PROJECT MAC   July 1, 1975

Computer Systems Research Division   Request for Comments No. 83

S.M. Thesis Proposal: IMPROVING INFORMATION STORAGE RELIABILITY USING A DATA NETWORK

by Arthur J. Benjamin

Abstract:

Backup and recovery methods using magnetic tapes are common in computer utilities, since information stored on-line is subject to damage. The serial access nature of tapes severely restricts the flexibility and simplicity of accessing and managing the stored data. A method using a data network will be studied, to develop a backup mechanism which takes advantage of a large, inexpensive, random access remote data storage facility to provide data access and management functions that are more flexible than those provided by a traditional mechanism. Although data transfer rates will be reduced, data access and management will be simplified, and system availability will be improved. A working model will be built on the Multics computer utility, using the ARPA network.

This note is an informal working paper of the Project MAC Computer Systems Research Division. It should not be reproduced without the author's permission, and it should not be referenced in other publications.

Thesis Proposal
A. J. Benjamin

June 24, 1975
V. 2.4

Improving Information Storage Reliability

Using a Data Network

I. Background and Introduction

The diversity of system resources available on a computer network makes it attractive to distribute the user's needs for services between those hosts best suited to providing those services. Applications programs are available as resources at some sites (e.g. MATHLAB at MIT); but there is much more to share than just programs. Probably the most important use of the network is in allowing data to be shared among network users; if we view programs as a class of data, then this is all the more true. Also, the availability of specific computer system architectures has made it attractive to divide and distribute one's applications among those systems that seem most adaptable to the needs of the problem. For example, Multics provides protection features and a hierarchical file system, TENEX provides concurrent parallel processing (multi-tasking) within a user process, and the Datacomputer [1, 2, 3] provides reliable, economical, high volume storage.

There has been recent interest in developing a better understanding of how to share network resources in a cooperative way, and eliminate the isolated use of resources by making it easier to distribute subtasks among hosts while maintaining the ability to communicate, coordinate, and control results. The RSEXEC protocol is one attempt at providing a useable facility for controlled sharing of network resources. The National Software Works is another area of interest directed specifically to these issues.

A resource-sharing network needs a facility for managing distributed data. In general, this is a very large problem area, with no complete solution existing today. As a subset of the distributed data problem on network computers, I have been looking at data base management facilities and applications, and how they look from a network environment. In particular, I have been considering file systems as data base management applications. By restricting the problem to managing uniform use of differing file systems in the network, we can see some partial solutions, e.g. the RSEXEC file system on TENEX, and the FTP protocol. In order to substantially restrict the complexity of the problem, I wish to consider a specific application which has features germane to the general problems of managing distributed data, but which also has workable solutions.

I have chosen to consider the use of network resources to enhance the reliability of a file system by providing a backup facility. Many existing systems currently employ such a facility; however, there are features which a sharable network facility could provide which make this approach more attractive than the traditional methods utilizing locally accessible magnetic tape. Foremost among these is the increased flexibility for data storage and retrieval afforded by a random access network facility, contrasted with the more restricted serial access methods using magnetic tape. Although data transfer rates will be reduced, data access and management will be greatly simplified. Since the backup function in and of itself is important, this area will be studied from two complementary points of view.

First, as a facility useful to existing systems with on-line file storage systems, backup will be studied as a mechanism itself, to find out what such a mechanism might require of network resources, and what network backup can provide for the user. Secondly, the nature of a network implementation will be studied as a specific example of a restricted application of distributed data management. Since this research is just beginning, I will only raise some of the issues in these two areas, and provide my current thoughts as to why they are important.

The rest of this paper is presented in five sections. I will provide further motivation for studying information storage reliability, followed by a discussion of current solutions. Then I will consider the problem in a network environment and how this can be a fruitful approach, followed by a discussion of specific issues worthy of investigation. Finally, I will outline the research plan.

