链滴

分布式 Java 中间件

作者：someone33881

原文链接：https://ld246.com/article/1542543279077

来源网站：链滴

<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/bing/20180619.jpg?imageView2/1/w/960/h/520/interlace/1/q/100">
关键词：java 线程
    中间件是为软件应用提供操作系统所提供的服务之外的服务的，如远程过程调
和对象访问中间件、消息中间件、数据访问中间件
一、Java 线程
    线程操作的 jvm 内存是主内存和工作内存这两个概念的！！！
    1、**线程池：**有效复用线程而不用每次都创建线程，ThreadPoolExecutor、
cheduledThreadPoolExecutor（定时）、Executors.newCachedThreadPool（没有线程上限）</
>
    2、**可见性：**指在一个线程中修改变量的值之后，在其他线程中能够看到这
值
    2、**synchronized 关键字：**声明方法（静态方法则同步锁属于类、成员方
则同步锁属于对象）、代码块（该关键字后的参数，用于同步的锁所属的对象，可以是任意对象） =
gt;(线程间互斥作用、块中变量的可见性作用【在主内存与工作内存中同步变量的值，因此
可见的】)【独占锁】
    3、ReentrantLock 类：类似于修饰代码块时的 syschroni
ed，不过需要显示地进行 unlock（一般写在 finally 中）；另有 tryLock 方法
且构造函数可选择构造公平锁（严格按照顺序，所以效率相对低些）与非公平锁（可抢占，可能会导
饿死）；ReentrantReadWriteLock 读写锁，主要用于读多写少且读不需要互斥的场景
    4、volatile 关键字：只是保证所修饰的同一个变量在多线
中的可见性，常用于修饰作为开关状态的变量；同一个变量线程间的可见性与多个线程中操作互斥是
码事；【显示锁和原子变量】
    对于一般的变量，如 get 方法的调用获取的都是当前线
工作内存中的副本，因此读不一定是最新的值；
    被 volatile 修饰的变量，变量不会有线程的本地副本，只会放在主存中，因此<s
rong>读一定是最新的；
    被 sychronized 修饰的变量，则是可以保证线程的本地副本与主存的同步，因
读也一定是最新的；
    5、**Atomic*类：**提供一些原子操作，比较明显地提升性能，主要在于如 At
micInteger 内部通过 JNI 的方式使用了硬件支持的 CAS 指令；
    6、wait、notify、notifyAll 方法：均是 Object 对象的方
，wait 是进行等待的，notify 和 notify 都是唤醒调用同一个对象 wait 方法的线程，区别在于前者唤
一个等待线程而后者唤醒全部，，且这三个方法的调用都必须在 sychronized 块中</stro
g>
    7、CountDownLatch 类：当多个线程都到达了预期状态或完成预期工作时触
事件，其他线程可以等待这个事件来触发自己后续的工作；
    8、CyclicBarrier 类：循环屏障，可以协同多个线程并让多个线程在这个屏障
等待，直到所有线程都到达了这个屏障时，再一起继续执行后面的动作；
    9、Semaphore 类：用于管理信号量的，构造的时候传入可供管理的信号量的
值，总数也即控制并发的数量
    10、Exchanger 类：用于在两个线程之间进行数据交换
    11、**Future 接口和 FutureTask 类：**回调
二、并发容器：
    并发容器是线程安全的一种，但是更加强调的是容器的并发性，也即不仅仅只
求线程安全，还要考虑并发性，提升并发环境下的性能；
    加锁互斥：线程安全，但降低了并发性，其实就是串行了
    CopyOnWrite：是在更改容器的时候，把容器写一份进行
改，保证正在读的线程不受影响，适合于读多写少的场景会非常好，实质上在写的时候重
了一次容器
    Concurrent：实现思路是尽量保证读不加锁，
且修改时不影响读，所以会达到比使用读写锁更高的并发性能
三、动态代理：
    代理模式

    静态代理
    动态代理：Proxy.newProxyInstance()   invoke()
四、反射
    Java 反射机制是指在运行状态中，对于任意一个类，都能够知道这个类的所有性和方法；对于任意一个对象，都能够调用它的任意一个方法和属性
五、网络通信
    三个模型：BIO、NIO、AIO
    通信框架：MINA，Netty
【读书系列】
    《大型网站系统与 Java 中间件实践》，曾宪杰，电子工业出版社