

Applications

Project 1

IntelliJ IDEA CE (pid 52595)

VisualVM

org.jetbrains.jps.cmdline.Launcher (pid 1273)

ru.job4j.gc.prof.GarbMainMenu (pid 1274)

Remote

VM Coredumps

JFR Snapshots

Snapshots

VisualVM, 9:10:04 PM

Start Page

ru.job4j.gc.prof.GarbMainMenu (pid 1274)

Overview

Monitor

Threads

Sampler

Profiler

## ru.job4j.gc.prof.GarbMainMenu (pid 1274)

Monitor

☒ CPU ☐ Memory ☐ Classes ☐ Threads

Uptime: 3 min 52 sec

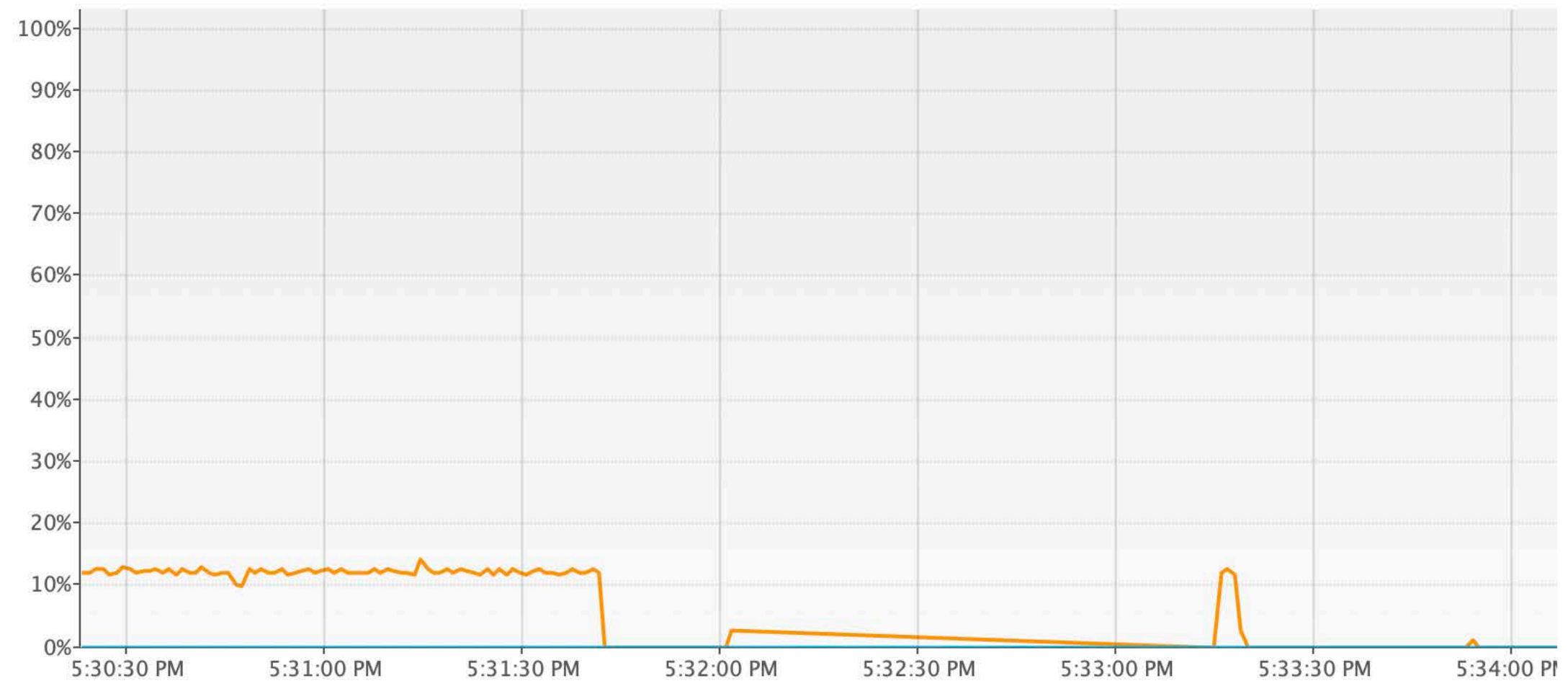
Perform GC

Heap Dump

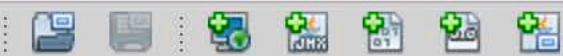
CPU

CPU usage: 0.0%

GC activity: 0.0%



CPU usage GC activity



Applications

- Local
  - IntelliJ IDEA CE (pid 52595)
  - VisualVM
  - org.jetbrains.jps.cmdline.Launcher (pid 1273)
  - ru.job4j.gc.prof.GarbMainMenu (pid 1274)
- Remote
- VM Coredumps
- JFR Snapshots
- Snapshots
  - VisualVM, 9:10:04 PM

Start Page

ru.job4j.gc.prof.GarbMainMenu (pid 1274)



Overview



Monitor



Threads



Sampler



Profiler

## ru.job4j.gc.prof.GarbMainMenu (pid 1274)

Monitor



CPU



Memory



Classes



Threads

Uptime: 4 min 01 sec

Perform GC

Heap Dump

Heap

Metaspace

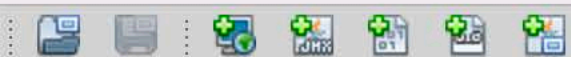
Size: 25,165,824 B

Used: 12,098,944 B

Max: 25,165,824 B



Heap size Used heap



Applications

- Local
  - IntelliJ IDEA CE (pid 52595)
  - VisualVM
  - org.jetbrains.jps.cmdline.Launcher (pid 1273)
  - ru.job4j.gc.prof.GarbMainMenu (pid 1274)**
- Remote
- VM Coredumps
- JFR Snapshots
