AD-A045 623 COMPUTER CORP OF AMERICA CAMBRIDGE MASS F/G 9/2
VERY LARGE DATABASES.(U)
AUG 77 R H DORIN, J Z SATTLEY N00014-76-C-0991
UNCLASSIFIED CCA-77-10 NL

1 OF 2
AD
A045623

A045

1.0

1.1

1.25

1.4

1.6

MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

Very Large Databases: Final Technical Report

Technical Report
CCA-77-10
August 30, 1977

Robert H. Dorin
Joanne Z. Sattley

Computer Corporation of America
575 Technology Square
Cambridge, Massachusetts 02139

Very Large Databases.
Final Technical Report. 1 Jul 76 - 30 Jun 77,

July 1, 1976 to June 30, 1977

30 Aug 77

95p.

Robert H. Dorin
Joanne Z. Sattley

Technical Report
CCA-77-10

387 285

This research was supported by the Defense Advanced Research Projects Agency of the Department of Defense and was monitored by the Office of Naval Research under Contract No. N00014-76-C-0991. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

Abstract

This report summarizes work performed by Computer Corporation of America under the ARPA Very Large Databases program. The report discusses two separate and quite disjointed tasks.

Section 1, written by Robert H. Dorin, describes the Datacomputer Technology Transfer project (DTT). This project was aimed at expansion of the Datacomputer user community by increasing the awareness of this unique tool in the government community. The plan for the project involved a series of general publicity efforts, presentations at potential user sites, and extensive technical support of the beginning steps of new users. The result of this project was a substantial increase in Datacomputer utilization, to wit:

- 180% increase in total storage used;
- 181% increase in monthly processor utilization; and
- 171% increase in the number of active system users.

Section 1 of this report discusses the activities performed under the DTT project and elaborates on its results.

Section 2, written by Joanne Z. Sattley, describes the Message Archiving and Retrieval System (MARS) project. The MARS effort was directed toward the design and implementation of a prototype system to provide economical storage and convenient retrieval of Arpanet mail. The system has been fully implemented and is operational at CCA on an experimental basis. This report describes the MARS prototype as well as concepts for future extensions of the system.

Table of Contents

1. Datacomputer Technology Transfer Project

1.1 Introduction

This section describes the activities and results of the Datacomputer Technology Transfer (DTT) project of the VLDB program. The goal of this project was the expansion of the user community of the Datacomputer, a network data utility developed by Computer Corporation of America (CCA), and sponsored by the Defense Advanced Research Projects Agency under contracts MDA903-74-C-0225 and MDA903-74-C-0227.

Section 1.2 of the report, "Project Activities", describes the tasks performed under the project. Section 1.3 discusses the key result of the project: a substantial increase in the level of system utilization. The balance of this section briefly describes the Datacomputer and discusses the plan of attack for the DTT project.

1.1. The Datacomputer

The Datacomputer is a complete hardware/software data handling system designed to support:

1. data sharing among heterogeneous computers in a network environment and

2. economic storage through use of a mass storage system.

The Datacomputer has been offering service to computers on the Arpanet since April, 1973. The first complete, working version of the system was available in August, 1975, and an Ampex Terabit Memory (TBM) system was installed in mid-1976 making the Datacomputer the first data management system to offer convenient, ready-to-use access to a mass memory. Version 2 of the Datacomputer which included a TBM system with an on-line capacity of 200 billion bits (about 60 times that of the Version 1 system) has been available to Arpanet users since October, 1976. More details of the system configuration and its extensive database management facilities are available in working papers and technical reports from CCA [CCA-a,CCA-b,CCA-c].

1.1.2 Plan of the DTT Project

The intent of the DTT project was to expand the user community of the Datacomputer by increasing the level of awareness of the system's capabilities among potential users. The plan for achieving this objective involved the following three steps:

1. Identify and make initial contact with potential users. This step involved the wide distribution of capsule Datacomputer information through mailings, conference presentations, and articles. The reader response to this activity identified a significant number of user prospects.

2. Present more detailed information and discuss applications with identified potential users. This activity was allocated much of the effort under the DTT project. It involved site visits, overview presentations, and day-long training seminars.

3. Support and advise new users. When organizations identified through steps 1 and 2 begin to use the Datacomputer, their efforts are supported by the Datacomputer staff under a separate contract: MDA903-77-C-0183.

