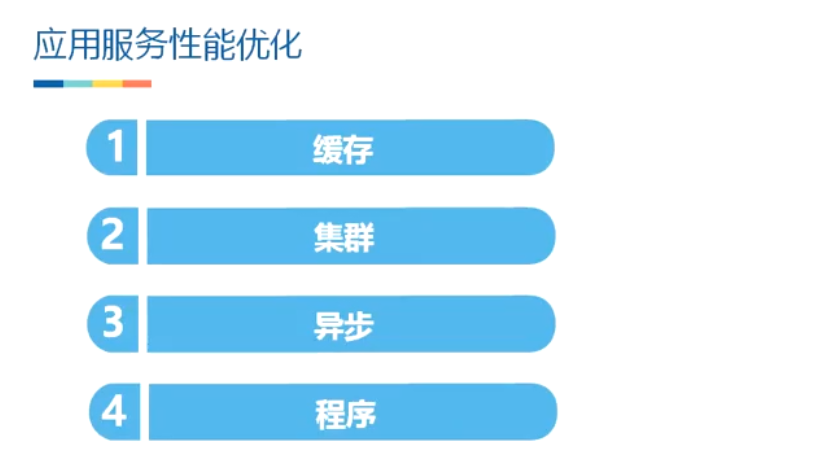
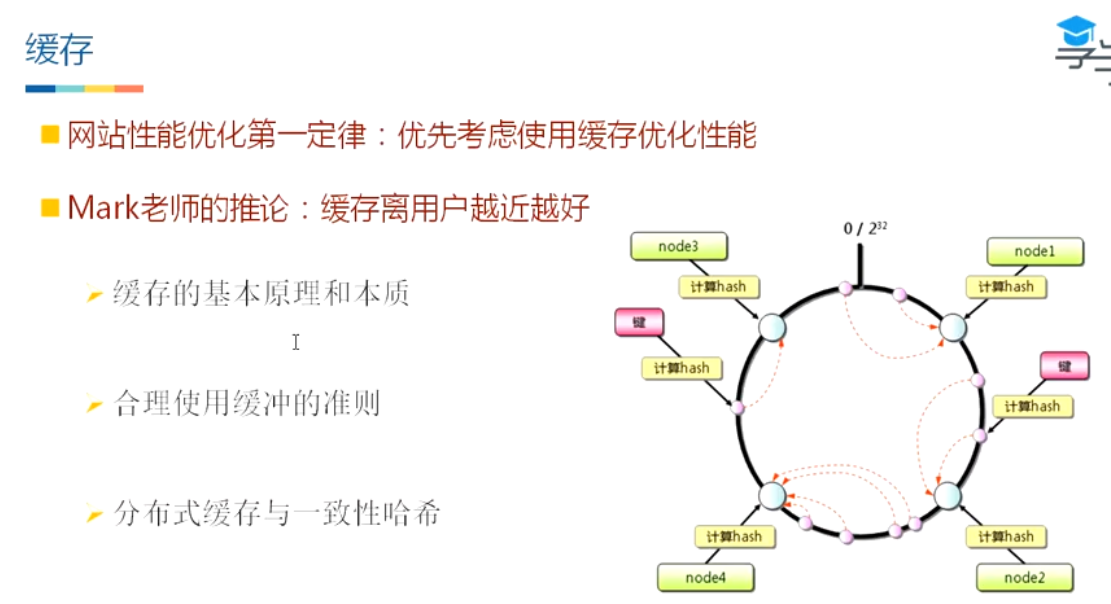


