Java Thread Analyse

Threads

- Ein Thread ist ein unabhängiger Ausführungspfad in einem Programm.
- In JBoss existieren viele Threads gleichzeitig
- Stati:
  - NEW: Noch nicht gestartet
  - RUNNABLE: Läuft aktiv in der JVM
  - BLOCKED: Blockiert und wartet auf einen Monitor-Lock (synchronized)
  - WAITING: Wartet unbegrenzt auf eine Aktion eines anderen Threads (Object.wait(), Thread.join() - Object.notify())
  - TIMED_WAITING: Wartet eine gewisse Zeit auf eine Aktion eines anderen Threads
  - TERMINATED: Thread ist beendet

Thread Dump

```
"http-/0.0.0.0:8080-Acceptor-0" daemon prio=10 tid=0x00007f62d8b9f000 nid=0x7d36 runnable [0x00007f62d1be3000]
   java.lang.Thread.State: RUNNABLE
    at java.net.PlainSocketImpl.socketAccept(Native Method)
    at java.net.PlainSocketImpl.accept(PlainSocketImpl.java:408)
    - locked <0x00000000ba4ad580> (a java.net.SocksSocketImpl)
    at java.net.ServerSocket.implAccept(ServerSocket.java:462)
    at java.net.ServerSocket.accept(ServerSocket.java:430)
    at org.apache.tomcat.util.net.DefaultServerSocketFactory.acceptSocket(DefaultServerSocketFactory.java:61)
    at org.apache.tomcat.util.net.JIoEndpoint$Acceptor.run(JIoEndpoint.java:322)
    at java.lang.Thread.run(Thread.java:662)

"http-/0.0.0.0:8080-Poller" daemon prio=10 tid=0x00007f62d8601000 nid=0x7d35 in Object.wait()
[0x00007f62d1ce4000]
   java.lang.Thread.State: TIMED_WAITING (on object monitor)
    at java.lang.Object.wait(Native Method)
    - waiting on <0x00000000ba4ade78> (a org.apache.tomcat.util.net.JIoEndpoint$Poller)
    at org.apache.tomcat.util.net.JIoEndpoint$Poller.run(JIoEndpoint.java:743)
    - locked <0x00000000ba4ade78> (a org.apache.tomcat.util.net.JIoEndpoint$Poller)
    at java.lang.Thread.run(Thread.java:662)

"EJB default - 7" prio=10 tid=0x00007f62f8347000 nid=0x7dde waiting on condition [0x00007f62f4bf9000]
   java.lang.Thread.State: TIMED_WAITING (sleeping)
    at java.lang.Thread.sleep(Native Method)
    at test.PerfCall.warte(PerfCall.java:33)
    at sun.reflect.GeneratedMethodAccessor19.invoke(Unknown Source)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
    at java.lang.reflect.Method.invoke(Method.java:597)
    at org.jboss.as.ee.component.ManagedReferenceMethodInterceptorFactory$ManagedReferenceMethodInterceptor.
    at org.jboss.invocation.InterceptorContext.proceed(InterceptorContext.java:288)
    at org.jboss.invocation.InterceptorContext$Invocation.proceed(InterceptorContext.java:374)
    at org.jboss.as.weld.ejb.Jsr299BindingsInterceptor.doMethodInterception(Jsr299BindingsInterceptor.java:129)
    at org.jboss.as.weld.ejb.Jsr299BindingsInterceptor.processInvocation(Jsr299BindingsInterceptor.java:137)
    at org.jboss.as.ee.component.interceptors.UserInterceptorFactory$1.processInvocation
    at org.jboss.invocation.InterceptorContext.proceed(InterceptorContext.java:288)
    at org.jboss.invocation.WeavedInterceptor.processInvocation(WeavedInterceptor.java:53)
```

