Overview of Data Acquisition at the SNS

Richard Riedel
Oak Ridge National Laboratory

1.0 Introduction

When the Spallation Neutron Source (SNS) becomes operational it is expected to generate an unprecedented neutron flux. Not only is the neutron source brighter than extant pulsed sources, the neutron instruments will be among the most efficient having larger angular detector coverage and long neutron guide runs. The high data rates, up to 60million events/second, places extreme requirements upon the data acquisition system that must also handle a wide variety of hardware components in a seamless and easy to understand fashion. In this paper we provide an overview of the architecture of the data acquisition system that will be in use on neutron scattering instrumentation at the SNS.

2.0 Black Box Viewpoint

From a black box viewpoint, each instrument's data acquisition system[1] can be considered a control system whose primary output is neutron scattering data and whose primary inputs are control commands describing either an experimental "run" or a simple change in a controllable variable. (I.e. detector position, sample environment parameters etc.). The input and output data sets are very different. In the case of neutron scattering data, the data sets and data rates can be very large of the order of gigabyte files, and of the order of megabytes/second. In the case of control commands, the data sets are small and infrequent. Each of these data interfaces has different security needs. While data files can be locked in a read only mode, control of the data acquisition system requires careful control of what is being requested and authentication of the requestor. Additionally it is desirable to have an interface that allows the user to interface specialized equipment he may need for an experiment with the data acquisition system. Figure one shows a diagram of these interfaces.

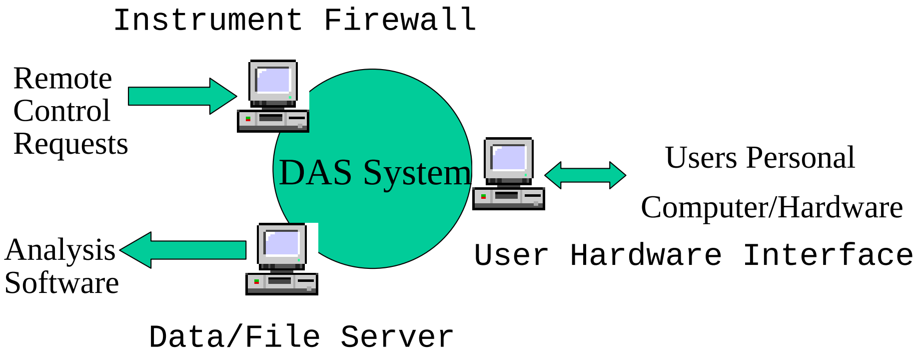

Figure one showing a block diagram of the SNS data acquistion system interfaces. The primary interfaces are the instrument firewall and the data/file server which together allow intelligent control of the DAQ system. A third interface, the user hardware interface allows the connection of specialized user equipment.

---

[1] Each instrument has its own independent data acquisition system.

Two of the external interfaces, the firewall and the data/file server, are integral to remote viewing and control of the instrument a topic of much interest to users and operators. The firewall which provides a gateway to remote services offered by the instrument's data acquisition (DAQ) system also allows each instrument to operate on a private network whose IP traffic loading and computer connections can be tightly controlled. Low bandwidth information such as temperature, vacuum, detector position, etc. is made available to the external user via this interface and is viewable with a simple web page application. Changes to single experimental parameters can also be made with this web page. More complex control of an instrument is done by downloading Python script files. Applications that can control the instrument are denied until the requestor of the operation is authenticated. Authentication is also required before certain standard ports, such as remote desktop, are made available. Also, script files are validated for syntax before execution. The instrument scientist can disable remote control at anytime. The firewall also provides NAT translation redirecting remote applications to the appropriate DAS server.

The data/file server allows external access to semi-real time neutron scattering data[2]. The native format of neutron scattering and control data output by the DAQ system is known as pre-NEXUS and consists of a set of binary and XML files. These files are transformed by a process running on the data/file server into a NEXUS5 file which can be accessed by the user. The data/file server has a RAID disk system with a capacity of about 1 terabyte and is used for short term storage of scattering data. A file backup process running on the data/file server is coordinated with the work load of the data/file server ensuring that analysis programs requiring access to semi-real time data have priority. The data/file server and the firewall interface together create a method for external intelligent control of the DAQ system. By analyzing the scattering data a program can then create a new set of control variables, form the changes into a Python script and download the file to the instrument via the firewall.

