

JVM面试课

个人公号:技术万花筒
欢迎关注!



JVM

01

jvm

02

gc/类加载

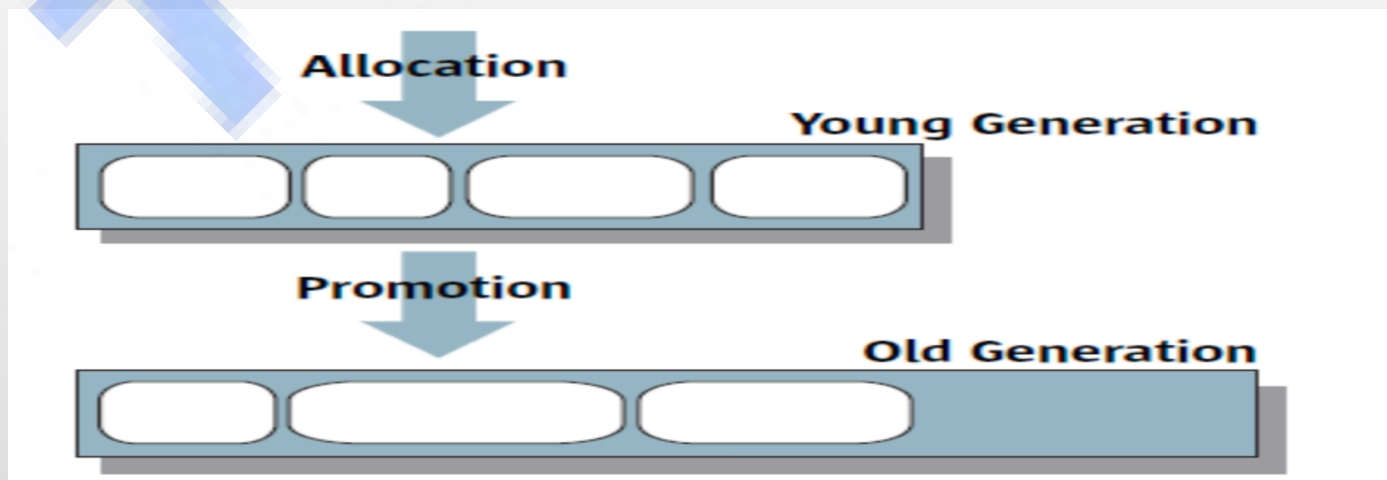
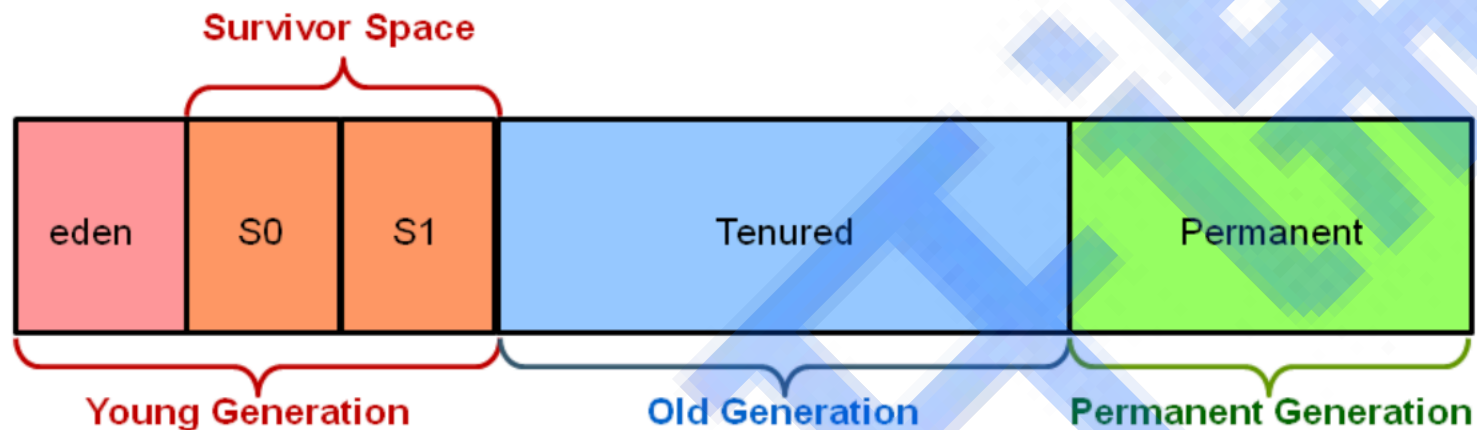
03

