GlassFish Monitoring with JMX, Jconsole, Glassbox and AMX

Sun Community Days
Sydney, Australia
06 March 2008

Chris Fleischmann
Sun Microsystems, Inc.

Agenda

- GlassFish Monitoring OOTB
- JConsole and JMX Monitoring
- Open Source project Glassbox
- AMX and GlassFish Management Rules

Agenda

- **GlassFish Monitoring OOTB**
- JConsole and JMX Monitoring
- Open Source project Glassbox
- AMX and GlassFish Management Rules

GlassFish Monitoring OOTB

Features include:

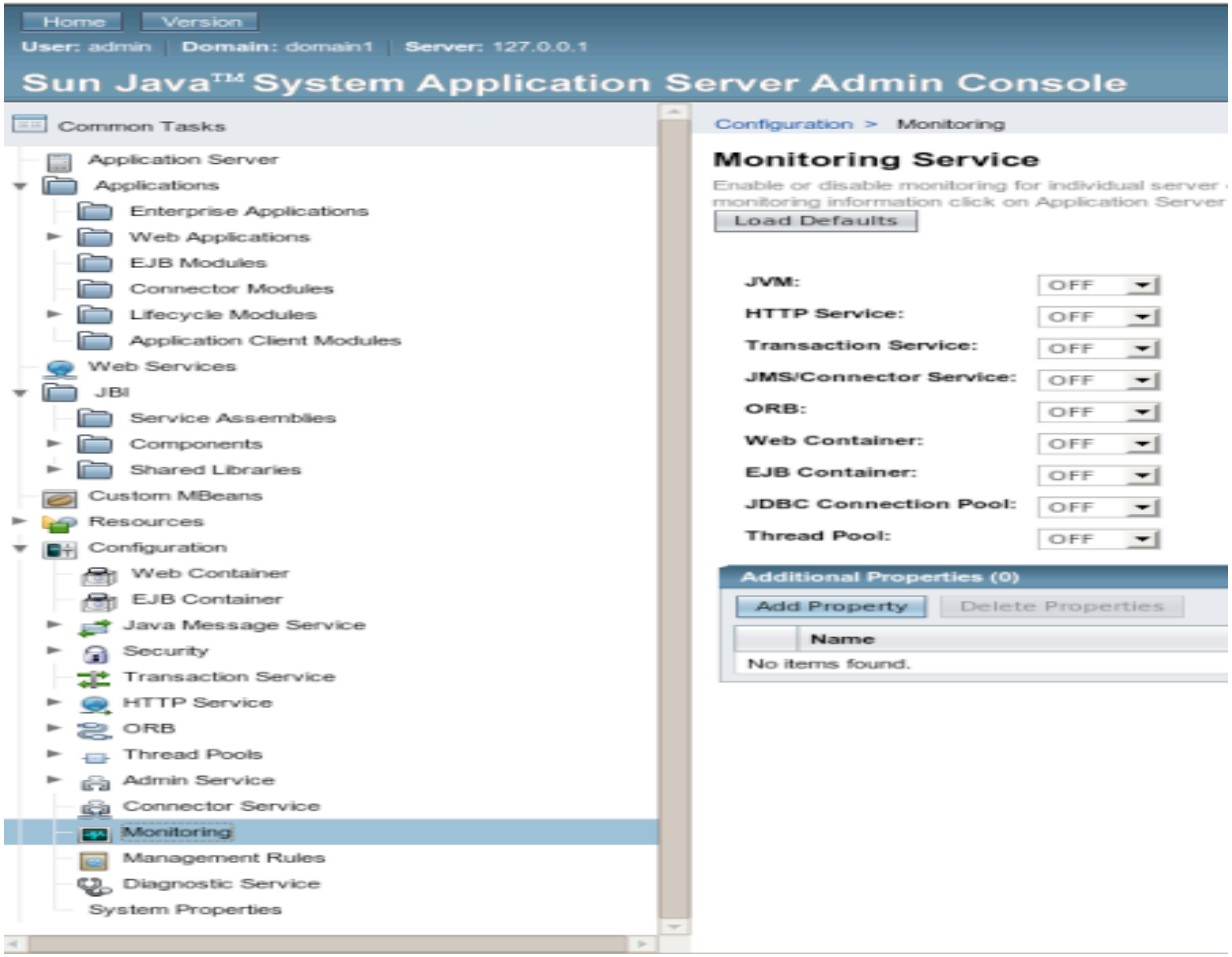
- CallFlow Monitoring
- Run-Time Monitoring
- Application Monitoring
- Resource Monitoring
- Transaction Monitoring

GlassFish Monitoring OOTB

Configuration

```
asadmin start-callflow-monitoring [–filtertype type=value:type=value] instancename
```

OR



Monitoring Service

Enable or disable monitoring for individual server
monitoring information click on Application Server

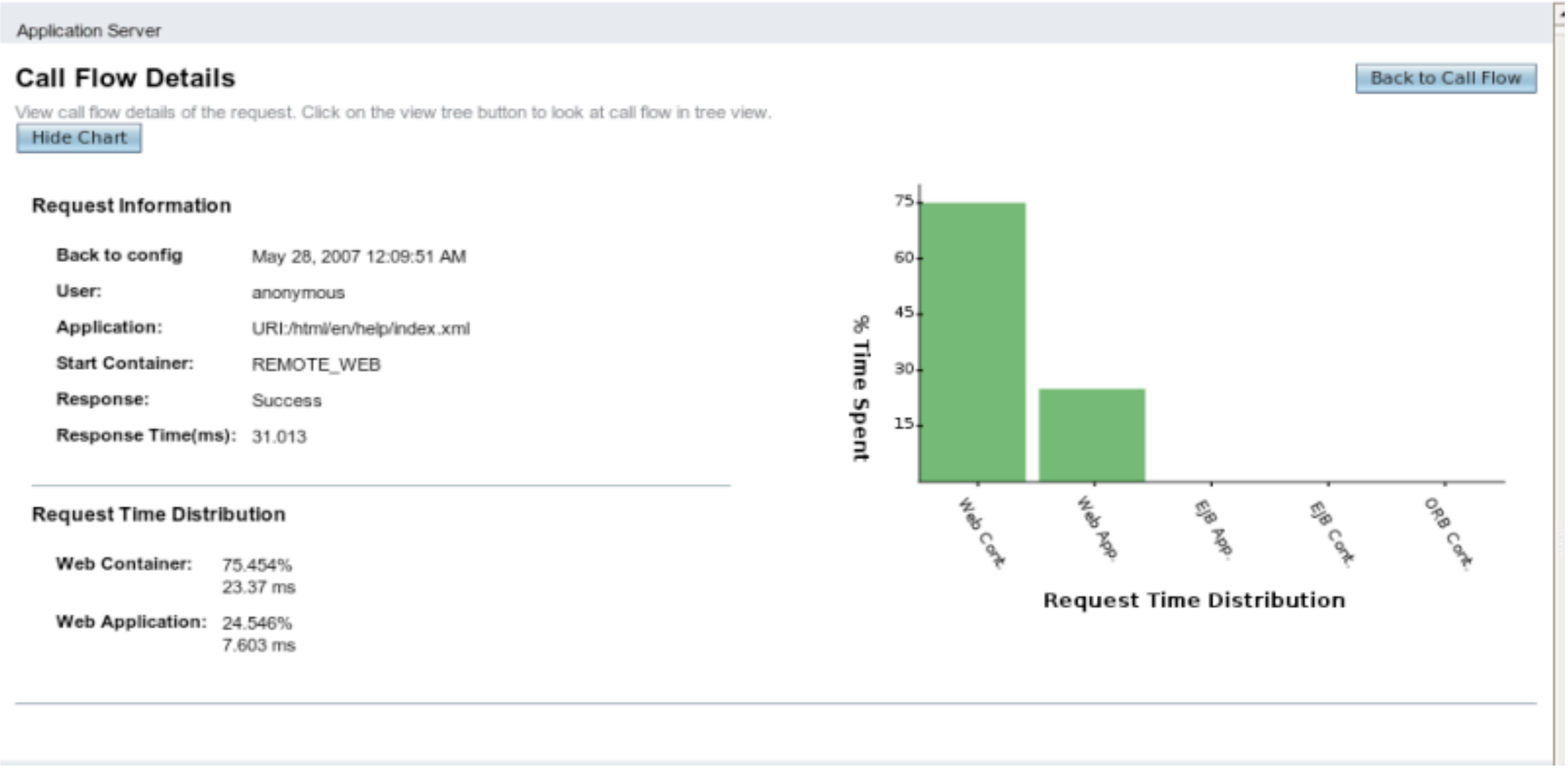
JVM:	OFF
HTTP Service:	OFF
Transaction Service:	OFF
JMS/Connector Service:	OFF
ORB:	OFF
Web Container:	OFF
EJB Container:	OFF
JDBC Connection Pool:	OFF
Thread Pool:	OFF

Additional Properties (0)

Name
No items found.

GlassFish Monitoring OOTB

Call Flow Monitoring



Application Server

Call Flow Details

Back to Call Flow

View call flow details of the request. Click on the view tree button to look at call flow in tree view.