The user hardware interface is the third external interface of interest. There are occasions when the user will need to bring in specialized equipment for a particular experiment. Although there is a well defined communications protocol for equipment control, the user is not unlikely to be familiar with the requirement nor are they likely to expend effort into learning the protocols. The user computer allows automated control of specialized user equipment by two different means. For custom equipment that make a hardware control port (such as a serial port) available for the reading or writing of information it is a simple matter, taking a few hours to modify a standard "data pass through" application creating a connection between the users equipment and the DAQ system. A second method needs no physical connection between the DAQ system and the custom equipment, but does require an XML configuration file to be placed on the user hardware interface computer. This "configuration" file lets the DAQ system know that there is specialized equipment attached and identifies Python script names that can be used in a script file. For example suppose the user has a variable he wants to change at a certain time during a sequence of runs. A single statement placed in the script file will cause a dialog to pop up when the statement is executed directing the user to change

---

[2] The semi-real time data files are updated every 5 to 30 seconds depending upon the size of the data set.

the variable.[3] The advantage of this is that the change is automatically recorded by the data acquisition system.

3.0 Inside the Data Acquisition System

As figure two shows the data acquisition consists of a number of distinct subsystems. The subsystems are denoted as real time, soft real time and slow control subsystems. The three different subsystems are designed to handle data with different guaranteed delivery times. The real time data system consists of the detector electronics and any other electronics that must have timing of the order of 100nsec. The soft real time data system consists of a cluster of high end P.C.s and fiber optical communication cards to handle data delivery with latency of the order of 100usec. The slow control subsystem has a control latency of the order of 100milseconds or less. The division into these different timing requirements is dictated by the pulse rate at the SNS neutron scattering facility which is 60Hz or 16.7msec. The slow control and soft real time subsystems consist entirely of networked P.C.s operating over multiple private IP subnets., while the real time subsystem is primarily custom electronics with custom high speed data networks.

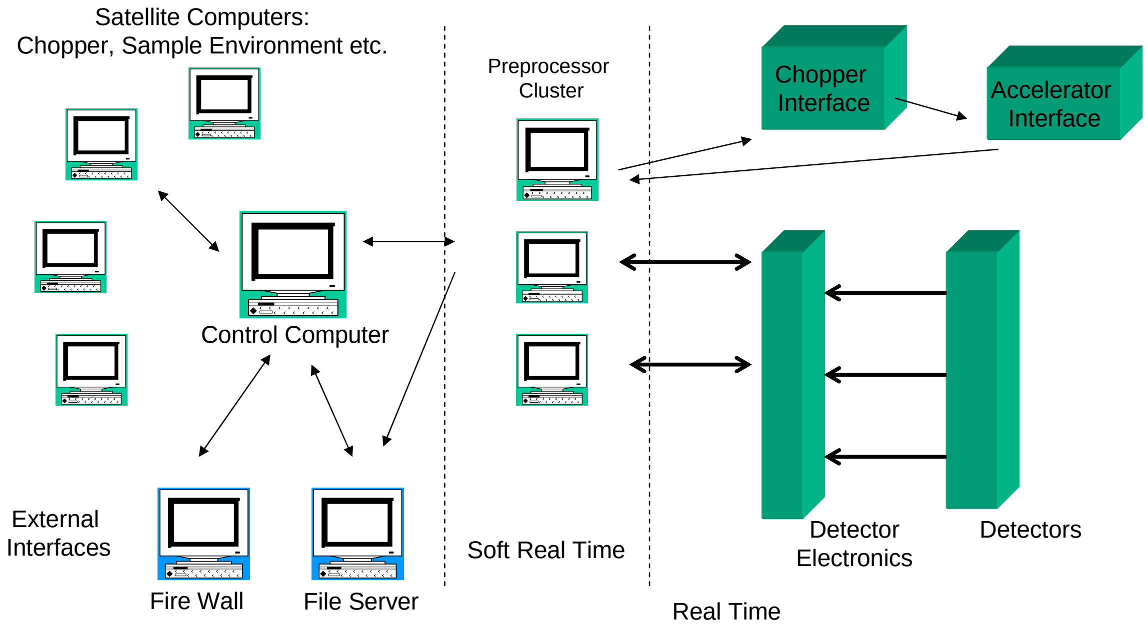

Figure two showing a block diagram of the SNS data Acquistion System. The detector electronics and timing module are part of the real time data acquisition sub-system, while the preprocessor and preprocessor communication links are part of the soft real time data acquisition subsystem. The control computer, ancillary controls and file server are part of the slow controls data acquisition subsystem.

