



ParNew GC算法原理

淘宝核心系统-专用计算组-洪熙

淘宝网

目录

- **GC算法介绍**
- JVM对象模型
- JVM Heap的分代
- JVM内存分配
- ParNew GC算法
- Promotion failure
- ParNew调优

GC算法介绍

- JVM种类
 - SUN hotspot , Harmony , IBM J9 , BEA Jrocket
- GC算法介绍 (Hotspot为例)
 - PS算法
 - CMS算法
 - G1算法

目录

- GC算法介绍
- **JVM对象模型**
- JVM Heap的分代
- JVM内存分配
- ParNew GC算法
- Promotion failure
- ParNew调优

JVM对象模型介绍

- Java对象模型 (x86_64)

markOop(8B)	Klass(8B)
field	Field
field	field

- Java对象的真实内存数据

```
(gdb) x/20xg obj
0x2aaaae0f2ee0: 0x0000000000000001      0x00002aaac7102588
0x2aaaae0f2ef0: 0x00002aaaae0f2f08      0x0000000700000000
0x2aaaae0f2f00: 0x000000001f46f70b      0x0000000000000001
```

```
(gdb) x/20xg obj
0x2aaaae262b08: 0x00002aaab985812b      0x00002aaac74b2de6
0x2aaaae262b18: 0x00073b0200000383      0x0000000000000000
0x2aaaae262b28: 0x00002aaaae2672e8      0x00002aaaae267c48
0x2aaaae262b38: 0x00002aaaae267b20      0x00002aaaafc834a0
0x2aaaae262b48: 0x00002aaab87aad10      0x0000000000000001
```

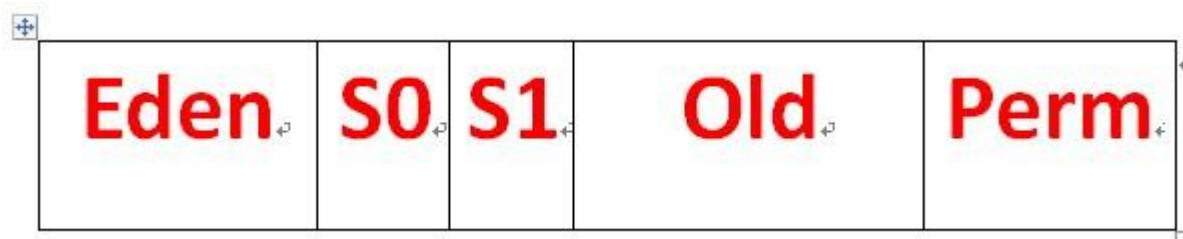
目录

- GC算法介绍
- JVM对象模型
- **JVM Heap的分代**
- JVM内存分配
- ParNew GC算法
- Promotion failure
- ParNew调优

JVM Heap的分代

- Java Heap内存申请

- mmap整体申请
- 设置每个gen的边界
- JVM虚拟内存管理
- GC后会swap so和s1
- ParNew 回收eden+survrior
- CMS回收old+perm (需要加参数)
- MSC回收整个heap