Hide Chart

Request Information

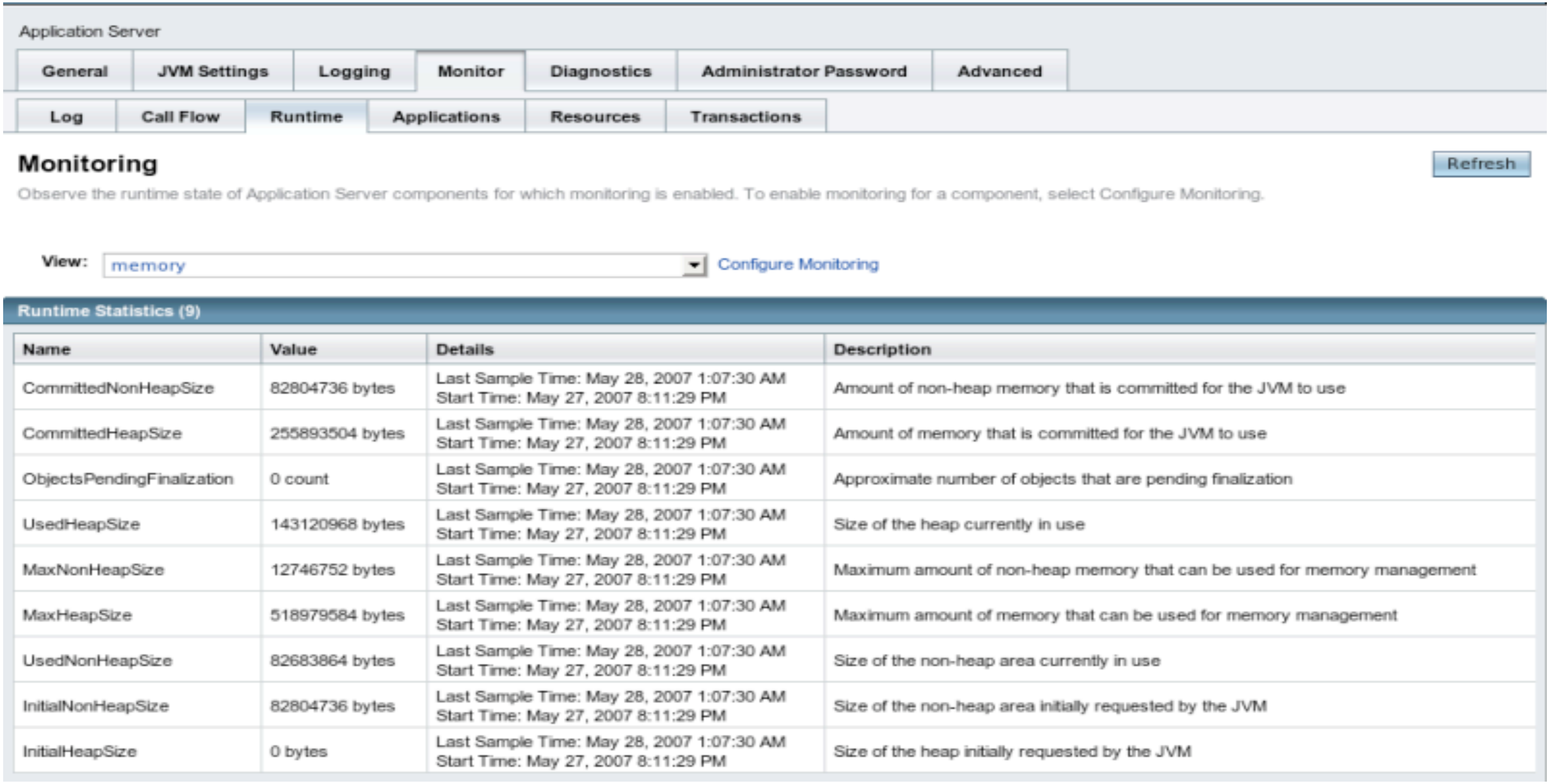
Back to config	May 28, 2007 12:09:51 AM
User:	anonymous
Application:	URI:/html/en/help/index.xml
Start Container:	REMOTE_WEB
Response:	Success
Response Time(ms):	31.013

Request Time Distribution

Web Container:	75.454% 23.37 ms
Web Application:	24.546% 7.603 ms

GlassFish Monitoring OOTB

Runtime Monitoring



Monitoring

Observe the runtime state of Application Server components for which monitoring is enabled. To enable monitoring for a component, select Configure Monitoring.

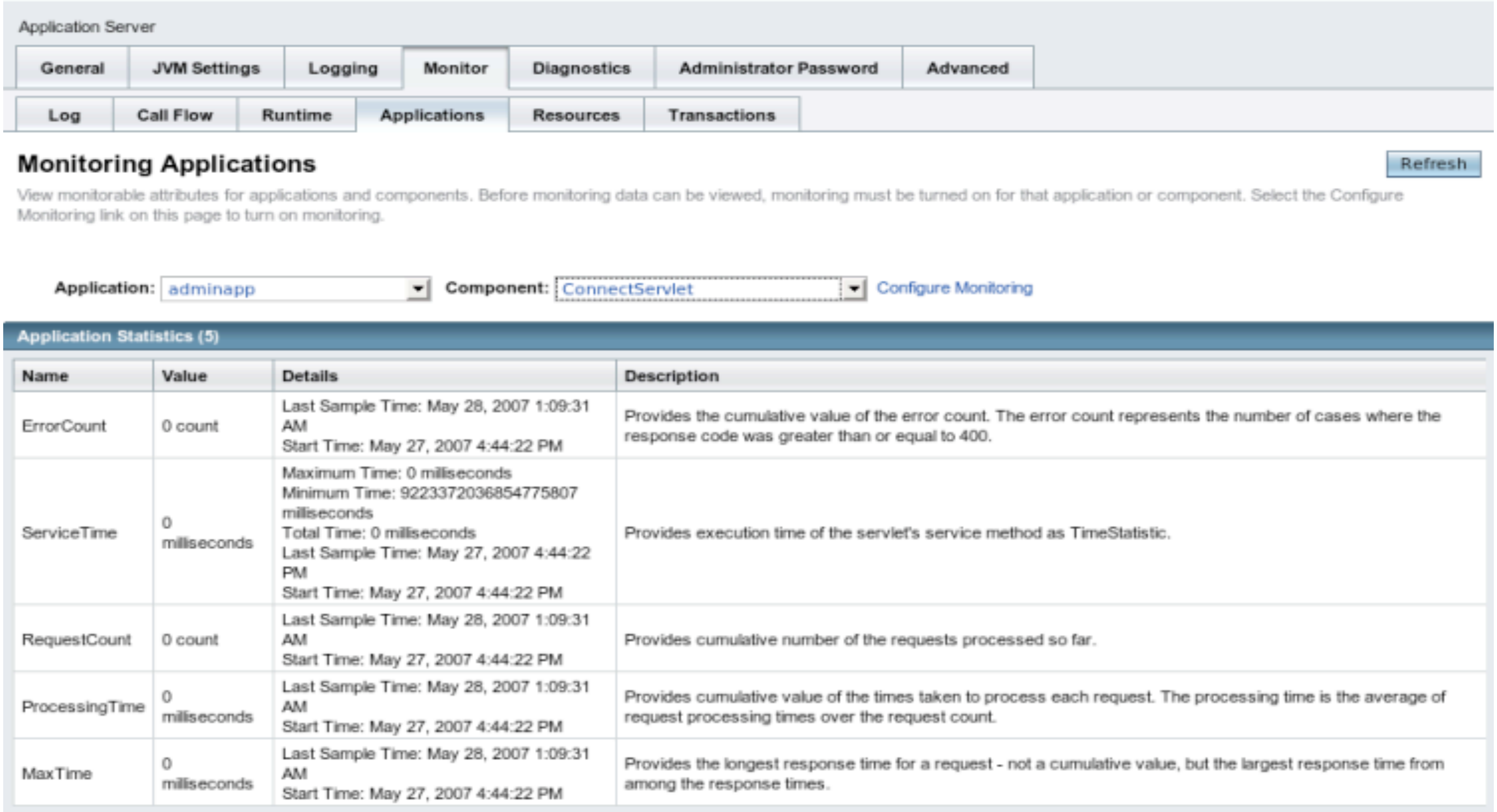
View: memory

Runtime Statistics (9)

Name	Value	Details	Description
CommittedNonHeapSize	82804736 bytes	Last Sample Time: May 28, 2007 1:07:30 AM Start Time: May 27, 2007 8:11:29 PM	Amount of non-heap memory that is committed for the JVM to use
CommittedHeapSize	255893504 bytes	Last Sample Time: May 28, 2007 1:07:30 AM Start Time: May 27, 2007 8:11:29 PM	Amount of memory that is committed for the JVM to use
ObjectsPendingFinalization	0 count	Last Sample Time: May 28, 2007 1:07:30 AM Start Time: May 27, 2007 8:11:29 PM	Approximate number of objects that are pending finalization
UsedHeapSize	143120968 bytes	Last Sample Time: May 28, 2007 1:07:30 AM Start Time: May 27, 2007 8:11:29 PM	Size of the heap currently in use
MaxNonHeapSize	12746752 bytes	Last Sample Time: May 28, 2007 1:07:30 AM Start Time: May 27, 2007 8:11:29 PM	Maximum amount of non-heap memory that can be used for memory management
MaxHeapSize	518979584 bytes	Last Sample Time: May 28, 2007 1:07:30 AM Start Time: May 27, 2007 8:11:29 PM	Maximum amount of memory that can be used for memory management
UsedNonHeapSize	82683864 bytes	Last Sample Time: May 28, 2007 1:07:30 AM Start Time: May 27, 2007 8:11:29 PM	Size of the non-heap area currently in use
InitialNonHeapSize	82804736 bytes	Last Sample Time: May 28, 2007 1:07:30 AM Start Time: May 27, 2007 8:11:29 PM	Size of the non-heap area initially requested by the JVM
InitialHeapSize	0 bytes	Last Sample Time: May 28, 2007 1:07:30 AM Start Time: May 27, 2007 8:11:29 PM	Size of the heap initially requested by the JVM

GlassFish Monitoring OOTB

Application Monitoring



Monitoring Applications

View monitorable attributes for applications and components. Before monitoring data can be viewed, monitoring must be turned on for that application or component. Select the Configure Monitoring link on this page to turn on monitoring.

