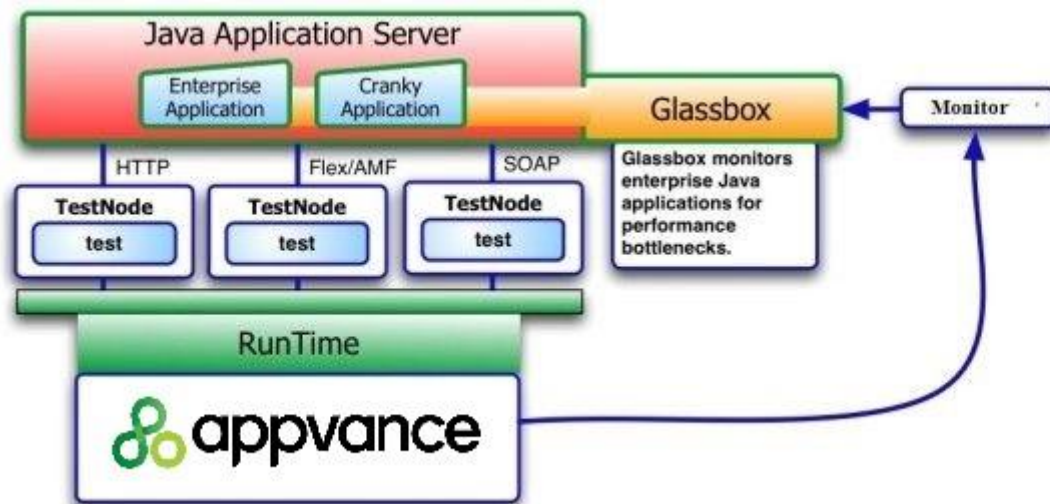


Architecture

AIQ is a distributed test and monitor environment. As a load and performance test operates Glassbox observes performance metrics of a Java enterprise application, including memory use, thread contention, database connections, and object operation. Appvance Monitor (Appvance Monitor) communicates with Glassbox over Java management (JMX) protocols. AIQ gathers Glassbox data and correlates Glassbox findings into AIQ reports.



For example, Appvance operates a load and performance test on one or more TestNodes. A TestNode is a Web service that receives and operates a test. The TestNode is in your QA lab, data center, cloud computing environment, or both. Glassbox installs on a Java application server and uses Aspect-Oriented Technology to observe enterprise applications. The AIQ monitor system (Appvance Monitor) communicates over JMX protocols to Glassbox as the test operates. The Appvance Enterprise Results Analysis Engine correlates the Glassbox observations to the load test results for root cause analysis and mitigation.

Glassbox provides a Troubleshooter to view the live monitored data and performance alerts for your application. The following shows the Glassbox Troubleshooter interface.

Status	Analysis	Server	Operation	Application	Avg. Time	Executions
SLOW	Slow Java Method	local	DerbyTimedQueryServlet	/cranky	30.26 sec.	1
SLOW	CPU Hog	local	DefaultServlet	cranky	3.11 sec.	1
SLOW	Slow Dispatch/Helper	local	DispatcherServlet	glassbox	1.03 sec.	1
OK		local	ClientController	glassbox	190 ms	1
OK		local	ClientController	Glassbox Web Client	95 ms	1
OK		local	ConnectionHelper.getConnections	glassbox	61 ms	1
OK		local	ColumnHelper.getColumns	glassbox	48 ms	2
OK		local	OperationHelper.updateCheck	glassbox	42 ms	1
OK		local	DWRServlet	glassbox	7.0 ms	12
OK		local	OperationHelper.updateCheck	Glassbox Web Client	6.3 ms	1
OK		local	DefaultServlet	Glassbox Web Client	6.2 ms	9
OK		local	OperationHelper.getOperations	Glassbox Web Client	6.1 ms	704
OK		local	ConnectionHelper.getConnections	Glassbox Web Client	5.8 ms	1
OK		local	DefaultServlet	glassbox	5.6 ms	5
OK		local	DWRServlet	Glassbox Web Client	4.2 ms	9
OK		local	ColumnHelper.getColumns	Glassbox Web Client	3.4 ms	2

PRODUCTS **COMPANY** **CONTACT US** **DOWNLOAD** **COMMUNITY**

Please try the latest version of the open source Glassbox 2.0 [here](#).