多线程/linux

04

案例

分代复制



年轻代

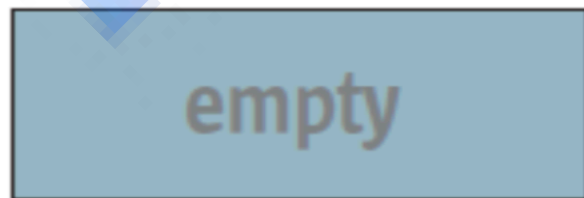
Young Generation



From



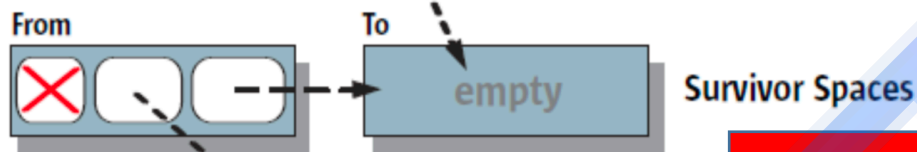
To



Survivor Spaces

Serial-年轻代

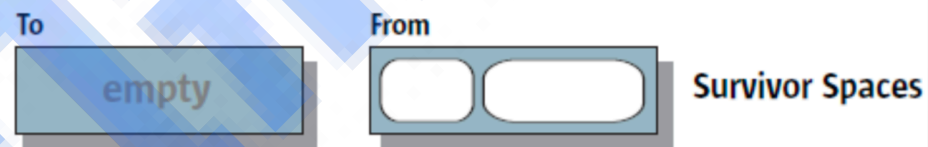
Young Generation



Old Generation



Young Generation



Old Generation



复制

适合Client型机器

-XX:+UseSerialGC=Serial+Serial Old

Serial-年老代

a) Start of Compaction



b) End of Compaction



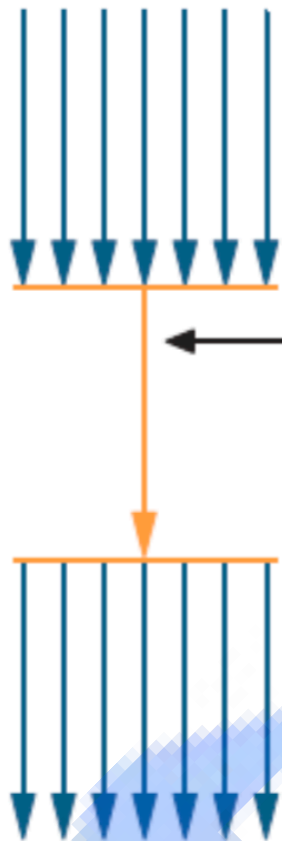
old:mark-sweep-compact

适合Client型机器

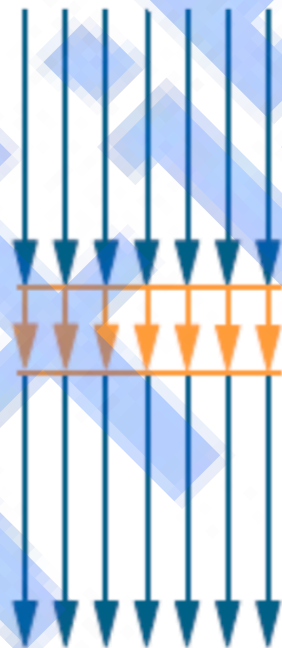
-XX:+UseSerialGC

Parallel-年轻代

Serial Collector



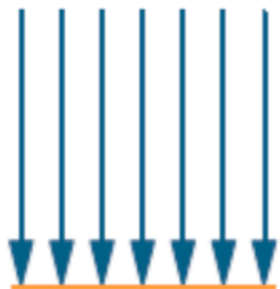
Parallel Collector



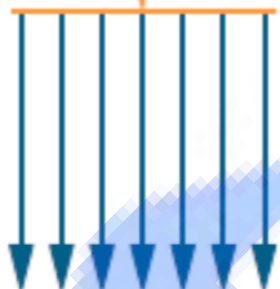
串行收集的并行版本
old+perm:mark-sweep-compact
适合多核Server型机器
-XX:+UseParNewGC =
ParallelNew+SerialOld(不推荐)

Parallel Scavenge-年轻代

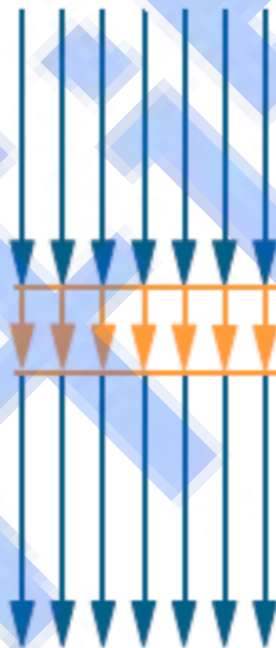
Serial Collector



Stop-the-world pause



Parallel Collector



吞吐量优先

old+perm:mark-sweep-compact

适合多核Server型机器

-XX:+UseParallelGC=

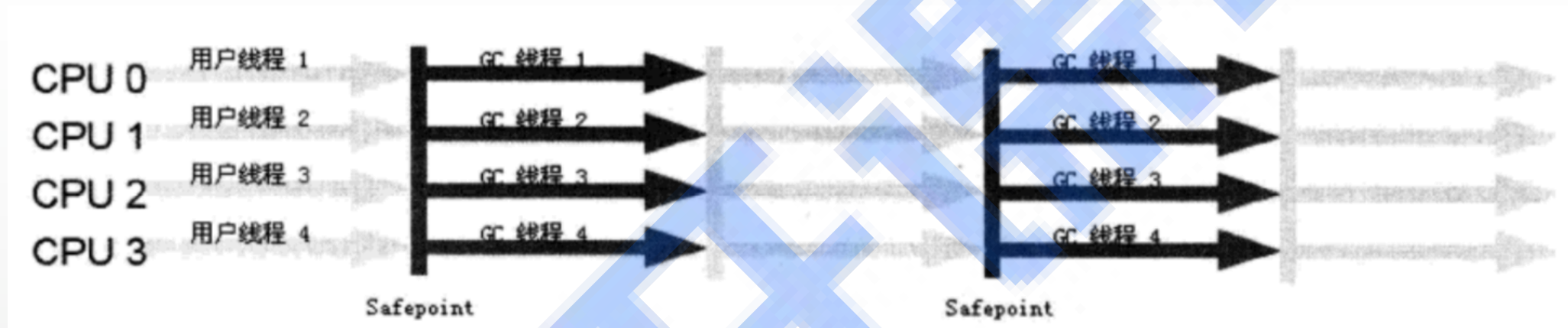
Parallel Scavenge+SerialOld(不推荐)

