利用 LabVIEW 优化多核处理器环境中的自动化测试应用

概览

LabVIEW 为自动化测试应用提供了独特的、易于使用的图形化编程环境。然而，真正改善在多核处理器上的执行速度，却是其动态分配代码到各种 CPU 核的能力。本文将学习如何利用并行编程技术以优化 LabVIEW 应用。

多线程编程的挑战

迄今，处理器技术领域中的创新已经使得计算机具备了工作于更高时钟速率的中央处理器单元（CPU）。然而，随着时钟速率逼近其理论上的物理极限，具有多个（而不是单个）处理核的新型处理器正处于研发之中。利用这些新型多核处理器，自动化测试应用利用并行编程技术，便可以达到最佳性能和最高吞吐量。Edward Lee 博士——加州大学伯克利分校电气与计算机工程系的杰出教授——这样描述并行处理的技术优势：

"*许多技术专家预言，回应摩尔定律的终结的将是日趋并行的计算机架构。如果我们希望继续提高计算性能，计算机程序必须能够利用这种并行机制。*"

而且，利用多核处理器的编程应用是一个巨大的编程挑战，这是广为接受的。比尔盖茨——微软公司的缔造者——关于这一挑战有这样一段话：

"*要想充分利用并行工作的处理器的威力，...软件必须能够处理并发性问题。但正如任何一位编写过多线程代码的开发者告诉你的那样，这是编程领域最艰巨的任务之一。*"

幸运的是，LabVIEW 为多核处理器提供了一个理想的编程环境，因为它为创建并行算法提供了一个直观的环境，而且它可以动态指派多个线程至一项给定的应用。事实上，利用多核处理器的自动化测试应用，可以方便地被优化以获取最佳性能。而且，PXIe 模块化仪器增强了这一技术优势，因为 PCIe 总线使高数据传输速率成为可能。从多核处理器和 PXIe 仪器的两个具体应用是：多通道信号分析和线上处理（硬件在环）。在本文的后续部分，我们将评估各种并行编程技术，并刻画每项技术所带来的性能优势。

实现并行测试算法

得益于并行处理的一项常见自动化测试应用（ATE），便是多通道信号分析。由于频率分析是一项占有处理器较多的操作，通过并行化处理测试代码使得每个通道的信号处理被分配至多个处理器核，可以提高执行速度。从编程人员的角度来看，为获得这一技术优势，所需的唯一改变便是仅仅重构测试算法。

为描述这一过程，我们将比较用于多通道频率分析（傅立叶变换或 FFT）的两个算法的执行时间，它们分别位于一个高速数字化仪的两个通道上。在该测试中，我们使用 PXIe-5122 14-位高速数字化仪

的两个通道，以最高采样率（100 MS/s）采集信号。首先，我们这一操作在 LabVIEW 中的传统的顺序编程模型。

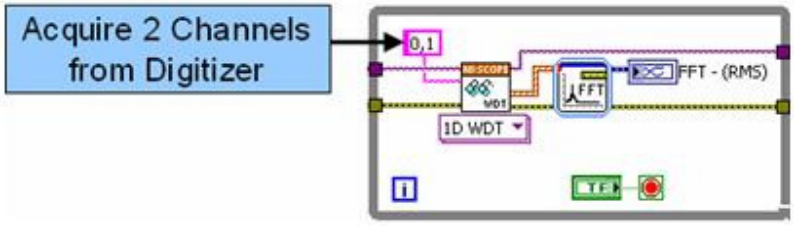

图 1. 利用顺序执行的 LabVIEW 代码

采集来自数字化仪的两个通道的信号

在上述模块框图中，两个通道的频率分析均在一个 FFT 快速 VI 中完成，它顺序分析每个通道信号。虽然上述算法也可以在多核处理器中有效执行，但仍存在通过并行处理每个通道提高算法性能的可能。

如果我们剖析上述算法，我们会发现完成 FFT 所需的时间要比从高速数字化仪采集数据长得多。通过每次获取一个通道的数据并并行执行两次 FFT，我们可以显著降低处理时间。下图表示了一个采用并行方法的新的 LabVIEW 模块框图。

