables and AM schema), as recommended for security reasons, this trigger should be applied so that each AM user session can be set with these parameters.

Oracle users who can be kept in CHOOSE or ALL_ROWS optimizer mode are users doing reporting on all datasets.

Microsoft SQL Server

Basic settings for performance

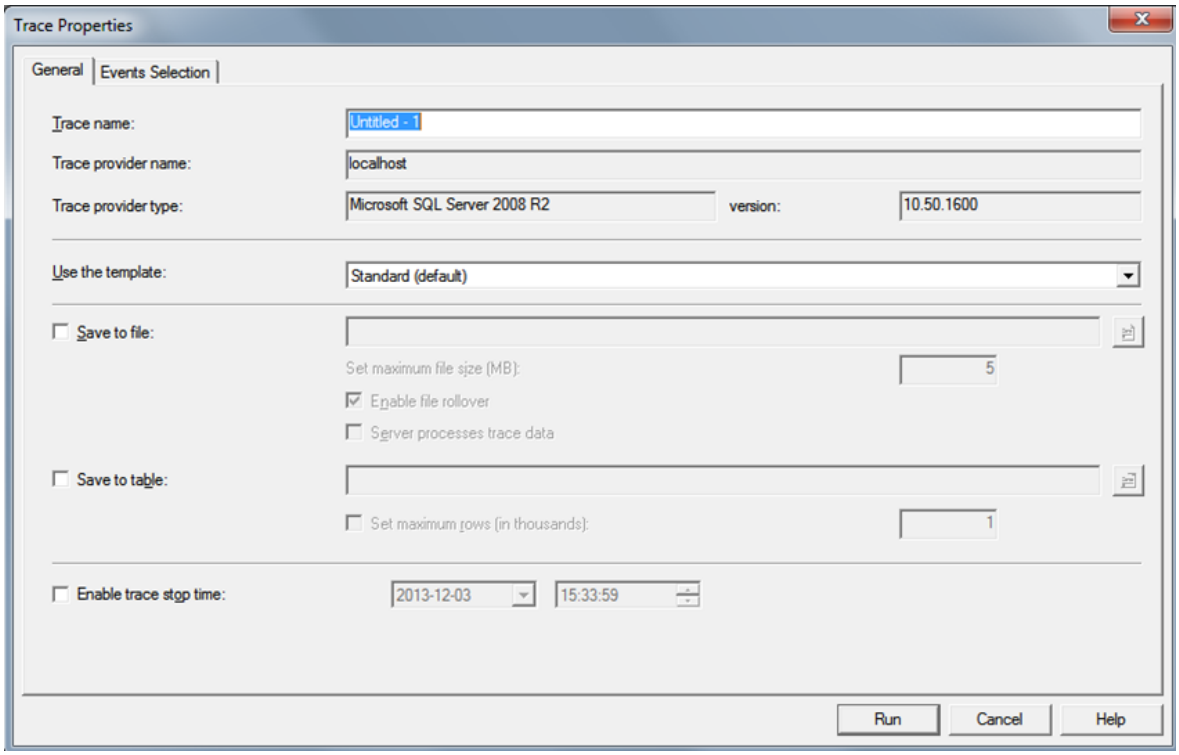
Microsoft SQL Server also offers some tuning and monitoring tools depending on the version.

To have good performance with MS SQL Server, some parameters and settings should always be turned on:

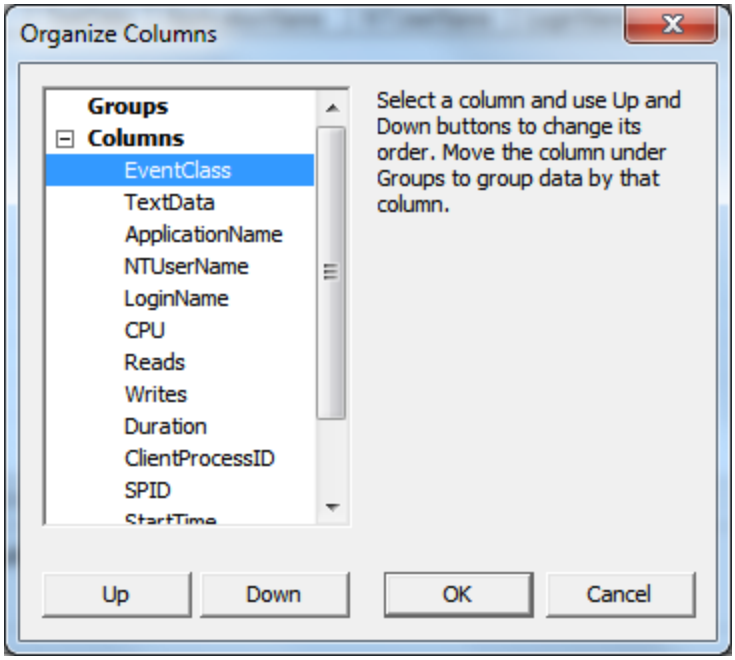
- Implement alternate counter stored procedure (up_getcounterval) in Asset Manager. For more information, see the Tuning guide.
- Use the alternate isolation level on:
  - ```
    ALTER DATABASE AMDemo52en SET READ_COMMITTED_SNAPSHOT ON
    ALTER DATABASE AMDemo52en SET ALLOW_SNAPSHOT_ISOLATION ON
    GO
    ```
- In the Asset Manager Widows client, set the database options in your database as follows:
  - **Isolation command before starting a write transaction**: set transaction isolation level snapshot.
  - **Isolation command for returning to data browsing read mode**: set transaction isolation level read uncommitted.

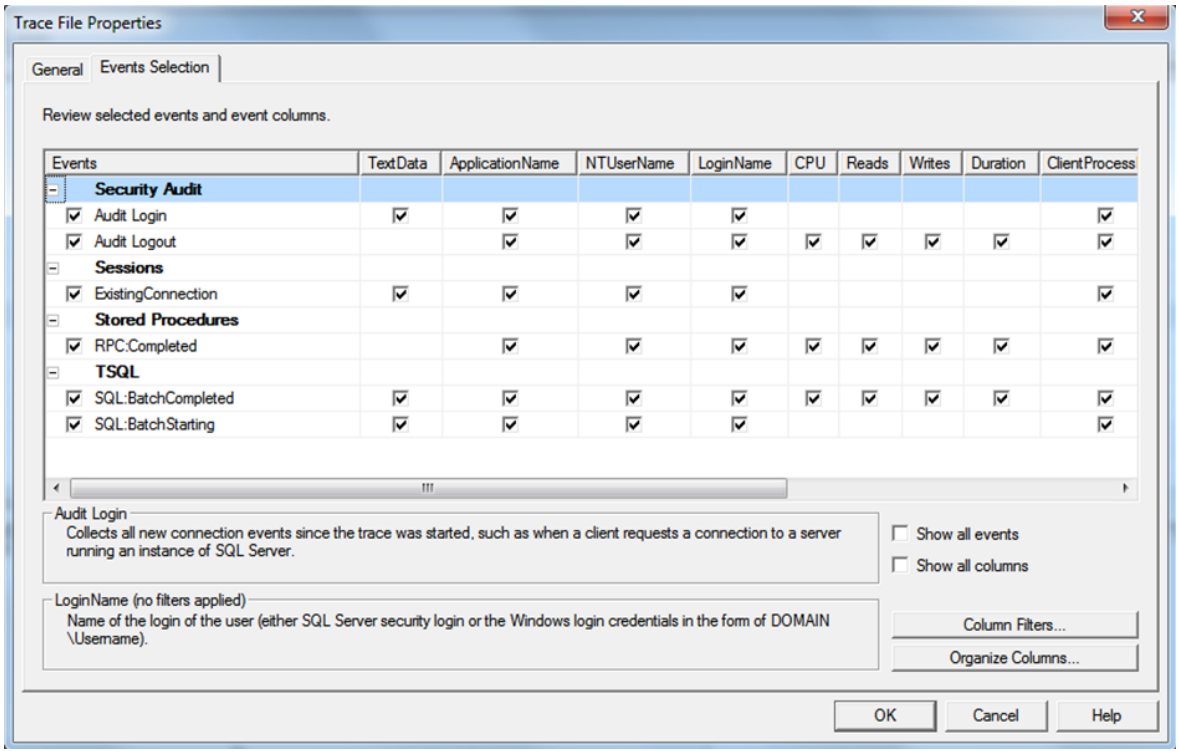
Monitoring tools

As a base line, it is recommended to turn on the MS SQL Server Profiler and generate a trace to capture poor performance queries.
    ```

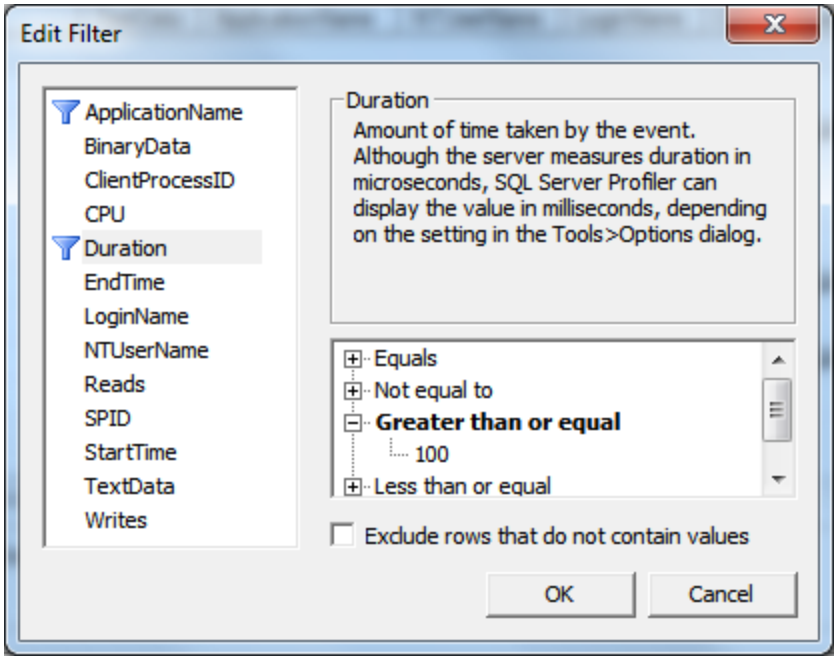


Parameters of the trace events can be modified.





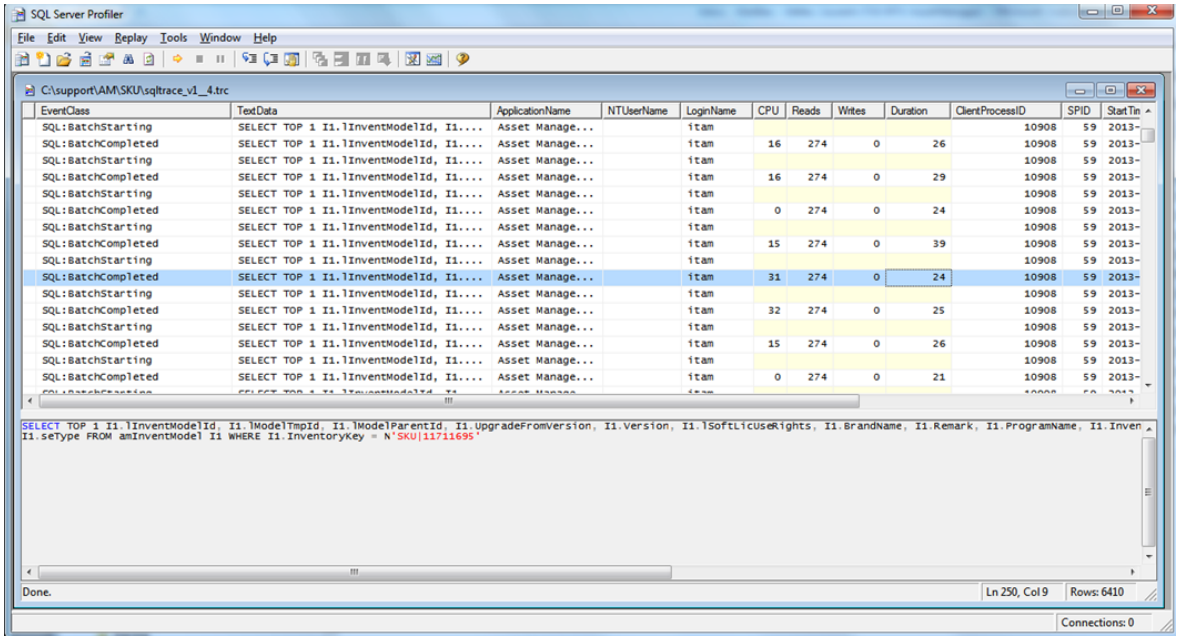
In addition, filters can be set to only capture slow queries or all queries. Slow queries can be defined on a variety of settings, below is an example on duration (it could be on disk or memory reads, CPU time and other parameters).



If the trace generated is containing only slow queries, it is possible to get the access plan to data, with a significant performance overhead. We recommend that IT uses explain plan with caution due to the performance impact on the system. However, it can be provided that the filter is set accordingly to

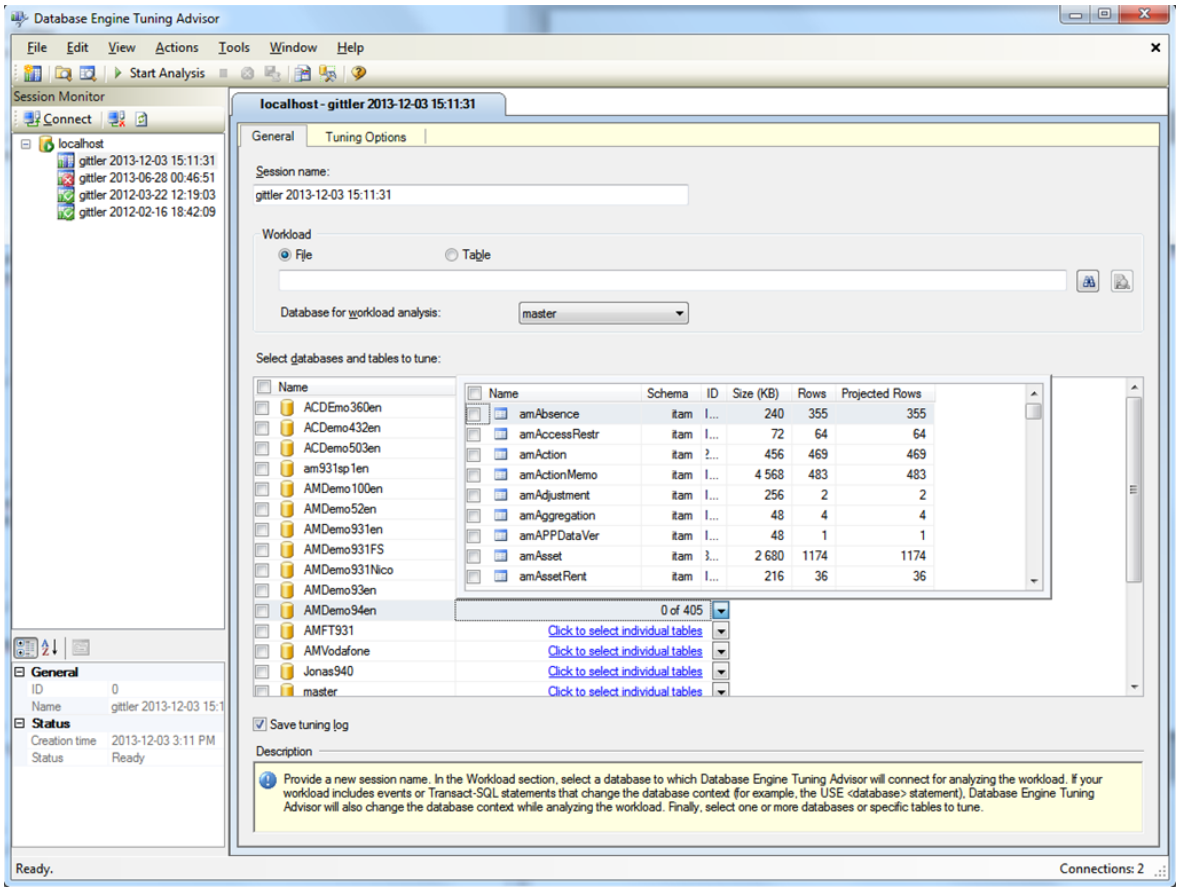
capture only the most offending queries (for example, queries that take longer than 3 second to complete).

The following is an example of a capture trace that does not have very selective filters turned on, therefore capture of the explain plan is turned off.



Database Engine Tuning Advisor

Database Engine Tuning Advisor is a MS SQL Server tool that works with a SQL Profiler trace file as a workload to analyze. It generates tuning and index creation recommendations. Given optimization can be considered to enhance Asset Manager performance, however, they should not be taken for granted. Some of the possible recommendations are not applicable for Asset Manager. They should be always carefully reviewed before they are applied.



The following is an example.