II. Motivation

Most large contemporary operating systems are too complex to understand sufficiently well to feel confident that they will operate correctly. One of the most complex (and important) components of an operating system is its file storage system. Since file system components interact in such a complex (and hence somewhat dangerous) manner, and since most hardware is subject to a non-zero failure rate, data stored on-line is subject to damage by both program and hardware errors. This results in a lack of absolute confidence in the reliability of data storage and motivates the need to safeguard against loss caused by either or both of the above mentioned possibilities for error.

The effort invested to supplement less-than-absolute reliability is determined by the cost of a system failure to the user. The method employed to supplement reliability is to provide redundancy of those system components whose chances of loss are to be minimized. In the case of computer systems, this generally means duplicating hardware and maintaining backup copies of data. In Multics, the ability to dynamically reconfigure the hardware allows a degree of on-line recovery from hardware failures, if defective subsystems can be replaced or removed, i.e. if there are redundant system components [4]. Similarly, an attempt is made to maintain redundant copies of all on-line data, so that lost or damaged data can be replaced in the event of a storage system malfunction [5]. The current backup system manages the redundant storage of data on sets of magnetic tapes. I propose that a data network (in particular the ARPA network) can provide simpler and more flexible facilities for implementing this function in a setting that has some interesting implications for computer utilities, such as Multics, which are part of such a network. I will discuss some of the issues in this paper.

III. Possible Approaches to More Reliable Data Storage

As already mentioned, reliability is usually enhanced through redundancy of system components. In addition to this approach, we will also consider a technique of fault tolerant computing, in which reliability may be improved by duplication of function (especially critical or frequently used functions) with different mechanisms. This is a generalization of the redundancy method, whereby duplicate system components need only preserve functionality, and not necessarily methodology. In fact, by using different methods, it is felt that complete recovery from malfunctions will be more likely, and hence system operation will be more reliable, in the sense that there will be fewer malfunctions which could leave the system in a state from which recovery is impossible. A combination of the different methods to be discussed in this section would provide this functional redundancy.

There are several practical considerations in designing a system that provides redundant data storage. Two usual conflicting measures of the system's value are its operating cost (demands on system resources) vs. its sophistication (ease of use, versatility). Most systems utilize magnetic tape as a storage medium because of its low cost and high storage capacity and channel bandwidth for reading and writing data. Some systems may use random access secondary storage devices to a lesser extent, because this is usually more expensive, but very few systems use remotely located large, shared, inexpensive storage facilities available in a data network. The latter method is the major subject of this research.

The essential function of the backup mechanism is to store copies of data for possible retrieval at a later time. However, the mechanism for doing this and the flexibility for performing useful related functions depend on characteristics of the backup device, some of which will be discussed next. The goal is to implement a virtual backup device, using a data network, that provides a flexible mechanism for managing data storage and retrieval.

One of the most important characteristics of the backup device is the way in which the data it stores can be accessed. Sequential access is a severe restriction inherent in tape systems, since it confines the order in which data items can be accessed or else access becomes time consuming. Given the availability of a large, cheap, reliable storage medium (e.g. the Datacomputer), a network-implemented backup device should allow random access to data, and remove this restriction. However, limited bandwidth for data transfers in the network will introduce new constraints. A local disk system would also allow random access to data, but would probably be much more expensive than a large sharable network-wide backup resource. A tape or disk system could be used to supplement a network implementation, increasing reliability through functional redundancy, as described earlier. Since the network seems to offer the most interesting set of possibilities, it is selected as the basis for the virtual backup device. The implications on the design of the backup facility and the issues of distributed data management that are confronted are discussed in the next two sections.

