Register | Login | Logout | Contact Us

Java-Success.com

Industrial strength Java/JEE Career Companion to open more doors

search here … Go

Home | Java FAQs | 600+ Java Q&As | Career | Tutorials | Member | Why?

Can u Debug? | Java 8 ready? | Top X | Productivity Tools | Judging Experience?

Home › Interview › Pressed for time? Java/JEE Interview FAQs › Java Key Area Essentials › ♦ Monitoring/Profiling Java applications interview Q&A

♦ Monitoring/Profiling Java applications interview Q&A

Posted on October 7, 2014 by Arulkumaran Kumaraswamipillai — No Comments ↓

0 Like Share 0 G+1

Q1 How will you go about monitoring your Java application for potential performance or memory issues?
A1

1. Visual VM for monitoring Java memory, CPU, threads, etc

VisualVM is a visual tool integrating several command-line JDK tools and lightweight profiling capabilities. Designed for both production and development time use, it further

600+ Full Stack Java/JEE Interview Q&As ♥Free ♦FAQs

open all | close all
- Ice Breaker Interview
- Core Java Interview (
- JEE Interview Q&A (3
- Pressed for time? Jav
  - Job Interview Ice B
  - FAQ Core Java Jol
  - FAQ JEE Job Inter
  - FAQ Java Web Se
  - Java Application Ar
  - Hibernate Job Inter
  - Spring Job Intervie
  - Java Key Area Ess
    - ♦ Design pattern
    - ♥ Top 10 causes
    - ♥♦ 01: 30+ Writir
    - ♦ 12 Java desigr
    - ♦ 18 Agile Devel
    - ♦ 5 Ways to debu
    - ♦ 9 Java Transa
    - ♦ Monitoring/Pro
    - 02: ♥♦ 13 Tips tc

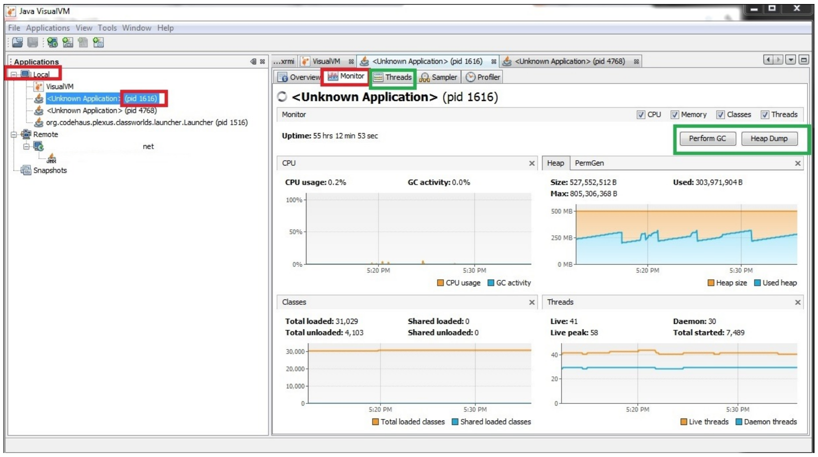
enhances the capability of monitoring and performance analysis for the Java SE platform. It is packaged as an exe file.

Step 1: You can start the visual vm by double clicking on %JAVA_HOME%/bin/jvisualvm.exe from Java 1.6 version onwards.

Step 2: Your local processes will be monitored under the local tab. The Visual vm can also used to open the heap dump files i.e *.hprof files to analyze the menory usages. You can find out the process ids of your local Java applications via

1. netstat -anp | grep 8088 or
2. In windows via the “Windows Task Manager” –> Processes tab. You need to click on View –> Select Columns and then “tick” PID (Process Identifier) check box.

Double click on the relevant PID in Visual VM console.


Java VisualVM
File Applications View Tools Window Help
Applications
Local
VisualVM
<Unknown Application> (pid 1616)
<Unknown Application> (pid 4768)
org.codehaus.plexus.classworlds.launcher.Launcher (pid 1516)
Remote
net
Snapshots
...xrmi
VisualVM
<Unknown Application> (pid 1616)
<Unknown Application> (pid 4768)
Overview
Monitor
Threads
Sampler
Profiler
<Unknown Application> (pid 1616)
Monitor
☑ CPU ☑ Memory ☑ Classes ☑ Threads
Perform GC
Heap Dump
CPU
Heap
PermGen
100%
50%
0%
500 MB
250 MB
0 MB
5:20 PM
5:30 PM
🟧 CPU usage 🟦 GC activity
🟧 Heap size 🟦 Used heap
Classes
Threads
30,000
20,000
10,000
0
40
20
🟧 Total loaded classes 🟦 Shared loaded classes
🟧 Live threads 🟦 Daemon threads