```
use [AMDemo94en]
go

CREATE NONCLUSTERED INDEX [_dta_index_amModel_18_1044250825__K1_45_86] ON [itam].[a
mModel]
(
	[lModelId] ASC
)
INCLUDE ( [Name],
[lUseUnitId]) WITH (SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OF
F, ONLINE = OFF) ON [PRIMARY]
go

CREATE STATISTICS [_dta_stat_1044250825_1_86] ON [itam].[amModel]([lModelId], [lUse
UnitId])
go

CREATE NONCLUSTERED INDEX [_dta_index_amPortfolio_18_1988254188__K1_14_27_32_35] ON
[itam].[amPortfolio]
(
	[lPortfolioItemId] ASC
```

```
)
INCLUDE ( [fQty],
[lAstId],
[lIconId],
[lModelId]) WITH (SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF,
ONLINE = OFF) ON [PRIMARY]
go

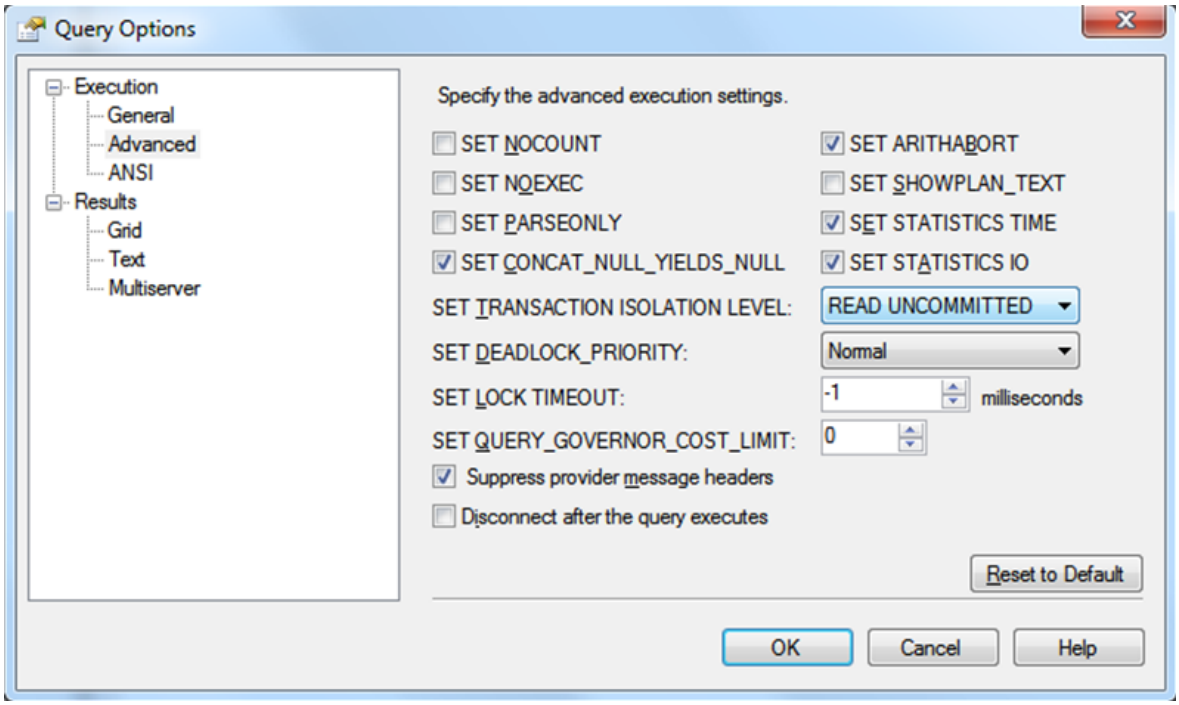
CREATE STATISTICS [_dta_stat_1988254188_27_35_1] ON [itam].[amPortfolio]([lAstId],
[lModelId], [lPortfolioItemId])
go

CREATE NONCLUSTERED INDEX [_dta_index_amAsset_18_8387099__K1_5] ON [itam].[amAsset]
(
       [lAstId] ASC
)
INCLUDE ( [AssetTag]) WITH (SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTI
NG = OFF, ONLINE = OFF) ON [PRIMARY]
go
```

How to get the performance impact of a specific query

In the query tool of SQL Server Management Studio, some options can be set in **Query**/**Query Options**.

One of the most useful one is the Statistics IO.



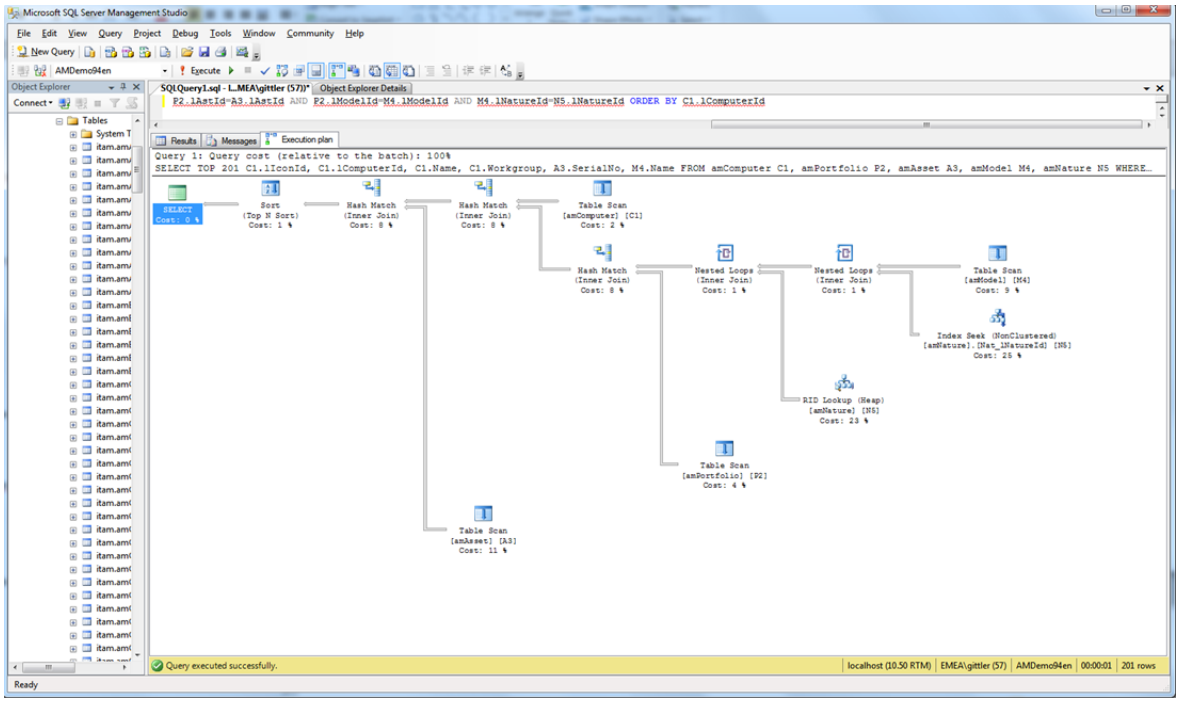
For example, you have the following query:

```
SELECT TOP 201 C1.lIconId, C1.lComputerId, C1.Name, C1.Workgroup, A3.SerialNo, M4.N
ame FROM amComputer C1, amPortfolio P2, amAsset A3, amModel M4, amNature N5 WHERE (
(N5.seComputerType = 0) OR (N5.seComputerType = 3)) AND C1.lItemId=P2.lPortfolioIte
mId AND P2.lAstId=A3.lAstId AND P2.lModelId=M4.lModelId AND M4.lNatureId=N5.lNature
Id ORDER BY C1.lComputerId
```

Run it in the query tool, statistics IO will provide statistics on how many reads are required on each table in the query:

```
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, read-ahead read
s 0, lob logical reads 0, lob physical reads 0, lob read-ahead reads 0.
Table 'amAsset'. Scan count 1, logical reads 172, physical reads 0, read-ahead read
s 0, lob logical reads 0, lob physical reads 0, lob read-ahead reads 0.
Table 'amPortfolio'. Scan count 1, logical reads 56, physical reads 0, read-ahead r
eads 0, lob logical reads 0, lob physical reads 0, lob read-ahead reads 0.
Table 'amNature'. Scan count 0, logical reads 5589, physical reads 0, read-ahead re
ads 0, lob logical reads 0, lob physical reads 0, lob read-ahead reads 0.
Table 'amModel'. Scan count 1, logical reads 131, physical reads 0, read-ahead read
s 0, lob logical reads 0, lob physical reads 0, lob read-ahead reads 0.
Table 'amComputer'. Scan count 1, logical reads 23, physical reads 0, read-ahead re
ads 0, lob logical reads 0, lob physical reads 0, lob read-ahead reads 0.
```

Explain plan (CTRL-M) will show the actual plan of the query executed.



MS SQL Server clustered index

MS SQL server has some very specific settings to generate special clustered indices that should be used with caution.

- Clustered indices physically order the rows in a table based on the clustered index.
- Because rows are physically ordered by the clustered index, only one clustered index can be applied on each table.
- Clustered indices are not supported nor recognized by AMDBA. Therefore, they should be limited to very specific tuning cases.
- When a table has a clustered index, full table scans on that table are replaced by a clustered table scan.
- Clustered indices generally slow down the performance on updates and inserts if the column updated is part of the clustered index because tables are physically ordered according to the values of its clustered index.
- Clustered indices enhance read performance, especially when access to data on the table can take advantage of the clustered indices.
- Clustered (or non-clustered) indices can also avoid some blocking locks on inserts (known as the open range issue without the index).

Parallel queries

- Parallel queries can define how many CPU cores of the MS SQL Server box can be used to execute a query (the greater the faster). This number is given by the **Max degree of parallelism** setting in MS SQL Server (1= No parallelization).
- Parallelization maximizes query performance for slow queries, but reduces the number of concurrent users while a parallel query is being executed (the more CPU cores are taken for one query, the less CPU cores are left available for other concurrent queries).
- Parallel queries can be defined at the MS SQL Server level (in properties of the server).

Data maintenance (true for every DBMS)

- It is very important to optimize and rebuild index regularly (daily/weekly).
- Compute statistics (daily/weekly for fast changing tables).
- Defragment (reorganize) tables to enhance performance, however, this task should be done less often.
- Check database integrity.
- Back up database.
- In development, use simple log method to automatically have the log truncated (not a full log that would make the log file grow indefinitely). In production, it depends on the backup method and the

process used.

- Back up database and/or logs.

Index creation consideration

Selectivity

Index creation follows the same rule on many DBMS. Consider generating multi-column index that will have the most selective column first.

A selective column contains a value that only appears in a few rows in a table.

For example, you have a Boolean column (with value 0 or 1), and you have millions of rows.

- If many rows have the value 0 for the Boolean column, and many rows have the value 1, it is not a selective column.
- If most rows have the value 0 and only a few rows have the value 1, value 1 is selective and a query using a WHERE clause in which the query is looking for the value 1 can potentially be optimized using an index. On the contrary, if the filter in the WHERE clause is using a condition where rows to be retrieved have the value 0, most of the rows will be retrieved and it will be faster to perform a full table scan than trying to retrieve a large quantity of rows through the index.

Statement above is general, having selectivity can be a lot different according to the field type and the number of rows matching each value of the column. Oracle calls this histogram of values.

A non-selective column can also be used as one of the fields of the index (but not the first column of the index) to enhance performance. The first field of the index refers to a selective value in a column.

Index for linguistic sort

Since Asset Manager 9.41, AMDBA includes a feature to create index with a linguistic sort order (with Oracle databases).

For an Oracle database, you can now set the case sensitivity of the database with AMDBA (**Database**/ **Enable case insensitive** in the Application Designer). If you select this option, all string fields in the Oracle database will become case-insensitive. This option has an impact on performance but is valuable for many users.

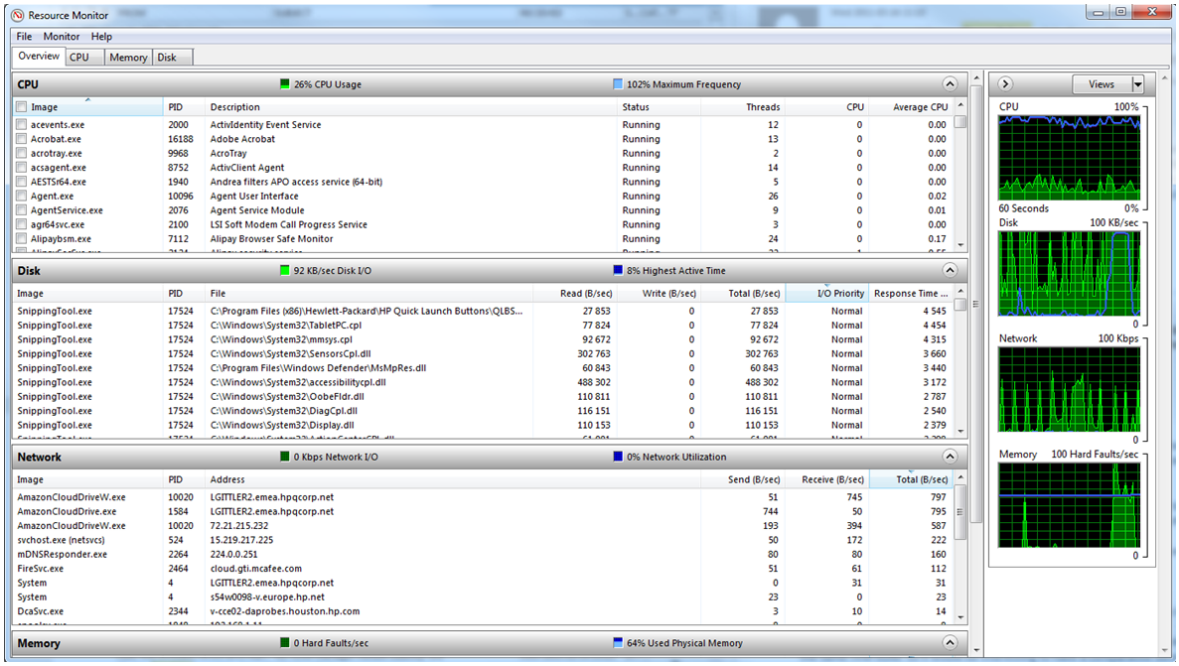
Changes are regularly introduced in Asset Manager. You can check the release notes of the version and all release notes of patches (one release notes per patch) so that you can get all changes since the release. Patches are cumulative, but release notes are not.

Monitoring in production

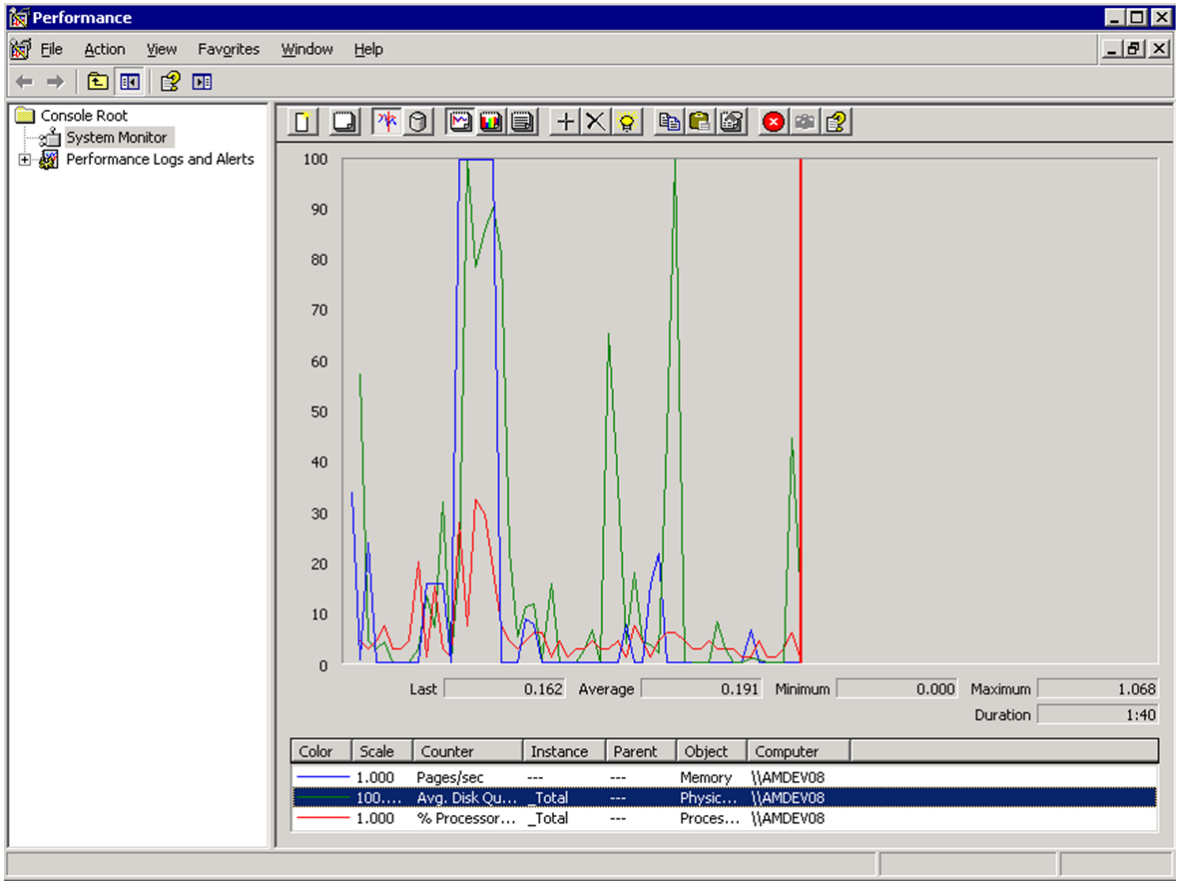
Web server

Operating System

You can use the Windows Performance Monitor counters or Windows Resource Monitor to monitor the operating system.



You can also use perfmon.msc.



System monitoring configuration

For instructions to set up and view performance monitors, see http://technet.microsoft.com/en-us/library/dd744567(v=ws.10).aspx and http://technet.microsoft.com/en-us/library/cc749249.aspx.

System memory

- Memory: % Committed Bytes In Use
- Memory: Page Faults/sec

System Disks (can be logical if the disk is on a Storage Area Network or Grid)

- PhysicalDisk: Disk Read Bytes/sec
- PhysicalDisk: Disk Reads/sec
- PhysicalDisk: Disk Write Bytes/sec
- PhysicalDisk: Disk Writes/sec

- PhysicalDisk: Avg. Disk Queue Length
- PhysicalDisk: Avg. Disk Read Queue Length
- PhysicalDisk: Avg. Disk Write Queue Length
- PhysicalDisk: Queue Size. This is the number of outstanding requests on the disk while the performance data is collected. It also includes requests in service at the time of the collection. This is an instantaneous snapshot, not an average value over the time interval. Multi-spindle disk devices can have multiple requests that are active at one time, but other concurrent requests are awaiting services. This counter might reflect a transitory high or low queue length, but if there is a sustained load on the disk drive, it is likely that this will be consistently high. Requests experience delays proportional to the length of this queue minus the number of spindles on the disks. For good performance, this average difference should be less than two.

System CPU

Notice that CPU performance numbers are not always accurate when hyper-threading is enabled. Hyper-threading is actually doubling the number of logical cores but not the computing power. It increases it by 10%-15%. In case of issues, you can simply disable hyper-threading on CPUs. There are also some CPU generation for which hyper-threading is more sophisticated, as a result, there might be non-linear relations between the used percentage of CPU time and actual computing power factor.

- Processor: % Idle Time. There should be some idle time on the server. If idle time is less than 10%-15%, the CPU of the server is saturated. Notice that some anti-virus or other specific processes reuse the idle time to perform background tasks while minimizing CPU performance. In that case, you can monitor the CPU time of the specific process(es) reusing CPU idle time and use that metric instead to measure idle time. Idle time is also used by some distributed computing processes like SETI, World Community Grid BOINC, and so on.
- Processor: Interrupts/sec
- System: Threads
- System: Queue Length. This is the instantaneous length of the processor queue in units of threads. All processors use a single queue in which threads wait for processor cycles. After a processor is available for a thread waiting in the processor queue, the thread can be switched onto a processor for execution. A processor can execute only a single thread at a time. Notice that faster CPUs can handle longer queue lengths than slower CPUs. The number of threads in the processor queue shows the ready threads only, not the threads that are running. Even multiprocessor computers have a single queue for processor time. Therefore, for multiprocessors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than two threads per processor is normally acceptable, depending on the workload.

System network

Ethernet

- Network: Output Queue Length. This is the length of the output packet queue (in packets). If this is longer than two, there are delays and the bottleneck should be found and eliminated, if possible. If network requests are queued by the Network Driver Interface Specification (NDIS), you should measure it instead.
- Network: Current Bandwidth. This is an estimate of the current bandwidth of the network interface in bits per second (BPS). For interfaces that do not vary in bandwidth or for those where no accurate estimation can be made, this value is the nominal bandwidth. It should be as large as possible.
- Network: Packets Outbound Errors. This is the number of outbound packets that cannot be transmitted because of errors. This should be 0 on a LAN unless there are network issues.
- Network: Packets Received Errors. This is the number of inbound packets that contains errors preventing them from being deliverable to a higher-layer protocol. This should be 0 on a LAN unless there are network issues.

TCP loopback interface

TCP Loopback interface is the network interface used between the Tomcat AM web service and web tier. Same for the inter-process (ICP, …) loopback interface.

For more information, see http://blogs.technet.com/b/wincat/archive/2012/12/05/fast-tcp-loopback-performance-and-low-latency-with-windows-server-2012-tcp-loopback-fast-path.aspx

Tomcat process

For each Tomcat instance (also known as Tomcat process), you need to monitor the process (name of the process to monitor is chosen when installing Tomcat):

- Processor time. In case there is a system CPU issue (Processor: % Idle Time is less than 15 to 10 %), this setting will indicate whether the web tier, web service (separated instance), or AM web (WS+WT in a single Tomcat instance) is taking too much CPU time. This parameter provides you with a baseline to look into issues. You may need to add some CPUs to the server. Ideally, in a stress test, total WS+WT CPU time should be around 80 % or more; in production, that number should vary across user activity and should be bigger when AM web performance is slowing down. If that is not the case, there are probably some operations slowing down the response time on the DBMS server side (lock, slow SQL queries taking time at the Oracle Server time, Server Memory swapping issues, or less probably Network or Disk issues when there is no memory swap problem).
- Virtual bytes. This is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries. It can be

large on the web service when the connection pool is set so that a big number of connections are opened. If that number is not large on the web service, enhance the number of connections left opened in the connection pool (idlesize).

- Working Set. This is the current size, in bytes, of the Working Set of this process. The Working Set is the set of memory pages recently touched by the threads in the process. If the free memory in the computer is above a threshold, pages are left in the Working Set of a process even if they are not in use. When free memory falls below a threshold, pages are trimmed from Working Sets. If they are needed, they will then be soft-faulted back into the Working Set before leaving main memory.

- Thread count. This is the number of threads that are currently active in this process. An instruction is the basic unit of execution in a processor, and a thread is the object that executes instructions. Every running process has at least one thread. It can be used to figure out how many Tomcat sessions are left opened by Tomcat, and how many connections are left opened in the connection pool.

- Page File Bytes. This is the current amount of virtual memory, in bytes, that this process has reserved for use in the paging file(s). Paging files are used to store pages of memory used by the process that are not contained in other files. Paging files are shared by all processes, and the lack of space in paging files can prevent other processes from allocating memory. If there is no paging file, this counter reflects the current amount of virtual memory that the process has reserved for use in physical memory.

- Page Faults/sec. This is the rate at which page faults by the threads executing in this process are occurring. A page fault occurs when a thread refers to a virtual memory page that is not in its working set in main memory. This may not cause the page to be fetched from disk if it is on the standby list and hence already in main memory, or if it is in use by another process with whom the page is shared. It should be very low on both the web service and web tier.

Other specific processes:

- If there are specific processes set up (anti-virus, security stacks, and so on), monitor their CPU time occupation and memory in the same way Tomcat is monitored.

Tomcat

Session statistics on Asset Manager web service

If Asset Manager Web is installed locally with Tomcat on port 8080, the following URL displays the default session timeout for the Asset Manager web tier, and the number of currently active sessions that fall within ten-minute ranges of their actual timeout times.

