Object Oriented Languages and Environments

CS 474 Homework 1

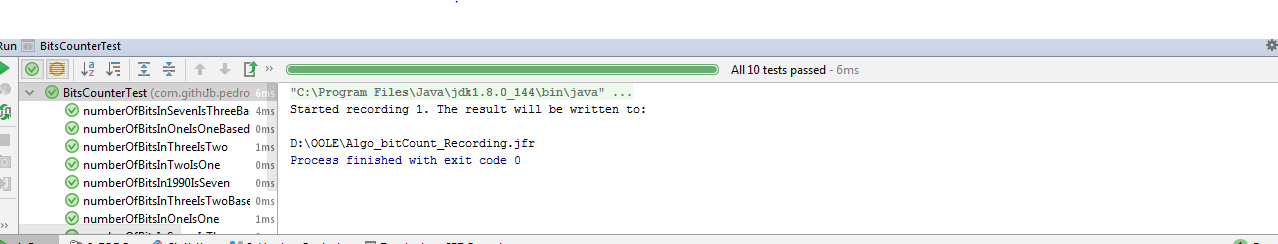
Tathagata Ganguly

|  |  |
| --- | --- |
| 1 | Java Mission Control |
| 2 | jcmd utility |
| 3 | Java VisualVM |
| 4 | JConsole utility |
| 5 | jmap utility |
| 6 | jps utility |
| 7 | visualgc utility |

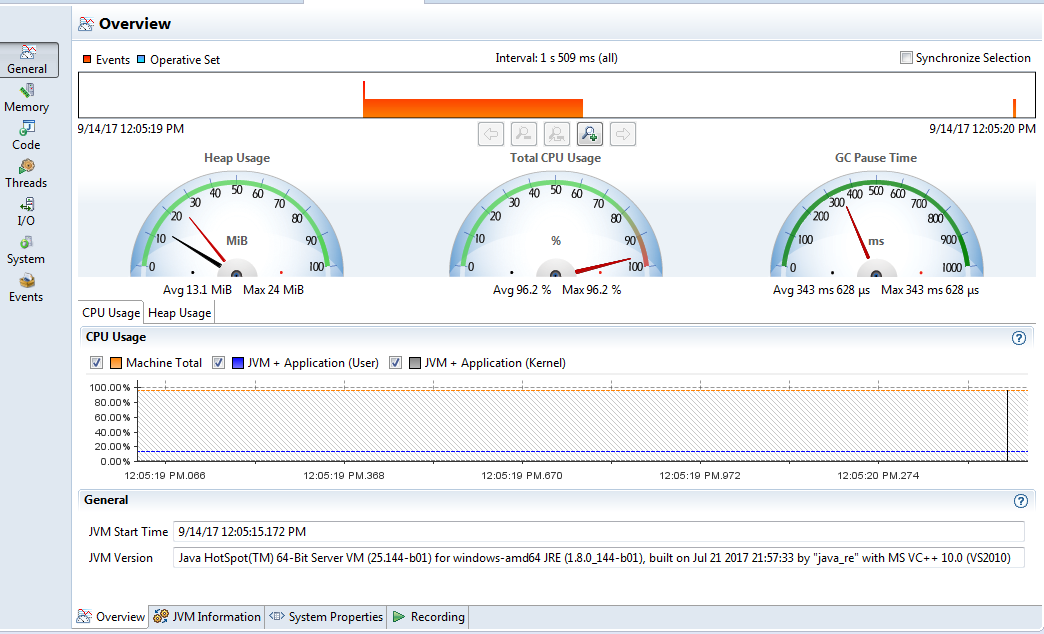
**1)Java Mission Control**

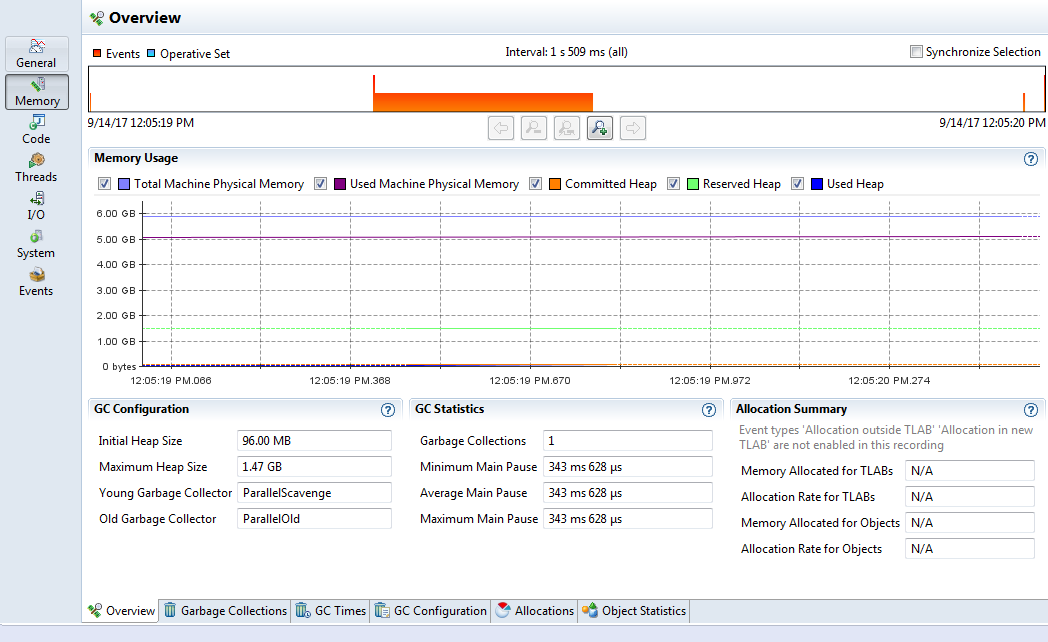
Step 1: Run JMC in your terminal

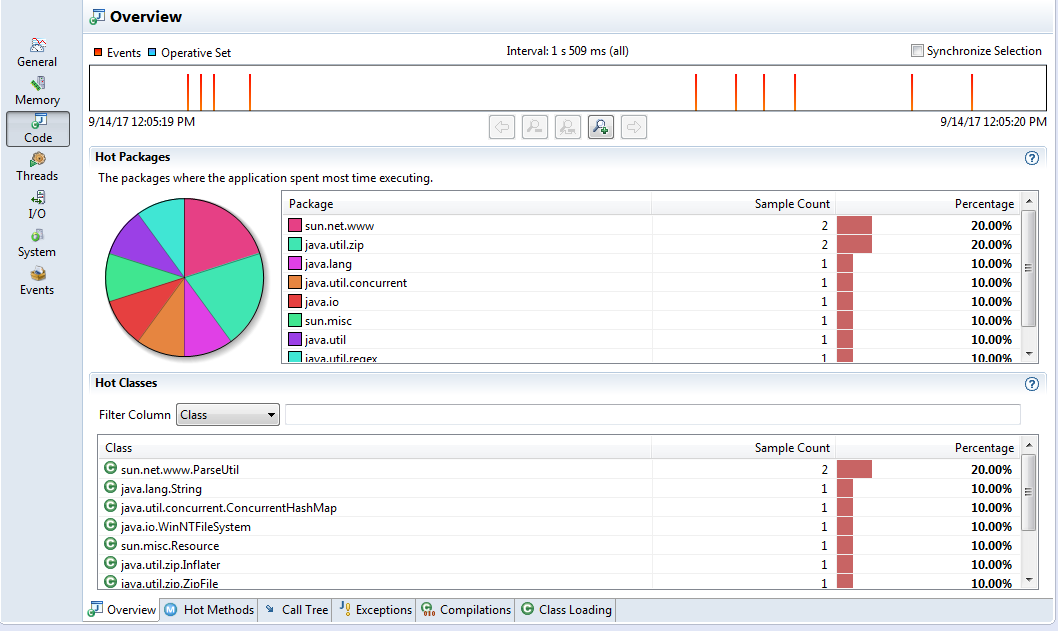
Step 2:Inside the project in Intellij, RUN -> EDIT CONFIGURATIONS ->  
VM OPTIONS : -XX:+UnlockCommercialFeatures -XX:+FlightRecorder -XX:StartFlightRecording=duration=2m,filename= File\_name (D:\OOLE\Algo\_bitCount\_Recording.jfr for me)

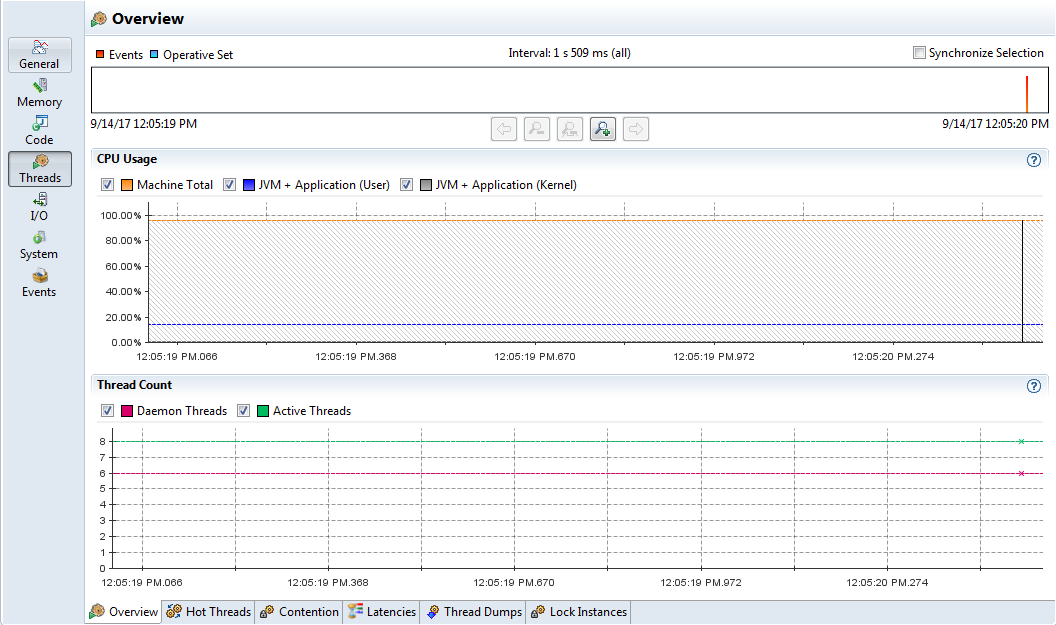


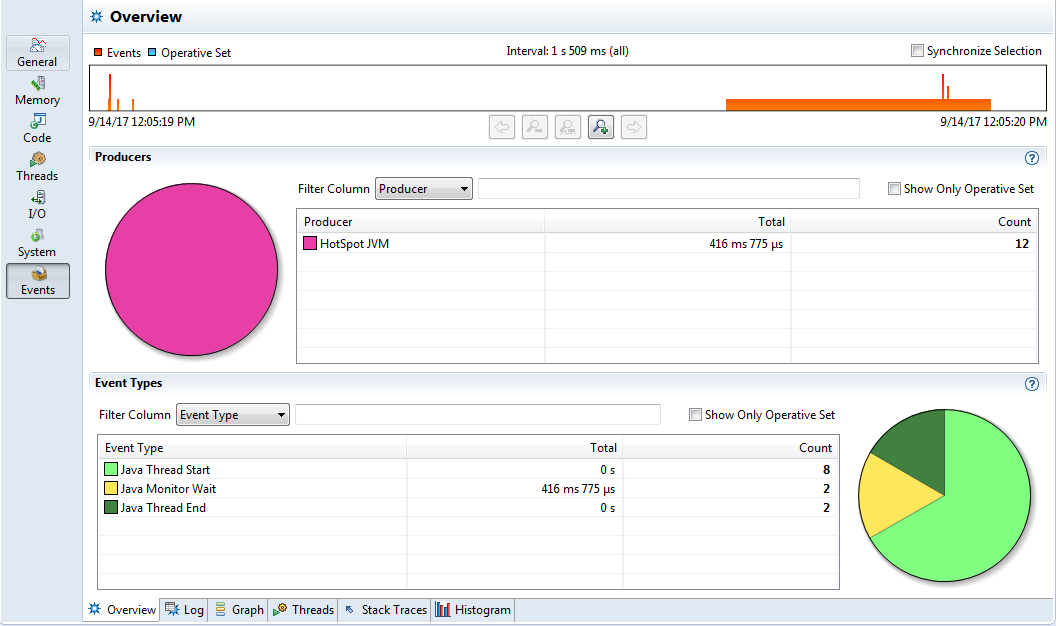
The recording can be viewed at JMC.