IV. Backup in a Network Environment

Traditional backup facilities operate by copying on-line storage onto magnetic tapes. This requires a tape management system to map file system entities into tape images, and vice-versa. The sequential access nature of the tape medium results in a hierarchical file system being collapsed into a linear list of data blocks, each list corresponding to some file of the on-line file storage system, along with identifying information (e.g. pathname, status, etc.). Recent changes to files are recorded periodically as new copies of on-line storage are made. Old copies are usually retained for a period of time, so that even if on-line files are intentionally deleted, backup tape copies may still be available. Some of the problems with such a tape system arise from the sequential organization of the data, and the resulting difficulty in accessing it efficiently. On the other hand, a large, shared, inexpensive data storage utility available on the network (viz. the Datacomputer), could be used to implement a backup device. Availability of this facility as a network resource would make it easier and more economical to build a more versatile backup system. Access to data could be random, and a more appropriate data organization could be designed to more closely parallel the on-line file system organization. Backup functions could be automated, so that no operator action would be needed in handling tapes. A more useful user interface to the backup system could also be constructed to allow more direct control over backup functions, such as prescribing when a consistent copy of a file or set of files can be made, or specifying how many older versions of a file should be retained in the backup system.

By considering the backup facility as an extension to the file storage system, useful for improving reliability, a more general file system emerges. The network provides the mechanism for studying this generalization, which entails some changes to current backup methods. First, we would like to retain advantageous features currently available in the tape backup system. These include: 1) the side effect of saving copies of several versions of a segment so that even if an old version is deleted from on-line storage, a copy may still be available on backup tapes; 2) it is possible to keep abreast of changes to on-line storage by recording them on tape as soon as is feasible, and we do not want to degrade this reliability parameter when using network facilities; 3) finally, the safeguarding of backup data is accomplished by physical controls over the tapes themselves, which may be subject to sparse errors but probably not catastrophic ones, and we do not want a network implementation to degrade the degree of either physical security or reliability currently available.

In addition, a network backup facility offers advantages not present in a tape facility. A less elaborate bookkeeping function will probably suffice for keeping track of backup data, perhaps stored among several Datacomputers on the network. Essentially, I want to create what functionally appears to be an image of the portion of the on-line file storage hierarchy which is to be backed up. To accomplish this, there needs to be a mapping from on-line storage to backup copy, as well as an inverse mapping from backup copy to on-line storage, and these can be simpler to handle with a random access network facility then with a serial access tape facility, because information describing stored data can be accessed independently of the stored data itself. The accessibility of a network facility not requiring human aid in performing any of its functions also makes it easier to automate backup, and to more easily provide interfaces for user control, as already mentioned.

V. Issues and Ideas

A goal of work to be done is to gain better insight and understanding of a restricted application of network distributed file systems. Several issues and suggestions associated with a network implementation of backup will be discussed here.

Since the bandwidth of a network connection is substantially lower than the bandwidth of a tape channel, a serious bottleneck could occur, for example, if all of on-line storage needed to be retrieved quickly from backup storage. However, if we consider the backup facility as a slower (but still random access) level in the memory hierarchy (with special properties, to be described later), we can think of the data management function in terms already familiar in multi-level hierarchical memories. For example, the restoring of on-line storage could proceed via two (or more) simultaneous connections, one operating in an autonomous manner at a rate that depends on system load, and another operating in response to user demand, but invisible to his process, just as demand paging is hidden from the user. This latter mechanism would be invoked in response to a "segment not stored here" fault, and would implement an extended attribute of the file system; the former mechanism is not essential, but is useful from a performance viewpoint.