The primary user interface applications reside on the control computer. The control computer is responsible for the control of an experimental run, issuing commands to the various other computer of the slow control system, along with commands to the

---

[3] It is the information in the XML configuration file that makes this operation possible.

various other subsystems. Control can be accomplished via Python script files or by direct input into a graphical user interface (GUI). While the control computer is responsible for the coordination of experimental runs, actual control of hardware is handled by satellite computers, or in the case of the detector system, the preprocessor cluster. The control computer and the various other P.C.s of the data acquisition system form a client-server relationship with the client applications running on the control computer. For computers on the slow control network control information is passed between computers using the National Instruments datasockets protocol. In the datasockets protocol control variables are associated with a URL name. For example a temperature reading on a Lakeshore controller may be associated with the URL "dstp://sampleenv/temperatureA". The URL itself does not necessarily provide enough information to be useful in identifying what is being measured or controlled. In the example above temperatureA could be the sample temperature, or some other temperature associated with sample environment. To associate the control variable with a meaningful "friendly name" a set of configuration files residing on the satellite computer provide meta-data associated with the control variable. The advantage of using a level of indirection is that a single GUI or application can be associated with different abstract variables simple by editing an XML file. For example a simple motor control program could be written. That same program could be associated with detector position, slit width, or even sample position by editing the associated XML configuration file. GUI programs written for the control program are written with this flexibility in mind, with a variables friendly name appearing as a label in the GUI.

While the control computer is the user's primary interface to the DAQ system the satellite computers are the expert operator's primary interface to the instrument's hardware. The typical suite of satellite computers in the slow control subsystem are the sample environment, chopper, and ancillary computer. As their names imply, the sample environment and chopper satellite computers control the sample environment and chopper hardware while the ancillary computer is responsible for the movement of motors. If specialized user equipment is needed, the user hardware interface described in the previous section would be present. Depending upon the complexity of hardware controlled, additional satellite computers may be warranted.

The control computer is also a client of the preprocessor cluster which is part of the soft real time system. The preprocessor is the interface to the detector electronics providing access to detector control parameters and to neutron scattering data. The preprocessor consists of one or more P.C.s communicating with each other and the control computer via commercial high end optical Gigabit Ethernet cards. The communication between the computers occurs on a private network separate from the slow controls network. This separation allows tight control of the sequencing of Ethernet traffic on the network. The number of P.C.s in the preprocessor cluster can vary between one to eight or more depending upon the expected data rate from the detector electronics. Certain data transformation such as position and time to d-spacing or histogramming of neutron data can be done in soft real time by the preprocessor. Whenever more than one computer belongs to the preprocessor cluster, the control computer communicates with a cluster master. The cluster master coordinating information requests by the control computer and transfer of neutron scattering data to the data/file server.

Not only must the preprocessor system histogram neutron scattering data in soft real time, but because the SNS DAQ system allows operation in event mode where each individual neutron event is time-stamped and associated with a pulse ID the soft real time system must deliver the pulse ID to the preprocessor in time to be associated with data delivered by the detector electronics. Because the pulse ID and other important pulse information are delivered by the accelerator system, a method is required to broadcast this information to the preprocessor. The system that accomplishes this information transfer in under a millisecond consists of an embedded timing circuit (ETC) card developed by accelerator systems, a P.C., an ETC driver developed by instrument systems, and a high thread priority system application also written by instrument systems. This combination of hardware and software transmits in soft real time the pulse I.D. along with a small amount of other accelerator data to the preprocessor unit via a broadcast UDP protocol on a private network who's only traffic is the transfer of accelerator data. The performance of the UDP interface is shown in figure three which shows how tight control of Ethernet traffic an use of preemptive interrupts can be combined to achieve soft real time delivery of information.

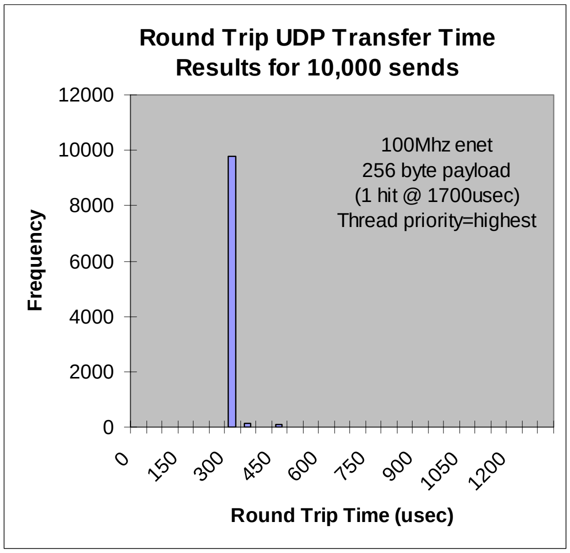

Figure three showing the results of a benchmark study on the speed of transfer of a small data block across a dedicated private network using a UDP communications protocol. The listed times are total round trip times for the delivery of data and receipt of a response packet.

