Tuning HP-UX and SAS Enterprise BI Server 9.1.3 on HP Integrity servers – technical brief

Introduction

All benchmarks and tuning parameters depend on the application. Each application has characteristics of its own and thus tuning will be specific to that application, the version of the software, the type of users, the operating system, and the system's hardware.

This document provides details about the HP-UX 11i v2 and SAS Enterprise BI performance parameters for the multiple SAS tiers running on HP Integrity servers. Due to the variety of implementation options and customer usage possibilities the parameters and their values should be viewed as a reference for tuning your environment.

SAS Midtier

These tunings reference using BEA WebLogic Server version 8.1 SP3 as the Midtier server installed on an Integrity server running HP-UX 11i v2.

For system configuration settings recommended by BEA, see the HP-UX supported configurations section of their website[1].

Kernel

Kernel parameter changes

More information for the kernel tunable parameters can be found at hp.com[2].

```
STRMSGSZ                        65535
max_thread_proc  Number of Users X 5
maxdsiz                    0xfffff000
streampipes                        64
vx_ninode                        8000
```

Discussion

Parameter	Description
`STRMSGSZ`	Limits the number of bytes of message data that can be inserted by putmsg() or write() in the data portion of any streams message on the system. If the tunable is set to zero, there is no limit on how many bytes can be placed in the data segment of the message.
`max_thread_proc`	Defines the maximum number of concurrent threads allowed per process.
`maxdsiz`	Defines the maximum size (in bytes) of the data segment for any user process. This tunable defines the maximum size of the static data storage segment for 32-bit and 64-bit processes. The data storage segment contains fixed data storage such as globals, arrays, static variables, local variables in main(), strings, and space allocated using `sbrk()` and `malloc()`. In addition, any files memory mapped as private and shared library per-invocation data also resides in the data segment.
`streampipes`	This tunable determines the type of pipe that is created by the pipe() system call. If set to the default value of zero, all pipes created by pipe() are normal HP-UX file-system pipes. If the value is non-zero, pipe() creates STREAMS-based pipes, and STREAMS modules can be pushed onto the resulting stream.

[1] HP-UX documentation for WLS 8.1 SP3 is found at
http://edocs.bea.com/platform/suppconfigs/configs81/hpux11_itanium/81sp3.html
[2] HP-UX kernel parameter (section 5) manuals can be found at http://docs.hp.com/en/B2355-60105/

vx_ninode	The VxFS file system caches the inodes in an inode table. The kernel tunable vx_ninode determines the number of inodes in the inode table to help VxFS in caching. The vx_ninode static tunable is initialized when a system is booted. Thus the changes in the vx_ninode table will take effect only during the next system reboot.

Network tuning

Network tuning changes

```
ndd -set /dev/tcp tcp_conn_request_max 15000
ndd –set /dev/tcp tcp_xmit_hiwater_def 1048576
ndd –set /dev/tcp tcp_rexmit_interval_initial 4000
ndd –set /dev/tcp tcp_keepalive_interval 900000
ndd –set /dev/tcp tcp_ip_abort_interval 60000
ndd -set /dev/tcp tcp_smallest_anon_port 32768
ndd -set /dev/tcp tcp_naglim_def 1
```

