UNIX and Real-Time Telemetry

Robert Querido
Loral Instrumentation
San Diego, CA

Introduction/Abstract

This paper discusses the benefits of using UNIX in a telemetry and satellite control product and some specific features implemented in UNIX-based workstations and file servers. Features discussed include real-time disk archiving and playback using UNIX and single-point-of-failure issues.

Benefits of UNIX

The two benefits of using UNIX are the extensive amount of industry software standards available and the tremendous price/performance of the UNIX hardware platforms.

Standards enhance portability and allow a developer to use large software building blocks in constructing an application. There is also a large pool of trained and highly talented engineers that know how to work with these software building blocks. Virtually all UNIX-based workstations and file servers support the following standards and defacto standard software building blocks:

- "C" Compiler
- TCP/IP Networking Software
- Network File System (NFS) for transparent file access across a network
- X Window System Graphics
- Remote Procedure Call (RPC) for interprocess communication on a heterogeneous network
- External Data Representation (XDR) for interprocess communication on a heterogeneous network
- SQL-based relational databases

The System 500 product uses, in varying degrees, all of the standards above.

Another key reason for using a UNIX platform is to leverage leading edge workstation and file server hardware performance in a commercial product.

In order to compete in the UNIX marketplace, computer manufacturers must keep up with the leading edge hardware price/performance. By using UNIX-based workstations and file servers, a commercial product rides the technology wave. The price/performance of the new DECstation 5000 is a good case in point.

The DECstation 5000 is a 24 MIP CPU (24 times faster than a VAX 11/780) that can be configured with 120 MB of memory up to 7.2 GB (six 1.2 Gbyte disks). The list price of the CPU with 16 MB of memory and a 19" 8-plane color monitor is $29.5K. Third party SCSI 1.2 GB disks can be purchased for under $5K. This level of competitive pressure does not exist in the proprietary operating system marketplace. A DEC VMS-based workstation in the price range of DECstation 5000 buys approximately 3 MIPS and memory expandable to 32 MB.

Archive Functionality

The System 500 product consists of one or more telemetry/satellite control front-end processor chassis connected by Ethernet to one or more UNIX-based workstations or file servers. The System 500 has a fully distributed architecture that allows front-ends and workstations to be incrementally added to the network. The product is unusual in that it includes a UNIX-based archiving capability.

Users access archiving capabilities by selecting the Datafile Manager option from a mouse-driven menu interface. The Datafile Manager is a state-of-the-art, X Windows-based interactive display. Figure 1 shows an example of a Datafile Manager display.

The user interacts with the display by making selections from a menu. The Datafile Manager allows the user to select the gather type, rate, box, file size limit, and parameters.

The gather type is used to select how the data is collected in the telemetry front-end and how it is sent over the Ethernet network to the workstation or file server. The gather types supported include CVT, STATGAT, and GATALL.

The CVT sends the current value for selected parameters at the time interval specified by the rate value. The STATGAT sends the minimum, maximum, average, current value, and number of samples read for the specified parameters and the specified time interval. The GATALL data gather sends all samples of the specified parameters. The box selection is used to specify from which front-ends on the network data is to be collected.

Once the user selects the desired setup options, he saves the setup selections in the database under a unique name. Once a setup is saved, the user can then start archiving into a data file by using an arm command. The archiving continues until the user issues an unarm command or the data file reaches a user-specified size limit. The Datafile Manager provides data loss failure information to the user, thus giving the user confidence in the integrity of the archiving.

In addition to the standard gather types described, the end user can write his own gathers. The user-created gathers are simple "C" language routines written for the Field Programmable Processor (FPP). The FPP, an embedded RISC processor in the front-end, is used for real-time processing and gathering. A user-created gather need not deal with network protocols; the "C" code simply performs a UNIX-like write request. The write request causes the data to be transferred from the front-end to a workstation or file server. The Datafile Manager display supports archiving for user-created gathers. In addition, there are workstation/file server "C" programmer interfaces (DGLIB) for initializing a gather, suspending a gather, resuming a gather, and terminating a gather. The end user uses the DGLIB to interface his workstation or file server software to front-end telemetry data and to interface to third party software packages.

