codecraft 11.1k

聊聊jvm的PermGen与Metaspace

jvm 发布于 2017-12-25

序

本文主要讲述一下jvm的PermGen与Metaspace

java memory结构

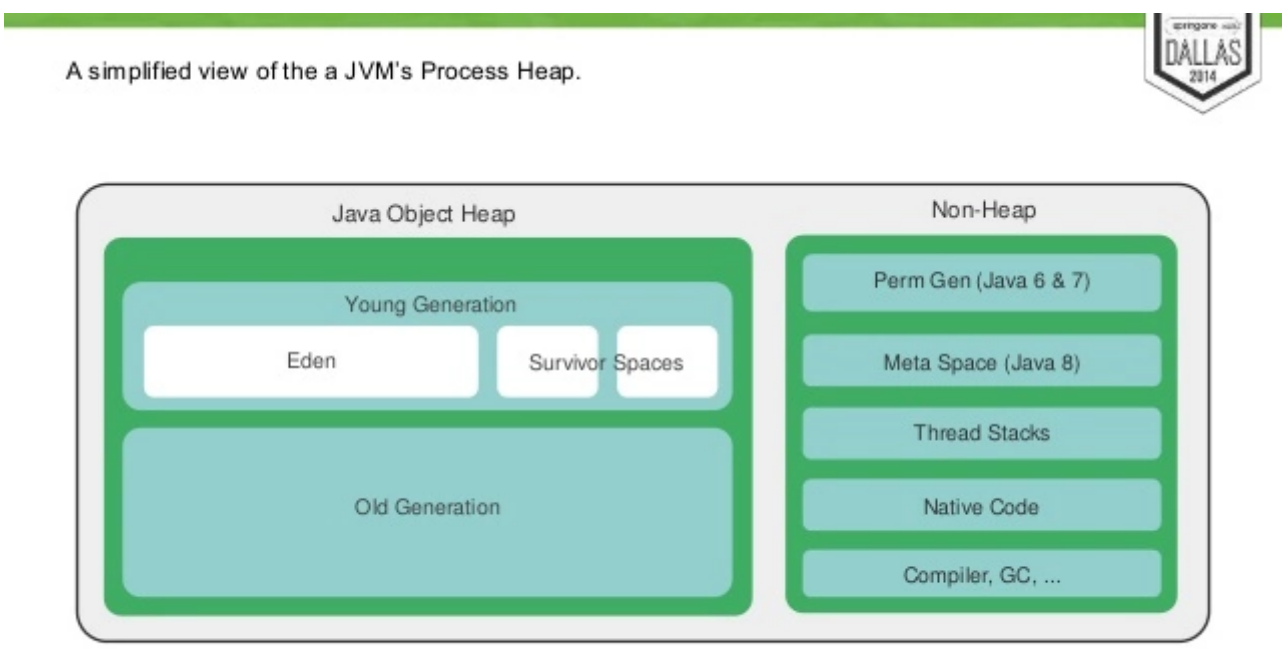
分代概念

对于垃圾收集算法来说，分代回收是高级算法之一。对象按照生成时间进行分代，刚刚生成不久的年轻对象划为新生代（Young gen-eration），而存活了较长时间的对象划为老生代（Old generation）。根据具体实现方式的不同，可能还会划分更多的代。比如有的把永久代也算做一个代。

memory划分

java memory主要分heap memory 和 non-heap memory，其计算公式如下：

```
Max memory = [-Xmx] + [-XX:MaxPermSize] + number_of_threads * [-Xss]
```



- heap结构

> 按分代，分young-eden,young-survivor,old
> 用-Xmn,-Xms,-Xmx来指定

- non-heap结构

> 包括metaspace,thread stacks,compiled native code,memory allocated by native code
>
> > -XX:PermSize或-XX:MetaspceSize,-Xss或-XX:ThreadStackSize

PermGen与Metaspace

字符串常量池的变化

- 在java7的时候将字符串常量池则移到java heap

> 所有的被intern的String被存储在PermGen区.PermGen区使用-XX:MaxPermSize=N来设置最大大小，但是由于应用程序string.intern通常是不可预测和不可控的，因此不好设置这个大小。设置不好的话，常常会引起

```
java.lang.OutOfMemoryError: PermGen space
```