Discussion

Parameter	Description
tcp_conn_request_max	The maximum number of pending connection requests for any listening end point. This tunable is also known as the maximum depth of the "listen queue". The actual maximum for any given TCP endpoint in the LISTEN state will be the MINIMUM of the tcp_conn_request_max and the value the application passed-in to the listen() socket call. For this parameter to take effect, it must be set BEFORE an application makes its call to listen(). So, if you use ndd to set this value after the application has started, it will have no effect unless you can get the application to recreate its LISTEN endpoint(s).
tcp_xmit_hiwater_def	For every TCP connection a buffer is allocated. The application writes into this buffer and TCP is responsible for sending it to the distant host. Sometimes it happens that the other host is not able to receive further data, so TCP can not send more data out on the interface. In this case the allocated buffer fills up and at one point we reach a limit where we must stop the application from sending more data to the buffer. This higher limit is called the high-water mark.
tcp_rexmit_interval_initial	This sets the initial value for the round trip time, from which the retransmit timeout is computed. The round trip time is fundamental for TCP's timeout and retransmission which is experienced on a given connection. This value is given in milliseconds
tcp_keepalive_interval	The parameter tcp_keepalive_interval determines the amount of time that TCP waits for an idle connection with no unacknowledged data before sending keepalive packets.
tcp_ip_abort_interval	This sets the TCP packet retransmission timers for established connections and will trigger to close the whole connection.
tcp_smallest_anon_port	Smallest anonymous port number to use.
tcp_naglim_def	Initial value for the Nagle limit.

To review BEA recommendations for HP-UX, go to the URL:
http://edocs.bea.com/platform/suppconfigs/configs81/hpux11_itanium/81sp3.html

Additional settings recommended by BEA

The following command helps maximize usage of the network bandwidth by setting the maximum transmission unit on the network card to 1500 bytes. This is the maximum packet size unless you are using jumbo frames.

```
/sbin/ifconfig lo mtu 1500
```

HP-UX process modifications

Increase amount of process memory and enable shared library segment merging

Execute the following commands:

cd /opt/java1.4/bin/IA64N/

chart +as mpas +mergeseg enable java

Set real time scheduling priority

1. As root edit the /etc/privgroup file. If it does not exist, create it. Add the following entry to the file:

   bea_group_name RTPRIO

   where *bea_group_name* is the group name containing the user that starts the BEA WebLogic Server instances.

2. In the shell, execute the following command: setprivgrp –f /etc/privgroup
3. You must restart the BEA WebLogic Server instances for this change to take affect.

HP JRE SE 1.4.2

The following parameters were set for HP JVM on the HP-UX 11i v2 system. For more detailed information, see the HP Java™ website[3].

```
-Xmx1280m
–Xms1280m
-XX:+ForceMmapReserved
-XX:PermSize=96m
–XX:MaxPermSize=96m
-XX:SurvivorRatio=13
-XX:-UseOnStackReplacement
-XX:SchedulerPriorityRange=SCHED_RTPRIO
-Xss192k
-XX:NewSize=282m
-XX:MaxNewSize=282m
-XX:+DisableExplicitGC
-Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000
-Djava.awt.headless=true
```

[3] http://www.hp.com/products1/unix/java/infolibrary/prog_guide/hotspot.html

Discussion

Parameter	Description
Xmx1280m Xms1280m	Java heap size information to minimize the garbage collection. The entries made after setting AggressiveHeap override the changes made to it.
XX:+ForceMmapReserved	Tells the JVM to reserve the swap space for all large memory regions used by the JVM. Use this option to reserve the space for all large memory regions used by the JVM. This includes the Java Heap, which is an `mmap`'ed space. Starting with HP-UX 11.11, the default behavior is that the memory regions be reserved lazily. Most large server-side applications will use all of the space, so improved performance can be obtained by reserving the space at program initialization. If this option is not used, 4K pages will be allocated for the `mmap`'ed regions. This can put pressure on the chip's translation buffer when it needs to translate the virtual page's address to its corresponding physical address.
XX:PermSize=96m XX:MaxPermSize=96m	Specifies the initial size and maximum, in bytes, of the Permanent Space memory allocation pool. This value must be a multiple of 1024 greater than 1MB. Append the letter k or K to indicate kilobytes, or m or M to indicate megabytes.

