81-1-278
高エネ研図書室

CERN-Data Handling Division
DD/80/10
G.R. Hoffmann
J.G. Lee
March 1980

Support of File Editing using Networks

(submitted to Das Rechenzentrum)

DD-jt

SUPPORT OF FILE EDITING USING NETWORKS

Geerd-R. Hoffmann[1],
Gordon Lee[2]
Data Handling Division, CERN, Geneva.

1. INTRODUCTION

Over the past 5 years, the availability of cheap, good quality VDU terminals has meant that cards and remote input/output stations have been replaced by file editing/job submission systems. During the coming 5 years, it is very likely that this trend towards an increasing use of VDU terminals will continue. Many large scientific and university centres may have to support hundreds of active VDU terminals over the next few years. This is true, for example, of the computer centre at the European Organization for Nuclear Research (CERN), Geneva, which operates several large CDC and IBM machines.

In parallel with this development in terminal usage, there has been a general trend towards computer networking where mainframes are connected via high speed data links. Such links operate at effective end-to-end data transmission rates of at least 20 Kbytes per second. The CERN network, CERNET (see [2]), is an example of this.

In the following feasibility study, we investigate some possibilities for fulfilling such a demand for terminal support. In particular, we examine the way in which high speed packet switching networks could be used to relieve the I/O load created by file editing from large mainframe machines. The advantages and disadvantages of this approach will be examined.

---

[1] Geerd-Rüdiger Hoffmann, born in 1945, got his degree as Diplom-Mathematiker after studying mathematics, physics, statistics and philosophy at the university of Tübingen. He is Akad. Rat at the Zentrum für Datenverarbeitung der Universität Tübingen, currently on leave of absence for a fellowship in the Data Handling Division of CERN.

[2] John Gordon Lee, born in 1946, graduated in physics at Imperial College, London, in 1967. He then joined the High Energy Physics group at Imperial College and in 1973 was awarded a PhD in physics. Since 1973 he has been a member of staff in the Data Handling Division of CERN.

2. OUTLINE OF POSSIBLE STRATEGIES

2.1 CONSTRAINTS

In studying future file editing facilities, we chose to work with the following constraints:

1. The system must be capable of handling up to 300 simultaneous terminal users.

2. Any solution must take account of the existing structure of the computing facilities. In particular, one must allow for more than one large mainframe with its file base and almost certainly several smaller machines and their file bases.

3. One must ensure that there is a certain continuity at the user interface level. In particular, when an out-dated mainframe machine is to be replaced there is a need for maintaining some sort of compatibility at the editing level. One should not expect the users to familiarise themselves with a new editor and associated utility processors every time such a replacement occurs.

4. One must allow for future hardware developments, in particular in the area of VDU displays and "intelligent" terminals. The relatively long life of a VDU terminal inevitably means that there will be a variety of terminal types at an installation in spite of rationalisation attempts. One should therefore aim to support several types of terminal.

5. The popularity of an editor with the user community is very often related to the ease of use of the system. In a scientific environment, where most computer users are dealing with program source files, reliability and ease of use are more important than extreme sophistication. Often, a line number based editor is quite adequate for the application. Note however, that this does not imply that this should be the only mode of editing available. Ideally, one should aim to provide several interfaces to the editing system and let the user decide which of these best suits his experience and application.

2.2 A CENTRALISED SOLUTION

Adopting a centralised solution means expanding the central computing facilities to handle up to 300 active users. This would probably mean running several mainframes since 300 terminal users load the I/O capacity of a machine rather than the CPU and even with a very large machine, the I/O capacity is limited. Moreover, increasing the I/O power of large machines is often expensive compared to the price of a modern minicomputer. Furthermore, the reliability of the service depends on the stability of the central computer's hardware and software. Any errors or failures will affect all users, since there is rarely any redundancy. Software and hardware maintenance can thus become a great problem, especially when new features are to be implemented.

On the other hand, a centralised solution has the advantage that the editing service can be closely integrated with the particular operating system.