- java7，8的字符串常量池在堆中实现

字符串常量池被限制在整个应用的堆内存中，在运行时调用String.intern()增加字符串常量不会使永久代OOM了。

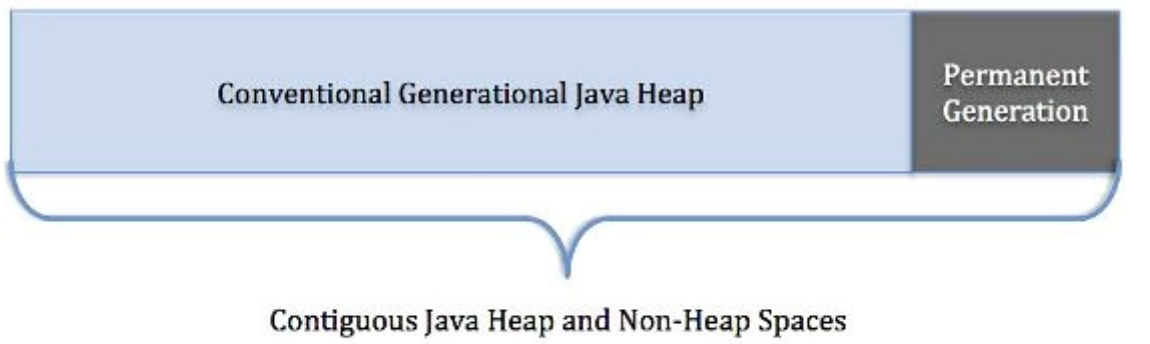
方法区的变化

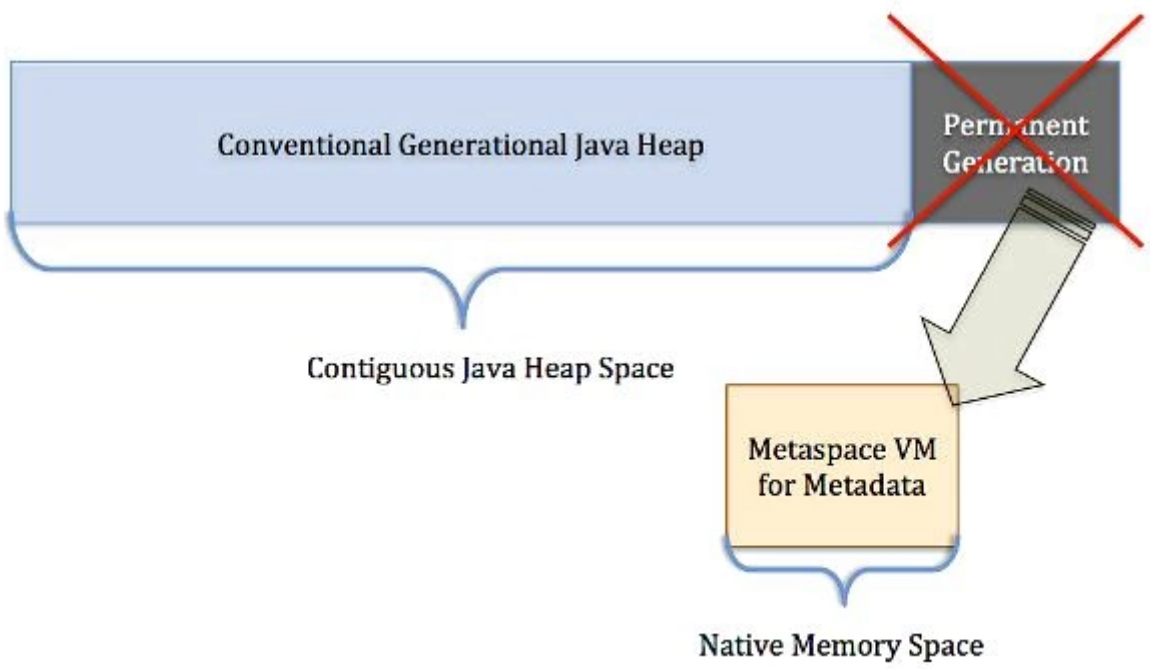
- java8的时候去除PermGen，将其中的方法区移到non-heap中的Metaspace

> move name and fields of the class, methods of a class with the bytecode
> of the methods, constant pool, JIT optimizations etc to metaspace

- Metaspace属于non-heap

> Metaspace与PermGen之间最大的区别在于：Metaspace并不在虚拟机中，而是使用本地内存。





> 如果没有使用-XX:MaxMetaspaceSize来设置类的元数据的大小，其最大可利用空间是整个系统内存的可用空间。JVM也可以增加本地内存空间来满足类元数据信息的存储。
> 但是如果没有设置最大值，则可能存在bug导致Metaspace的空间在不停的扩展，会导致机器的内存不足；进而可能出现swap内存被耗尽；最终导致进程直接被系统直接kill掉。

- OOM异常

如果类元数据的空间占用达到MaxMetaspaceSize设置的值，将会触发对象和类加载器的垃圾回收。

```
java.lang.OutOfMemoryError: Metaspace space
```

