

利用 NI LabVIEW 与多核处理器优化自动化测试应用

LabVIEW 为自动化测试应用提供了独特的、简单易用的图形化编程环境。它能够动态地将代码分配到多个 CPU 核上运行，从而提高在多核处理器上的执行速度。下面让我们来学习如何利用并行编程技术，对 LabVIEW 应用程序进行优化。

多线程编程的挑战

直到最近，在处理器技术上的革新使得计算机的 CPU 能够以更高的时钟频率工作。然而，随着时钟频率逐渐接近理论物理极限，各制造公司都在开发多核的新型处理器。有了这些新型的多核处理器，开发自动化测试应用的工程师们可以借助并行编程技术，获得最高的性能和最大的吞吐量。Edward Lee 博士是美国加州大学伯克利分校的电气与计算机工程教授，他这样描述并行处理的优点。

“许多技术专家预测摩尔定律的最终答案是逐渐提高并行度的计算机体系结构。如果我们希望继续得到计算性能的提升，那么程序就必须能够利用并行机制。”

此外，行业专家认为编写能够利用多核处理器的应用程序是一个严峻的挑战。微软公司的创立者 Bill Gates 如下解释：

“要充分利用并行处理器的能力，软件必须解决并发性问题。但是正如所有编写过多线程代码的开发员所说的一样，这是编程中最为困难的任务之一。”

值得庆幸的是，NI LabVIEW 软件提供了一个理想的多核处理器编程环境，它包含了直观的 API，这些 API 可以创建并行算法，这些算法可以在一个应用中动态调度多个线程。这样，通过使用 LabVIEW，您就可以使用多核处理器优化自动化测试应用程序，达到最佳性能。

此外，PXI Express 模块化仪器利用 PCI Express 总线所提供的高数据传输速率，其结果使得这个优点更突出。能够从多核处理器和 PXI Express 仪器中获益的两个应用是多通道信号分析与在线处理（硬件在环）。本白皮书评价了多个并行编程技术，并且对每个技术所带来的性能提升进行了描述。

实现并行测试算法

多通道信号分析是能够从并行处理中获益的一个常见的自动化测试应用。由于频率分析是一个处理器密集型的任务，通过并行运行测试代码，可将各通道的信号处理分布到多个处理器核上，这就可以提高执行速度。从程序员的角度出发，只需要对测试算法进行简单的重构就可以充分享有并行处理的好处。

例如，可以通过比较一个高速数字化仪的两个通道上的两个多通道频率分析（快速傅立叶变换，即 FFT）算法的执行时间来说明这一优点。NI PXIe-5122 14 位高速数字化仪使用两个通道获得最高采样速率（100 MS/s）的信号。首先，在 LabVIEW 中用传统的顺序执行程序模型来执行这一操作。

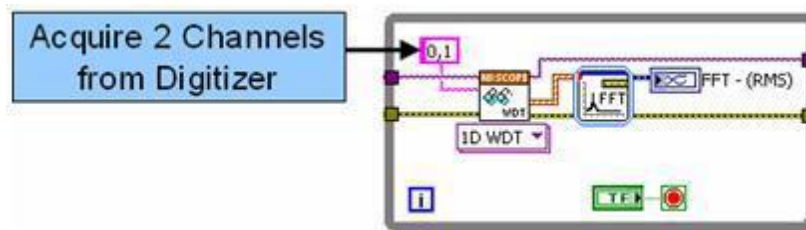


图 1 LabVIEW 使用顺序执行代码从数字化仪对两个通道进行采集

在图 1 中，两个通道的频率分析都在 FFT Express VI 中完成，其针对每个通道的分析是顺序执行的。由于上

述的算法还可以在多核处理器中有效地执行，那么就可以通过并行得处理各个通道来提高算法性能。

如果对算法性能进行分析，就会发现 FFT 计算相对从高速数字化仪采集数据而言，需要花费更多的时间完成。通过同时对各个通道进行采集，并将两个 FFT 并行计算，就可以大大减少处理时间。图 2 是使用并行方案的 LabVIEW 程序框图。