Uptime: 55 hrs 12 min 53 sec

CPU usage: 0.2% **GC activity:** 0.0%

Size: 527,552,512 B **Used:** 303,971,904 B
Max: 805,306,368 B

Total loaded: 31,029 **Shared loaded:** 0
Total unloaded: 4,103 **Shared unloaded:** 0

Live: 41 **Daemon:** 30
Live peak: 58 **Total started:** 7,489

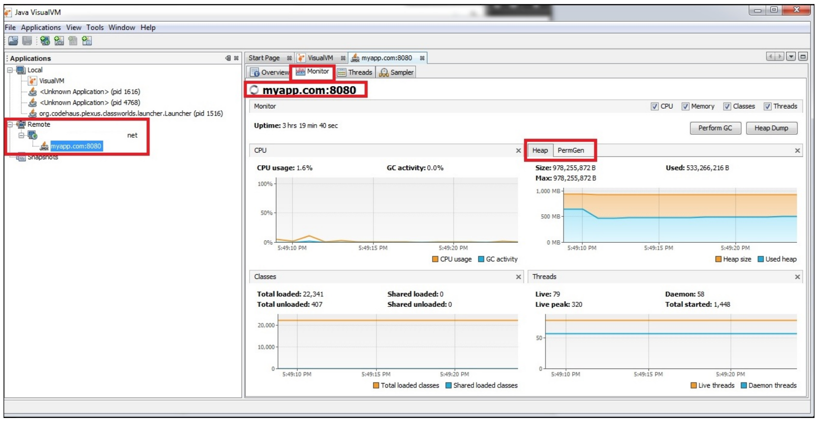
Visual VM

You can add remote processes by following the steps shown below.

1. Right click on “Remote” and then select “Add Remote Host…”.
2. Provide the host name like “myapp.com”.
3. It searches and adds the host.
4. Right click on the added host name and then select “Add

15 Security key a
4 FAQ Performa
4 JEE Design Pa
5 Java Concurre
6 Scaling your Ja
8 Java memory i
OOP & FP Essentia
Code Quality Job I
SQL, XML, UML, JSC
Hadoop & BigData Int
Java Architecture Inte
Scala Interview Q&As
Spring, Hibernate, & I
Spring (18)
Spring boot (4)
Spring IO (1)
Spring JavaConf
01: ♥♦ 13 Spring
01b: ♦ 13 Spring
02: ► Spring DII
03: ♥♦ Spring DI
04 ♦ 17 Spring b
05: ♦ 9 Spring Be
06: ♥ Debugging
07: Debugging S
Spring loading p
Hibernate (13)
01: ♥♦ 15+ Hiber
01b: ♦ 15+ Hiber
02: Understandin
03: Identifying ar
04: Identifying ar
05: Debugging H
06: Hibernate Fi
07: Hibernate mi
08: Hibernate au
09: Hibernate en
10: Spring, Java
11: Hibernate de
12: Hibernate cu
AngularJS (2)

JMX Connection" and in the "Connection" field type the hostname:JMX port number like myapp.com:8083.
5. Double click on this JMX connection to monitor CPU, memory, threads, Garbage collection, etc.



myapp.com:8080

Uptime: 3 hrs 19 min 40 sec

CPU usage: 1.6% GC activity: 0.0%

Size: 978,255,872 B Used: 533,266,216 B
Max: 978,255,872 B

Total loaded: 22,341 Shared loaded: 0
Total unloaded: 407 Shared unloaded: 0

Live: 79 Daemon: 58
Live peak: 320 Total started: 1,448

Visual VM

VisualVM's profiler works by "instrumenting" all of the methods of your code. This injects extra bytecode to your methods for recording when they're invoked, and how long they take to execute each time.

VisualVM's sampler takes a thread dump of all executing threads on regular basis, and uses the thread dumps to work out how much CPU time is consumed by each method.

2. Monitoring and gathering statistics with JMX based jconsole and Yammer metrics

When you are running long term applications like web applications, batch jobs, or stand-alone status update jobs, it is good to know some statistics about them, like number of requests served or request duration. You can also gather more generic information like the state of your internal collections, how many times some portion of code is being executed, or health checks like database availability, or any kind of connection to an external system.

