Lessons Learned in Using COTS for

Real Time High Speed Data Distribution

Bob Downing
Jim Bretz
SPARTA Incorporated
Flight Support Division
43210 Gingham Avenue Suite 6
Lancaster, California 93535
(805) 723-3148

ABSTRACT

Currently, there is a large effort being placed on the use of commercial-off-the-shelf (COTS) equipment to satisfy dedicated system requirements. This emphasis is being pursued in the quest of reducing overall system development costs. The development activity discussed in this paper consisted of determining some of the boundaries and constraints in the use of COTS equipment for high speed data distribution. This paper will present some of the lessons learned in developing a real-time high speed (greater than 1 MByte/sec) data distribution subsystem using COTS equipment based on industry accepted standards and POSIX P1003.1 operating system compliance.

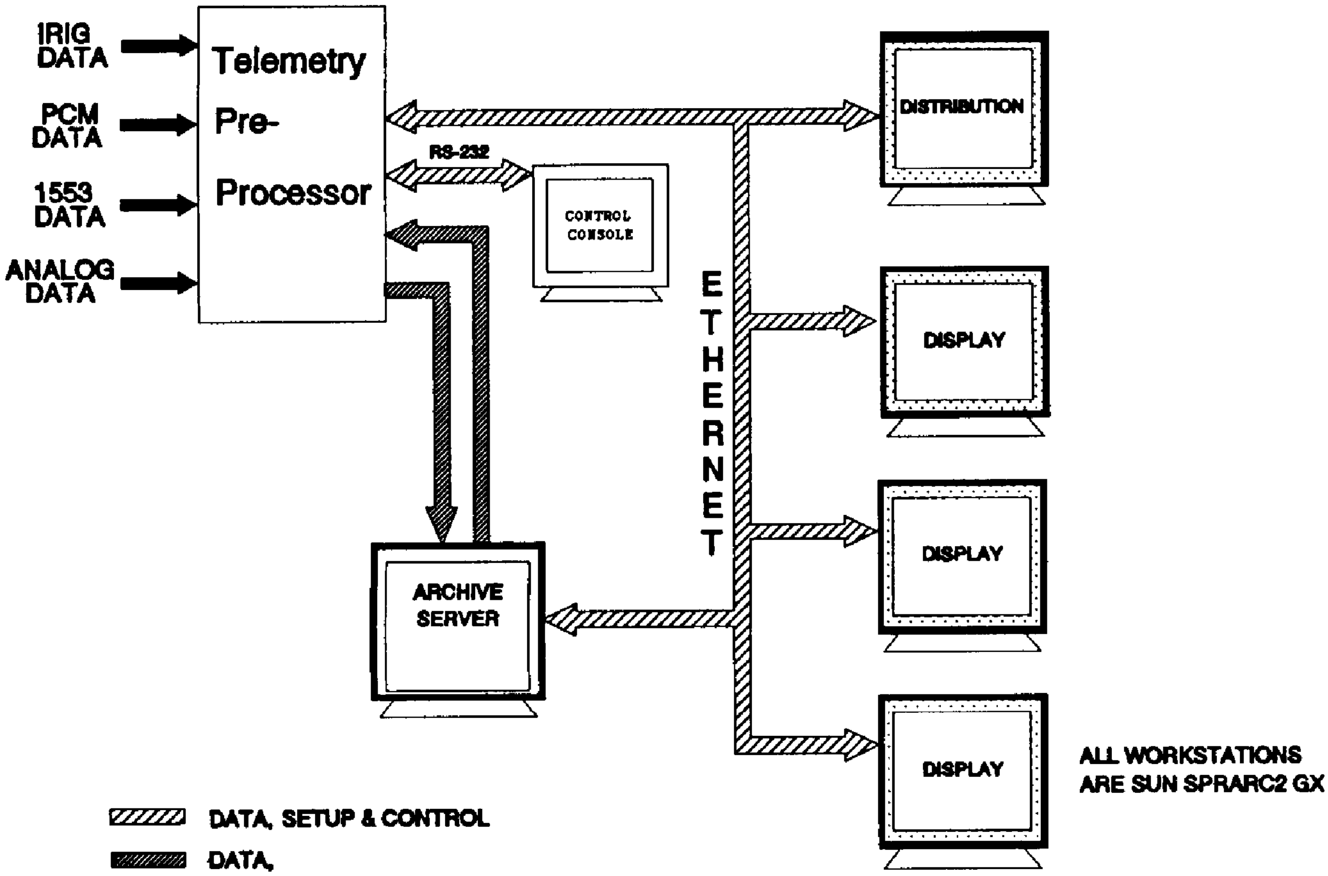
KEYWORDS

Real Time Data Distribution, COTS, POSIX

INTRODUCTION

The data distribution subsystem described in this paper is part of a larger data acquisition, processing, and display system. This system is designed to collect data from a system under test (SUT) within an anechoic chamber test environment. This system must record and monitor the response of the SUT to external stimulus and must support a wide range of interfaces currently in use and planned for the future in the testing of high bandwidth data sources. Refer to Figure 1 for a block diagram of the original system architecture.

Figure 1 - System Block Diagram