2.3 AN "INTELLIGENT TERMINAL" SOLUTION

Many powerful terminals are now available on the market which incorporate their own microcomputer and editing software. This type of terminal might in the future form the basis of editing systems. To provide fast access to mainframe file bases, however, terminals must be connected by means of high speed networks, because a simple transmission line of speeds up to 9.6 Kbits/second is too slow to take full advantage of the features offered by the terminal hardware. At the time being, the development of "bus" or "ring" networks necessary for a system based on "intelligent" terminals has not yet reached a stage where they are reliable and commercially available. Furthermore, it is not clear how one would support the "workfile", also called active file, of the editor. Since the size of the workfile can easily be about a Mbyte, it can be assumed that external storage will be needed. Another reason for assuming an external storage medium is the problem of recovery after system crashes. The available local floppy discs, however, are very limited in their capacity and in their speed at random access. Therefore, more sophisticated mass storage will have to be considered.

On the other hand, this strategy for file editing offers a system with a high level of redundancy in the event of hardware failures. It also has the advantage that the software development effort is almost certainly less than for the other options.

2.4 A DISTRIBUTED SOLUTION

The basic approach here is to off-load the work currently performed by editors running in central machines into a separate, dedicated machine. In this note, we shall refer to such a machine as a <u>File-Edit</u> machine. File-Edit machines would have only a limited local file base but would be able to access remote files on the central computers by means of a data communications network. The aim of this note is to examine this third solution in more detail and to see if such an approach can satisfy the needs of terminal users for several years to come.

3. USER INTERFACE FOR THE FILE-EDIT MACHINE

The File-Edit machine would provide a means of access to any file base which offers remote file access facilities using the communications network. As a by-product, it would also provide a simple means for the transfer of text files between network file bases.

The response time of the system for short commands (e.g. LIST, MODIFY, etc.) would not exceed 3 seconds.

As regards the human interface for the editor, it is intended that the File-Edit machine would provide editing at several user interface levels.

3.1 A LINE NUMBER ORIENTED EDITOR

This would be a line number based editor which, at the user level, would be similar to the WYLBUR[3] editor. It would provide the user community with a tool that is easy to learn and which is adequate for most file editing work. It should be noted that the emphasis is to provide a File-Edit/Job Submission system and not a text processing system. This is probably best left to other existing software products.

The line based editor would be the first stage to be implemented in any File-Edit machine project. Included in this first version would be a command file facility where commands are taken from a file rather than the terminal. Such "EXECUTE" files are used extensively in the WYLBUR editor at CERN.

3.2 FULL SCREEN EDITING

Full screen editing, which is more convenient for the entry and treatment of natural language text, is an essential part of any File-Edit machine system. It is also preferred by some people for the treatment of card-image program source since one always has a clear view of the context in which any changes in the file are made. The approach here would be to follow what has been done elsewhere (see e.g. [1] and [5]) and implement the full screen editing as a software layer above the line based editor.

3.3 INTELLIGENT TERMINALS

Terminals with substantial built-in processing power will become more commonplace in the next few years. They will therefore have a bearing on the way in which a File-Edit system is designed. Although detailed studies have not been

----

[3] Wylbur/370 Reference Manual. Stanford University. Nov 1975.

carried out, it is thought that the layered structure of the editor software will make the addition of such terminals to the system relatively easy. One should point out, however, that such terminals (sometimes with their own editor built in) are not intended as the main type of terminal to be supported by the File-Edit machine.

4. SPECIFICATIONS FOR A FILE-EDIT MACHINE

Limited network data transfer rates and the intended modularity of the File-Edit system determine an upper limit to the size of a single File-Edit machine. Experience at CERN[4] with the WYLBUR editor indicates that a 40 terminal File-Edit machine would create an average network load of 8 Kbytes per second (with peak rates of 25 Kbytes per second). In order to keep the requirements for the network transmission rates low, we will therefore assume that each machine will handle about 40 terminals. It is also assumed in this specification that the editor would be written in a high level language and that the machine hardware does not put any restrictions on the structure of the editor program. Preferably, one would implement the editor as a reentrant task. The configuration should therefore consist of the following:

4.1 <u>HARDWARE</u>

1. 40 terminal interfaces with line speeds up to 9.6 Kbits/second. File-Edit machines could be distributed geographically according to local requirements.