Stack Trace

Source Code Referenz

Stack Trace

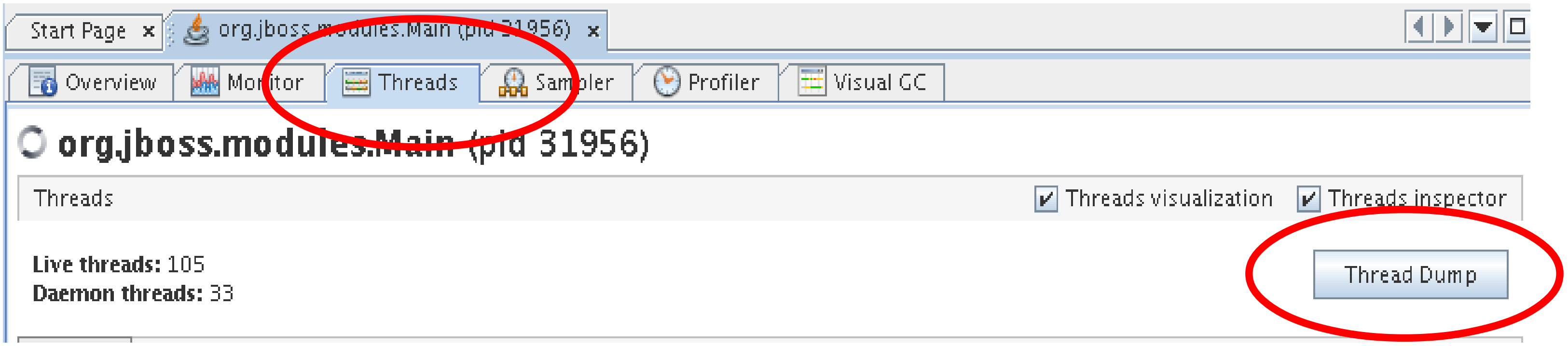
```
    at java.lang.Thread.sleep(Native Method)
    at test.PerfCall.warte(PerfCall.java:33)
    at sun.reflect.GeneratedMethodAccessor19.invoke(Unknown Source)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
    at java.lang.reflect.Method.invoke(Method.java:597)
    at org.jboss.as.ee.component.ManagedReferenceMethodInterceptorFactory$ManagedReferenceMethodInterceptor.
    at org.jboss.invocation.InterceptorContext.proceed(InterceptorContext.java:288)
    at org.jboss.invocation.InterceptorContext$Invocation.proceed(InterceptorContext.java:374)
    at org.jboss.as.weld.ejb.Jsr299BindingsInterceptor.doMethodInterception(Jsr299BindingsInterceptor.java:129)
    at org.jboss.as.weld.ejb.Jsr299BindingsInterceptor.processInvocation(Jsr299BindingsInterceptor.java:137)
    at org.jboss.as.ee.component.interceptors.UserInterceptorFactory$1.processInvocation
    at org.jboss.invocation.InterceptorContext.proceed(InterceptorContext.java:288)
    at org.jboss.invocation.WeavedInterceptor.processInvocation(WeavedInterceptor.java:53)
```

```
 public class PerfCall implements Perf {
   ...
      public String warte(long n){
        if ( log.isDebugEnabled() ){
            log.debug("Schlafe " + n + " Millisec");
        }
        try {
            Thread.sleep(n);
        } catch (InterruptedException e) {
            throw new RuntimeException("Interrupt in warte!", e);
        }
        return "Geschlafen: " + n + " Millisec";
    }
```

Zeile: PerfCall.java:33

Thread Dumps erzeugen

- jstack *pid*
- kill -QUIT *pid* Prozess-ID (pid) mit ps oder jps auslesen
- jvisualvm