-XX:MaxGCPauseMillis : GC停顿

-XX:GCTimeRatio : GC时间占比

-XX:+UseAdaptiveSizePolicy : 自适应

Parallel-年老代



Parallel Scavenge的老年代版本

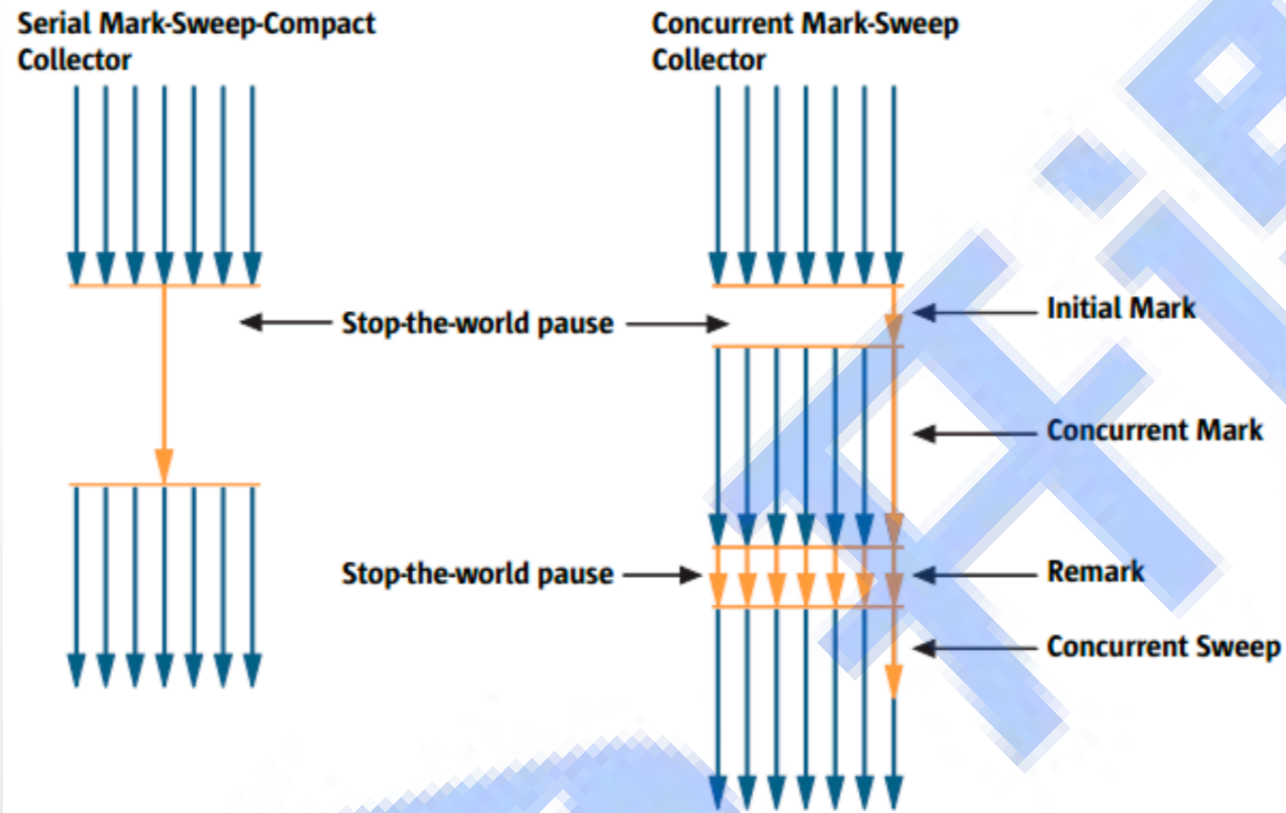
吞吐量优先

old+perm:mark-sweep-compact

适合多核Server型机器

-XX:+UseParallelOldGC

CMS



Concurrent Mark-Sweep

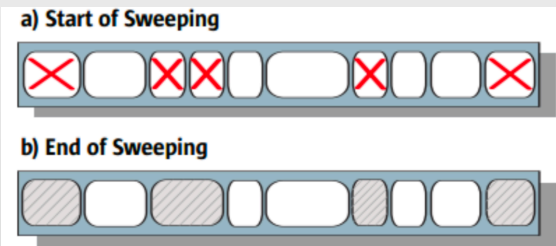
- 1、初始标记
- 2、并发标记
- 3、重新标记
- 4、并发清扫

-XX:+UseConcMarkSweepGC=
ParaNew+CMS+Serial Old

适合

- 1、更短的GC暂停
- 2、长生命周期对象
- 3、多核CPU

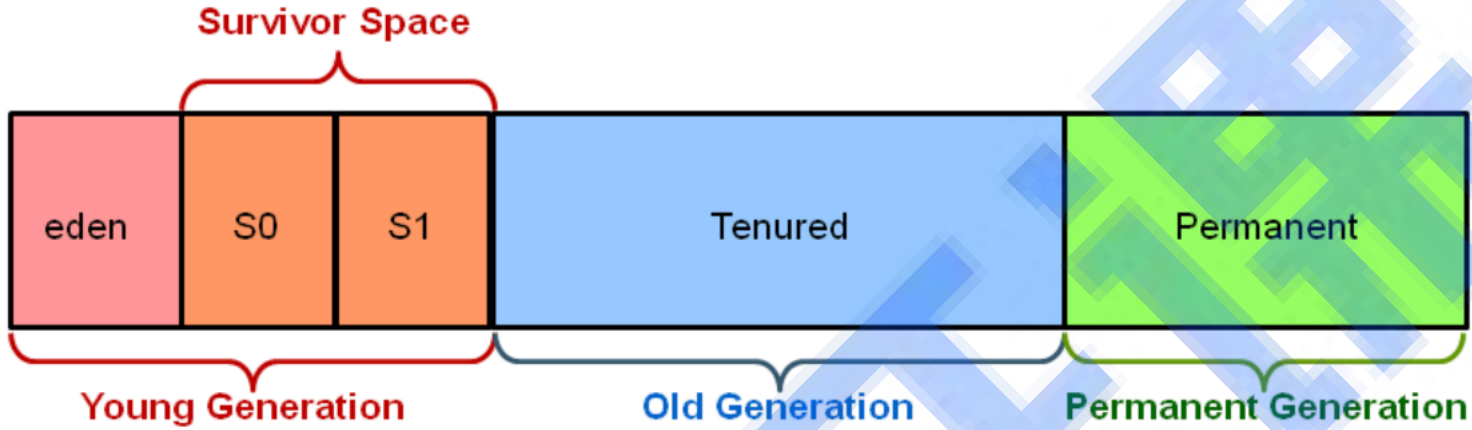
碎片



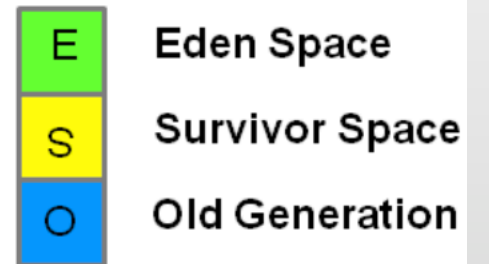