- Simple Web Based Client
- Merges the Troubleshooter and Inspector functionality
- Drop it in and diagnose apps with a click

Pinpoints problems clearly
Adapts automatically to apps
Works without code changes

[Updates](#) [Documents](#) [Forums](#) [Contact Us](#)

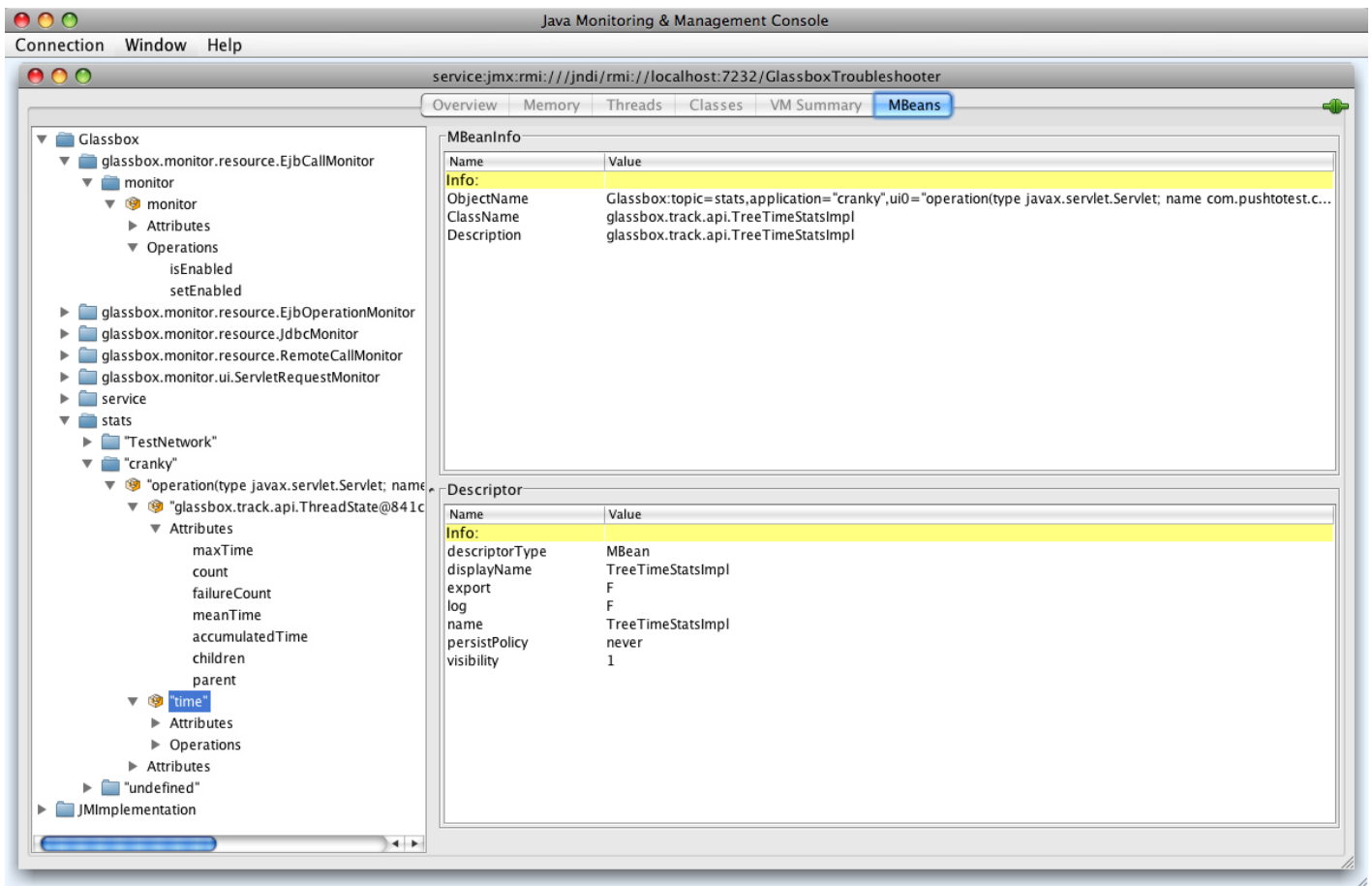
Glassbox Troubleshooter highlights problems with your application achieving your criteria for acceptable performance criteria. For example, Glassbox identifies slow object performance by setting a threshold in milliseconds. Use the [Glassbox com \(Hyperlink\)](#) figuration to set the criteria.

Appvance designed AIQ's Results Analysis Engine to provide drill-down functions on any of the Glassbox performance observations. [Contact Appvance Global Services](#) to learn about additional charting and analysis functions.

Glassbox provides a Java management (JMX) interface to the Troubleshooter's extended performance monitor data. Access this data through the [Java Management Console](#) (JConsole). JConsole comes with the [Java Development Kit](#) (JDK.) With the JDK installed run the JConsol in a shell or terminal window by typing the following:

```
jconsole service:jmx:rmi:///jndi/rmi://localhost:7232/GlassboxTroubleshooter
```

JConsole connects to the Glassbox Troubleshooter and displays a graphical interface shown in the following.



Last updated: June 02, 2023

If you have any feedback on this topic, please contact us at documentation@appvance.ai

Revision #2

Created 2026-07-31 16:29:46 UTC by Admin

Updated 2026-07-31 16:43:05 UTC by Admin