Dongguabai
码龄6年 暂无认证

567	2万+	2万+	302万+	
原创	周排名	总排名	访问	等级
2万+	340	709	552	2291
积分	粉丝	获赞	评论	收藏

私信 关注

搜博主文章

热门文章

解决由于ntoskrnl.exe导致的蓝屏（Win10） 267161

Linux查看历史命令 69195

HTTP请求返回304状态码 55835

JDBC Connection XXX will not be managed by Spring 49580

SpringBoot自动装配原理分析 49311

最新评论

高并发（水平扩展，垂直扩展）
m0_51967950: 讲的不错，通俗易懂

原生AspectJ用法分析以及Spring-AOP原...
z103566940: 孔乙己应该脱下长衫

基于 CAS 操作实现一个简单的锁
Dongguabai: 这只是个简单的Demo，诸如ABA问题这里暂时不考虑

基于 CAS 操作实现一个简单的锁
毛毛虫呜呜: 有个问题：如果线程t1,t2同时去获取锁，就会同时执行compareandset...

JVM安全退出（如何优雅的关闭java服务）
迷路剑客: 为啥不直接注册shutdownHook

您愿意向朋友推荐“博客详情页”吗？

强烈不推荐 不推荐 一般般 推荐 强烈推荐

最新文章

永远不要使用双花括号初始化实例，否则就会OOM！

MySQL 同时删除和添加多个字段

分布式集群数据同步之 ZooKeeper

2023年 1篇	2022年 35篇
2021年 163篇	2020年 171篇
2019年 180篇	2018年 480篇

Java8中@Contended和伪共享

转载 Dongguabai 于 2020-12-29 16:20:39 发布 835 收藏 版权
分类专栏： jvm &amp; jdk

jvm & jdk 专栏收录该内容 2 订阅 73 篇文章 订阅专栏

Java8中@Contended和伪共享

> Java8引入了@Contented这个新的注解来减少伪共享(False Sharing)的发生。本文介绍了@Contented注解并解释了为什么False Sharing是如何影响性能的。

缓存行

> CPU读取内存数据时并非一次只读一个字节，而是会读一段64字节长度的连续的内存块(chunks of memory)，这些块我们称之为缓存行(Cache line)。

- 假设你有两个线程(Thread1和Thread2)都会修改同一个volatile变量x:

```
volatile long x;
```

- 如果Thread1先改变x的值，然后Thread2又去读它：

- 那么x所在缓存行上的所有64个字节的值都要被重新加载，因为CPU核心间交换数据是以缓存行为最小单位的。当然Thread1和Thread2是有可能在同一个核心上运行的，但我们此处假设两个线程在不同的核心上运行。
- 已知long类型占8个字节，缓存行长度为64个字节，那么一个缓存行可以保存8个long型变量，我们已经有了一个long型的x，假设x所在缓存行里还有其他7个long型变量，v1到v7:x, v1, v2, v3, v4, v5 ,v6 ,v7.

伪共享(False Sharing)

> 这个缓存行可以被许多线程访问。如果其中一个修改了v2，那么会导致Thread1和Thread2都会重新加载整个缓存行。你可能会疑惑为什么修改了v2会导致Thread1和Thread2重新加载该缓存行，毕竟只是修改了v2的值啊。虽然说这些修改逻辑上是互相独立的，但同一缓存行上的数据是统一维护的，一致性的粒度并非体现在单个元素上。这种不必要的数据共享就称之为“伪共享”(False Sharing)。

填充(Padding)

> 一个CPU核心在加载一个缓存行时要执行上百条指令。如果一个核心要等待另外一个核心来重新加载缓存行，那么他就必须等在那里，称之为stall(停止运转)。减少伪共享也就意味着减少了stall的发生，其中一个手段就是通过填充(Padding)数据的形式，来保证本应有可能位于同一个缓存行的两个变量，在被多线程访问时必定位于不同的缓存行。

```java
public class FalseSharingWithPadding {
    public volatile long x;
    public volatile long p2; // padding
    public volatile long p3; // padding
    public volatile long p4; // padding
    public volatile long p5; // padding
    public volatile long p6; // padding
    public volatile long p7; // padding
    public volatile long p8; // padding
    public volatile long v1;
}