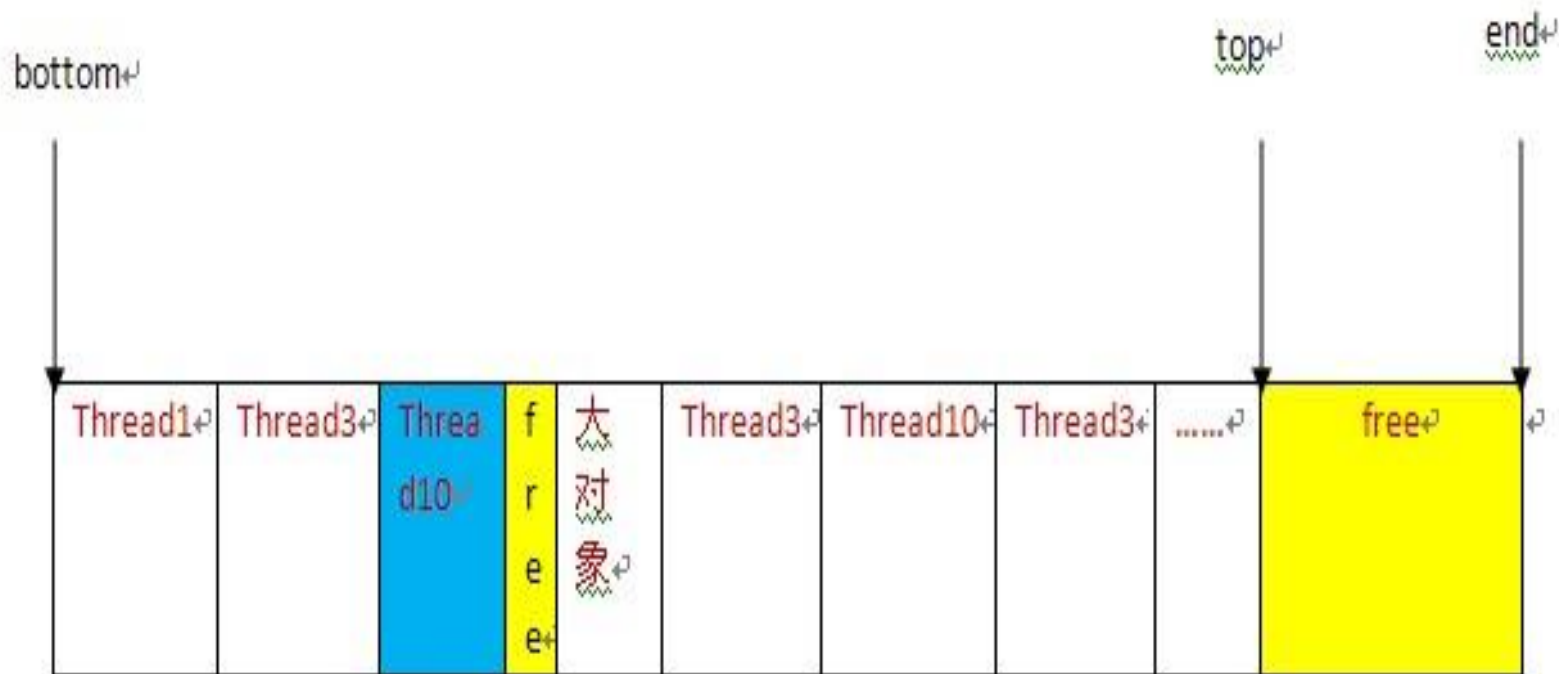


目录

- GC算法介绍
- JVM对象模型
- JVM Heap的分代
- **JVM内存分配**
- ParNew GC算法
- Promotion failure
- ParNew调优

Java对象的内存分配方法

- TLAB的概念(-XX+UseTLAB)
- 大对象与TLAB
- Minor GC的触发



目录

- GC算法介绍
- JVM对象模型
- JVM Heap的分代
- JVM内存分配
- **ParNew GC算法**
- Promotion failure
- ParNew 调优

