

并发编程网 - ifeve.com

让天下没有难学的技术

MAR 2013

09

42,169 人阅读

古圣昌

并发译文

★★★★★

(16 votes, average: 4.81 out of 5)

9 comments

多线程的代价

原文链接：<http://tutorials.jenkov.com/java-concurrency/costs.html>

作者：[Jakob Jenkov](#) 翻译：古圣昌 校对：欧振聪

从一个单线程的应用到一个多线程的应用并不仅仅带来好处，它也会有一些代价。不要仅仅为了使用多线程而使用多线程。而应该明确在使用多线程时能多来的好处比所付出的代价大的时候，才使用多线程。如果存在疑问，应该尝试测量一下应用程序的性能和响应能力，而不只是猜测。

设计更复杂

虽然有一些多线程应用程序比单线程的应用程序要简单，但其他的一般都更复杂。在多线程访问共享数据的时候，这部分代码需要特别的注意。线程之间的交互往往非常复杂。不正确的线程同步产生的错误非常难以被发现，并且重现以修复。

上下文切换的开销

当CPU从执行一个线程切换到执行另外一个线程的时候，它需要先存储当前线程的本地的数据，程序指针等，然后载入另一个线程的本地数据，程序指针等，最后才开始执行。这种切换称为“上下文切换”(“context switch”)。CPU会在一个上下文中执行一个线程，然后切换到另外一个上下文中执行另外一个线程。

上下文切换并不廉价。如果没有必要，应该减少上下文切换的发生。

你可以通过维基百科阅读更多的关于上下文切换相关的内容：

http://en.wikipedia.org/wiki/Context_switch

增加资源消耗

线程在运行的时候需要从计算机里面得到一些资源。除了CPU，线程还需要一些内存来维持它本地的堆栈。它也需要占用操作系统中一些资源来管理线程。我们可以尝试编写一个程序，让它创建100个线程，这些线程什么事情都不做，只是在等待，然后看看这个程序在运行的时候占用了多少内存。

原创文章，转载请注明： 转载自[并发编程网 – ifeve.com](#) **本文链接地址：**
[多线程的代价](#)

热门文章

[Google Guava官方教程（中文版）](#) 581,149 人阅读

[Java NIO系列教程（一）Java NIO 概述](#) 403,309 人阅读

[Java并发性和多线程介绍目录](#) 281,400 人阅读

[Java NIO 系列教程](#) 267,318 人阅读

[Java NIO系列教程（十二）Java NIO与IO](#) 226,295 人阅读

[Java8初体验（二）Stream语法详解](#) 207,619 人阅读

[Java NIO系列教程（六）Selector](#) 201,199 人阅读

[Java NIO系列教程（三）Buffer](#) 197,245 人阅读

[Java NIO系列教程（二）Channel](#) 195,423 人阅读

[《Storm入门》中文版](#) 177,558 人阅读

[69道Spring面试题和答案](#) 167,383 人阅读

[Netty 5用户指南](#) 161,710 人阅读

[面试题](#) 145,543 人阅读

[并发框架Disruptor译文](#) 144,531 人阅读

[Java 7 并发编程指南中文版](#) 138,649 人阅读

[Java NIO系列教程（八）SocketChannel](#) 128,284 人阅读

[\[Google Guava\] 3-缓存](#) 122,684 人阅读

[\[Google Guava\] 2.3-强大的集合工具类：ja...](#) 121,994 人阅读

[\[Google Guava\] 1.1-使用和避免null](#) 112,340 人阅读

[Java NIO系列教程（七）FileChannel](#) 110,871 人阅读

并发编程网 | 让天下没有难学的技术



长按，识别二维码，加关注

微信号: ifeves

About

Latest Posts



古圣昌

BGI 开发工程师

★[添加本文到我的收藏](#)

Related Posts:

1. [Java并发性和多线程介绍目录](#)
2. [看动画学并发编程](#)
3. [CPU缓存刷新的误解](#)
4. [剖析Disruptor:为什么会这么快？\(一\)锁的缺点](#)
5. [剖析Disruptor:为什么会这么快？（二）神奇的缓存行填充](#)
6. [聊聊我对Java内存模型的理解](#)
7. [Disruptor 2.0更新摘要](#)
8. [Java并发性和多线程介绍](#)
9. [有助于减少伪共享的@Contended注解](#)
10. [Netty源码解读（三）Channel与Pipeline](#)
11. [剖析Disruptor:为什么会这么快？（一）Ringbuffer的特别之处](#)
12. [CPU Cache Flushing Fallacy](#)
13. [Java NIO系列教程（一）Java NIO 概述](#)
14. [从JVM并发看CPU内存指令重排序\(Memory Reordering\)](#)
15. [通过Axon和Disruptor处理1M tps](#)

Write comment

Comments RSS

Trackback are closed

Comments (9)



夕水溪下

03/12. 2013 10:16am

[Log in to Reply.](#) | [QUOTE](#)

之前有过测试，如果在本地同时创建500个线程，运行后，eclipse会直接挂掉。

[Log in to Reply.](#) | [QUOTE](#)

RECENT POSTS

[Dubbo-从入门到深入](#)

[Leader-Follower线程模型概述](#)

[《Apache Thrift官方文档》简介](#)

[《RabbitMQ官方指南》安装指南](#)

[在Windows上安装RabbitMQ](#)

[动手实现一个 LRU cache](#)