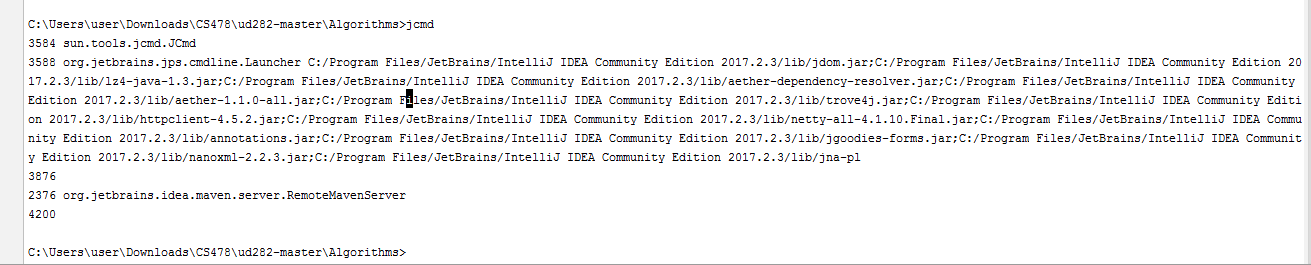




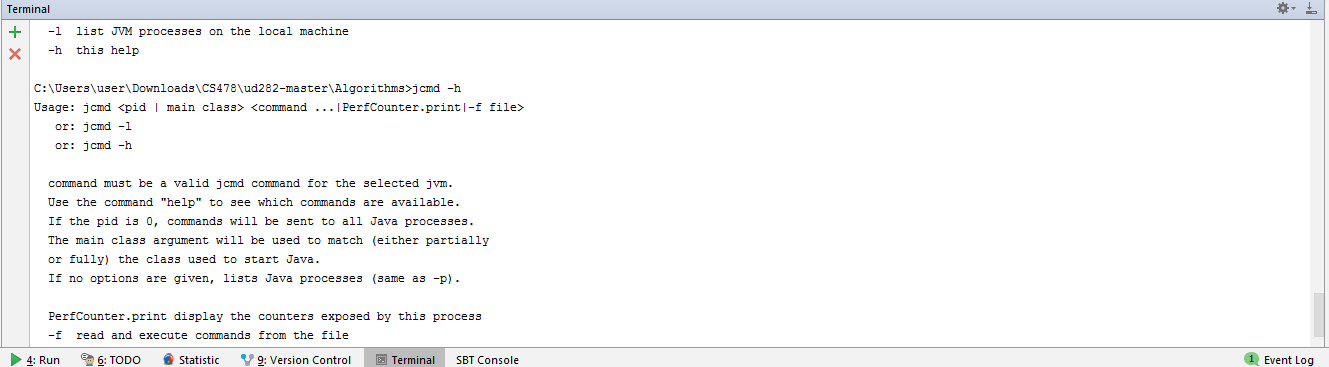


**2 ) jcmd Utility**

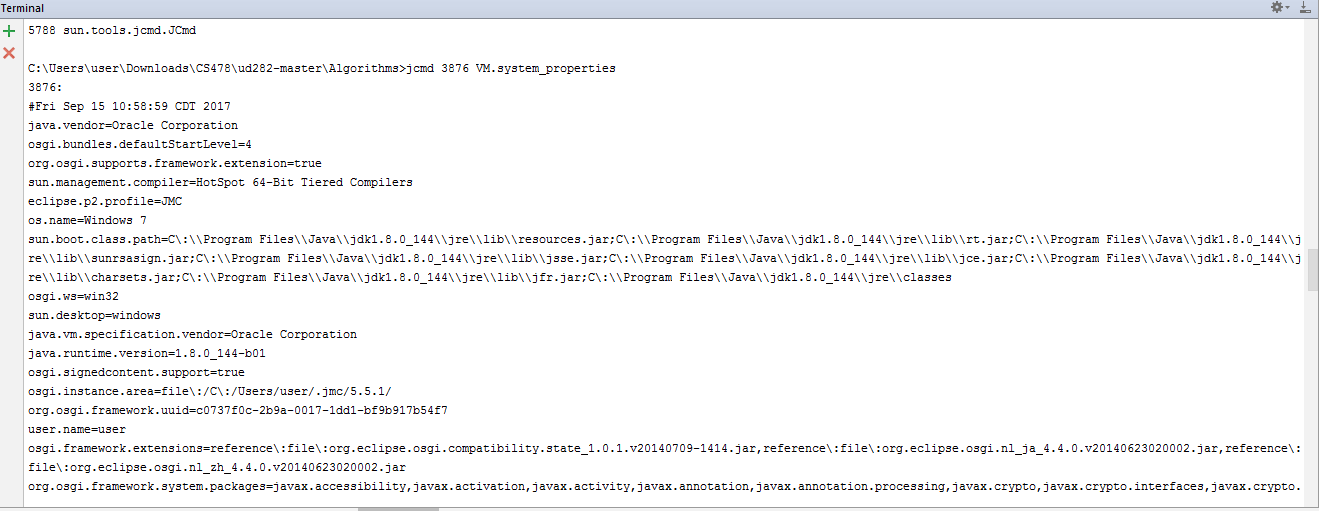
i. Run 'jcmd' /'jcmd -l'in the terminal

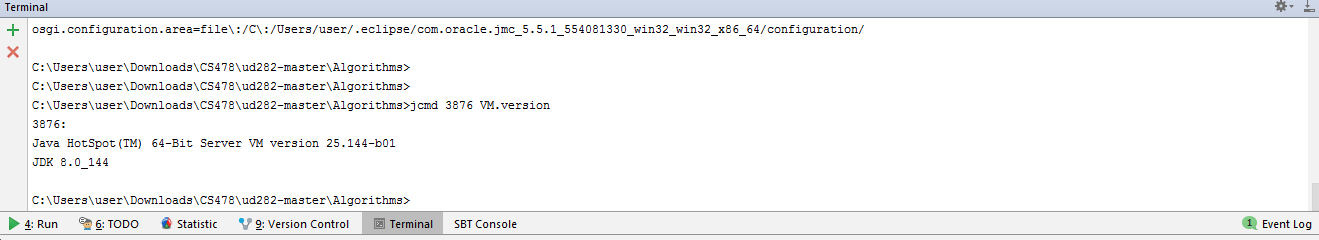


ii.Run 'jcmd -h' for help



iii.Run 'jcmd <pid> VM.system\_properties' to view all the system properties of VM

