Document Type: Tutorial
NI Supported: Yes
Publish Date: Feb 1, 2011

Building Distributed Monitoring and Control Systems With LabVIEW

Overview

This paper examines the benefits of creating distributed monitoring and control systems as well as the fundamentals of building these systems. Engineers can use the concepts presented in this paper to design reliable, high-performance systems that can easily integrate with future technologies.

Table of Contents

Introduction

By implementing a distributed measurement and control system, you can optimize the processes running on each machine and over the network, creating a more reliable and higher-performance system. Distributed systems are made up of two pieces: the backbone of the system and the nodes.

When designing a system, first define what you need to accomplish. Will the system need to both control and monitor? Which unique business processes do you want to integrate into the system, for example, a purchasing database or quality tracking database? Identifying the integral processes up front is important to properly architect the system. Choosing the right tools can help you build commonly used features and achieve the flexibility to incorporate custom processes or equipment from third-party vendors. NI LabVIEW is a flexible development environment that is designed to help you easily integrate many disparate components into a complete monitoring and control system.

Although this paper concentrates primarily on software tools, National Instruments also offers general-purpose data acquisition hardware as well as optimized devices for control and measurement analysis.

Defining the Backbone

When thinking about a distributed system, consider the entire system first. At the top level is the backbone of the system. The backbone of a distributed system can be simplified to key servers and the network. The software running on the key servers must manage network transfers, data management, data visualization, alarms and events, and security. A key characteristic of the backbone is that it must be able to communicate with the rest of the hardware through a common protocol, such as TCP/IP. The software used at each machine in the network must support the same communication protocols. In addition, the key servers should work with a variety of communication protocols so they can interface with legacy and next-generation machines in the system. Choosing a software package, such as LabVIEW, that supports many industry-standard protocols is critical to reducing development time for each machine on the network and integrating networked devices with the key servers.

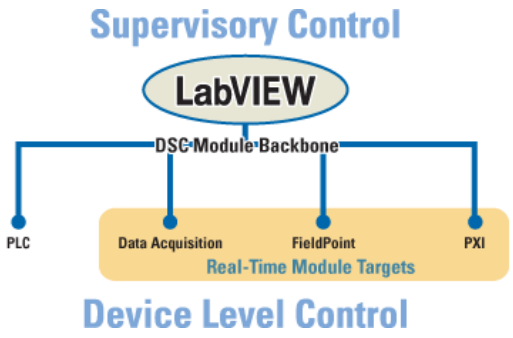

Figure 1. Diagram of a Distributed System

Networking

As a data acquisition system grows, you can store data on multiple computers and monitor it centrally. Or you can store it on one central server. The most difficult challenge is to communicate with live data. To implement this easily, you must integrate the software tools with the operating system's native networking technology as transparently as possible. Networking tools should be designed to maximize throughput and should be stable and reliable if there are disruptions in the network. OLE for Process Control (OPC) is an industry-standard interface through which software and hardware can communicate irrespective of the manufacturer. LabVIEW provides support for OPC through the LabVIEW Datalogging and Supervisory Control (DSC) Module. But OPC may not operate at the level of performance you need for higher speeds. For this requirement, LabVIEW DSC also includes a built-in networking protocol for optimal throughput.

Data Management

The key servers should also be able to log data. Logging is the process of acquiring data and storing it in a file or a database. The more complicated an application is, the more critical it becomes to have tools that do this well. You can easily store small amounts of data in text or spreadsheet files; however, larger amounts of data benefit from more sophisticated data storage formats.

You can choose from many different ways to store large data sets -- each has its own advantages and disadvantages. The two primary types of databases used are relational and streaming databases. Relational databases are more traditional databases often used in business applications. Although extremely flexible, they are not optimized for disk space and fast throughput. Streaming databases, on the other hand, are designed for quickly storing large amounts of data to disk. However, you cannot define tables or other structures and, therefore, you lose some

flexibility and search capabilities. When building a database, some of the most common challenges are designing an easy-to-use, scalable file structure and data structure. Without the right tools, you can spend hours designing and modifying your data structures. LabVIEW DSC features a streaming database designed to meet the needs of measurement and control applications. With built-in data hierarchy and data structure definition, the built-in database provides superior space utilization and data throughput.

