

Haben Sie Ihre WebLogic Umgebung im Griff
Daniel Joray
Trivadis AG
Bern

Keywords

WebLogic, JMS, Performance, J2EE, Monitoring

About Monitoring

Monitoring is a proactive - not reactive - approach to supervise a system

You have to ensure that...

- All components of your applications are up and running
- Your applications do not produce errors
- Your application setup is optimal to get out the maximum performance
- In a WebLogic Domain, there are hundreds of MBeans with thousands of metrics attributes
- It is not possible to monitor all those metrics

The purpose of this presentation is to define a limited set of metrics that allow you to monitor your WebLogic domain

JVM

The beauty of JVM technology is also its biggest challenge: nothing can be assumed with a "write once, run anywhere" application. Rather than optimizing for one use case, one application, and one specific user load, the JVM constantly tracks what is going on in a Java application and dynamically optimizes accordingly.

A JVM also takes care of dynamic resource management for running Java applications. This means it handles allocating and de-allocating memory, maintaining a consistent thread model on each platform, and organizing the executable instructions in a way that is suited for the CPU architecture where the application is executed.

Garbage collection is the JVM's process of freeing up unused Java objects in the Java heap. The Java heap is where the objects of a Java program live. It is a repository for live objects, dead objects, and free memory. When an object can no longer be reached from any pointer in the running program, it is considered "garbage" and ready for collection.

The memory management is not a task of WebLogic Server, this is the work of the Java Virtual Machine (JVM).

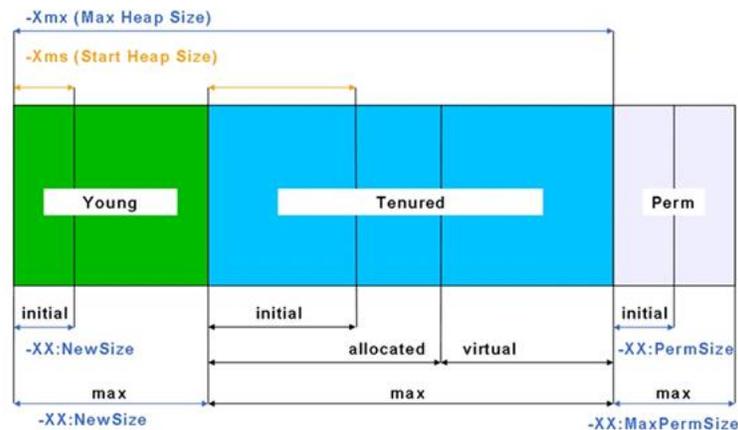
The heap memory is the runtime data area from which the Java VM allocates memory for all class instances and arrays. The heap may be of a fixed or variable size. The garbage collector is an automatic memory management system that reclaims heap memory for objects.

- Eden Space: The pool from which memory is initially allocated for most objects.

- Survivor Space: The pool containing objects that have survived the garbage collection of the Eden space.
- Tenured Generation: The pool containing objects that have existed for some time in the survivor space.

By default, NewRatio for the Server JVM is 2, the old generation occupies 2/3 of the heap while the new generation occupies 1/3

Memory management configuration are set on startup from JVM



```
$ java -Xms128 -Xmx256M -XX :NewRatio=3 -XX:MaxPermSize=128M
-Xms Initial size from heap
-Xmx Max size from heap
```

The most important metrics to check the JVM Memory allocation are:

- Percent use of the heap size
- Percent of time in Garbage Collection
 - GC is a stop the world process
 - It is very important to verify the system is not spending too much time in this state
- Max/ Mean time for the garbage collection

Another important concept of Java is the thread. A thread is a program's path of execution. Most programs written today run as a single thread, causing problems when multiple events or actions need to occur at the same time.

A thread pool is a group of threads initially created that waits for jobs and executes them. The value of the ThreadPoolSize attribute equals the number of simultaneous operations that can be performed by WebLogic Server. You can degrade performance by increasing this value unnecessarily.

The key metrics for the thread are:

- Idle Thread Counts
 - This count does not include standby threads and stuck threads. The count indicates threads that are ready to pick up new work when it arrives
- Thread Pool Queue length
 - The number of pending requests in the priority queue
- Hogging Thread Counts
 - Hogging threads are candidates for stuck threads.
 - Threads that “might” get stuck. These threads will be declared “stuck” after StuckThreadMaxTimeout seconds which usually is 600secs.