1.2 Project Activities

This subsection describes the tasks performed under the DTT project. It deals with steps 1 and 2 of the project plan:

1. identifying potential users and

2. providing them with detailed system information.

1.2.1 Initial Identification and Contact

Many of the users in the Arpanet community were either unaware of the Datacomputer and its availability, or somewhat misinformed about the system. Some knew of it as a storage resource but did not know about the system's extensive software facilities. Others were aware of its existence but unaware of its potential availability for their use. This step of the project was intended to increase general awareness so that organizations with potential applications would at least know enough to inquire further.

The techniques used to achieve this broader awareness were: a mass mailing (section 1.2.1.1), presentations at conferences (section 1.2.1.2), and a trade publication article (section 1.2.1.3).

1.2.1.1 Mailing to Arpanet Community

A letter describing the Datacomputer was mailed, along with an overview article [MARILL and STERN] and a reply card, to Arpanet users in the U.S. Government, particularly within the Department of Defense. This mailing effort produced a considerable response including inquiries from the following organizations.

- Defense Communications Agency
- Air Force Systems Command (Andrews AFB)
- Defense Mapping Agency
- Rome Air Development Center
- National Library of Medicine
- National Bureau of Standards
- U.S. Army CERL
- Rock Island Arsenal

- Tooele Army Depot
- Department of Commerce
- Naval Coastal Systems Laboratory
- Naval Ship Engineering Center
- NASA Ames Research Center
- Air Force Logistics Command
- Naval Surface Weapons Center
- U.S. ARRCOM

Those who responded were sent additional documentation, and were telephoned to discuss applications and arrange a presentation.

1.2.1.2 Presentations at Conferences

CCA was invited to participate in the Third IEEE-CS sponsored Workshop on Mass Storage Systems in Palo Alto, California on April 5-6, 1977. A variation of the standard Datacomputer presentation (see section 1.2.2.1) was delivered in a session consisting of users of mass memory devices. CCA also participated in a panel discussion on the use of mass storage to support distributed processing.

This conference yielded additional organizations with potential Datacomputer applications including the U.S. Army Engineering Topographic Laboratory.

CCA was invited to submit a paper to the IEEE International Symposium on Computer Aided Seismic Analysis and Discrimination on its work in support of a large seismic database. A paper entitled "Use of the Datacomputer in the Vela Seismological Network" was written, and presented to this conference of seismologists on June 9, 1977 in Falmouth, Massachusetts. A copy of this paper is enclosed as Appendix A.

This conference yielded two additional contacts: the National Science Foundation and the U.S. Geological Survey.

1.2.1.3 Articles in Trade Publications

A publicity article describing the Datacomputer was written, and arrangements were made for it to appear in the Computerworld weekly newspaper. A copy of this article from the May 9, 1977 issue is included as Appendix B. Inquiries for more information were made from 20 organizations. Discussions are underway to have a similar article printed in the Government Data Systems publication.

1.2.2 Presentation of Datacomputer Technology

Once prospective users were identified, it was important to make them aware of the technology of the Datacomputer, particularly the benefits provided by the system. In addition to the hundreds of copies of articles, manuals, and working papers which were requested by representatives from government and industrial organizations, the DTT project sought to meet in person in order to discuss the Datacomputer and its potential use. To prepare for these meetings, an overview presentation (40 minutes) and a training seminar (1 day) were developed. These are discussed below.

1.2.2.1 Datacomputer Presentations

A 40 minute presentation with color slides was prepared. The text and slides of this talk appear in Appendices C and D. The talk was designed for application managers, programmers and researchers. The presentation emphasized features of the system which are unique and which characterize those applications most likely to enjoy substantial benefits from use of the system.

The Datacomputer talk was delivered at several government installations:

- Defense Mapping Agency Topographic Center
- Defense Communications Engineering Center, DCA
- National Bureau of Standards
- Rome Air Development Center

as well as other organizations:

- University of Maryland, Department of Information Systems Management
- Rand Corporation
- Digital Equipment Corporation, Federal Systems Group

On several occasions, visits to CCA were arranged for contacts either local to or passing through the Boston area. In addition to the Datacomputer presentation, a tour of the system installation was given. Representatives who visited CCA during the project came from:

- the Rand Corporation
- the Department of Transportation
- MIT Nuclear Engineering Department
- the Israeli Armed Forces
- The Analytical Sciences Corporation
- a research group from Technical University in Braunschweig, Germany

1.2.2.2 Seminars

Some Datacomputer prospects were familiar with the concepts of the system, and wanted to learn more about the system and its use. A full-day training seminar was prepared for such prospects, and two seminars were held during the DTT project. They are described below.

