多核编程策略：流水线

流水线介绍

流水线是一项应用广泛的技术，用于顺序软件任务的性能提升。简单说来，流水线是将顺序任务分步处理的方式，就如同产品装配线的模式。

考虑以下例子，假设在汽车装配线上生产汽车，您的任务是装配一辆完整的汽车。这时您可以将此过程具体分成三个步骤：搭建车身结构，安装汽车配件（如引擎等），对成品汽车喷漆。

假设车身、配件、喷漆每道工序需要一小时，因此装配一辆完整的汽车需要 3 小时（见图 1）。

图 1. 在本例中，从装配线上下线一辆汽车需要花费 3 小时

这个过程如何改进呢？我们是否可以设立一个工作站用于生产车身结构，第二个用于安装配件，第三个用于喷漆？如此一来，当第一辆车在喷漆时，第二辆车可进行配件安装， 第三辆车正在搭建车身结构。

流水线如何提升性能

尽管在新的工序下每辆车仍需要三个小时完成，但现在每小时就有一辆车下线，相比每三小时下线一辆车，汽车生产的吞吐量提升了 3 倍。

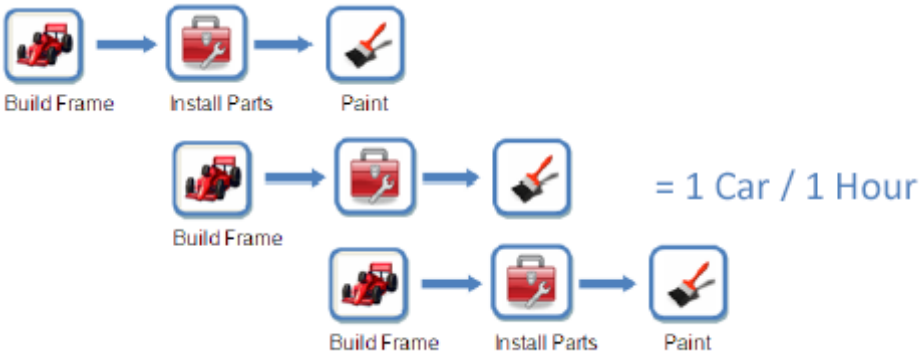


图 2. 流水线能显著提升应用的吞吐量

LabVIEW 中的流水线

同样的流水线概念可以形象地用于 LabVIEW 应用中的顺序任务执行。从本质上，我们可以通过 LabVIEW 的移位寄存器和反馈节点在任何程序中创建类似“装配线”流水线结构。

譬如，考虑从数据采集设备中采集数据的应用，实现快速傅立叶变换(FFT)分析，然后将分析结果保存到磁盘。值得注意的是，这是一个典型的顺序任务，不可能在对数据进行 FFT 分析的同时又在保存分析结果，但您可以利用多核系统构架的优势，通过流水线方式来实现。

现在必须返回到之前提到的“级”概念，我们的应用案例可以分三个具体步骤：采集、分析、存储。假设每个过程需要 1 秒钟来完成，也就是说，不采用流水线的情况下，每个循环需要 3 秒的时间来执行。我们可以认

