

Performance Testing using Visual VM

Hprof

File Name: java.hprof.txt

Command

```
E:\Marist\Semester2\SoftwareVerificationAndMaintenance\roomScheduler\target>java -
agentlib:hprof=cpu=samples, interval=20,depth=3 -jar RoomScheduler-0.0.1-SNAPSHOT.jar
Dumping CPU usage by sampling running threads ... done.
```

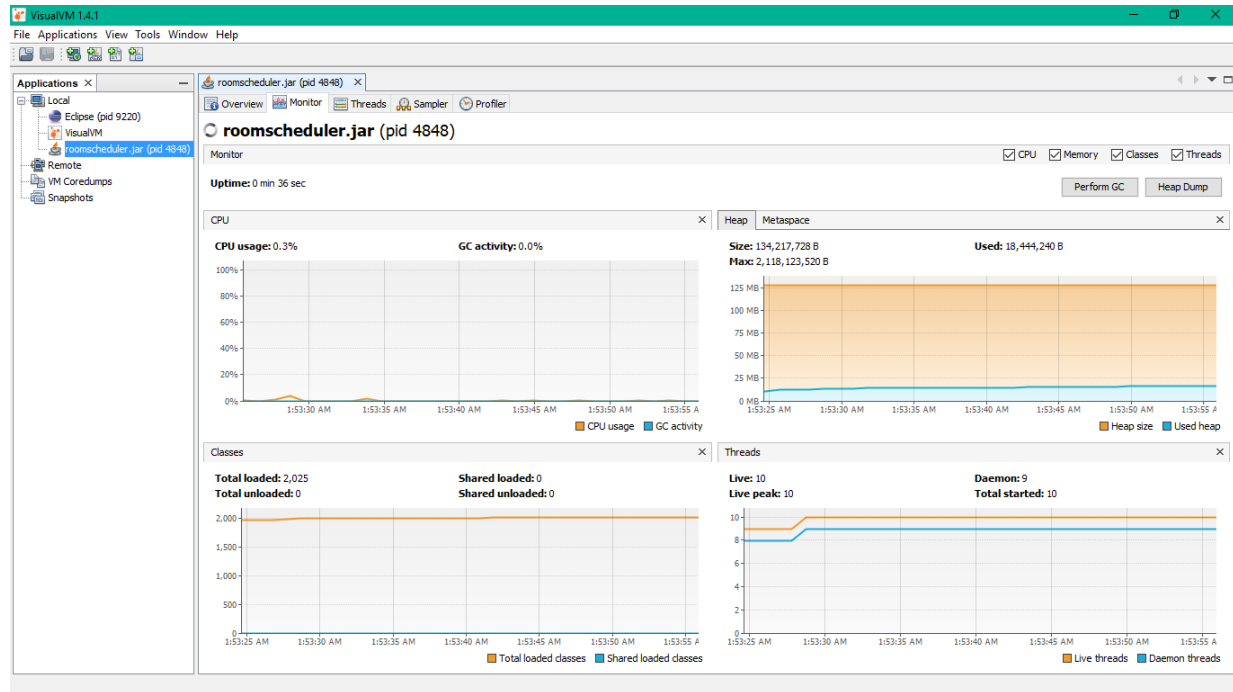
Inside the java.hprof.txt

```
THREAD START (obj=50000198, id = 200001, name="main", group="main")
THREAD END (id = 200001)
TRACE 300070:
    java.io.WinNTFileSystem.canonicalizeWithPrefix0(<Unknown Source>:Unknown line)
    java.io.WinNTFileSystem.canonicalizeWithPrefix(<Unknown Source>:Unknown line)
    java.io.WinNTFileSystem.canonicalize(<Unknown Source>:Unknown line)
    java.io.File.getCanonicalPath(<Unknown Source>:Unknown line)
TRACE 300026:
    java.io.FileOutputStream.close0(<Unknown Source>:Unknown line)
    java.io.FileOutputStream.access$000(<Unknown Source>:Unknown line)
    java.io.FileOutputStream$1.close(<Unknown Source>:Unknown line)
    java.io.FileDescriptor.closeAll(<Unknown Source>:Unknown line)
TRACE 300050:
    sun.misc.Perf.createLong(<Unknown Source>:Unknown line)
    sun.misc.PerfCounter.<init>(<Unknown Source>:Unknown line)
    sun.misc.PerfCounter.newPerfCounter(<Unknown Source>:Unknown line)
    sun.misc.PerfCounter$CoreCounters.<clinit>(<Unknown Source>:Unknown line)
CPU SAMPLES BEGIN (total = 3) Sat May 12 01:32:08 2018
rank  self  accum   count trace method
   1  33.33%  33.33%       1 300070 java.io.WinNTFileSystem.canonicalizeWithPrefix0
   2  33.33%  66.67%       1 300026 java.io.FileOutputStream.close0
   3  33.33% 100.00%       1 300050 sun.misc.Perf.createLong
CPU SAMPLES END
```

