

VisualVM

Durchblick für den Admin

und Entwickler ;-)

DEMO

Der Vortrag bestand fast ausschließlich aus einer praktischen Demonstration der VisualVM, Einsatz während eines Lasttest und der Performanceoptimierung aus den Erkenntnissen.

JDK Admin-Werkzeuge

- **Console**

- jps

- jinfo

- jstack

- jmap

- **Graphisch**

- jconsole

- jvisualvm

Woher?

- im JDK (auch OpenJDK) ab 1.6 enthalten
- Standalone:
 - <https://visualvm.dev.java.net/>
- Runtime-Voraussetzung: JDK 1.6+

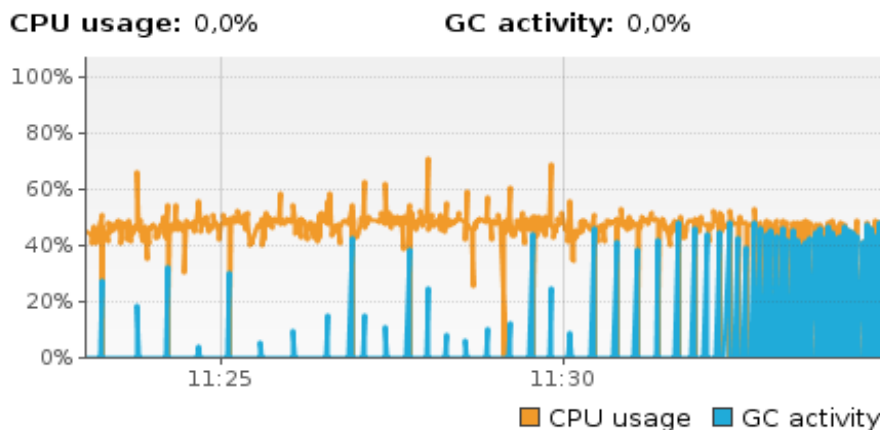
```
[tobi@fedora ~]$ which java
/usr/bin/java
[tobi@fedora ~]$ ls -l /usr/bin/java
lrwxrwxrwx. 1 root root 22 27. Jun 2008 /usr/bin/java -> /etc/alternatives/java
[tobi@fedora ~]$ ls -l /etc/alternatives/java
lrwxrwxrwx 1 root root 30 27. Jun 15:50 /etc/alternatives/java -> /usr/java/jdk1
.6.0_20/bin/java
[tobi@fedora ~]$ /usr/java/jdk1.6.0_20/bin/jvisualvm █
```

org.eclipse.jdt.internal.junit.runner.RemoteTestRunner (pid 15834)