- XX:CMSInitiatingOccupancyFractio
- XX:+UseCMSCompactAtFullCollection
- XX:+CMSFullGCsBeforeCompaction

http://docs.oracle.com/javase/8/docs/technotes/guides/vm/gctuning/cms.html#concurrent_mark_sweep_cms_collector

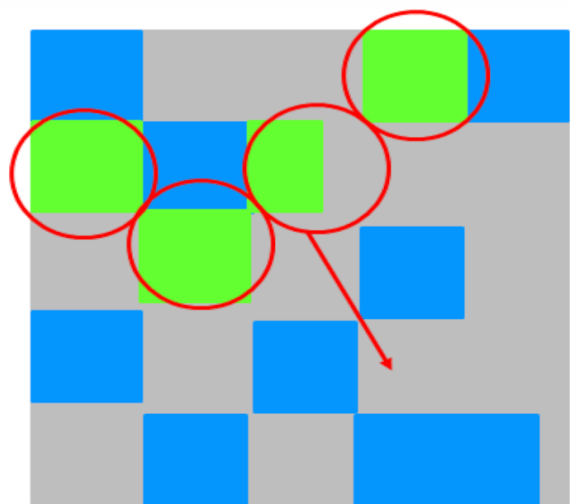
G1



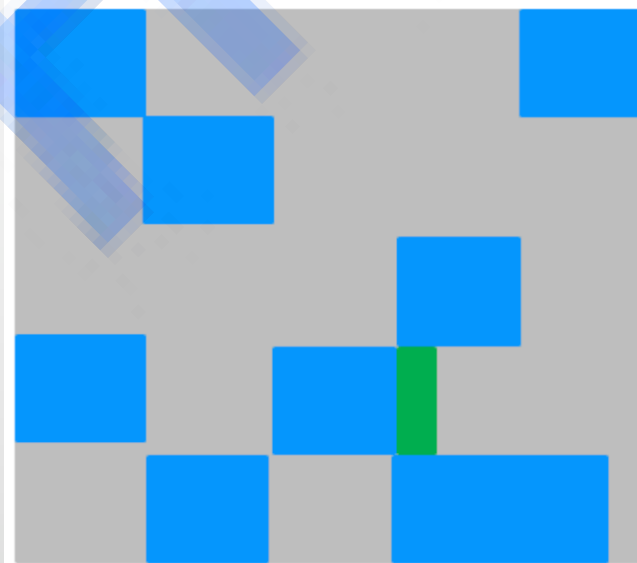
-XX : + UseG1GC



G1-新生代收集

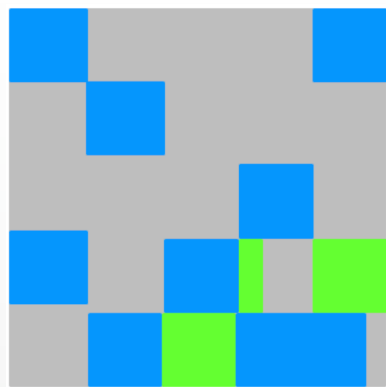


- Non-Allocated Space
- Young Generation
- Old Generation
- Recently Copied in Young Generation
- Recently Copied in Old Generation