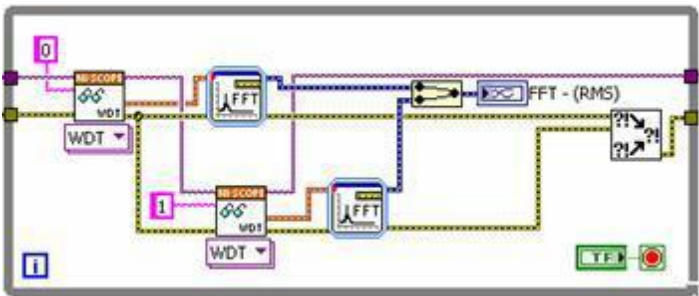


图 2 使用并行执行结构的 LabVIEW 代码

每个通道都是通过数字化仪顺序采集的。请注意如果数据采集都是来自独立的仪器，就可以完全并行地执行这些操作。然而，由于 FFT 是处理器密集的任务，您也可以通过简单地将信号处理并行运行来提高程序性能。这样，总执行时间也会降低。图 3 显示了两个所需的执行时间的比较。

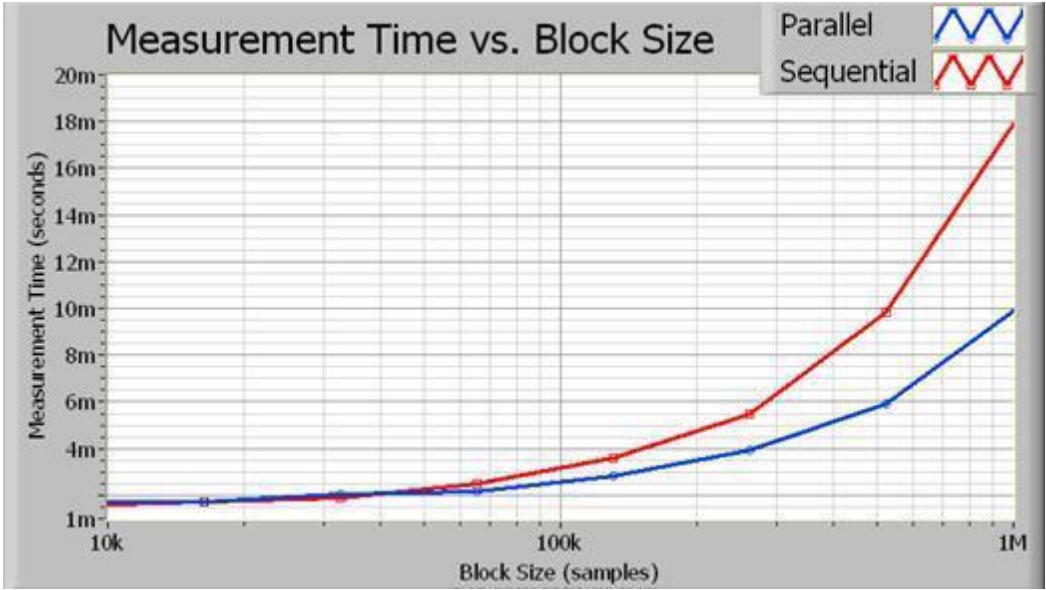


图 3 随着块大小的增加，通过并行执行所节省的处理时间变得越来越显著

实际上，对于较大的数据块而言，使用并行算法能够使性能提高达到两倍。图 4 显示了性能提升的比例关于采集大小（采样点数）的函数关系。

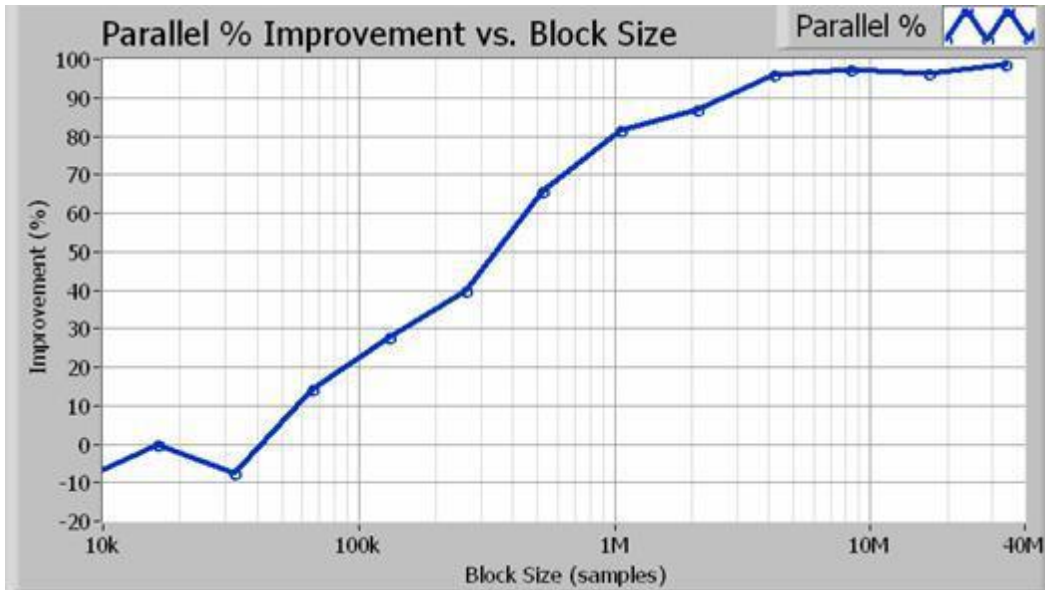


图 4：对于超过 100 万采样点的块大小而言（100 Hz 分辨率带宽），使用并行方法能够得到 80% 甚至更多的性能提升。