XX:SurvivorRatio=13	Ratio of eden/survivor space size. The default is 8, meaning that eden is 8 times bigger than from and to, each. Here are some examples:
	`Xmn / (SurvivorRatio + 2) = size of from and to, each`
	`( Xmn / (SurvivorRatio + 2) ) * SurvivorRatio = eden size`

XX:-UseOnStackReplacement	On stack replacement enables the interpreter to go into compiled code while it is executing the same instance of the method call.
XX:SchedulerPriorityRange=SCHED_RTPRIO	Use real time thread scheduler priority. This option can be used to both select the scheduling policy and map the Java thread priorities, 1 (low) through 10 (high), to the underlying HP-UX thread priorities. See the Java website[4] for details.
Xss192k	Sets the Java stack size. This increased value keeps the JVM from crashing during deep recursions, such as those observed in the ACC.
XX:NewSize=282m XX:MaxNewSize=282m	Sets the Java new generation heap size. The "new generation" is the first generation in HotSpot's generational garbage collector.
XX:+DisableExplicitGC	Ignore requests for garbage collection from within the Java code. Leaves it to the JVM to make all decisions about garbage collection.

[4] http://www.hp.com/products1/unix/java/infolibrary/prog_guide/hotspot.html?jumpid=reg_R1002_USEN#-XX:SchedulerPriorityRange=SCHED

Dsun.rmi.dgc.client.gcInterval=3600000	When it is necessary to ensure that DGC clean calls for unreachable remote references are delivered in a timely fashion, the value of this property represents the maximum interval (in milliseconds) that the RMI runtime will allow between garbage collections of the local heap.
Dsun.rmi.dgc.server.gcInterval=3600000	When it is necessary to ensure that unreachable remote objects are unexported and garbage collected in a timely fashion, the value of this property represents the maximum interval (in milliseconds) that the RMI runtime will allow between garbage collections of the local heap.
Djava.awt.headless=true	This allows server applications to access core graphics objects, such as fonts, colors and images.

SAS execute queue

Create an execute queue for each BEA WebLogic Instance with the following settings:

ExecuteQueue Name="sas.wrs.default"

ThreadCount="(# web users per instance) / 2.5"

ThreadsIncrease="0"

WebLogic instance timeout settings

There are many timeouts available for your web server instance. You will need to monitor your average user response times to determine the best settings to use in your environment. Ideally you want to set the highest settings listed below to be greater than the average users response times for transactions.

A good rule of thumb in setting this is to ensure that timeouts decrease in duration in the following order:

(Average User Web Transaction Time)

HTTP Message Timeout

Idle Connection Timeout

Complete Message Timeout

If using a front-end server set the following additional timeouts in decreasing order:

HP-Apache Timeouts (KeepAliveSecs & Timeout)

WebLogic Plug-In for Apache Timeout (KeepAliveSecs)

WebLogic Front-End Server Instance Timeouts (KeepAliveSecs & PostTimeoutSecs)

In our testing we had a front-end server and set our timeouts as:

Apache Timeouts: Timeout 360, KeepAliveTimeout 360

WebLogic HTTP Message Timeout: 320

WebLogic Idle Connection Timeout: 240

WebLogic Plug-In to Apache Timeouts: KeepAliveSecs 240

WebLogic Complete Message Timeout: 180

WebLogic Front-End Server Timeouts: KeepAliveSecs=90 & PostTimeoutSecs=90

XML parser

SAS works better with a different XML parser than the default. To use the new parser you should following these steps:

Copy the files xercesImpl.jar and xmlParserAPIs.jar (located in `SASWebReportStudio/WEB-INF/lib` directory) to your $JAVA_HOME/jre/lib/ext directory. Grant these files rights for the sas and bea users.

Register the XML parser in BEA:

In BEA WebLogic console click on Services > XML, right click and select "Configure a new XML Parser"

Configure the new XML Registry:

Name XercesXMLParser
Document Builder Factory: org.apache.xerces.jaxp.DocumentBuilderFactoryImpl
SAX Parser Factory: org.apache.xerces.jaxp.SAXParserFactoryImpl
Transformer Factory: org.apache.xalan.processor.TransformerFactoryImpl
When To Cache: cache-at-initialization
Target and Deploy to all web instances.