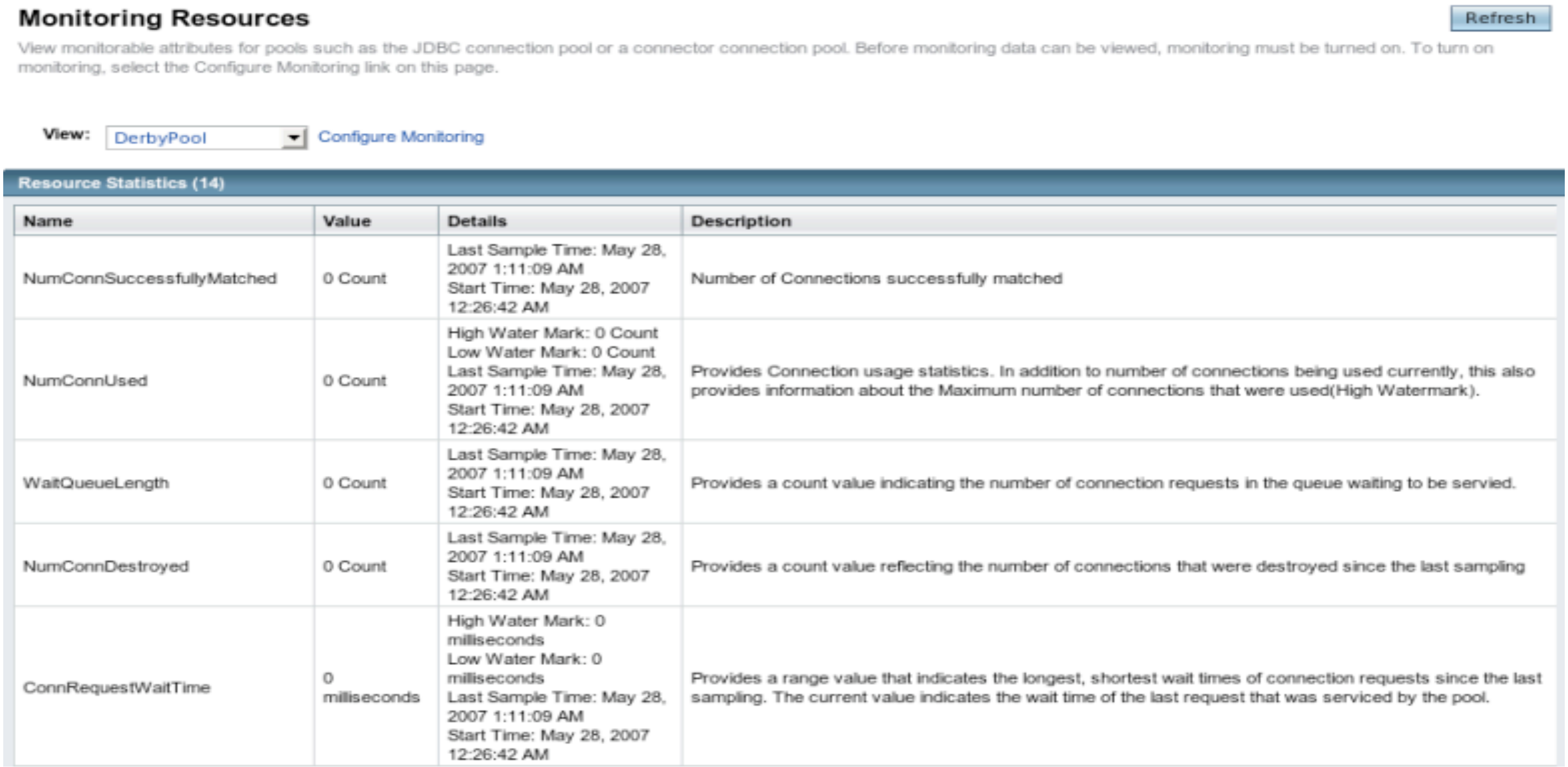
Application: adminapp **Component:** ConnectServlet

Application Statistics (5)

Name	Value	Details	Description
ErrorCount	0 count	Last Sample Time: May 28, 2007 1:09:31 AM Start Time: May 27, 2007 4:44:22 PM	Provides the cumulative value of the error count. The error count represents the number of cases where the response code was greater than or equal to 400.
ServiceTime	0 milliseconds	Maximum Time: 0 milliseconds Minimum Time: 9223372036854775807 milliseconds Total Time: 0 milliseconds Last Sample Time: May 27, 2007 4:44:22 PM Start Time: May 27, 2007 4:44:22 PM	Provides execution time of the servlet's service method as TimeStatistic.
RequestCount	0 count	Last Sample Time: May 28, 2007 1:09:31 AM Start Time: May 27, 2007 4:44:22 PM	Provides cumulative number of the requests processed so far.
ProcessingTime	0 milliseconds	Last Sample Time: May 28, 2007 1:09:31 AM Start Time: May 27, 2007 4:44:22 PM	Provides cumulative value of the times taken to process each request. The processing time is the average of request processing times over the request count.
MaxTime	0 milliseconds	Last Sample Time: May 28, 2007 1:09:31 AM Start Time: May 27, 2007 4:44:22 PM	Provides the longest response time for a request - not a cumulative value, but the largest response time from among the response times.

GlassFish Monitoring OOTB

Resource Monitoring



Monitoring Resources

View monitorable attributes for pools such as the JDBC connection pool or a connector connection pool. Before monitoring data can be viewed, monitoring must be turned on. To turn on monitoring, select the Configure Monitoring link on this page.

View: DerbyPool

Resource Statistics (14)

Name	Value	Details	Description
NumConnSuccessfullyMatched	0 Count	Last Sample Time: May 28, 2007 1:11:09 AM Start Time: May 28, 2007 12:26:42 AM	Number of Connections successfully matched
NumConnUsed	0 Count	High Water Mark: 0 Count Low Water Mark: 0 Count Last Sample Time: May 28, 2007 1:11:09 AM Start Time: May 28, 2007 12:26:42 AM	Provides Connection usage statistics. In addition to number of connections being used currently, this also provides information about the Maximum number of connections that were used(High Watermark).
WaitQueueLength	0 Count	Last Sample Time: May 28, 2007 1:11:09 AM Start Time: May 28, 2007 12:26:42 AM	Provides a count value indicating the number of connection requests in the queue waiting to be servied.
NumConnDestroyed	0 Count	Last Sample Time: May 28, 2007 1:11:09 AM Start Time: May 28, 2007 12:26:42 AM	Provides a count value reflecting the number of connections that were destroyed since the last sampling
ConnRequestWaitTime	0 milliseconds	High Water Mark: 0 milliseconds Low Water Mark: 0 milliseconds Last Sample Time: May 28, 2007 1:11:09 AM Start Time: May 28, 2007 12:26:42 AM	Provides a range value that indicates the longest, shortest wait times of connection requests since the last sampling. The current value indicates the wait time of the last request that was serviced by the pool.

