

多核编程策略：数据并行化

硬件设计师通过使用多核处理器来改善计算能力，软件程序员也必须找到新的编程策略来驾驭并行计算的能力。一项技术的出现有效地利用了多核处理器并行化处理数据的优势，我们称之为“分治策略(Divide and Conquer)”技术。

数据并行化

数据并行化是一种将大块数据集分割成小块并行处理的编程技术。完成数据处理后，再将其组合成单个数据集。这项技术使程序员最大限度利用多核处理器，有效使用所有的处理能力。

如图 1 中的情况，大数据集经由单独处理器处理。此时另外三个 CPU 核处于空闲状态，由第一个处理器独立承担所有数据集的处理任务。

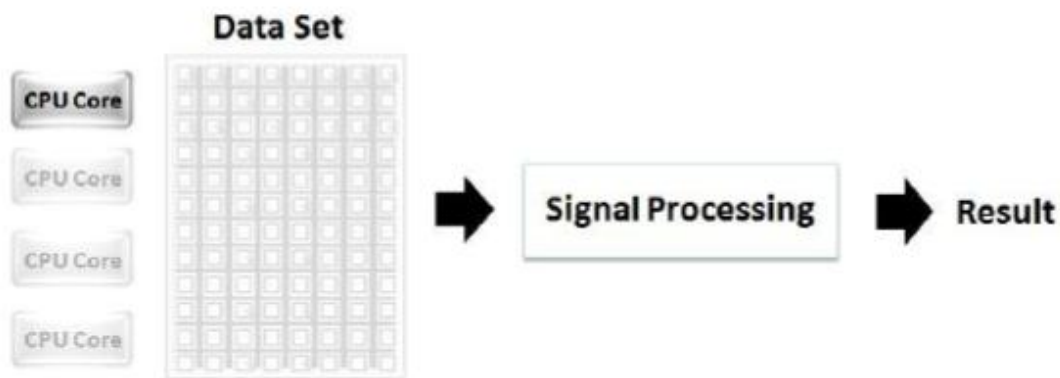


图 1 单 CPU 处理大数据集

现在看一下图 2 中的实现情况，通过数据并行化来最大限度利用 4 核处理器的处理能力。此时大数据集被分割为四个子集，每个子集被分配给不同的处理器来处理。完成所有处理后，再将这些子集重组成完整的数据集。

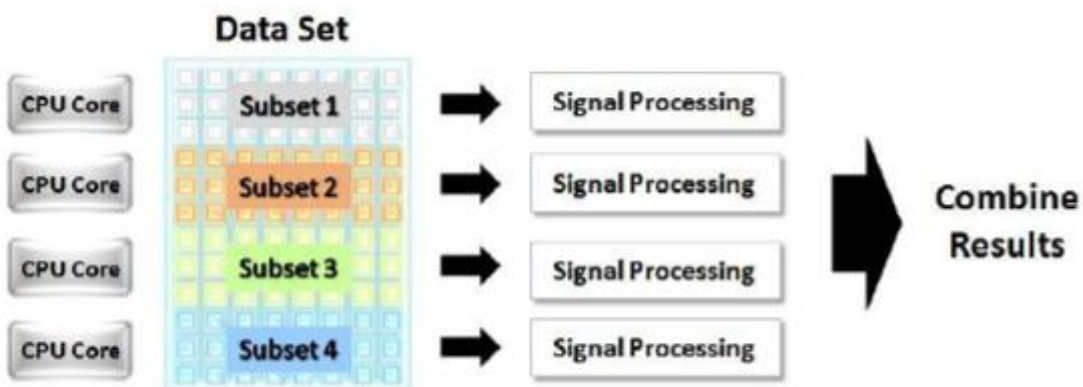


图 2 4 核并行处理大数据集

图中使用的 NI LabVIEW 图形化编程方式是并行数据构架的典范。通过 NI LabVIEW 实现并行处理相比于传统基于文本的编程语言来说更直观、简单，而采用后者创建多线程应用会要求更高的编程知识。

LabVIEW 中的数据并行化

图 3 中的代码为两个矩阵（矩阵 1 和矩阵 2）的乘法运算。这是 LabVIEW 中两个大矩阵乘法的标准实现方法。

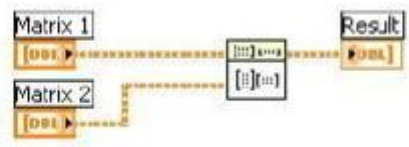


图 3 不采用数据并行化在 LabVIEW 中实现矩阵乘法

完成上述的操作需要大量的计算时间，尤其当数据量特别大时。图 3 中的代码并未体现出多核处理器的优势，除非矩阵乘法 VI 恰好是多线程，且针对多核处理器优化过。相对的，图 4 中的代码利用了数据并行化，从而在双核处理器上具有更快的执行速度。

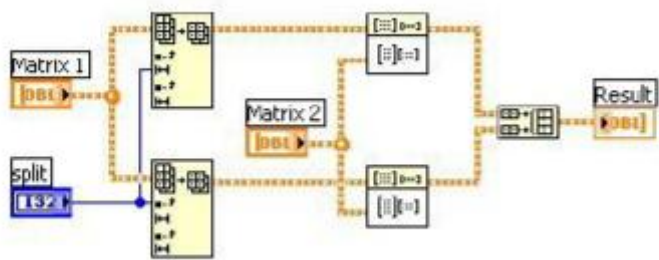


图 4 采用数据并行化在 LabVIEW 中实现矩阵乘法运算

在 LabVIEW 中使用该项技术提高多核处理器上的性能的同时，很重要一个前提是:矩阵乘法 VI 必须是可重入的。不可重入的矩阵乘法 VI 将不能独立、并行地执行。

数据并行化的应用实例

德国慕尼黑马克斯·普朗克研究院的研究人员采用数据并行化的 LabVIEW 编程，用于德国最先进的核聚变装置 ASDEX tokamak 中的等离子控制。该程序运行于一台 8 核的服务器上，在 8 个 CPU 核上并行的执行高强度矩阵运算，实现了 1ms 控制循环。首席研究员 Louis Giannone 表示：“在使用 LabVIEW 完成首阶段控制应用编程设计后，我们在 8 核处理器上实现了单核处理器 20X 的速度，成功达到 1ms 闭环控制速率的要求”