Tell each web instance to use the new XML parser:

XMLEntityCache="XMLCacheMBean_web" XMLRegistry="XercesXMLParser"

For more details on XML parsers for SAS:
http://e-docs.bea.com/wls/docs81/xml/xml_admin.html

Xythos WebFile Server

Kernel

These tunings reference using Xythos WebFile Server version 4.0.48 installed on an Integrity server running HP-UX 11i v2.

Kernel parameter changes

More information for the kernel tunable parameters can be found at hp.com[5].

```
STRMSGSZ         65535
max_thread_proc   1000
streampipes         64
vx_ninode         8000
```

Discussion

Parameter	Description
STRMSGSZ	Limits the number of bytes of message data that can be inserted by putmsg() or write() in the data portion of any streams message on the system. If the tunable is set to zero, there is no limit on how many bytes can be placed in the data segment of the message.
max_thread_proc	Defines the maximum number of concurrent threads allowed per process.
streampipes	This tunable determines the type of pipe that is created by the pipe() system call. If set to the default value of zero, all pipes created by pipe() are normal HP-UX file-system pipes. If the value is non-zero, pipe() creates STREAMS-based pipes, and STREAMS modules can be pushed onto the resulting stream.
vx_ninode	The VxFS file system caches the inodes in an inode table. The kernel tunable vx_ninode determines the number of inodes in the inode table to help VxFS in caching. The vx_ninode static tunable is initialized when a system is booted. Thus the changes in the vx_ninode table will take effect only during the next system reboot.

Java heap size

Increase Java heap size to minimize garbage collection.

Edit $XYTHOS_HOME/appserver-4.0.48/bin/setclasspath.sh

JAVA_OPTS="-server -Xms768m -Xmx768m"

Xythos thread settings

You will want to manage the number of threads that are available based on the amount of traffic your servers receive and on the number of CPUs that your server has at its disposal. Tomcat Connectors listen for incoming requests. A pool of threads is available to handle these requests. The number of threads available can be managed with the `minProcessors` and `maxProcessers` parameters in the `server.xml` configuration file.

[5] HP-UX kernel parameter (section 5) manuals can be found at http://docs.hp.com/en/B2355-60105/

Edit $XYTHOS_HOME/appserver-4.0.48/conf/server.xml

<!-- Define a non-SSL HTTP/1.1 Connector -->

<Connector className="org.apache.catalina.connector.http.HttpConnector"

port="8300" minProcessors="**100**" maxProcessors="**200**"

SAS Metadata Server

These tunings reference using SAS Metadata Server version 9.1.3 SP3 installed on an Integrity server running HP-UX 11i v2.

Kernel parameter changes

More information for the kernel tunable parameters can be found at hp.com[6].

```
STRMSGSZ      65535
maxfiles_lim   8192
streampipes      64
vx_ninode      8000
timeslice        20
```

Discussion

Parameter	Description
STRMSGSZ	Limits the number of bytes of message data that can be inserted by putmsg() or write() in the data portion of any streams message on the system. If the tunable is set to zero, there is no limit on how many bytes can be placed in the data segment of the message.
maxfiles_lim	Specifies the system hard limit for the number of file descriptors that a process is allowed to have for open files at any given time.
vx_ninode	The VxFS file system caches the inodes in an inode table. The kernel tunable vx_ninode determines the number of inodes in the inode table to help VxFS in caching. The vx_ninode static tunable is initialized when a system is booted. Thus the changes in the vx_ninode table will take effect only during the next system reboot.
streampipes	This tunable determines the type of pipe that is created by the pipe() system call. If set to the default value of zero, all pipes created by pipe() are normal HP-UX file-system pipes. If the value is non-zero, pipe() creates STREAMS-based pipes, and STREAMS modules can be pushed onto the resulting stream.
timeslice	Defines the scheduling time interval that a thread may execute on a processor before the kernel scheduler will context switch out the thread for other same priority threads to run. When a thread starts executing on a processor, the thread is set up to run for the number of ticks in the timeslice tunable. On every clock interrupt that a thread is found executing, the time quantum balance for the thread is decremented, and when the balance reaches zero, the thread is context switched out. The timeslice value controls one method of user preemption that the operating system implements.