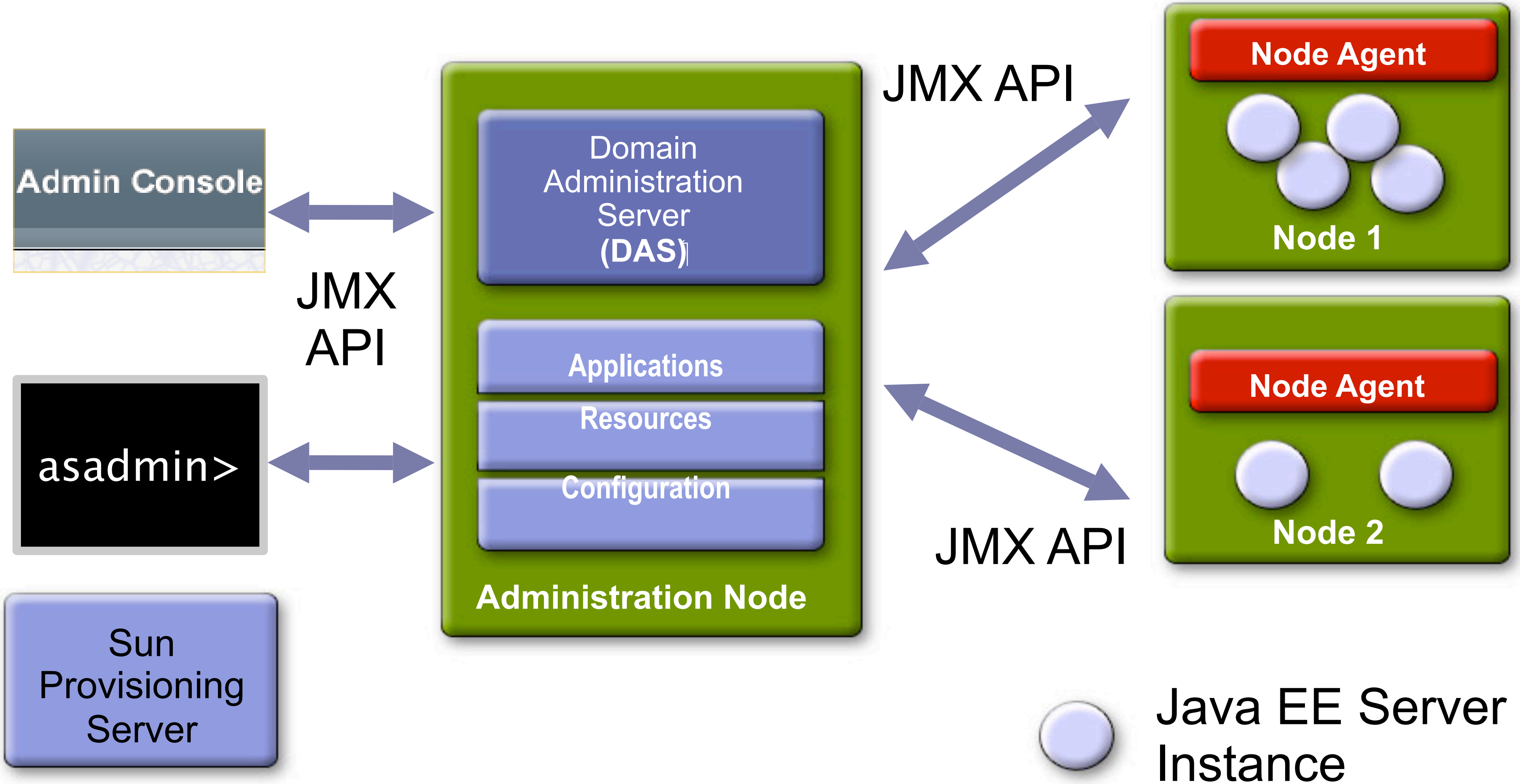
GlassFish Monitoring OOTB

Quick Demo

Agenda

- GlassFish Monitoring OOTB
- **JConsole and JMX Monitoring**
- Open Source project Glassbox
- AMX and GlassFish Management Rules

JMX Support in GlassFish v2

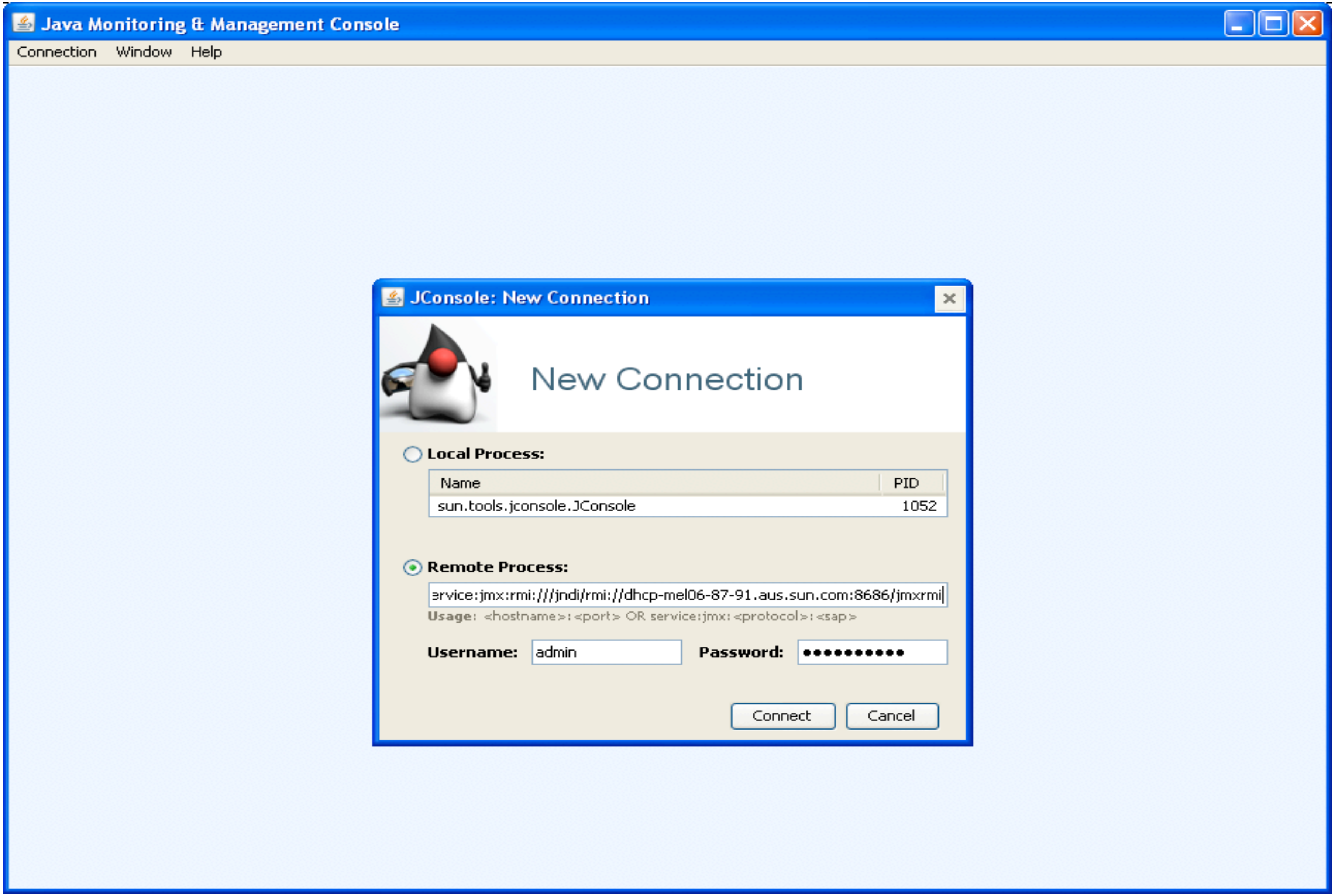


JMX = Java Management Extensions

JConsole and JMX Monitoring

- JConsole Features Include:
  - Overview: Displays overview information about the Java VM and monitored values.
  - Memory: Displays information about memory use.
  - Threads: Displays information about thread use.
  - Classes: Displays information about class loading.
  - VM: Displays information about the Java VM.
  - MBeans: Displays information about MBeans.

