

JVM相关

笔记本： 工作笔记

创建时间： 2020/11/4 19:30

更新时间： 2020/11/5 16:13

作者： 438842220@qq.com

URL： about:blank

JMM/类加载器/垃圾回收/java内存区域/调优

JMM:

<https://www.jianshu.com/p/8a58d8335270>

现代计算机的内存模型：

为了平衡CPU和内存之间速度的矛盾引入了缓存。从而引发了缓存一致性问题，当多个线程共享同一块内存时这个问题就会出现。为了解决这个问题，就需要缓存一致性协议。

为了平衡写的速度，引入了写缓冲区。但写缓冲区只对自己可见。因此可能导致读写的顺序不一致。

java内存模型（JMM）：

JMM是隶属于JVM的，定义了JVM在内存中的工作方式。

JAVA线程 -----工作内存-----save/load操作-----主内存

JAVA线程 -----工作内存-----↑

JAVA线程 -----工作内存-----↑

JAVA线程 -----工作内存-----↑

栈区内的(原始类型、对象指针)
线程独占、线程共享 不是

本身就是一个, 不存在同步等问题

变量有: 对象、对象的任何成员

传值 → not 修 变量、Static

共享以上, 须注意并发时的读 RW、WW、RW 问题。

因此: Synchronized, Synchronized.

要以类、Static、对象成员为入参。

修饰: 需要这些的地方: 代码块 or 方法

因此 volatile 修饰的: 类和类内的变量
Static.

而: 方法内的对象指针, 原始类型不必修饰。

类加载器:

我们需要了解JAVA字节码定义, 以及有了字节码之后的类加载流程。

然后再去了解类加载器的树形结构。最后自己定义类加载器。这样就基本完成了。

垃圾回收:

JAVA内存区域:

调优:

