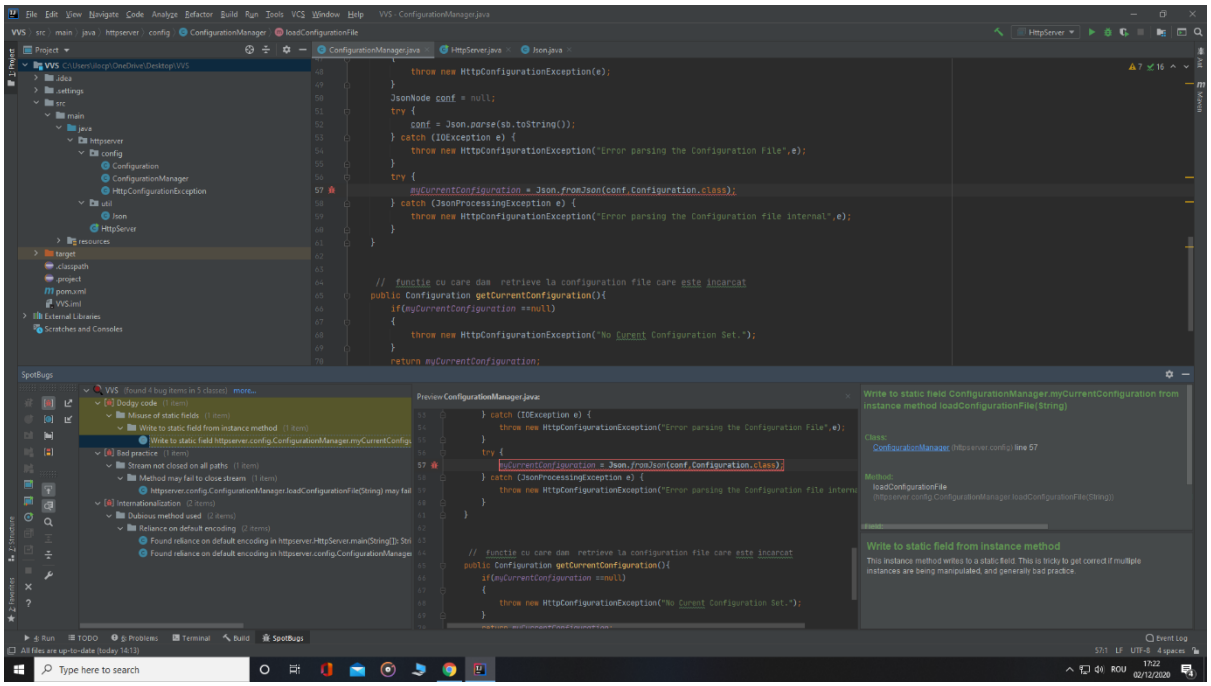
Analiza statica

Am folosit extensia SpotBugs in Intellij in detrimentul extensiei FindBugs deoarece aceasta am gasit-o ca fiind compatibila.

