Comparing GPIB, LAN/LXI, PCI/PXI Measurement Performance in Hybrid Systems

Alex B. McCarthy
National Instruments
11500 N. Mopac Expressway
Austin, TX 78759
512-683-5310
alex.mccarthy@ni.com

Faya Peng
National Instruments
11500 N. Mopac Expressway
Austin, TX 78759
512-683-6754
faya.peng@ni.com

Abstract - When designing automated test and measurement systems, developers can optimize system throughput by evaluating instrument control buses. Part of the optimization process includes defining a benchmark test, selecting the appropriate instruments and determining which instrument control bus provides optimal performance. While there are a variety of factors to consider when selecting an instrument, the instrumentation bus used has a large impact on the instrument performance. Many instrumentation buses are available for instrument control, such as GPIB, HS488, VXI, USB, LAN/LXI, PCI, PXI, and others. This paper briefly discusses a benchmarking technique for comparing instruments buses and shows actual measurement performance results across several instruments. Based on the testing methodology and sample results, system developers can measure actual bus performance and optimize their test systems.

INTRODUCTION

Engineers can use various techniques to optimize measurement performance in hybrid, or multi-platform, systems to decrease test times and increase throughput using various techniques. Determining which instruments and which instrumentation bus to use is part of the optimization process. In automated test systems the instrumentation bus used has a big impact on system performance. Many instrumentation buses are available for instrument control, such as GPIB, HS488, RS232, VXI, USB, Ethernet/LAN, PCI, and PXI (which uses the PCI and PCI Express buses). There is no ideal bus for every application - each bus has strengths and weaknesses that make it more or less suitable to a particular application.

With GPIB, users benefit from a proven instrumentation bus technology and the widest variety of instruments; with USB, developers can take advantage of the wide availability of USB ports and easy connectivity; and with Ethernet/LAN, system developers can create distributed applications across long distances. By using a modular instrumentation architecture such as PXI, developers can take advantage of open, multi-vendor standards and software flexibility to create a user-defined solution for their specific application needs.

LATENCY AND BANDWIDTH IMPACT ON BUS PERFORMANCE

While each bus has its strengths, specific buses offer better performance for certain applications than others. When evaluating bus performance, two important factors are latency and bandwidth. Latency measures the data transmission time, typically in seconds, and bandwidth measures the rate at which data is sent across the bus, typically in MB/s. Lower latency improves the performance of small data transfers that include digital multimeter (DMM) measurements, switching, and instrument configuration. Higher bandwidth is important in applications such as waveform generation and acquisition or RF measurements. Figure 1 compares the latency