Data Visualization
Logging data to disk is not the only challenge you face when designing distributed systems. Engineers and operators also need visibility into the data, either as the signals are acquired or after the acquisition is complete. You can view data in two different ways.

First, you can view it from the server during the acquisition. This is also known as live data. Engineers need to look at live data to monitor the status of a running system.

Viewing live data from a single machine is a relatively straightforward operation. To monitor live data, you can create a user interface, such as a LabVIEW front panel, and view it on a local monitor or embed it in a Web page using Web Services and LabVIEW Web UI Builder.

Viewing live data from multiple machines is more challenging. Each machine viewing data must be running a client, whose job is to query for the required data when asked. It is important to understand what type of server the application is going to be connecting to and to choose a software package that helps make the communication easy. One of the most common servers is OPC. Finding a software package with a built-in OPC server/client can save valuable development time.

The second way to view data is after it has been stored in a file, also known as historical visualization. Use this data for post-acquisition analysis and presentation. The challenges associated with this are similar to data logging. Without the right tools, you must know the file and data structures before analyzing and viewing the data. Choosing a software development tool that does this can, again, save time.

LabVIEW DSC builds on LabVIEW analysis and presentation functions to include built-in OPC connectivity and visualization tools to help you easily view live and historical data from multiple machines.

Managing Alarms and Events
When acquiring large amounts of data or data over long periods of time, engineers are typically less interested in the value of each data point and more interested in significant changes in the data values. You can monitor these changes using alarms and events. It is important to preserve the history of these alarms and events so they can be analyzed at a later date. The key features you need to support alarms include the ability to generate an alarm, store it along with the associated data, and recall the alarm and all of the relevant information after the acquisition. Relevant information may include when the alarm was triggered, who acknowledged it, and at what time it was acknowledged. With LabVIEW DSC, you can configure and monitor alarms using the HMI wizard, allowing them to focus on what causes the alarms rather than programming an alarm infrastructure yourself.

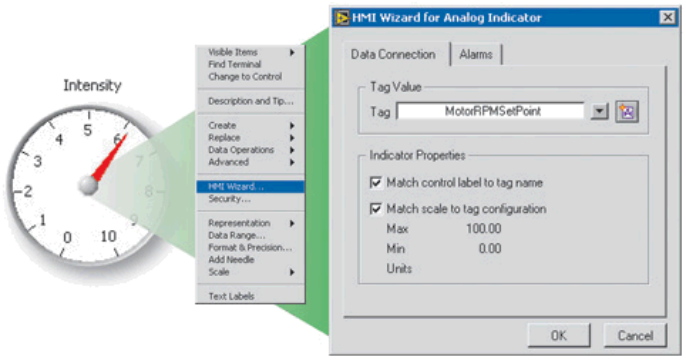

Figure 2. The HMI Wizard encapsulates configuration data and graphical programming.

Security
Handling sensitive data often raises security questions. Who should have access to the data and to which parts? Should everyone be able to modify the file or database? Probably not. Developing security code can be time-consuming and cumbersome. By defining the system needs up front, you can choose tools to help you do this. With LabVIEW DSC, you can define user profiles that limit access to different specific user interface controls on the application as well as to different and sensitive sections of the data.

Integration

Integrating components into an existing or new backbone is often the most difficult and time-consuming part of building a system. For the integration to go as smoothly as possible, it is important to gather the requirements at the beginning of the process and choose software tools that make this process easier. Looking for tools that are open and flexible is crucial to making this easier. Open software tools use industry protocols, such as OPC and TCP/IP, and work with other vendors' instruments to make integration easier for the end user. Flexible software tools empower engineers with the capability to easily add different components to the backbone of the system.

