

jimichan

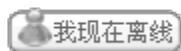
永久域名 <http://jimichan.javaeye.com>

jimichan

浏览: 42820 次

性别:

来自: 上海

[详细资料](#)[留言簿](#)

搜索本博客

最近访客  
[客](#)[>>更多访客](#)[liwanfeng](#)[zhangjunyuan](#)[Foging](#)[ceceyu](#)

博客分类

[Jetty6 指南书 - 第一章 Jetty介绍](#)[| IE 下面 80020101 错误的一个原因](#)

0 顶

0 踩

2010-02-04

## [正确使用ThreadPoolExecutor](#)

文章分类: [Java编程](#)

一直都是使用 `Executors.newFixedThreadPool(10)` 这样的方式来创建线程池。这个工厂方法实际创建 `ThreadPoolExecutor` 对象，其构造函数有三个主要参数 `corePoolSize` `maximumPoolSize` 和 `workQueue`

`newFixedThreadPool` 的时候 `ThreadPoolExecutor` 的三个构造参数分别为 10, 10, `new LinkedBlockingQueue<Runnable>()`;

一直都 **错误地理解** 他的工作方式，及新加入一个任务后，如果当前线程数达到 `maximumPoolSize` 后，就放入队列中等待执行。想当然地认为只要参数设置为 1, 10, `new LinkedBlockingQueue<Runnable>()` 后，线程池就会最大同时运行 10 个任务，没任务的时候就只有一个线程。

**错! 错! 错!**

正确的是-----当有新任务时，若当前线程数等于或大于 `corePoolSize` 时，任务会被加入到队列中，如果加入成功，则稍后有空闲线程执行；如果失败，则创建一个新的线程来执行，若线程数达到 `maximumPoolSize` 则任务会被拒绝执行。又由于 `LinkedBlockingQueue` 是个无界队列，所以这里 `maximumPoolSize` 设置是无效的。

呀! 终于用对了!

javadoc 文档的一部分:

所有 [BlockingQueue](#) 都可用于传输和保持提交的任务。可以使用此队列与池大小进行交互:

- 如果运行的线程少于 `corePoolSize`，则 `Executor` 始终首选添加新的线程，而不进行排队。
- 如果运行的线程等于或多于 `corePoolSize`，则 `Executor` 始终首选将请求加入队列，而不添加新的线程。
- 如果无法将请求加入队列，则创建新的线程，除非创建此线程超出 `maximumPoolSize`，在这种情况下，任务将被拒绝。

排队有三种通用策略:

1. 直接提交。工作队列的默认选项是 [SynchronousQueue](#)，它将任务直接提交给线程而不保持它们。在此，如果不存在可用于立

- [全部博客 \(36\)](#)
- [java \(12\)](#)
- [javascript \(4\)](#)
- [UI \(3\)](#)
- [其他 \(2\)](#)
- [google app engine \(5\)](#)
- [python \(3\)](#)
- [jetty \(5\)](#)

我的相册



m1.JPG

[共 2 张](#)

我的留言簿 [>>更多留言](#)

- 我的request里面一个文件也没有值啊[http://jimichan.javaee ...](http://jimichan.javaee...)  
-- by [xueshuanglong](#)
- 上传到哪个目录由你服务器端程序决定, 通过FileItem item对象获得被上传的 ...  
-- by [xueshuanglong](#)
- <http://www.javaeye.com/topic/111z> 我照 ...  
-- by [大力水手](#)

其他分类

- [我的收藏 \(6\)](#)

即运行任务的线程, 则试图把任务加入队列将失败, 因此会构造一个新的线程。此策略可以避免 在处理可能具有内部依赖性的请求集合时出现锁定。直接提交通常要求无界 `maximumPoolSizes` 以避免拒绝新提交的任务。当命令以超过队列所能处理的平均数连续到达时, 此策略允许无界线程具有增长的可能性。