- Snapshots
  - VisualVM, 9:10:04 PM

Start Page

ru.job4j.gc.prof.GarbMainMenu (pid 1274)

Overview

Monitor

Threads

Sampler

Profiler

## ru.job4j.gc.prof.GarbMainMenu (pid 1274)

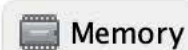
Sampler

Settings

Sample:



CPU



Memory



Stop

Status: memory sampling in progress

Heap histogram

Per thread allocations

Results:



Collected data:



Snapshot

Perform GC

Heap Dump

Name	Live Bytes	Live Objects
<b>int[]</b>	4,444,864 B (44.3%)	2,061 (1.7%)
<b>byte[]</b>	1,666,576 B (16.6%)	21,055 (17.5%)
<b>java.lang.String</b>	449,760 B (4.5%)	18,740 (15.5%)
<b>java.lang.Object[]</b>	388,520 B (3.9%)	8,752 (7.3%)
<b>java.lang.Class</b>	340,528 B (3.4%)	2,811 (2.3%)
<b>char[]</b>	189,912 B (1.9%)	640 (0.5%)
<b>java.util.HashMap\$Node</b>	178,368 B (1.8%)	5,574 (4.6%)
<b>java.util.TreeMap\$Entry</b>	160,000 B (1.6%)	4,000 (3.3%)
<b>java.lang.reflect.Method</b>	137,368 B (1.4%)	1,561 (1.3%)
<b>java.util.concurrent.ConcurrentHashMap\$Node</b>	134,176 B (1.3%)	4,193 (3.5%)
<b>java.util.HashMap\$Node[]</b>	126,104 B (1.3%)	1,197 (1%)
<b>java.util.LinkedHashMap\$Entry</b>	66,920 B (0.7%)	1,673 (1.4%)
<b>java.lang.String[]</b>	58,456 B (0.6%)	1,626 (1.3%)
<b>java.lang.Class[]</b>	54,112 B (0.5%)	2,338 (1.9%)
<b>java.util.ImmutableCollections\$List12</b>	51,696 B (0.5%)	2,154 (1.8%)
<b>java.util.HashMap</b>	42,864 B (0.4%)	893 (0.7%)
<b>java.util.concurrent.ConcurrentHashMap\$Node[]</b>	40,656 B (0.4%)	121 (0.1%)