LabVIEW was designed from the beginning to work with third-party hardware. It has the underlying architecture of a traditional programming language, so you can properly work with physical I/O, real-time constraints, and hardware configuration. LabVIEW also features a large user community to share ideas and resources.

Defining the Nodes

As you can see, there are a lot of tasks that the key servers need to handle. And the more they are asked to do, the larger the processors need to be and the more efficient the code needs to be. Otherwise, the computer can easily become overburdened by these different applications and tasks. This is where the concept of distributed computing comes in. By offloading some of the tasks such as data acquisition, analysis, and control to the node level, you can more efficiently use your available resources.

At the node level of the system is the hardware with specific tasks, such as a conveyor belt controller or a safety shutdown monitor. With certain hardware components, intelligence can be incorporated at the I/O level. The control routine executing at each node can have a faster response time because the networking dependencies are eliminated. In addition, offloading the node processing from the key servers reduces the number of tasks they need to run, which frees them up to monitor a larger system. The gains in processing efficiency are most evident when the node level control system integrates and analyzes varied I/O such as motion control, image acquisition, analog data acquisition, and serial data transfers.

Real-Time Operating Systems
Many node-level control systems are implemented with dedicated controllers using technology such as real-time operating systems. This platform is ideal for the node control because the real-time operating system offers deterministic performance, high reliability, and stand-alone operation. Determinism is the quality of having a task always execute at a certain time or within a certain amount of time. The advantage of this is a high level of precision in the control loop. Real-time operating systems are more reliable than other operating systems because they are dedicated to only one application or process at a time. With fewer tasks to distract the operating system, the real-time system experiences more robust operation. Finally these dedicated targets can operate as stand-alone or embedded systems. This offers the advantage of not needing an operator to interact with the process at all times.

There are several different real-time operating systems available on the market today. When deciding which one to use, you should consider how you will develop the code to run on that operating

system. One option is to use a traditional text-based language. This can involve a steep learning curve and special skills for optimizing control code. Another option is to use LabVIEW and the LabVIEW Real-Time Module.

The LabVIEW Real-Time Module extends the LabVIEW development environment with the ability to deploy and execute applications to a dedicated target running a real-time operating system. The built-in LabVIEW tools for measurement and control systems deliver software and hardware functionality specific to real-time applications, such as PID control functions and advanced timing and synchronization capabilities. With a unique approach to real-time development, LabVIEW delivers a highly productive platform for engineers to create deterministic and embedded systems.

Incorporating Analysis Into Monitoring and Control

To implement control at the I/O level, you must be able to embed analysis functions in the I/O code. Embedding these functions in the control code allows users to extract valuable information from data, make decisions on the process, and obtain results. Unfortunately, combining analysis with data acquisition and data presentation is not always a straightforward process. Application software packages typically address one component of the application but seldom address all aspects. LabVIEW was designed to address the requirements for a start-to-finish, fully integrated solution so you can seamlessly integrate all phases of your application in a single environment.

LabVIEW provides powerful analysis libraries, routines, and algorithms that range from basic math to advanced signal processing, which can be seamlessly integrated with all other functions in LabVIEW. These functions, in conjunction with powerful data visualization capabilities, make LabVIEW the ideal tool for any application.

You can incorporate analysis into your applications and programs in different ways. You need to make certain considerations when determining the way in which analysis should be performed. The following paragraphs describe inline analysis, offline analysis, and additional analysis tools.

Inline Analysis
Inline analysis implies that the data is analyzed within the same application where it is acquired, such as at the control I/O. Decisions have to be made immediately and the results have direct consequences on the process, typically through parameters that need to be changed or actions that must be performed. When dealing with inline analysis, it is important to consider the amount of data because these routines can easily become computationally intensive and have an adverse effect on the performance of the application. LabVIEW offers analysis and mathematical routines that natively work together with data acquisition functions and display capabilities, so you can easily build them into any application.

