

Java架构师必备技能之 高并发及性能调优系列课程训练营

讲师：JackHu

时间：2020年9月9日

个人简介：

开课吧特约资深讲师&课程研究员
知学科技首席架构师&创始人
博学谷课程研究&技术负责人
51拜师技术架构师&负责人
药房网技术负责人
华北计算研究所技术负责人
机械工业出版社《源码修炼指南》
《RocketMQ实战内幕》



章节内容



1.为什么要进行JVM调优

2.深入分析JVM调优原理

3.JVM调优实践

4.GC日志分析 (观察YGC/FGC)

5.吞吐量优先调优

6.响应时间优先调优

7.G1垃圾回收器

01 /

为什么要进行JVM调优

kaikedaba

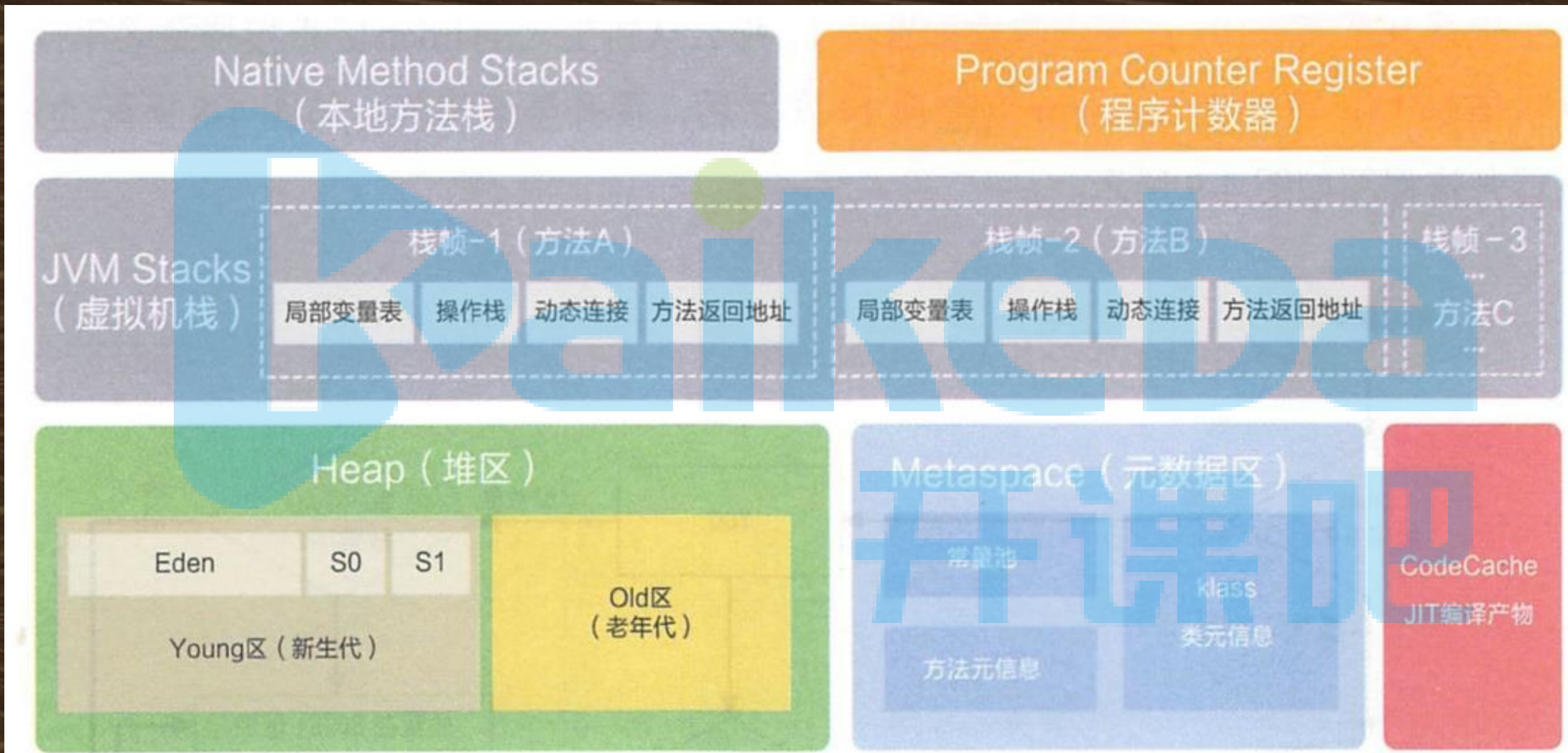
开课吧

思考问题：为什么要进行JVM调优？

开课吧

- 垃圾太多，内存占满
- 垃圾回收线程太多





Java虚拟机内存模型：垃圾回收对象主要在Heap（堆区）

思考问题：JVM可利用多大的内存空间？

开课吧

- 操作系统的位数决定寻址空间决定服务器内存大小
- 32位操作系统 (maxMemory:4GB)
- 64位操作系统 (maxMemory: 16384PB)