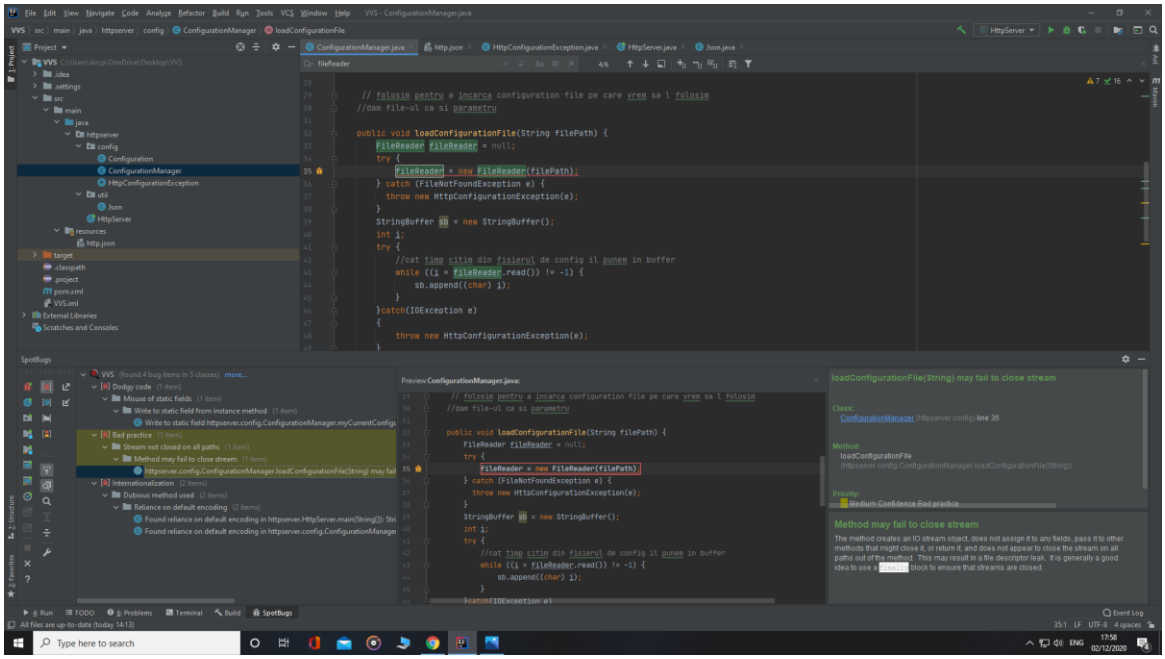
Mai jos sunt 2 cazuri de probleme semnalate in urma rularii programului.



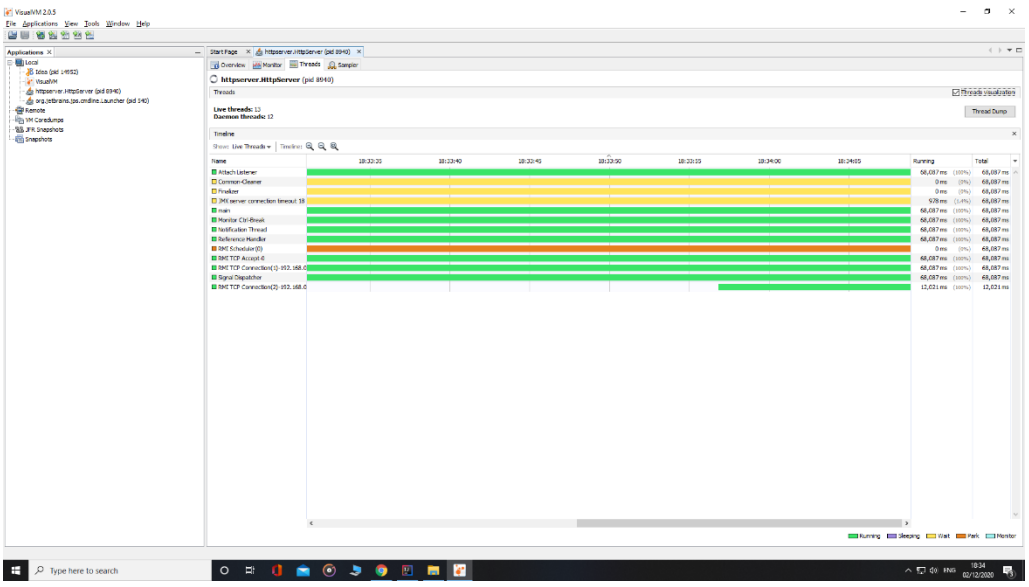
In cadrul primei probleme SpotBugs evidentiaza folosirea improprie a unei variabile statice care este folosita intr-o metoda nestatica. Este vorba de variabila myCurrentConfiguration caruia i se atribuie configuratia citita din fisierul Json, configuratie ce specifica host-ul si webroot-ul.

O varianta de inlaturare a acestei probleme ar fi schimbarea metodei in care folosim variabila intr-o metoda statica. Este vorba despre metoda loadConfigurationFile.

A 2-a problema semnalata apare din cauza ca stream-ul fileReader nu este inchis in momentul in care comunicatia nu mai are loc. Putem rezolva aceasta problema prin adaugarea la blocul try catch a unui bloc finally in care sa inchidem stream-ul.



Analiza dinamica



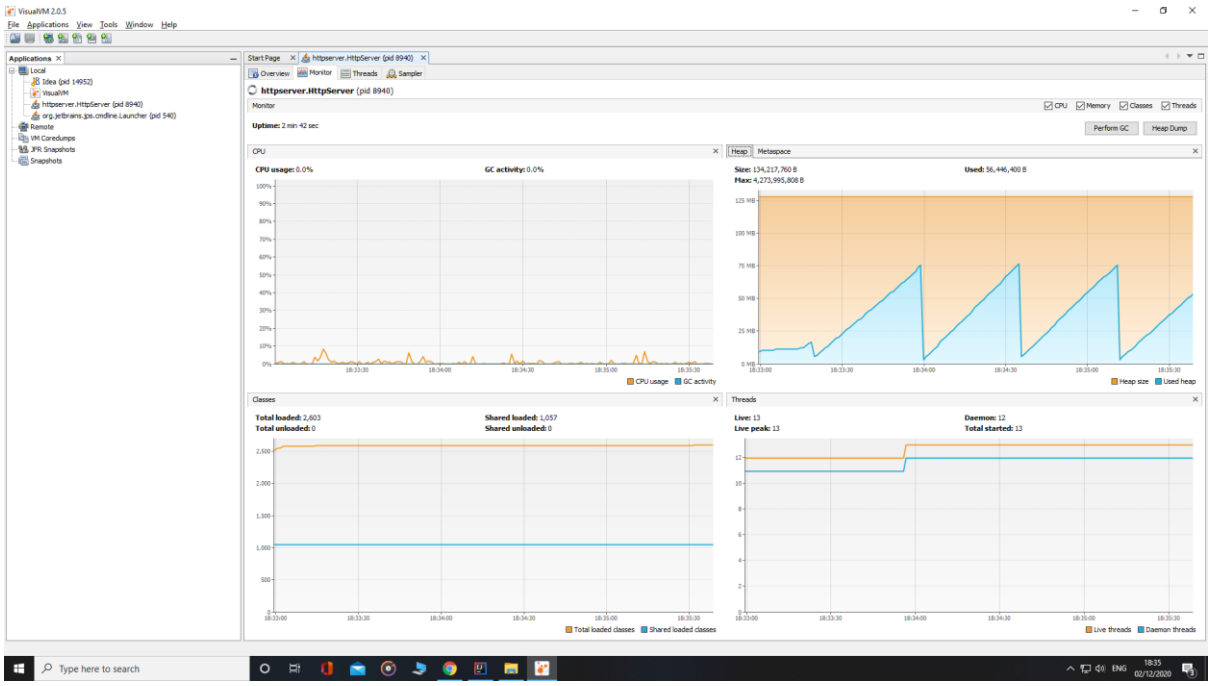
In tab-ul monitor avem descries utilizarea resurselor in felul urmator:

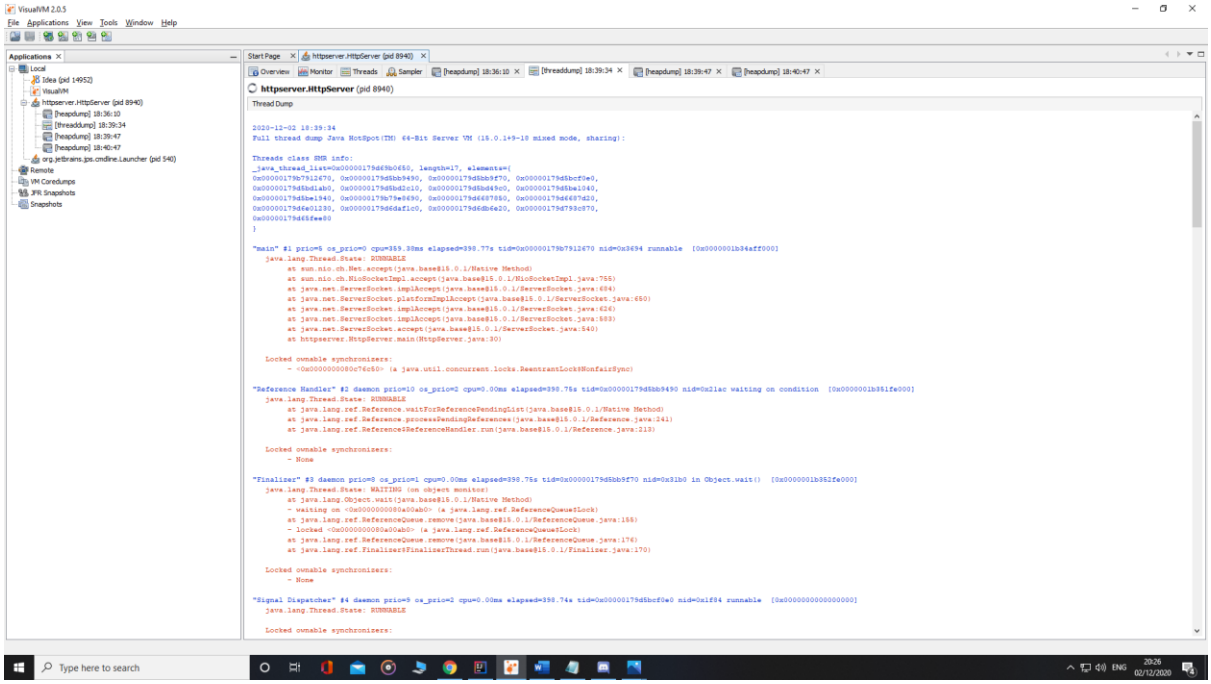
-procentul utilizat de catre processor pentru rularea aplicatiei,

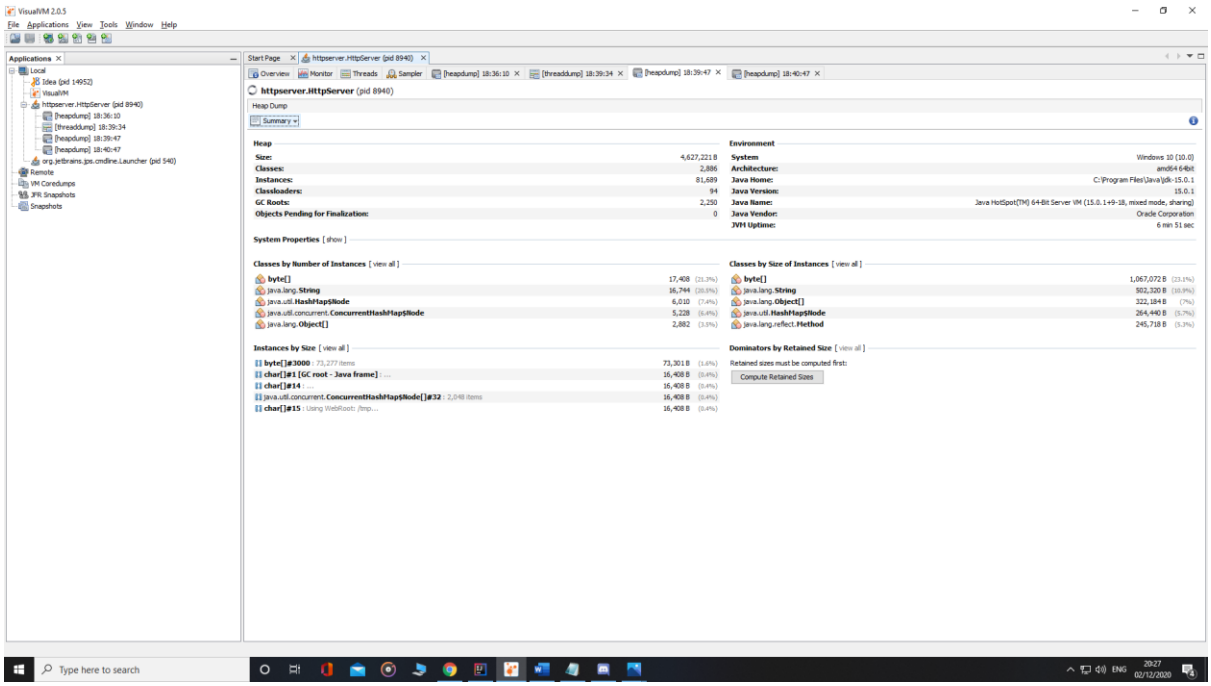
-memoria heap ocupata dar si dimensiunea de metadata folosita care apare in metaspace