[《Thrift官方文档》Thrift支持的语言](#)

[《Thrift官方文档》– docker构建说明](#)

[浅尝一致性Hash原理](#)

[Dubbo剖析-线程模型](#)

[分布式理论：CAP是三选二吗？](#)

[Jarslink1.6.1版本特性](#)

[《深入分布式缓存》之“缓存为王”](#)

[《Thrift官方文档》翻译邀请](#)

[《Apache RocketMQ用户指南》之定时消息示例](#)

[使用Spring框架实现远程服务暴露与调用](#)

[Dubbo剖析-服务消费方Invoker到客户端接口的转换](#)

[Dubbo剖析-服务消费方远程服务到Invoker的转换](#)

[Linux零拷贝原理](#)

[阿里再开源！模块化开发框架JarsLink](#)

[Dubbo剖析-服务提供方Invoker到Exporter的转换](#)

[Dubbo剖析-服务提供方实现类到Invoker的转换](#)

[Dubbo剖析-增强SPI中扩展点自动包装的实现](#)

[Dubbo剖析-服务消费端异步调用](#)

[Dubbo剖析-服务直连](#)

[Dubbo剖析-服务分组与服务版本号](#)

[Dubbo剖析-监控平台的搭建与使用](#)

[Dubbo剖析-增强SPI的实现](#)

[Dubbo剖析-整体架构分析](#)

[《Linkerd官方文档》在ECS中运行Linkerd](#)

CATEGORIES

[Android](#) (3)

[C++](#) (12)

[CPU](#) (2)

[Framework](#) (72)

[akka](#) (20)

[GO](#) (6)



方 腾飞
03/13. 2013 8:51am

起500个线程，如果每个线程启动时需要1m内存，jvm可能内存溢出。



vavio99
03/18. 2013 8:49pm

[Log in to Reply.](#) | [QUOTE](#)

可以试试jvm的-Xss参数



疯子_无醉
03/19. 2013 2:46pm

[Log in to Reply.](#) | [QUOTE](#)

”

vavio99 :
可以试试jvm的-Xss参数

”

设大了太占内存了。



Ulric Qin
05/18. 2013 8:05pm

[Log in to Reply.](#) | [QUOTE](#)

虽然有一些多线程应用程序比单线程的应用程序要简单，但其他的一般都更复杂。

所以说，虽然现在是在学习多线程编程，但是能不用多线程还是少用，除非真的有显著效果，否则光调试以及处理莫名其妙的问题就够受的了



E网情深
10/21. 2013 5:59pm

[Log in to Reply.](#) | [QUOTE](#)

让它创建100个线程，这些线程什么事情都不做，只是在等待

这里等待指的什么意思？不是sleep吧，还是这样？ new Thread(new

```
Runnable(){  
  
public void run(){  
  
}  
  
})
```

谢了



packagecomponent
03/08. 2014 11:51am

[Log in to Reply.](#) | [QUOTE](#)

如果是同一个进程中的线程的切换，开销比较小。不同进程间的线程切换，开销比较大，上文这一点没有说明



fangqiang08
09/12. 2014 9:20pm

[Log in to Reply.](#) | [QUOTE](#)

请问编程中怎么判断、设置线程属于某个进程？



forward
01/23. 2018 4:33pm

[Log in to Reply.](#) | [QUOTE](#)

[groovy](#) (6)

[guava](#) (23)

[JAVA](#) (824)

[JVM](#) (40)

[linux](#) (9)

[microservices](#) (1)

[Netty](#) (31)

[react](#) (6)

[redis](#) (23)

[Scala](#) (11)

[spark](#) (19)

[Spring](#) (23)

[storm](#) (44)

[thinking](#) (3)

[Velocity](#) (10)

[Web](#) (18)

[zookeeper](#) (1)

[公告](#) (5)

[大数据](#) (33)

[好文推荐](#) (31)

[并发书籍](#) (97)

[并发译文](#) (410)

[感悟](#) (3)

[技术问答](#) (12)

[敏捷管理](#) (6)

[本站原创](#) (87)

[架构](#) (32)

[活动](#) (6)

[网络](#) (7)

TAGS

[actor](#) [Basic](#) [classes](#) [collections](#)
[concurrency](#) [Concurrent](#) [concurrent](#)
[data](#) [structure](#) [Customizing](#) [Executor](#)
[Executor](#) [framework](#) [False](#) [Sharing](#) [faq](#) [fork](#)
[Fork/Join](#) [fork](#) [join](#) [Framework](#) [Functional](#)
[Programming](#) [Guava](#) [IO](#) [JAVA](#) [java8](#)
[jmm](#) [join](#) [JVM](#) [lock](#) [Memory](#) [Barriers](#) [Netty](#)
[NIO](#) [OAuth](#) [2.0](#) [pattern-matching](#) [RingBuffer](#) [Scala](#)
[service](#) [mesh](#) [slf4j](#) [spark](#) [spark官方文档](#) [stm](#)
[Storm](#) [synchronization](#) [Synchronized](#)
[thread](#) [tomcat](#) [volatile](#) [多线程](#) [并发译文](#)
[文](#) , [Java](#) , [Maven](#)

不正确的线程同步产生的错误非常难以被发现，并且重现以修复。

这句的翻译有点拗口

You must be [logged in](#) to post a comment.

重入锁死

java并发面试题(一)基础