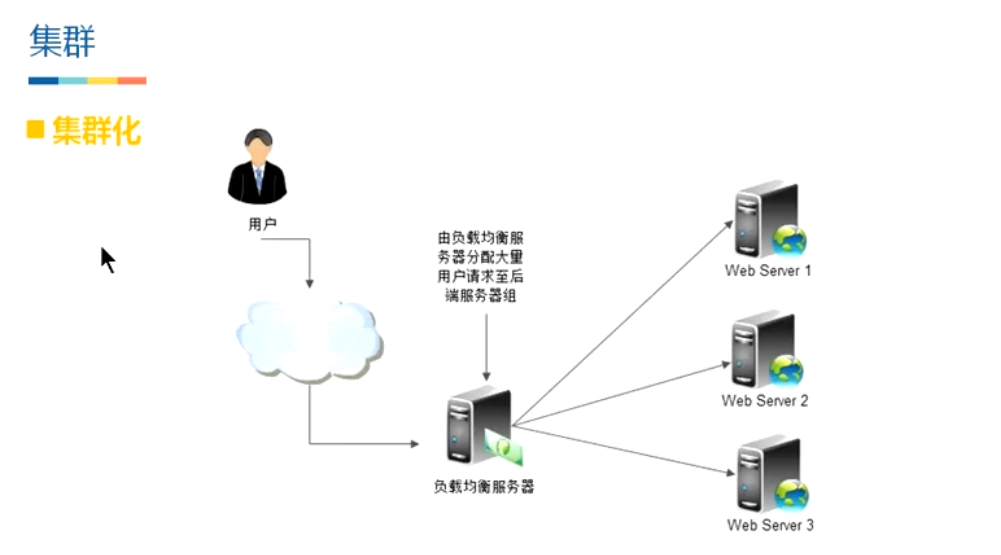
不同服务使用不同的域名，可以提高浏览器加载数据的并发量

下载完CSS后才会渲染，浏览器在下载完js后马上执行











即时编译—JIT Just in Time

-server 编译的晚一些

-XX:+TieredCompilation 分层编译 jdk1.7引入 默认关闭， jdk1.8默认打开

-Xint 禁用jit

-XComp 最快

-Xmixed 混合模式



-XX:ReservedCodeCacheSize=N 设置代码缓存大小，缺省JDK1.7 32-48M之间；JDK8 240M

所以使用JDK7时可以适当调大这个参数

使用ReservedCodeCacheSize 会提前申请一块内存

如果代码缓存满了后，会报错： CodeCache is full

编译阀值：

方法调用计数器：

循环回边计数器：



调整的影响不大，建议使用默认值。

编译线程：建议不调整

方法内联： -XX:inline 来开启关闭内联

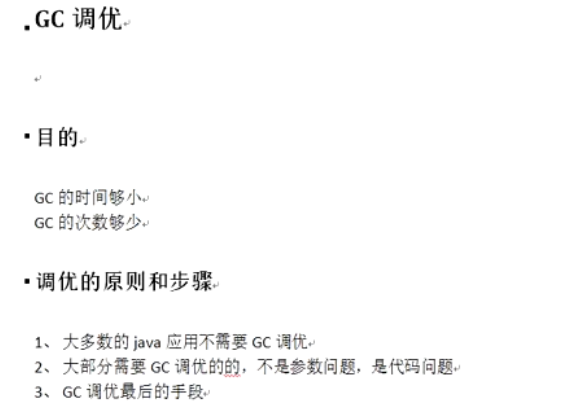
默认开启，不要关闭。可以减少栈帧，提高性能，性能提升大概50%，一个方法是否内联通过热点分析来判断。

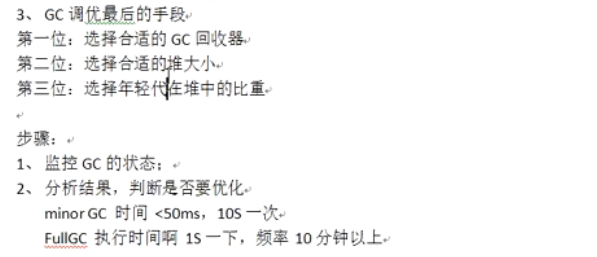
只有方法字节码小于325字节才会内联 -XX:MaxFreqInlinesSize=N 来设置内联的阀值。

方法小于35个字节，一定会内联 –XX:MaxInlineSize

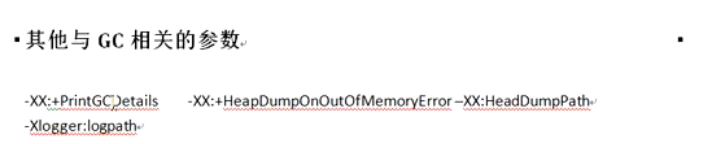
逃逸分析：尽量不要动，默认开启







GC 调优的参考点： 执行时间、执行频率



生产环境开启： 发送OOM时候打印日志，并且指定打印的位置

-XX:+HeapDumpOnOutOfMemoryError -XX:HeadDumpPath

这两个参数选择使用

-XX:PrintHeapAtGC 查看堆回收前后的状态

-XX:TraceClassLoading