1-4244-0052-X/06/$20.00 ©2006 IEEE

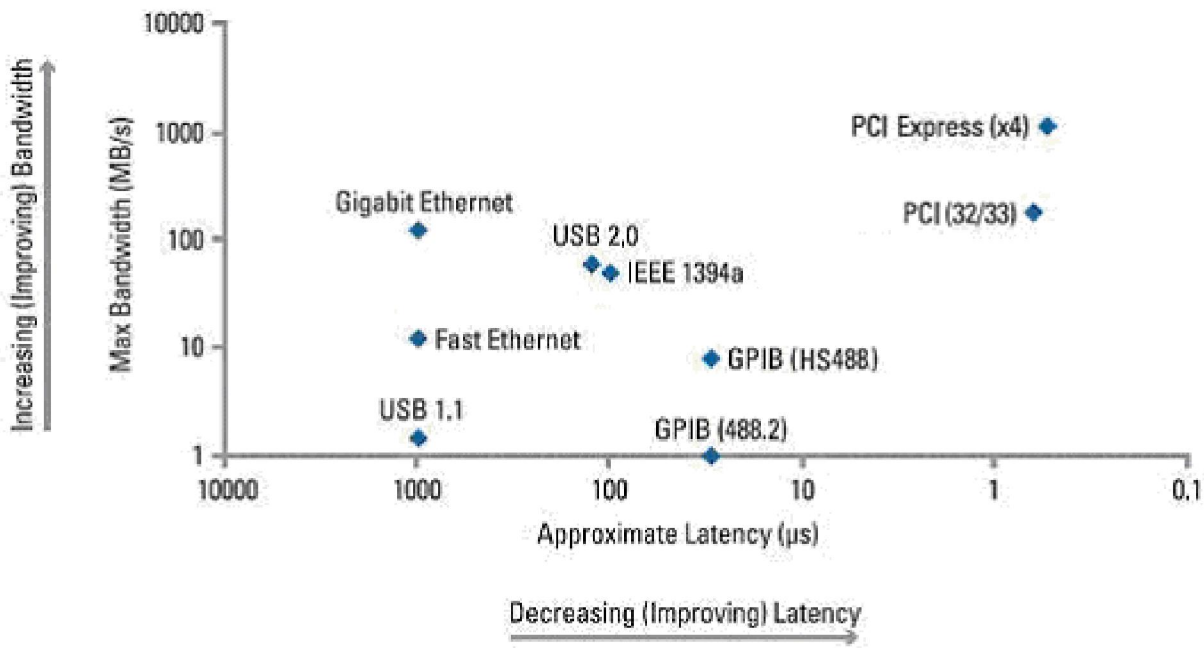

Figure 1. PCI Express, including PXI Express, provides the best combination of high bandwidth and low latency.

and bandwidth of various instrumentation buses. Note that improving or increasing bandwidth moves up while improving or decreasing latency moves to the right. By becoming aware of the sample size and type of communication used, developers can factor in the latency and bandwidth of the buses and choose the appropriate instrumentation bus to optimize the throughput of their system.

BENCHMARK TEST METHODOLOGY

This paper investigates how the instrumentation bus affects instrument throughput by evaluating the results from benchmarks for acquiring a waveform. To measure the impact of the bus, developers conducted benchmark tests using three different oscilloscopes or digitizers across five different buses. The results largely reflect instrumentation bus performance based on latency and bandwidth but are also dependent on instrument implementation and the specific application usage.

These benchmarks measured the performance seen from the user's perspective in acquiring a waveform. The time measured for acquiring a waveform includes the complete round-trip process of these four phases:

1. Sending a command from the application programming interface (API),
2. Digitizing the waveform and loading the output buffer,
3. Sending the data across the bus, and
4. Receiving and displaying the data on the computer

Developers conducted these benchmarks on a 3.0 GHz, Pentium 4 computer running Windows XP using the LabVIEW application development environment. To communicate with the Agilent MSO6032A and the Tektronix TDS 5104 instruments, VISA commands provided direct I/O communication to the instruments. To send commands to the NI 5122, the developers used the driver provided with the instrument.

By using a systematic benchmark test, developers can record the degree to which changes in the instrumentation bus and/or instrument affects system throughput. Developers used a high-precision timer and a standard set of typical instrument control tasks. To properly measure the time to acquire a waveform, these benchmark tests took advantage of the high-resolution timer provided Windows XP. Two functions provide access to the counter value and frequency. QueryPerformanceCounter provides the current counter value and the QueryPerformanceFrequency provides the frequency or rate at which the counter increments in counts/s. These calls provide a high-resolution counter because the counter frequency is usually a derivative of the CPU frequency. The counter accuracy is affected by the clock handling hardware, BIOS software, and Windows adds delays and jitter. The resulting timer resolution is machine dependent and is usually in the low µs range. It is important to determine timer resolution as part of developing the benchmark. To measure the time required to perform a waveform acquisition, the developers followed the process shown in Figure 2.

