

多核线程-volatile原理与技巧 PDF转换可能丢失图片或格式，
建议阅读原文

https://www.100test.com/kao_ti2020/350/2021_2022__E5_A4_9A_E6_A0_B8_E7_BA_BF_E7_c104_350191.htm 为什么使用volatile比同步代价更低？同步的代价，主要由其覆盖范围决定，如果可以降低同步的覆盖范围，则可以大幅提升程序性能。而volatile的覆盖范围仅仅变量级别的。因此它的同步代价很低。volatile原理是什么？volatile的语义，其实是告诉处理器，不要将我放入工作内存，请直接在主存操作我。（工作内存详见java内存模型）因此，当多核或多线程在访问该变量时，都将直接操作主存，这从本质上，做到了变量共享。volatile的有什么优势？1，更大的程序吞吐量 2，更少的代码实现多线程 3，程序的伸缩性较好 4，比较好理解，无需太高的学习成本 volatile有什么劣势？1，容易出问题 2，比较难设计 volatile运算存在脏数据问题 volatile仅仅能保证变量可见性，无法保证原子性。volatile的race condition示例：

```
public class TestRaceCondition { private volatile int i = 0. public void increase() { i. } public int getValue() { return i. } }
```

当多线程执行increase方法时，是否能保证它的值会是线性递增的呢？答案是否定的。原因：这里的increase方法，执行的操作是 $i = i + 1$ ；针对 $i = i + 1$ ，在多线程中的运算，本身需要改变i的值。如果，在i已从内存中取到最新值，但未与1进行运算，此时其他线程已数次将运算结果赋值给i. 则当前线程结束时，之前的数次运算结果都将被覆盖。即，执行100次increase，可能结果是一一般来说，这种情况需要较高的压力与并发情况下，才会出现。100Test 下载频道开通，各类考试题目直接

下载。详细请访问 www.100test.com