The initial data distribution design used Ethernet with TCP/IP protocol to distribute data to the Display Workstations. The individual parameters were selectively routed to the Display Workstations in Current Value Table (CVT) format for real time display of the data. This data distribution path is shared with the capability to command and control system components. The effective bandwidth of Ethernet for display data distribution was approximately 225 KBytes per second for point-to-point communication using sockets for data transmission. The effective throughput of the distribution system is a function of the number of display workstations (throughput divided by the number of workstation nodes). The effective throughput for three display workstations is less than 100 KBytes per second, which is significantly less than desired. The long-term requirement for data distribution was for 2 MBytes per second of CVT data to all workstations.

To meet this requirement a design was needed that would not disrupt operations and be transparent to the system user making maximum use of COTS equipment. This paper will address some of the lessons learned in dealing with this development. These lessons include non-deterministic performance problems of the standard workstation operating system and memory caching.

Each of the Display Workstations was a SUN SPARCstation 2, which uses the SUN SBus as its primary interface. At the time of this development, the SBus was not widely supported by Industry. To solve the problem, a distribution workstation was dedicated to the job of receiving data sequentially from the data source over a parallel DR11-W interface, then distributing that data to the display workstations via a high speed distribution medium. To solve the problem of interfacing to the Display Workstations, a conversion to a widely supported I/O bus was necessary. A Versa Module Europa (VME) solution was selected as the I/O bus, and the Solflower SBus-to-VME converter was chosen to provide the most transparent bus conversion to the SunOS, while maintaining the performance of the VME bus.