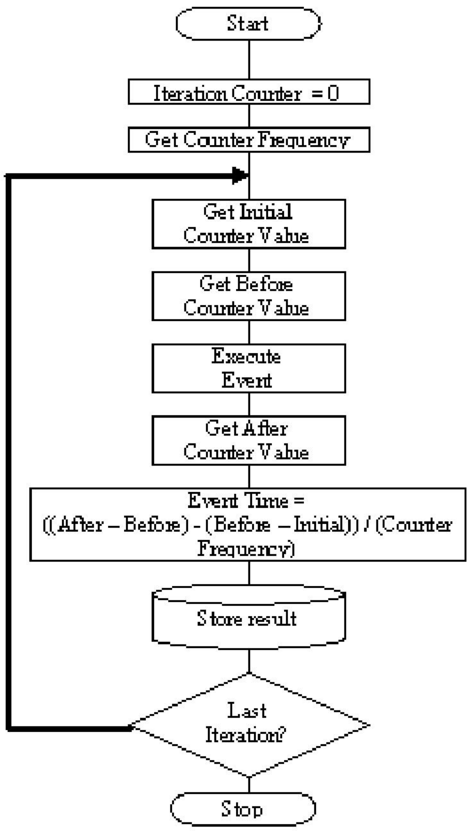

Figure 2. Instrument Throughput Benchmark Methodology

First, the developers used the QueryPerformanceFrequency function to obtain the counter frequency. Then the QueryPerformanceCounter function is called three times; the first time, it provides the Initial Counter Value, and then it is immediately called again to provide the Before Counter Value, the counter value before the command to acquire the waveform is sent. Once the data has been retrieved by the computer, it is called to provide the Final Counter Value. The QueryPerformanceCounter call overhead is calculated by subtracting the Initial Counter Value from the Before Counter Value. Then, the actual measurement time is calculated by taking the difference from the Final Counter Value and the Before Counter Value and then subtracting the call overhead.

Using these functions, the developers recorded the time required for the complete round-trip process to fully depict the time required to acquire a waveform from sending the command to receiving the data.

TEST RESULTS AND OBSERVATIONS

Using three instruments, the benchmarks compare GPIB, HS488, Ethernet/LAN, USB, and PCI/PXI, for acquiring waveforms of three sample sizes: 500, 1000, and 1,000,000 samples. The Tektronix TDS 5104 oscilloscope supports GPIB, HS488, and Ethernet while the Agilent MSO6032A oscilloscope supports GPIB, Ethernet, and USB. With an NI 5122 digitizer, the test acquired benchmark data for PCI and is a good surrogate for PXI.

To gain more insight from these benchmarks, developers can evaluate which phases of the waveform acquisition are uniform and which vary depending on the bus and instrument:

1. Sending a command from the application programming interface (API);
2. Digitizing the waveform and loading the output buffer;
3. Sending the data across the bus; and
4. Receiving and displaying the data on the PC.

With phases 1 and 4, developers can assume that the PC time is constant; that is, the time taken to send and process the command as well as the time taken for the computer to receive and parse the data is comparable in all cases. In addition, the time required for phase 2 includes a standard minimum measurement time that is independent of the bus and instrument which is calculated from the sample size and sampling rate. Phase 3 reveals the differences between the buses and instruments since it largely depends on bus performance in transferring the data and the method in which the instrument acquires and loads the output buffer. The raw results are displayed in Table 1.

Sample Size	Instrument	Bus	Waveform Time (µs)	Throughput (MB/s)
500	NI 5122	PCI[PXI]	7,033.19	0.0711
	Tek TDS 5104	GPIB	8,877.71	0.0563
	Tek TDS 5104	HS488	9,233.42	0.0542
	Tek TDS 5104	Ethernet	14,294.39	0.0350
	Agilent MSO6032A	Ethernet	69,947.38	0.0071
	Agilent MSO6032A	GPIB	70,940.31	0.0070
	Agilent MSO6032A	USB	71,694.10	0.0070
1,000	Tek TDS 5104	HS488	6,676.78	0.1498
	NI 5122	PCI[PXI]	6,901.53	0.1460
	Tek TDS 5104	GPIB	7,078.84	0.1413
	Tek TDS 5104	Ethernet	9,983.38	0.1002
	Agilent MSO6032A	Ethernet	70,725.43	0.0141
	Agilent MSO6032A	USB	71,712.53	0.0139
	Agilent MSO6032A	GPIB	72,369.68	0.0138
1,000,000	NI 5122	PCI[PXI]	48,342.02	20.6879
	Agilent MSO6032A	USB	272,040.17	3.6759
	Tek TDS 5104	Ethernet	880,624.36	1.1356
	Agilent MSO6032A	Ethernet	1,109,983.93	0.9009
	Tek TDS 5104	HS488	1,325,664.74	0.7543
	Agilent MSO6032A	GPIB	1,406,901.97	0.7108
	Tek TDS 5104	GPIB	1,554,999.33	0.6431

Table 1. Benchmark Comparison of Instruments and Buses Based on Sample Size

Based on these four phases, a few observations can be made when examining the waveform time required for the various sample sizes and sampling rates. At 500 samples with a 250 kS/s sampling rate, the minimum time possible for this measurement is 2000 µs, calculated by (500 samples)/ (250 kS/s), which yields 2000 µs. This value serves as a baseline when comparing how the various buses perform in transferring the data across the buses. Figure 3 shows the performance for each instrument and bus combination and compares it to the baseline minimum measurement time. Because the longer waveform time of the Agilent MSO6032A is seen for all three buses used (GPIB, Ethernet, and USB), it is most likely attributed to a common implementation bottleneck rather than individual bus performance.