```

> 在你考虑使用填充之前，必须要了解的一点是JVM可能会清除无用字段或重排无用字段的位置，这样的话，可能无形中又会引入伪共享。我们也没有办法指定对象在堆内驻留的位置。

> 为了避免无用字段被消除，通常我们会用volatile修饰一下。个人建议只需为处于激烈竞争状态的类进行填充处理，而且一般只有通过对其性能分析才能发现性能上的不同。通常在性能分析时，最好在对其迭代访问10000次之后再去采样，这样可以消除JVM本身的运行时优化策略带来的影响。

Java8和@Contended

> 除了对字段进行填充之外，还有一个比较清爽的方法，那就是对需要避免陷入伪共享的字段进行注解，这个注解暗示JVM应当将字段放入不同的缓存行，这也正是JEP142的相关内容。

- 该JEP引入了@Contented注解。被这个注解修饰的字段应当和其他的字段驻留在不同的位置。

```java
    public class Point {
        int x;
        @Contended
        int y;
    }

```

> 上面的代码将x和y置于不同的缓存行。@Contented注解将y移动到远离对象头部的地方，(以避免和x一起被加载到同一个缓存行)。

文章知识点与官方知识档案匹配，可进一步学习相关知识

Java技能树 > 首页 > 概览 110419 人正在系统学习中

【并发编程】— 伪共享（False Sharing）底层原理及其解决方式 12-22
文章目录1 什么是伪共享1.1 从疑问说起 — 难道JMM中不存在局部性原理？？？ 1.2 缓存行的概念1.3 伪共享（False Sharing）的概念 + 其可能引发的性...

缓存--伪共享问题 最新发布 qq_33014341的博客 247
伪共享问题指两个线程A和B，他们俩写入同一个cache block的不同变量时，会导致另一个cpu核心的缓存失效的问题。 解决伪共享有两种办法 一般而言...

Java8中@Contended和伪共享 进行缓存行填充_我的编程备忘录的博客-CSDN... 3-23
Java8引入了@Contented这个新的注解来减少伪共享(False Sharing)的发生。本文介绍了@Contented注解并解释了为什么False Sharing是如何影响性能...

对齐填充@Contended_Lucifer Zhao的博客 4-2
Idea中配置开启@Contented: 1) idea工具栏中 Help --> Edit Custom Vm Options 2) -XX参数中 boolean类型, -XX:+RestrictContended为该属性设置为true,...

真实字节二面：什么是伪共享？ 艾小仙的专栏 318
这个问题来自最近一个朋友字节面试碰到的，最后他也成功拿到了字节offer，这个问题我想可能挺多人不太清楚，所以想拿出来单独说一说。 好了，让我...

聊聊java8中的@sun.misc.Contended与伪共享 dhaibo1986的专栏 299
文章目录1.伪共享1.1 CPU的缓存机制1.2 缓存行和伪共享2.代码验证2.1 不做任何处理2.2 缓存填充2.3 通过Contended3.总结 在前面学习ConcurrentHash...

慢慢学JVM之缓存行和伪共享_青铜大神的博客 3-24
第二种办法,使用JDK8新增的@Contented,使用@Contented注解后会增加128字节的padding,需要启动时增加-XX:-RestrictContented选项才能生效。

如何使用@Contented关键字避免伪缓存行与调优与解决Idea注解报错问题 一瓶绿茶三元钱 660
如何使用@Contented关键字避免伪缓存行与调优与解决Idea注解报错问题，idea@Contented报错。 1.注解概述 1.1使用范围 Contented注解的Target包含...

Java虚拟机初学习（二） qq_43357821的博客 101
Java对象的内存布局 对象头 每个对象都有一个对象头，对象头包括两部分，标记信息和类型指针。 标记信息包括哈希值，锁信息，GC信息。类型指针指...

Java8的@sun.misc.Contended注解解决伪共享问题 dongcheng_2015的博客 266
本文源自转载：Java8的@sun.misc.Contended注解 目录 一、@sun.misc.Contended 介绍 二、单独使用一个缓存行有什么作用——避免伪共享 三、@su...

