



链滴

Java 高并发笔记

作者: [CM](#)

原文链接: <https://ld246.com/article/1486373784238>

来源网站: [链滴](#)

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

并发的概念

- 阻塞
- 非阻塞：
 - 1.无障碍
 - 2.无锁
 - 3.无等待

阻塞

只允许一个线程访问临界区（共享资源）。

非阻塞：

- 允许多个线程进入临界区。非阻塞下的无障碍、无锁以及无等待方式对于释放资源的约束条件不同。
- 【无障碍】又被成为乐观锁。无竞争时，线程能在有限的步骤内完成操作。而有竞争的时候则回滚数据，尝试重新发起操作。
- 【无锁】是在无障碍的基础上，保证每一次竞争有一个线程能胜利，并完成操作，可避免活锁的情况（多个线程因只能获取部分资源而无法进行操作，而陷入循环进出临界区的情况）。
- 【无等待】是无锁的基础上，要求每个线程都必须在有限的步骤内完成操作。所以也必定是无饥饿。

并行的两个定律

Amdahl(阿姆达尔定律)

【加速比】 $= 1 / F - [(1 - F) * 1 / n]$
其中F为程序中串行模块的比重
n为处理器的个数

要点

只有提高并行模块的比重再合理地添加处理器才能以最小的投入，提高加速比。

Gustafson（古斯塔夫森定律）

执行时间：串行时间+并行时间 $(a+b)$
总执行时间：串行时间+处理器个数*并行时间 $(a+n*b)$
加速比：总执行时间/执行时间 $(a+b)/(a+n*b)$

定义：F=串行时间/执行时间（串行比例 $F = a/a+b$ ）

则加速比 $S(n) = (a + n \cdot b) / (a + b) = a / (a + b) + n \cdot b / (a + b)$
 $= F + n[(a + b - a) / (a + b)]$
 $= F + n[1 - (a / (a + b))]$
 $= F + n(1 - F)$
 $= F + n - nF$
 $= n - F(n - 1)$

要点

串行比例很小的时候，加速比和CPU的个数是成正比的。