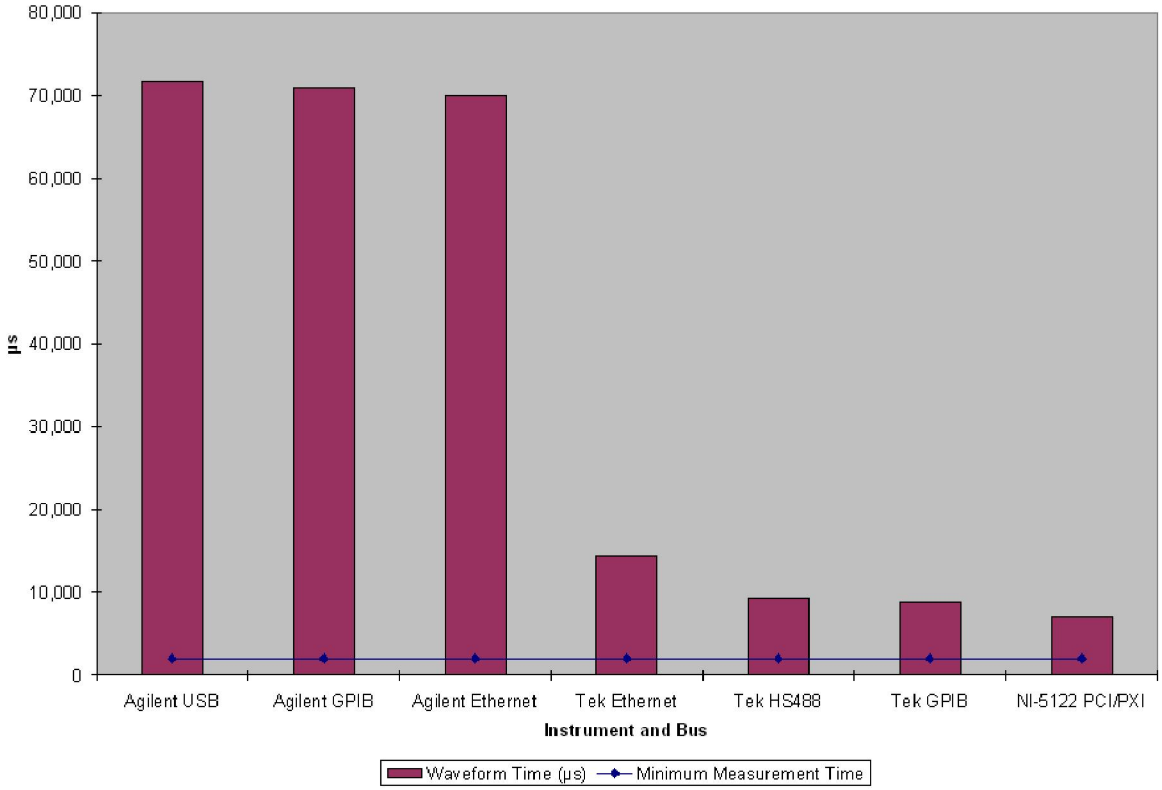

Figure 3. Instrument and Bus Performance for 500 samples (S)

Comparing the Tektronix TDS 5104 performance on GPIB, HS488, and Ethernet with the NI 5122 on PCI/PXI provides some insight about bus performance. Because the waveform size is smaller, the latency of the bus has a larger impact on the performance. Typically, parallel buses such as PCI/PXI and GPIB have better latency than a serial bus such as Ethernet. Because PCI/PXI has the lowest latency of all the buses, it is reasonable to expect the NI 5122 on PCI/PXI perform 1.27X better than the Tektronix TDS 51204 on GPIB and 2X better than the Tektronix TDS 51204 on Ethernet for a 500 sample waveform.

Figure 4 shows the results when comparing the data for the 1000 sample waveform at a sampling rate of 500 kS/s. Many of the same ideas apply. To use as a baseline reference, the minimum measurement time is also 2000 µs. The slower performance for the Agilent MSO6032A for all three buses is once again likely attributed to the instrument implementation. To compare the speeds of different buses, it is more appropriate to compare the performance of the Tektronix TDS 5104 with the NI 5122. The results from the Tektronix TDS 5104 on Ethernet are still noticeably slower than that of GPIB, HS488, and PCI/PXI because measurements of this sample size are still primarily dependent on the latency of the bus, and the latency of Ethernet is slower than the other buses, as seen in Figure 1. The benchmark results for 1000 sample waveform demonstrate the benefit of HS488 and PCI/PXI.