为应用的吞吐量是每 3 秒完成 1 次计算。

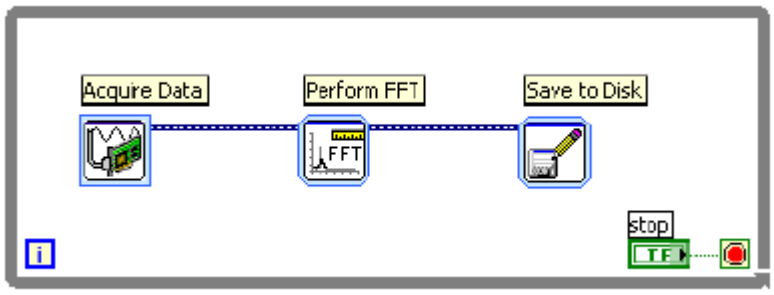


图 3 不采用流水线时 LabVIEW 应用程序执行需要 3 秒

图 3. 在每个 LabVIEW 步骤中放置反馈节点，便可以轻松实现流水线程序。

插入反馈节点后，每个流水线级在获得前一过程数据（前一次循环）后开始操作。如此一来，流水线的每次循环都能完成一次计算，将吞吐率提升了 3 倍。

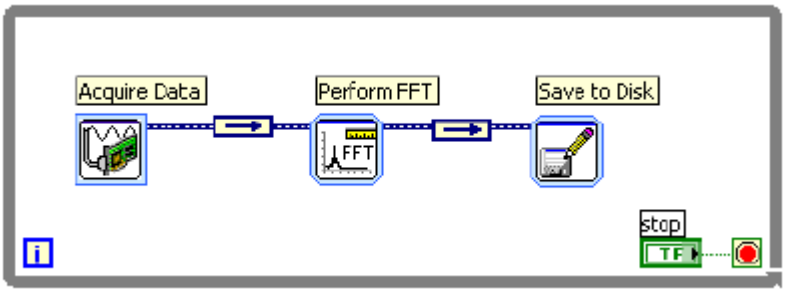


图 4 流水线应用中采用了反馈节点，使吞吐率提高 3 倍

另一种在 LabVIEW 中实现流水线的方法是采用移位寄存器来代替反馈节点（两种方法的功能是等价的）。

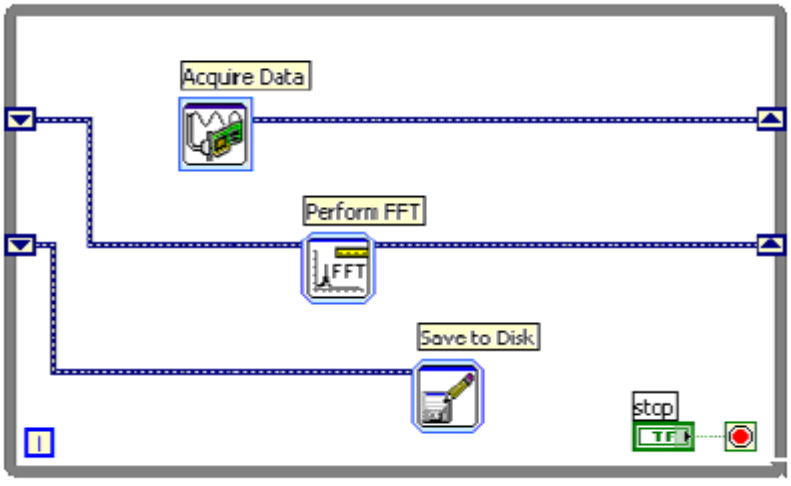


图 5. 使用移位寄存器的流水线应用将吞吐率提升 3 倍

重要考虑因素

流水线间平衡

在上述的汽车制造和 LabVIEW 案例中，每个流水级都假设为同样的执行时间，我们认为这样的各级流水线是平衡的。

然而在真实应用中很难做到各级平衡。考虑下述的框图：如果第一级的执行时间是第二级的 3 倍，那么两级的流水线只能获得很小的性能提升。

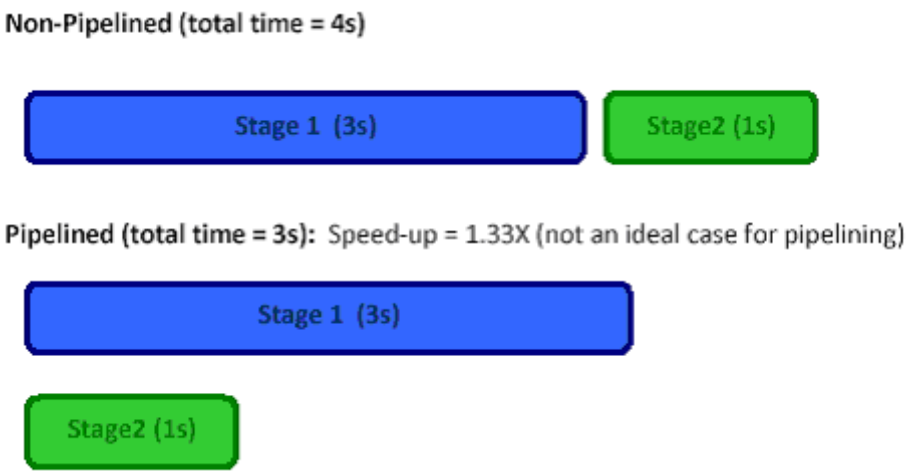
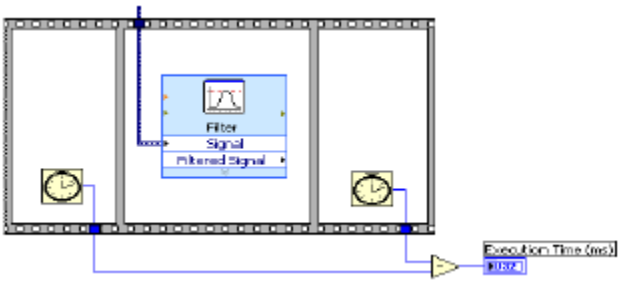


图 6 确定执行时间来识别平衡的应用

这种情况下需要做一定修改，必须将任务从第一级中移到第二级，直至两级的执行时间大致相当。当流水线级数很高时，这一修改就变得非常困难了。在 LabVIEW 中，确认每级流水线基准时间很重要，这样可以确保各级流水线的平衡。采用关联中的平铺式顺序结构以及 Tick Count(ms)函数可以很方便地确定基准，见下图。



处理器核间的数据传输

我们应当尽可能避免在流水线两级之间传输大量数据。因为流水线中的不同级可能运行于不同处理器核上，各级流水线间的数据传输实际上就相当于在处理器核的物理内存间传输数据。由于处理器核并不共享缓存（或者传输数据大小超过了缓存大小），这就导致，从整体应用的用户角度看来流水线的效率降低了。

结论

总的来说，流水线技术能够提高原本顺序执行应用的效率，使程序员从多核处理器中获益。为了从流水线中获得最大的性能提升，必须使各级流水线平衡，不允许某级执行时间比其它级长许多。此外，需要最小化各级流水线间的数据传输，避免因处理器核内存见的互相读写而导致性能反而降低。