

多核编程策略：任务并行化

任务并行化

任务并行化是在软件中实现独立任务的并行处理。考虑单核处理器下同时运行网络浏览器和文字处理程序。虽然这些应用程序通过不同线程运行，但它们最终仍会共享同一处理器。现在来考虑另一种情形，两个程序运行在双核机器上，两个应用程序在本质上是独立的。尽管它们并非彻底独立，因为还需要共享一些通用资源，但双核能更有效地处理两个并行任务。

LabVIEW 中的任务并行化

LabVIEW 图形化编程方式允许开发人员通过绘制并行代码部分或分支代码来实现任务的并行化。两个没有数据交换的独立任务无需额外编程即可独立运行。

图 1 显示的简单数据采集过程中，上部分代码为模拟电压输入任务，下部分代码为数字输出任务。

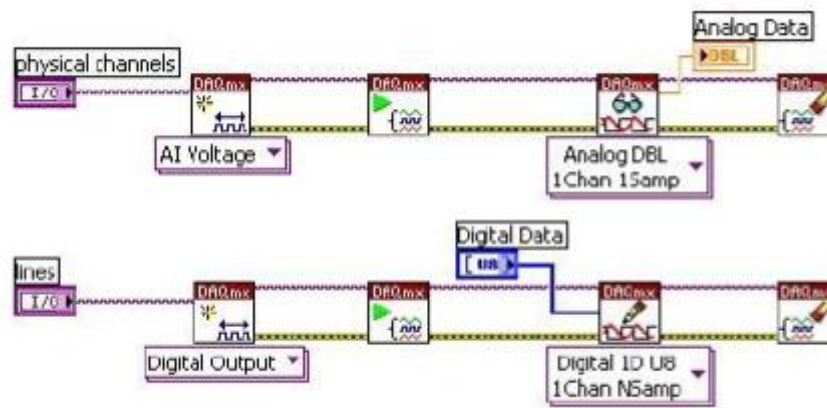


图 1 LabVIEW 中的独立并行模拟输入和数字输出任务

两个独立的数据采集任务必须共享单核的 CPU。而多核处理器中，每个线程可在不同 CPU 内核中运行，极大提升了执行速度。

在应用中使用数据并行时，很重要一点是确保两个任务不使用可能造成瓶颈的共享资源，就如同图 2 中的代码。

任务并行化的应用实例

Eaton 公司的一项 LabVIEW 应用中，基于状态机架构的传输测试包含三个主要任务，它能够以数据采集通道的四倍速运行。该应用由三个异步循环来完成并行的采集、测试控制、和用户界面任务。在方案实现之前，团队设计人员只能通过依次分离工作量或同时使用多台单核笔记本来满足测试要求。使用标准的商业桌面系统后，设计人员能够最小化功率消耗、散热、及测试时间，从而降低总成本。他们还将测试系统装入“可移动的测试车”，使用多核笔记本实现车载代码运行。