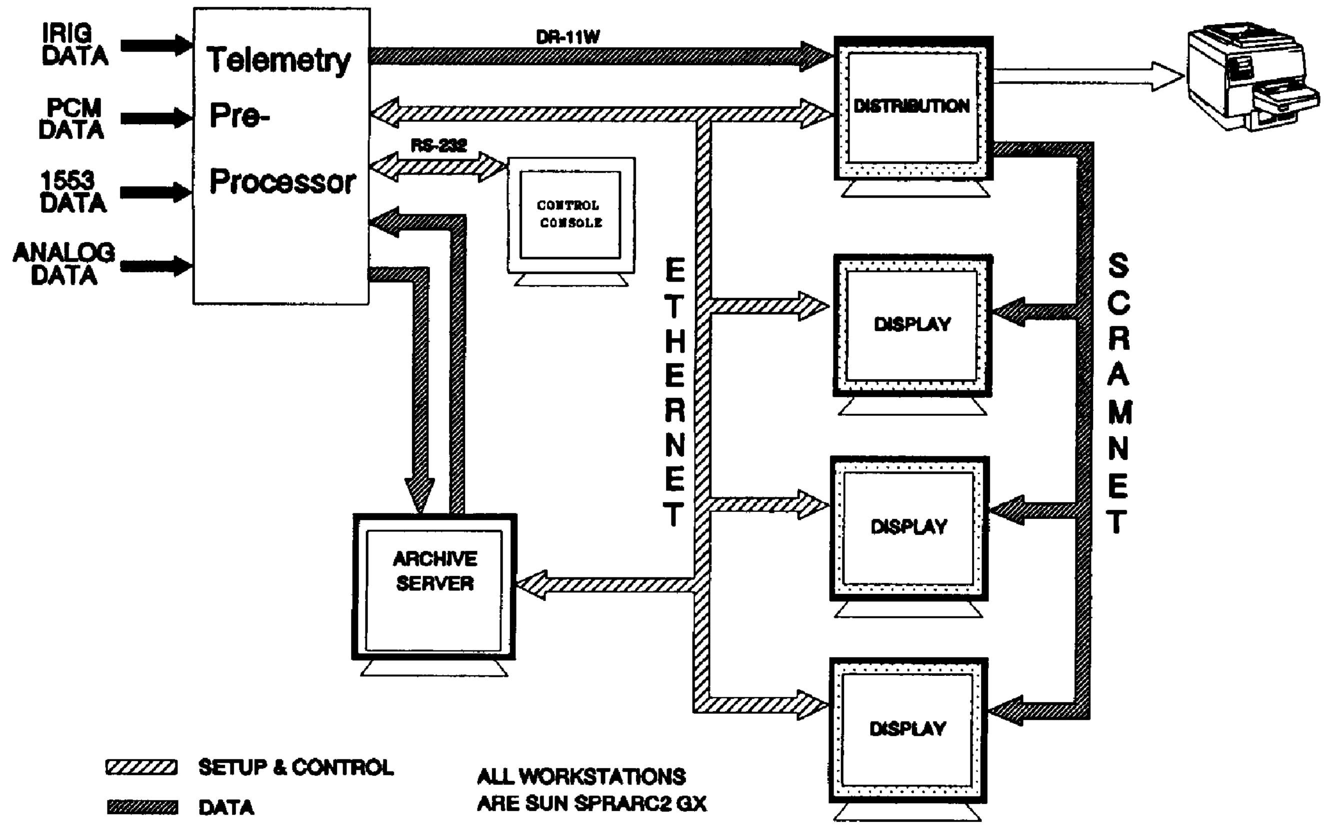
A distribution workstation was dedicated to the job of receiving data sequentially from the Telemetry Preprocessor (TPP) over a parallel DR11-W interface, then distributing that data to the display workstations via a high speed distribution medium. A shared memory system from SYSTRAN Corporation was selected as the network distribution medium. With the selected network bandwidth rated well above the requirements and the shared memory configuration ideal for the maintenance of a CVT, the VME bus conversion and shared memory network pair provided a viable solution. Each display workstation periodically transfers the selected data received from the shared memory network to the graphical display process. Software was developed to distribute data at the distribution workstation and interface shared memory to the COTS display hardware and software. Refer to figure 2 for a block diagram of the system architecture.

PROBLEM DESCRIPTIONS and RESOLUTIONS

Several problems were encountered in using the UNIX Operating System (SunOS) to handle the realtime distribution function, but the prominent problem encountered in this development was one regarding operating system memory caching. Initially, this data distribution development concept was of Direct Memory Accessing (DMA) from the TPP to the distribution workstation via DR11W. Problems occurred while attempting to address a block of memory. The SUN addressing is virtual, not the physical address as expected, adding an extra layer to the data transfer pathway. Furthermore, the DMA memory is cached upon reading. This resulted in occasional blocks of data that are not updated in real time, even when the operating system returned a word count implying that the data transfer took place for that memory location. This prevented data from being transferred from the TPP to the CVT in a real-time manner.

The lack of true DMA support by the SunOS prompted the development of an alterative method of accumulating data. This method required the data be captured in

Figure 2 - System Block Diagram



buffers and transferred to the CVT in discrete groups. This group size is dynamic; the groups get larger during higher data rate transfers and smaller during low rate transfers. This method transfers data at rates up to 2000 KBytes per second. The downfall of this approach is that the group buffer size is changing when the data rate changes. This solution reduced the overall throughput of the data transfer, but still met the 2000 KByte/second throughput requirement. The conversion to a real-time operating system would increase this rate and the buffering scheme would be removed.

OS Caching Problem - The initial development concept was of Direct Memory Addressing (DMA) from the TPP Preprocessor to the Data Distribution Workstation CVT via DR11W. Problems occurred while attempting to address a block of memory. The SUN addressing is virtual, not the actual physical address expected, adding an extra layer to the data transfer pathway. Furthermore, the DMA memory is cached upon reading. This resulted in occasional blocks of data that are not updated in real time, even when the operating system returned a word count implying that the data transfer took place for that memory location. This prevented data from being transferred from the Telemetry Preprocessor to the Current Value Table (CVT) in a real-time manner. Tests were run utilizing newer versions of the OS, modifying parameters in the OS, and implementing different algorithms which only showed minimal improvement. Continuing inquires with SUN and other technical support areas have given some insight into the problem but have not revealed a functional solution.

