DE 93 1019 19

GA–A21404

ACCESS TO DIII–D DATA LOCATED IN MULTIPLE FILES AND MULTIPLE LOCATIONS

by

B.B. McHARG, Jr.

OCTOBER 1993

GENERAL ATOMICS

DISCLAIMER

GA–A21404

ACCESS TO DIII–D DATA LOCATED IN MULTIPLE FILES AND MULTIPLE LOCATIONS

by

B.B. McHARG, Jr.

This is a preprint of a paper to be presented at the 15th IEEE Symposium on Fusion Engineering, October 11–15, 1993, Cape Cod, Massachusetts, and to be printed in the *Proceedings.*

Work supported by
U.S. Department of Energy
Contract DE–AC03–89ER51114

GENERAL ATOMICS PROJECT 3466
OCTOBER 1993

GENERAL ATOMICS

ACCESS TO DIII–D DATA LOCATED IN MULTIPLE FILES AND MULTIPLE LOCATIONS

B.B. McHarg, Jr.
General Atomics
P.O. Box 85608, San Diego, California 92186-9784

ABSTRACT

The General Atomics DIII–D tokamak fusion experiment is now collecting over 80 MB of data per discharge once every 10 min, and that quantity is expected to double within the next year. The size of the data files, even in compressed format, is becoming increasingly difficult to handle. Data is also being acquired now on a variety of UNIX systems as well as MicroVAX and MODCOMP computer systems. The existing computers collect all the data into a single shot file, and this data collection is taking an ever increasing amount of time as the total quantity of data increases. Data is not available to experimenters until it has been collected into the shot file, which is in conflict with the substantial need for data examination on a timely basis between shots. The experimenters are also spread over many different types of computer systems (possibly located at other sites). To improve data availability and handling, software has been developed to allow individual computer systems to create their own shot files locally. The data interface routine PTDATA that is used to access DIII–D data has been modified so that a user's code on any computer can access data from any computer where that data might be located. This data access is transparent to the user. Breaking up the shot file into separate files in multiple locations also impacts software used for data archiving, data management, and data restoration.

INTRODUCTION

DIII–D is a large tokamak research experiment operated by General Atomics (GA) under contract with the Department of Energy (DOE). Primary goals of the experiment include studies of the properties of high temperature plasmas approaching fusion reactor like conditions. These studies are particularly important for providing data for next generation fusion devices. There have been substantial changes in the last two years to both the quantity of experimental data acquired in the experiment and to the topology of the computer systems where data is acquired. These changes have thus led to upgrades in the computer systems [1] and some substantial changes in the software for accessing and managing data on these systems.

DIII–D is a pulsed experiment that takes "shots" about once every 10 min with each shot lasting up to 10 sec. There may be 40 to 50 shots in an operating day. In the last two years the quantity of data has doubled from 40 MB/shot (the largest being 50 MB) to currently 80 MB/shot (the largest being 90 MB). This quantity is expected to double again within the next one to two years. Until now data has always been collected from various sources and combined into a single shot data file for each shot of the experiment. Historically this was logical, as the data quantity was smaller and the data was in the shot file and available for examination shortly after the shot. Also when data was restored for later analysis, that analysis used data from most diagnostics. As the quantity of data increases, the size of the data files even in compressed format has become increasingly difficult to handle. In addition there are substantial delays in data availability as data waits to be placed into the shot file. These delays often extend into the next shot cycle thus delaying between shot data examination.

Another major change in the DIII–D computer systems has been the introduction of UNIX computer systems to perform data acquisition and analysis. The UNIX systems use TCP/IP for networking. These systems as well as VAX computer systems need to access data in a timely manner between shots. Also, some of these systems can potentially be at a remote site other than at DIII–D.

For all of these reasons it has been decided to break up the shot file into a number of separate shot files. A particular plasma diagnostic system may then create its own shot file locally. In many cases, data analysis may only require data from that shot file while in other cases analysis may require data from shot files from several different systems. The problem for data access then is to know where data is located and in what file and how to access it given the two different networking protocols which exist among the systems, some having only one of the protocols and some having both.

CURRENT DATA ACQUISITION AND DATA ACCESS CONFIGURATION

Fig. 1 is a generic diagram illustrating the configuration of the computers and network at DIII–D. The User Service Center (USC) is primarily used for data analysis and data restoration. DIII–D is primarily used for data generation, some real time analysis, and diagnostic and tokamak control of the experiment. All shot data files reside on the computer interconnect (CI) disks attached to the primary (CI) nodes. The USC is about a

Manuscript received October 12, 1993. This is a report of work sponsored by the U.S. Department of Energy under Contract No. DE-AC03-89ER51114.