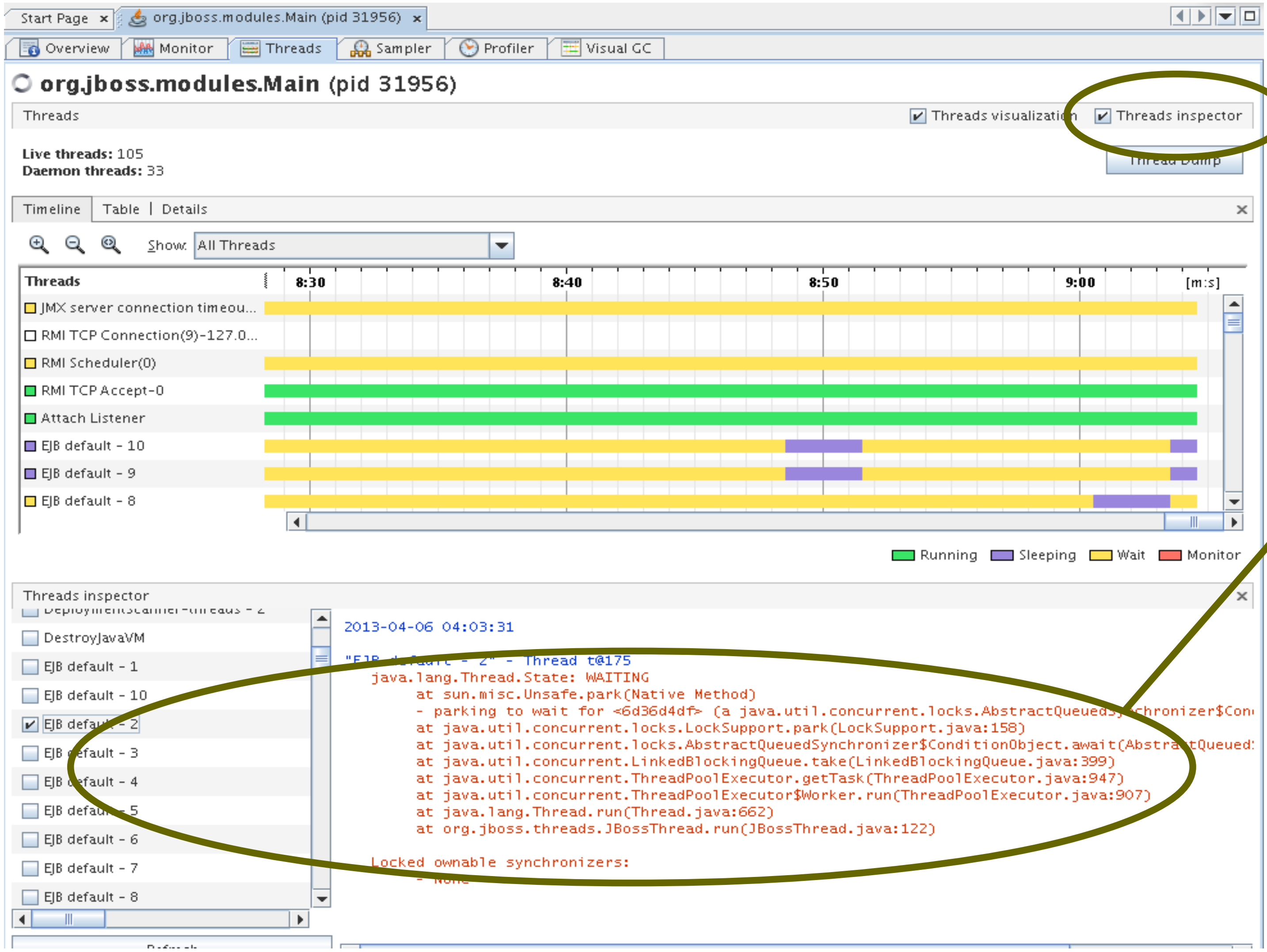


org.jboss.modules.Main (pid 31956)

Threads

Live threads: 105
Daemon threads: 33

Threads beobachten mit JVisualVM



org.jboss.modules.Main (pid 31956)

Threads

Live threads: 105
Daemon threads: 33

Threads	8:30 – 9:00 [m:s]
JMX server connection timeou...	🟨
RMI TCP Connection(9)-127.0...	
RMI Scheduler(0)	🟨
RMI TCP Accept-0	🟩
Attach Listener	🟩
EJB default - 10	🟨 🟪 🟨 🟪
EJB default - 9	🟨 🟪 🟨 🟪
EJB default - 8	🟨 🟪 🟨

Threads inspector

- DestroyJavaVM
- EJB default - 1
- EJB default - 10
- [✓ EJB default - 2]
- EJB default - 3
- EJB default - 4
- EJB default - 5
- EJB default - 6
- EJB default - 7
- EJB default - 8