JConsole and JMX Monitoring



○ **Local Process:**

Name	PID
sun.tools.jconsole.JConsole	1052

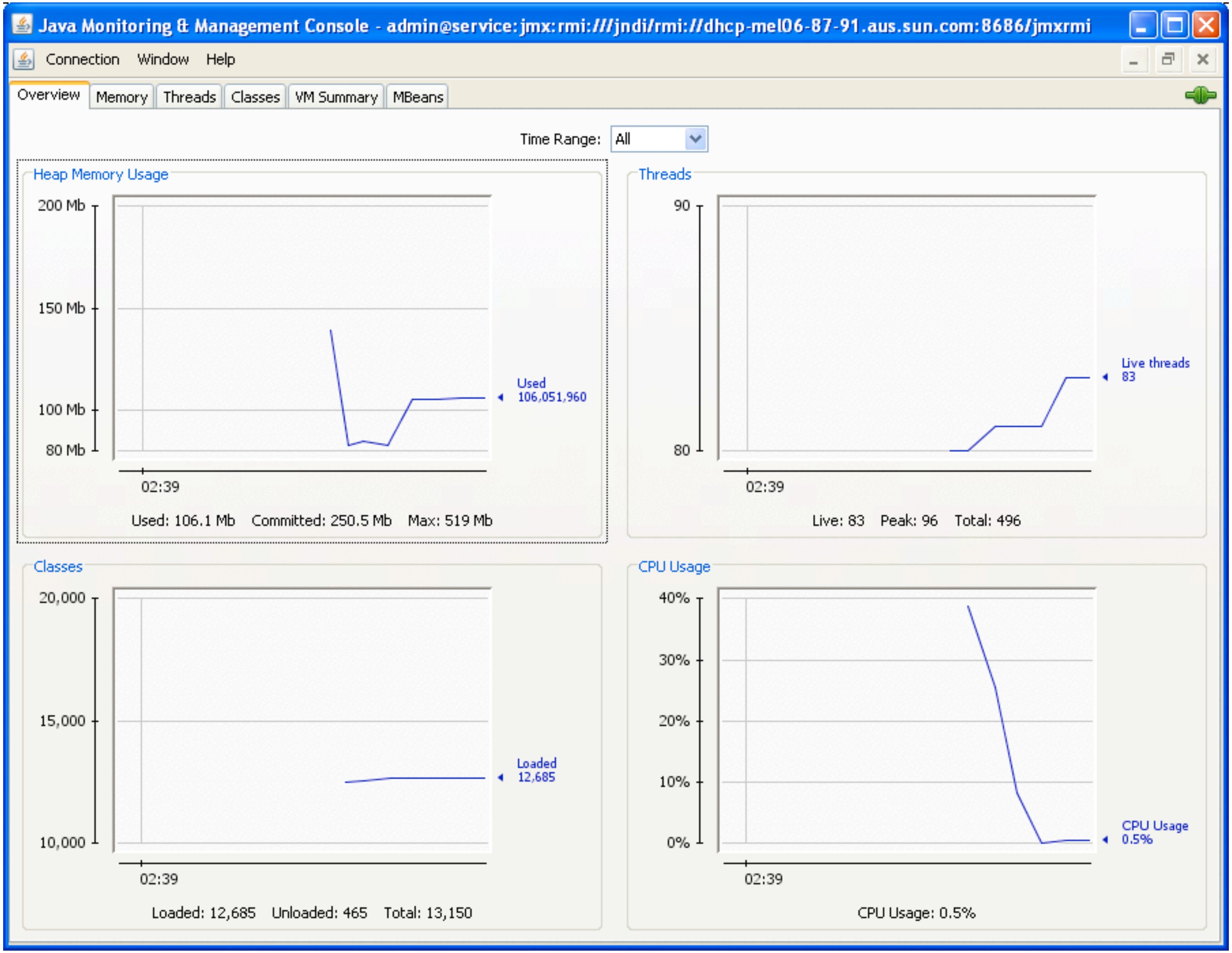
● **Remote Process:**

ervice:jmx:rmi:///jndi/rmi://dhcp-mel06-87-91.aus.sun.com:8686/jmxrmi

Usage: <hostname>:<port> OR service:jmx:<protocol>:<sap>

Username: admin **Password:** ●●●●●●●●●●●

JConsole and JMX Monitoring



Java Monitoring & Management Console - admin@service:jmx:rmi:///jndi/rmi://dhcp-mel06-87-91.aus.sun.com:8686/jmxrmi

Time Range: All

Used: 106.1 Mb Committed: 250.5 Mb Max: 519 Mb

Live: 83 Peak: 96 Total: 496

Loaded: 12,685 Unloaded: 465 Total: 13,150

CPU Usage: 0.5%

JConsole and JMX Monitoring

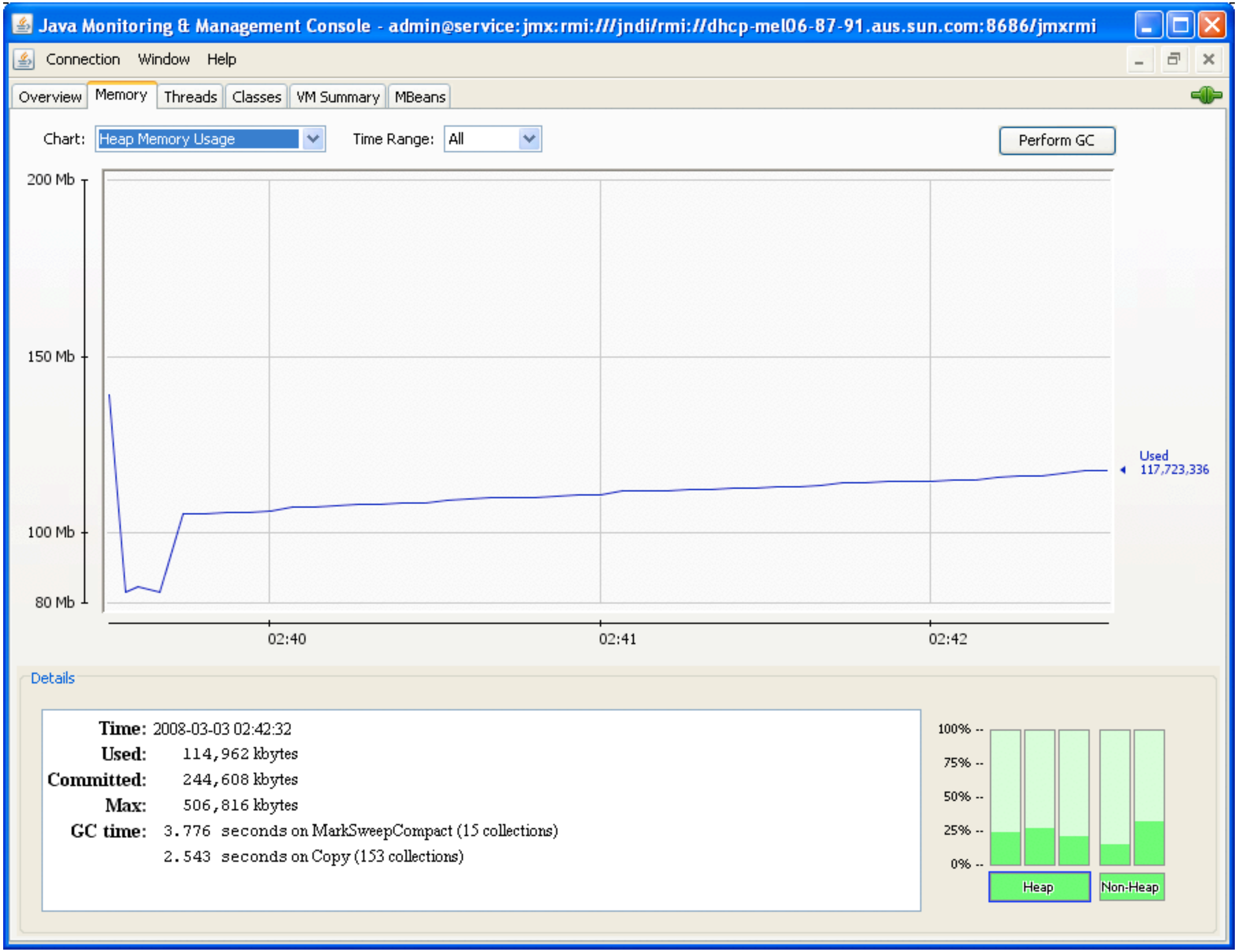
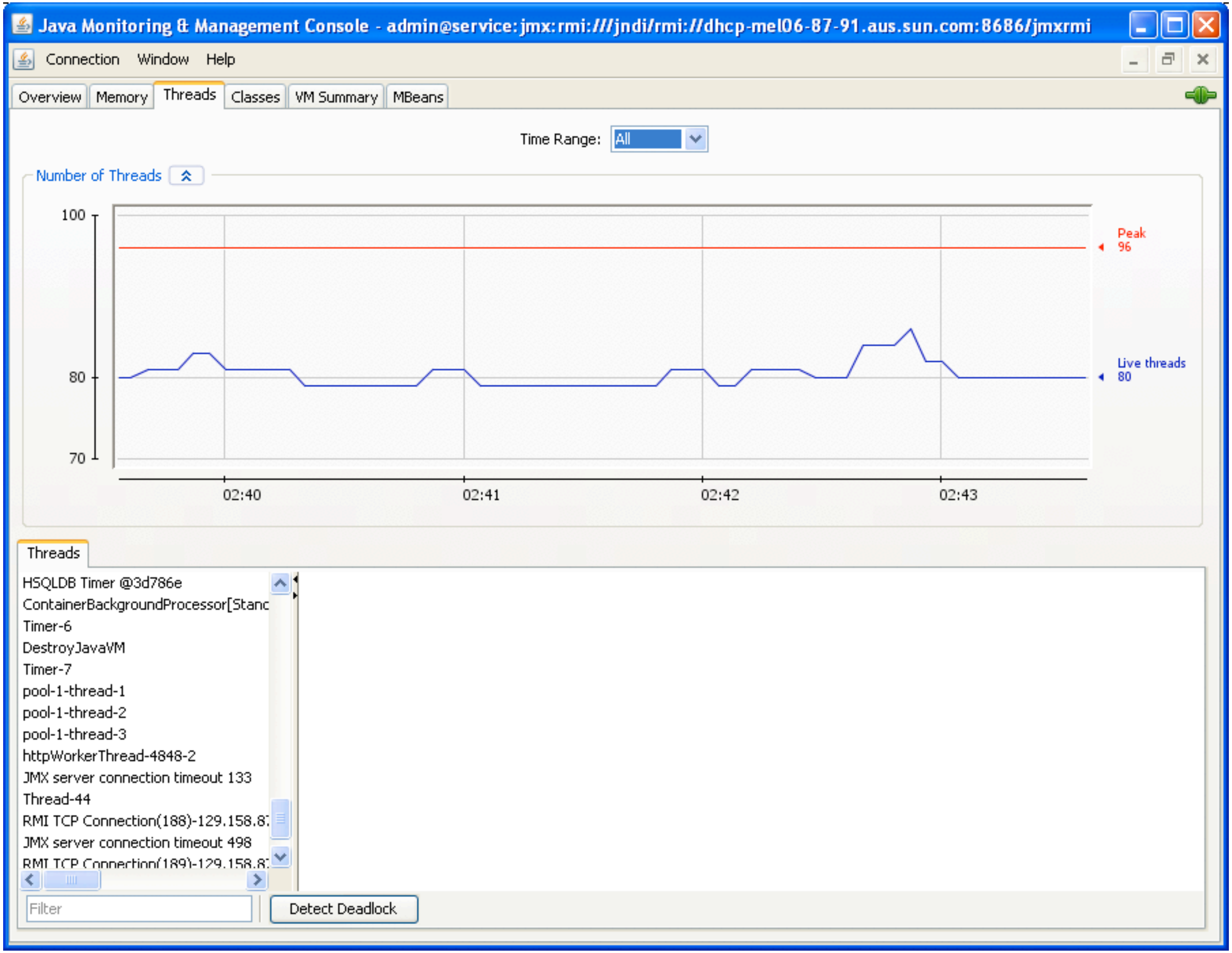