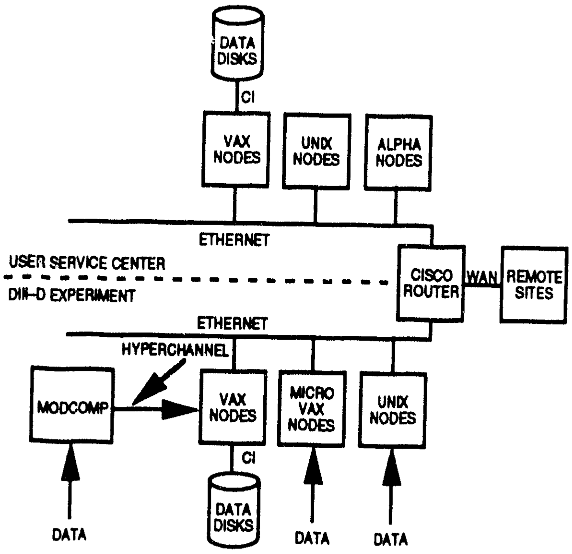

FIG. 1. DIII–D computer system generic block diagram.

mile from the DIII–D site and is connected to it by a fiber optic ethernet link. A Cisco router connects the separate ethernets of the two computer centers. The router also connects to a wide area network (WAN) where data may be accessed from remote locations.

Currently there are four methods by which data is acquired and placed into the shot file (Fig. 2). These are

1. MODCOMP collected data from the Hyperchannel link.
2. MicroVAX generated pointname files from DECnet.
3. UNIX system generated pointname files from DECnet.
4. Pointname collection from a disk NFS mounted by UNIX systems.

In this figure the solid lines represent hardware network connections while the dotted lines represent software virtual connections.

Approximately one-half of the DIII–D data is collected by the MODCOMP data acquisition system [2,3]. Pointnames are grouped into logical experiments, and as each experiment is collected it is written over the Network Systems Hyperchannel data link to a VAX 6410 computer (VAXS). The VAX then writes out the data to the shot file where it is then available for access and examination. Because so much data is acquired, this process continues over most of the shot cycle, and in some cases can limit the shot cycle.

Methods 2, 3, and 4 are similar in that they all involve a system writing a file for each pointname collected. Those files are then read by a task on VAXS which in turn adds the pointname to the shot file.

There are numerous MicroVAX computers in the DIII–D VAX cluster which collect data. These systems

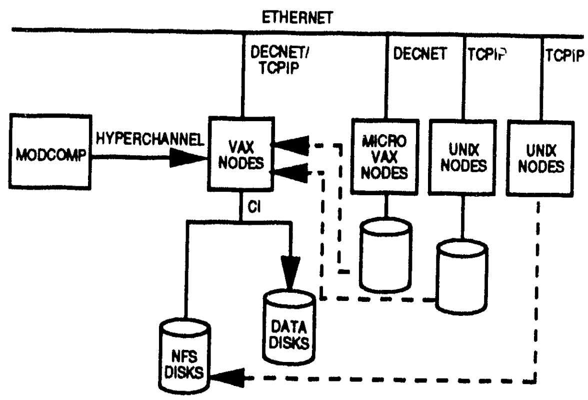

FIG. 2. Data input to shot files.

write the pointname files onto a disk local to the MicroVAX. These systems are part of the VAX cluster and their disks are mounted via the VAXcluster software on VAXS. These files are then picked up by VAXS and incorporated into the shot file [4–6].

The UNIX systems are used for data analysis. Originally such systems that were acquiring data and needed to create pointname files were required to have DECnet because VAXS did not have the TCP/IP capability that all UNIX systems have. It was then possible to acquire data from the UNIX systems using DECnet just as was done for the MicroVAX systems.

More recently TCP/IP has been added to VAXS and it is possible for UNIX systems to write pointname files to an NFS mounted disk that is local to VAXS. The pointname files are then incorporated into the shot file. However there is an increasing number of such systems, often with a large number of pointnames.

The methods for acquiring these pointname files were originally intended to only handle a few tens of pointnames. Now, partly due to the UNIX systems that are coming on line to acquire data, the number of pointname files has grown into the hundreds. Because the MODCOMP acquired data has a higher priority for getting into the shot file, the MicroVAX and UNIX acquired data is generally delayed getting into the shot file and thus is not accessible until later.

