

Java 应用在 Linux 操作系统下，性能监控

作者: [wangschi](#)

原文链接: <https://ld246.com/article/1474811649500>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

Java应用可能因多线程锁竞争，导致性能问题，或无法扩展。用pidstat -w监控上下文切换，结合一般准则判定Java应用是否存在锁竞争。挂起（如：在遇到锁时阻塞）和恢复，会导致操作系统的让步式上下文切换（Voluntary Context Switch）。所以，锁竞争严重的应用，会出现大量的让步式上下文切换。

本例中，没有任何上下文切换。假设cswch/s为3500即，让步式上下文切换每秒3500次。CPU为核心，CPU频率是3GHz。计算方法：用 $3500/2=1750$ ，耗费的时钟周期 $1750 * 80\ 000$ （每次让步上下文切换的时钟周期）= $140\ 000\ 000$ 。 $140\ 000\ 000/3\ 000\ 000\ 000=4.7\%$ 。根据一般准则，让步式上下文切换时钟周期占用 $3\% \sim 5\%$ ，说明Java应用面临锁竞争。

抢占式上下文切换

让步式上下文切换，通常是线程主动释放CPU（如：在遇到锁时阻塞），抢占式上下文切换代表线程因为被分配的时间片用尽被迫放弃CPU或被其他优先级更高的线程所抢占。抢占式上下文切换率高，说明预备运行的线程数多于可用的虚处理器数，用vmstat可以看到很长的运行队列、很高的CPU使用率、很大的线程迁移数。

减少应用线程数，用taskset将应用分配给特定的处理器组，可以减少抢占式上下文切换。