The lack of true DMA support by the SUN operating system prompted the development of an alterative method of accumulating data. The real time data is captured in buffers and transferred to the CVT in these buffers. The implemented buffers are dynamically sized; the buffers get larger for higher data rates and smaller for low data rates. This alterative is operating very well with tested speeds to lMB/s. The downfall of this method is that at very high speeds data is buffered in larger blocks. At low speeds, it may require several iterations for the buffer to adjust, and at very low speed , an occasional pause is detected while the CVT waits for a buffer to be filled

Non-deterministic OS - The lack of a true real-time operating system surfaced again on the display workstation. The display workstation has two basic tasks, to read the CVT from shared memory and then display that data. Reading the CVT in a deterministic fashion has failed due to process swapping by the operating system and non-deterministic interrupt handling. When a particular parameter is to be updated at 0.1 seconds, for example, the operating system may update that parameter at 0.1, 0.5, 0.55 ..., due to time sharing of the processor among the different tasks. Even when we created our own timing algorithm that would watch the system clock and signal when the parameter was due to be updated, the OS time shared the process and buffered the created signals. The new release of the SUN Operating System (Solaris 2.0) should allow timed deterministic processes, but this new release was not available at the time of development.

A work around was developed that consists of requesting the shared memory network to send an interrupt on each data parameter received provided the best resolution. This method was tested with one parameter running at approximately 400 samples/second. Using such a high frequency of interrupts caused interference with other programs. Additionally, using the shared memory network interrupts severely limits the number of different update rates. Each update rate would require a different interrupt number, of which there are few. The final design settled on using the X-Event timer in hopes that future updates to the SUN OS would improve this capability.

Displaying the data has presented another problem. The display workstation did not have enough processing power to update many parameters each with a different update rate. Tests with two display graphs, one at 200 sample/sec the other at 20 samples/sec, but at the same display update rate of 1 millisecond, were beyond the capabilities of the system. This solution for the display update requirement was not satisfied and the requirement was relaxed for the use of COTS equipment.

CONCLUSION

COTS equipment provided an off-the-shelf solution that integrated well into the distributed architecture for the data system. This effort provided a means to distribute high speed CVT data to standard workstations. The software implementation could be simplified and the data rate increased with the additional of a real time operating system. The overall philosophy of using commercial off-the-shelf was validated and provided a cost and schedule effective solution. The use of COTS to satisfy system requirements is an effective way to reduce overall system development costs. Industry standards insure flexible, easily upgradeable architectures with satisfactory performance. Choosing standards sometimes means that the implementation can be behind the performance delivered by a proprietary system. With the maturation of standards and as the rate of COTS performance increase, this gap is closing fast. In most instances, it is no longer viable to "reinvent the wheel" in order to satisfy system requirements. It is more advantageous to use interchangeable, tested, industry standard, system components that can be incrementally upgraded. This philosophy insures responsiveness to new technologies and requirements changes.

The ultimate success of the system development using COTS lies in the acceptance of tradeoffs. During the design phase, there must be tradeoffs made between the system requirements and COTS supplied capabilities at that given point in time. For instance, under a POSIX P1003.1 compliant operating system (note: most UNIX based operating systems are POSIX Pl003.1 compliant), there is no standard way to address interrupt or process priority assignments; this problem is being addressed in POSIX P1003.4 real-time extensions, which should remedy most of the problems encountered.

With the scope of work shifted from component development to component integration, problems are consequently centered around integration issues. Not all standards are equal, adherence to the standard does not insure component compatibility. In some cases, it is recommended that these discrepancies be localized and well documented, until a more robust or uniformly supported standard is available.

Some of the boundaries and constraints for design of a high speed data distribution system using COTS were discussed in this paper. The benefits of development using COTS were found to greatly outnumber the drawbacks. It was the authors' intention to educate future developers in this area as well as promote the use of COTS in overall reduction of system development costs.