因为使用 LabVIEW 可动态地分配每个线程，所以在多核处理器上很容易提高自动化测试应用程序的性能。实际上，您并不需要编写使用多线程的专门代码。而只需要极少地进行程序修改就可以让并行测试应用得益于多核处理器。

配置自定义并行测试算法

并行信号处理算法帮助 LabVIEW 对多核划分处理的使用。图 5 显示了 CPU 处理算法各个部分的次序。

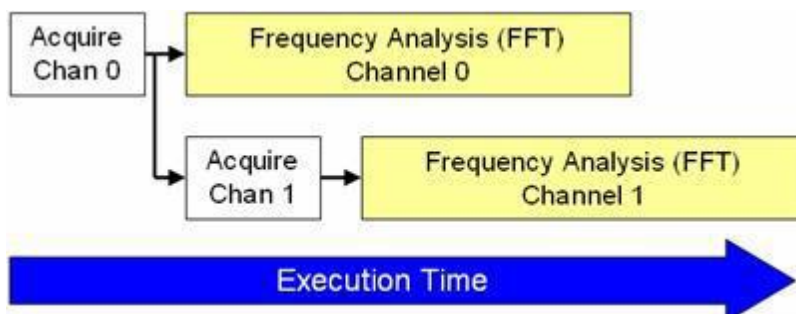


图 5：LabVIEW 能够并行处理大部分采集到的数据，从而节省了执行时间

并行处理要求 LabVIEW 为每个信号处理子程序制作一个复本（克隆）。在默认状态下，许多 LabVIEW 信号处理算法都被配置为“可重入执行”。这意味着 LabVIEW 将动态地分配每个子程序的独立例程，包括独立的线程和内存空间。因此，您必须将子程序配置为以可重入方式运行。要做到这点只需要在 LabVIEW 进行一步简单的配置。要设置这个属性，选择文件》VI 属性，选择执行类别。然后选择如图 6 所示的可重入执行标志。