- GC时间足够小（堆内存设置小）
- GC次数足够少（堆内存设置大）
- 发生full GC的周期足够长

开课吧

02/

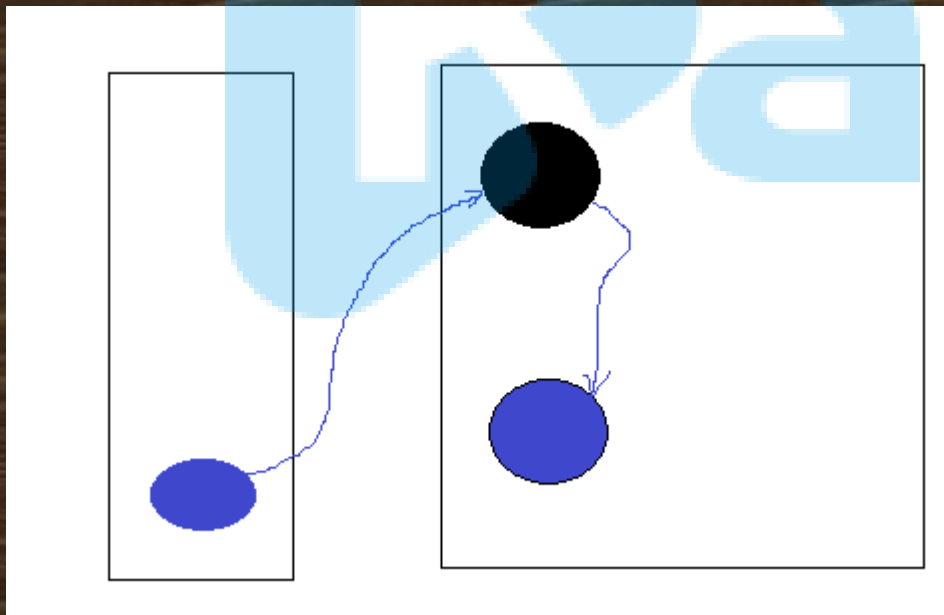
深入分析JVM调优原理

kaikedaba

开课吧

灵魂拷问：什么是垃圾？？

- 是指那些不可能再被任何途径使用的对象所占的内存空间（哪些对象），就是垃圾

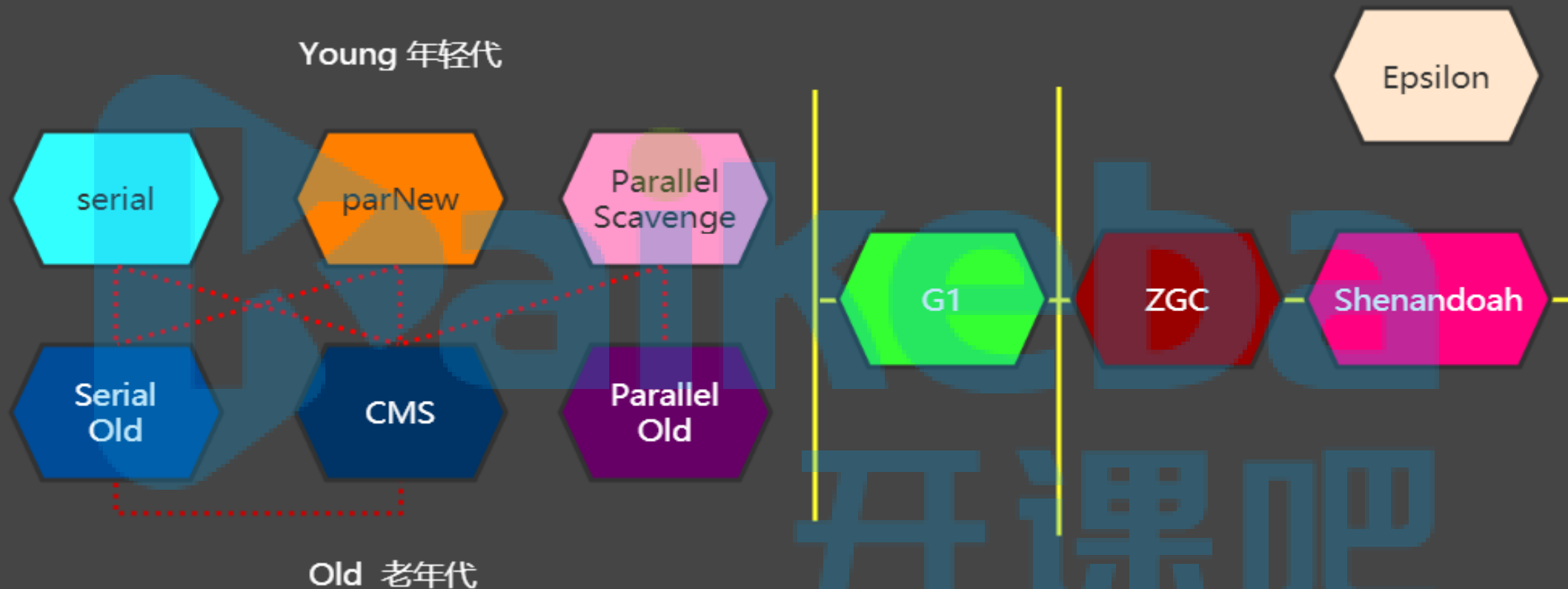


- 引用计数算法