Chart: Heap Memory Usage

Time Range: All

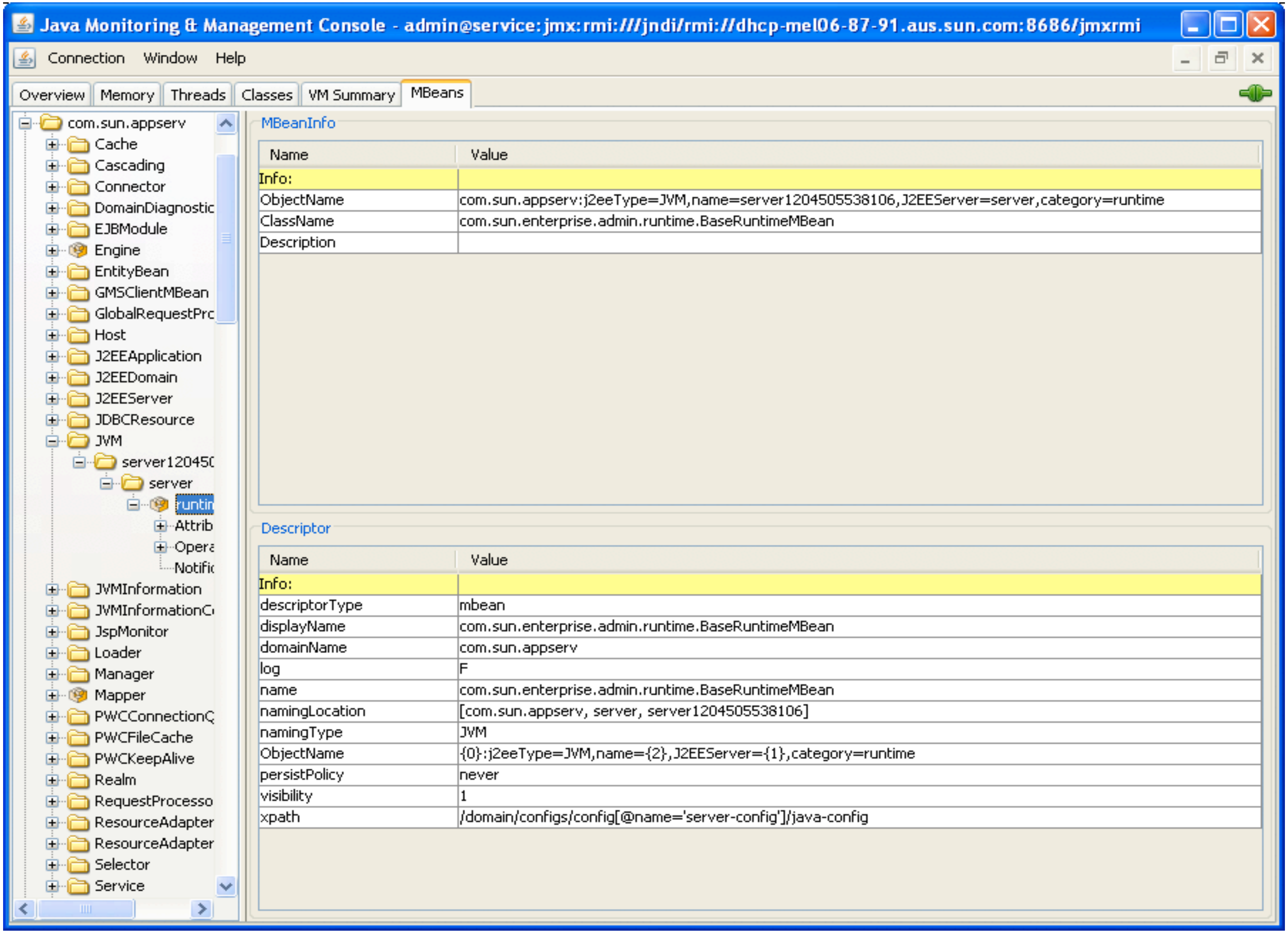
```
      Time: 2008-03-03 02:42:32
      Used:     114,962 kbytes
 Committed:     244,608 kbytes
       Max:     506,816 kbytes
   GC time:   3.776  seconds on MarkSweepCompact (15 collections)
              2.543  seconds on Copy (153 collections)
```

JConsole and JMX Monitoring



```
HSQLDB Timer @3d786e
ContainerBackgroundProcessor[Stanc
Timer-6
DestroyJavaVM
Timer-7
pool-1-thread-1
pool-1-thread-2
pool-1-thread-3
httpWorkerThread-4848-2
JMX server connection timeout 133
Thread-44
RMI TCP Connection(188)-129.158.87
JMX server connection timeout 498
RMI TCP Connection(189)-129.158.87
```

JConsole and JMX Monitoring



MBeanInfo

Name	Value
Info:	
ObjectName	com.sun.appserv:j2eeType=JVM,name=server1204505538106,J2EEServer=server,category=runtime
ClassName	com.sun.enterprise.admin.runtime.BaseRuntimeMBean
Description	

Descriptor

Name	Value
Info:	
descriptorType	mbean
displayName	com.sun.enterprise.admin.runtime.BaseRuntimeMBean
domainName	com.sun.appserv
log	F
name	com.sun.enterprise.admin.runtime.BaseRuntimeMBean
namingLocation	[com.sun.appserv, server, server1204505538106]
namingType	JVM
ObjectName	{0}:j2eeType=JVM,name={2},J2EEServer={1},category=runtime
persistPolicy	never
visibility	1
xpath	/domain/configs/config[@name='server-config']/java-config

JConsole and JMX Monitoring

Quick Demo

Agenda

- GlassFish Monitoring OOTB
- JConsole and JMX Monitoring
- **Open Source project Glassbox**
- AMX and GlassFish Management Rules

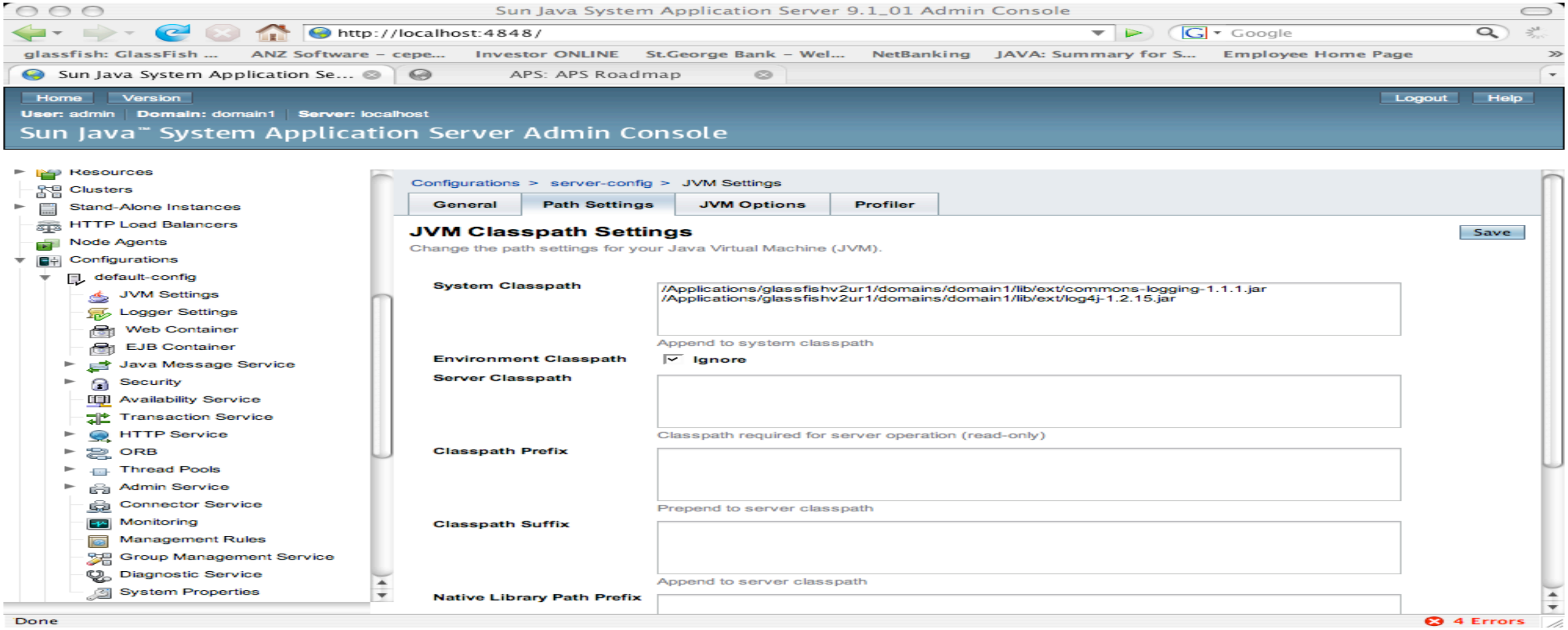
Open Source Project Glassbox

- Glassbox Inspector 2.0 - an open source non-intrusive and light-weight Java application monitoring package with:
  - Monitoring metrics collection based on Aspect-Oriented Programming (AOP) approach using AspectJ 5
  - Collected monitoring metrics exposed via the Java Management Extensions (JMX) Managed Beans (MBeans)

Open Source Project Glassbox

Configuration

- Both commons-logging_1.1.x and log4j_1.2.x must be added to the system classpath for GlassFish
- The Glassbox JVM parameters must be added to the launch configuration



JVM Classpath Settings

Change the path settings for your Java Virtual Machine (JVM).