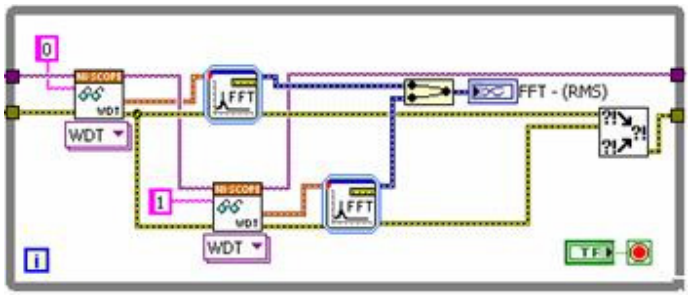

图 2. 利用并行执行的 LabVIEW 代码

如上面代码所示，将顺序获取数字化仪的每个通道的数据。注意，如果两次数据获取均来自不同的仪器，那么完全可以并行完成这些操作。然而，由于傅立叶变换占用大量的处理器时间，我们仍可以仅通过将信号处理并行化改善性能。故而减少了总的执行时间。两种实现的执行时间如下所示：

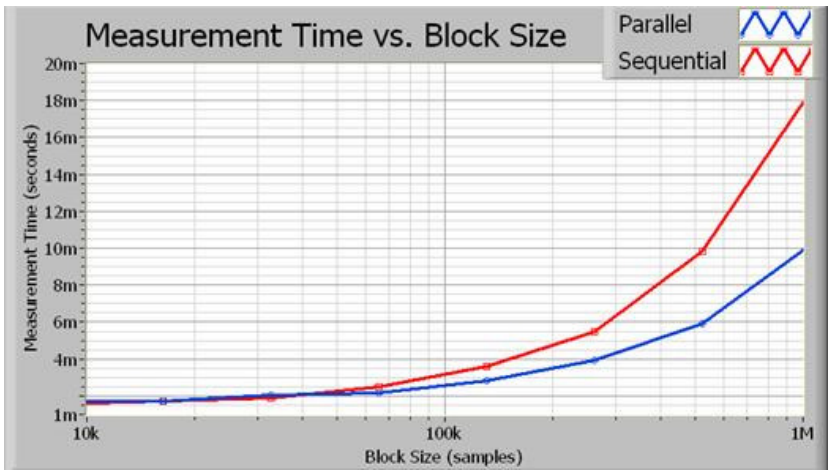

图 3. 顺序算法与并行算法（时间）的执行时间比较

Measurement Time vs. Block Size 测量时间与数据块大小
Parallel 并行
Sequential 顺序
Measurement Time (seconds)测量时间（秒）
Block Size (samples)数据块大小（采样数）

如上图所示，随着数据块大小（每次获取的采样数）的增加，通过并行处理节约的处理时间愈为显著。事实上，对于更大的数据块，并行算法实现近 2 倍的性能改进。下图描述了性能增长的精确百分比随采集数据块大小（以采样数为单位）的变化。

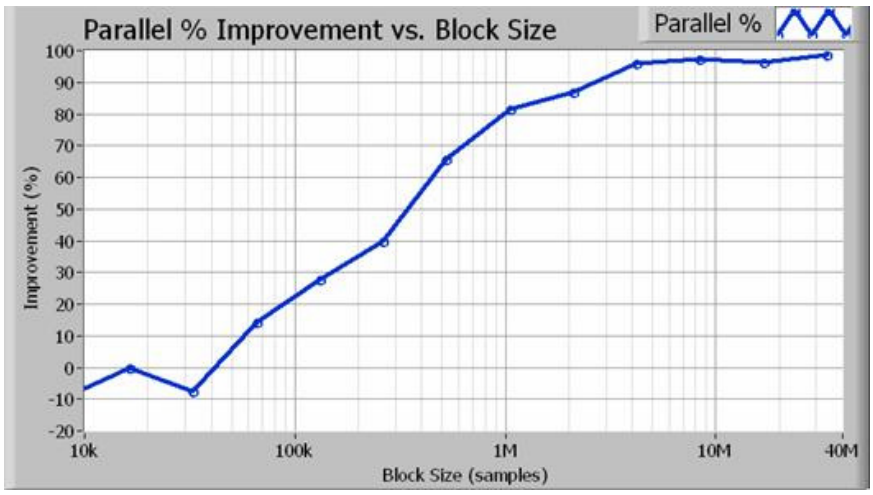

图 4.并行算法带来的性能增长（百分比）

Parallel % Improvement vs. Block Size 并行性能增长百分比与数据块大小
Parallel 并行
Improvement 性能增长
Block Size (samples)数据块大小（采样数）
图 4 显示，当数据块大于 1 百万采样（100 Hz 精度带宽）时，并行方式实现 80%或更高的性能增长。