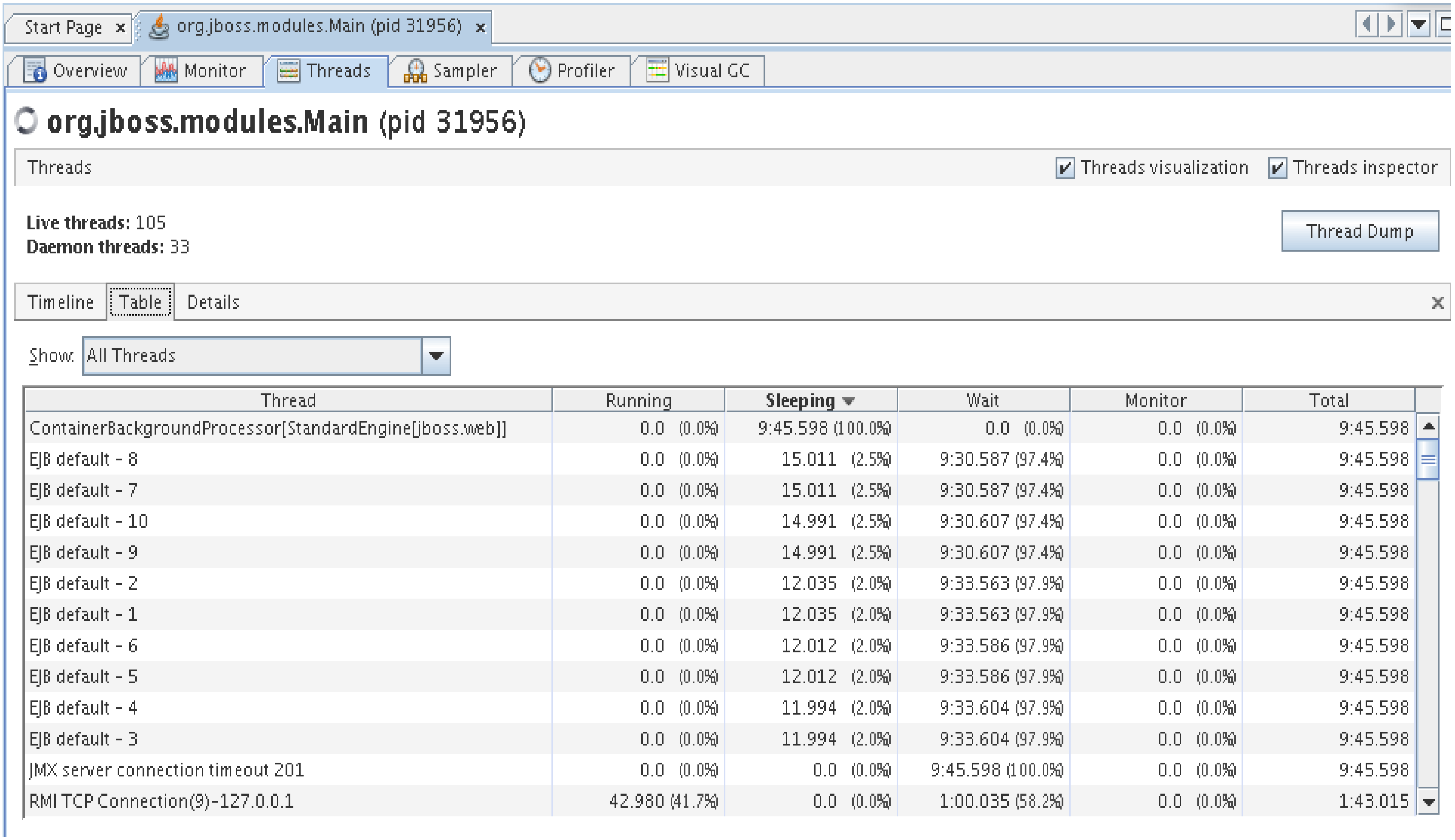
```
2013-04-06 04:03:31

"EJB default - 2" - Thread t@175
   java.lang.Thread.State: WAITING
        at sun.misc.Unsafe.park(Native Method)
        - parking to wait for <6d36d4df> (a java.util.concurrent.locks.AbstractQueuedSynchronizer$Con
        at java.util.concurrent.locks.LockSupport.park(LockSupport.java:158)
        at java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.await(AbstractQueued
        at java.util.concurrent.LinkedBlockingQueue.take(LinkedBlockingQueue.java:399)
        at java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:947)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:907)
        at java.lang.Thread.run(Thread.java:662)
        at org.jboss.threads.JBossThread.run(JBossThread.java:122)

   Locked ownable synchronizers:
        - None
```

Threads Inspe
Plugin

Threads beobachten mit JVisualVM



org.jboss.modules.Main (pid 31956)