Once data is in the shot file it is available for data access from any system via a PTDATA call. PTDATA is the universal data interface for DIII–D data. It was originally written for the VAX but has been transported to UNIX systems for accessing data residing either on the DIII–D VAX cluster or the USC VAX cluster. The user calls PTDATA with a shot number, a data type, a pointname (which must be a unique name), and certain parameters describing what data is to be returned.

NEW DATA ACCESS CONFIGURATION AND PTDATA SEARCH ALGORITHM

There are a number of plasma diagnostics that generate a substantial amount of data, and analysis of this data may not require other data. Thus if analysis can be performed on the same computer on which the data is acquired, the results would be available sooner rather than waiting until the data has become part of the shot file.

The new configuration involves creating a shot file on a system's local disk and writing pointnames to that shot file. All existing shot files have names of the form shot.PLA where shot is the shot number. These new shot files will have an extension other than .PLA, with a name appropriate to the particular diagnostic. Once written, these shot files need to be accessible from anywhere just as the main shot.PLA file is accessible from anywhere. This has led to considerable software additions to PTDATA in order to be able to find data and access it as efficiently as possible.

There are a number of characteristics of the computer systems which must be taken into account in designing the search algorithm for PTDATA. One crucial item is that there are two different networking protocols that may be used, DECnet and TCP/IP. Some systems only have DECnet and thus can not directly access a system that only has TCP/IP. This is similarly true for a system only having TCP/IP networking. There are some systems, however, which have both. Because all systems do not have both protocols, in some cases it is necessary to use a system having both protocols (such as VAXS) as a gateway in order to access the data. Another characteristic is that MicroVAX systems in the DIII–D VAX cluster, have the central data disks mounted via the VAXcluster software as if they were local disks. Similarly, the central nodes (VAXS, VAXT) have the MicroVAX disks mounted as if they were local disks. Thus a PTDATA call on VAXS for data on a clustered MicroVAX would access that data as if it were local rather than accessing via DECnet. Although both methods involve accessing data over ethernet, the VAXcluster software is much more efficient than DECnet. Another characteristic involves the specification of the file extension in the PTDATA call. It is possible that a pointname may be under one extension now, but in the future will be under another extension. Also, a user may have just specified the wrong extension. In order to alleviate the burden on the user to always specify the right extension, PTDATA must be able to determine the correct extension if necessary. Another important point to consider is that access to data on a local disk is much more efficient than accessing data via a network protocol.

Based on these considerations, the following "rules" have been developed in order to allow PTDATA to find requested data.

1. No matter where PTDATA is looking for data, if it is not found in the specified shot.extension, then PTDATA will search any other extensions of the same shot number that may be present at that location. For example, shot.PLA and shot.SXR might be in the same location, and the user has requested a pointname in shot.PLA, and PTDATA has found it. Now the user requests another pointname but is still specifying shot.PLA when in reality it is in shot.SXR. Since PTDATA does not find it in the .PLA file and since it is already connected to that location, PTDATA will then check the shot.SXR file which is located there and thus find the pointname.
2. If the previous PTDATA call has found the requested pointname in a particular location, then when the user asks for another pointname, then PTDATA will look in that previous location first. As in item 1, the two pointnames may be in two different files, but the caller may have specified .PLA in both cases. It is more efficient to go ahead and check that location first rather than checking elsewhere and having to come back to that location.
3. If data was not found at the previous location, then PTDATA will check the local disk for any shot file with the correct shot number. Data is normally found under the logical name SYS$D3 on the VAX or the path SYS_D3 on UNIX systems. In addition, on a MicroVAX system in the DIII–D cluster, data may be found under the logical D3DATA. Local disk access is significantly faster than network access, so there is little penalty in searching first locally. There are a couple of peculiarities to the meaning of local. If the caller is on VAXS or VAXT in the DIII–D cluster and the requested extension is known to exist on a MicroVAX in the cluster, then that is considered to be local data and PTDATA will look there (under logical name node$$D3DATA where node is the MicroVAX node name) as well as in the central data area (SYS$D3). If the caller is on a MicroVAX in the DIII–D cluster, then PTDATA will look on its local disk (D3DATA), but will also look in the central data area disks (SYS$D3) which are mounted via the VAXcluster software as local disks.
4. If data is not found on local disks, then PTDATA needs additional information. The USC maintains a table of currently known shots and extensions and their locations. This table provides information on which computer system a particular shot.extension is located, and also what computer system generates data with a particular extension. Since PTDATA will have already looked locally, this new location is expected to be remote and thus a network connection will be made to that remote location.
5. If data is not present at the known location or there was no known location, then PTDATA will attempt further network connections. PTDATA will try the

DIII–D central data disks, then the USC central data disks. Finally, if there is a known location for this shot file extension, then PTDATA will try to link directly to that computer system. If the location does not have the same protocol as the caller, then it is necessary to gateway through VAXS or VAXT, which have both network protocols.