When dealing with control processes where high-speed, deterministic, point-by-point data acquisition is present, a specific type of inline analysis is needed: point-by-point. Any time resources are dedicated to real-time data acquisition, point-by-point analysis becomes a necessity as acquisition rates and control loops are increased by orders of magnitude. The point-by-point approach simplifies the design, implementation, and testing process because the flow of the application closely matches the natural flow of the real-world processes that the application is monitoring and controlling. Point-by-point analysis is also streamlined and stable because it directly ties into the acquisition and analysis process.

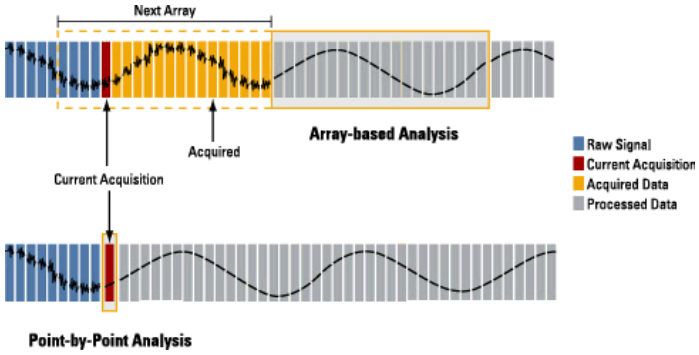

Figure 4. Point-by-Point Analysis

By adding these powerful algorithms and routines into your applications, you can eliminate the guess work and create intelligent processes that can analyze results during run time, improving efficiency and iteratively correlating input variables to experiment or process performance. This type of analysis allows the acquisition and analysis process to move closer to the point of control in embedded controllers and dedicated CPUs.

Offline Analysis
Offline applications do not typically feature the demand for results to be obtained in a real-time fashion to make decisions on the process. Offline analysis applications require only that sufficient computational resources are available. The main intent of such applications is to identify the cause and effect of variables affecting a process by correlating multiple data sets. These applications generally require importing data from custom binary or ASCII files and commercial databases such as the LabVIEW DSC Module historical database, Oracle, Access, and other SQL/ODBC-enabled databases.

Once the data is imported into LabVIEW, users perform several of hundreds of available analysis routines, manipulate the data, and arrange it in a specific format for reporting purposes. LabVIEW provides functions to access any type of file format and database, SQL functionality in order for engineers to communicate with existing business databases through the LabVIEW Database Connectivity Toolkit, and support for the latest data-sharing technologies such as XML, Web-enabled data presentation, and ActiveX.

Additional Analysis Tools
In addition to the built-in measurement analysis libraries, engineers rely on add-on toolkits and modules to reduce development time for specialized application needs. By incorporating toolkit components into custom applications, you reduce the need for the particular expertise commonly associated with developing more vertical applications such as PID control, vibration measurements, and image processing.

PID Control
The LabVIEW PID and Fuzzy Logic Toolkit adds sophisticated control algorithms to control applications. By combining the PID and fuzzy logic control functions in this toolkit with the measurement analysis functions in LabVIEW, you can quickly develop programs for automated control. In addition, by integrating these control tools with NI data acquisition hardware, you can create powerful, robust control systems.

Sound and Vibration Analysis
The NI Sound and Vibration Measurement Suite provides the basic tools required to quickly create custom applications for sound and vibration analysis. Use it to extend LabVIEW with functions and visualization tools for handling calibration, order analysis, frequency analysis, transient analysis, sound-level measurements, and fractional-octave analysis. The Sound and Vibration Measurement Suite has also been optimized for running on embedded real-time controllers to allow for faster processing and larger data sets.

Vision/Image Processing
NI Vision Acquisition Software adds high-level machine vision and image processing to LabVIEW. You can use it in machines as well as factory automation operations that require extremely reliable, high-speed vision systems.