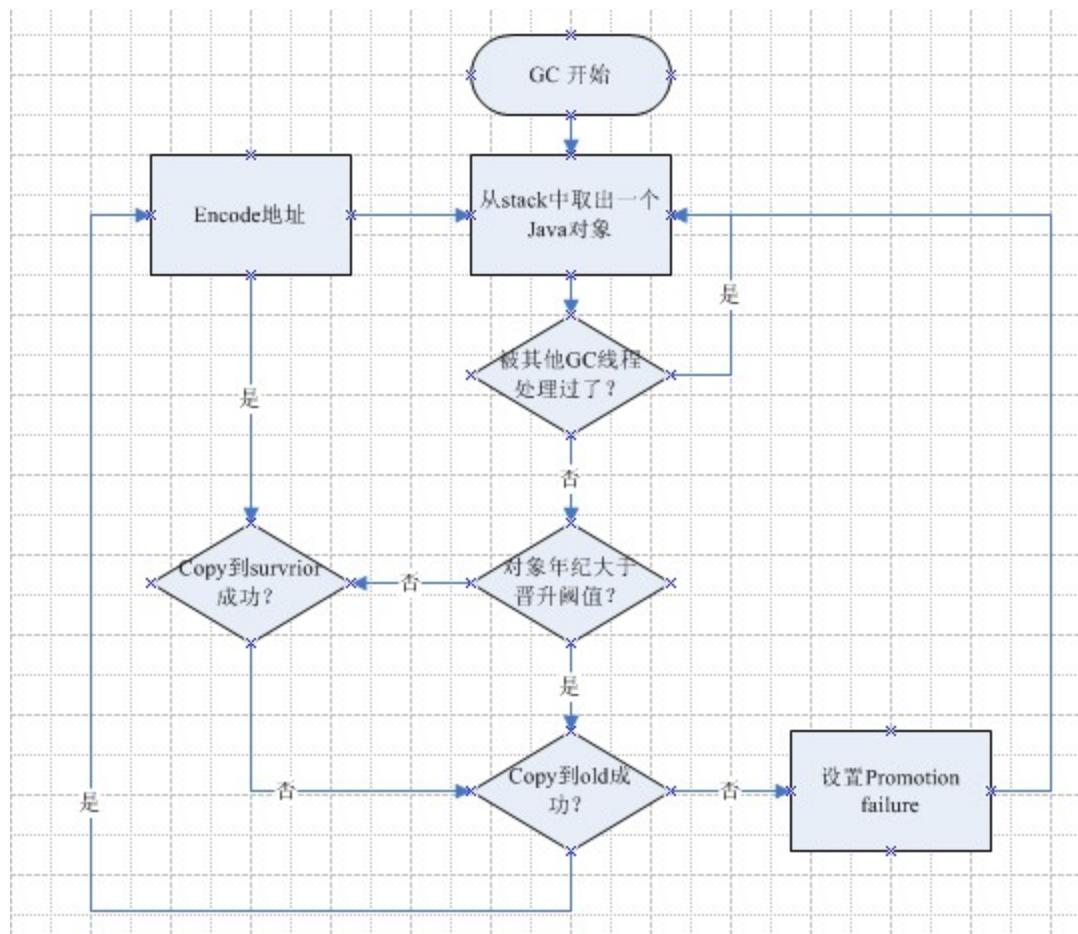
ParNew GC算法基本原理（日志）

- 通过-XX:+UseConcMarkSweepGC 来开始使用ParNew
- ParNew算法特点
- 一段ParNew的GC日志
2013-01-07T09:03:46.728+0800: 26.294: [GC 26.294: [ParNew:
1462912K->35625K(1755456K), 0.0869560 secs]
1462912K->35625K(4950336K), 0.0871000 secs] [Times:
user=0.22 sys=0.02, real=0.09 secs]
- 从上面的日志可以看出
 - GC开始时间
 - GC算法类型
 - GC前后young及整个Heap的使用量的变化
 - GC耗时
- 一次ParNew的这些时间到底消耗在哪里？

ParNew GC算法（总览）

- ParNew GC算法流程
 - 从root对象开始标记所有活的对象
 - 处理Card Table（old到young对象的引用）
 - 处理reference