System Classpath:

```
/Applications/glassfishv2ur1/domains/domain1/lib/ext/commons-logging-1.1.1.jar
/Applications/glassfishv2ur1/domains/domain1/lib/ext/log4j-1.2.15.jar
```

Append to system classpath

Environment Classpath: ☑ Ignore

Server Classpath:

Classpath required for server operation (read-only)

Classpath Prefix:

Prepend to server classpath

Classpath Suffix:

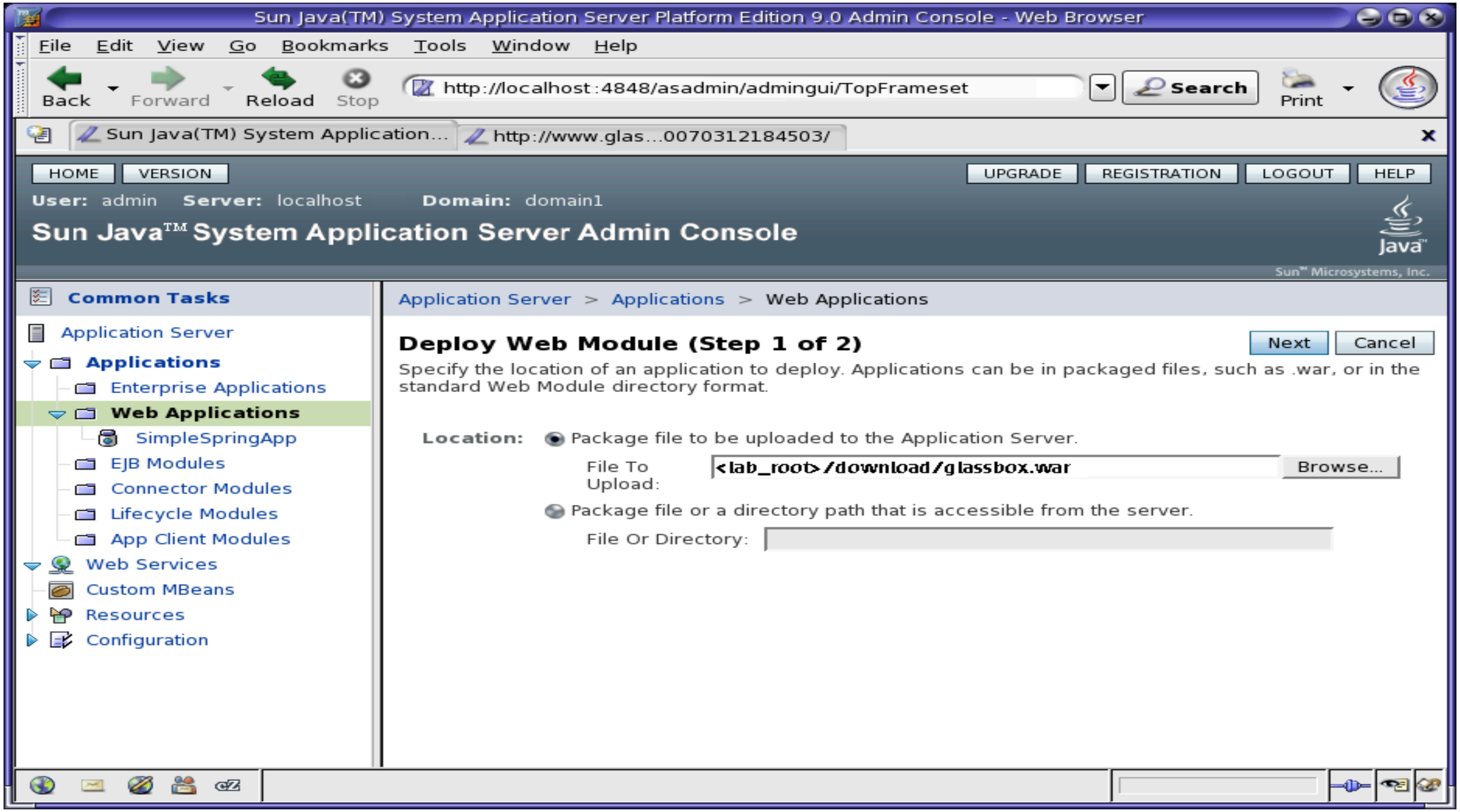
Append to server classpath

Native Library Path Prefix:

Open Source Project Glassbox

Configuration (Cont.)

- Deploy glassbox.war



Sun Java(TM) System Application Server Platform Edition 9.0 Admin Console - Web Browser
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://localhost:4848/asadmin/admingui/TopFrameset
Search
Print
Sun Java(TM) System Application...
http://www.glas...0070312184503/
HOME VERSION
UPGRADE REGISTRATION LOGOUT HELP
Sun Java™ System Application Server Admin Console
Java
Sun™ Microsystems, Inc.
Common Tasks
Application Server
Applications
Enterprise Applications
Web Applications
SimpleSpringApp
EJB Modules
Connector Modules
Lifecycle Modules
App Client Modules
Web Services
Custom MBeans
Resources
Configuration
Application Server > Applications > Web Applications
Next
Cancel
Browse...

User: admin Server: localhost Domain: domain1

Deploy Web Module (Step 1 of 2)

Specify the location of an application to deploy. Applications can be in packaged files, such as .war, or in the standard Web Module directory format.

Location: ● Package file to be uploaded to the Application Server.

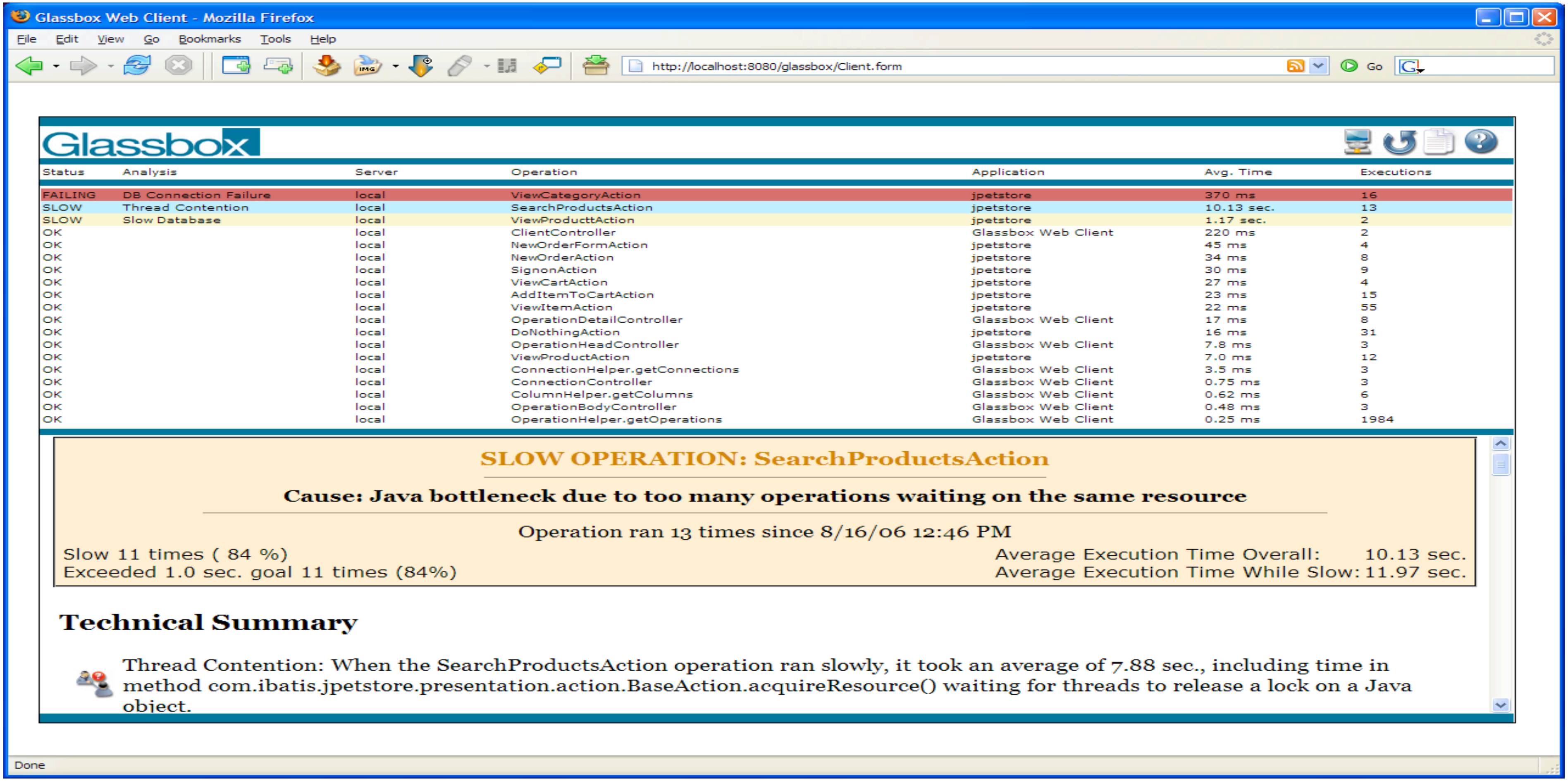
File To Upload: <lab_root>/download/glassbox.war