6. At this point there is again a need for further information. This step has not yet been implemented, however the plan is to maintain a table of pointnames and a list for each pointname of any extension that the pointname has appeared under. It is expected that a pointname will only appear under a very few possible extensions. Once the extensions are known, then along with the known locations of those extensions, those locations can be searched.

A number of pieces of software are necessary for implementing data access by PTDATA in addition to PTDATA itself. Any computer system that supports data locally on its own system needs to have a server process (PTSERVER) present in order to serve data to a caller on another computer system. Most systems only support one protocol. However, the primary nodes at the USC and DIII–D support both, and PTSERVER must determine whether the connection being serviced uses DECnet or TCP/IP protocol.

The USC is the server for these various tables of information which PTDATA may use in finding data. Whenever a file is "created" somewhere, its location is sent to the server on the USC, so that there will be a known location for that shot.extension. A file is created whenever it is created as a brand new file, or the file is compressed into a compressed data file, or a file is restored to disk from tape. The presumption is made that, when a file is "created," the new location should be the known one. Note however, that a system having data on its local disk would continue to access the data on the local disk even though the known location for the data might be elsewhere. There is a server process at the USC that serves requests for data locations and receives information for new files that have been created.

This paper has primarily emphasized the software changes to PTDATA for finding data in multiple files over multiple systems. However, a considerable amount of other software has had to be modified or written. Software for writing out the pointnames to the shot file is quite different on the UNIX systems as compared to the VAX, but will create the files in the same format. VAX format is always used for integer and real numbers, and PTDATA makes appropriate conversions. The data compression code had to be ported to the UNIX systems. Data management of the data areas for the UNIX systems still needs to be developed.

CONCLUSIONS AND FUTURE DIRECTIONS

Breaking up the shot data file into multiple files in multiple locations is expected to provide considerable benefits in allowing data to be accessible sooner. Also, more data is allowed to be present on disk since a restore request may only need data from a particular shot.extension rather than all extensions for a shot. The changes made to PTDATA have enabled users to continue to access data from any location regardless of where the data resides.

Modifications to PTDATA have been an evolutionary process that will continue into the future. As data quantity increases, and more data is accessed via the network, a considerable slowdown is often seen in accessing the data. There are a number of potential improvements that could be made to PTDATA (or related codes) that may improve this situation. One idea that is relatively straight forward to implement is to keep network links open. Another improvement might be to copy shots (or partial shots) to another system in some automatic manner thus providing local access. A related mechanism would be the caching of data. Considerable software development would be needed for these methods.

ACKNOWLEDGMENT

The development of concepts for the new PTDATA implementation has resulted from extensive discussions with P. Henline of General Atomics and with D. Butner, B. Meyer, and J. Moller of Lawrence Livermore National Laboratory.

REFERENCES

[1] P.A. Henline, "Use of open systems for control, analysis, and data acquisition of the DIII–D Tokamak," these proceedings.

[2] B.B. McHarg, Jr., "Real-time diagnostic data acquisition system for the Doublet III tokamak fusion experiment," in *Proc. IEEE Transactions on Nuclear Science*, vol. 30, p. 3787, 1983.

[3] A.S. Glad, P.. Henline, B. McHarg, D. Drobnis, S. Karin, and G. Shepard, "Acquisition, archiving, and analysis of Doublet III diagnostic data on a distributed computer system," in *Proc. 11th Symp. on Fusion Technology, Oxford*, Paper BP2-23, 1980.

[4] B.B. McHarg, Jr., "The use of a VAX cluster for the DIII–D data acquisition system," in *Proc. 14th Symp. on Fusion Engineering*, p. 852, 1991.

[5] B.B. McHarg, Jr., "Data file management in the DIII–D data acquisition and analysis computer systems," in *Proc. 13th Symp. on Fusion Engineering*, p. 185, 1989.

[6] B.B. McHarg, Jr., "The use of networking in the DIII–D data acquisition and analysis computer systems]," in *Proc. 12th Symp. on Fusion Engineering*, p. 636, 1987.

END

DATE FILMED

12/6/93