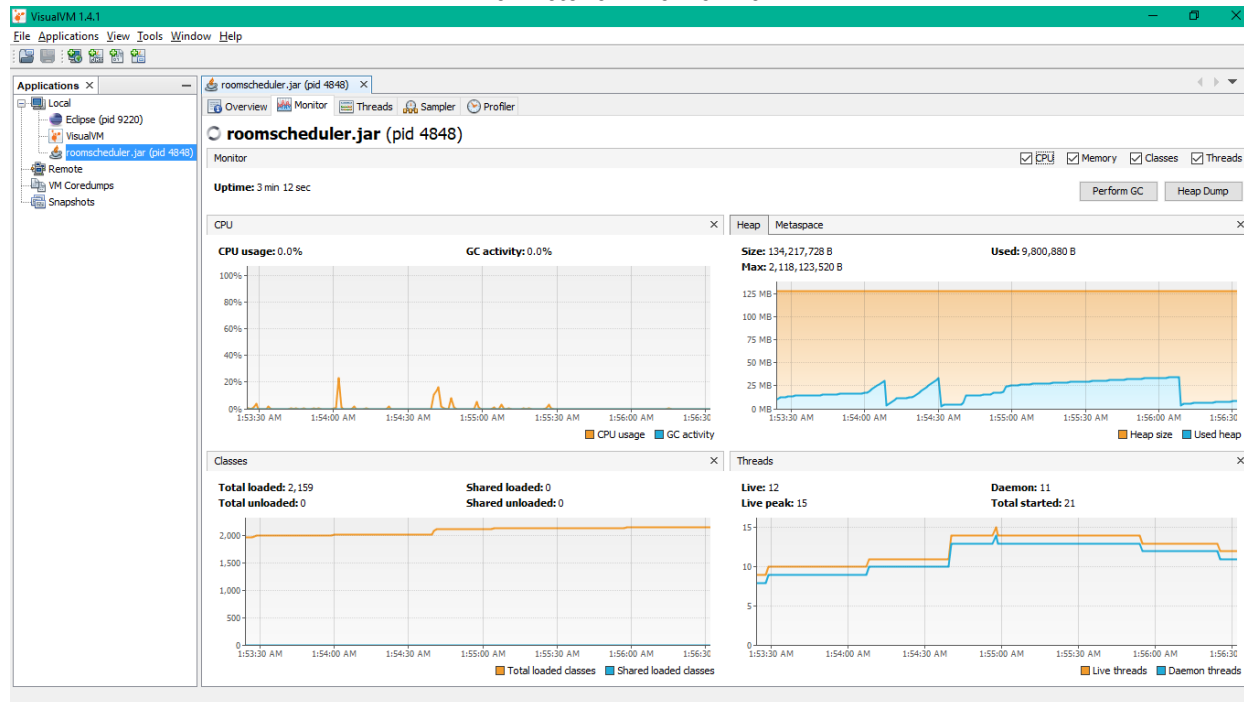
Visual VM results

Before & After execution views in order fashion for Importing and listing rooms multiple times

Before Execution - Monitor View



After Execution - Monitor View



Before Execution – Sampler View

VisualVM 1.4.1

File Applications View Tools Window Help

Applications X

Local

- Eclipse (pid 9220)
- VisualVM
- roomscheduler.jar (pid 4848)

Remote

- VM CoreDumps
- Snapshots

roomscheduler.jar (pid 4848)

Overview Monitor Threads **Sampler** Profiler

roomscheduler.jar (pid 4848)

Sampler

Sample:

Status: CPU sampling in progress

CPU samples Thread CPU time

Results: View: Collected data: Snapshot

Thread Dump

| Name | Total Time | Total Time (CPU) |
|--|-----------------|------------------|
| main | 7,017 ms (100%) | 7,017 ms (100%) |
| roomscheduler.RoomScheduler.main() | 7,017 ms (100%) | 7,017 ms (100%) |
| roomscheduler.RoomScheduler.mainMenu() | 7,017 ms (100%) | 7,017 ms (100%) |
| java.util.Scanner.hasNextInt() | 7,017 ms (100%) | 7,017 ms (100%) |
| Self time | 0.0 ms (0%) | 0.0 ms (0%) |
| Self time | 0.0 ms (0%) | 0.0 ms (0%) |

After Execution – Sampler View

VisualVM 1.4.1

File Applications View Tools Window Help

Applications X

Local

- Eclipse (pid 9220)
- VisualVM
- roomscheduler.jar (pid 4848)

Remote

- VM CoreDumps
- Snapshots

roomscheduler.jar (pid 4848)

Overview Monitor Threads **Profiler**

roomscheduler.jar (pid 4848)

Profiler

Profile:

Status: profiling running (29 methods instrumented)