Threads

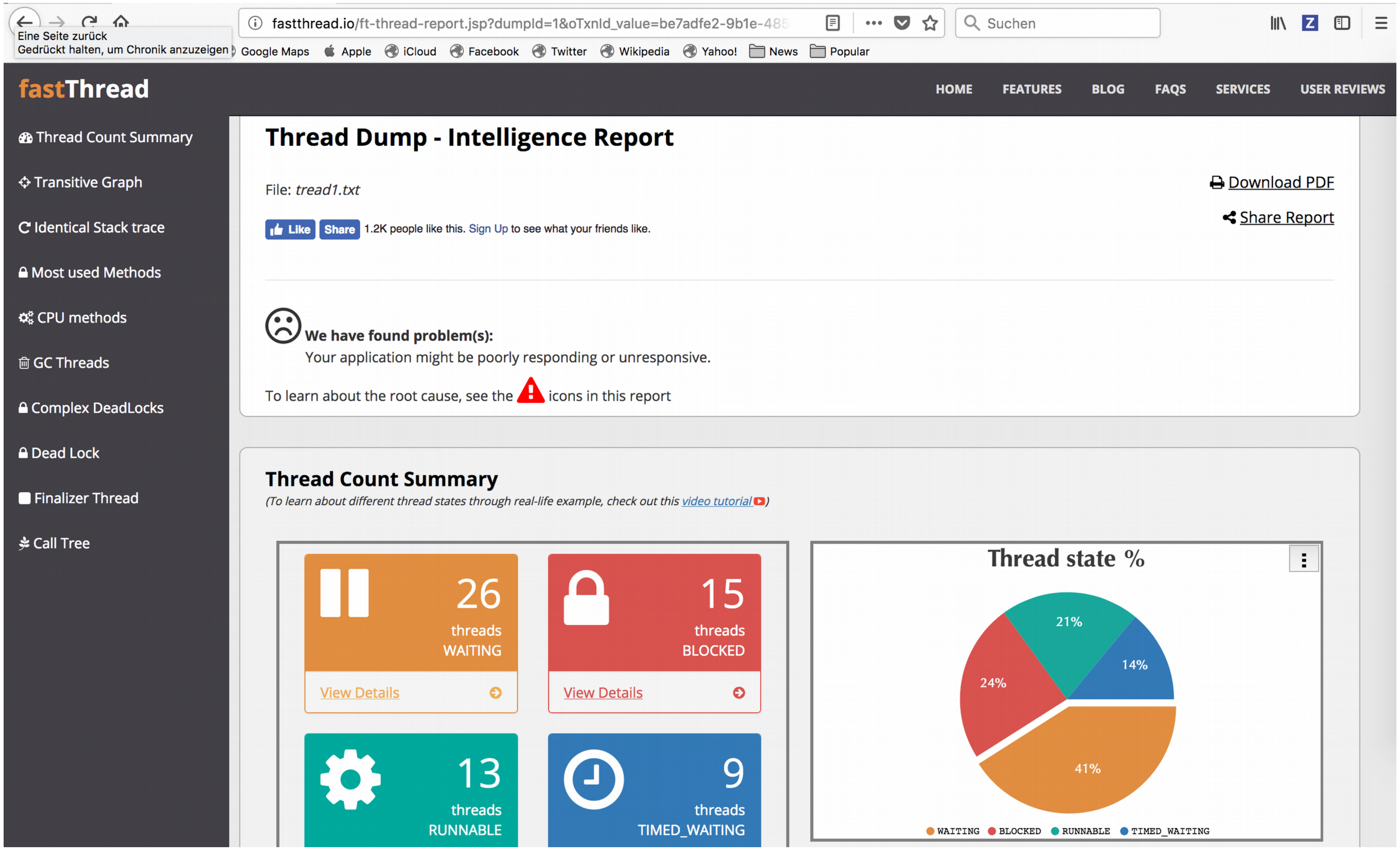
Threads visualization [✓] Threads inspector [✓]