在多核处理器之上，我们可以方便地实现自动化测试应用的性能改进，因为 LabVIEW 动态地分配每一个线程。事实上，用户不需要创建特殊的代码以支持多线程，而是通过最少的编程调整，并行测试应用便可以获益于多核处理器。

配置定制的并行测试算法

将信号处理并行化的技术优势在于它支持 LabVIEW 在多个处理器核中划分 CPU 的用度。在下图中，我们描述了 CPU 处理算法每一部分的次序。

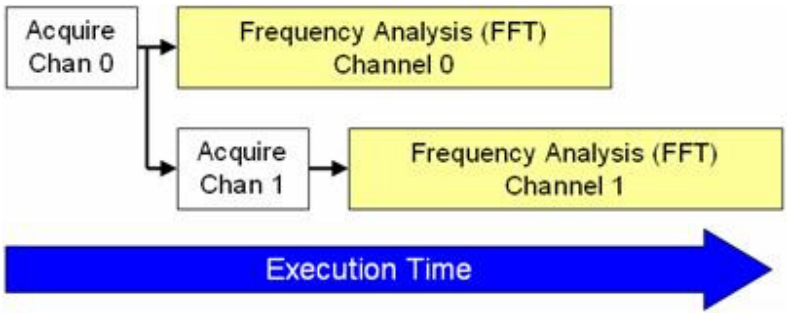

图 5. CPU 的处理执行

Acquire Chan 0 采集通道 0
Frequency Analysis （FFT） Channel 0 通道 0 的频率分析（FFT）
Acquire Chan 1 采集通道 1
Frequency Analysis （FFT） Channel 1 通道 1 的频率分析（FFT）
Execution Time 执行时间

如图所示，LabVIEW 能够并行处理许多采集数据，从而节省了执行时间。对于 LabVIEW，并行处理的需求之一便是拷贝（或克隆）每个信号处理子例程。缺省情况下， LabVIEW 的许多信号处理算法配置为"重入执行"。这就意味着 LabVIEW 将动态分配每个子例程的一个不同实例，包括独立线程和存储空间。因而，定制子例程必须被配置为工作于重入方式。这可以通过 LabVIEW 中一个简单的配置步骤完成。欲设置这一属性，选择文件菜单下 VI 属性并选中"执行"栏；然后，选中"重入执行"标记（如下所示）。

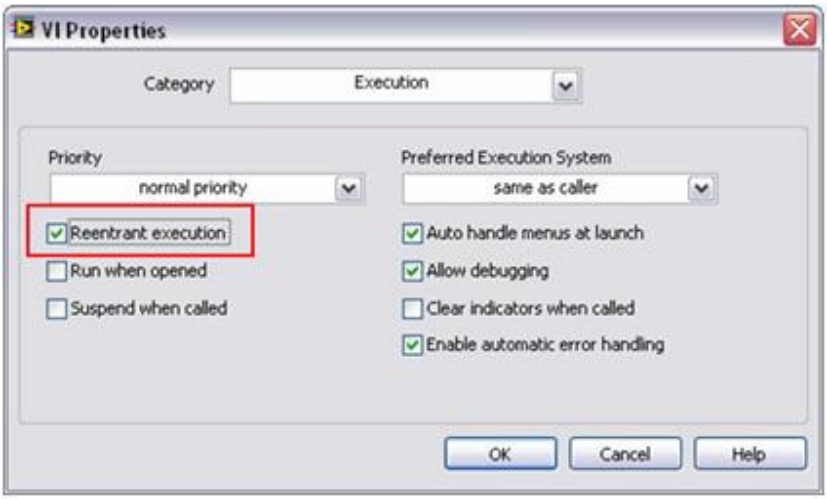

图 6. 在 LabVIEW 中配置重入执行属性

通过上图所示的简单步骤，我们可以并行执行多个定制子例程，就如同标准 LabVIEW 分析函数那样。因此，在多核处理器之上，自动化测试应用通过简单的编程技术就可以实现性能的改进。

优化硬件在环应用

得益于并行信号处理技术的又一个应用便是为同时输入与输出使用多个仪器。一般，这些应用被称为硬件在环（HIL）或在线处理应用。在此场景下，高速数字化仪或高速数字 I/O 模块用于信号采集，其软件执行数字信号处理算法。最后，通过另一个模块化仪器生成结果。其典型模块框图如下所示：