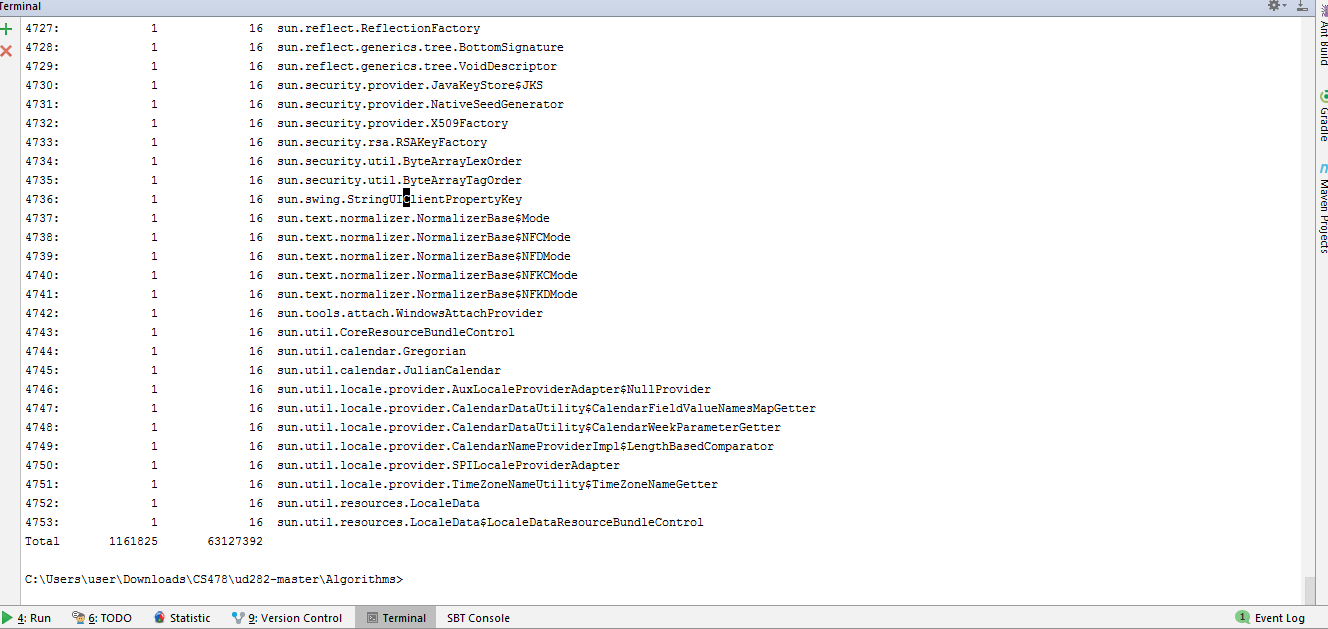


iv. Run 'jcmd <pid> VM.version' to view all the version of VM

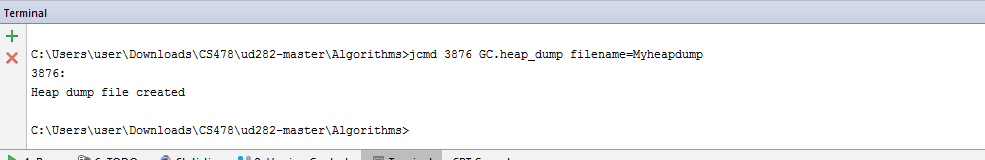
iv. Run 'jcmd <pid> VM.uptime' to view all the uptime of VM



v. Run 'jcmd<pid> GC class\_histogram to create a class histogram

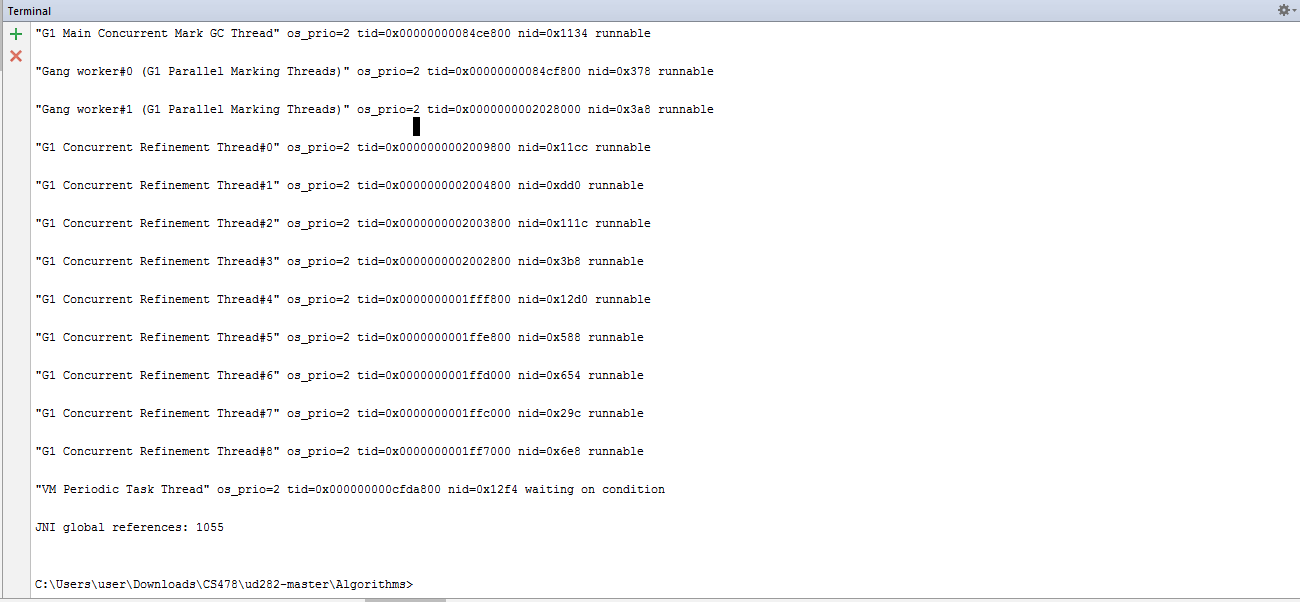


vi. Run 'jcmd <pid> GC.heap\_dump filename=Myheapdump to create a heap dump file.

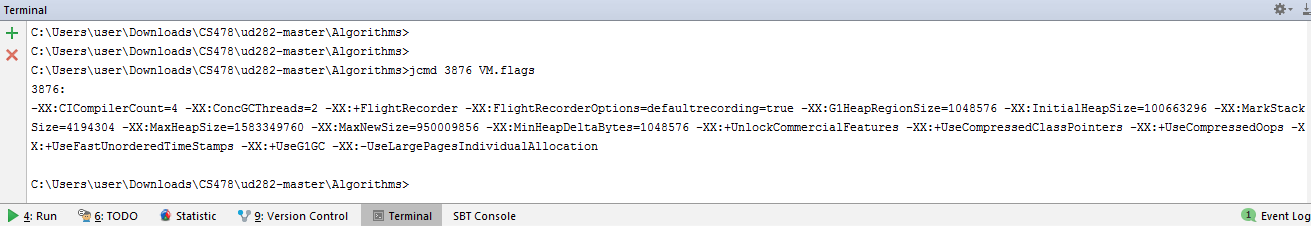


vii.Run jcmd <process id/main class> GC.class\_histogram filename=Myheaphistogram to create a heap histogram file.

viii. Run jcmd <pid> Thread.print to print all threads with stack traces



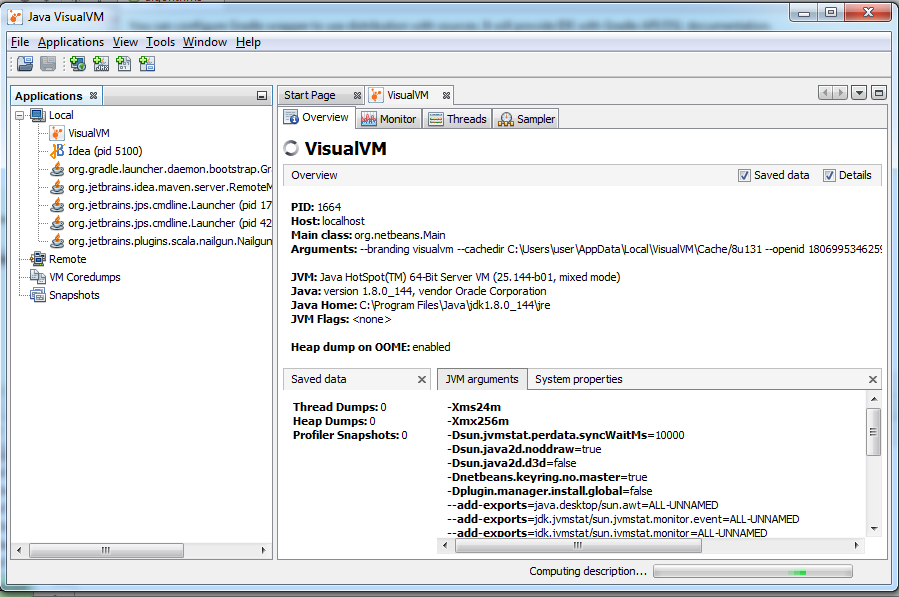
ix. Run jcmd <pid> VM.flags to print all the flags used for the VM