Live threads: 105
Daemon threads: 33

Show: All Threads

Thread	Running	Sleeping	Wait	Monitor	Total
ContainerBackgroundProcessor[StandardEngine[jboss.web]]	0.0 (0.0%)	9:45.598 (100.0%)	0.0 (0.0%)	0.0 (0.0%)	9:45.598
EJB default - 8	0.0 (0.0%)	15.011 (2.5%)	9:30.587 (97.4%)	0.0 (0.0%)	9:45.598
EJB default - 7	0.0 (0.0%)	15.011 (2.5%)	9:30.587 (97.4%)	0.0 (0.0%)	9:45.598
EJB default - 10	0.0 (0.0%)	14.991 (2.5%)	9:30.607 (97.4%)	0.0 (0.0%)	9:45.598
EJB default - 9	0.0 (0.0%)	14.991 (2.5%)	9:30.607 (97.4%)	0.0 (0.0%)	9:45.598
EJB default - 2	0.0 (0.0%)	12.035 (2.0%)	9:33.563 (97.9%)	0.0 (0.0%)	9:45.598
EJB default - 1	0.0 (0.0%)	12.035 (2.0%)	9:33.563 (97.9%)	0.0 (0.0%)	9:45.598
EJB default - 6	0.0 (0.0%)	12.012 (2.0%)	9:33.586 (97.9%)	0.0 (0.0%)	9:45.598
EJB default - 5	0.0 (0.0%)	12.012 (2.0%)	9:33.586 (97.9%)	0.0 (0.0%)	9:45.598
EJB default - 4	0.0 (0.0%)	11.994 (2.0%)	9:33.604 (97.9%)	0.0 (0.0%)	9:45.598
EJB default - 3	0.0 (0.0%)	11.994 (2.0%)	9:33.604 (97.9%)	0.0 (0.0%)	9:45.598
JMX server connection timeout 201	0.0 (0.0%)	0.0 (0.0%)	9:45.598 (100.0%)	0.0 (0.0%)	9:45.598
RMI TCP Connection(9)-127.0.0.1	42.980 (41.7%)	0.0 (0.0%)	1:00.035 (58.2%)	0.0 (0.0%)	1:43.015

Thread Analysis - fastthread.io



fastthread.io/ft-thread-report.jsp?dumpId=1&oTxnId_value=be7adfe2-9b1e-485

Eine Seite zurück
Gedrückt halten, um Chronik anzuzeigen

Google Maps · Apple · iCloud · Facebook · Twitter · Wikipedia · Yahoo! · News · Popular

fastThread — HOME · FEATURES · BLOG · FAQS · SERVICES · USER REVIEWS

- Thread Count Summary
- Transitive Graph
- Identical Stack trace
- Most used Methods
- CPU methods
- GC Threads
- Complex DeadLocks
- Dead Lock
- Finalizer Thread
- Call Tree

Thread Dump - Intelligence Report

File: *tread1.txt*

Download PDF
Share Report

Like Share 1.2K people like this. Sign Up to see what your friends like.

We have found problem(s):
Your application might be poorly responding or unresponsive.

To learn about the root cause, see the ⚠ icons in this report

Thread Count Summary
(To learn about different thread states through real-life example, check out this video tutorial)

26 threads WAITING — View Details	15 threads BLOCKED — View Details
13 threads RUNNABLE	9 threads TIMED_WAITING

gepardec

CPU Verbrauch

- Prozessspezifisch
  - ps -o '%cpu,%mem' -p *pid*

```
[esiegl@devjava4 ~]$ ps -eo pid,ppid,cmd,%mem,%cpu --sort=-%mem | head
  PID  PPID CMD                         %MEM %CPU
28190 28118 /usr/java/jdk1.8.0_144/bin/ 11.5 56.8
 2012     1 /usr/libexec/packagekitd     3.5  0.0
 2471  2320 /usr/bin/gnome-shell         3.1  0.0
  623     1 /usr/lib/systemd/systemd-jo  1.9  0.0
 1778  1755 gnome-shell --mode=gdm --wa  1.4  0.0
```

