

# 第十章 高性能计算 和并行算法

## § 10. 1 引言

计算机的运算速度在日新月异地增长，计算机的市场价格却不断地下降。

当前的计算机技术仍然远远不能满足物理问题计算的需要。

高性能计算机是一个所有最先进的硬件，软件，网络和算法的综合概念，“高性能”的标准是随着技术的发展而发展的。

高性能计算系统中最为关键的要素是单处理器的最大计算速度，存储器访问速度和内部处理器通讯速度，多处理器系统稳定性，计算能力与价格比，以及整机性能等。

传统的计算机是冯·纽曼(Von Neumann)计算机，它是由中央处理器、内存器和输入/输出设备构成。

冯·纽曼“瓶颈”。

为了要超越这个“瓶颈”，人们发展了两种计算机体系结构和相关软件技术的应用原则。一个是并行算法(parallelism)，另一个是流水线技术(pipelining)。

由于高性能计算机与当前能够应用的新计算技术相关联，因而它与并行算法和流水线技术有着密切的联系。

## § 10. 2 并行计算机和并行算法

并行计算机是由多个处理器组成，并能够高速、高效率地进行复杂问题计算的计算机系统。

串行计算机是指只有单个处理器，顺序执行计算程序的计算机，也称为顺序计算机。

并行计算作为计算机技术，该技术的应用已经带来单机计算能力的巨大改进。

并行计算就是在同一时间内执行多条指令，或处理多个数据的计算。并行计算机是并行计算的载体。

**为什么要采用并行计算呢？**

- **并行计算可以大大加快运算速度，即在更短的时间内完成相同的计算量，或解决原来根本不能计算的非常复杂的问题。**
- **提高传统的计算机的计算速度一方面受到物理上光速极限和量子效应的限制，另一方面计算机器件产品和材料的生产受到加工工艺的限制，其尺寸不可能做得无限小。因此我们只能转向并行算法。**
- **并行计算对设备的投入较低，既可以节省开支又能完成计算任务。实际上，许多物理计算问题本身就具有并行的特性，这就是需要并行算法的最朴素的原因。**

**通常的冯·纽曼计算机是属于 SISD (Single Instruction Single Data stream computers) 单指令单数据流计算机类型计算机，它的结构只有一个处理器，同时可以处理一个单数据流。**

并行计算机若采用与某种网络相关的多处理器结构，则会使其性能得到改善。如今广泛使用的结构分成以下类型：

- SIMD (Single Instruction Multiple Data stream computers) 单指令多数据流计算机；
- MIMD (Multiple Instruction Multiple Data stream computers) 多指令多单数据流计算机。

单/多指令指的是可以同时计算机上执行的指令数，而单/多数据流指的是计算机同时可处理的一个或多个数据流。SIMD 和 MIMD 体系结构对并行计算机十分重要。

- SIMD 计算机具有处理器器件阵列，它通过单个控制单元来控制功能设备。这个控制单元向所有处理器发出相同的执行指令，随后所有处理器对进入的不同数据流进行相同的操作。
- MIMD 计算机具有可以同时独立运行多条指令，对不同的数据进行操作。例如对数组的如下运算  $A=B*C+D+E+F/G$ ，可以分解为乘法 ( $B*C$ )，加法 ( $D+E$ ) 和除法 ( $F/G$ ) 直接交给 MIMD 计算机的直接处理部分同时进行计算。

按照同时执行程序的不同并行计算机又可以分为：

SPMD (Single Program Multiple Data Stream Computers) 单程序多数据流并行计算机，SPMD 并行计算机一般是由多个相同的计算机或处理器构成；

MPMD (Multiple Program Multiple Data Stream Computers) 多程序多数据流并行计算机。MPMD 并行计算机内计算机或处理器

的地位是不同的，根据分工的不同，它们擅长完成的工作也不同，因此可以根据需要将不同的程序放到 MPMD 并行计算机上执行，使得这些程序协调一致地完成给定任务

这种划分依据的执行单位不是指令而是程序。针对 SPMD 和 MPMD 并行计算机所设计的并行算法的思路与前面介绍的并行算法的思路有很大不同。

**并行计算机也可以按照内存的结构来划分：**

**“分布式内存”和“共享式内存”两种基本结构的并行计算机。**

**根据并行计算任务的大小分为三类：**

- **粗粒度并行算法；** 它所含的计算任务有较大的计算量和较复杂的计算程序。
- **细粒度并行算法；** 它所含的计算任务有较小的计算量和较短的计算程序。
- **中粒度并行算法；** 它所含的计算任务的大小和计算程序的长短在粗粒度和细粒度两种类型的算法之间。

**根据并行计算的基本对象可以分为：**

**数值并行计算和非数值并行计算。**

**根据并行计算进程间的依赖关系可以分为：**

- (1) 同步并行算法；该算法是通过一个全局的时钟来控制各部分的步伐，将任务中的各个部分计算同步地向前推进。**
- (2) 异步并行算法；它执行的各部分计算步伐之间没有关联，互不同步。**

**对于 SIMD 并行计算机一般适合用同步并行算法，而 MIMD 并行计算机则适合用异步并行算法。**

## § 10. 3 并行编程

通常编程设计过程可以分为四步：

- **任务划分 (Partitioning)**。该阶段将整个使用域或功能分解成一些小的计算任务，它的目的是要揭示和开拓并行执行的机会；
- **通信 (Communication) 分析**。由任务划分产生的各项任务一般都不能独立完成，可能需要确定各项任务中所需交换的数据和协调各项任务的执行。通信分析可以检测在任务划分阶段划分的合理性；
- **任务组合 (Agglomeration)**。按照性能要求和实现的代价来考察前两个阶段的结果，必要时可以将一些小的任务组合成更大的任务以提高执行效率和减少通信开销；
- **处理器映射 (Mapping)**。这是设计的最后阶段。要决定将每一个任务分配到哪个处理器上去执行。目的是要最小化全局执行时间和通讯成本，并最大化处理器的利用率。

上述过程简称为 PCAM 设计过程。

**目前最重要的并行编程模型有两种：**

- **数据并行 (Data Parallel) 模型** (它的初衷是为了 SIMD 并行机),
- **消息传递 (Message Passing) 模型** (其初衷是为了多计算机)。

还有共享变量模型、函数式模型等，但是它们的应用都没有前面两种那样普遍。

数据并行的含义是将相同的操作同时作用于不同数据，因此不但适用于在 SIMD 计算机上实现，也可以在 SPMD 并行计算机上运行，这取决于粒度大小。SIMD 程序着重开发指令级中细粒度的并行性，SPMD 程序着重开发子程序级中粒度的并行性。

并行语言的产生有三种方式：

- **设计全新的并行语言；**
- **扩展原来串行语言的语法成分，使它支持并行特征；**
- **为串行语言提供可调节的并行库，并不改变串行语言。**

目前常用的是第 (2) 和 (3) 两种方式，最常用的是第 (3) 种。对 FORTRAN 和 C 语言的扩充是最常见的并行语言产生办法。

**The End**