```
http://localhost:8080/manager/text/sessions?path=/AssetManager
```

The following URL does the same for Tomcat sessions to the Asset Manager web service.

```
http://localhost:8080/manager/text/sessions?path=/AssetManagerWebService
```

For example, after restarting Tomcat and then executing one of the JSP samples in the `/examples` web app, you may get something like this.

```
OK - Session information for application at context path /examples
Default maximum session inactive interval 30 minutes
30 - <40 minutes:1 sessions
```

For more information about Tomcat session statistics, see http://tomcat.apache.org/tomcat-7.0-doc/manager-howto.html#Session_Statistics.

Server status

- From the Server Status tomcat console link, you can view the information about the server.
  - Server and JVM version number, JVM provider, OS name and number followed by the architecture type.
  - Several information about the memory usage of the JVM (available, total and max memory).
- A second list gives information about the Tomcat AJP and HTTP connectors. The same information is available for both of them:
  - Threads information: Max threads, min and max spare threads, current thread count and current thread busy.
  - Request information: Max processing time and processing time, request and error count, bytes received and sent.
- A table showing Stage, Time, Bytes Sent, Bytes Receive, Client, VHost, and Request. All existing threads are listed in the table. The following is the list of the possible thread stages:
  - "Parse and Prepare Request": The request headers are being parsed or the necessary preparation to read the request body (if a transfer encoding has been specified) is taking place.
  - "Service": The thread is processing a request and generating the response. This stage follows the "Parse and Prepare Request" stage and precedes the "Finishing" stage. There is always at least one thread in this stage (the server-status page).
  - "Finishing" : The end of the request processing. Any remainder of the response still in the output buffers is sent to the client. This stage is followed by "Keep-Alive" if it is appropriate to keep the connection alive or "Ready" if "Keep-Alive" is not appropriate.
  - "Keep-Alive": The thread keeps the connection open to the client in case the client sends another request. If another request is received, the next stage will be "Parse and Prepare Request". If no request is received before the keep alive times out, the connection will be closed and the next stage will be "Ready".
  - "Ready": The thread is at rest and ready to be used.

For more information, see http://tomcat.apache.org/tomcat-7.0-doc/manager-howto.html#Server_Status.

Monitoring

Tomcat 7 includes a monitoring documentation:

http://tomcat.apache.org/tomcat-7.0-doc/monitoring.html

Implying mostly JMX, which can be used by Jconsole, through:

manager-jmx — Access to JMX proxy interface and to the "Server Status" page as seen in:

http://tomcat.apache.org/tomcat-7.0-doc/manager-howto.html

Load balancing

Several ways can be used to implement and monitor the Tomcat web balancing. As a baseline, refer to the how-to documentation: http://tomcat.apache.org/tomcat-7.0-doc/balancer-howto.html

Load Balancing using the JK native Tomcat connector is documented as follows (for Tomcat 7):

How to: http://tomcat.apache.org/connectors-doc/generic_howto/loadbalancers.html

Status Worker: http://tomcat.apache.org/connectors-doc/reference/status.html

For more information about Apache HTTP server, see Using Apache HTTP Server 2.x with mod_proxy.

Java engine (JVM)

It is useful to monitor the JVM by tuning the Java parameters (Eden Space, Heap, Stack, Garbage collector) in a production or test environment.

Monitoring will be somehow limited on other topics like profiling and tuning performance on the code side, because this class of monitoring is limited to developers using the source code and reverse engineering is prohibited on Asset Manager.

However, some tools can be used as a good baseline to monitor JVM performance and tune JVM settings.

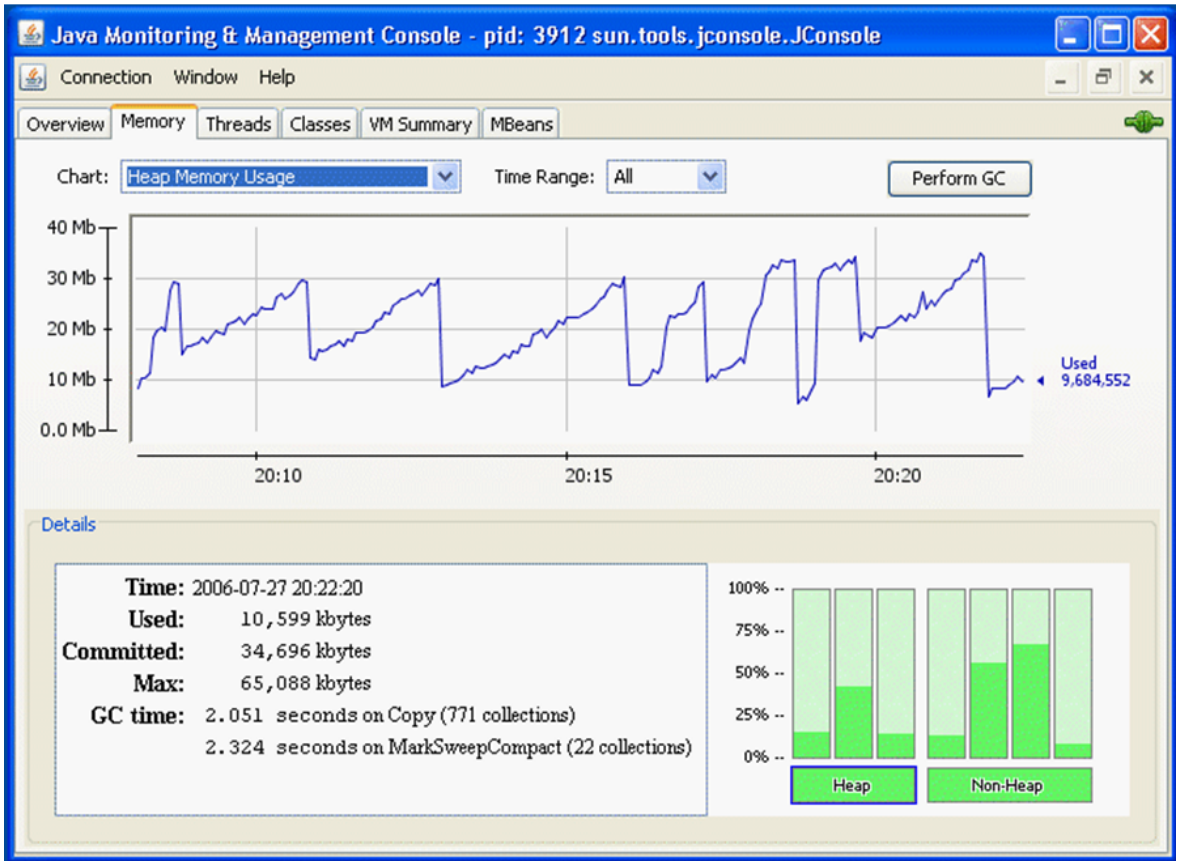
Jconsole

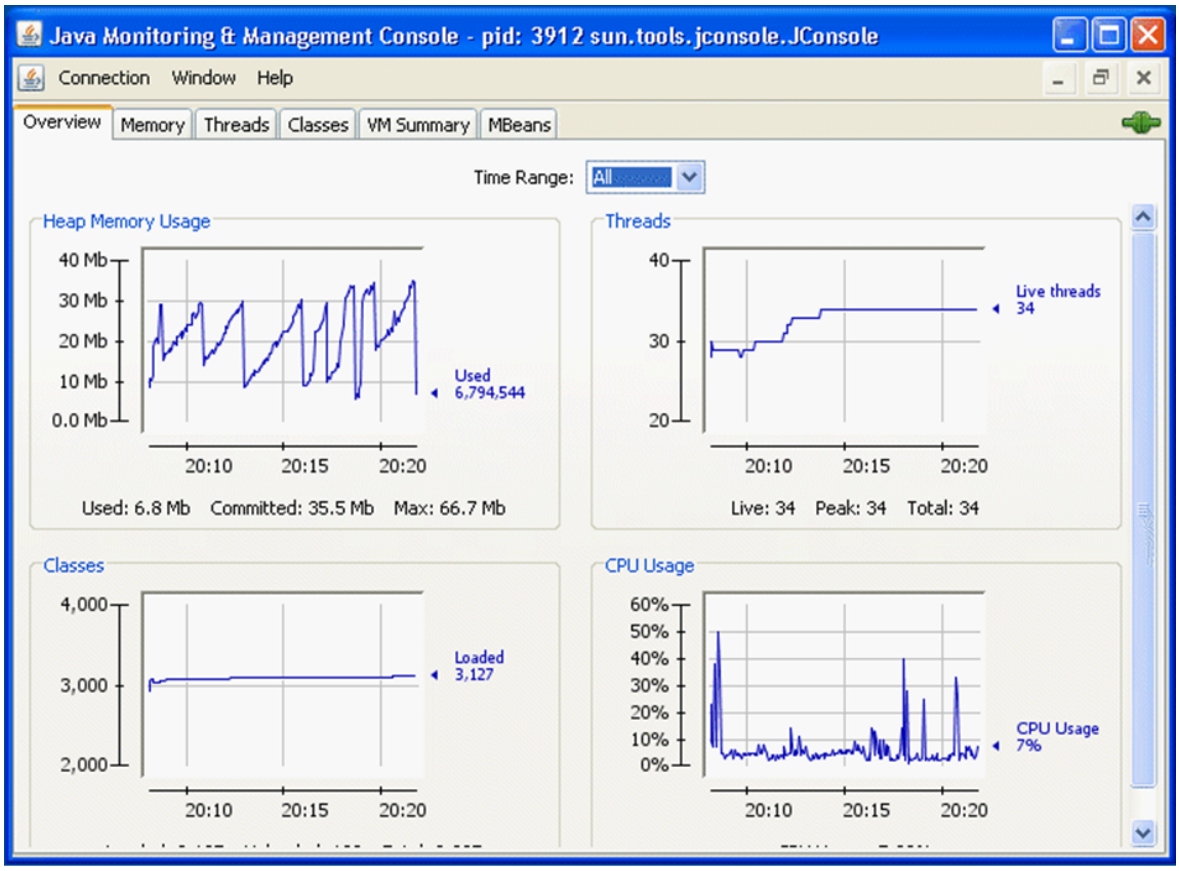
Jconsole can be used to monitor the Tomcat JVM that runs AssetManager web service and web tier.

Some JMX connectors can be defined in the Java starting command like:

```
-Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.port=8484
-Dcom.sun.management.jmxremote.ssl=false
```

Then, it will be possible to start Jconsole as follows:


Java Monitoring & Management Console - pid: 3912 sun.tools.jconsole.JConsole
Connection
Window
Help
Overview
Memory
Threads
Classes
VM Summary
MBeans
Chart:
Heap Memory Usage
Time Range:
All
Perform GC
40 Mb
30 Mb
20 Mb
10 Mb
0.0 Mb
20:10
20:15
20:20
Used
9,684,552
Details
Time: 2006-07-27 20:22:20
Used: 10,599 kbytes
Committed: 34,696 kbytes
Max: 65,088 kbytes
GC time: 2.051 seconds on Copy (771 collections)
2.324 seconds on MarkSweepCompact (22 collections)
100%
75%
50%
25%
0%
Heap
Non-Heap



JVM monitoring outside Jconsole

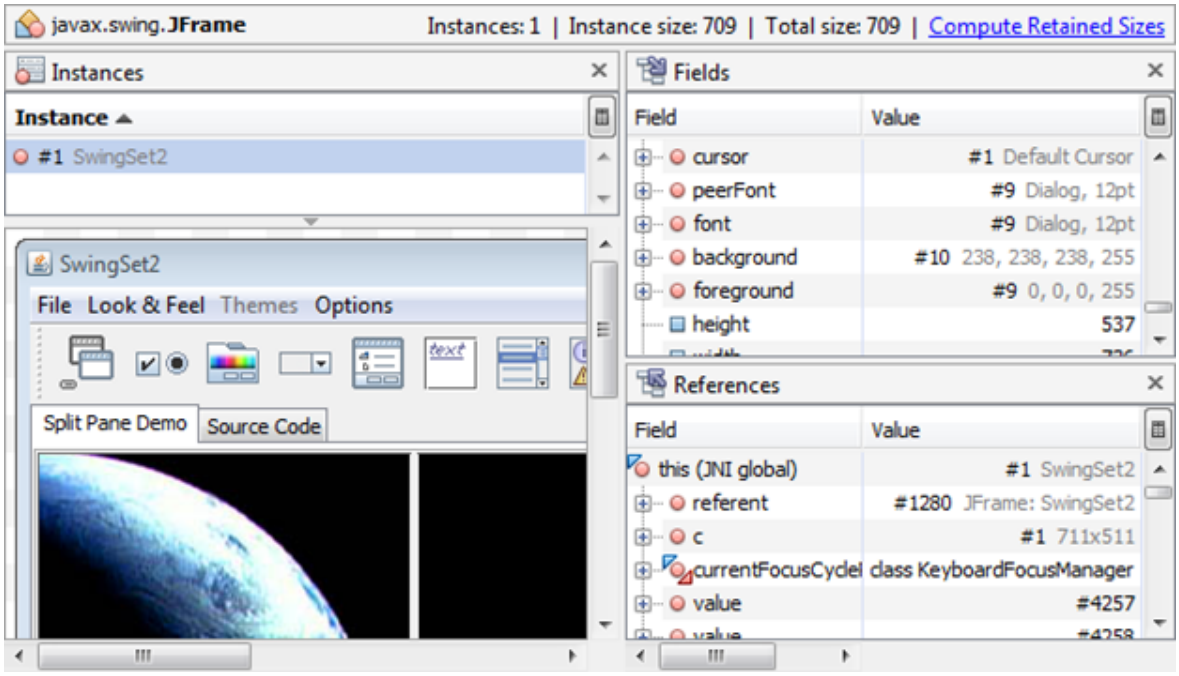
- Monitor full and partial garbage collection (can be set by Java JVM flags)
- Monitor heap memory usage
- Monitor stack usage
- Monitor overall memory consumption (see system monitoring above).
  - Eden Space (heap): The pool from which memory is initially allocated for most objects.
  - Survivor Space (heap): The pool containing objects that have survived the garbage collection of the eden space.
  - Tenured Generation (heap): The pool containing objects that have existed for some time in the survivor space.
  - Permanent Generation (non-heap): The pool containing all the reflective data of the virtual machine itself, such as class and method objects. With Java VMs that use class data sharing, this generation is divided into read-only and read-write areas.
  - Code Cache (non-heap): The HotSpot Java VM also includes a code cache, containing memory that is used for compilation and storage of native code.

For more information, see http://docs.oracle.com/javase/7/docs/technotes/guides/management/jconsole.html.

An alternative to Jconsole

In alternative, VisualVM can be considered to monitor the JVM among a variety of JVM monitoring tools.

- Application Developer: Monitor, profile, take thread dumps, browse heap dumps
- System Administrator: Monitor and control Java applications across the entire network
- Java Application User: Create bug reports containing all the necessary information



Note for virtualized environments

For virtual machines, all the measures above apply to the logical machine itself (logical power and logical memory allocation).

Other monitoring tools must be used to ensure that virtual machine resource metering is significant. Those tools will be used to meter the measure the resource pool resource to determine the actual CPU power, memory allocation, disk resource.

For example, a logical system showing idle time of 10% might mean the CPUs of the physical server in the resource pool are saturated. Typically, when the CPU of a physical member of the resource pool hosts several VMs, only a fraction of the host CPU power will be allocated to the AM web server VM, but the CPU of the VM might be seen with its CPU saturated.

For more information, see:

- http://www8.hp.com/us/en/software-solutions/software.html?compURI=1172041#.U8l4lfmSw8k
- http://www.hp.com/hpinfo/newsroom/press_kits/2011/HPFortCollins/HP_Data_Center_Smart_Grid.pdf
- http://www8.hp.com/us/en/software-solutions/it-automation-orchestration/
- http://technet.microsoft.com/en-us/library/hh831415.aspx
- http://www.vmware.com/fr/products/vcenter-operations-management

DBMS monitoring

There are many enterprise monitoring consoles and tools for Oracle.

This manual describes a few methods and tools to monitor the database performance.

Oracle provides three widely used monitoring tools:

- Automated Workload Repository (AWR)
- Automatic Database Diagnostic Monitor (ADDM)
- Oracle Enterprise Console

An oracle suite: Oracle Diagnostics Pack 11g provides the required tools. For more information, see http://www.oracle.com/us/products/enterprise-manager/diagnostic-pack-11g-ds-068465.pdf.

There are a lot of other tools to monitor Oracle performance, including SQL scripts that can be run directly.

Optimizer_MODE

DBA: Before going more in depth in the Oracle tuning, carefully choose the Optimzer_MODE. Make sure that the Optimizer_MODE is set to FIRST_ROWS_1 ~ FIRST_ROWS_1000 (FIRST_ROWS_10 is a good choice to start). Changes to FIRST_ROWS_1 or FIRST_ROWS_100 might be tested as early iterations of tuning sessions.

DBA: Optimizer_index_cost_adj can be set to 25 (by default, it is 100) to instruct the optimizer to use index more often than table scans.

Automatic Database Diagnostic Monitor (ADDM)

Automatic Database Diagnostic Monitor (ADDM) starts its analysis by focusing on the activities that cost the most time of the database and then drills down through a sophisticated problem classification tree to determine the root causes of problems.

ADDM can report symptoms, and most often discover the actual cause behind performance problems. ADDM embeds a problem classification tree built on performance tuning experience of Oracle's own performance experts, and it is designed to accurately diagnose the most frequently seen problems,

such as CPU/IO bottlenecks, poor connection management, undersized memory, resource intensive SQL statements, lock contention, and so on.

Automated Workload Repository (AWR)

The Automatic Workload Repository (AWR) replaces STATPACKS.

AWR contains operational statistics about a particular database and other relevant information. At regular intervals (once an hour by default), the database takes a snapshot of all its vital statistics and workload information and stores them in AWR.

AWR is very useful for problem detection and self-tuning purposes. Data is stored in both the memory and the database. The collected data can be displayed in both reports and views.

The statistics collected and processed by AWR include:

- Object statistics that determine both access and usage statistics of database segments.
- Time model statistics based on the time usage for activities, displayed in the V$SYS_TIME_MODEL and V$SESS_TIME_MODEL views.
- Some of the system and session statistics collected in the V$SYSSTAT and V$SESSTAT views.
- SQL statements that are producing the highest load on the system, based on criteria such as elapsed time, CPU time, and read operations on disk (long elapsed time) or in Memory (large CPU time).
- Active Session History (ASH) statistics, representing the history of recent activities of sessions.

For more information, see http://docs.oracle.com/cd/E11882_01/server.112/e16638/autostat.htm#PFGRF027.

Oracle Diagnostics Pack 11g comprehensive system monitoring

Oracle Diagnostics Pack 11g automatically examines the vital signs of different components, such as databases, individual instances, and host operating systems. It stores the required historical information to provide administrators with a long-term view of their system behavior. It also helps them administer service level goals more effectively.

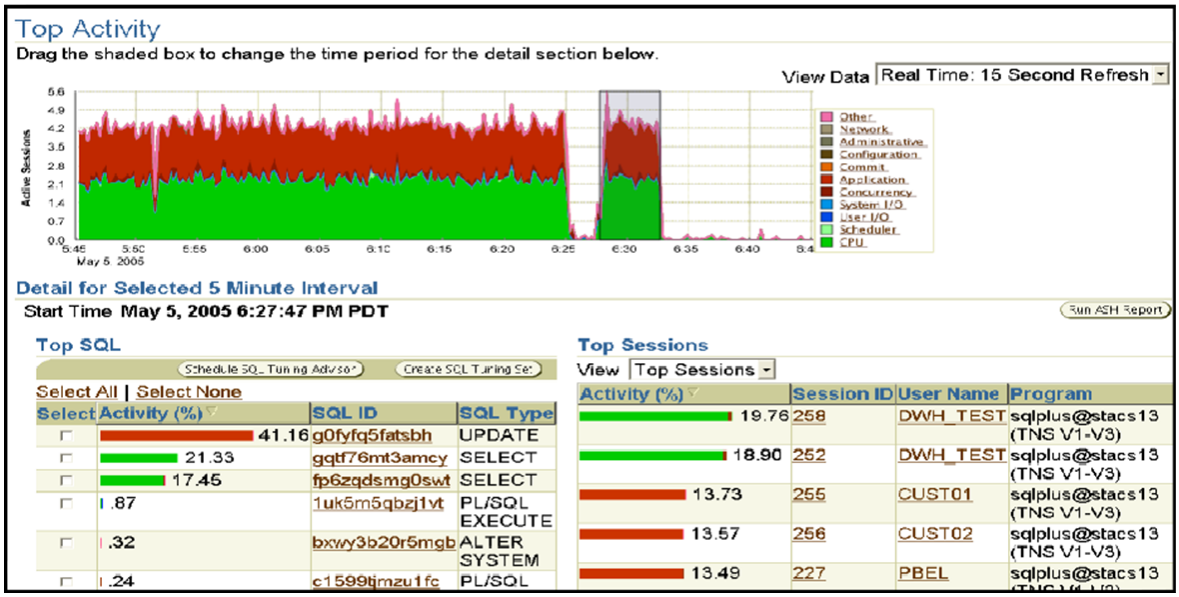
The Database Performance Page displays host information, user activity, and throughput information on a common screen, for easy correlation. With this information, the DBA can verify that the machine has sufficient CPU and memory resources before analyzing the database.

Active Sessions graph shows how much CPU resource the users are consuming and if there are users waiting for resources instead of running on the CPU. A throughput graph can be used to determine if the throughput is affected by machine resources, CPU consumption, or resource contention.

The Host Performance Page provides a quick glimpse of the CPU, memory, and disk bandwidth utilization at the machine level. By using the information presented on this page and associated drill-downs, the administrator can find out details about how machine resources are being used and which users or applications are consuming most of the system resources.

ASH

Active Session History (ASH) is a key component of AWR. ASH samples the current state of all active sessions every second and stores it in memory. The data collected in memory can be accessed by a V$ view. This sampled data is also pushed into AWR every hour for performance diagnoses. ASH enables performance analysis of problems that occur for a very short duration. It replaces the need to use utilities like SQL trace.



For DBAs, see *Appendix 4* for a SQL script that performs most historical statements.

Advanced Event Notification

The Oracle Database 11g provides a built-in, push-based alerting mechanism.

Oracle Diagnostics Pack 11g extends this alerting capability by allowing administrators to be notified when they are away from their desks.

As alerts are being generated, the Oracle Enterprise Manager framework provides an advice-driven intuitive response system that walks the administrator through alerts resolution, including capabilities to set up automated responses appropriately.

Oracle cache ratios

DBAs can find AWR or SQL scripts in *Appendix 1: Oracle instance Monitoring SQL scripts* report cache ratios statistics.

- The Database hit cache ratio should be greater than 95 % (the closer to 100%, the better).
- The library hit cache ratio should be greater than 98 % (the closer to 100%, the better).

- The number of sorts on disk should be as small as possible. Sorts on disk are very expensive for the Oracle server, they slow down the performance and queue disks I/Os.

SQL query performance tuning

For DBAs, AWR, ADDM, and SQL scripts such as given in *Appendix 1: Oracle instance Monitoring SQL scripts* report slowest queries.

For DBAs and AMAPPs, when SQL queries with a big impact on the DBMS server are listed in several lists of most offending queries in the AWR report, we recommend that you address them first.

AMAPP and DBA Teams should work together to address those queries and tune them.

You can use the following methods to tune the queries:

- Add Oracle hints (AM has limited capabilities), only in lists with the force index check box in the list configuration.
- Add a multi-column index.
- Change the query whenever possible.
- Use a stored execution plan.

Tuning in production is an iterative process that changes depending on database volume (table volume) and processes. It can be done through DBAs for queries, and can be done also by the AMAPP team whenever the SQL generated can be changed customizing AM Queries, list configuration, filters, access rights, wizard and script code.

At the beginning, it is often seen that some queries are generating a large percentage of the server CPU and disk usage. Tuning those queries in priority can dramatically improve performance.

Multi-column indices are considered when there is a way to set the most selective column first.

When the solution is to add a multi-column index, you can test it by adding the index directly in the DBMS backend, and then see the performance boost afterwards in new AWR snapshots or Oracle tkprofed server traces (see appendix for a way to perform these). When the multi-column index has been proven to be effective through traces, we recommend that you add it in production through AMDBA whenever possible. If this is not possible like an index on a function, add it on the DBMS side and maintain a SQL generation script that will be applied on the DBMS side.

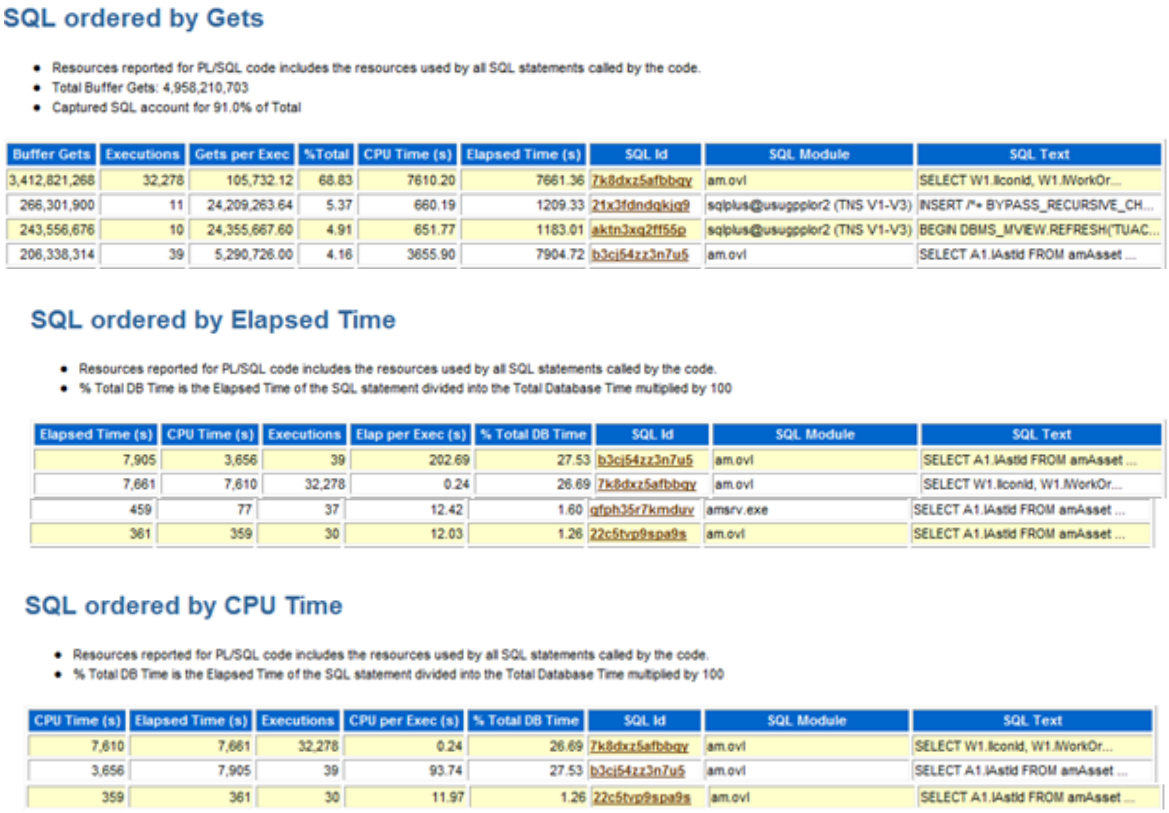
In addition, we always recommend that you generate an index rebuilt script for all AM indices (those created by AMDBA and others) to do maintenance tasks. It can reduce fragmentation and maintain performance.

An example of SQL tuning

Monitoring and Analysis

Below is a small example of queries of Asset Manager in an AWR report.