During 1976, CCA had been in touch with several laboratories of the Energy Research and Development Administration concerning use of the Datacomputer. To further these discussions a full-day seminar was held at CCA in Cambridge, Massachusetts on January 21, 1977. Detailed information about the Datacomputer was presented and specific database applications within ERDA were discussed. Invitations were extended to several ERDA labs, and representatives attended from:

- Lawrence Berkeley Laboratory
- Brookhaven National Laboratory
- Argonne National Laboratory

- MIT's Laboratory for Nuclear Science
- ERDA Headquarters in Washington, D.C.
- ERDA research group at UCLA
- ERDA research group at NYU

The agenda consisted of functional and architectural overviews of the Datacomputer, a tour and demo at the CCA installation, a tutorial on Datalanguage, and a discussion of ERDA applications. One large Datacomputer application, involving a weather database, was planned at this seminar, and will be discussed in section 1.3.3.

Many Arpanet users are located in California so another full-day seminar was arranged on July 12, 1977 at the Stanford Research Institute with the help of Ms. Elizabeth Feinler of the Network Information Center. The agenda was quite similar to the ERDA seminar. Several existing Datacomputer applications were discussed rather than specific applications of those in attendance. Time was taken after the seminar and during the breaks to discuss the applications of some members of the audience.

The audience of 45 included representatives from:

- Stanford Research Institute

- Stanford University
- Lawrence Berkeley Laboratory
- Lawrence Livermore Laboratory
- NASA Ames Research Center
- Xerox Palo Alto Research Center

Several of these contacts are currently being followed up.

1.3 Datacomputer Usage

Overall system utilization increased significantly during the first six months of 1977. Much of this increase resulted from the greater awareness of the Datacomputer within the Arpanet community.

This increased utilization has occurred within the relatively short duration of the DTT project. This was possible because many new users were able to avail themselves of Datacomputer software which was already available and well suited to their task. Others required new software development. These latter applications are now in the development stage. While they did not have an immediate impact on system usage, over the long run they will add substantially to the user community.

Following a discussion of overall system usage, examples of short and long term applications will be presented.

1.3.1 System Utilization

Three factors were chosen to measure the Datacomputer utilization.

1. Storage - This is a significant indicator of utilization of a database system. Furthermore since the Datacomputer is a remote facility, the storage utilization represents very directly a burden which has been removed from the user's local resources.

2. CPU time - Since the data management facilities of the Datacomputer include sequential searching, indexed access and computational features, processor utilization is a relevant measure of system work.