⊞ Git & SVN (6)
⊞ JMeter (2)
⊞ JSF (2)
⊞ Maven (3)
⊞ Testing & Profiling/Sa
⊞ Other Interview Q&A f
⊞ ▶ Free Java Interviev

16 Technical Key Areas

open all | close all
⊞ Best Practice (6)
⊞ Coding (26)
⊞ Concurrency (6)
⊞ Design Concepts (7)
⊞ Design Patterns (11)
⊞ Exception Handling (3
⊞ Java Debugging (21)
⊞ Judging Experience Ir
⊞ Low Latency (7)
⊞ Memory Managemen
⊞ Performance (13)
⊞ QoS (8)
⊞ Scalability (4)
⊞ SDLC (6)
⊞ Security (13)
⊞ Transaction Managen

80+ step by step Java Tutorials

open all | close all
⊞ Setting up Tutorial (6)
⊞ Tutorial - Diagnosis (2
⊞ Akka Tutorial (9)
⊞ Core Java Tutorials (2
⊞ Hadoop & Spark Tuto

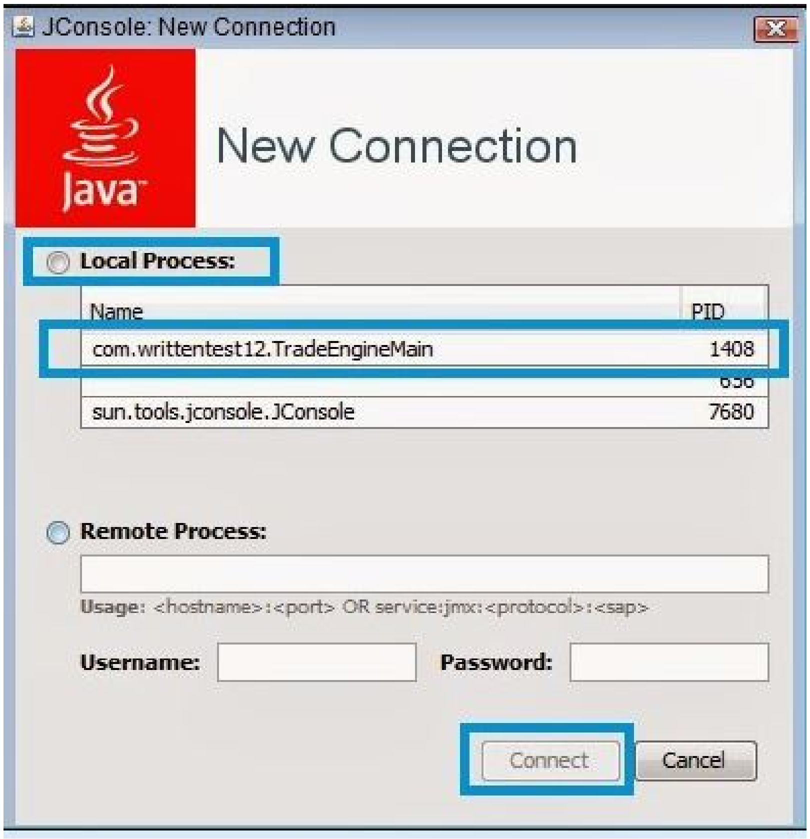
All this kind of instrumentation can be achieved by using native JMX and modular project like "Yammer Metrics". Metrics provides a powerful way to measure the behavior of your critical components and reporting them to a variety of systems like, JConsole (i.e via JMX), System Console, Ganglia, Graphite, CSV, or making them available through a web services as JSON data.

Step 1: Run the TradeEngineMain application.

Step 2: From the DOS command prompt, type jconsole

```
C:\Users\myapp>jconsole
```

Step 3: The jconsole window pops up as shown below:



Name	PID
com.writtentest12.TradeEngineMain	1408
	656
sun.tools.jconsole.JConsole	7680

jconsole

Step 4: You can monitor the statistics via the MBeans tab as shown below.