-numarul de clase incarcare care sunt folosite

-numarul de thread-uri care ruleaza




VisualVM 2.0.5
File Applications View Tools Window Help
Applications
Local
Idea (pid 14952)
VisualVM
httpserver.HttpServer (pid 8940)
[heapdump] 18:36:10
[threaddump] 18:39:34
[heapdump] 18:39:47
[heapdump] 18:40:47
org.jetbrains.jps.cmdline.Launcher (pid 540)
Remote
VM Coredumps
JFR Snapshots
Snapshots
Start Page
httpserver.HttpServer (pid 8940)
Overview
Monitor
Threads
Sampler
[heapdump] 18:36:10
[threaddump] 18:39:34
[heapdump] 18:39:47
[heapdump] 18:40:47
httpserver.HttpServer (pid 8940)
Thread Dump
2020-12-02 18:39:34
Full thread dump Java HotSpot(TM) 64-Bit Server VM (15.0.1+9-18 mixed mode, sharing):
Threads class SMR info:
_java_thread_list=0x00000179d69b0650, length=17, elements={
0x00000179b7912470, 0x00000179d5bb9490, 0x00000179d5bbcf0e0,
0x00000179d5bd1ab0, 0x00000179d5bd2c10, 0x00000179d5bd49c0,
0x00000179d5be1940, 0x00000179b75e8490, 0x00000179d6697850,
0x00000179d6697d20,
0x00000179d6e01230, 0x00000179d6dafic0, 0x00000179d6db4e20,
0x00000179d793c870,
0x00000179d65fee80
}
"main" #1 prio=5 os_prio=0 cpu=359.38ms elapsed=398.77s tid=0x00000179b7912670 nid=0x3694 runnable [0x0000001b34aff000]
java.lang.Thread.State: RUNNABLE
at sun.nio.ch.Net.accept(java.base@15.0.1/Native Method)
at sun.nio.ch.NioSocketImpl.accept(java.base@15.0.1/NioSocketImpl.java:755)
at java.net.ServerSocket.implAccept(java.base@15.0.1/ServerSocket.java:684)
at java.net.ServerSocket.platformImplAccept(java.base@15.0.1/ServerSocket.java:650)
at java.net.ServerSocket.implAccept(java.base@15.0.1/ServerSocket.java:626)
at java.net.ServerSocket.implAccept(java.base@15.0.1/ServerSocket.java:603)
at java.net.ServerSocket.accept(java.base@15.0.1/ServerSocket.java:540)
at httpserver.HttpServer.main(HttpServer.java:30)
Locked ownable synchronizers:
- <0x0000000080c76c50> (a java.util.concurrent.locks.ReentrantLock$NonfairSync)
"Reference Handler" #2 daemon prio=10 os_prio=2 cpu=0.00ms elapsed=398.76s tid=0x00000179d5bb9490 nid=0x2lac waiting on condition [0x0000001b351fe000]
java.lang.Thread.State: RUNNABLE
at java.lang.ref.Reference.waitForReferencePendingList(java.base@15.0.1/Native Method)
at java.lang.ref.Reference.processPendingReferences(java.base@15.0.1/Reference.java:241)
at java.lang.ref.Reference$ReferenceHandler.run(java.base@15.0.1/Reference.java:213)
Locked ownable synchronizers:
- None
"Finalizer" #3 daemon prio=8 os_prio=1 cpu=0.00ms elapsed=398.75s tid=0x00000179d5bb9f70 nid=0x31b0 in Object.wait() [0x0000001b352fe000]
java.lang.Thread.State: WAITING (on object monitor)
at java.lang.Object.wait(java.base@15.0.1/Native Method)
- waiting on <0x00000000080a80ab0> (a java.lang.ref.ReferenceQueue$Lock)
at java.lang.ref.ReferenceQueue.remove(java.base@15.0.1/ReferenceQueue.java:155)
- locked <0x00000000080a80ab0> (a java.lang.ref.ReferenceQueue$Lock)
at java.lang.ref.ReferenceQueue.remove(java.base@15.0.1/ReferenceQueue.java:174)
at java.lang.ref.Finalizer$FinalizerThread.run(java.base@15.0.1/Finalizer.java:170)
Locked ownable synchronizers:
- None
"Signal Dispatcher" #4 daemon prio=9 os_prio=2 cpu=0.00ms elapsed=398.74s tid=0x00000179d5bcf0e0 nid=0x1f84 runnable [0x0000000000000000]
java.lang.Thread.State: RUNNABLE
Locked ownable synchronizers:
Type here to search
ENG
20:26
02/12/2020


