The Star, a Dynamically Configured Dataflow Director for Realtime Control*

M. Bickley and J. Kewisch
Continuous Electron Beam Accelerator Facility
12000 Jefferson Avenue, Newport News, VA 23606-1909 USA

Abstract

The CEBAF accelerator is controlled by an automated system consisting of 50 computers connected to machine hardware and another 20 to 30 computers used for displaying machine data. The control system communication software must manage the inter-machine communication of these computers. Each of the different segments of software that make up the machine control system is treated as data sources and data sinks, with a single process mediating the transfer of all data between any data source/data sink pair. The mediating process is called the Star. This dynamically configured process keeps track of all available machine data posted by data sources and of all data requested by data sinks. Data transmission rates through the Star are kept low by sending only data that is requested by other control software, and then only when the value of the data changes. The system is entirely response-driven, with the Star process taking action only at the request of either a data source or a sink. The software for the communication is written using standard C code and TCP/IP sockets, making the communication software platform independent.

I. INTRODUCTION

The control system in place at CEBAF in the fall of 1991 used reflected memory to transport data between the data sources and data sinks. The machine information that was produced at the data sources was stored in a block of shared memory. This block of shared memory was copied from the data source machines to the data sink machines by custom software which ran continuously, cycling constantly. Some fraction of the total CPU time on every machine was devoted to performing this operation. In addition, all applications which used the data were forced to poll the appropriate memory locations in order to determine if the value of the data had changed.

Although this technique was effective for small systems, it became a burden as the amount of machine data to be transferred increased. Once the complete CEBAF injector was installed, the cryomodules that are part of the injector increased the volume of machine data dramatically. The increase in machine data caused a corresponding increase in the time taken to pass the blocks of shared memory over the network. This resulted in a significant increase in response time to operator input. It was clear that as more of the accelerator was installed, the performance of the system would degrade so much that the machine would be very painful to operate. A new data transfer paradigm was needed.

II. A COMMUNICATIONS PARADIGM

In order to create the new data transfer paradigm, the strengths and weaknesses of the existing system were examined. The following requirements of the communication protocol were deemed necessary for the new software:

- The communications should be connection-based, so it would be clear at either end of a connection when the other end had been closed. This would greatly simplify the bookkeeping associated with the data transfer.
- The database organization associated with data transfer should be dynamically allocated. It should be clear when new machine data sources are added to the system and are available to data sinks.
- The complexities of the network transfer should be concentrated on a single computer. This increases the average size of the network packets, making the network data transfers more efficient. It also minimizes the CPU effort devoted to communications on the data source and data sink computers.
- The data flow should be based on a single request-multiple reply query protocol. This minimizes the overhead associated with maintaining a data flow channel.
- Data should be transferred only as it changed. It made no sense to burden the communication software with the delivery of data that provided a data sink with no additional information.
- The communications software should be as general as possible, to simplify porting to new computer platforms.

In order to fulfill the desired requirements of the communications paradigm, the logical topology to use is a star. This organization, with a single central computer, allows the total system throughput to be governed by the performance of that computer. The central computer (which was named, rather simply, the "Star") could be a system with as much CPU horsepower as was needed to adequately handle the expected load of the complete accelerator.

III. FUNCTION OF THE STAR

The Star computer is fundamentally just a manager of the dataflow between the data sources and the data sinks. Its principal task is to forward data requests from data sinks to data sources, and to forward data responses from the sources back to the sinks. In addition, the intelligence for the control of data traffic is kept in the Star. This allows the sources and sinks to be ignorant. The data source does not have to keep track of the destination of all of the data flowing out, and the data sink does not have to be aware of the source of the data that comes into it.

In order to accomplish the management of the dataflow, the Star keeps three interrelated data structures. The

* Supported by DOE contract DE-AC05-84ER40150.

0-7803-1203-1/93$03.00 © 1993 IEEE

first data structure is a hash table that contains a reference to every piece of data available from any data source. The second is a sink table, used to associate the current data sinks with the data sources from which the data items originate. The third data structure is a source table, which associates the current data sources with the data sinks that have made requests for data values.

The hash table associates with each table entry an identifier that indicates from which data source the relevant data item originates. The hash table is used when a data sink makes a request for a data element. If the data element is found in the hash table then a request for the data value is forwarded to the appropriate data source. If the data element is not found in the hash table, then the data sink is informed that the data is not available.

Once a data value has been requested by a data sink process and the data element is found in the hash table, the data sink table and data source table must be altered. The sink table is used to keep track of which data elements have been requested by which sinks. In the event that one of the sinks terminates, this table is used to identify those data elements which the data sink had requested.