A slightly modified view of file system organization helps support this extension. The structure of the storage system would be implicit in the mapping between names known via the on-line storage system, and backup copy data, and corresponds to the directory hierarchy in Multics. The major backup function then is concerned with managing segments that are terminal nodes in the hierarchy, i.e. the "real data", and subsidiary descriptive information (e.g. bit count, dates, names, locations in the hierarchy, etc.) can be considered part of the mapping, which would also be stored on the backup device. After part or all of on-line storage is damaged, the salvager function amputates injured and irreparable subtrees, restores all information not in terminal nodes, i.e. restores from the backup device a skeleton of on-line storage for the injured subtrees, and leaves restoration of actual data segments (terminal nodes) to the background or demand processes. Damage to the file system need not reduce system availability. This can be viewed as a dynamic reconfiguration of injured parts of the storage system, using the network as the redundant backup device, but in a way that repairs or heals the original injury without risking damage to the backup information. In other words, access to the backup device is once-only and by a different mechanism than is used to access on-line storage, but the functionality of data access as perceived by the user would be identical. (1)

Other issues relate to restoring or reconfiguring of the file system. For example, what happens if the backup device is unavailable, due to problems with the network or at a remote Datacomputer? How does

(1) However, access time would probably increase.

recovery relate to the backup mechanisms? For example, will it still be useful to keep a bit-by-bit physical copy of the on-line storage devices? Perhaps a hybrid mechanism can be used, whereby all but terminal nodes are stored locally (e.g. on tapes), and terminal nodes are stored through the network. I do not like this approach, but perhaps some application of the current backup mechanism can be used to duplicate a function of a network facility by a different mechanism, to add additional reliability.

It will be easier to allow a user more direct control over backup functions, since these will be automatic and independent operations. For example, he can control the frequency of dumping, or the number of backup versions to keep (multiple versions of data is one of the special properties of the backup copy of the on-line file system). The issues of how a user might want to control these functions, what control he should be given, and what is an appropriate interface, will be studied later. The user implementation of backup and retrieval policies is an issue for further research.

Another problem arises in connection with data security, in controlling access to data stored on the backup device. Adaptations of current techniques in access control will probably be sufficient to solve this problem. Which techniques are best and how to adapt them in an actual implementation will be studied in the course of the research.

Finally, there are some fundamental issues that are applicable to the general distributed data problem. It seems possible to formulate and test a small number of mechanisms upon which the network backup system, as a subset of the general problem, can be based, which can also support other (related) facilities. For example, mapping names of local objects into network objects is necessary in the backup system; in addition, it would be nice if the mechanism for doing this would also work for building an extended network-wide file system. By solving a smaller problem, we can at least provide a test case example of some of the issues a general solution to the distributed data problem will have to resolve.

VI. Research Plan

The proposed research will be conducted in four phases. The first, which has been in progress for some time, consists of a background study to discover current issues in distributed data management, computer system reliability, and distributed file systems, backup, and retrieval, in particular.

The second phase, which is just beginning, will be to design the virtual backup device and an appropriate interface that is general enough so that its primitives will be useful to consider as a special case of a more general distributed file system. Also to be designed is an operational plan for performing the backup and retrieval functions. This phase should take six weeks to complete.

The third phase, which will partly occur in parallel with the second, will be to implement an ARPA network working model on Multics, based on the proposed design. Since design and implementation are often mutually dependent, it is likely that some iteration will occur between phases two and three before a stable design/model results. This should be achieved in ten weeks.

The final phase will be to analyze the results of phases two and three, and to assess their significance. The thesis will be completed during this phase, with a target completion date of January, 1976.

Bibliography

1. Computer Corporation of America, Datacomputer Version 0/11 User Manual, Datacomputer Project Working Paper No. 10; December 1, 1974, Cambridge, Ma.

2. Computer Corporation of America, Further Datalanguage Design Concepts, Datacomputer Project Working Paper No. 8; December 15, 1973, Cambridge, Ma.

3. Marill, Thomas, and Dale Stern, The Datacomputer: A Network Data Utility, Computer Corporation of America; January 7, 1975, Cambridge, Ma.

4. Schell, Roger R., Dynamic Reconfiguration in a Modular Computer System, Project MAC Technical Report No. 86; June 1971, Cambridge, Ma.

5. Stern, Jerry A., Backup and Recovery of On-Line Information in a Computer Utility, Project MAC Technical Report No. 116; January 1974, Cambridge, Ma.