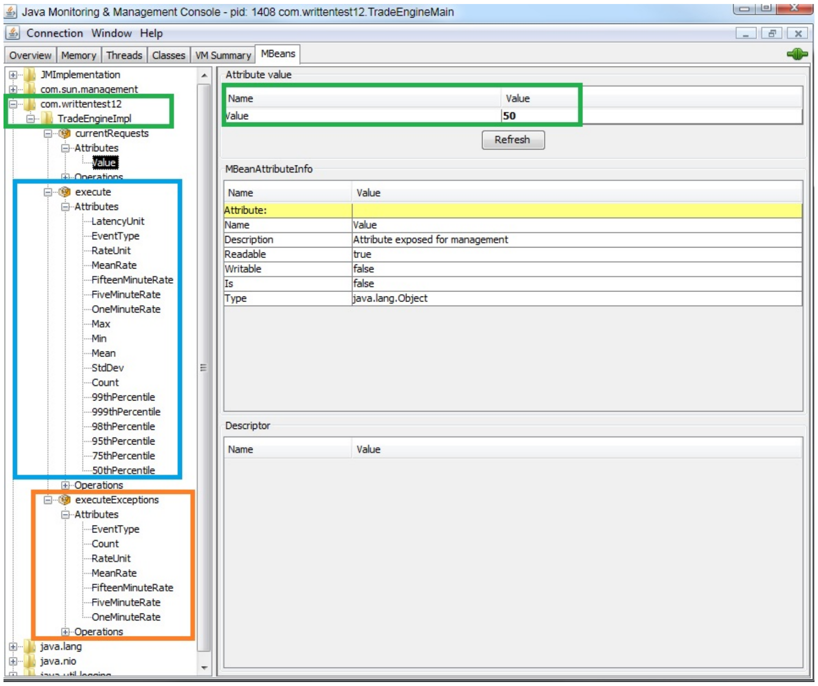
⊞ JEE Tutorials (19)
⊞ Scala Tutorials (1)
⊞ Spring & HIbernate Tu
⊞ Tools Tutorials (19)
⊞ Other Tutorials (45)

100+ Java pre-interview coding tests

open all | close all
⊞ Can you write code?
⊞ ♦ Complete the given
⊞ Converting from A to I
⊞ Designing your classe
⊞ Java Data Structures
⊞ Passing the unit tests
⊞ What is wrong with th
⊞ Writing Code Home A
⊞ Written Test Core Jav
⊞ Written Test JEE (1)

How good are your .....?

open all | close all
⊞ Career Making Know-
⊞ Job Hunting & Resum



Attribute value

Name	Value
Value	50

MBeanAttributeInfo

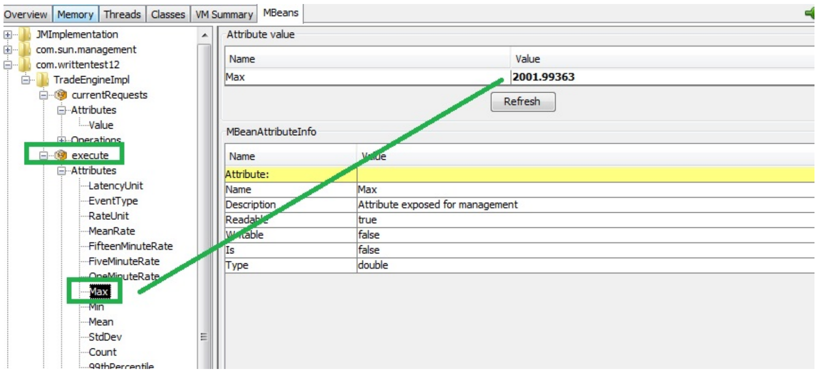
Name	Value
Attribute:	
Name	Value
Description	Attribute exposed for management
Readable	true
Writable	false
Is	false
Type	java.lang.Object

Descriptor

Name	Value

jconsole

Drill down into each and observe the values.



Attribute value

Name	Value
Max	2001.99363

MBeanAttributeInfo

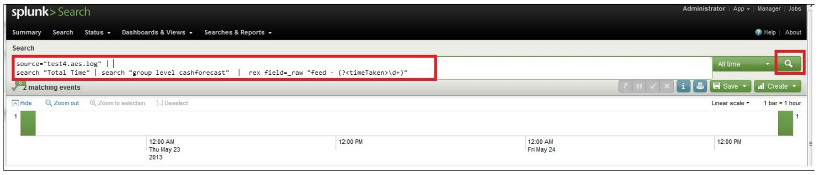
Name	Value
Attribute:	
Name	Max
Description	Attribute exposed for management
Readable	true
Writable	false
Is	false
Type	double

jconsole

With Yammer metrics, you can also output metrics to CSV, as JSON via RESTful web service, etc. It is quite a handy tool to monitor your applications.