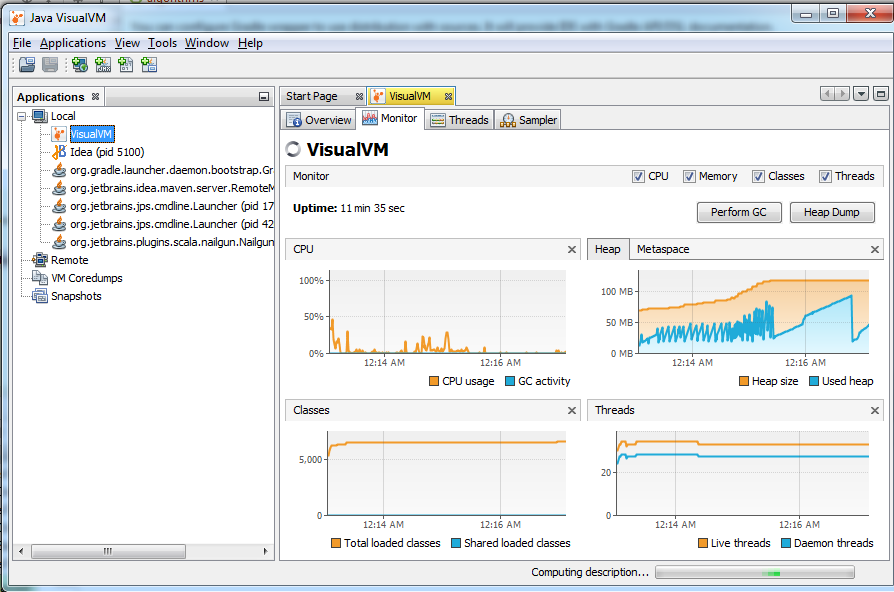


**3)JAVA VISUAL VM**

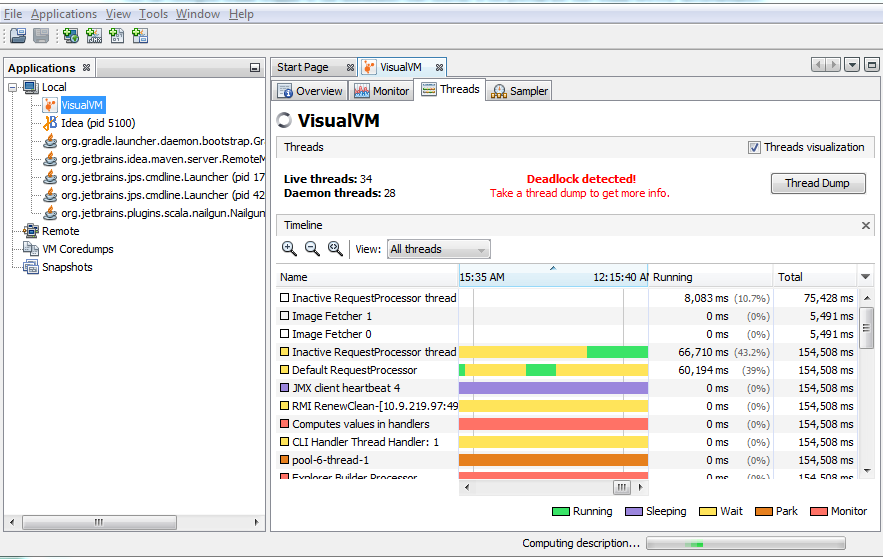
Step 1: Open settings ->Install Visual VM plugin and restart Intellij

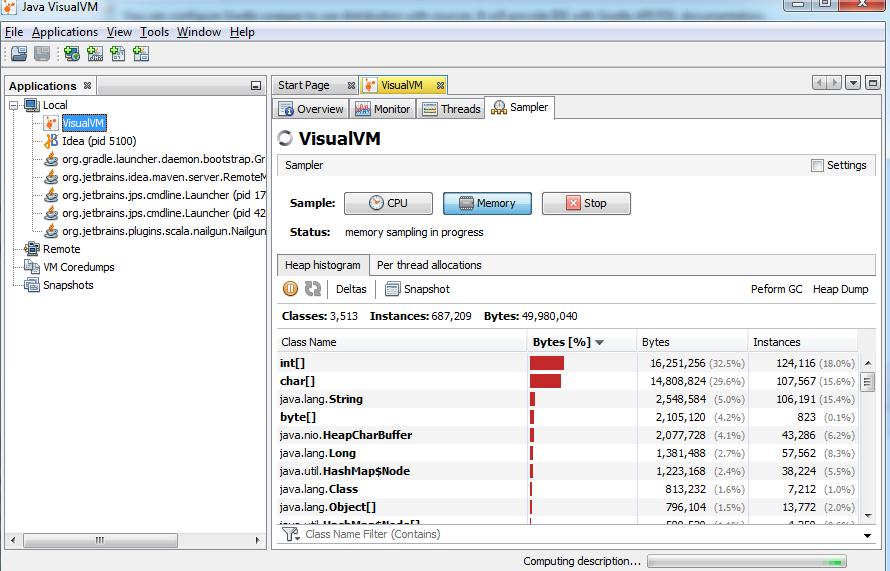
Step 2: Open settings-> Open Visual VM launcher ->Run program