The last of the DAQ subsystems is the real time subsystem that consists of the detector electronics and associated real time signal distribution electronics. The detector electronics is responsible for signal conditioning of the analog signal from the detector, analog to digital conversion, and assigning a unique position and time stamp for each detected neutron. The detector electronics also contains the electronics required for transmitting the neutron event data to the preprocessor computer cluster. This transmission is accomplished via a bi-directional high-speed optical data link.

The high-speed data link is a fiber optic based data link designed by the SNS data acquisition group. Data bytes representing detected neutron events are formatted in the detector electronics, serialized and sent through the optical link in data frames. The data link also provides for data to be sent to the detector electronics from the preprocessor. The optical interface uses a custom protocol on top of the first two fiber channel layers, and is capable of transferring data at a rate of 200Mbytes/sec.

The detector electronics consists of the electronics and firmware code needed to record the time and location of scattered neutrons. In addition it must be capable of determining via computer control the correct values for any threshold or offset settings required for calibration of the detector system.[4] By using a set of networked connected P.Cs (preprocessor cluster)., it is a simple matter to expand the DAQ system to handle a variety of detector systems. For example a low rate detector system (500Kevent/sec) may communicate with a preprocessor cluster having only a single P.C. while a high rate detector system (60Mevent/sec) may require a preprocessor cluster of ten P.C.s. Figure four below shows a hypothetical connection diagram of the detector electronics and the preprocessor cluster.

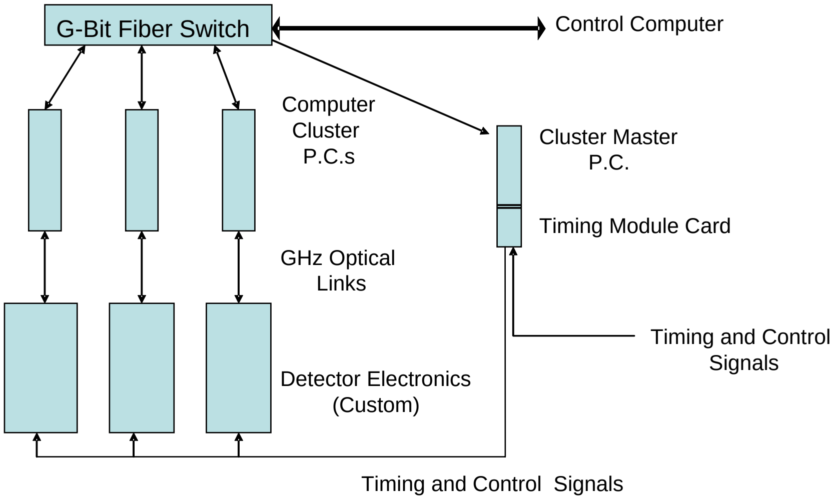

Figure four showing a hypothetical detector electronics preprocessor cluster system. The detector electronics is sectioned into one or more subsections depending upon the expected event rate. Each detector subsection is linked via a high speed 2.1GHz optical link to a computer of the preprocessor cluster. If there is more than one detector electronics subsection, and cluster master is required which maintains the coordination of information between the control computer and the detector electronics. The preprocessor cluster communicates with each other and the control computer via a Gigabit fiber optic private network.

While each detector system must communicate to an optical communications card, how this is best implemented may be different for different detector systems. For example some detector systems may be able to use components already designed by SNS. For other detector systems it may be necessary to design a custom interface or daughter board to interface the detector electronics to the optical communications card. This method was done for a set of XY gas detectors built by Brookhaven National Labs. A third possibility is to use a P.C. with an optical communication card with custom software

---

[4] Calibration is required to be automated. Currently there are no planned detector systems requiring manual adjustment of potentiometers.

to communicate to the preprocessor system. In all cases the detector electronics is required to perform a standard set of operations and must transmit data in a defined format regardless of the detector system.

Also part of the real time subsystem is the real time timing distribution network which carries real time digital signals between the timing module, the chopper optical distribution board. Figure five shows the various parts of the timing loop.