2. 1 high speed network interface.

3. 1 Mbyte of main memory excluding operating system and network interface requirements. This figure is estimated on the following assumptions:

   a) In the interests of rapid access to file records, large pieces of the active file should be able to reside in memory. Allowing 4 KBytes per user, implies about 160 KBytes of memory.

   b) Each connection to a data base host machine should consist of at least two links to provide high transfer speed and a low wait time for user files. The buffer size should be 8KByte, and each link should have 2 buffers available for double

---

[4] L. M. Robertson : Some Statistics on Disk Usage and Job Submission / Retrieval by WYLBUR Users. CERN Internal Document. SW-IBM Work Note 102. 1979.

buffering. Allowing for 4 remote data base machines, the memory required for links is 128 Kbytes.

c) The EDITOR program is believed to take about 512 KByte.

d) Terminal buffers (1.5 KByte per terminal) and terminal handler take about 168 KByte.

e) Queues, internal management etc. take another 12 KByte.

4. 60 MByte disk space accessible via at least two different access paths.
The amount of disk space is estimated as follows:

a) Allowing an average active file per user of 500 KBytes, (i.e. 6000 records of 80 characters) 40 users require about 20 Mbytes.

b) In order to provide fail safe recovery from system crashes, a LOG file will be kept to allow the active file to be reconstructed after a crash. For this feature, and other information necessary to ensure that, on system crashes, no user loses more than his last command, we allow 20 MBytes.

c) The remaining 20 MBytes are used for program development, operating system, storage of task modules, gathering of statistical data etc.

5. The time to read one block of 1 KByte from disk should not exceed 50 msec. This figure is calculated as follows:
If 40 users request one line of their active file which is currently not in core, 40 buffers have to be read. If the processing time for getting the display ready is in the order of 25 msec, then a READ time of 50 msec per block is necessary to give a response time of at most 3 seconds.

6. The most complicated EDIT command working on one block of data must not take longer than 25 msec. This implies that substring searching in 1 KB does not exceed this figure.

4.2 SOFTWARE

The operating system of the File-Edit machine should provide the following services:

1. An interface to a high speed computer network allowing remote file access, job submission, output file retrieval and some job status requests.

2. Interleaving I/O operations on disks.

3. Interleaving READ/WRITE operations for network data links.

4. An inter-task communication mechanism operating via shared memory (i.e. no copying of buffers).

5. Possibility to synchronize tasks via a "wait for event" mechanism.

6. High level language support for terminal I/O. This includes full duplex support of terminals and special key support.

7. After system crashes it must be possible to recover partially written files which were being used at the time of the crash.

8. Support of reentrant routines would be extremely advantageous. If this feature were available, there would be an editor task per terminal implying about 50 active tasks to be handled by the scheduler. Thus, task switching should not cause an excessive overhead. To switch from one task to another should not take more than about 0.1 msec.

9. A high level language which produces efficient machine code (e.g. PASCAL) should be available.

5. ADVANTAGES

The following are some advantages which emerge from a File-Edit machine system:

1. File editing becomes decoupled from other developments at the compter centre. Changes in the mainframe machines or the addition of new (or specialised) machines do not imply a change for file editing or the appearance of yet another editor. In this way, one is able to maintain a standard user interface for editing irrespective of where the files are actually stored.

2. The File-Edit machine can support many different terminal classes since its editor will provide several different interface levels. In this way, it is possible to support "intelligent" terminals as well as the many different types of ordinary displays that will inevitably be present at most installations.

3. The distribution of work between the File-Edit machine and the host file base machine could be

changed by implementing a more sophisticated host interface. This could go some way to reducing the data link traffic by allowing file fragments to be transferred in the case of very long files. This development would depend upon the result of studies of user behaviour from the statistical data gathered in the File-Edit machine.