Playback Functionality

The Display Builder allows users to create custom real-time, quick-look displays with a set of graphic display objects such as bar charts, strip charts and current value boxes. All such created displays have a data file button. Users can dynamically switch between real-time quick-look and data file playback by pressing the data file button. The user can playback a data file while recording is in progress. After pressing the data file button, the user selects the name of the data file to be viewed. The user can optionally select a start and end IRIG time for viewing portions of the data file. After the data file is selected, a scroll bar appears. The user uses the mouse to move the slider on the scroll bar which in turn moves through the data file and display points on the screen. The data can be played back in both forward and reverse directions. Users can play back data a single point at a time in forward or reverse direction by simply clicking the mouse button on the right and left direction arrows on the scroll bar. Note the single stepping maintains the exact order the data had in real-time. A user can have two identical quick-look displays on the same screen with one display showing real-time data while the other display is showing recorded data.

The real-time, quick-look displays also support a history buffer for reviewing the most recent (n) real-time samples that have been displayed. The value of (n) is user configurable. The user activates playback of the history buffer by pressing the playback button on the display. Once activated, a scroll bar appears. The scrolling works in the same manner as the data file scroll bar.

The quick-look displays have a data loss indicator. A data loss bit map picture is displayed if data loss occurs and a border around the data loss bit map picture is latched to indicate data loss history. A mouse click clears the data loss latch. The data loss indicator feature is intended to give the user confidence in the integrity of the data being he is viewing. Figure 2 shows a sample Display Builder screen.

Design

The data file implementation relies heavily on UNIX-based features such as TCP/IP and NFS. Figure 3 shows the data paths for archiving. A FPP-based, user-created gather algorithm in the front-end collects the data to be archived. When the gather is ready to transmit data from the front-end to the workstation or file server, the FPP gather algorithm transfer function executes a UNIX-like write. The Ethernet processor software then transfers the data to the requesting workstation, using the UDP protocol. Data can also be sent reliably using a simple ACKing on top of the UDP protocol. The reliable transmission incurs a 5% overhead.

The design supports a fully distributed architecture. Any workstation or file server on the network can archive data. In a file server configuration, any workstation on the network can initiate archiving on the file server. For redundancy, a front-end can have two identical gathers running, with each gather sending its data to a different workstation or file server on the network.

NFS-mounted files can be used for the locations where data files are recorded, allowing other workstations on the network to transparently access the data files, even during recording. NFS allows the Display Builder data file playback display to read across the network just the data required to drive the display. In playback, this avoids having to copy an entire data file file across the network. The data file formats are straightforward and are documented so that end users and third party software packages can access the files.

The UNIX workstation-based archiving performance is limited by LAN speed and by the protocol used on the LAN. The UNIX-based archive is not intended to replace high-speed disk archiving in the front-end. In fact, the UNIX archiving can nicely complement front-end, real-time, high-speed disk archiving. During postprocessing, the UNIX archiving can be used to upload the data from the front-end, high-speed disk to the UNIX workstation for analysis.

UNIX I/O and Process Scheduling

The UNIX kernel uses I/O buffers to optimize disk I/O. The I/O buffer size for a specific configuration is primarily a function of total memory and code size. On state-of-the-art workstations, this buffer size ranges from 4 MB upward. With the use of an elevator algorithm, data is written into the I/O buffers and actual I/Os are reordered to optimize head movement. The I/O buffer allows a workstation or file server to handle burst data since the data is first written to memory. When playing back archive data concurrently with recording, the read operation checks the I/O buffers for the required data. If the data is found in the I/O buffers, no disk read operation is required.

UNIX was originally implemented as a time sharing operating system. Several computer manufacturers have added priority real-time extensions to the kernel. There is currently an IEEE UNIX real-time standardization effort underway. The UNIX archiving feature described in this paper does not require any real-time extensions.

The following section explains how the UNIX archiving works with a UNIX time-sharing operating system.

- The workstations and file servers are not used as general purpose, time-sharing machines but as special purpose telemetry application processors. As such, the worst-case workstation process loading and process delay can be empirically derived.