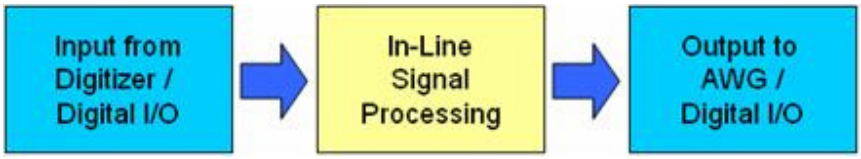

图 7. 在线信号处理（HIL）模块框图

Input from Digitizer/ Digital I/O 来自数字化仪/数字 I/O 的输入
In-Line Signal Processing 在线信号处理
Output to AWG/Digital I/O 输出至 AWG/数字 I/O

常见 HIL 应用包括在线数字信号处理（滤波、插值等）、传感器仿真和定制组件模拟。在这篇特别准备的白皮书中，我们将探究用于在在线数字信号处理应用中获得最佳吞吐量的技术。

通常可以使用两种基本的编程结构，单循环结构和带有队列的管道式多循环结构。单循环结构实现简单，对于小数据块具有低时延。相比之下，多循环结构能够支持高得多的吞吐量，因为它们能够更好地利用多核 CPU。

对于传统的单循环方式，一个高速数字化仪的读函数、信号处理算法和高速数字 I/O 顺次组织。如下面模块框图所示，这些子例程中的每一个都必须按照 LabVIEW 数据流编程模型确定的顺序执行。

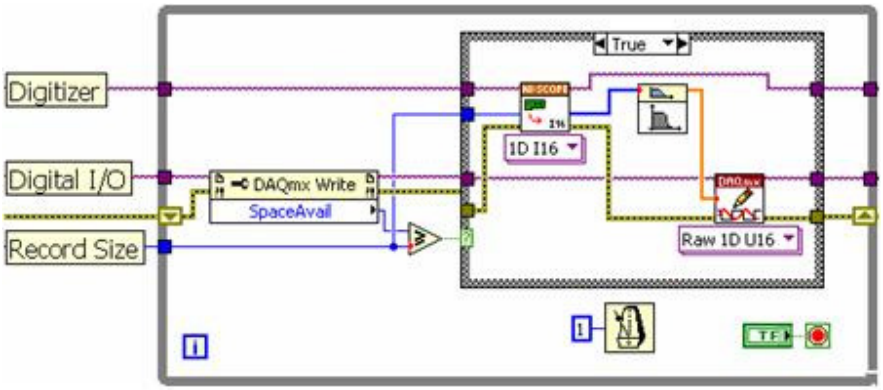

图 8. 在循环中依单循环方式进行处理

Digitizer 数字化仪
Digital I/O 数字 I/O
Record Size 记录大小

单循环结构受限于几个因素。由于顺序执行每一环节，处理器在处理数据的同时受限无法执行仪器 I/O。在这种方式下，由于处理器一次只能执行一个函数，所以无法有效利用多核 CPU。因而，在应用中仅使用了多核 CPU 的一个核。虽然单循环结构足以处理较低的采集速率，但是要想得到较高的数据吞吐量仍需要采用多循环方式。

多循环架构使用队列结构实现 while 循环间的数据传递。下面，我们展开论述在 while 循环间采用一个队列结构进行数据流编程的概念。

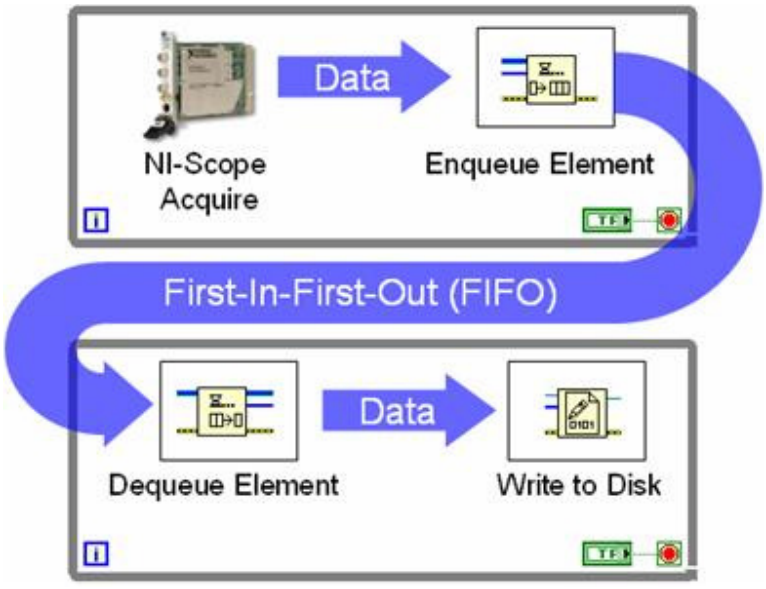

图 9. 队列结构支持多循环间的数据共享

Data 数据

NI-Scope Acquire　　NI-Scope 采集
Enqueue Element 入队元素
First-In-First-Out （FIFO）先进先出
Dequeue Element 出队元素
Write to Disk 写入磁盘