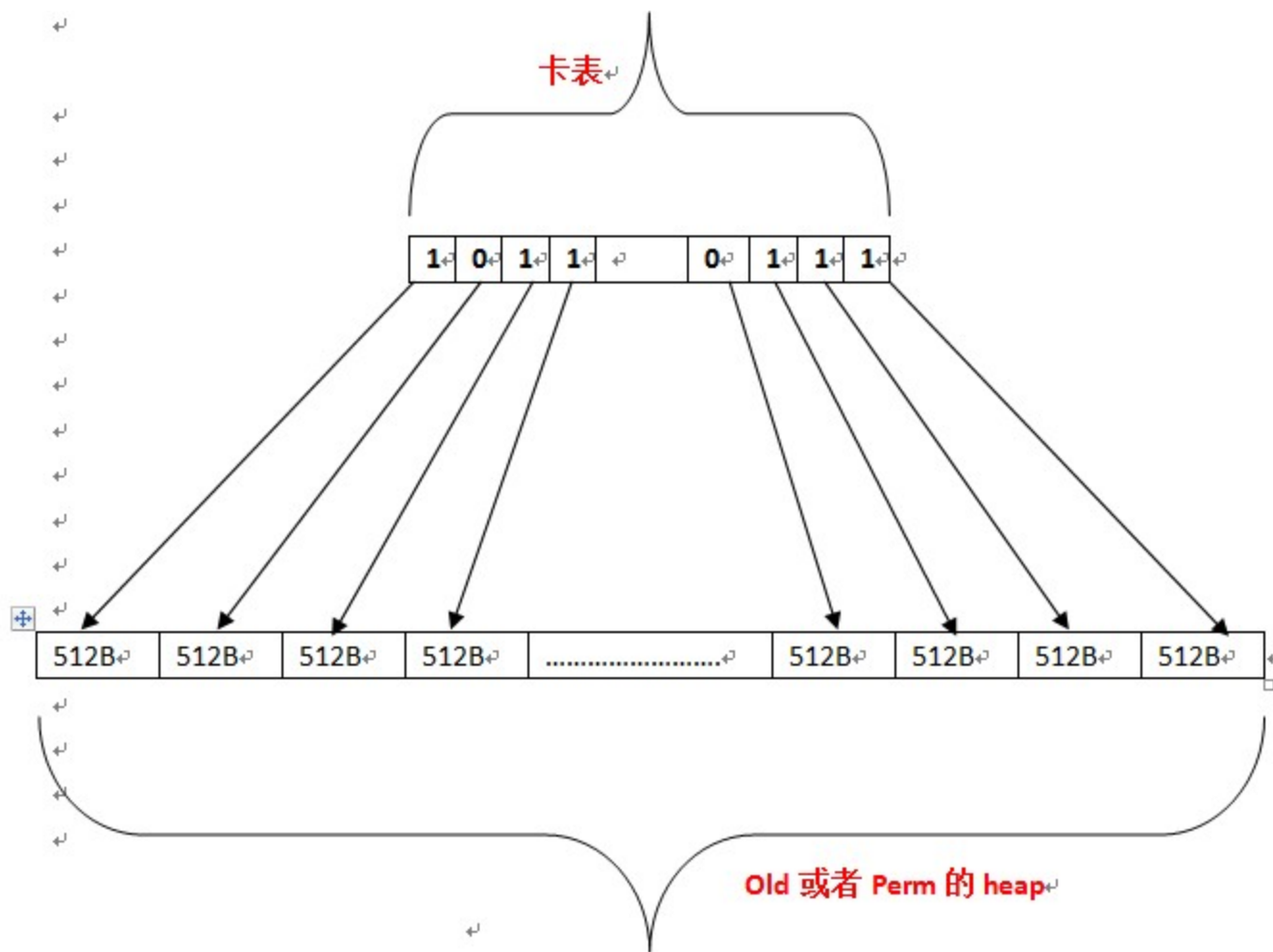
ParNew GC算法（对象遍历）



ParNew GC算法（卡表）

- 什么是Card Table（卡表）
- 卡表的内存分配
- 卡表与GC算法
- ParNew 对卡表的处理
 - 多线程优化
 - 卡表边界的处理（BOT表）

ParNew GC算法 (卡表)



ParNew GC算法（数据统计）

- 先看一个真实线上环境的[GC log](#)数据分析
- 从上面的log基本上可以得出
 - GC线程的工作量
 - 一次Minor GC中root遍历所有对象耗时少
 - 一次minor GC中处理CardTable的时间占大多数

目录

- GC算法介绍
- JVM对象模型
- JVM Heap的分代
- JVM内存分配
- ParNew GC算法
- **Promotion failure**
- ParNew 调优

Promotion Failure

- Promotion failure产生的原因
 - 堆的配置不合理
 - 相关GC参数配置不合理
 - 瞬间Load大量的数据
- Promotion failure发生后JVM的处理
 - Full GC
 - 长时间Stop the world
- 尽量避免Promotion failure
 - 合理设置CMSInitiatingOccupancyFraction
 - 和应用结合，优化应用代码，避免内存泄露

目录

- GC算法介绍
- JVM对象模型
- JVM Heap的分代
- JVM内存分配
- ParNew GC算法
- Promotion failure
- **ParNew 调优**

ParNew 调优

- 借助GC log和工具jstat等观察ParNew情况
 - ParNew频率多
 - 单次ParNew的耗时长
 - 每次ParNew后堆的变化
- 调优策略
 - Young大小的调整
 - SurvriorRatio的调整
 - 和CMS相结合来考虑
 - 优化应用，尽量避免new无谓的对象

谢谢！