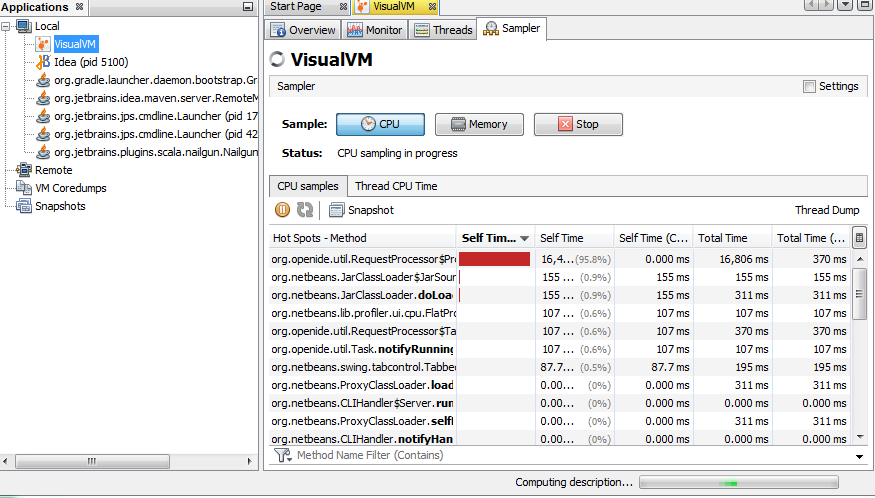
**VISUAL VM OVERVIEW**

****

**MONITOR**

**THREADS**

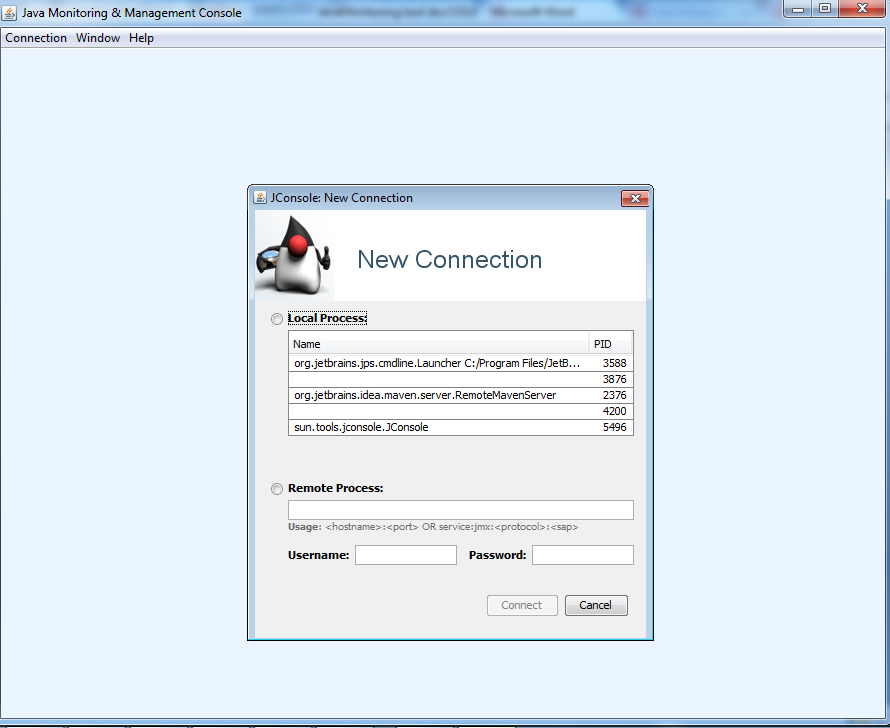
**SAMPLER MEMORY**

**SAMPLER CPU**

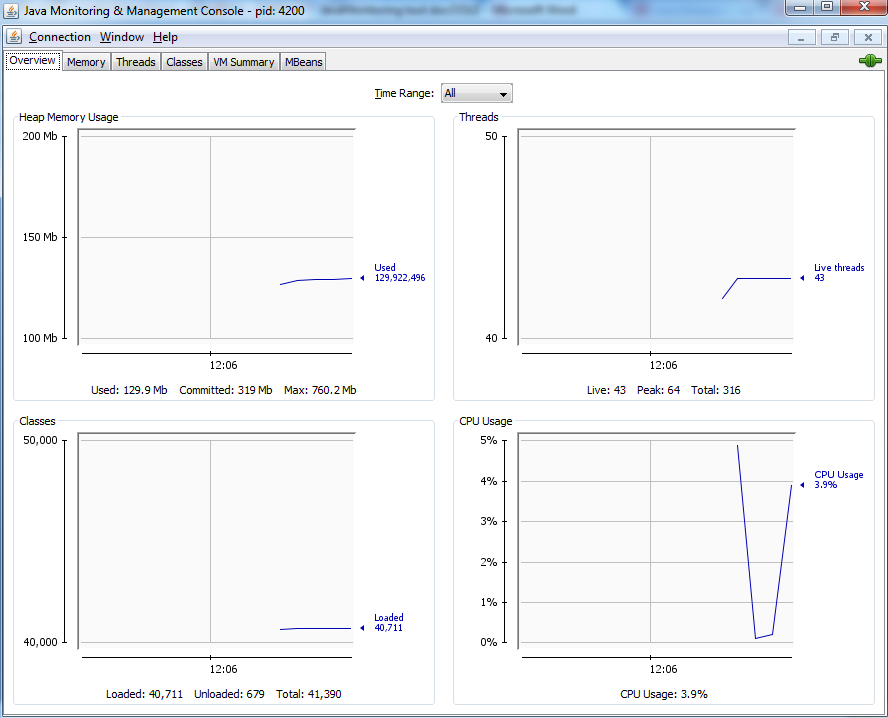
**4)JConsole Utility**

Run 'JConsole in the terminal

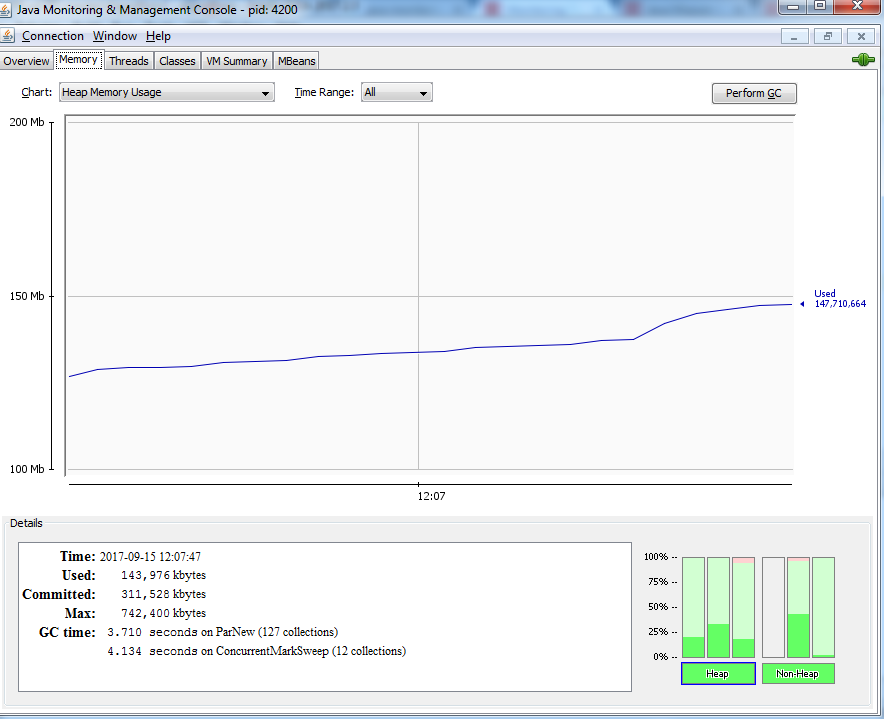
Java Monitoring and Management console pops up