[6] HP-UX kernel parameter (section 5) manuals can be found at http://docs.hp.com/en/B2355-60105/

Threading model

Modify the threading model SAS uses.

Create a file $SAS_HOME/libpthread.properties which contains the line:

```
force_scope_process 1
```

Edit the profile file for the user that starts the SAS metadata server process to include:

```
export PTHREAD_PROPERTY_FILE=$SAS_HOME/libpthread.properties
export PTHREAD_TUNE=1
```

SAS process and thread tuning

Edit the SAS process and thread pool to optimize performance by ensuring that the number of available threads is compatible with the number of threads that can be accepted concurrently by the server.

- The MAXACTIVETHREADS server configuration option specifies the maximum number of threads that are allowed to run concurrently on the metadata server. The number of processors available on the server host determines the number of queries that can be active at any point in time.

- The THREADSMIN (TMIN) and THREADSMAX (TMAX) objectserverparms options, which are specified in the metadata server start command, specify the number of threads that are maintained in the server's thread pool. The thread pool defines the total number of threads that are available to a server, regardless of the number of user requests that are received or accepted.

Edit MetadataServer.sh and in startup command add

```
instantiate tmin=(#cpu * 4 + 2) tmax=(same as tmin)
```

Edit omaconfig.xml

```
MAXACTIVETHREADS="(number of cpus in system)"
TMIN="(#cpu * 4 + 2) "
TMAX="(#cpu * 4 + 2) "
```

SAS OLAP Server

Kernel

These tunings reference using SAS OLAP Server version 9.1.3 SP3 installed on an Integrity server running HP-UX 11i v2.

Kernel parameter changes

More information for the kernel tunable parameters can be found at hp.com[7].

```
STRMSGSZ         65535
max_thread_proc   1000
streampipes         64
vx_ninode         8000
```

Discussion

Parameter	Description
STRMSGSZ	Limits the number of bytes of message data that can be inserted by putmsg() or write() in the data portion of any streams message on the system. If the tunable is set to zero, there is no limit on how many bytes can be placed in the data segment of the message.
max_thread_proc	Defines the maximum number of concurrent threads allowed per process.
streampipes	This tunable determines the type of pipe that is created by the pipe() system call. If set to the default value of zero, all pipes created by pipe() are normal HP-UX file-system pipes. If the value is non-zero, pipe() creates STREAMS-based pipes, and STREAMS modules can be pushed onto the resulting stream.
vx_ninode	The VxFS file system caches the inodes in an inode table. The kernel tunable vx_ninode determines the number of inodes in the inode table to help VxFS in caching. The vx_ninode static tunable is initialized when a system is booted. Thus the changes in the vx_ninode table will take effect only during the next system reboot.

[7] HP-UX kernel parameter (section 5) manuals can be found at http://docs.hp.com/en/B2355-60105/

Increase process priority

Grant the SAS process realtime scheduling priority

Edit /etc/privgroup

sasgroup RTPRIO RTSCHED

restart OLAP server using the following command:

rtsched -s SCHED_NOAGE -p 178 $OLAP_HOME/OLAPServer/OLAPServer.sh start

Increase process memory

Increase the SAS process memory amount

Edit OLAPserver.sh, in the startup command add

–memsize (actual system memory size)

For more information

HP's HP-UX URL: www.hp.com/go/hpux11i

HP and SAS documentation URL: www.hp.com/go/sas

SAS support URL: support.sas.com

BEA's documentation URL: http://edocs.bea.com

4/2006