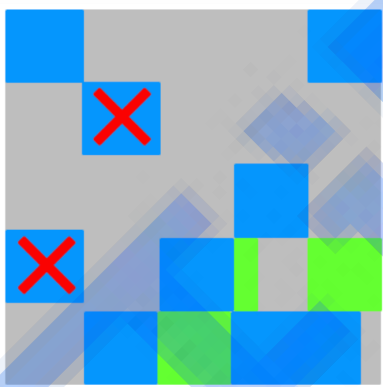
- Non-Allocated Space
- Young Generation
- Old Generation
- Recently Copied in Young Generation
- Recently Copied in Old Generation

G1-年老代收集

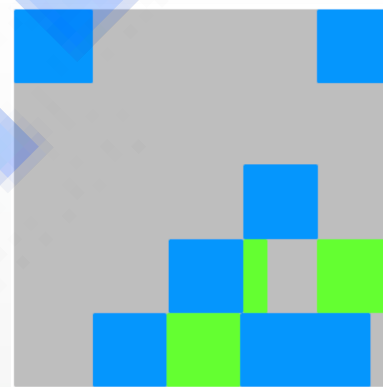


初始标记

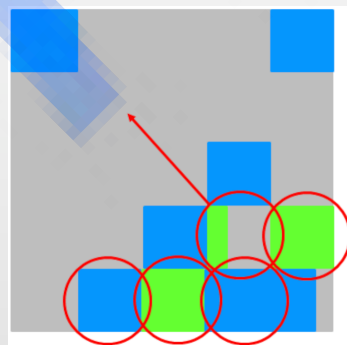
GC pause (young)(initial-mark)



并发标记

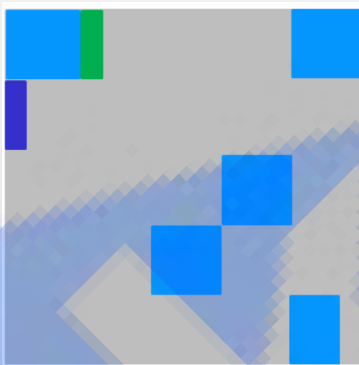


重新标记



复制/清除

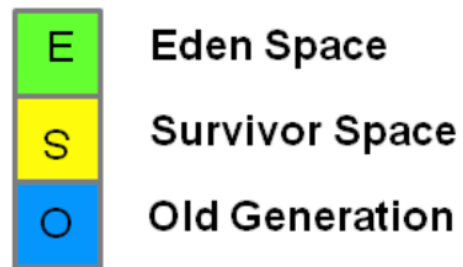
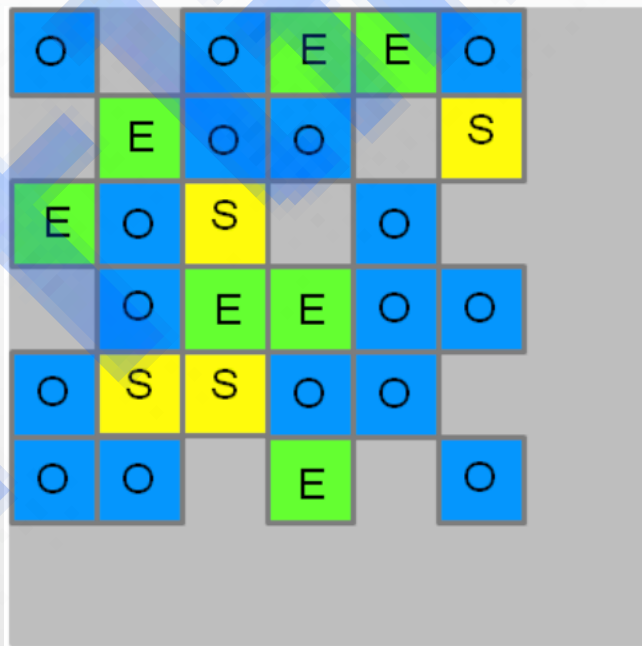
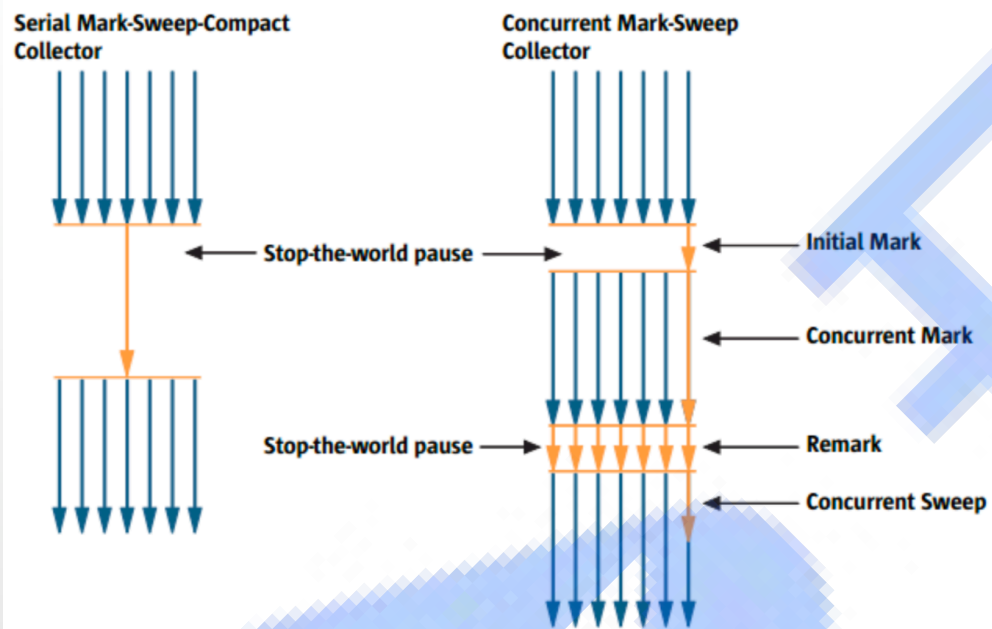
GC pause (mixed)