4. Because the File-Edit machine will be programmed using a high level language, its maintenance becomes easier. Since modularity and layer structure are design objectives, adapting to new features in terminal hardware should not present major problems.

5. As a single File-Edit machine will only support up to 40 terminals multiple File-Edit machines should be envisaged. This provides for expansion of the system to accommodate more terminals in a modular manner. By only purchasing extra hardware as it is required, one can benefit from the rapidly falling cost of computer hardware.

   Assuming the data bases are at least partially distributed, the breakdown of one machine, either a main computer or a File-Edit machine, would not cause a complete breakdown in the user service. Since the File-Edit machine will be reasonably cheap (e.g. about 350K SFr), a stand-by machine could be purchased which will be used for development work if not needed for ordinary service.

6. DISADVANTAGES

The following gives a list of possible disadvantages of a File-Edit machine:

1. The network load created by the File-Edit machines may become unacceptably high since each machine will add an average traffic of about 8 KBytes per second with peak rates of 25 KBytes per second. This problem of network traffic is especially important in the case where most file editing commands are directed to one particular file base. However, future network developments or the addition of a high speed channel connection may provide a way round this limitation.

2. The remote file access software in the file base machines might become overloaded in the case where the file base is not distributed. For 7 File-Edit machines representing 280 active terminals, the file base machine has, in the worst case, to deal with

   a) 56 KByte per second average data transfer rate

b) 2 file open operations per second

c) 2 job status enquiries per second

3. Remote file access software must be available on all machines with data bases.

4. The performance of a fully loaded File-Edit machine, in terms of response times, would not be as good as the centralised solution for commands involving the transfer of a file from a file base to the File-Edit machine.

7. IMPLEMENTATION ESTIMATES

If a File-Edit machine solution were to be implemented, then a small implementation team of about 4 people should produce a first version of the system after about 18 months. The initial design and definition of the software components and their interfaces would occupy the team for the first 6 months of this period. Following the implementation period, the first File-Edit machine would be available to users. After a suggested 6 month running-in period, a second File-Edit machine could be introduced and, in the long term, editors on the host machines would only be used for specialised applications.

8. CONCLUSIONS

A File-Edit machine can be considered as a viable alternative for supporting large numbers of editing terminals. It offers a simple, modular structure which provides great flexibility in adapting to changes in the central computer system. This is important when there are likely to be significant hardware developments in the next few years. Furthermore, there is a general trend for mainframes to become specialised to certain applications (e.g. high speed arithmetic processors) whilst general purpose machines remain about their current size and one increases capacity by acquiring more machines and coupling them via a network. This is all compatible with the File-Edit machine solution since it is a natural superstructure to build above a data communication network.

It is therefore believed that File-Edit machines can meet the editing needs of large computer centres for several years into the future by offering a technically advanced, modular, and cost effective solution.

9. ACKNOWLEDGEMENTS

The authors would like to thank their colleagues of the Data Handling Division at CERN for many helpful discussions in the preparation of this paper. In particular, we would like to thank T. Bloch, L.M. Robertson, T.C. Streater and D. Wiegandt.

LITERATURE

[1] Fraser, C.W. : A Compact, Portable CRT-based Text Editor. Software - Practice and Experience, Vol 9(1979). pp. 121-125.

[2] Gerard, J.M. : The Development and Use of a High Speed Packet Switching Network in a High Energy Physics Laboratory. Proceedings EURO IFIP 1979. pp. 375-381.

[3] Hillman, A.L., Schofield, D. : EDIT - An interactive Network Service. Software - Practice and Experience, Vol. 7(1977). pp. 595-611.

[4] Ivie, E.L. : The Programmer's Workbench - A Machine for Software Development. Communications of the ACM, Vol 20, Number 10, October 1977.

[5] Macleod, Ian : Design and Implementation of a Display Oriented Text Editor. Software - Practice and Experience, Vol. 7(1977). pp. 771-778.