- 根可达算法



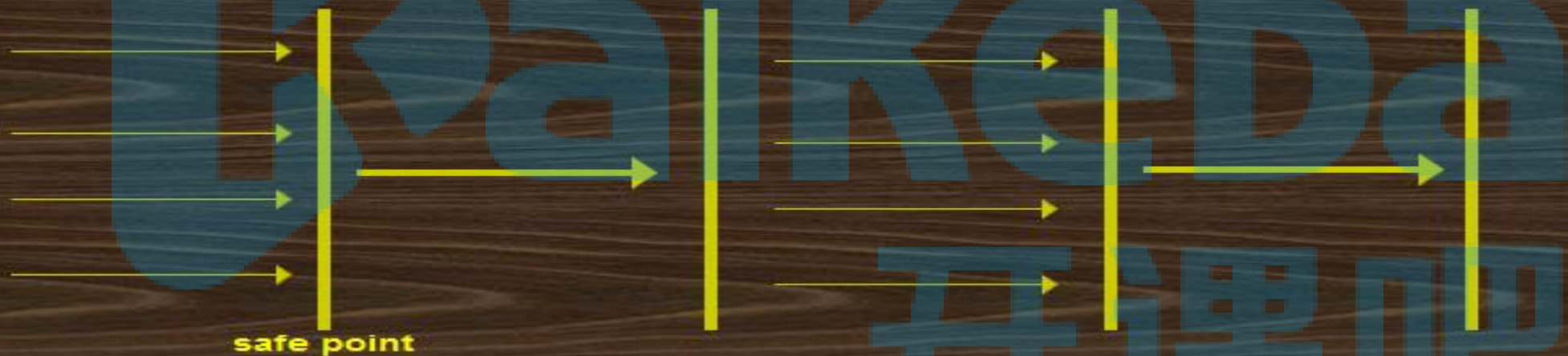
- Mark-Sweep 标记清除法
- Copying 拷贝算法
- Mark-Compact 标记压缩算法

开课吧



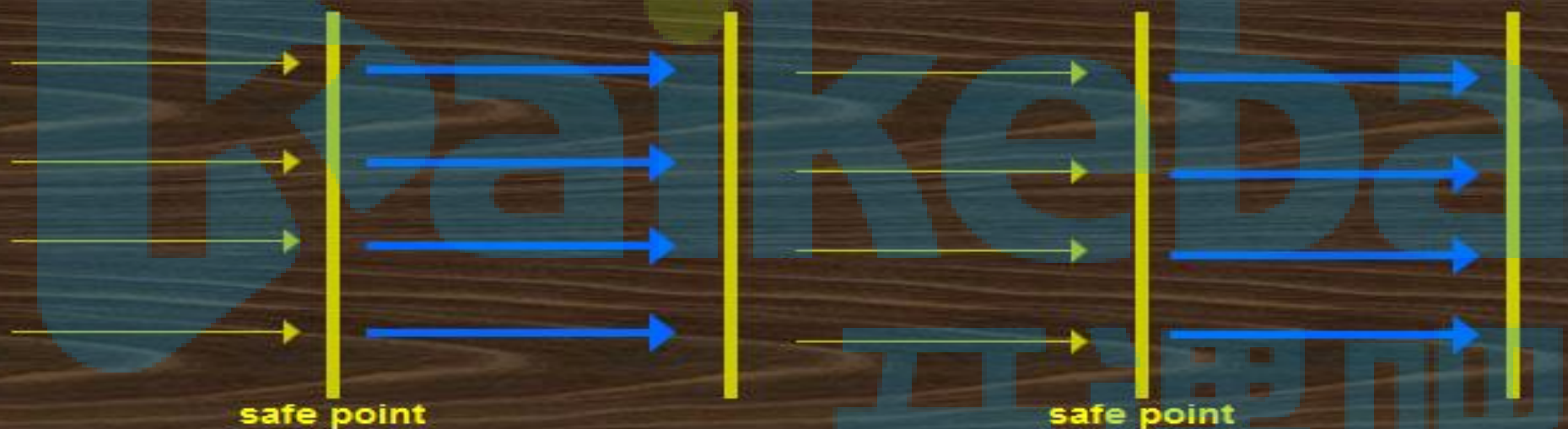
- 串行回收器：Serial、Serial old
- 并行回收器：ParNew、Parallel Scavenge、Parallel old
- 并发回收器：CMS、G1（分区算法）