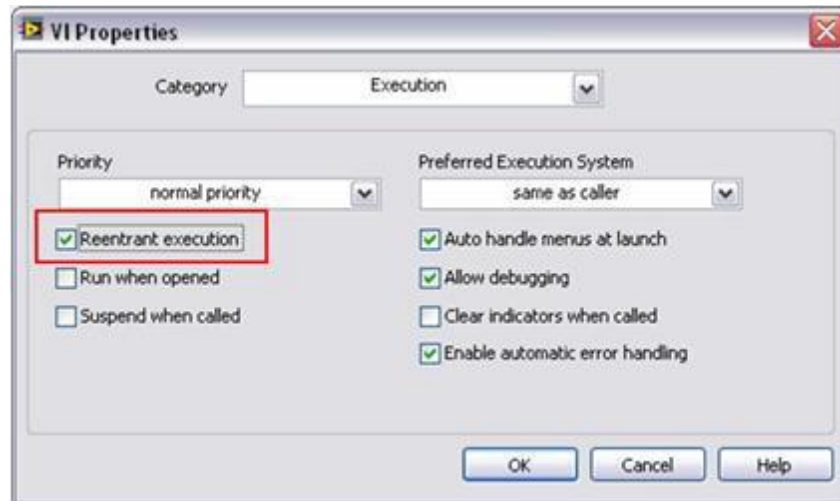


图 6 在这个简单的步骤中，您可以和标准 LabVIEW 分析函数一样，并行执行多个自定义子程序

这样，只需要使用简单的编程技巧，就可以利用多核处理器提高自动化测试应用程序的性能。

优化硬件在环应用

另一个能够从并行信号处理技术中获益的是使用多个仪器进行同时输入输出。通常，这被称作硬件在环（HIL）或在线处理应用。在这个情形下，您需要使用高速数字化仪或高速数字 I/O 模块采集信号。然后在软件中完成数字信号处理算法。最后，使用另一个模块化仪器生成结果。图 7 是典型的程序框图。

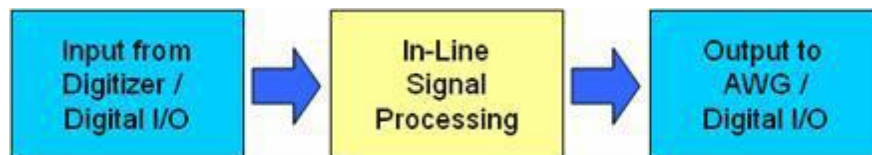


图 7：这个程序框图展示了典型硬件在环（HIL）应用中的步骤

通常的硬件在环应用包含在线数字信号处理（例如滤波和插值）、传感器仿真以及自定义元件模拟。此时，可以使用多种在线数字信号处理应用中技巧，以获得最大吞吐量。

一般来说，可以将基本编程结构分为两种——简单循环结构和包含队列的流水线多循环结构。对于较小的数据块来说，简单循环结构易于实现，并可以提供较低的延迟。另一方面，由于多循环体系结构可更好得多核处理器，结果能够提供更高的吞吐量。

使用传统简单循环方法，您可以顺序布置高速数字化仪读函数、信号处理算法和高速数字 I/O 写函数。如图 8 的程序框图所示，LabVIEW 编程模型确定了每个子程序必须按照顺序执行。