The other data structure, the source table, is similar to the sink table. The source table is used to keep track of which data elements are being supplied by each data source. In the event that a data sink process terminates, this table is used to inform the associated data sinks that the requested data elements are no longer available. In addition, the source table is used to track how many requests have been made for each data element. When a data element is requested by multiple sinks, only a single request is made of the responsible data source. It is not until the data is no longer needed by any sink that the request for data from the source is terminated.

IV. TESTING OF THE STAR

It was necessary to make a good estimate of the ultimate load on the Star once the complete CEBAF accelerator was installed and running. This was done by extrapolation from the portion of the machine in use at that time, the injector. There were a total of 10,000 data channel read backs in use, and of those 10,000 approximately 13% changed per second. The cycle rate of the control algorithms on the data source computers was 3 Hz, but the plan was to increase the rate to 10 Hz. Once the performance of the data sources was enhanced, it was expected that the number of data source changes per second would triple, resulting in about 40% of the read backs changing per second.

Furthermore, on a typical CEBAF display page, there are about 100 data channels, of which 40 are set points and the other 60 are read backs. Taken together with the fact that a change rate of 40% of the total number of read backs per second was predicted, the anticipated system load was 24 changes per second per display page. The completed accelerator was expected to have a total of 100 display pages in use at any time, giving an expected system load of 2,400 changes per second.

Preliminary testing of the Star computer was performed using an HP720 computer. Software was written which functioned as a simple data source, where the total volume of data out of the data sources was controllable in each of two ways. First, the interval between data transfers was variable, at rates up to 20 Hz. Second, the number of data points sent per transfer was variable, from 100 to 1000 data values. Each data value consisted of a floating point value, of length 4 bytes. The data sinks used for the testing were the same graphical display software used in the normal control of the accelerator.

For the testing, the number of data values sent per transfer from each of 9 artificial data sources was varied from 100 to 1000. For each transfer volume, the transfer rate was increased from 1 Hz to either the maximum transfer rate of 20 Hz, or until the system saturated. System saturation is the point at which the Star computer can no longer keep up with total volume of traffic, so buffer overflow occurs. For each volume/rate pair the average time between the issuance of a new value by a data source and its receipt by a data sink was measured. The performance of the Star computer with respect to signal load was linear for all tests, with system one-way travel time ranging from 0.0025 to 0.08 seconds as the signal load increased from less than 1,000 changes per second to 24,000 changes per second. At the upper end of the range, above 24,000 changes per second, the one-way travel time degraded worse than linearly. Once the upper limit of 27,000 changes per second was reached, the system broke down completely, with buffer overflow in the Star. See Figure 1.

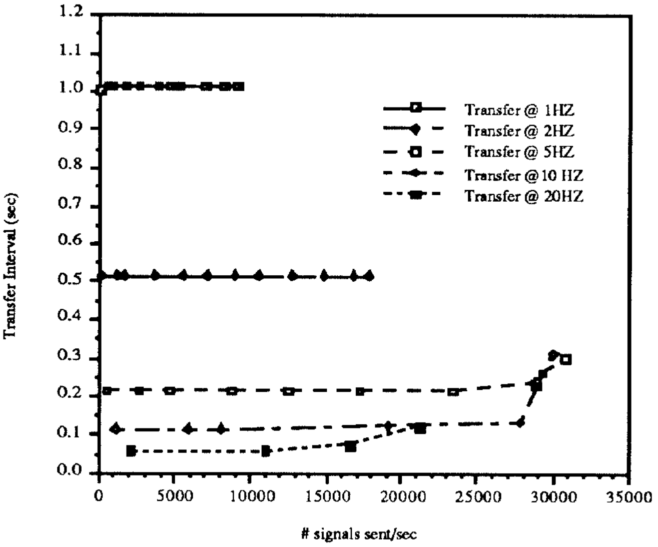

Figure 1. Data transfer interval vs. # signals sent.

The testing demonstrated that on an HP720 computer, rated at 56 MIPS, the Star process could support a dataflow rate of 10 times the expected load of the complete accelerator. Similar testing was done with different computers in order to determine if the performance of the Star process scaled with the performance of the computer. The most extensive testing was done with an HP835 computer. The results of the testing, along with an extrapolation of the expected performance from an HP735 computer, are shown in Figure 2.

The Star process, running on an HP735 computer rated at 125 MIPS, should support a dataflow rate of 25 times the expected load of the CEBAF control system. Given these favorable test results, the Star process was implemented as part of the CEBAF control system, and was used for the commissioning of the CEBAF North Linac and East Arc from August of 1992 until April of 1993.