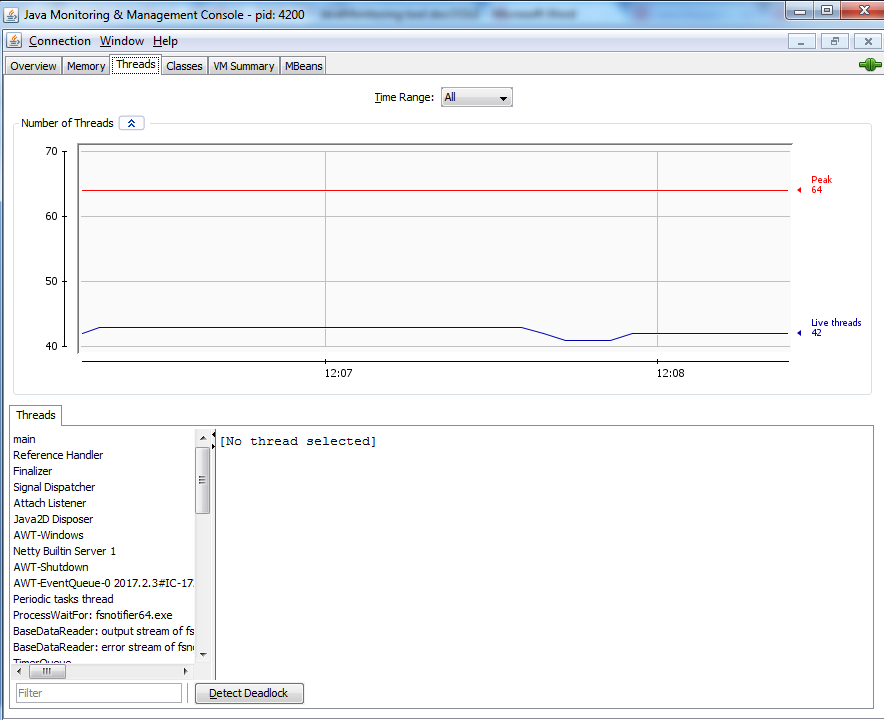
Choose any free PID to connect



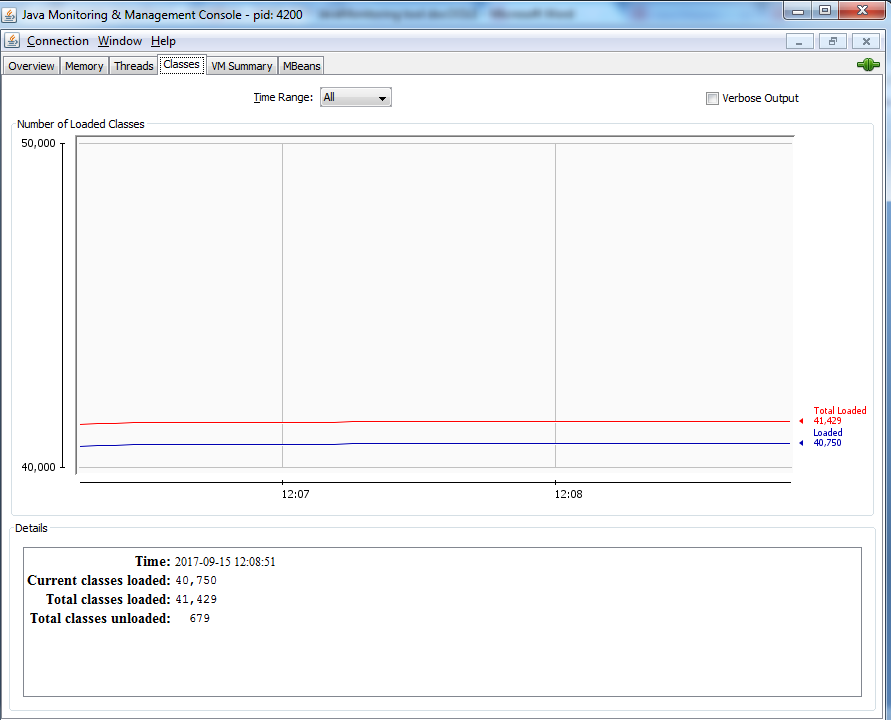
**Overview**

****

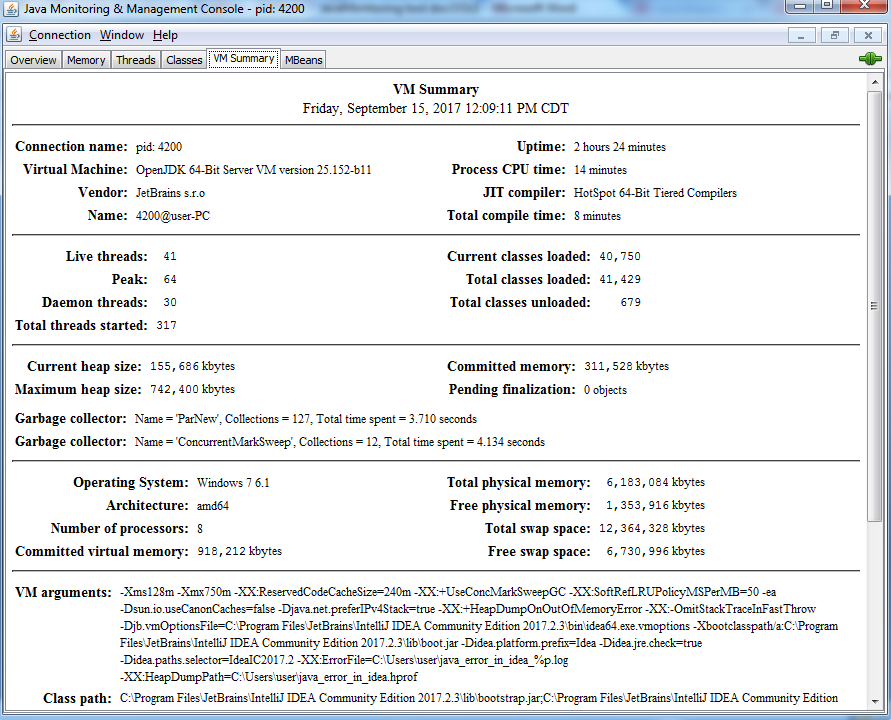
**MEMORY**

****

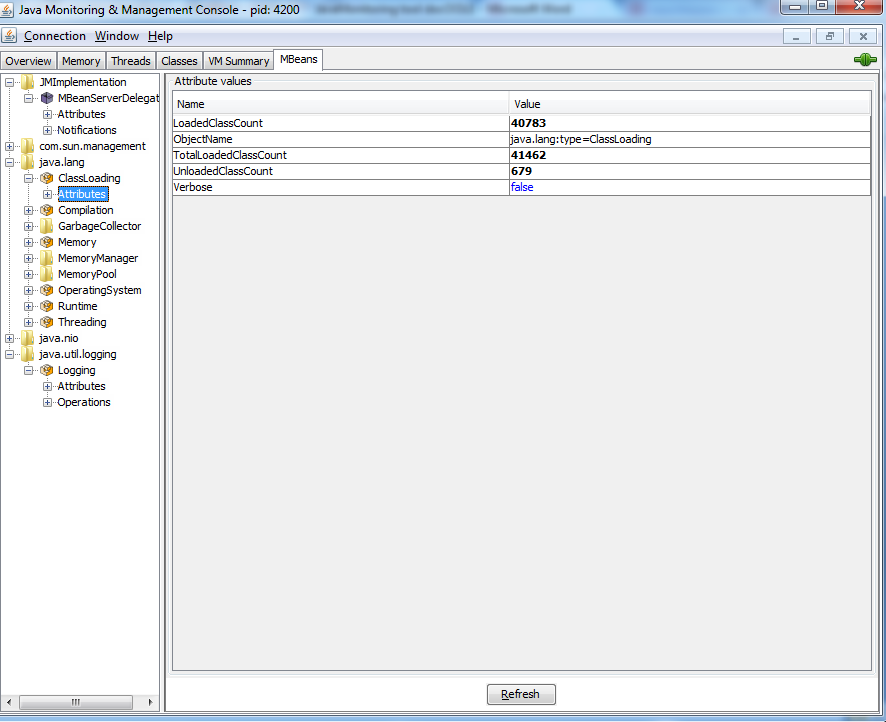
**THREADS**

****

**CLASSES**

****

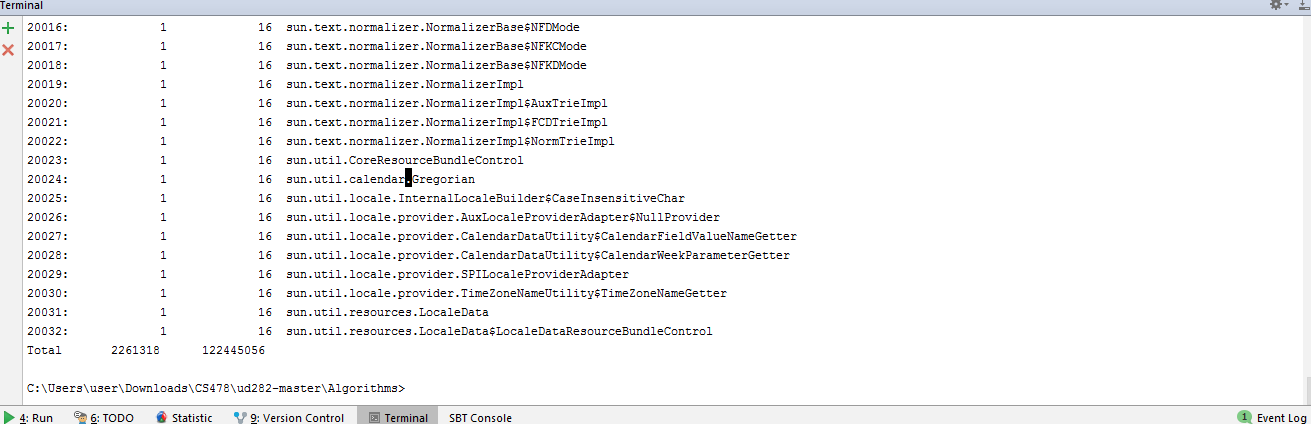
**VM SUMMARY**



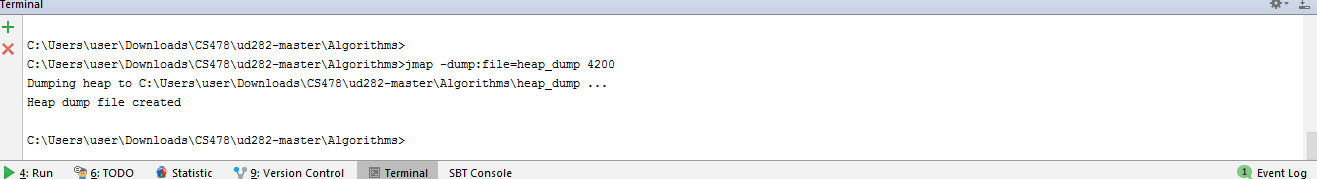
**You can also use Mbeans for further analysing the data.**

**5)jmap Utility**

jmap -histo <pid> to create a heap histogram



jmap -dump:file=<file> <pid> to create a heap dump



**6)jps Utility**

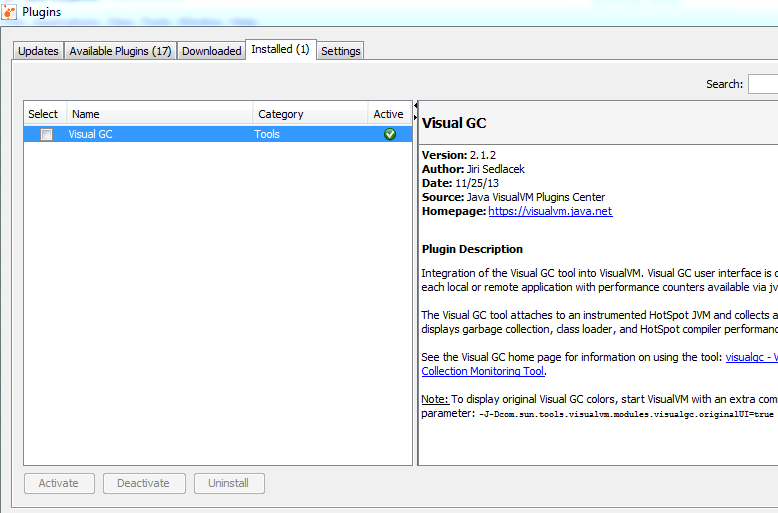