- The FPP is a real-time RISC processor with 512 kB of code and data memory. The FPP memory is currently being expanded to 2 MB. There can be up to 8 FPPs in a single front-end. The FPP memory provides the required buffering if the workstation falls behind. The buffer size is a function of the data archiving rate and the worst-case workstation file server process delay.
- The front-end Ethernet processor and workstation Ethernet must be able to burst data to catch up if the archiving workstation ever falls behind. The burst rate should ideally be able to empty the buffer within one second.

Single Point of Failure

Satellite applications and some telemetry applications have a requirement that no single point of failure exist. The System 500 product satisfies this requirement with a dual logical network configuration. A logical net consists of at least one workstation or file server and at least one front-end. There can be any number of logical networks on the same physical Ethernet. In the dual logical network configuration there is one file server on each logical net. Any workstation on the Ethernet can control and display data from any front-end on either logical network. In satellite applications and some telemetry applications, it is not acceptable to allow any user on any workstation to control any front-end. The graphical resource allocation display allows a user with the proper access level to allocate and deallocate specific workstations and front-ends for exclusive use.

There are three critical components to consider in the single point of failure issue:

- Workstation
- File Server
- Front-End Telemetry/Satellite Processor Hardware

In the event of a failure at a workstation, the user can recover by logging onto any other workstation on the network (since any workstation on the network can control any front-end).

A failed file server causes the system to run in a degraded mode. Specifically, the front-ends controlled by the crashed file server cannot have their hardware setup data base modified. The front-ends continue to send redundant archive data to the other file server. Quick-look telemetry displays continue to be updated on workstations and satellite commanding by the workstation continues.

The front-ends that were controlled by the crashed file server can be switched over to the other file server. The switch-over does require downloading the hardware setup data base again and is accomplished via the resource allocation display.

By downloading a setup from a backup front-end, a user can recover from a crashed front-end. If no delay can be tolerated, two front-ends can be downloaded with the same setup, with the second front-end used as a hot backup. Note that on a single workstation screen, the user can display data and control two front-ends that are on different logical networks.

Conclusion

The availability of UNIX-based standard software building blocks as well as leading edge UNIX hardware platform performance continues to make UNIX a very attractive platform for telemetry and satellite control products. This paper has presented a brief description of some features of one such product, Loral Instrumentation's System 500.

