并发编程网 - ifeve.com
让天下没有难学的技术

MAR 07
2013

多线程的优点

61,016 人阅读

古 圣昌

并发译文

(27 votes, average: 4.89 out of 5)

11 comments

原文：http://tutorials.jenkov.com/java-concurrency/benefits.html

作者：Jakob Jenkov 翻译：古圣昌 校对：欧振聪

尽管面临很多挑战，多线程有一些优点使得它一直被使用。这些优点是：

资源利用率更好

程序设计在某些情况下更简单

程序响应更快

资源利用率更好

想象一下，一个应用程序需要从本地文件系统中读取和处理文件的情景。

比方说，从磁盘读取一个文件需要5秒，处理一个文件需要2秒。处理两个文件则需要：

```
5秒读取文件A
2秒处理文件A
5秒读取文件B
2秒处理文件B
---------------------
总共需要14秒
```

从磁盘中读取文件的时候，大部分的CPU时间用于等待磁盘去读取数据。在这段时间里，CPU非常的空闲。它可以做一些别的事情。通过改变操作的顺序，就能够更好的使用CPU资源。看下面的顺序：

```
5秒读取文件A
5秒读取文件B + 2秒处理文件A
2秒处理文件B
---------------------
总共需要12秒
```

CPU等待第一个文件被读取完。然后开始读取第二个文件。当第二文件在被读取的时候，CPU会去处理第一个文件。记住，在等待磁盘读取文件的时候，CPU大部分时间是空闲的。

总的说来，CPU能够在等待IO的时候做一些其他的事情。这个不一定就是磁盘IO。它也可以是网络的IO，或者用户输入。通常情况下，网络和磁盘的IO比CPU和内存的IO慢的多。

程序设计更简单

在单线程应用程序中，如果你想编写程序手动处理上面所提到的读取和处理的顺序，你必须记录每个文件读取和处理的状态。相反，你可以启动两个线程，每个线程处理一个文件的读取和操作。线程会在等待磁盘读取文件的过程中被阻塞。在等待的时候，其他的线程能够使用CPU去处理已经读取完的文件。其结果就是，磁盘总是在繁忙地读取不同的文件到内存

热门文章

Google Guava官方教程（中文版） 581,155 人阅读

Java NIO系列教程（一） Java NIO 概述 403,316 人阅读

Java并发性和多线程介绍目录 281,406 人阅读

Java NIO 系列教程 267,320 人阅读

Java NIO系列教程（十二） Java NIO与IO 226,296 人阅读

Java8初体验（二）Stream语法详解 207,624 人阅读

Java NIO系列教程（六） Selector 201,200 人阅读

Java NIO系列教程（三） Buffer 197,246 人阅读

Java NIO系列教程（二） Channel 195,425 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 161,711 人阅读

面试题 145,544 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 138,651 人阅读

Java NIO系列教程（八） SocketChannel 128,285 人阅读

[Google Guava] 3-缓存 122,685 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 112,341 人阅读

Java NIO系列教程（七） FileChannel 110,873 人阅读

中。这会带来磁盘和CPU利用率的提升。而且每个线程只需要记录一个文件，因此这种方式也很容易编程实现。

程序响应更快

将一个单线程应用程序变成多线程应用程序的另一个常见的目的是实现一个响应更快的应用程序。设想一个服务器应用，它在某一个端口监听进来的请求。当一个请求到来时，它去处理这个请求，然后再返回去监听。

服务器的流程如下所述：

```
while(server is active){
    listen for request
    process request
}
```

如果一个请求需要占用大量的时间来处理，在这段时间内新的客户端就无法发送请求给服务端。只有服务器在监听的时候，请求才能被接收。另一种设计是，监听线程把请求传递给工作者线程(worker thread)，然后立刻返回去监听。而工作者线程则能够处理这个请求并发送一个回复给客户端。这种设计如下所述：

```
while(server is active){
    listen for request
    hand request to worker thread
}
```

