What is Garbage Collection log, Thread dump & Heap dump?

By Ram Lakshmanan

	GC Log	Thread dump	Heap dump
What is it?	GC Log contains garbage collection events related information. It will indicate how many GC events ran, what type of GC events they are (i.e. Young GC or Full GC), how long did each GC event pause the application, how much objects did each GC event reclaim.	Thread dump is snapshot of all threads running in the application at point in time. It contains all the information about each thread in the application such as: thread state, thread Id, native Id, thread name, stack trace, priority.	Heap dump is a snapshot of your application's memory in a point in time. It contains information such as what are the objects in memory, what values do they carry, what is their size, what other objects do they reference.
How does it look?	Sample garbage collection log file can be found here.	Sample thread dump can be found here.	Sample heap dump can be found here. (Note: It is going to be in binary format. So you actually can't read it).
Where it is used?	Garbage collection logs are used to optimize the GC pause times, it's used to identify optimal memory size for your application, it's also used to trouble -shoot memory related problems.	Thread dumps are primarily used for troubleshooting production problems such as CPU spikes, unresponsiveness in the application, poor response time, hung threads, high memory consumption.	Heap dumps are primarily used for troubleshooting memory related, OutOfMemoryError problems.
How to generate it?	You can generate garbage collection log, by passing following JVM arguments: For Java versions until 8: **-XX:+PrintGCDetails** **-XX:+PrintGCDateStamps** **-Xloggc:<file-path>** For Java version starting from 9: **-Xlog:gc*:file=<file-path>** where file-path: is the location where Garbage Collection log file will be written	Thread dumps can be captured using 8 different options. Most common option to take thread dump is to use the 'jstack' tool. jstack tool is shipped in JDK_HOME\bin folder. Here is the command that you need to issue to capture thread dump: **jstack -l <pid> > <file-path>** where pid: is the Process Id of the application, whose thread dump should be captured. file-path: is the file path where thread dump will be written in to.	Heap dump can be captured using 7 different options. Most common option to take heap dump is to use the 'jmap' tool. jmap tool is shipped in JDK_HOME\bin folder. Here is the command that you need to issue to capture: **jmap -dump:format=b, file=<file-path> <pid>** where pid: is the Java Process Id, whose heap dump should be captured. file-path: is the file path where heap dump will be written in to.
How to understand it?	Here is a video tutorial, which attempts to help you to understand the GC log file format.	Here is a video tutorial, that gives good detailed overview on how to understand the thread dumps.	Heap dump files are in binary format and tend to be large in size. Besides that, their format heavily lacks documentation. Thus, you have to use the heap dump analysis tools (given in next question) to analyze and understand them.
What tools can be used for analysis?	Gceasy IBM GC & Memory visualizer HP Jmeter Google Garbage Cat	fastThread Samurai IBM Thread & Monitor analyzer Visual VM	Eclipse MAT HeapHero JVisualVM