3. Splunk to analyse Java logs and other machine data

Splunk is an enterprise-grade software tool for collecting and analyzing “machine data” like log files, feed files, and other big data in terra bytes. You can upload logs from your websites and let Splunk index them, and produce reports with graphs to analyze the reports. This is very useful in capturing start and finish times from asynchronous processes to calculate elapsed times.



```
source="test4.aes.log" | |
search "Total Time" | search "group level cashforecast"  |  rex field=_raw "feed - (?<timeTaken>\d+)"
```

2 matching events

Splunk

4. Monitoring your Java application infrastructures with tools like Nagios, Tivoli from IBM and SiteScope from HP.

which provides complete monitoring of Java applications and servers. Nagios Offers effective JMX monitoring. Nagios is an open source computer system monitoring, network monitoring and infrastructure monitoring software application. With Nagios you can send heart beat messages to say web services to ensure that they are available. When a service is not available, the “heart beat” message fails and raise an alert for the support staff to react.

Tivoli Management Framework (TMF) is a CORBA-based network management framework from IBM. BM Tivoli Monitoring helps you optimize IT infrastructure performance and availability. here is a high level overview of using Tivoli to monitor Java log files for presence of some text like errors and then raise support tickets at different levels like p1,p2,p3, and p4. The “P” is for production environment and 1-4 are level of importance. Any p1 and p2 level issues need to be attended immediately. The SCIM is used to identify the source of the ticket, and it stands for System, Component, Item, and Module.

Note: When it come to free tools for profiling and monitoring Java applications the usual choices are **JConsole** and **VisualVM**. **AppDynamics** now provides a third option to help you with web application performance monitoring.

There are other free tools to monitor performance like

- HeapAnalyzer is handy Java heap leak through its heuristic search engine and analysis of the Java heap dump in Java applications. It analyzes Java heap dumps by parsing the Java heap dump, creating directional graphs, transforming them into directional trees, and executing the heuristic search engine.
- PerfAnal is a GUI-based tool for analyzing the performance of applications on the Java 2 Platform. You can use PerfAnal to identify performance problems in your code and locate code that needs tuning.
- Eclipse Memory Analyzer is a fast and feature-rich Java heap analyzer that helps you find memory leaks and reduce memory consumption.
- GCViewer is a free open source tool to visualize data produced by the Java VM options -verbose:gc and -Xloggc:. It also calculates garbage collection related performance metrics like throughput, pauses, longest pause, etc.

Popular Posts

♦ 11 Spring boot interview questions & answers

827 views

♦ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers

767 views

18 Java scenarios based interview Questions and Answers

400 views

001A: ♦ 7+ Java integration styles & patterns interview questions & answers

389 views

01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

296 views

♦ 7 Java debugging interview questions & answers

293 views

01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

286 views

♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers

279 views

♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

240 views

001B: ♦ Java architecture & design concepts interview questions & answers

202 views

Bio | **Latest Posts**

Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

About Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

‹ Unix remoting interview Q&A for Java developers

Auditing, data retention and archiving interview Q&A ›

Posted in Java Key Area Essentials, member-paid, Memory Management, Performance, QoS

Leave a Reply

Logged in as geethika. Log out?

Comment

Post Comment

Empowers you to open more doors, and fast-track

Technical Know Hows

☀ Java generics in no time ☀ Top 6 tips to transforming your thinking from OOP to FP ☀ How does a HashMap internally work? What is a hashing function?
☀ 10+ Java String class interview Q&As ☀ Java auto un/boxing benefits & caveats ☀ Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect

Non-Technical Know Hows

☀ 6 Aspects that can motivate you to fast-track your career & go places ☀ Are you reinventing yourself as a Java developer? ☀ 8 tips to safeguard your Java career against offshoring ☀ My top 5 career mistakes

Prepare to succeed

☀ Turn readers of your Java CV go from "Blah blah" to "Wow"? ☀ How to prepare for Java job interviews? ☀ 16 Technical Key Areas ☀ How to choose from multiple Java job offers?

Select Category ▼

© Disclaimer

The contents in this Java-Success are copy righted. The author has the right to correct or enhance the current content without any prior notice.

These are general advice only, and one needs to take his/her own circumstances into consideration. The author will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. No guarantees are made regarding the accuracy or usefulness of content, though I do make an effort to be accurate. Links to external sites do not imply endorsement of the linked-to sites.