结果

Grey	Non-Allocated Space
Green	Young Generation
Blue	Old Generation
Dark Green	Recently Copied in Young Generation
Dark Blue	Recently Copied in Old Generation

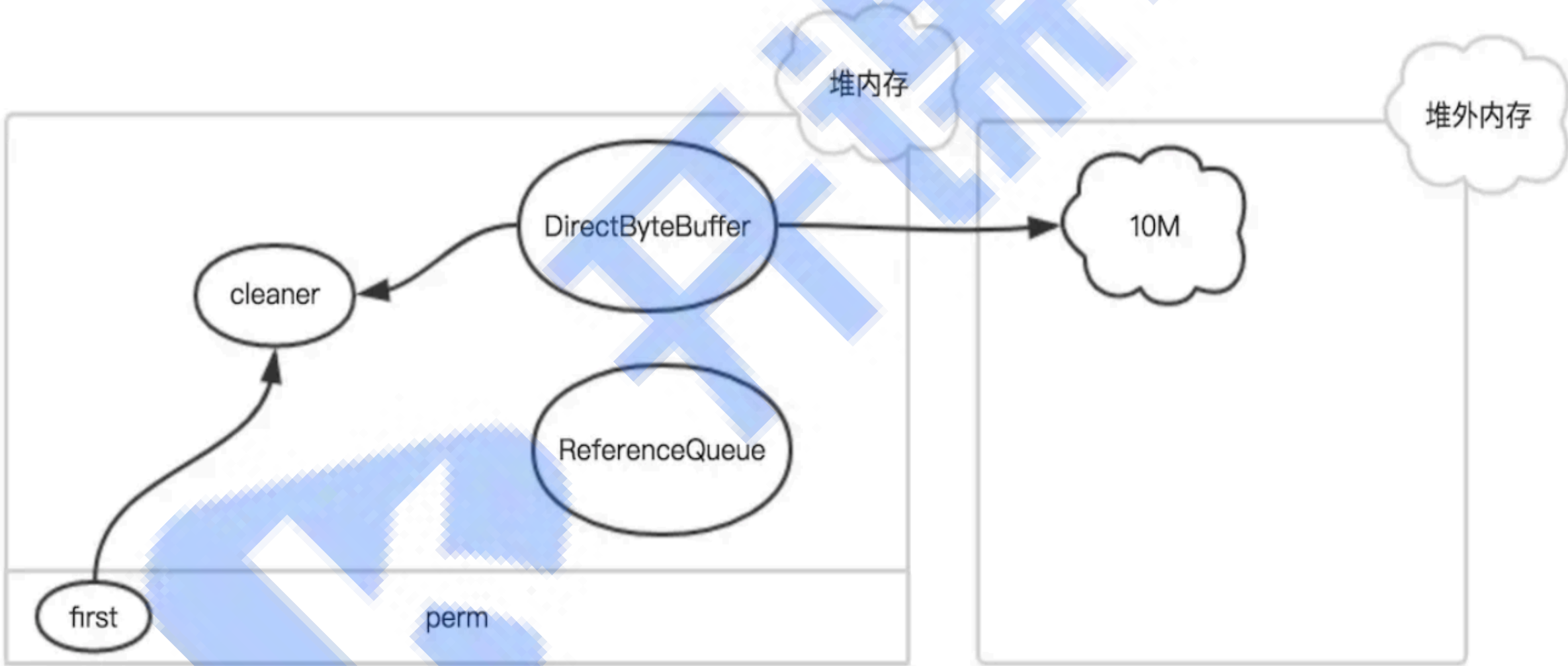
G1-CMS区别对比



G1-最佳实践

- **-Xmn** , 将禁用暂停时间限制XX : **MaxGCPauseMillis** , 且无法动态扩展
- **to-space overflow** , to-space exhausted
 - **增加堆大小**
 - **-XX:G1ReservePercent** , 增加保留空间
 - **-XX:InitiatingHeapOccupancyPercent** , 减少提前启动
 - **-XX:ConcGCThreads=n** , 增加标记线程数量