HS488, a higher-speed GPIB transfer protocol, scales the maximum data transfer rate of ANSI/IEEE Standard 488.1-1987 up to 8 MB/s by removing delays in the 3-wire IEEE 488.1 handshake. Using the HS488 protocol, the GPIB controller hardware can automatically detect compatible devices capable of using the HS488 handshake to transfer data. If the controller does not detect an HS488 capable device, it automatically defaults to the standard IEEE 488.1 3-wire handshake to complete the data transfer.

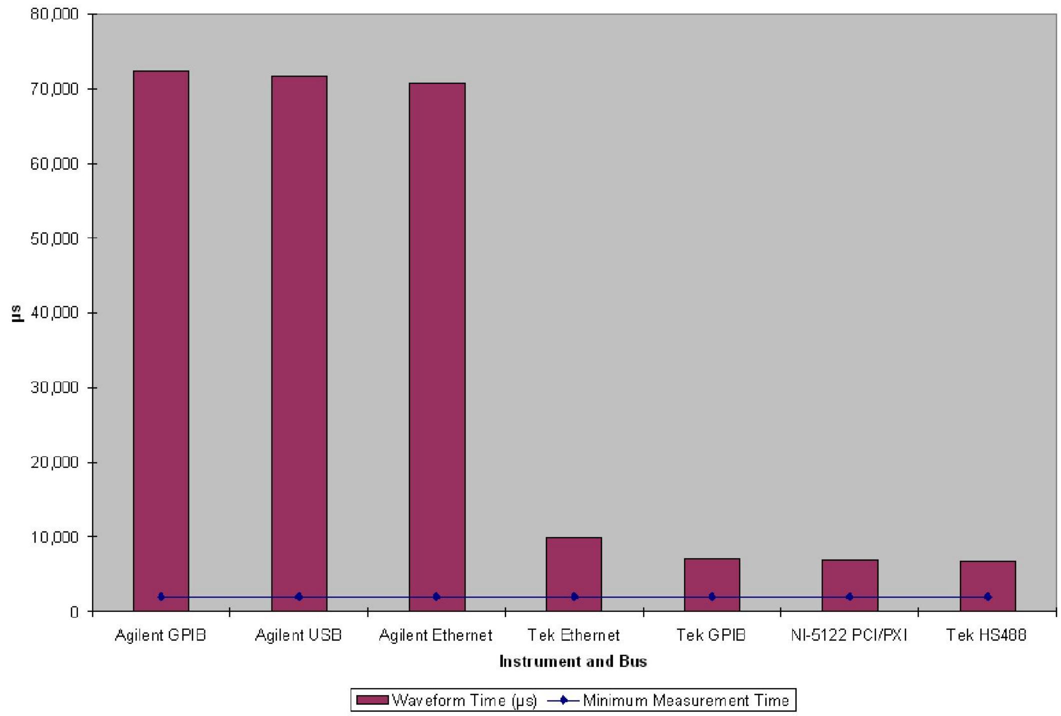

Figure 4. Instrument and Bus Performance for 1 kS

With the large sample size of 1 MS at 100 MS/s, shown in Figure 5, these benchmarks are more dependent on the bandwidth of the bus. Here the minimum measurement time is 10,000 µs, but because the waveform size is so large, the performance is largely based on the ability of the bus to quickly transfer the data. Because Ethernet and USB have higher bandwidth capabilities than GPIB, the better performance seen in the Tektronix TDS 5104 on Ethernet and the Agilent MSO6032A on Ethernet and on USB is expected. In addition, because PCI/PXI, has a much higher bandwidth and the NI 5122 efficiently implements retrieving the waveform data, an NI 5122 on PCI/PXI outperforms the Agilent MSO6032A on USB by 5.6X, the Tektronix TDS 5104 on Ethernet by 18.7X, and the Agilent MSO6032A on Ethernet by 22.8X.

CONSIDERTATIONS FOR IMPROVING PERFORMANCE

From these benchmarks, developers have a few considerations to improve the throughput of a system. The benchmark results demonstrated the importance of the first two considerations, latency and bandwidth. Because the instrumentation bus bandwidth and latency have a large impact on performance, developers can select a bus with bandwidth and latency more suitable for their applications. For applications with a small sample size, such as, 500 samples, using PCI/PXI and GPIB provides the best performance because these buses have lower (better) latency. When the application has a small-to-medium sample size, such as 1000 samples, latency is still of primary importance, but higher bandwidth does help. In this case, HS488, PCI/PXI, and GPIB perform well. For large sample sizes, such as 1 MS, it is important to use a bus with high bandwidth. With higher bandwidth, PCI/PXI and USB are well-suited to handle applications with large sample sizes.