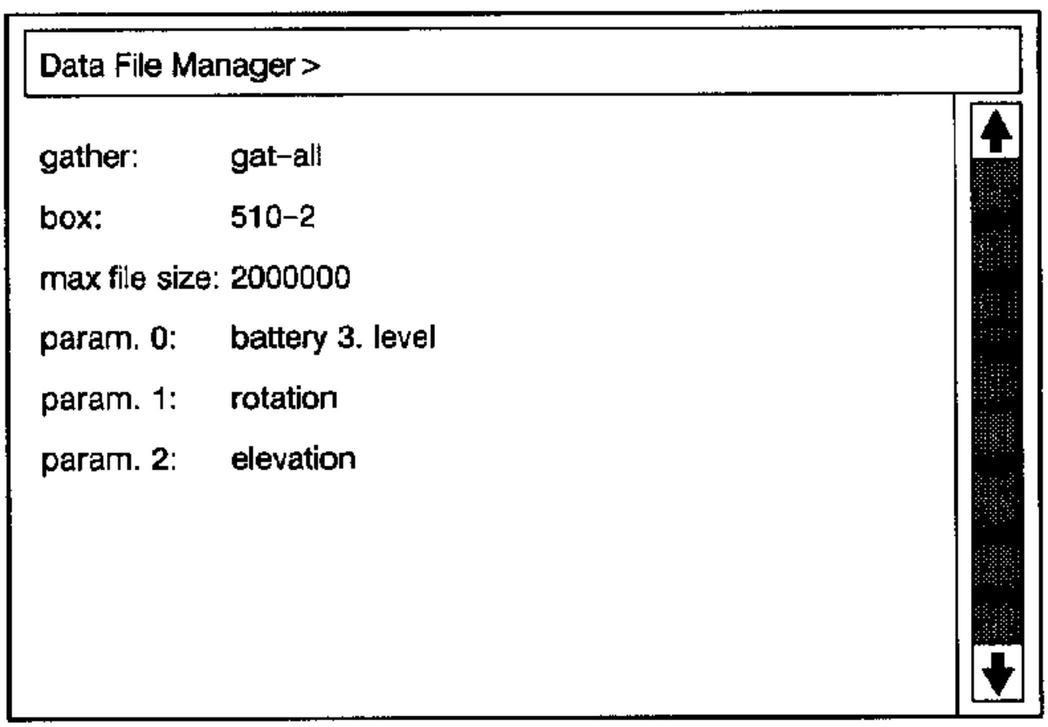
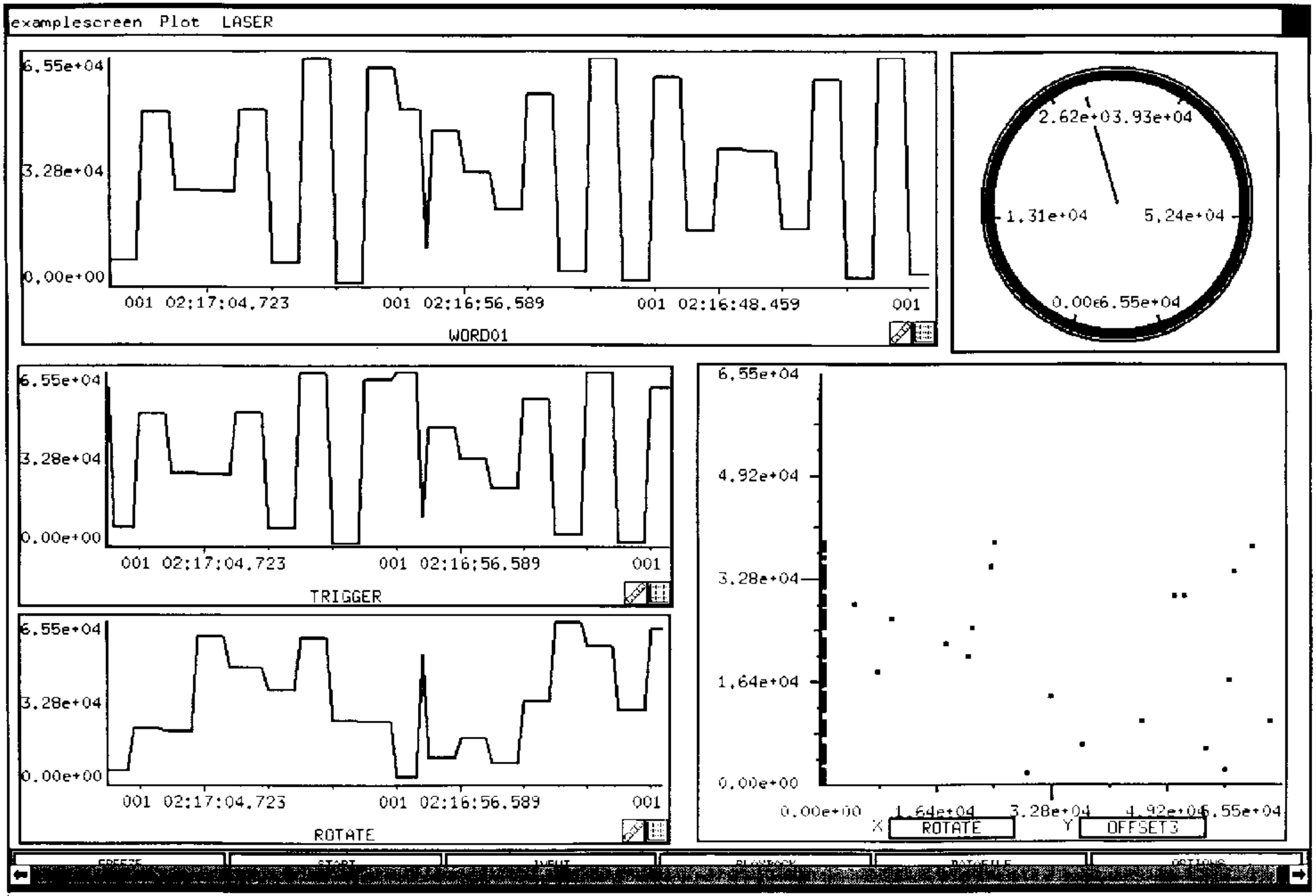