这种方式，服务端线程迅速地返回去监听。因此，更多的客户端能够发送请求给服务端。这个服务也变得响应更快。

桌面应用也是同样如此。如果你点击一个按钮开始运行一个耗时的任务，这个线程既要执行任务又要更新窗口和按钮，那么在任务执行的过程中，这个应用程序看起来好像没有反应一样。相反，任务可以传递给工作者线程（word thread)。当工作者线程在繁忙地处理任务的时候，窗口线程可以自由地响应其他用户的请求。当工作者线程完成任务的时候，它发送信号给窗口线程。窗口线程便可以更新应用程序窗口，并显示任务的结果。对用户而言，这种具有工作者线程设计的程序显得响应速度更快。

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:***

多线程的优点

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. **线程执行者（二）创建一个线程执行者**
3. **OAuth 2.0系列教程（七）请求和响应**
4. **Adopt Open JDK官方文档(五) Docker镜像**
5. **死锁**
6. **OAuth 2.0系列教程（十一）客户端证书请求和响应**
7. **线程执行者（三）创建一个大小固定的线程执行者**
8. **OAuth 2.0系列教程（十）资源拥有者密钥证书授权请求和响应**
9. **Java NIO系列教程（九） ServerSocketChannel**
10. **看动画学并发编程**
11. **并发网服务器迁移**
12. **OAuth 2.0系列教程（九）契约请求和响应**
13. **基本线程同步（七）修改Lock的公平性**
14. **Java ThreadLocal的使用**
15. **并发集合（三）使用阻塞线程安全的列表**

Write comment　　Comments RSS

Trackback are closed | Comments (11)

夕水溪下　　Log in to Reply | QUOTE

03/11. 2013 5:41pm

总的来说，多线程可以充分利用CPU资源，帮助我们编写出高性能的程序。

Ulric Qin　　Log in to Reply | QUOTE

05/18. 2013 7:59pm

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，Java ，Maven

说的简单易懂，很容易联想到web服务器了，谢谢作者

wolfer
06/09. 2013 11:49pm

Log in to Reply | QUOTE

讲得很好，浅显易懂，太爱你了

Yosemite
09/06. 2013 11:58am

Log in to Reply | QUOTE

很不错，浅显易懂

YanBit
09/12. 2013 1:58pm

Log in to Reply | QUOTE

用三个例子讲明白了，多线程的三个优点

资源利用率更好

程序设计在某些情况下更简单

程序响应更快

赞！~

zhengJackson
10/30. 2013 11:13pm

Log in to Reply | QUOTE

非常不错，谢谢楼主分享

绿荫/aiq/CHF
03/25. 2014 10:15am

Log in to Reply | QUOTE

讲的很浅显易懂，赞一个

yehong542983647
12/02. 2015 4:29pm

Log in to Reply | QUOTE

例子确实是不错，坚持，这里的知识之前看并发编程实践已经清楚了，觉得

文章比书籍好理解多了

javajack
02/16. 2016 11:07am

Log in to Reply | QUOTE

我表示很开心可以看到这样一系列的文章，毕竟，这技术含量是相对高的，

不仅在学习到知识了，还可以看到国外一些程序员分享的文章，一群可敬的

人。谢谢。

Rainyn
03/23. 2017 10:02am

Log in to Reply | QUOTE

您好，有个疑问，为什么等待数据下载的过程中CPU将会空闲大量时间？难

道下载这个操作不是CPU控制进行的吗？

_Gorgeous
04/11. 2017 9:30am

Log in to Reply | QUOTE

Hello~可以参考下这篇文章http://www.codeceo.com/article/life-

thread-learn.html#comments

You must be logged in to post a comment.

避免死锁 线程池

Copyright © 并发编程网 – ifeve.com | ICP号: 浙ICP备12046809号 | 浙公网安备 33010602005873号 |