Direct Memory



Direct Memory

- OOM->dump
- GC无效



GC算法组合

年轻代

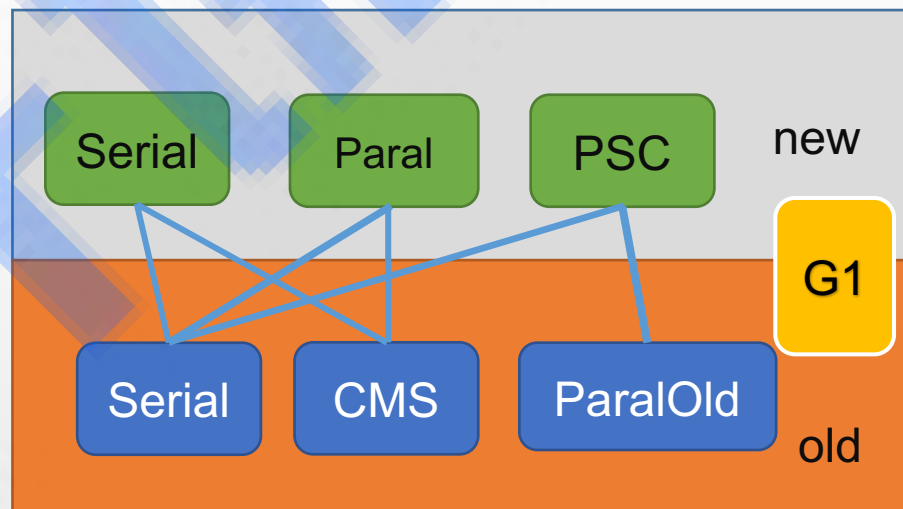
- Serial、ParNew、Parallel Scavenge

年老代

- SerialOld、ParallelOld、CMS

全堆收集

- G1



<http://www.oracle.com/technetwork/java/javase/tech/memorymanagement-whitepaper-1-150020.pdf>

jstat

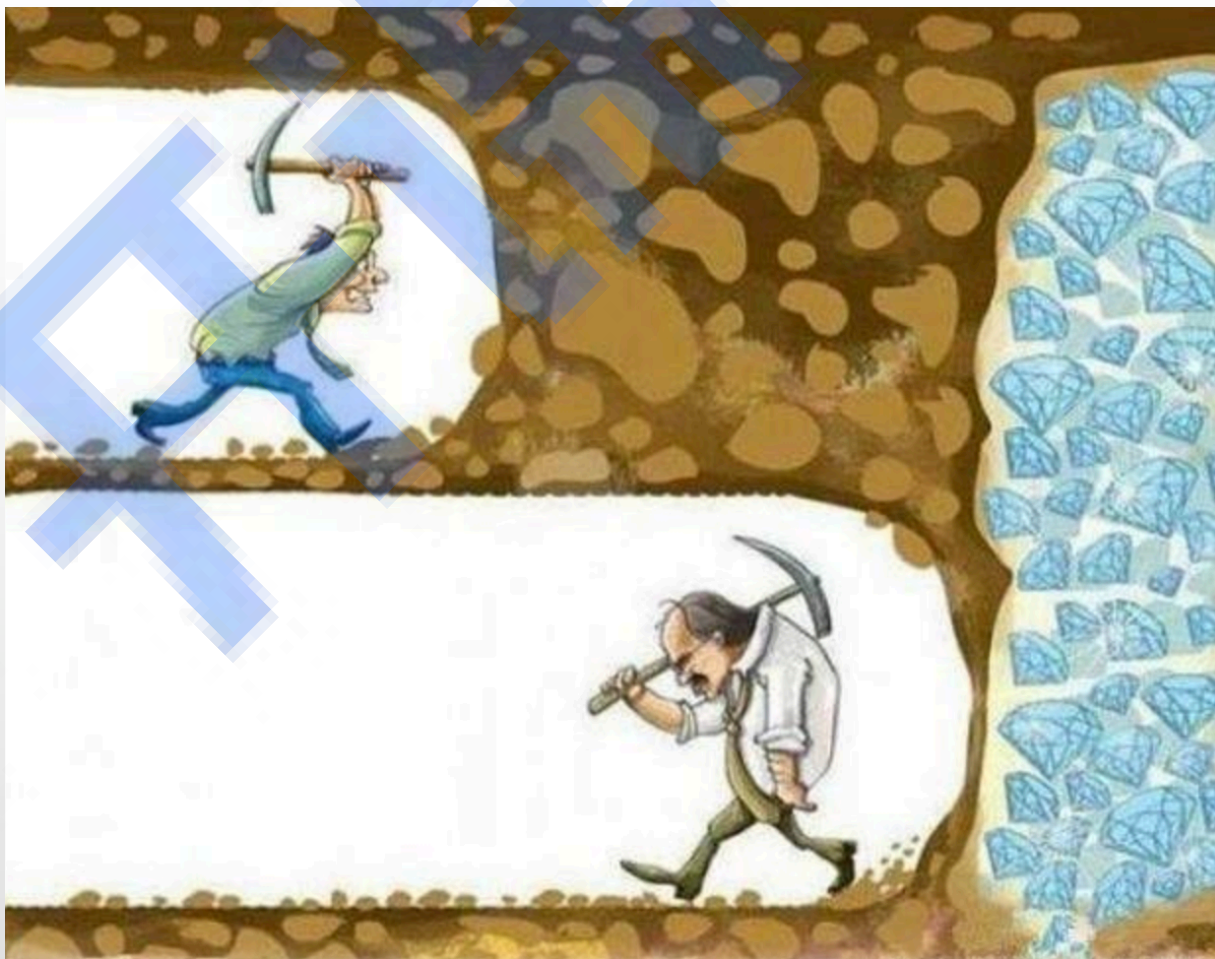
```
[~] $ jstat
invalid argument count
Usage: jstat -help|-options
       jstat -<option> [-t] [-h<lines>] <vmid> [<interval> [<count>]]

Definitions:
<option>      An option reported by the -options option
<vmid>        Virtual Machine Identifier. A vmid takes the following form:
               <lvmid>[@<hostname>[:<port>]]
               Where <lvmid> is the local vm identifier for the target
               Java virtual machine, typically a process id; <hostname> is
               the name of the host running the target Java virtual machine;
               and <port> is the port number for the rmiregistry on the
               target host. See the jvmsat documentation for a more complete
               description of the Virtual Machine Identifier.
<lines>       Number of samples between header lines.
<interval>    Sampling interval. The following forms are allowed:
               <n>["ms"|"s"]
               Where <n> is an integer and the suffix specifies the units as
               milliseconds("ms") or seconds("s"). The default units are "ms".
<count>       Number of samples to take before terminating.
-J<flag>      Pass <flag> directly to the runtime system.
```

```
[~] $ jstat -options
-class
-compiler
-gc
-gccapacity
-gccause
-gcmetacapacity
-gcnew
-gcnewcapacity
-gcold
-gcoldcapacity
-gcutil
-printcompilation
```