Java8使用@sun.misc.Contended避免伪共享 04stone37 396
转载：http://www.ideabuffer.cn/2017/05/12/Java8%E4%BD%BF%E7%94%A8-sun-misc-Contended%E9%81%BF%E5%85%8D%E4%BC%AA%E5%85...

java jdk注解 使用_JDK中注解@Contended是干什么的？怎么使用? weixin_34840783的博客 421
避免伪共享(false sharing)。先引用个伪共享的解释：缓存系统中是以缓存行(cache line)为单位存储的。缓存行是2的整数幂个连续字节，一般为32-256个...

性能优化之@Contended减少伪共享 西魏陶渊明的博客 257
说到伪共享,就要说CPU缓存,我们程序执行时候信息会被保存到CPU缓存中 而这些缓存中的数据可能被多线程访问,假如一个线程还没处理完

sass语法中的@content xqhys的博客 3233
@content这个是不是个类似占位用的，代表里面是自定义的的 @-webkit-keyframes #{$name} { @content; } `@content`用在`mixin`里面的，当定义一个`...

react中@content使用总结 Y_Cfeng的博客 1156
文章目录前言引入@connect是什么呢？？？ @connect介绍connect介绍类的装饰基本语法connect的注入定义模块区域 state 数据建立连接获取指定模块...

Disruptor以及@Contended注解 节至 3863
Disruptor感想 很早之前阅读过Disruptor的使用，这里有篇美团团队的文章很详细的介绍了这种队列的相关原理以及为什么这么设计和这么设计为什么处理...

高级装配 32
环境与profile 为了使一些 Bean 在特定情况注册而引入机制，若是使用不同的 XML 配置要为每种环境重新构建，也可能引入 BUG,使用 profile 则不会，因...

springmvc注解 contentJava的博客 337
一、@controller 用来表示一个web控制层bean，用来前后端的交互！类似struts的action或servlet 二、@service 用来表示一个业务层bean 三、@repos...

Http请求中Content-Type讲解以及在Spring MVC注解中produce和consumes配置详解 热门推荐 qq1164014750 1万+
转载于：https://blog.csdn.net/shinebar/article/details/54408020引言： 在Http请求中，我们每天都在使用Content-type来指定不同格式的请求信息，但是...

“相关推荐”对你有帮助么？

非常没帮助 没帮助 一般 有帮助 非常有帮助

目录

分类专栏

log	3篇
skywalking	2篇
elk	18篇
arthas	2篇
nacos	1篇
activiti	1篇
tool（dev）	7篇
feign	1篇
xa	4篇
atomikos	1篇
wrk	1篇
encryption	1篇
redission	
io	1篇
netty	1篇
eureka	4篇
socket	3篇
tool	1篇
aspectj	1篇
http	2篇
os	3篇
excel	1篇
lua	4篇
hystrix	1篇
browser	3篇
google	1篇
java 9	1篇
thymeleaf	2篇
safe	2篇
spark	1篇
c	2篇
gradle	
deliberate	3篇
undetermine	82篇
optimization	10篇
solr	9篇
view	27篇
exception	108篇
docker	2篇
spring boot	64篇
linux	43篇
r	1篇
jvm & jdk	73篇
spring security	8篇
redis	14篇
concurrent（j.u.c）	82篇
data structure & arithmetic	11篇
web server	17篇
zookeeper	39篇
maven	34篇
distribute & micro service	39篇
serialize	20篇
windows	6篇
web service	5篇
design pattern	12篇
dubbo	22篇
ide	54篇
version control	18篇
network protocol	8篇
servlet	13篇
testing	2篇
shiro	1篇
node.js	3篇
spring	60篇
spring mvc	25篇
mybatis	41篇
oracle	27篇
database connection pool	2篇
mysql	61篇
interview	9篇
class	5篇
activemq	4篇
rocketmq	3篇
kafka	10篇
nginx	2篇
guava	2篇
security	1篇
java 8	10篇
arithmetic	2篇
mac os	26篇
concurrent（solution）	1篇
office	1篇
vue	6篇
python	4篇
hibernate	4篇
jetty	2篇
ffmpeg	1篇
c++	