如图所示，队列支持多个循环间的数据共享。上图所表示的是典型的所谓生产者/消费者循环结构。这里，在一个循环中，一个高速数字化仪持续采集数据，并在每次迭代中将新的数据集传递至 FIFO 队列。消费者循环仅需监视队列的状态，当每个数据集可用时将其写入磁盘。采用队列的意义在于这两个循环均可相互独立执行。在上例中，高速数字化仪可以持续采集数据，即使这些数据写入磁盘时存在一定的延迟。与此同时，其它的采样仅需存储在 FIFO 队列中。通常，生产者/消费者管道式方法，通过支持更有效的处理器利用率，使更高的数据吞吐量成为可能。这一技术优势在多核处理器中甚至更为显著，因为 LabVIEW 可以动态分配 CPU 线程至每个处理器核。

对于一项在线信号处理应用，我们可以使用三个独立的 while 循环和两个队列结构，实现其间的数据传递。在此应用场景下，一个循环将从一台仪器采集数据，一个循环将专门执行信号处理，而第三个循环将数据写入到另一台仪器。描述这一方式的 LabVIEW 模块框图如下所示：

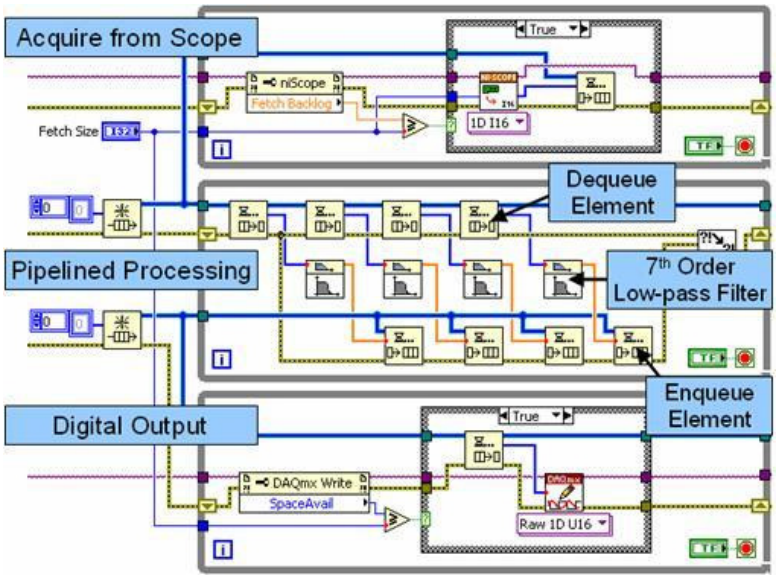

图 10. 具有多个循环和排队结构的管道式信号处理

Acquire from Scope 自 Scope 采集数据
Dequeue Element 出队元素
Pipelined Processing 管道式处理
7th Order Low-Pass Filter　　7 阶低通滤波器
Digital Output 数字输出
Enqueue Element 入队元素

在上图中，最上面的循环是一个生产者循环，它从一个高速数字化仪采集数据，并将其传递至第一个队列结构（FIFO）。中间的循环同时作为生产者和消费者工作。每次迭代中，它从队列结构中卸载（消费）若干个数据集，并以管道的方式独立对其进行处理。这种管道方式通过支持高达四个数据集

的独立处理，实现了多核处理器环境下的性能改进。注意，中间的循环同时也作为一个生产者工作，将处理后的数据传递至第二个队列结构。最后，最下面的循环将处理后的数据写入至高速数字 I/O 模块。

并行处理算法改善了多核 CPU 的处理器利用率。事实上，总吞吐量有赖于两个因素，处理器利用率和总线传输速度。通常，CPU 和数据总线在处理大数据块时工作效率最高。而且，我们可以进一步使用具有更快传输速度的 PXIe 仪器，缩减数据传输时间。因而，我们通过依采集数据大小（以采样数计）变化的采样率描述最大吞吐量，如下所示：