VisualVM 2.0.5
File Applications View Tools Window Help
Applications
Local
Idea (pid 14952)
VisualVM
httpserver.HttpServer (pid 8940)
[heapdump] 18:36:10
[threaddump] 18:39:34
[heapdump] 18:39:47
[heapdump] 18:40:47
org.jetbrains.jps.cmdline.Launcher (pid 540)
Remote
VM Coredumps
JFR Snapshots
Snapshots
Start Page
httpserver.HttpServer (pid 8940)
Overview
Monitor
Threads
Sampler
[heapdump] 18:36:10
[threaddump] 18:39:34
[heapdump] 18:39:47
[heapdump] 18:40:47
httpserver.HttpServer (pid 8940)
Heap Dump
Summary
Heap
Size: 4,627,221 B
Classes: 2,886
Instances: 81,689
Classloaders: 94
GC Roots: 2,250
Objects Pending for Finalization: 0
Environment
System Windows 10 (10.0)
Architecture: amd64 64bit
Java Home: C:\Program Files\Java\jdk-15.0.1
Java Version: 15.0.1
Java Name: Java HotSpot(TM) 64-Bit Server VM (15.0.1+9-18, mixed mode, sharing)
Java Vendor: Oracle Corporation
JVM Uptime: 6 min 51 sec
System Properties [show]
Classes by Number of Instances [view all]
byte[] 17,408 (21.3%)
java.lang.String 16,744 (20.5%)
java.util.HashMap$Node 6,010 (7.4%)
java.util.concurrent.ConcurrentHashMap$Node 5,228 (6.4%)
java.lang.Object[] 2,882 (3.5%)
Classes by Size of Instances [view all]
byte[] 1,067,072 B (23.1%)
java.lang.String 502,320 B (10.9%)
java.lang.Object[] 322,184 B (7%)
java.util.HashMap$Node 264,440 B (5.7%)
java.lang.reflect.Method 245,718 B (5.3%)
Instances by Size [view all]
byte[]#3000 : 73,277 items 73,301 B (1.6%)
char[]#1 [GC root - Java frame] : ... 16,408 B (0.4%)
char[]#14 : ... 16,408 B (0.4%)
java.util.concurrent.ConcurrentHashMap$Node[]#32 : 2,048 items 16,408 B (0.4%)
char[]#15 : Using WebRoot: /tmp... 16,408 B (0.4%)
Dominators by Retained Size [view all]
Retained sizes must be computed first:
Compute Retained Sizes
Type here to search
ENG
20:27
02/12/2020