- Treadspezifisch
  - top -H und jstack

Threads mit höchster CPU-Auslastung

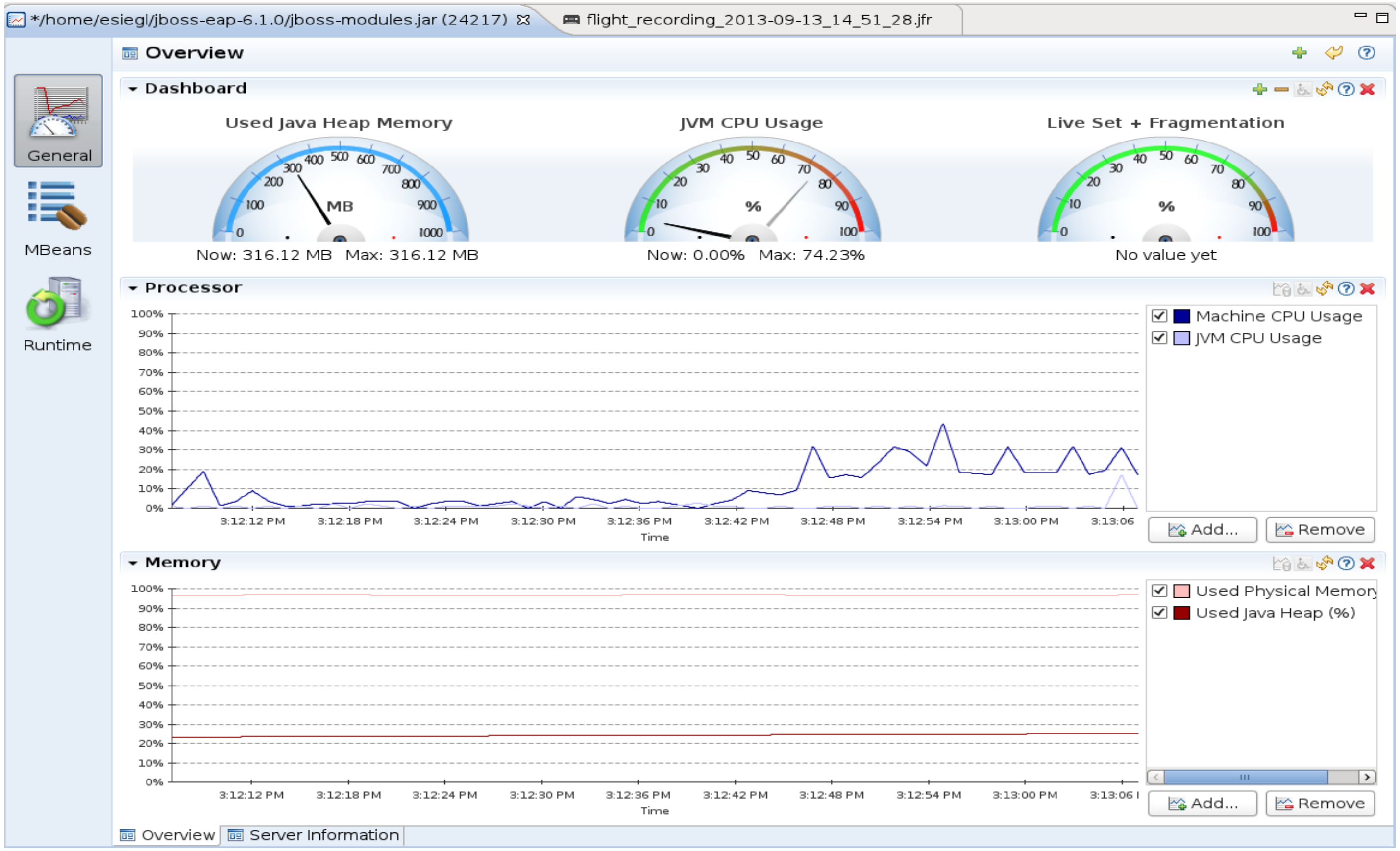
Thread-ID nach HEX umwandeln

Java-Thread suchen

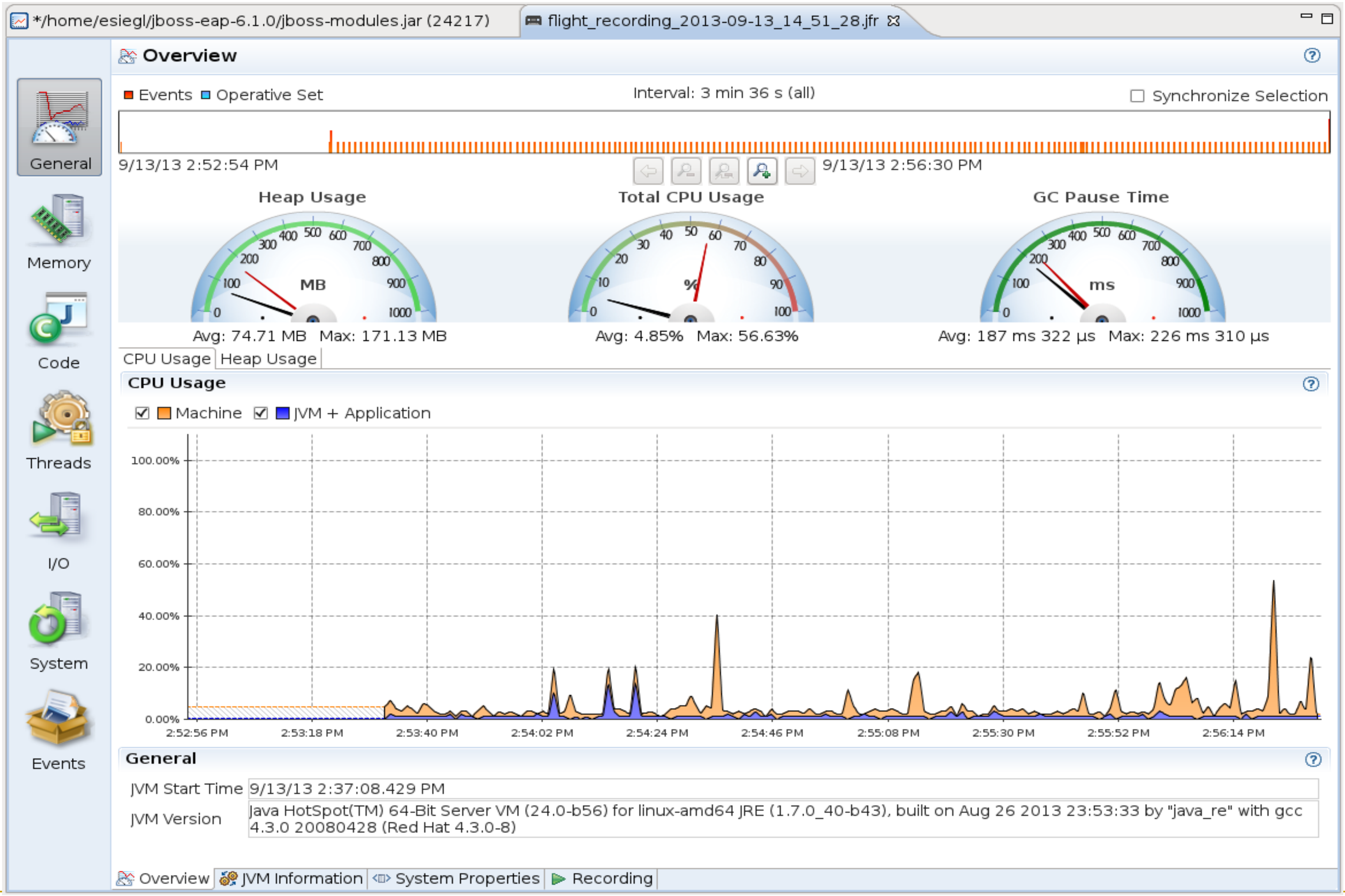
```
jboss:/home/jboss>top -H
jboss:/home/jboss>echo "obase=16; 58577" | bc
E4D1
jboss:/home/jboss>jstack 31201 | grep -i E4D1
"RequestHandlerThread[#3182]" daemon prio=10 tid=0x00007ffd9800b000 nid=0xe4d1
runnable [0x00007ffdfbbf9000]
```

Java Mission Control

- (JBoss) JVM starten mit:
  - JAVA_OPTS="$JAVA_OPTS -XX:+UnlockCommercialFeatures -XX:+FlightRecorder"



JMC Flight Recorder



*/home/esiegl/jboss-eap-6.1.0/jboss-modules.jar (24217)

flight_recording_2013-09-13_14_51_28.jfr

Overview

Interval: 3 min 36 s (all)

☐ Synchronize Selection

9/13/13 2:52:54 PM — 9/13/13 2:56:30 PM

Heap Usage — Avg: 74.71 MB Max: 171.13 MB

Total CPU Usage — Avg: 4.85% Max: 56.63%

GC Pause Time — Avg: 187 ms 322 µs Max: 226 ms 310 µs

General

JVM Start Time: 9/13/13 2:37:08.429 PM

JVM Version: Java HotSpot(TM) 64-Bit Server VM (24.0-b56) for linux-amd64 JRE (1.7.0_40-b43), built on Aug 26 2013 23:53:33 by "java_re" with gcc 4.3.0 20080428 (Red Hat 4.3.0-8)

Flight Recorder mit Kommandozeile

- Export Konfiguration aus jmc
  - File → Export → Flight Recording Configuration Template
- Starte Recording
  - jcmd `jboss pid` VM.unlock_commercial_features
  - jcmd `jboss pid` JFR.start duration=60s filename=recording1.jfr settings=/home/jboss/my.jfc
- Referenz:
  - https://docs.oracle.com/javacomponents/jmc-5-5/jfr-runtime-guide/comline.htm