FROM 15 MINUTES TO 7 MINUTES--A PROGRESS REPORT ON IMPROVING THE PERFORMANCE OF THE TANDEM MIRROR EXPERIMENT-UPGRADE (TMX-U) DIAGNOSTIC COMPUTER SYSTEM

H. H. Bell
M. D. Brown
J. M. Moller
W. H. Meyer

This paper was prepared for submittal to 11th Symposium on Fusion Engineering Proceedings
Austin, Texas
November 18-22, 1985

November 11, 1985

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

Unclassified
FROM 15 MINUTES TO 7 MINUTES--A PROGRESS REPORT ON IMPROVING THE PERFORMANCE OF THE TANDEM MIRROR EXPERIMENT-UPGRADE (TMX-U) DIAGNOSTIC COMPUTER SYSTEM

H. H. Bell, Jr., J. W. Hailer, A. W. Benway, M. D. Brown, and W. H. Meyer
Lawrence Livermore National Laboratory
P.O. Box 808, L-540
Livermore, CA 94550

Abstract

May 1983 marked the beginning of an intensive effort to both improve the operating reliability, and improve the performance of the TMX-U Diagnostic Computer System. At that time, the system was handling, (acquiring, storing, processing, plotting, displaying, and archiving) about 3 million bytes (Mb) of data per shot, with a 15-minute cycle time between shots. In addition, the system was fairly fragile, with frequent (about 5 times/day) crashes, requiring re-booting. At the present time, the system reliably handles about 5 Mb of data per shot, with a 7-minute cycle time between shots. This improvement was accomplished by a combination of new hardware, rearranging existing hardware, and new or revised software.

Hardware changes were made in two areas. First, the shared disks were rearranged into different domains to make more efficient use of locking features. Second, we purchased and installed a solid-state RAM disk emulator (8 megabytes) to provide extremely fast access to lists and files that must be accessed frequently.

In the software area, we made improvements in several areas. Initial effort went into finding bugs and optimizing existing code. We developed a template so that we could produce efficient code from applications that had first been developed on a very powerful, general-purpose scientific spread sheet. Also, we adopted a ticket book algorithm to allocate disk resources more efficiently.

"Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract number W-7405-ENG-48."

Introduction

The TMX-U Diagnostic Computer System consists of a network of seven Hewlett-Packard (HP) 1000/F computers.[1] This system acquires most of the physics data from TMX-U, stores the data temporarily, processes it, produces plots, lists and displays, and finally archives the data to tape. The computer system is presently connected to about 900 data channels or transducer elements, although about 500 channels are used on a typical shot. Considering both raw and processed data, the system handles about 5 megabytes (Mb) of data during a 7-minute cycle time (during the last experimental run).

The computer system is configured with five machines connected directly to the experiment (see Figs. 1 and 2). One machine is used for off-line data processing (see Fig. 5) and one machine is used for software development. All of the machines are inter-connected with DS-1000 serial data links, which are used primarily for file transfer and short synchronization. The five experiment computers are also connected to 720 Mb of shared disk (see Fig. 3). All of the experimental data is transferred between machines via the shared disks. Figure 4 shows the physics operations positions.

All diagnostic data are stored in disk files, one file per shot. All data are stored in the form of attributes, which are of two types: dependent and independent attributes. Dependent variables store the results of measurements or calculations, while independent variables describe the timebases and measurement geometries.

Hardware Changes

Disk Reconfiguration:

The TMX Data acquisition system uses the capabilities of the HP 13034C Multiple Access Controller (MAC) disks. Each MAC controller can connect one or more disk drives to one or more CPUs. Our five CPUs were connected to eight shared disk drives through three shared MAC controllers, to form three domains.
Disk access is affected by three kinds of contention: head contention, channel contention for each controller's single data channel, and lock contention. Head contention arises from the requirement for exclusive access to a file during an update operation. To support update access to shared files, each MAC controller can be locked.

To minimize head contention, our data files were already divided into multiple extents residing on separate disk drives. Critical tables were already partitioned among the shared controllers to minimize lock contention. Re-partitioning the disk drives among the shared controllers to minimize channel contention yielded a further modest 10-15% performance improvement.

MEDARAM Addition:

The shot queue and several tables are accessed very frequently by all five computers during acquisition and processing. For instance, the shot queue (about 200 Kb) is read from disk every time a computer completes a task (each computer can handle three tasks at a time in our multi-tasking environment). The computer examines the list, selects the next uncompleted task that it is capable of performing, "checks" it off, and writes the queue status back to disk. To speed up this process and to reduce contention for other disk resources, we purchased an 8 Mb solid-state RAM disk emulator from Imperial Technology. It was installed as a fourth domain on our shared disk subsystem. Since there are no moving parts, the head positioning latency and rotational latency were eliminated entirely. This saves between 50 and 250 msecs for every access of the queue and tables.

Software Changes

The most obvious place to start to improve reliability and performance in a computer system is to find and fix the bugs and optimize the code. These two areas were pursued vigorously, and the effort yielded significant results. In addition, several major software efforts were aimed directly at improving reliability and performance.

Checksumming:

A checksumming and authorship scheme was added to the processing queue code to provide a mechanism for detecting read/write errors to the shot queue and control tables. This new process provides a transparent method of recovering from corrupted writes or reads between extended memory area (EMA) and disk/MEDARAM. These errors occur due to transient hardware problems and possibly due to software bugs.

The basic scheme is that there are two tables which are used for every read/write access to the disk. One of these is a checksum table, and the other is an authorship table. Each table has 100 entries, as there are 100 pages in the shot queue. Each time a CPU writes to the shot queue on disk, each of the affected pages is checksummed. The matching entry in the checksum table receives this value, and the matching author page is updated with the CPU's node number.

When a read is made from disk to a CPU's EMA image of the shot queue on disk, the incoming pages are checksummed. This calculated value is compared to what is in the checksum table for that particular page. If these don't match, the recovery algorithm is entered. A re-read is attempted first to test for a transient hardware glitch. If that fails, then the
Ticket Book Algorithm:

One of the major software improvements for speeding up the TMX-U system was the implementation of a "ticket-book" method of allocating disk space. Tests showed that our five CPUs were often sitting idle, waiting for one of the shared disks that was locked by another CPU. We searched for a way to alleviate this problem and found a solution. The original method of allocating disk space for a particular shot required that one of the shared disks be locked, to allocate the necessary disk space, every time a header or data record was to be written to one of the shared disks. Then the shared disks had to be locked again to actually write that header or data to disk. The "ticket-book" method is a way of allocating extra large chunks of disk space (a ticket-book) to each CPU, and then that CPU doing local allocations of disk space (selling tickets) when required. During the course of processing a shot, the number of disk locks for allocations was about 2500. With the ticket-book method, this number went down to about 700 and our overall system speed improved by 15 percent.

The Future:

We are in the process of connecting powerful desk-top computers (workstations which handle complex diagnostics) to the existing system with modern data communications techniques. When completed, we should be able to handle 10 Mb of data in about five minutes.
Conclusion

The TMX-U Computer System improvements, undertaken over a 2-1/2 year period, were successful in improving reliability and ease of operation, and in increasing the processing performance by a factor of five. The largest part of this effort was carried out by three programmers, three physicists, and an electronics engineer. This was done while operating the system and making all of the other changes and additions that are required on a large, dynamic physics fusion experiment.

Acknowledgements

We would like to thank Minette Lewis and Raylene Cooper for typing the manuscript Camille Vandermeer for editing and artwork services.

References


DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.