JDBC DataSources

A DataSource object is the representation of a data source in the Java programming language. In basic terms, a data source is a facility for storing data. It can be as sophisticated as a complex database for a large corporation or as simple as a file with rows and columns. A data source can reside on a remote server, or it can be on a local desktop machine. Applications access a data source using a connection, and a DataSource object can be thought of as a factory for connections to the particular data source that the DataSource instance represents

The key metrics for the thread are:

- Number Available
 - allows you to validate that you have the correct amount of resources available to service the clients needs
- Number Active Connections
 - allows you to validate that you have not too many amount of resources available to service the client’s needs
- Number Unavailable Connections
 - The number of connections currently in use by applications or being tested in this instance of the data source

JMS

The Java Message Service (JMS) is a standard API for accessing Enterprise Messaging systems. Specifically, JMS, enables Java applications sharing a messaging system to exchange messages and simplifies application development by providing a standard interface for creating, sending, and receiving messages

The key metrics for the thread are:

- JMS Pending Message
 - A message present on a JMS destination can be either current or pending. A current message is available for immediate consumption by consumers
- JMS Oldest Message Age
 - Normally it doesn’t matter how many messages are on the queue, it is a good idea to pay close attention to how old the oldest message is
 - This is normally a key indication of issues, or shows that the system is being affected by an excessive message dump on the queue – something the system cannot keep up with the load

- JMS Consumers
 - If no consumers are visible, then we
 - don't process messages

Application

A WebLogic Server application consists of one of the following modules or applications running on WebLogic Server:

- Web application modules - HTML pages, servlets, JavaServer Pages, and related files
- Enterprise Java Beans (EJB) modules - entity beans, session beans, and message-driven beans
- Connector modules - resource adapters

The key metrics for the thread are:

- Open Sessions Current
 - Specifies the number of sessions that are currently active in the application
- Open Session High
 - Refers to the highest number of sessions that were open for an Application, this is great data for capacity planning
- Bean Access Total
 - Total number of times an attempt was made to get a bean instance from the free pool
- Waiter Current Count
 - Number of threads currently waiting for an available bean instance from the pool

Logfiles

In a WebLogic Domain, most components have logfiles

- Domain
- Managed Server
- Instance Log File stderr/stdout of the JVM
 - server_name.out
- Server Log File
 - server_name.log
- HTTP access log File
 - access.log
- JDBC subsystem
 - datasource.log
- JMS subsystem
 - jmsServers\JMSServer_server\jms.messages.log

The messages for all WebLogic Server subsystems contain a consistent set of fields the most important are

- Severity
 - INFO, WARNING, ERROR, NOTICE, CRITICAL, ALERT, EMERGENCY
- Subsystem
- Message ID

- Message Text

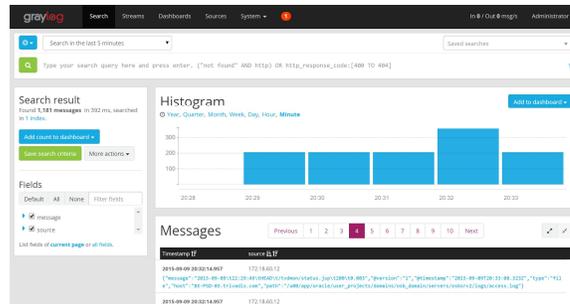
Logs are a critical part of any system, they give you insight into what a system is doing as well what happened. When your system grows to multiple hosts, managing the logs and accessing them can get complicated

Searching for a particular error across hundreds of log files on hundreds of servers is difficult without good tools

A common approach to this problem is to setup a centralized logging solution like Distributed Log Collectors

Splunk Light / Enterprise / Cloud makes it simple to collect, analyze and act upon the untapped value of the big data generated by your technology infrastructure, security systems and business applications

- Collect and Index Data
- Search and Investigate
- Correlate and Analyze
- Visualize and Report
- Monitor and Alert



Tools

To monitor / optimize your WebLogic Domain you can use:

- The WebLogic Console is probably the first place see the WebLogic metrics
- The jconsole is a JMX-compliant graphical tool for monitoring a Java virtual machine
- The jvisualvm is a tool that provides a visual interface for viewing detailed information about Java applications while they are running on a Java Virtual Machine (JVM), command allows you to query various aspects of specific virtual machines
- The Java Mission Control jmc and Java Flight Recorder together create a complete tool chain to continuously collect low level and detailed runtime information enabling after-the-fact incident analysis
- DomainHealth is a J2EE application which is deployed on a AdminServer, it shows the most key metrics
- GCViewer analyses logfile from the JVM garbage collection actions
- WLST – The WebLogic Scripting Tool
- Dynatrace / Appdynamics
 - Those softwares manages the availability and performance of software applications and the impact on user experience in the form of deep transaction tracing, synthetic monitoring, real user monitoring, and network monitoring.
- EM12c Middleware Monitoring Plug-Ins

- The WebLogic Server Management Pack Enterprise Edition greatly improves server as well as application performance by providing unique functionality to automatically detect performance bottlenecks; quickly diagnose these performance problems, and identify their root cause.

Summary

Each components from a WebLogic Domain should be monitored, but not each metrics and probably you need more than one tool.

Not all monitored metrics should send alarm, but in problem case or if you want plan the capacity of your system for the next months you should have enough information to do your job

Keep in mind monitoring requires system resources; you need to place the focus on the important metrics. Not optimal monitoring configuration will impact the application performance

Contact:

Daniel Joray

Principal Consultant
Infrastructure Managed Services

trivadis ag
Weltpoststrasse 5
CH-3015 Bern

Mobile +41-79-320-17-39
daniel.joray@trivadis.com
www.trivadis.com