Key Point

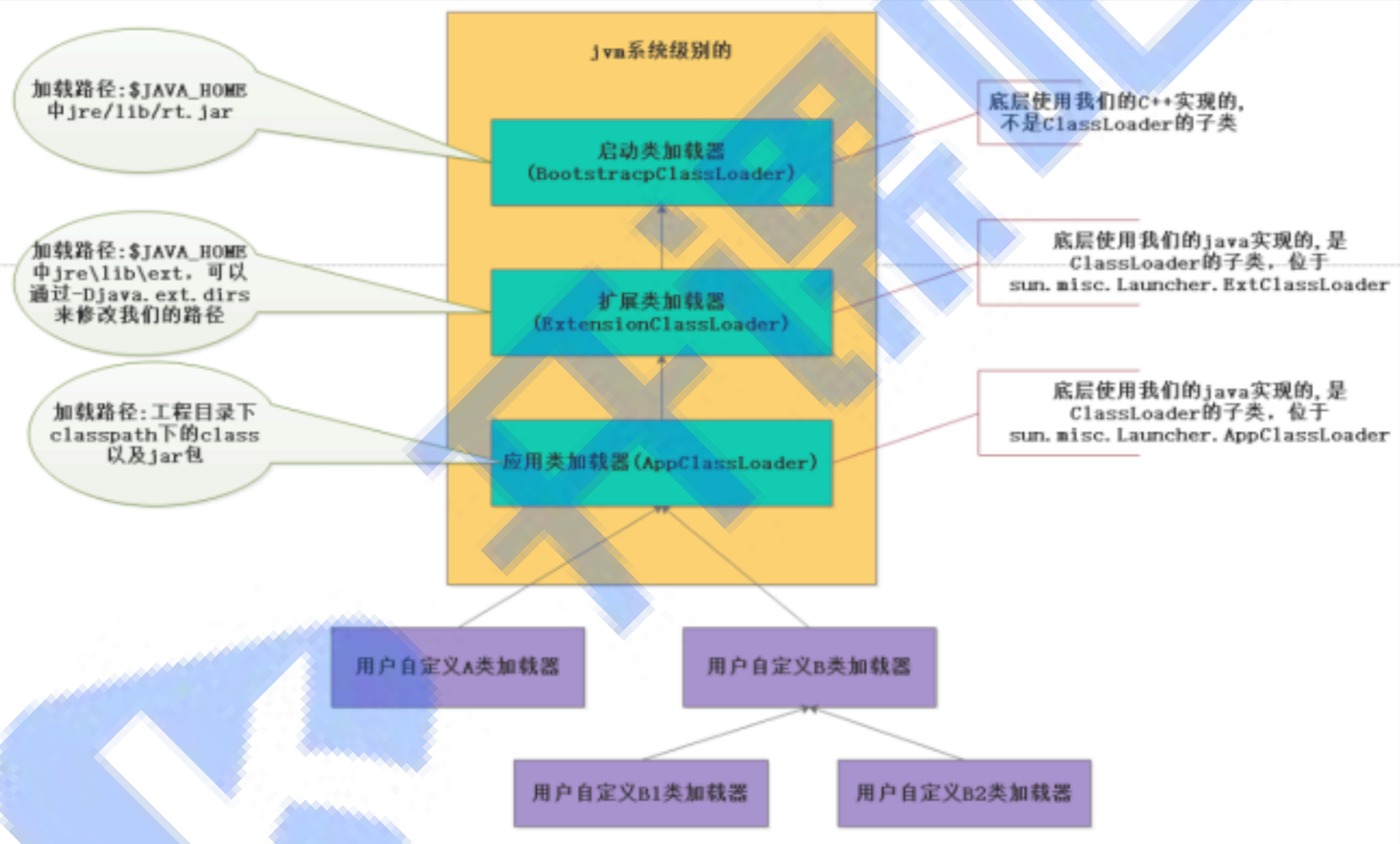
- 根据监控指标和表象切入
- 自定往下，顺藤摸瓜
- 必要的工具+客观理性的分析
 - 不选择性忽略
 - 一行代码都不要放过
- 一定要了解&结合业务**



jvm面试课

02

gc/类加载



THANK YOU FOR WATCHING