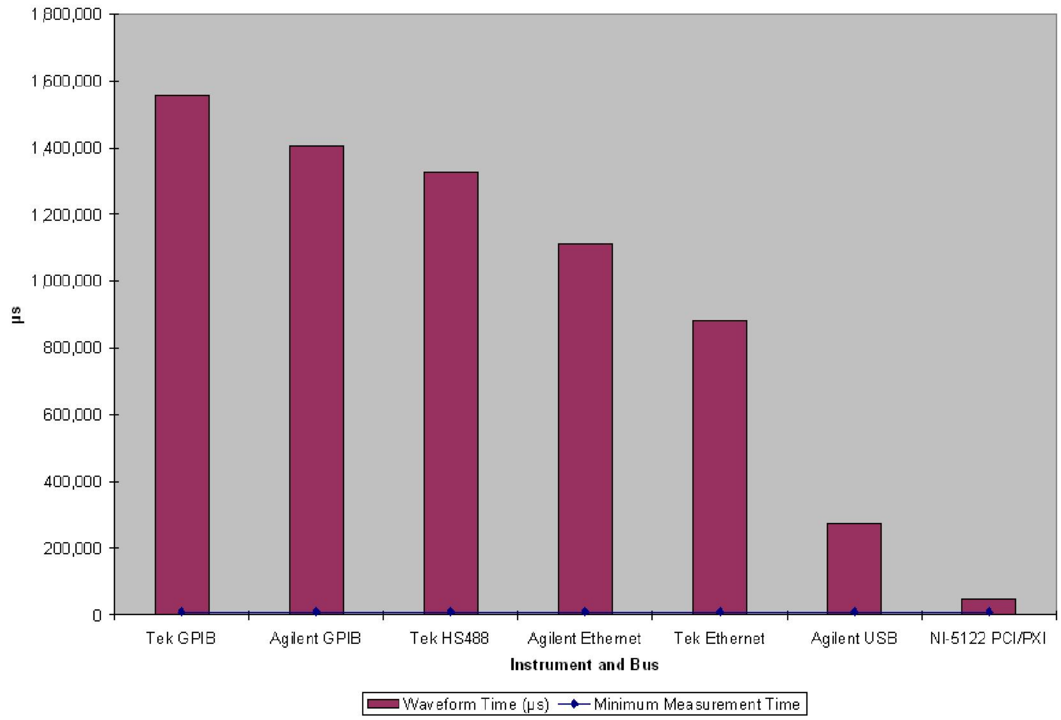

Figure 5. Instrument and Bus Performance for 1 MS

The third consideration to improve system performance is to evaluate instrument implementation for the types of sample sizes used in the application. Because of instrument implementation, the Agilent scope did not perform well for small and small-to-medium sample sizes, but had a good implementation for large sample sizes. Examining how the instrument implements various instrument control tasks provides guidance on optimal instrument selection.

A good way to balance the three factors of system performance – latency, bandwidth, and instrument implementation – is to profile the instruments available to determine which is best suited for the application. By using a high-precision timer such as those offered in Windows and a standard set of typical instrument control tasks, developers can develop a repeatable performance benchmark for measurement time for specific application.

CONCLUSIONS

Because of the variety of instruments available, developers have multiple considerations to evaluate in improving the throughput of their test and measurement applications. To improve throughput, developers should consider the bus latency, bus bandwidth, and instrument implementation when designing a system and evaluating instruments. For small sample sizes, PCI/PXI and GPIB provide high performance because of their low latency. For small-to-medium sample sizes, HS488, PCI/PXI, and GPIB performed well. Because performance for large sample sizes depends on bandwidth, PCI/PXI, USB, and Ethernet are good choices. Because of the variance in vendor implementations, developers also should consider the implementations of the instruments; for instance, some perform better for larger sample size than for smaller sample sizes. By factoring in the types of communication used for the application and profiling their instruments, developers can select instruments that perform better for those specific tasks to improve throughput. By taking into consideration these various factors, developers can improve the throughput of an existing system.