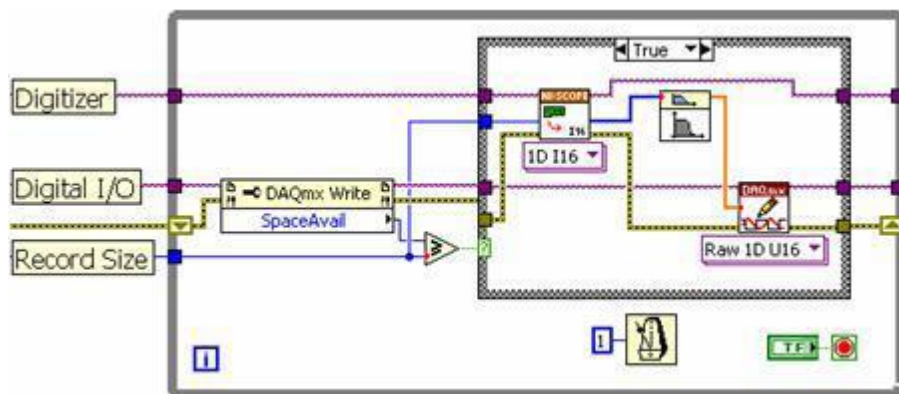


图 8 使用 LabVIEW 简单循环方法，每个子程序必须按照顺序执行

简单循环结构受到多种限制。由于每个阶段都是按照顺序执行的，处理器在处理数据的时候，就不能进行仪器 I/O。使用这个方法，由于处理器在同一时间只能执行一个函数，因此不能有效地利用多核 CPU。尽管对于低采样速率而言，使用简单循环已经足够，要达到更高的数据吞吐量，就需要使用多循环方法。

多循环体系结构使用队列在各 while 循环之间传递数据。图 9 展示了使用队列结构在 while 循环之间的编程方法。

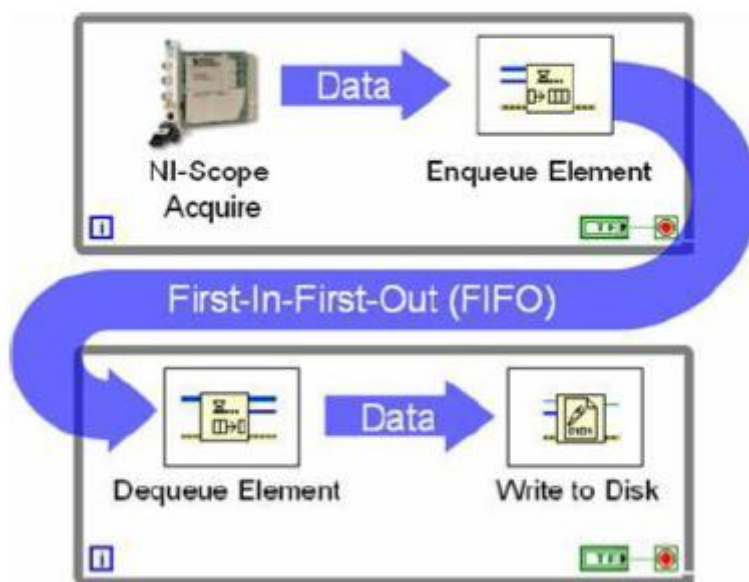


图 9 多循环间通过队列结构共享数据

图 9 所示是一个典型的生产者/消费者循环结构。结构中由一台高速数字化仪循环采集数据，并在每次循环中将新数据送入 FIFO 通道。消费者循环仅用于监测队列状态，在数据就绪时将其写入磁盘。使用队列的意义在于使两个循环彼此独立执行。

在上述例子中，尽管数据写入磁盘有一定的滞后，但高速数字化仪仍保持连续采集数据。期间未写入的数据将保存在 FIFO 通道中。一般来说，生产者/消费者流水线方式能够提供更大的数据吞吐量，优化处理器的使用率。这种优势对于多核处理器来说将更加显著，因为 LabVIEW 能在多核之间动态分配处理器线程。

