在高性能计算中，重要的不是你写代码有多快，而是你写的代码运行有多快。

这一切都与内存有关：您使用多少内存以及多久加载一次内存。

如果你加载一个值，你会得到八个或十六个。

就像买鸡蛋一样。你不能只得到一个。内存加载由512位的高速缓存行完成。对于8字节的双精度值，无论您是否需要，都将加载八个值。计划您的程序使用多个值，最好是八个连续的值，以获得最佳性能。当你吃的时候，用剩下的鸡蛋。

如果代码中有任何缺陷，并行化将暴露它们。

■了解并行计算机中的资源

■评估应用程序的性能和加速比

■查看软件工程对并行计算的特殊需求

■考虑数据结构的选择

■选择执行和并行化良好的算法

为什么是并行计算？

本章涵盖

什么是并行计算以及

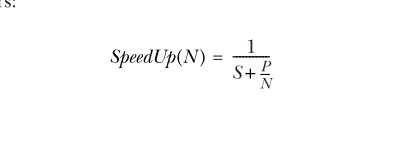
为什么它越来越重要

现代硬件中存在并行性的地方

为什么应用程序并行性的数量很重要

并行强调的是在**同一时刻**，多个任务同时执行

并发（Concurrency）才更符合你所说的 “一段时间上，多个任务的同时执行” 这种描述。



动态随机存取存储器（Dynamic Random Access Memory，简称 DRAM）用于存储信息或数据。计算核心（简称核心）执行算术运算（加、减、乘、除）、评估逻辑语句，并从 DRAM 中加载和存储数据。

两种并行架构，两块cpu共享一个区域，或者分开。

图形用户界面

AI 生成的内容可能不正确。

消息传递方法是为分布式内存架构开发的，它使用显式消息在进程之间移动数据。在这个模型中，你的应用程序产生了独立的进程，在消息传递中称为等级，拥有自己的内存空间和指令管道（图1.20）。

SIMD（单指令，多数据）和MIMD（多指令，多数据）

文本, 信件

AI 生成的内容可能不正确。

文本

AI 生成的内容可能不正确。

文本

AI 生成的内容可能不正确。

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

文本, 信件

AI 生成的内容可能不正确。

文本

AI 生成的内容可能不正确。

它具有以下特点：

* 对数组进行操作，而非单个数据项，这样可以避免函数调用开销以及指令和数据缓存未命中的问题。
* 在更多情况下，相较于结构体，更倾向于使用数组，以更好地利用缓存。
* 采用内联子例程，而非遍历深层次的调用层次结构。
* 控制内存分配，避免在后台进行无向的内存重新分配。
* 使用基于连续数组的链表，避免使用 C 和 C++ 中标准的链表实现方式，因为标准链表在内存中跳转频繁，数据局部性和缓存利用率较差。

缓存未命中的三个C：强制、容量、冲突

文本, 信件

AI 生成的内容可能不正确。

手机屏幕截图

AI 生成的内容可能不正确。

文本

AI 生成的内容可能不正确。