2. 无界队列。使用无界队列 (例如, 不具有预定义容量的 [LinkedBlockingQueue](#)) 将导致在所有 `corePoolSize` 线程都忙的情况下将新任务加入队列。这样, 创建的线程就不会超过 `corePoolSize`。(因此, `maximumPoolSize` 的值也就无效了。) 当每个任务完全独立于其他任务, 即任务执行互不影响时, 适合于使用无界队列; 例如, 在 Web 页服务器中。这种排队可用于处理瞬态突发请求, 当命令以超过队列所能处理的平均数连续到达时, 此策略允许无界线程具有增长的可能性。
3. 有界队列。当使用有限的 `maximumPoolSizes` 时, 有界队列 (如 [ArrayBlockingQueue](#)) 有助于防止资源耗尽, 但是可能较难调整和控制。队列大小和最大池大小可能需要相互折衷: 使用大型队列和小型池可以最大限度地降低 CPU 使用率、操作系统资源和上下文切换开销, 但是可能导致人工降低吞吐量。如果任务频繁阻塞 (例如, 如果它们是 I/O 边界), 则系统可能为超过您许可的更多线程安排时间。使用小型队列通常要求较大的池大小, CPU 使用率较高, 但是可能遇到不可接受的调度开销, 这样也会降低吞吐量。

Google 提供的广告

[Java](#)

[博客圈](#)

[博客洪晃](#)

[博客韩寒](#)

0

顶

0

踩

[Jetty6 指南书 - 第一章 Jetty 介绍](#)

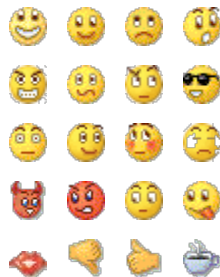
| [IE 下面 80020101 错误的一个原因](#)

10:23 | [浏览 \(82\)](#) | [评论 \(0\)](#) | 分类: [java](#) | [相关推荐](#)

评论

发表评论

表情图标



字体颜色:

字体大小:

对齐:

提示: 选择您需要装饰的文字, 按上列按钮即可添加上相应的标签

- [我的新闻](#) (1)
- [我的论坛主题贴](#) (39)
- [我的所有论坛贴](#) (28)
- [我的精华良好贴](#) (0)

最近加入圈子

- [图灵俱乐部](#)
- [MooTools](#)
- [Groovy on Grails](#)

存档

- [2010-02](#) (6)
- [2009-11](#) (2)
- [2009-04](#) (2)
- [更多存档...](#)

最新评论

- [spring事务管理在mysql数 ...](#)  
呵呵，居然也有哥们犯这个错误  
-- by [jimichan](#)
- [spring事务管理在mysql数 ...](#)  
终于在这找到原因了，谢谢。。。  
-- by [cheng888qi](#)

您还没有登录，请[登录](#)后发表评论(快捷键 Alt+S / Ctrl+Enter)

- [才发现的 纯css的圆角DI...](#)

bilang 写道 你可以看下

<http://malsup.com/jquery/> ...

-- by [oec2003](#)

- [python 中的print 是语句 ...](#)

文档里有6.6 The print

statement print\_stmt :: ...

-- by [supermarshal](#)

- [groovy MetaClass 剖析 1](#)

就我所知,系统类如内置数组类  
的处理方式是不一样的

-- by [jimichan](#)

评论排行榜

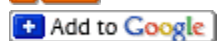
- [python 中的print 是语句还是函  
数](#)

- [正确使用ThreadPoolExecutor](#)

- [Jetty6 指南书 - 第一章 Jetty介  
绍](#)

- [Jetty6 指南和解析 - 第二章  
Jetty初探](#)

- [《Jetty6 指南书》前言和目录](#)



[\[什么是RSS?\]](#)

---

声明: JavaEye 文章版权属于作者, 受法律保护。没有作者书面许可不得转载。若作者同意转载, 必须以超链接形式标明文章原始出处和作者。

© 2003-2009 JavaEye.com. All rights reserved. 上海炯耐计算机软件有限公司 [ 沪ICP备05023328号 ]