A Proposal for Replacement of the PT-VME940 in the SDSS2 DAQ

Kurt Biery, Don Holmgren, Ron Rechenmacher, Margaret Votava

1 Introduction

As part of the 2005 upgrade of the SDSS data acquisition system, the existing PT-VME link between the online analysis computer (OAC) and the interface control computers (ICCs) will be replaced with TCP/IP messages.

This document describes the design and some of the implementation details of this new TCP/IP link.

2 Architectural Overview

In the existing system, the transfer of data between the OAC and the ICCs is handled by a PT-VME940 card in each VME crate. This card permits memory to be mapped between the UNIX system and the VxWorks systems across the VMEbus in the ICC crates. This memory map allows OAC applications to read and write data to individual ICCs without any intermediate processing.

A TCP/IP based communications channel will require an intermediate process, however, to manager the transfer of the data. We are proposing that each ICC have a simple server process that listens for read and write requests from the OAC, processes the requests as needed, and sends status information and data back to the OAC. On the OAC side, the existing read and write routines will need to be modified to send TCP messages to ICCs rather than simply reading or writing the data to the PT-VME address space.

3 Message Protocol

When the OAC sends a read request to an ICC in the TCP-based system, the ICC location and data length need to be sent in the request. We propose to send this information in a header block that will also include the type of the request and other useful information. For read requests, the only data in the message will be the header block. A reply to a read request will contain a reply header along with the data that was read from the ICC memory.

For a write request, the message from the OAC to an ICC will contain the request header and the data that will be written to the ICC memory. The reply to a write request will only contain a reply header.

We propose the following request and reply headers:

Message type	Field name	Field size	Description
Request	protocolVersion	16 bits	Defines the protocol version for each message. Useful if protocol needs to change at some point.
	requestType	16 bits	One of “read”, “write”, or “shadow-write”.
	continuationFlag	16 bits	Allows for grouping individual requests together (not currently used).
	userData	16 bits	Message-specific data.
	requestAddress	32 bits	ICC relative address for read or write.
	requestLength	32 bits	Length of data to be read or written.
Reply	protocolVersion	16 bits	Protocol version of reply
	replyType	16 bits	One of “read”, “write”, or “shadow-write”.
	statusCode	32 bits	Indicates the success or failure of the requested operation on the ICC.
	actualAddress	32 bits	Relative address at which the data was actually read or written.
	actualLength	32 bits	The length of the actual data transfer.

Are there any issues with VxWorks that require that transfers always contain an even number of bytes (or a multiple of 4 bytes)? If so, then we may need to pad the data blocks.

The addresses contained in the requests and replies will actually be offsets computed from the PT-VME address of the ICC as seen by the OAC. Each ICC will use this offset with its shared memory location to determine the final address for the read or write.

4 Locking Issues

The existing OAC code uses semaphores to prevent operations which need (temporary) exclusive access to a particular ICC. This functionality will be maintained in the upgraded system.

In addition, the server process on each ICC will only process a single request at a time.

If we find that the use of semaphores on the OAC becomes undesirable at some point, the continuation flag in the request header could be used by the ICC server to simulate locks by focusing on a single client’s requests until that client declares that it has finished its set of tasks. We do not plan on implementing this functionality initially, however.

5 Hostname Lookups

The existing way of referencing a particular ICC using crate and board numbers will need to be supported after the conversion. For this, we will need to maintain a map between the ICC host names (which the OAC TCP code will need when making socket connections) and the crate and board numbers (which users and existing scripts will use to reference ICCs).

For this, we propose to create a map at initialization based on the ICC configuration files in $ASTROBASE_DIR/system/<systemName>.

6 ICC Access Control

To prevent unauthorized clients from communicating with the ICCs, each ICC TCP server will have a list of authorized client IP addresses. Only clients on the list will be allowed to send read and write commands. The list will be specified for each ICC in its boot script.

7 Additional Features

To allow for consistency checking, it will be possible to transfer the data over TCP in a shadow mode. In this mode, the data transfer will use both the PT-VME940 and the TCP/IP link. The PT-VME data will be used for the operation of the system and the TCP data will be compared to it to test for errors. The operation of this shadow mode will be controlled by a compile-time flag in dscConfig.h.

To support measurements of transfer rates, we would like to provide utilities to time the PT-VME and TCP transfers which can be controlled by run-time flags.