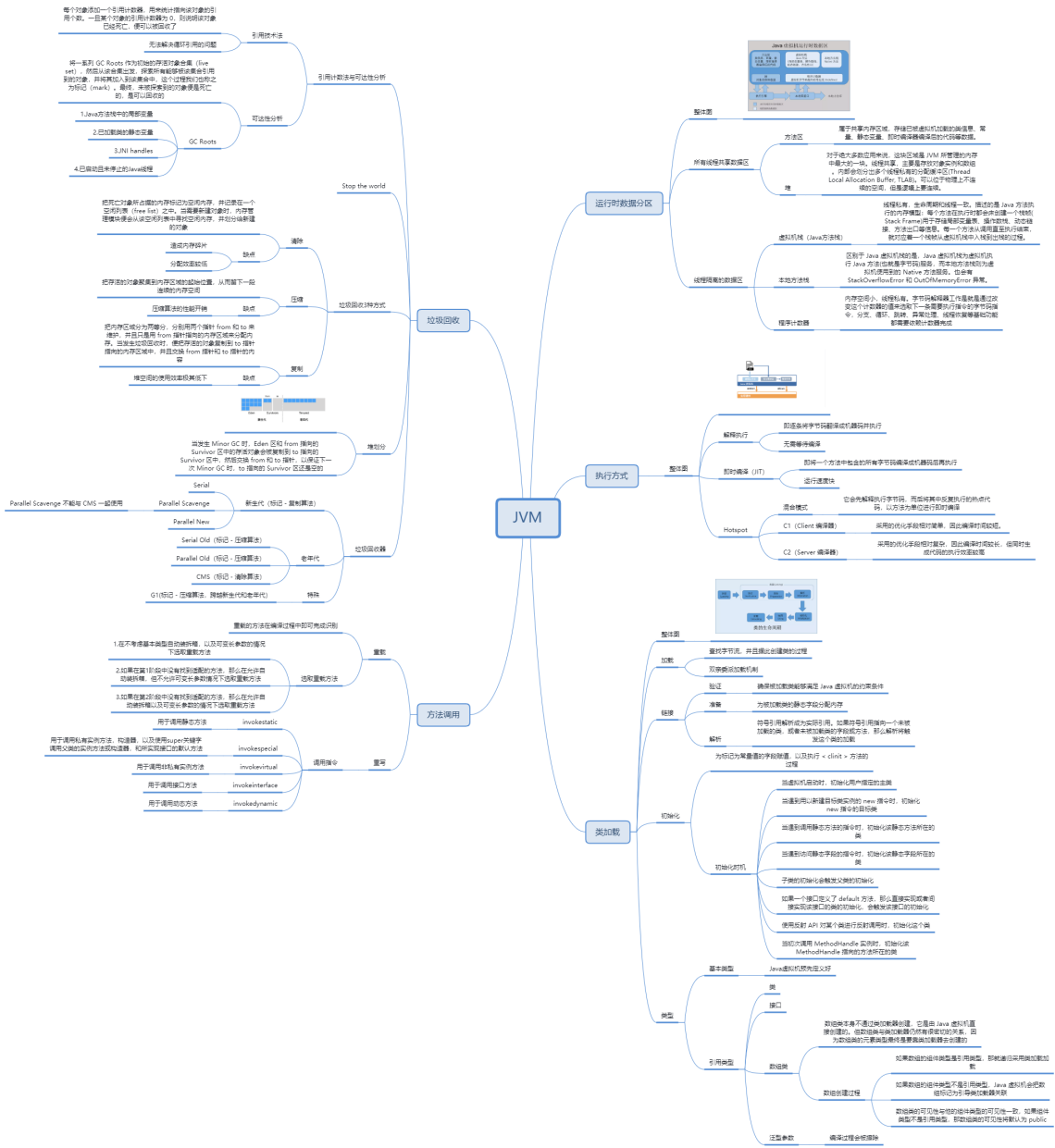


JVM总结

Java8中JVM内存结构：

Stack		Heap Space								Non-Heap Space						
Program Counter Register		Young Generation			Old Generation		Runtime Constant Pool			MetaSpace			Native Memory		Code Cache	
VM Stack	Native Stack	Eden TLAB	From Survivor 0	To Survivor 1	Tenured	Humongous	Symbolic Reference	Literal	Compressed Class Space	Compile Code	Field&Method Data	JNI Memory	Direct Memory	Stack Memory	JIT Compile	JIT Code

JVM相关基础知识：



JVM学习收获如下：

- 通过字节码技术，可以直观的看到Java的底层执行的逻辑，特别是现在代码糖日益变多，一个逻辑有很多的代码实现方式，那种方式很好，可以通过字节码技术透过现象看到本质，一窥究竟。
- Java程序写好之后，将GC日志放到gceasy网站，查看应用程序稳定运行时长长期存活对象在堆中占用的空间大小，也就是Full GC后堆中老年代占用空间大小，即活跃数据的大小。通过活跃数据和各分区之间的比例关系，推算各分区的大小。

空间	倍数
总大小	3-4 倍活跃数据的大小
新生代	1-1.5 活跃数据的大小
老年代	2-3 倍活跃数据的大小
永久代	1.2-1.5 倍Full GC后的永久代空间占用

- 结合学习的知识，将常VM命令（jmap、jps、jstat、jstack、jcmd）、jdk自带的图形化工具（jconsole、jvisualvm、jmc）、MemoryAnalyzerTool解决公司实际项目OOM问题。