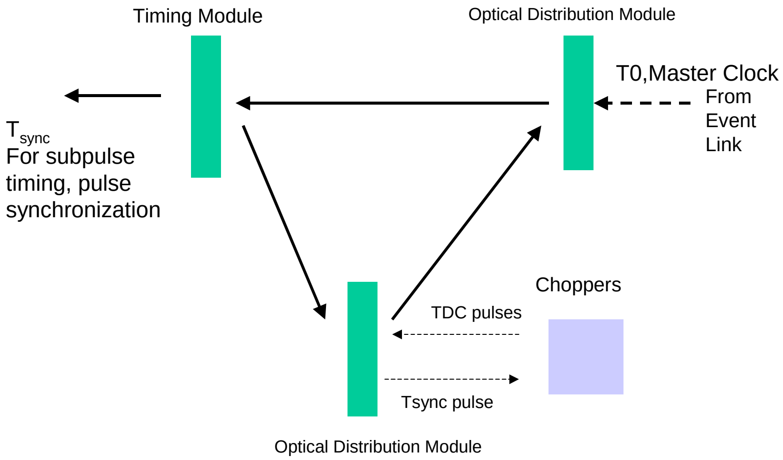

Figure five showing a block diagram of the SNS real time digital distribution scheme. The system forms a signal loop. This is required to insure that the optical circuitry maintains phase lock.

The timing module, a custom PCI bus card designed by the SNS DAS group, resides in the preprocessor cluster master and provides timing signals used by the detector electronics and other information that is needed for event mode collection of data such as chopper phase errors. Timing signals are transmitted via fiber optical links. These fiber optic links include connections to 1) the detector electronics, 2) the chopper timing system, and 3) the accelerator timing system. Internal to the timing module are various registers that control the configuration and operation of the module. These registers are accessed by various software applications using a custom windows driver. The optical distribution board converts up to 16 TTL or LVTTL signals to and from an optical serial stream and provides the required method of transferring real time digital signals for the DAQ system.

The timing module and distribution cards form an optical loop for transmission of digital signals with 5nsec of jitter. Some of the more important signals are the cycle start or extraction signal from the accelerator which defines the point in time of proton production. Also transmitted are the chopper top dead center pulses which allow one to determine the phasing of all choppers on a pulse by pulse basis. The distribution system allows other equipment requiring phasing to the accelerator to be added to the instrument. This is done by using a spare optical link or by using a set of spare Lemo connectors on the timing module. In this way novel ideas such as synchronizing sample stress with the production of neutron pulses can be realized.

4.0 Scripting Language and Software Design

As the last topic of this overview, we describe some aspects of software design and how it relates to the scripting language used by the DAQ system. As we have seen in the discussion above, the distributed control networks of the DAQ system form a number of client server relationships. In each case, the sever supplies data and meta-data rather than an application. This client server relationship allows the software developer to develop user interface applications for different classes of users. Because the control computer is the primary user interface to the DAQ system it is important to make these applications easy to understand, containing a minimum of obscure variable names. By using friendly variable names such as “sampletemperature”, “neutronwavelength” in both the control application and scripting language one gives the user a greater understanding of what is occurring during an experiment. While a simple application may be appropriate for the general user, it may not be appropriate for an expert operator who must diagnose mal-functioning hardware or set up various control loop parameters such as PID. Because the server application is a separate process running on a separate computer[5], one can tailor its look and feel to meet the needs of the expert operator. This separation of applications into two classes, each meeting a specific users needs, is used extensively in the DAQ system. Using a set of standard C++ classes we can now write applications for the control computer in about ½ to 1hour which can be fairly sophisticated in the way data is displayed. The client server model is also of great utility for interfacing with detector systems. By communicating with an application on the preprocessor cluster rather than with the detector system directly the client application can be design to be largely the same for all detector systems with the server application handling details that are needed for the different detector types

The interactions between software processes active on the control computer can be especially complex with up to six different multithreaded DAQ processes running at the same time. In addition to GUI applications the SNS DAQ system allows the use of Python has a scripting language using the standard Python interpreter. Inter-process communication is accomplished with the use of memory maps and event notification. Each process on the control computer is responsible for creating a Python file that is executed when Python is launched or if the system’s configuration changes. These files contain Python code to create instances of standard control classes where the instance names are the friendly names defined in the configuration files on the server computers. All processes, including Python, create a listening thread to handle notifications from other active processes. The use of a separate listening thread allows the Python script to respond to system events such has “pause” or “stop” in a controlled manner.

5.0 Conclusion

We have given an overview of the various hardware and software designs for use with the SNS data acquisition system. This system which has been under development for four years is now in the early stages of deployment. A set of software modules has been written and tested and will form the basis for specialized needs of the individual neutron scattering instruments. We feel confident that this software along with a number of state of the art hardware components will create a robust and user friendly environment for both the casual and expert user.

---

[5] Depending upon the loading of a computer it is possible to have both the client and server applications active on a single computer.