> JVM从Metaspace在捕获一个一个内存分配失败后抛出。

Metaspace相关参数

- -XX:MetaspaceSize，初始空间大小，达到该值就会触发垃圾收集进行类型卸载，同时GC会对该值进行调整：如果释放了大量的空间，就适当降低该值；如果释放了很少的空间，那么在不超过MaxMetaspaceSize时，适当提高该值。
- -XX:MaxMetaspaceSize，最大空间，默认是没有限制的。
- -XX:MinMetaspaceFreeRatio，在GC之后，最小的Metaspace剩余空间容量的百分比，减少为分配空间所导致的垃圾收集
- -XX:MaxMetaspaceFreeRatio，在GC之后，最大的Metaspace剩余空间容量的百分比，减少为释放空间所导致的垃圾收集

小结

将常量池从PermGen剥离到heap中，将元数据从PermGen剥离到元数据区，去除PermGen的好处如下：

- 将字符串常量池从PermGen分离出来，与类元数据分开，提升类元数据的独立性
- 将元数据从PermGen剥离出来到Metaspace，可以提升对元数据的管理同时提升GC效率。

> 在PermGen中元数据可能会随着每一次Full GC发生而进行移动。HotSpot虚拟机的每种类型的垃圾回收器都需要特殊处理PermGen中的元数据，分离出来以后可以简化Full GC以及对以后的并发隔离类元数据等方面进行优化。

- 为后续将HotSpot与JRockit合二为一做准备。

> PermGen是HotSpot的实现特有的，JRockit并没有PermGen一说

doc

- Java8内存模型—永久代(PermGen)和元空间(Metaspace)
- JVM内存调优相关的一些笔记（杂）
- Java PermGen 去哪里了
- 一个Tomcat配置参数引发的血案
- Java6,7,8中的String.intern() – 字符串常量池
- 升级Java8可能会踩到的坑

阅读 20k • 更新于 2017-12-25

赞 11 收藏 14 ¥ 赞赏 分享

本作品系原创， 采用《署名-非商业性使用-禁止演绎 4.0 国际》许可协议

codecraft
11.1k

关注作者

3 条评论

得票 • 时间

撰写评论 ...

提交评论

哇哦： 大佬产出好高。。可惜没目录分类，文章太多了。。

• 回复 • 2019-05-06

zazaluMonster： 感谢总结!
鄙人最近写了一篇Java的OOM异常总集篇! 对各类OOM异常都做了分析总结,希望可以帮助到最近遇到这类问题的小伙伴
https://zazalu.space/2019/09/...

• 回复 • 2019-09-19

只道： 你好，可以说一下，文中：move name and fields of the class, methods of a class with the bytecode of the methods, constant pool, JIT optimizations etc to metaspace，这句话的出处吗？感谢

回复 • 4月18日

推荐阅读

【深入浅出-JVM】（序）

本系列主要是让一个刚入门的 java 开发者，也能愉快的从零开始成为一个真正的 jvm 大神。 大纲 java 虚拟机的定义、总体架构、...

mousycoder • 阅读 1.5k • 5 赞

node-jvm —— 纯node.js 的 JVM 实现

node-jvm，使用纯node.js实现了JVM，太酷了！ 例如，下面是一段生成斐波纳契数列并计算生成时间的Java代码： {代码...} 将它...

思否编辑部 • 阅读 4.5k • 2 赞

两幅图帮你搞定JVM和JMM

仅此两幅图、希望对你理解JVM 和JMM 有帮助。如有错误欢迎指正。如感觉图片不清晰、请移步到 >>>>>> 这里有我对JVM 中...

阅历笔记 • 阅读 728 • 2 赞

JVM脑图总结

xmind源文件：[链接]

devon • 阅读 586 • 2 赞

JVM系列分享1-类加载

Henry • 阅读 163 • 1 赞

JVM内存模型

上图 jvm内存模型，虚拟的描述java操作内存的一个模型。java跨平台：jvm将每一条指令翻译成不同平台的机器码。 类加载机制 J...

李沁春 • 阅读 160 • 1 赞

JVM运行时数据区域

参考资料 Java Virtual Machine Specification | 2.5. Run-Time Data Areas Java Virtual Machine Specification | 5.3. Creation and ...

chanjarster • 阅读 376 • 1 赞

JVM启动参数

标准参数（-），所有的JVM实现都必须实现这些参数的功能，而且向后兼容； 非标准参数（-X），默认jvm实现这些参数的功能...

东瓜 • 阅读 449 • 1 赞

code-craft

用户专栏

spring boot , docker and so on 欢迎关注微信公众号: geek_luandun

708 人关注 **1816** 篇文章

关注专栏 专栏主页

热门专栏 PHP 开发课程 用户排行榜 广告投放 社区运营日志 隐私政策
热门课程 Python 开发课程 徽章 职位发布 市场运营日志 下载 App
最新活动 前端开发课程 帮助中心 讲师招募 团队日志
技术圈 移动开发课程 声望与权限 联系我们 社区访谈
酷工作 社区服务中心 合作伙伴
移动客户端

Copyright © 2011-2020 SegmentFault.
浙ICP备 15005796号-2 浙公网安备 33010602002000号 杭州堆栈科技有限公司版权所有
iTrust 安全验证