对于内嵌式的信号处理应用，可以使用三个独立的 while 循环和两个队列结构来传递数据。此时，一个循环用

于从仪器采集数据，第二个循环专用于信号处理，第三个循环将数据写入另一台仪器。

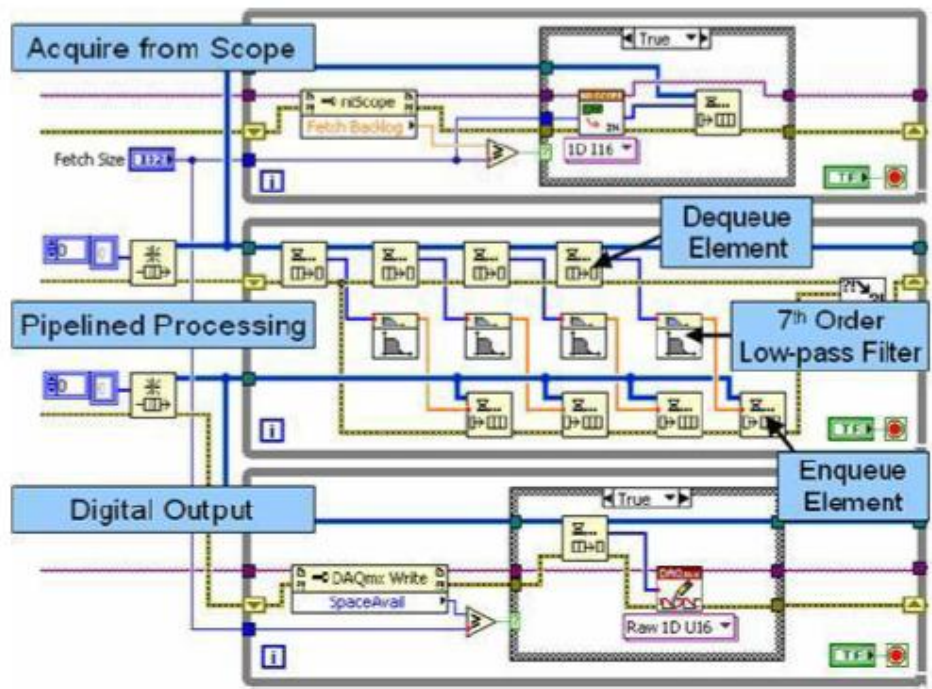


图 10. 程序框图显示多循环与队列结构的流水线信号处理

图 10 中，顶层循环是生产者循环，用于从高速数字化仪采集数据，并将其送到第一个队列结构(FIFO)。中间层循环同时作为生产者循环和消费者循环，它在每次重复中上载（消费 or 消耗？）若干来自队列结构的数据集，并以流水线方式独立处理这些数据集。这种流水线方式能配合多核处理器独立处理四个数据集，从而提高性能。需要注意的是，中间层循环还同时作为生产者，将处理后的数据送到第二个队列结构。最终，底层循环将处理后的数据送到高速数字化 I/O 模块。

并行处理算法提高了多核 CPU 的处理器利用率。事实上总吞吐率主要取决于两个因素：处理器利用率和总线传输速度。一般来说，CPU 和数据总线在处理大块数据时效率更高。当然，您还可以通过使用 PXI Express 仪器进一步缩短数据传输时间，获得更快的传输速度。