Run 'jps' in terminal to  list the instrumented Java HotSpot VMs on the target system.



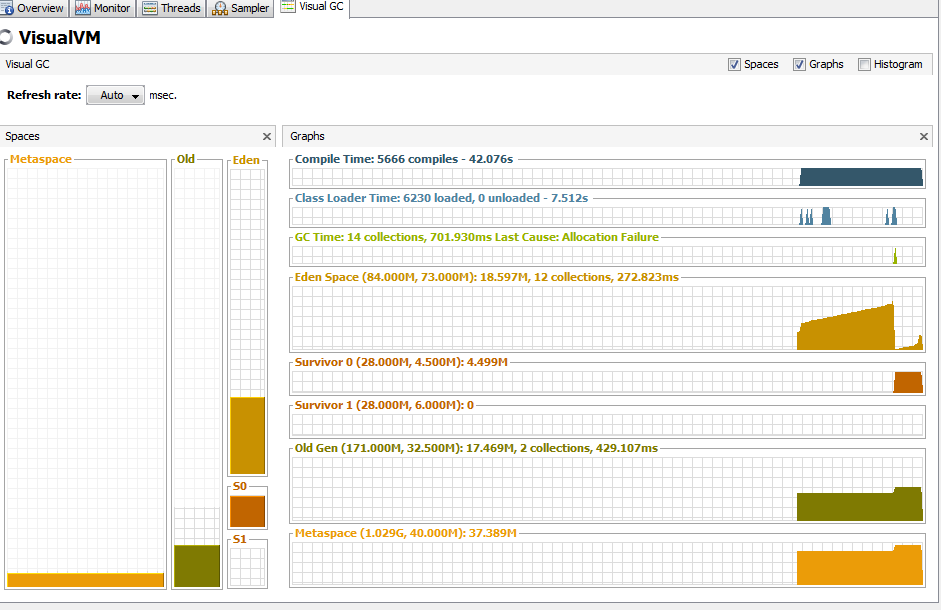
**7)Visual GC**

Run Java VisualVM, and click "Tools > Plugins"

Click the "Available Plugins" tab and install "Visual GC" from the list



Run the program and use visualVM to monitor it.



Citations

https://docs.oracle.com/javase/8/docs/technotes/guides/troubleshoot/tooldescr006.html#BABEHABG

https://www.javacodegeeks.com/2016/03/jcmd-one-jdk-command-line-toolrule.html

http://www.herongyang.com/Java-Tools/jvisualvm-Install-Visual-GC-Plugin.html