○ Package file or a directory path that is accessible from the server.

File Or Directory:

Open Source Project Glassbox

Sample Screenshot



Glassbox Web Client - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://localhost:8080/glassbox/Client.form
Go
Glassbox
Done

Status	Analysis	Server	Operation	Application	Avg. Time	Executions
FAILING	DB Connection Failure	local	ViewCategoryAction	jpetstore	370 ms	16
SLOW	Thread Contention	local	SearchProductsAction	jpetstore	10.13 sec.	13
SLOW	Slow Database	local	ViewProducttAction	jpetstore	1.17 sec.	2
OK		local	ClientController	Glassbox Web Client	220 ms	2
OK		local	NewOrderFormAction	jpetstore	45 ms	4
OK		local	NewOrderAction	jpetstore	34 ms	8
OK		local	SignonAction	jpetstore	30 ms	9
OK		local	ViewCartAction	jpetstore	27 ms	4
OK		local	AddItemToCartAction	jpetstore	23 ms	15
OK		local	ViewItemAction	jpetstore	22 ms	55
OK		local	OperationDetailController	Glassbox Web Client	17 ms	8
OK		local	DoNothingAction	jpetstore	16 ms	31
OK		local	OperationHeadController	Glassbox Web Client	7.8 ms	3
OK		local	ViewProductAction	jpetstore	7.0 ms	12
OK		local	ConnectionHelper.getConnections	Glassbox Web Client	3.5 ms	3
OK		local	ConnectionController	Glassbox Web Client	0.75 ms	3
OK		local	ColumnHelper.getColumns	Glassbox Web Client	0.62 ms	6
OK		local	OperationBodyController	Glassbox Web Client	0.48 ms	3
OK		local	OperationHelper.getOperations	Glassbox Web Client	0.25 ms	1984

SLOW OPERATION: SearchProductsAction

Cause: Java bottleneck due to too many operations waiting on the same resource

Operation ran 13 times since 8/16/06 12:46 PM

Slow 11 times (84 %)
Exceeded 1.0 sec. goal 11 times (84%)

Average Execution Time Overall: 10.13 sec.
Average Execution Time While Slow: 11.97 sec.

Technical Summary

Thread Contention: When the SearchProductsAction operation ran slowly, it took an average of 7.88 sec., including time in method com.ibatis.jpetstore.presentation.action.BaseAction.acquireResource() waiting for threads to release a lock on a Java object.

Open Source Project Glassbox

Quick Demo

Agenda

- GlassFish Monitoring OOTB
- JConsole and JMX Monitoring
- Open Source project Glassbox
- **AMX and GlassFish Management Rules**

AMX and GlassFish Management Rules

- AppServer Management EXtensions. The terms “AppServer” and “Management” should be self-explanatory. The term “Extensions” refers to the addition of a considerable number of interfaces beyond the basic structure defined by the J2EE Management specification (JSR 77).
- AMX is composed of several hundred Java interfaces (eg “public interface DomainRoot {...}). Each of this interfaces represents the runtime capabilities of a server-side JMX MBean residing in the AppServer.
- A good introductory article to read:

  http://today.java.net/pub/a/today/2008/01/15/dynamic-load-balancing-in-glassfish.html

AMX and GlassFish Management Rules

Objectives

- Manage complexity by self-configuring
- Improve ease-of-use by automating mundane management tasks
- Improve performance by self-tuning in unpredictable run-time conditions
- Improve availability by preventing and recovering (self-healing) from failures
- Improve security by taking self-protective actions when security threats are detected

AMX and GlassFish Management Rules

Features

- Using AMX we can:
  1. Change application server configurations—create resources, delete resources, enable or disable, etc.
  2. Manage servers, node agents, clusters, etc.
  3. Receive notifications for almost any event happening in the application server and react accordingly.
  4. Monitor the state of many objects that are hosted inside the application server. These include EJBs, web applications, enterprise applications, connection pools, etc.

AMX and GlassFish Management Rules

Features

- A management rule is a set of:
  - Event: An event uses the JMX notification mechanism to trigger actions. Events can range from an ***MBean*** attribute change to specific log messages.
  - Action: Actions are associated with events and are triggered when related events happen. Actions can be ***MBeans*** that implement the ***NotificationListener*** interface.

AMX and GlassFish Management Rules

Features

- Important types of events are as follows:
  - Monitor events: These type of events trigger an action based on an ***MBean*** attribute change.
  - Notification events: ***MBeans*** can implement ***NotificationBroadcaster*** in order to send notifications to all listeners that registered their interest on its event notifications.
  - System events: This is a set of predefined events that come from the internal infrastructure of GlassFish application server. These events include: lifecycle, log, timer, trace, and cluster events.

AMX and GlassFish Management Rules

How-To's

1) Connect to the Domain Admin Server (DAS)
   - The port used by AMX is an RMI port (default 8686), not the GUI (http/s) port. Connecting to the GUI port will not work.
   - You will need to know whether TLS (SSL) is enabled or not in order to connect.
2) Use method, ***public static com.sun.appserv.management.client.AppserverConnectionSource connect***
3) From that point on, call ***getDomainRoot()*** to get an instance of ***DomainRoot*** and call methods from there.

AMX and GlassFish Management Rules

A Simple (but useful) example

A snippet of code to retrieve the cluster “Cluster-1” from the default domain then retrieve an instance, “instance-01” from the cluster and set the Load Balancing Weight to 25.

```
---

AppserverConnectionSource ASConnection =  Connect.connectNoTLS("127.0.0.1", 8686, "admin", "adminadmin");

DomainRoot dRoot = ASConnection.getDomainRoot();

Map<String, ClusterConfig> clusters =  dRoot.getContaineeMap(XTypes.CLUSTER_CONFIG);

ClusterConfig clusterConf= clusters.get("Cluster-1");

Map <String, ClusteredServerConfig< servers = clusterConf.getClusteredServerConfigMap();

ClusteredServerConfig instance1 = servers.get("instance-01");


instance1.setLBWeight("25");
```

AMX and GlassFish Management Rules

A Simple (but useful) example

- Define your management rule in a configuration file.
- There are three cluster events fired; when a cluster is started, stopped, or has failed.
- Remember each management rule has an event that will trigger an action. An action is a ***MBean*** that implements the ***NotificationListener*** interface.
- When the management rule triggers the ***handleNotification*** action method will be called

AMX and GlassFish Management Rules

A Simple (but useful) example

Once you have written your rule file and your MBean class you can then deploy/register the MBean with GlassFish:

```
create-mbean --host <HOSTNAME> --port <PORT> --user <ADMIN USER< --name <BEAN NAME> --attributes ConfigurationFilePath=<CONFIGURATION FILE.XML> <CLASS FILENAME>
```

For Example:

```
create-mbean --host localhost --port 4848 --user admin --name ClusterInstanceWeightsManager --attributes ConfigurationFilePath=c\:/config.xml samples.glassfish.management.clustermanager.ClusterWeightManager
```

AMX and GlassFish Management Rules

A Simple (but useful) example

Lastly, define the management rule, which will automatically send the cluster start notification to the new MBean when the cluster starts.

```
create-management-rule --eventtype cluster --host <HOST> --port <PORT<  --user <ADMIN
    USER> --ruleenabled=true  --action <BEAN NAME> --event loglevel INFO --
    recordevent=true  --eventproperties name=<NAME OF RULE>
```

Resources

- http://weblogs.java.net/blog/kalali/archive/2007/05/glassfish_versi_1.html
- http://java.sun.com/developer/technicalArticles/J2SE/jconsole.html
- http://www.glassbox.com
- https://glassfish.dev.java.net/javaee5/amx/

Questions ?

Chris Fleischmann
Chris.Fleischmann@sun.com

SUN COMMUNITY DAYS 2007-2008

A Worldwide Developer Conference