Post Acquisition Analysis and Report Generation

Now that you have thought about the system design, the next question is how will you use the data? Will you need to pull data from a central database and create a report to share with colleagues, managers, or customers? If this is the case, then it is important to choose a software package that is tightly integrated with the rest of the system. One example is NI DIAdem, which you can use to import data from many different sources or databases, interactively analyze large sets of data, and easily create professional reports.

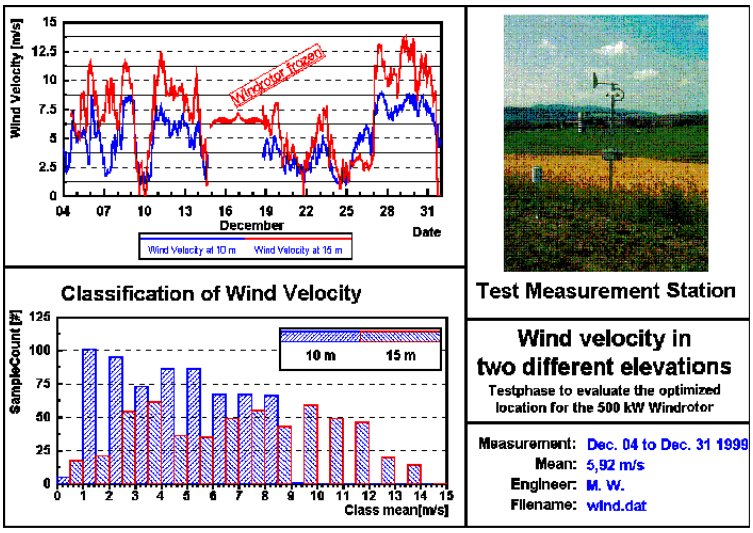

Figure 5. DIAdem Report

With DIAdem, you can easily retrieve data acquired by your LabVIEW application. Use a wizard to point to a data file for DIAdem to retrieve and determine the best way to visualize and present the content of the file. You also can implement this programmatically from within LabVIEW. Retrieving data from several files at once is simple, and you can quickly compare results and data sets obtained by different applications or different runs of the same application.

Unlike standard spreadsheet programs, large data sets are not an issue for DIAdem. In fact, DIAdem is capable of dealing with more than a billion data points. You can not only easily pull data from the LabVIEW DSC Module database but also retrieve data from SQL/ODBC and Microsoft ADO-enabled databases. DIAdem has integrated mechanisms with which data can easily be saved to and retrieved from these databases.

Once you have acquired and analyzed the data and obtained the desired results, you need to share the results with colleagues and customers. DIAdem makes this task simple. With DIAdem, you can interactively design reusable and fully customizable report templates. Any of the objects in the report can show data from the acquired or calculated channels. When you are satisfied with the report layout, you can store it for reuse and/or send it to a printer.

System Maintenance and Upgrades

Lastly, designing a system that is easy to scale and maintain from the beginning is extremely important. Considerations include minimizing the differences between the current system and the new one as much as possible, as well as automating as many steps as possible. Collaborating with the different groups responsible for maintenance at the beginning of the project and learning about their processes and expectations can help you design a system that minimizes maintenance steps. One way to do this is to build user interfaces for maintenance procedures and then automate them whenever possible. Although working with a cross-functional team may reveal some conflicting requirements, dealing with as many of these up front as possible results in a better system.

Conclusion

Distributed monitoring and control systems help you take advantage of the processing power of all available resources to more efficiently use the entire system. By choosing open and flexible tools for the backbone of the system, you can more easily integrate products from many different vendors. Choosing a real-time operating system and designing an embedded application with analysis help offload some of the computational requirements from the key servers as well as provide faster, more sophisticated control algorithms at the node level. LabVIEW with the LabVIEW DSC and LabVIEW Real-Time Modules gives you the tools you need to succeed when designing, building, and integrating distributed systems.