Figure 1. Data File Manager Display



examplescreen Plot LASER
6.55e+04
3.28e+04
0.00e+00
001 02:17:04.723
001 02:16:56.589
001 02:16:48.459
001
WORD01
2.62e+03.93e+04
1.31e+04
5.24e+04
0.00e6.55e+04
001 02:17:04.723
001 02:16:56.589
001
TRIGGER
ROTATE
4.92e+04
1.64e+04
0.00e+00
1.64e+04
3.28e+04
4.92e+06.55e+04
X
ROTATE
Y
OFFSET3

Figure 2.

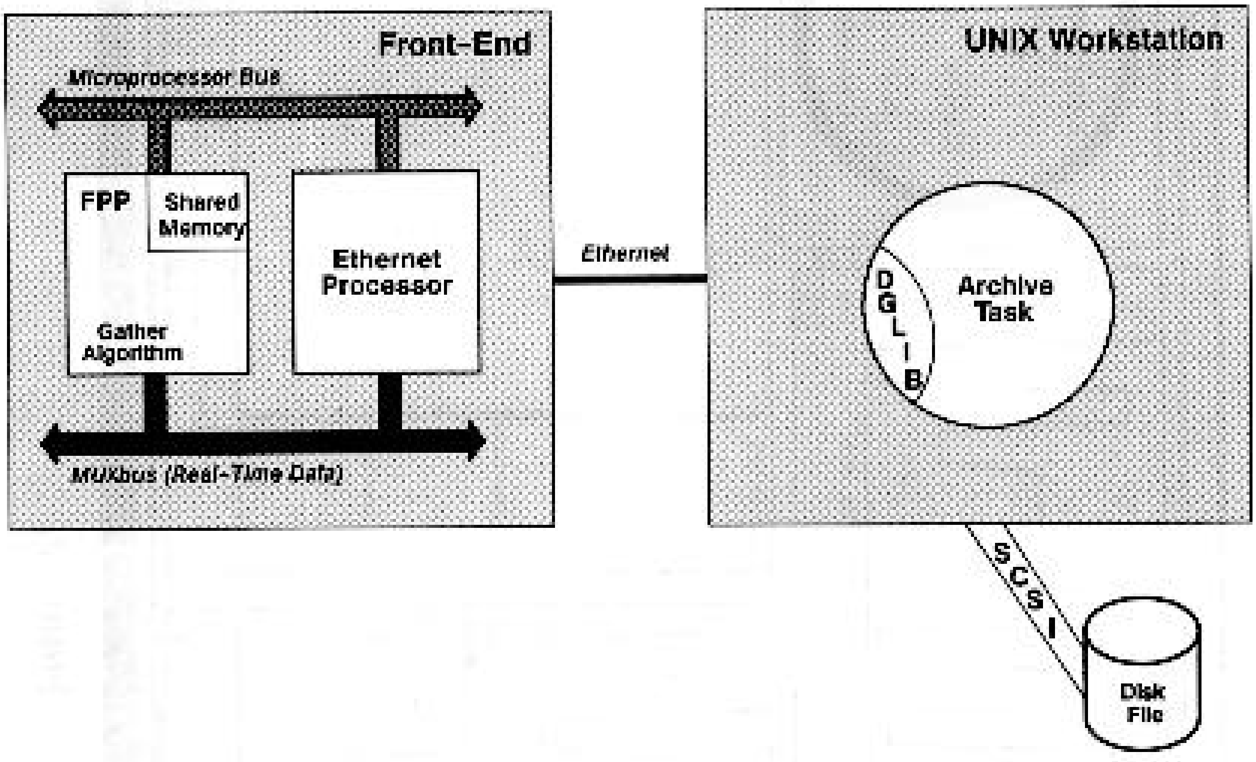


Figure 3. Archive Dataflow Diagram