Profiling results

Results: View: Collected data: Snapshot

CPU settings Memory settings JDBC settings

Profile classes:

<define classes to be profiled>

☒ Track only live objects

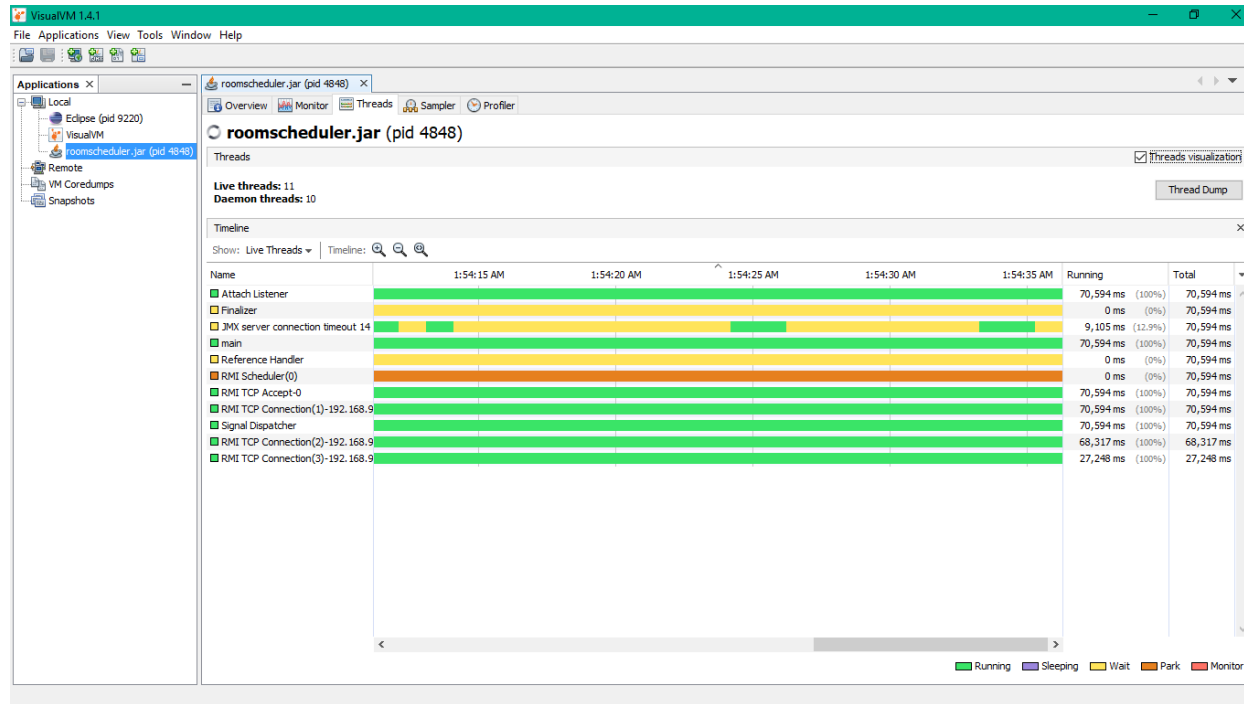
☒ Limit allocations depth: 10

Preset: Default Edit...

| Name | Total Time | Total Time (CPU) | Invocations |
|--|------------------|------------------|-------------|
| main | 8,124 ms (100%) | 0.0 ms (-%) | 8 |
| roomscheduler.RoomScheduler.main() | 9,050 ms (99.2%) | 0.0 ms (-%) | 4 |
| roomscheduler.RoomScheduler.mainMenu() | 73.9 ms (0.8%) | 0.0 ms (-%) | 4 |
| utility.Utility.clearScreen() | 67.5 ms (0.7%) | 0.0 ms (-%) | 4 |
| Self time | 3.49 ms (0%) | 0.0 ms (-%) | 4 |
| roomscheduler.RoomScheduler.mainMenu() | 2.91 ms (0%) | 0.0 ms (-%) | 4 |

| Name | Self Time (CPU) | Total Time (CPU) | Invocations |
|--|-----------------|------------------|-------------|
| roomscheduler.RoomScheduler.mainMenu() | 0.0 ms (-%) | 0.0 ms (-%) | 8 |
| roomscheduler.RoomScheduler.mainMenu() | 0.0 ms (-%) | 0.0 ms (-%) | 4 |
| roomscheduler.RoomScheduler.mainMenu() | 0.0 ms (-%) | 0.0 ms (-%) | 4 |
| utility.Utility.clearScreen() | 0.0 ms (-%) | 0.0 ms (-%) | 8 |
| utility.Utility.sleepFor(int) | 0.0 ms (-%) | 0.0 ms (-%) | 4 |
| utility.Utility.getCurrentDateTime() | 0.0 ms (-%) | 0.0 ms (-%) | 8 |

Before Execution – Threads View



After Execution – Threads View