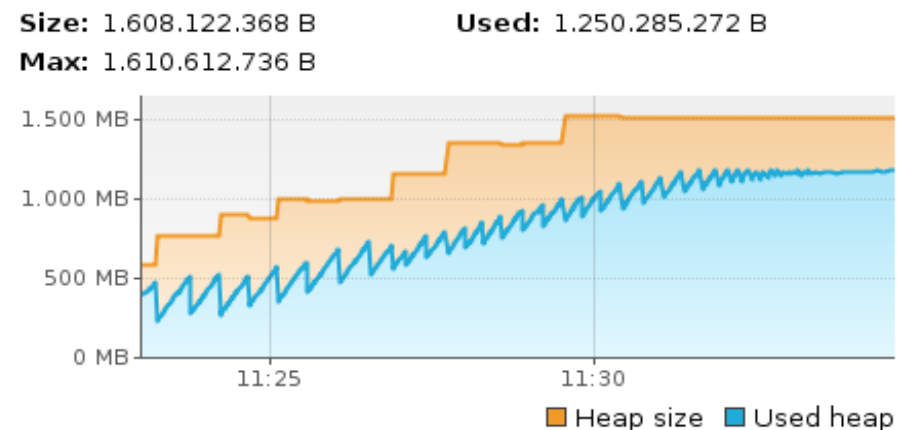
Monitor CPU Memory Classes Threads

Uptime: 19 min 54 sec Perform GC Heap Dump

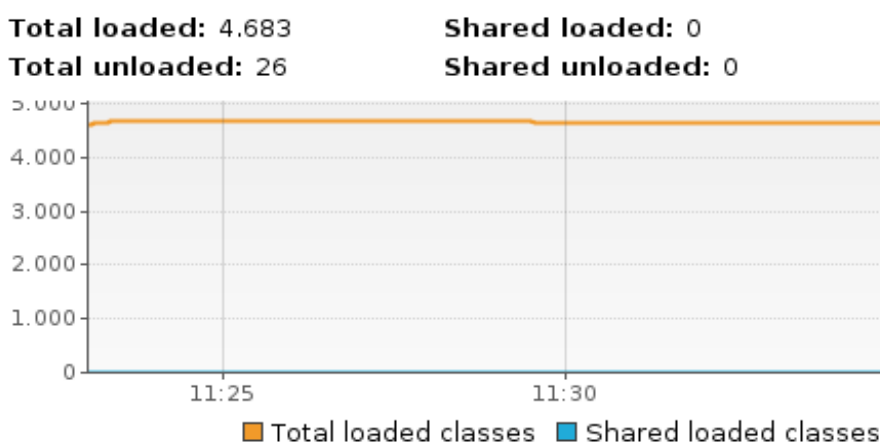
CPU ×



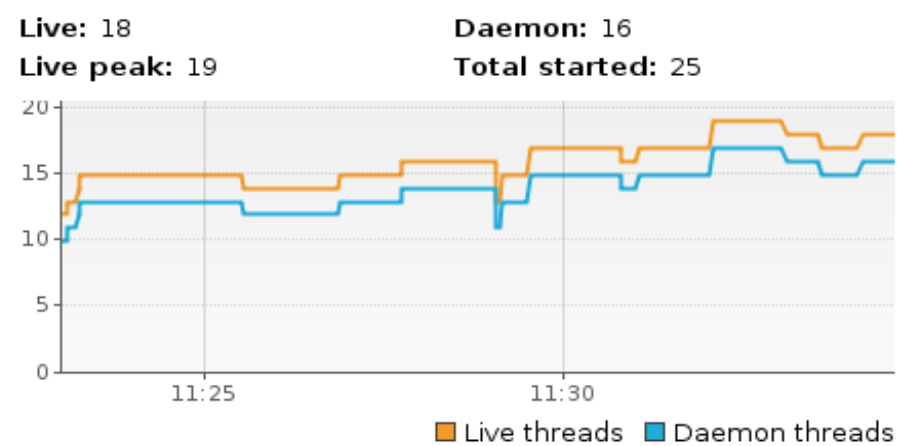
Heap PermGen ×



Classes ×



Threads ×



VisualVM

- Netbeans Application
- Netbeans Profiler

JDKs anbinden

- lokal
 - -Dcom.sun.management.jmxremote
- remote
 - -Dcom.sun.management.jmxremote.port=12345
 - -Dcom.sun.management.jmxremote.authenticate=false
 - -Dcom.sun.management.jmxremote.ssl=false
- Zugriff auf JDK 5 möglich

Grundfunktionen

- JVM
- CPU
- Speicher, Garbage Collection, HeapDump
- Threads, ThreadDumps
- Profiler

Snapshots

- halten den momentanen, sonst flüchtigen Zustand fest
- können als Datei zur Analyse transportiert werden
- Application Snapshots
- Profiler/Sampler Snapshots

Interessante Plugins

- VisualGC
- MBeans
- Sampler
- JConsole -> TopThreads
- ThreadDumpAnalyzer (TDA)

Tipps

- RMI-Name setzen
 - `-Djava.rmi.server.hostname=...`
- Profiler kann Bilder speichern

Viel Spaß mit VisualVM!