

[译]Java8官方GC调优指南 --(七)并发收集器 - 掘金

 juejin.cn/post/6844904053546319879

2020年1月28日

本套文章是Java8官方GC调优指南的全文翻译，[点击查看原文](#)，原文章名称《Java Platform, Standard Edition HotSpot Virtual Machine Garbage Collection Tuning Guide》

7 The Mostly Concurrent Collectors 并发收集器

JDK 8提供两个并发收集器：

- Concurrent Mark Sweep (CMS) Collector:这个回收器倾向于降低gc停顿时间和共享处理器资源来加快gc。
- Garbage-First Garbage Collector:这个服务器级的回收器适用于多处理和超大内存。可以满足最大停顿时间和高吞吐量。

Overhead of Concurrent 并发带来的额外消耗

并发收集器使用多处理器资源来减少major collection的停顿时间。最显眼的消耗单处理器和多处理器在并发收集阶段的区别。在一个N核的操作系统上，并发收集阶段会使用K/N个核， $1 \leq K \leq \text{ceiling}\{N/4\}$ (N/4的上限)。除了在并发阶段使用处理器之外，还需要额外的开销来支持并发。因此，虽然并发收集器的垃圾收集暂停时间通常要短得多，但应用程序吞吐量也往往比其他收集器略低。

在一个多核的机器上，并发收集阶段还有其他的CPU核可以让应用继续运行，所以并发收集器线程不会暂停整个应用程序。这通常也反映在更短暂的停顿时间上，不过再次强调，应用程序正常运行时将会使用更少的CPU核数(因为有一些核给并发收集器用了)，所以程序会稍微慢一些。**CPU核数N越多，并发收集带来的吞吐量降低也就越少，并发收集器的收益也会越来越大。**Concurrent Mark Sweep (CMS) Collector中的章节Concurrent Mode Failure会讨论这种扩展中的潜在限制。

因为并发收集阶段会使用最少一个CPU核心，所以并发收集器在单核机器上不会有什么好处。不过，CMS有一个不受影响的模式可以在单核或双核CPU上获得更低的停顿时间；具体可以看Concurrent Mark Sweep (CMS) Collector 中的 Incremental Mode章节。这个特性在Java SE 8 被标记为过期，而且未来可能就要被干掉了。

我也觉得没什么用，单核就不用CMS了

扩展阅读

[The Garbage-First Garbage Collector](#)
[Garbage-First Garbage Collector Tuning](#)