- Stop-The-World
- Copying collector which uses a single GC Threads



单CPU效率最高，是client模式的默认垃圾回收器

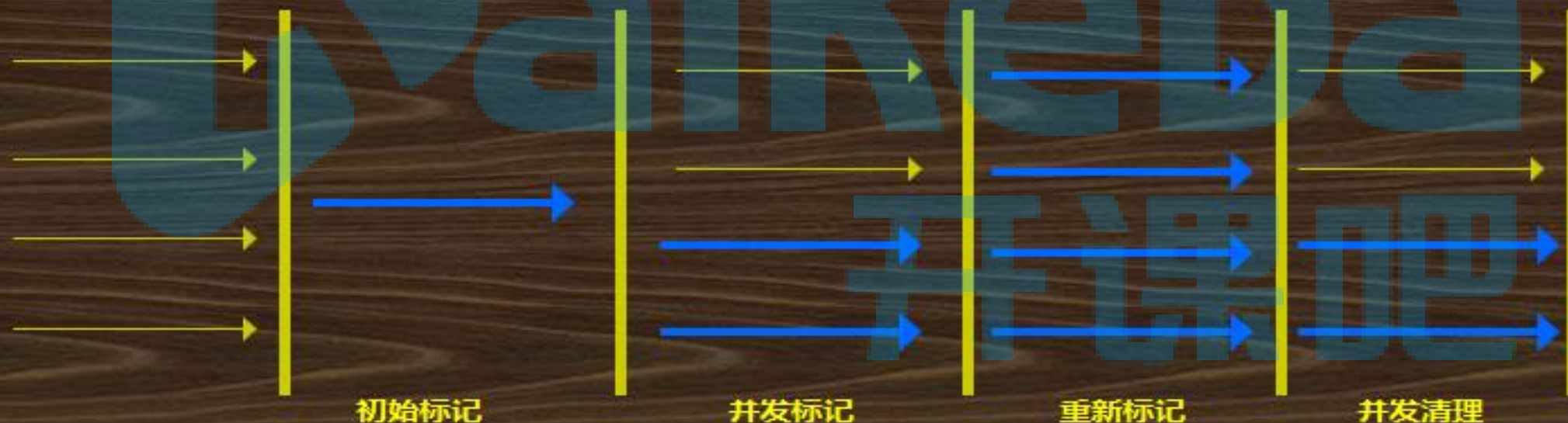
- Stop-The-World
- Copying collector which uses multiple GC Threads



并行回收器：

ParNew
Parallel Scavenge
Parallel old

- Stop-The-World
- concurrent mark sweep , A mostly concurrent , low-pause collector



- Stop-The-World
- Garbage First (G1)



使用G1收集器时，它将整个Java堆划分成约2048个大小相同的独立Region块，每个Region块大小根据堆空间的实际大小而定，整体被控制在1MB到32MB之间，且为2的N次幂，即1MB, 2MB, 4MB, 8MB, 16MB, 32MB。可以通过-XX:G1HeapRegionSize设定。所有的Region大小相同，且在JVM生命周期内不会被改变；

ZGC是一款JDK 11中新加入的具有实验性质的低延迟垃圾收集器，ZGC源自于是Azul System公司开发的C4（ Concurrent Continuously Compacting Collector ）收集器。

Supported Platforms

Platform	Supported	Since	Comment
Linux/x64	✓	JDK 11	
Linux/AArch64	✓	JDK 13	
macOS	✓	JDK 14	
Windows	✓	JDK 14	Requires Windows version 1803 (Windows 10 or Windows Server 2019) or later.

ZGC是一款JDK 11中新加入的具有实验性质的低延迟垃圾收集器，ZGC源自于是Azul System公司开发的C4（Concurrent Continuously Compacting Collector）收集器。ZGC收集器是一款基于Region内存布局的，暂时不设分代的，使用了读屏障、颜色指针等技术来实现可并发的标记-整理算法的，以低延迟为首要目标的一款垃圾收集器。

ZGC的Region可以具有大、中、小三类容量：

小型Region（Small Region）：容量固定为2MB，用于放置小于256KB的小对象。

中型Region（Medium Region）：容量固定为32MB，用于放置大于等于256KB但小于4MB的对象。

大型Region（Large Region）：容量不固定，可以动态变化，但必须为2MB的整数倍，用于放置4MB或以上的大对象。

Goals

TB

Multi-terabyte heaps

10_{ms}

Max GC pause time

15%

Max application throughput reduction



Lay the foundation for future GC features