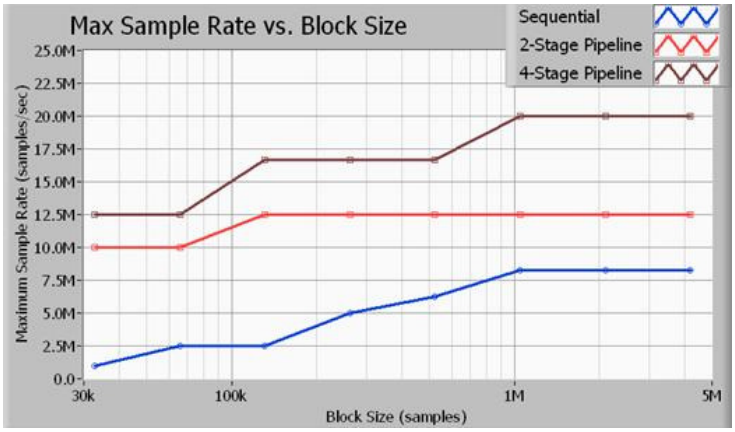

图 11. 多循环结构与单循环结构的吞吐量比较

Max Sample Rate vs. Block Size 最大采样率与数据块大小
Maximum Sample Rate (Samples/sec)最大采样率（采样数/秒）
Block Size (samples) 数据块大小（采样数）
Sequencial 顺序
2-Stage Pipeline　　2 阶段管道
4-Stage Pipeline　　4 阶段管道

该图中所描述的所有标定都是围绕 16-位采样进行的。此外，所采用的信号处理算法为一个截至频率为采样率的 0.45 倍的 7 阶巴特沃兹低通滤波器。如数据显示，4 阶段管道式（多循环）方式支持最大数据吞吐量。注意，2 阶段信号处理方式获得了比单循环方式（顺序）更好的性能，但其 CPU 的利用不及 4 阶段方式有效。上面所列的采样率为 PXIe-5122 高速数字化仪和 PXIe-6537 高速数字 I/O 模块的输入和输出的最大采样率。注意，当采样率为 20 MS/s 时，应用总线的输入和输出的数据传输率均为 40 MB/s，所以总的总线带宽为 80 MB/s。

应当考虑的是，管道式处理方式在输入与输出之间确实引入了时延。所引入的时延取决于几个因素，包括数据块的大小和采样率。下表比较了单循环和 4 阶段多循环架构中实测所得的时延随数据块大小和最大采样率的变化情况。

Single Loop Latency Benchmarks

Block Size	Sample Rate (Max)	Latency (milliseconds)
32k	1 MS/s	2.50 ms
64k	2.5 MS/s	5.62 ms
128k	2.5 MS/s	11.56 ms
256k	5 MS/s	22.03 ms
512k	6.25 MS/s	44.22 ms
1M	8.25 MS/s	85.63 ms
2M	8.28 MS/s	169.52 ms
4M	8.25 MS/s	199.62 ms

4-Stage Pipeline Latency Benchmarks

Block Size	Sample Rate (Max)	Latency (milliseconds)
32k	12.5 MS/s	38.78 ms
64k	12.5 MS/s	45.41 ms
128k	16.67 MS/s	38.27 ms
256k	16.67 MS/s	44.86 ms
512k	16.67 MS/s	55.17 ms
1M	20 MS/s	148.85 ms
2M	20 MS/s	247.29 ms
4M	20 MS/s	581.15 ms

表 1 和 2. 单循环和 4 阶段管道的时延标定

Single Loop Latency Benchmarks 单循环时延标定
Block Size 数据块大小
Sample Rate (Max)采样率（最大）
Latency (milliseconds)时延（毫秒）
4-Stage Pipeline Latency Benchmarks 4 阶段管线时延标定

正如人们可以预知的，当 CPU 的使用率接近 100%时时延也随之增加。这一点在采样率为 20 MS/s 的 4 阶段管道范例中尤为明显。相比之下，任何一个单循环范例的 CPU 使用率都几乎不会超过 50%。

总结

基于 PC 的仪器系统，如 PXI 和 PXIe 模块化仪器，从多核处理器技术的进步和数据总线速度的提高中获益匪浅。当新型 CPU 通过添加多个处理核改进性能时，并行或管道式处理结构成为最大化 CPU 效率所必需。幸运的是，LabVIEW 通过将处理任务动态分配至单个处理核；为这一编程挑战提供了一种上佳的解决方案。如上面数据显示，将 LabVIEW 算法结构化以利用并行处理，可以带来显著的性能提高。