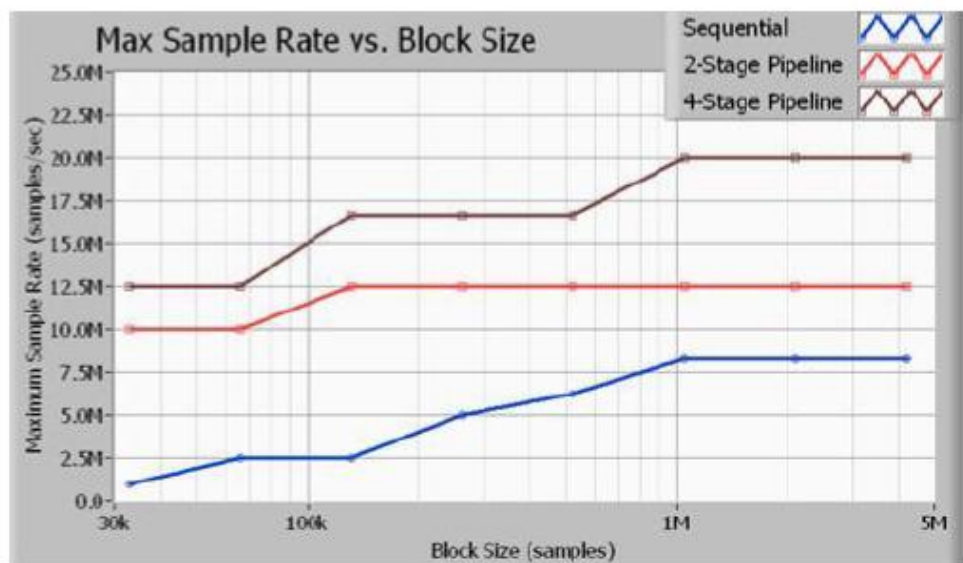


图 11. 多循环结构的吞吐量比单循环结构大许多

图 11 根据采样点数据量的大小，显示了不同采样率下的最大吞吐量。上述曲线基准均为 16 位采样。此外，信号处理算法均使用 7 阶巴特沃斯低通滤波器，截止频率为 0.45x 采样频率。数据显示，4 级流水线（多循环）方式下的数据吞吐量最大。需要注意的是，2 级信号处理方式能获得比单循环方法（顺序）更优的性能，但对处理器的利用率低于 4 级流水线方式。列出的采样率是 NI PXIe-5122 高速数字化仪和 NI PXIe-6537 高速数字 I/O 模块的最大输入输出采样率。注意在 20 MS/s 时，应用总线的输入输出数据率均为 40 MB/s，此时总线的总带宽为 80 MB/s。

流水线处理方式会导致输入与输出间的延迟，这个现象需要重点考虑。该延迟取决于几个因素，包括数据块大小和采样率。表 1 和表 2 比较了不同数据块大小和不同采样率下，对单循环和 4 级循环结构的延迟测量。

Single Loop Latency Benchmarks			4-Stage Pipeline Latency Benchmarks		
Block Size	Sample Rate (Max)	Latency (milliseconds)	Block Size	Sample Rate (Max)	Latency (milliseconds)
32k	1 MS/s	2.50 ms	32k	12.5 MS/s	38.78 ms
64k	2.5 MS/s	5.62 ms	64k	12.5 MS/s	45.41 ms
128k	2.5 MS/s	11.56 ms	128k	16.67 MS/s	38.27 ms
256k	5 MS/s	22.03 ms	256k	16.67 MS/s	44.86 ms
512k	6.25 MS/s	44.22 ms	512k	16.67 MS/s	55.17 ms
1M	8.25 MS/s	85.63 ms	1M	20 MS/s	148.85 ms
2M	8.28 MS/s	169.52 ms	2M	20 MS/s	247.29 ms
4M	8.25 MS/s	199.62 ms	4M	20 MS/s	581.15 ms

表 1、表 2. 表中显示了单循环与 4 级循环基准下的延迟

正如预期中的，随 CPU 利用率越接近 100%，延迟逐渐增大。在 20 MS/s 采样率的 4 级流水线结构中上述现象尤其明显。与此相对的，单循环中 CPU 利用率很少高于 50%。

结论

先进的多核处理器技术及更高传输速度的数据总线使得基于 PC 的仪器，诸如 PXI 及 PXI Express 模块化仪器受益良多。随着多核技术的发展，新 CPU 性能的不断提高，并行或流水线处理结构将成为最大化 CPU 利用

率的关键。LabVIEW 恰恰通过动态分配处理器任务解决了对编程的挑战。正如上文所述，使用 LabVIEW 算法结构实现并行处理能够获得极大的性能提升。