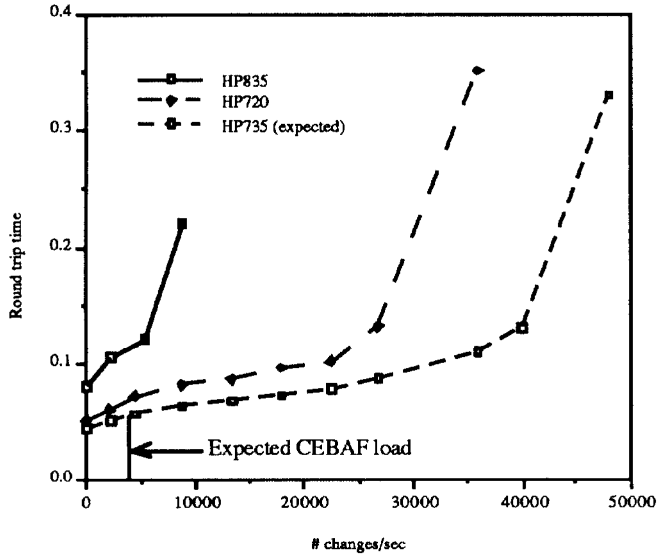

Figure 2. Round trip time vs. changes/sec for different computers.

V. OBSERVED PERFORMANCE

Performance statistics for the Star were kept during the commissioning period mentioned above. There were two goals of the data collection during this time. The first goal was a demonstration that the signal load extrapolations made when examining the injector were valid when scaling the system to include more of the CEBAF accelerator. The second goal was a measure of how the system performed in a real-world situation. It was necessary to find out if the system performed as expected, and also to see if other Ethernet traffic affected the performance of the Star. The data that was tracked during the commissioning consisted of two elements. The first element was the total volume of data traffic through the Star computer, and the second was the average data round trip time.

The traffic volume through the Star process was trivial to measure. A counter, which was incremented as each data item passed through the system, was added to the Star software. After ten seconds had passed, the total data load measured was saved, and the counter was reset. The system load was tracked over time and compared to the average data round trip time.

Determining the average round trip time was a more complicated procedure. Two simple programs were written, one a data source and one a data sink. The data source was a source for only one data item. The data sink was a sink for that one item. The data sink started a timer, then requested the value of the data item. When the value of the data was received, the data sink made a request to the Star that the value of the item be changed. This sort of request is akin to a machine operator turning a knob or otherwise changing a machine set point. The data source changed the value of the data item, then sent the new value to the data sink. When the updated value was received by the data sink (indicating the request had been honored) the sink requested a different new value. The data sink made a total of fifty change requests, then stopped the timer. The total time elapsed then included a total of fifty round-trip passages of the data through the Star. The data sink waited five seconds after completion of the measurement and started the process over.

During a typical month of commissioning, from January 2 to January 29, 1993, the total data load and average data round trip time measurements were recorded. This data is reflected in Figure 3.

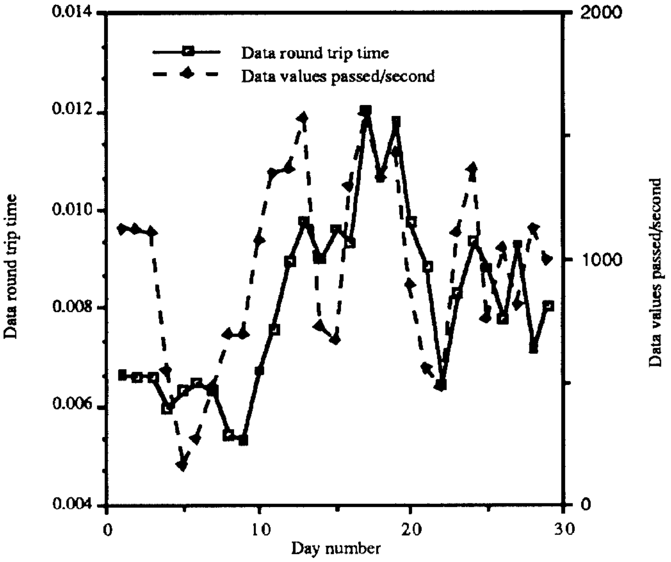

Figure 3. Observed Star performance.

During this running period, approximately 25% of the CEBAF accelerator hardware had been installed, and 12 of the expected 50 display consoles were in place. The average data load during the period was slightly more than 1,000 data elements per second, with peaks as high as 2,000 data elements per second. The data round trip time under these loads was always well under 0.1 seconds, and typically was under 0.02 seconds. The measured load was greater than was expected when extrapolating from the injector, as mentioned in section IV, but was still well within the capabilities of the Star process running on the HP720 computer used during the commissioning.

It appears that the completed accelerator will have a signal load of 4,000 data elements per second, with peaks as high as 8,000 per second. Given this updated data traffic information, the Star process running on an HP735 computer will be capable of supporting 5 times the expected peak load and 10 times the typical load when controlling the finished CEBAF accelerator.

VII. REFERENCES

[1] R. Bork, C. Grubb, G. Lahti, E. Navarro, J. Sage, T. Moore, "CEBAF Control System," CEBAF-PR-89-013