```
Summary of ARW \thursday\awrrpt_1_17221_17225.html\awrrpt_1_17221_17225.html
```



SQL ordered by Gets

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- Total Buffer Gets: 4,958,210,703
- Captured SQL account for 91.0% of Total

Buffer Gets	Executions	Gets per Exec	%Total	CPU Time (s)	Elapsed Time (s)	SQL Id	SQL Module	SQL Text
3,412,821,268	32,278	105,732.12	68.83	7610.20	7661.36	7k8dxz5afbbqy	am.ovl	SELECT W1.lIconId, W1.lWorkOr...
266,301,900	11	24,209,263.64	5.37	660.19	1209.33	21x3fdndqkjq9	sqlplus@usugpplor2 (TNS V1-V3)	INSERT /*+ BYPASS_RECURSIVE_CH...
243,556,676	10	24,355,667.60	4.91	651.77	1183.01	aktn3xq2ff55p	sqlplus@usugpplor2 (TNS V1-V3)	BEGIN DBMS_MVIEW.REFRESH('TUAC...
206,338,314	39	5,290,726.00	4.16	3655.90	7904.72	b3cj54zz3n7u5	am.ovl	SELECT A1.lAstId FROM amAsset ...

SQL ordered by Elapsed Time

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- % Total DB Time is the Elapsed Time of the SQL statement divided into the Total Database Time multiplied by 100

Elapsed Time (s)	CPU Time (s)	Executions	Elap per Exec (s)	% Total DB Time	SQL Id	SQL Module	SQL Text
7,905	3,656	39	202.69	27.53	b3cj54zz3n7u5	am.ovl	SELECT A1.lAstId FROM amAsset ...
7,661	7,610	32,278	0.24	26.69	7k8dxz5afbbqy	am.ovl	SELECT W1.lIconId, W1.lWorkOr...
459	77	37	12.42	1.60	qfph35r7kmduv	amsrv.exe	SELECT A1.lAstId FROM amAsset ...
361	359	30	12.03	1.26	22c5tvp9spa9s	am.ovl	SELECT A1.lAstId FROM amAsset ...

SQL ordered by CPU Time

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- % Total DB Time is the Elapsed Time of the SQL statement divided into the Total Database Time multiplied by 100

CPU Time (s)	Elapsed Time (s)	Executions	CPU per Exec (s)	% Total DB Time	SQL Id	SQL Module	SQL Text
7,610	7,661	32,278	0.24	26.69	7k8dxz5afbbqy	am.ovl	SELECT W1.lIconId, W1.lWorkOr...
3,656	7,905	39	93.74	27.53	b3cj54zz3n7u5	am.ovl	SELECT A1.lAstId FROM amAsset ...
359	361	30	11.97	1.26	22c5tvp9spa9s	am.ovl	SELECT A1.lAstId FROM amAsset ...

The common queries in several lists are b3cj54zz3n7u5 and 22c5tvp9spa9s.

b3cj54zz3n7u5:

```
SELECT A1.lAstId FROM amAsset A1, amPortfolio P2 WHERE Upper(A1.SerialNo) = :1 AND
A1.seFinManagement IN (:2, :3, :4) AND P2.lAstId > :5 AND P2.lCompanyId IN ( SELECT
R3.lCpyId FROM amRelServedGrante R3, amEmplDept amEmplDept_CurrentUser WHERE R3.lEm
plDeptId = amEmplDept_CurrentUser.lEmplDeptId AND amEmplDept_CurrentUser.lEmplDeptI
d = 139187643 ) AND A1.lAstId=P2.lAstId
```

22c5tvp9spa9s:

```
SELECT A1.lAstId FROM amAsset A1, amPortfolio P2 WHERE Upper(A1.BarCode) = :1 AND A
1.seFinManagement IN (:2, :3, :4) AND P2.lAstId > :5 AND P2.lCompanyId IN ( SELECT
R3.lCpyId FROM amRelServedGrante R3, amEmplDept amEmplDept_CurrentUser WHERE R3.lEm
plDeptId = amEmplDept_CurrentUser.lEmplDeptId AND amEmplDept_CurrentUser.lEmplDeptI
d = 139187643 ) AND A1.lAstId=P2.lAstId
```

Tuning

We recommend that you add the following indices directly in the DBMS backend. Those indices cannot be created in AMDBA as they are indices implying a function (column).

For the queries b3cj54zz3n7u5 and 22c5tvp9spa9s:

- Create index AstUSerialNo on amAsset(Upper(SerialNo));
- Create index AstUBarCode on amAsset (Upper(BarCode));

Oracle OS Monitoring

Like Oracle database, the operating system of the server should be monitored as well.

If the database server is running on Windows, you can refer to the above section for OS monitoring on the AM web server.

When the Oracle database is running on UNIX, the monitoring utilities depend on the UNIX flavor. There are also numerous monitoring tools in the market, some of which are HP software.

When monitoring the key parameter, you will find the same counters as for Windows above (CPU, Memory swap, Context switch, threads, and processes).

There are also some specific parameters to monitor to make sure that the OS is performing well and optimized for the database operations:

- Processor queue length (vmstat)
- Disk cache (% memory allocated to disk cache, if Oracle database is not set on a raw disk volume)
- Network tuning (TCP packet wait, as known as TCP delay)
- Disk usage
- Fragmentation and journaling on disk, depending on the file system
- Temp space
- Disk system for Oracle logging

vmstat, iostats, sar, fsstat, swap, netstat, prstat/top, gnome-perfmeter, and gnome-system-monitor are some utilities among many others that can be run directly on the system to capture information. For more information, see:

- http://www.dba-oracle.com/t_monitoring_unix_cpu.htm
- https://wikis.oracle.com/display/OTNTaskFinder/Performance+Monitoring

For more information about OS specific tuning on UNIX (SAR, top, vmstat, and so on), see http://www.dba-oracle.com/t_monitoring_unix_cpu.htm.

All those parameters can be extended according to the OS specific documentation (and IBM Red Books regarding AIX).

Asset Manager database monitoring

ID generation

For AMAPP, make sure that the database IDs for each ID field in tables (for example, lEmplDeptId of the amEmplDept table) (incremental sequence am) is not getting close to 1 G.

If that is the case, the Asset Manager database needs to be defragmented (amdba – iddefrag <parameters>).

Software installations

For AMAPP, we recommend that you set the software installations as bcompact. As documented in the Asset Manager documentation.

When software installations are set as non-compact, each software installation is generating a record in both amPortfolio and amSoftinstall tables. When software installations are compact, one records is generated in amSoftinstall, but no record is generated in amPortfolio.

Temporary tables

For AMAPP and DBAs, there are a number of tables storing temporary data that needs to be cleaned out periodically, depending on the time required to store those data in the database (2 weeks is a good baseline value).

Tables:

- Workflows events (amWfOccurEvent, amWfInstance). Workflow should be set as transient when not trace is required.
- Receive process for items that have been fully received amItemReceived.
- SAM counter storage in SAM calculation processes.
- Software installs for non-retrieved software (bdisappeared flag turned on amSoftinstall).

Database volume

For AMAPP and DBAs, when tables are getting large, items can probably be archived. Refer to the AM documentation for more information.

Tables need to be partially archived when there are performance issues on the DBMs on those tables, and SQL queries are getting poor performance with no tuning capability.

Database backup

For DBAs, we recommend that you keep regular backups of the database through the DBMS backend.

Concurrent operations on database schema change

For AMAPP, there should be no concurrent changes on the database schema in AMDBA done concurrently. AMDBA should be used by 1 single user in case of change of structure.

A change of structure in AMDBA will require all programs to be shut down (including Connect-it, amsrv and AM web server) before the change is done.

Maintenance

For DBAs, regular DBA operations should be performed to rebuild index and defragment the tables when required.

All index having a blevel of 3 should be rebuilt (see Appendix 1).

Logs

Logs should be checked for errors.

This includes both application logs and OS/DBMs logs:

- Tomcat logs
- Amsrv logs
- Connect-It logs
- All other external logs dealing with the am database, including Oracle logs (alert.log)

Appendix 1: Oracle instance monitoring SQL scripts

Oracle Instance key numbers

REM Oracle instance parameters

```
 select name, value from v$parameter where ISDEFAULT <> 'TRUE';
 col reads_per_exec   format 999,999,999
 col disk_reads   format 999,999,999
 col gets_per_exec   format 999,999,999
 col buffer_gets   format 999,999,999
 col hit_ratio   format 999,999,999
 col sorts   format 999,999,999
```

REM GET Most offending Queries

```
 BREAK ON EXECUTIONS ON READS_PER_EXEC ON DISK_READS ON GETS_PER_EXEC ON BUFFER_GET
S ON HIT_RATIO ON SORTS ON USERS_EXECUTING ON ADDRESS SKIP 2

 SELECT rownum, executions, round(disk_reads / executions, 2)
        reads_per_exec,
        disk_reads,
        round(buffer_gets / executions, 2) gets_per_exec,
        buffer_gets,
        round((buffer_gets - disk_reads) / buffer_gets, 2)*100
        hit_ratio,
        sorts,
        users_executing,
        PARSING_SCHEMA_ID,
        OPTIMIZER_MODE,
        t.address,
        t.sql_text
 FROM v$sqlarea a, v$sqltext t
 WHERE executions >0
       AND (buffer_gets >100000 or disk_reads>100000)
       AND a.hash_value=t.hash_value
 ORDER BY t.hash_value, t.piece;

set pagesize 9999
set linesize 2000
```

Rem Database Hit Ratio

```
select    (1 - (sum(decode(name, 'physical reads',value,0)) /
          (sum(decode(name, 'db block gets',value,0)) +
          sum(decode(name, 'consistent gets',value,0)))))
```

```
          * 100 "Hit Ratio"
from      v$sysstat;
```

Rem

```
select    (1-(sum(getmisses)/sum(gets))) * 100 "Hit Ratio"
from      v$rowcache;

Rem Hit ratio of Librray cache
select    Sum(Pins) / (Sum(Pins) + Sum(Reloads)) * 100  "Hit Ratio"
from      v$librarycache;
```

Rem Disk SORTS vs Memory sorts

```
select     a.value "Disk Sorts", b.value "Memory Sorts"
from       v$sysstat a, v$sysstat b
where      a.name = 'sorts (disk)'
and           b.name = 'sorts (memory)';



 col PHYRDS  format 999,999,999
 col PHYWRTS  format 999,999,999
 col READTIM   format 999,999,999
 col WRITETIM   format 999,999,999
 col name format a40

select    name, phyrds, phywrts, readtim, writetim
from      v$filestat a, v$dbfile b
where     a.file# = b.file#
order by  readtim desc;


col Value   format 999,999,999,999,999
Rem v$sysstat raw numbers
select * from v$sysstat;
```

Rem v$sgastat raw numbers

```
select * from v$sgastat;
```

REM identify index for which blevel=3 (need to rebuild those), could possibly consider 2 and 3 as well

```
select TABLE_NAME,
INDEX_NAME,
 INDEX_TYPE,
 BLEVEL,
 STATUS
from
 dba_indices
where
```

```
 blevel>2 and
 TABLE_OWNER= Upper('&&AC_Db_Owner') Order by TABLE_NAME, INDEX_NAME;
```

REM Capture OS Busy Time (OS Usage)

```
select
   'OS Usage' series, to_char(snaptime, 'yyyy-mm-dd hh24 hh24:mi') snap_time,
   round(busydelta / (busydelta + idledelta) * 100, 2) "CPU Use (%)"
from
   (select
      s.begin_interval_time snaptime,
      os1.value - lag(os1.value) over (order by s.snap_id) busydelta,
      os2.value - lag(os2.value) over (order by s.snap_id) idledelta
   from
      dba_hist_snapshot s,
      dba_hist_osstat os1,
      dba_hist_osstat os2
   where
      s.snap_id = os1.snap_id
   and
      s.snap_id = os2.snap_id
   and
      s.instance_number = os1.instance_number
   and
      s.instance_number = os2.instance_number
   and
      s.dbid = os1.dbid and s.dbid = os2.dbid
   and
      s.instance_number = (select instance_number from v$instance)
   and
      s.dbid = (select dbid from v$database)
   and
      os1.stat_name = 'BUSY_TIME'
   and
      os2.stat_name = 'IDLE_TIME');
```

REM INDEX USAGE

```
col c1 heading ?Object|Name?       format a30
col c2 heading ?Operation?         format a15
col c3 heading ?Option?            format a15
col c4 heading ?Index|Usage|Count? format 999,999

break on c1 skip 2
break on c2 skip 2

select
   p.object_name c1,
   p.operation   c2,
   p.options     c3,
   count(1)      c4
```

```
from
   dba_hist_sql_plan p,
   dba_hist_sqlstat s
where
   p.object_owner =Upper('&&AC_Db_Owner')
and
   p.operation like '%INDEX%'
and
   p.sql_id = s.sql_id
group by
   p.object_name,
   p.operation,
   p.options
order by
   1,2,3;
spool off;
```

REM Oracle index monitoring

Ensure index monitoring capability can be turned on on the server, see Oracle documentation.

REM STEP 1

Generate Oracle script to turn Index monitoring on.

```
set pages 999;
set heading off;
set echo off;
spool c:\temp\run_monitor.sql
select
   'alter index '||owner||'.'||index_name||' monitoring usage;'
from
   dba_indices
where
   owner = Upper('&&AC_Db_Owner') and index_name not like 'SYS_%';
spool off
```

REM STEP 2

Run the script.

```
set echo on
spool c:\temp\run_monitor_state.LOG
@c:\temp\run_monitor.sql
spool off;
```

REM STEP 3 (to do some days later)

Retrieve index usage for monitored index.

```
spool c:\temp\index_usage.log
select
```

```
   index_name
   mon,
   used
from
   v$object_usage;


col c1 heading ?Object|Name?       format a30
col c2 heading ?Operation?         format a15
col c3 heading ?Option?            format a15
col c4 heading ?Index|Usage|Count? format 999,999

break on c1 skip 2
break on c2 skip 2

select
   p.object_name c1,
   p.operation   c2,
   p.options     c3,
   count(1)      c4
from
   dba_hist_sql_plan p,
   dba_hist_sqlstat s
where
   p.object_owner =Upper('&&AC_Db_Owner')
and
   p.operation like '%INDEX%'
and
   p.sql_id = s.sql_id
group by
   p.object_name,
   p.operation,
   p.options
order by
   1,2,3;

spool off
```

Appendix 2: Example of a query optimization

Consider a query for a tuning session

This query is not impacting the server too much in terms of Oracle memory read (buffer_gets). However, it is executed very often, at the end of the day, it takes more than 4% of the Oracle Server CPU and can be optimized.

```
SELECT W1.lWorkItemId, W2.lWfInstanceId FROM amWfWorkItem W1, amWfInstance W2 WHERE
W1.lDocRecordId = 28836763 AND W1.lActivId = 1417 AND W1.seStatus = 0 AND W2.seStat
us = 0 AND W1.lWfInstanceId=W2.lWfInstanceId;
```

Before index

DBA:

Execution Plan

```
0      SELECT STATEMENT Optimizer=CHOOSE (Cost=1195 Card=1 Bytes=36 )
   1    0   NESTED LOOPS (Cost=1195 Card=1 Bytes=36)
   2    1     TABLE ACCESS (FULL) OF 'AMWFWORKITEM' (Cost=1193 Card=1   Bytes=27)
   3    1     TABLE ACCESS (BY INDEX ROWID) OF 'AMWFINSTANCE' (Cost=2  Card=1 Bytes
=9)
   4    3       INDEX (UNIQUE SCAN) OF 'WFINST_LWFINSTANCE' (UNIQUE) ( Cost=1 Card=
1)
Statistics..
      12400  consistent gets
      12389  physical reads
…
          0  sorts (memory)
          0  sorts (disk)
          0  rows processed
```

Analysis

For DBA, you can consider a multi-column index.

- Always place the more selective field (i.e. the condition for which the less rows will be retrieved at the first place, then the second more selective one at the second place, and so on).
- For this example, the choice will be a covering index. An index on just (ldocrecordid, lactivid, sestatus) can be considered as well.

Tested solution is to add a multi-column index.

```
create unique index workwfitem_20051124 on amwfworkitem(ldocrecordid, lactivid, ses
tatus, lwfinstanceid, lworkitemid);
```

After index

```
SELECT W1.lWorkItemId, W2.lWfInstanceId FROM amWfWorkItem W1, amWfInstance W2 WHERE
W1.lDocRecordId = 28836763 AND W1.lActivId = 1417 AND W1.seStatus = 0 AND W2.seStat
us = 0 AND W1.lWfInstanceId=W2.lWfInstanceId;

Execution Plan
----------------------------------------------------------
   0      SELECT STATEMENT Optimizer=CHOOSE (Cost=4 Card=1 Bytes=36)
   1    0   NESTED LOOPS (Cost=4 Card=1 Bytes=36)
   2    1     INDEX (RANGE SCAN) OF 'WORKWFITEM_20051124' (UNIQUE) (Cost=2 Card=1 B
ytes=27)
   3    1     TABLE ACCESS (BY INDEX ROWID) OF 'AMWFINSTANCE' (Cost=2 Card=1 Bytes=
9)
   4    3       INDEX (UNIQUE SCAN) OF 'WFINST_LWFINSTANCE' (UNIQUE) (Cost=1 Card=
1)
Statistics
…
          3  consistent gets (was >12000 before)
          2  physical reads (was >12000 before)
…
          1  SQL*Net roundtrips to/from client
          0  sorts (memory)
          0  sorts (disk)
          0  rows processed
```

This index can boost the query performance by a warp factor of 4000 in terms of memory reads. In terms of CPU usage, it will boost the performance by a warp factor of more than 100.

Implementation in production

- DBA: Drop the index
- AMAPP: Add the same index in AMDBA.
- DBA: Execute a daily rebuilt of the index, and create an index in the rebuild SQL script in case the rebuild fails.

After a week, measures show that the query takes less 0.1 % of the Oracle Server CPU (was 4% before tuning).

Appendix 3: Oracle trace /Tkprof

DBA: Oracle trace start

```
REM Notice the %am% will detect am.* (AM Windows Gui), amsrv (AM Automated Process
Manager), amdba (AM application Designer), amexp (AM Export utility)
REM this parameter should be changed to capture the AM Web Service depending on the
name and way Tomcat is started (can be java, tomcat, …)
REM Machine name can also be set to filter the server to trace in a multiple AM Web
server environment

SQL> connect / as sysdba
SQL> set pagesize 900
SQL> select 'EXECUTE dbms_system.set_ev ('||s.sid||','||s.serial#||', 10046, 4,'
 || '''''' || ')  ;' from v$session s,v$process p  where s.paddr=p.addr and lowe
r(s.machine) like '%' and lower(s.program) like '%am.%';
'EXECUTEDBMS_SYSTEM.SET_EV('||S.SID||','||S.SERIAL#||',10046,4,'||''''''||');'
--------------------------------------------------------------------------------
EXECUTE dbms_system.set_ev (42,1214, 10046, 4,'')  ;
SQL> EXECUTE dbms_system.set_ev (42,1214, 10046, 4,'')  ;
```

DBA: Oracle trace stop

```
SQL> connect / as sysdba
SQL> set pagesize 900
SQL> select 'EXECUTE dbms_system.set_ev ('||s.sid||','||s.serial#||', 10046, 0,'
 || '''''' || ')  ;' from v$session s,v$process p  where s.paddr=p.addr and lowe
r(s.machine) like '%' and lower(s.program) like '%am.%';
'EXECUTEDBMS_SYSTEM.SET_EV('||S.SID||','||S.SERIAL#||',10046,0,'||''''''||');'
--------------------------------------------------------------------------------
EXECUTE dbms_system.set_ev (42,1214, 10046,0,'')  ;
SQL> EXECUTE dbms_system.set_ev (42,1214, 10046, 4,'')  ;
```

DBA: Format Trace TKPROF

```
Start on DOS command
tkprof h:\app\administrator\diag\rdbms\orcl\orcl\trace\orcl_ora_xxxxx.trc g:\suppor
t\UBISorcl_ora_xxxx.txt explain=UBIS/password@orcl sort fchqry
```

AMAPP and DBA: Analyze the result.

Appendix 4: Adblog

Asset Manager Windows client

Setup:

1. Open Asset Manager Windows client.
2. Open the **Help** > **About Asset Manager** > **More**, and note the location of the am.ini file. The heading for the entry is `INI`.
3. Close all instances of Asset Manager on the local machine.
4. Open the associated configuration (ini) file and create the following entry. If there is already an `[Option]` block, do not create a new one, instead, add any missing lines to it.

```
[OPTION]
/ADBLOG=1
/ADBLOG/ADBLOGFILENAME=C:\ASSETLOGS\TESTLOG.TXT
<---- MAKE SURE YOU CHANGE THIS TO A VALID PATH ON YOUR MACHINE
/ADBLOG/ADBLOGFILESIZE=8388610
/ADBLOG/ADBLOGSTARTUP=0
/ADBLOG/EXPLAINPLAN=0
/ADBLOG/LOGAPICALLS=1
/ADBLOG/LOGVIEWER=NOTEPAD
```

In the example above, the log file is created directly on the C drive. Make sure that you set the path and file name in an existing directory on a disk with a lot of remaining free space.

By default, the log file is opened with Notepad. If another program is preferred, this can be set by editing the following line (this example would set UltraEdit as the default editor):

```
/AdbLog/LogViewer=uedit32.exe
```

Restart the Asset Manager process and recreate the error. In all cases, the following keystrokes can be used to produce a more targeted log file:

- F3 : Toggle the trace mode ON or OFF.
- Shift+F3: Reset the log file (makes it empty). It is useful to capture the SQL generated for just a given operation in AM.
- Ctrl+F3: Display (using Notepad by default) the current log file.

Definition of parameters

- AdbLog=: Enable or disable adblogging. 1 means adblogging is active (enabled); 0 means it is inactive (disabled).
- /AdbLog/AdbLogFileName=: Full path and name of the log file. The directory structure must exist; it is not created automatically. If the file does not exist, it is created. If the file already exists, it is opened in the Append mode, it means that all new log information is added to the existing file contents.
- /AdbLog/AdbLogFileSize: Maximum log file size in bytes.
- /AdbLog/AdbLogStartup=: Enable/disable adblogging at startup. ALWAYS leave this set to 0, unless you are instructed by HP OpenView Support to set it to 1. Setting the parameter to 1 causes significant performance problems.
- /AdbLog/ExplainPlan=: Enable/disable SQL Explain Plan logging. ALWAYS leave this set to 0, unless you are instructed by HP OpenView Support to set it to 1.
- /AdbLog/LogViewer=: Editor used to open the log file when you press CTRL+F3 to view it.

How to enable adblog for Asset Manager API

1. Log on to the host where the user will execute the Tomcat instance running the AM web service.
2. Open Asset Manager Windows client.
3. Note the path shown for INI files in **Help** > **About Asset Manager** > **More**.

   For example:

   INI: C:\Documents and settings\Administrator\ApplicationData\HP\AssetManager\conf\am.ini

   (for Windows 2003)

   INI: C:\Users\Administrator\AppData\Roaming\HP\AssetManager\conf

   (for Windows 2008)
4. The aamapi9x.ini (AM version 9.xx) file in the path shown above has the following parameters:

   ```
   [Option]
   /AdbLog/AdbLogFileName=C:\temp\adblog.log
   (Enter any file/folder for this option; the folder should exist)
   /AdbLog/AdbLogFileSize=83886080
   (The file will grow to 80 GB before renamed to adblog.bak)
   /AdbLog/LogApiCalls=1
   /AdbLog/AdbLogStartup=1
   ```
5. If the user is using Asset Manager web service on the same box (for the web client), make sure to

stop it before proceeding.

6. Empty the Conitgui.log file.
7. Enable monitor log in Connect-it GUI.
8. Open the scenario and produce the document(s).

Note: API logging writes extensively, this can usually cause a performance issue. Disable adblog as soon the task is complete. To stop adblog, first stop and close the scenario builder, and then remove the adblog options from the aamapi9x.ini file or set the values of the parameters to 0:

/AdbLog/AdbLogStartup=0

/AdbLog/LogApiCalls=0

How to enable adblog for Asset Manager web server

1. Log on to the host where the user will execute the Tomcat instance for the AM Web Service
2. Open Asset Manager Windows client.
3. Note the path shown for INI files in **Help** > **About Asset Manager** > **More**.

   For example:

   INI: C:\Documents and settings\Administrator\ApplicationData\HP\AssetManager\conf\am.ini

   (for Windows 2003)

   INI: C:\Users\Administrator\AppData\Roaming\HP\AssetManager\conf

   (for Windows 2008)
4. The aamapi9x.ini (AM version 9.xx) file in the path shown above has the following parameters:

```
[Option]
/AdbLog/AdbLogFileName=C:\temp\adblog.log
(Enter any file/folder for this option; the folder should exist)
/AdbLog/AdbLogFileSize=83886080
(The file will grow to 80 GB before renamed to adblog.bak)
/AdbLog/LogApiCalls=0
/AdbLog/AdbLogStartup=0
```

5. If the user is using a Connect-It scenario on the same box, with the same user account as the user account running Tomcat, make sure to stop Connect-It before proceeding.

Note: API logging writes extensively, this can usually cause a performance issue. Disable adblog

as soon the task is complete. To stop adblog, first stop and close the scenario builder, and then remove the adblog options from the aamapi9x.ini file or set the values of the parameters to 0:

/AdbLog/AdbLogStartup=0

/AdbLog/LogApiCalls=0

To stop the adblog in AM web

To start the adblog in AM web (dramatically impact on performance), call the web service as follows:

```
http://<amWSserver>:<WSport>/AssetManagerWebService/adblog?start=-1
```

For example,

```
http://localhost:8080/AssetManagerWebService/adblog?start=-1
```

To stop the adblog in AM web, call the web service as follows:

```
http://<amWSserver>:<WSport>/AssetManagerWebService/adblog?stop=-1
```

For example,

```
http://localhost:8080/AssetManagerWebService/adblog?stop=-1
```

Send Documentation Feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on Asset Manager Web Monitoring for Windows x64, Oracle, MS SQL Server and Tomcat (Asset Manager 9.50)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to ovdoc-ITSM@hp.com.

We appreciate your feedback!