3. Connect time - This measure represents a rough gauge on user activity.

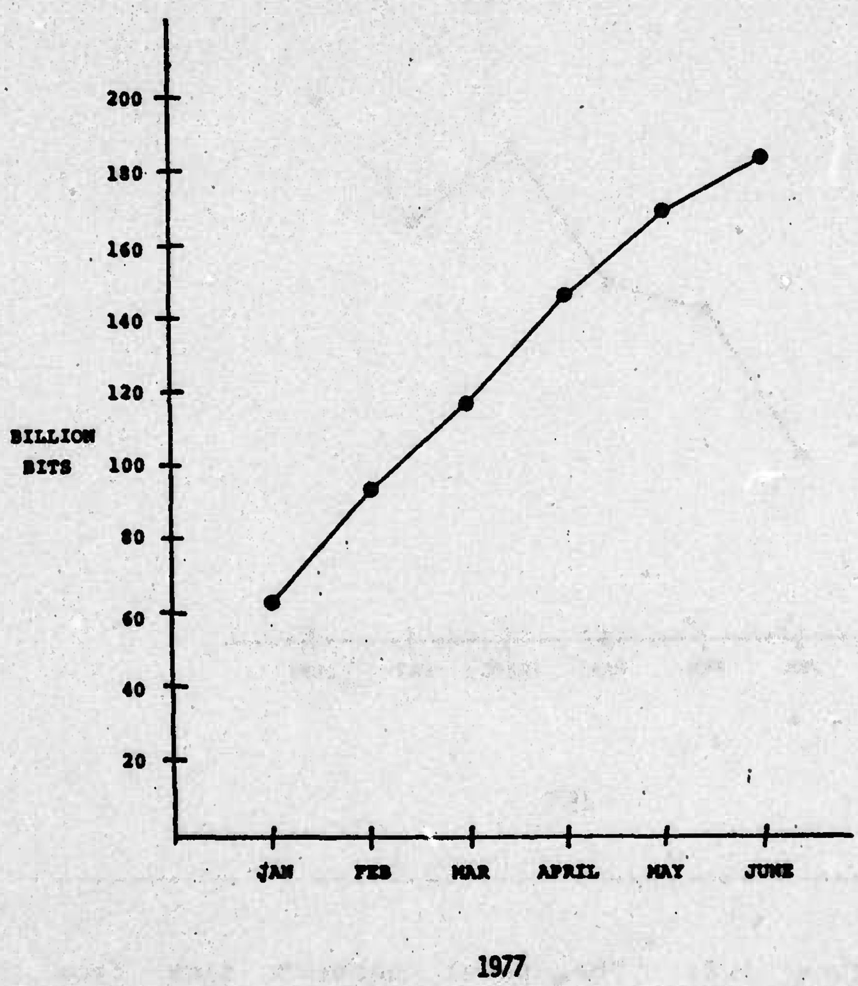
All three of these factors showed substantial increases over the measured period. The storage utilization in the Datacomputer increased by 180% from 66 billion bits in

January to 185 billion bits in June. This is the equivalent of approximately 230 3330-type disk packs. The

---

Storage Utilization

Figure 1.1



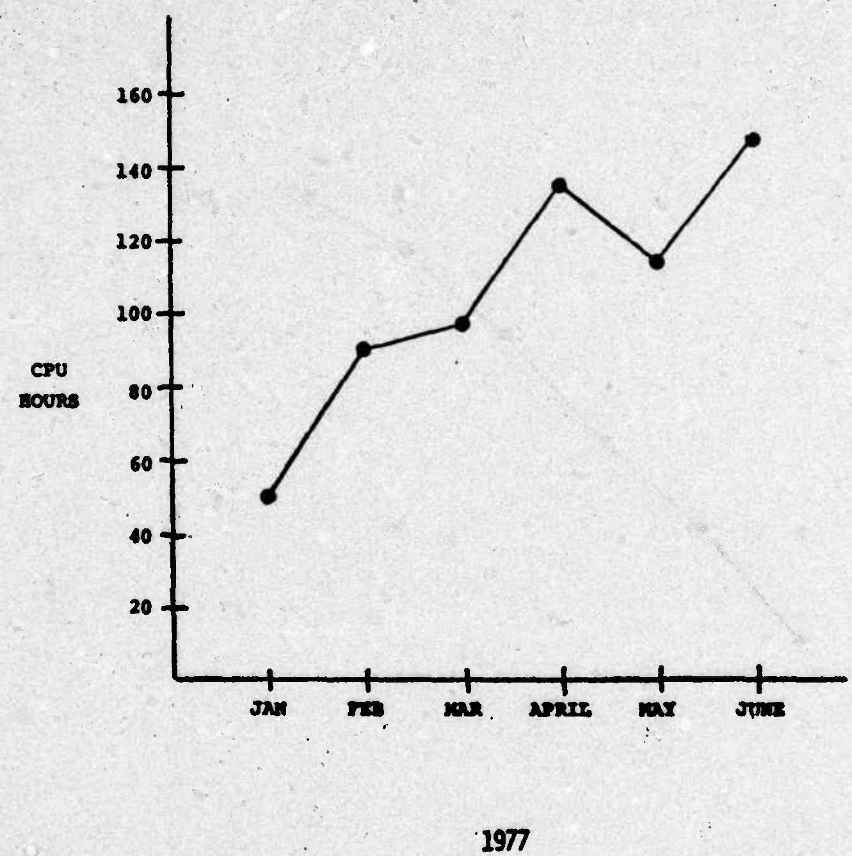
---

monthly storage charges are shown in Figure 1.1.

The central processing utilization increased by 181% from 51.9 CPU hours in January to 146.1 CPU hours in June, and

---

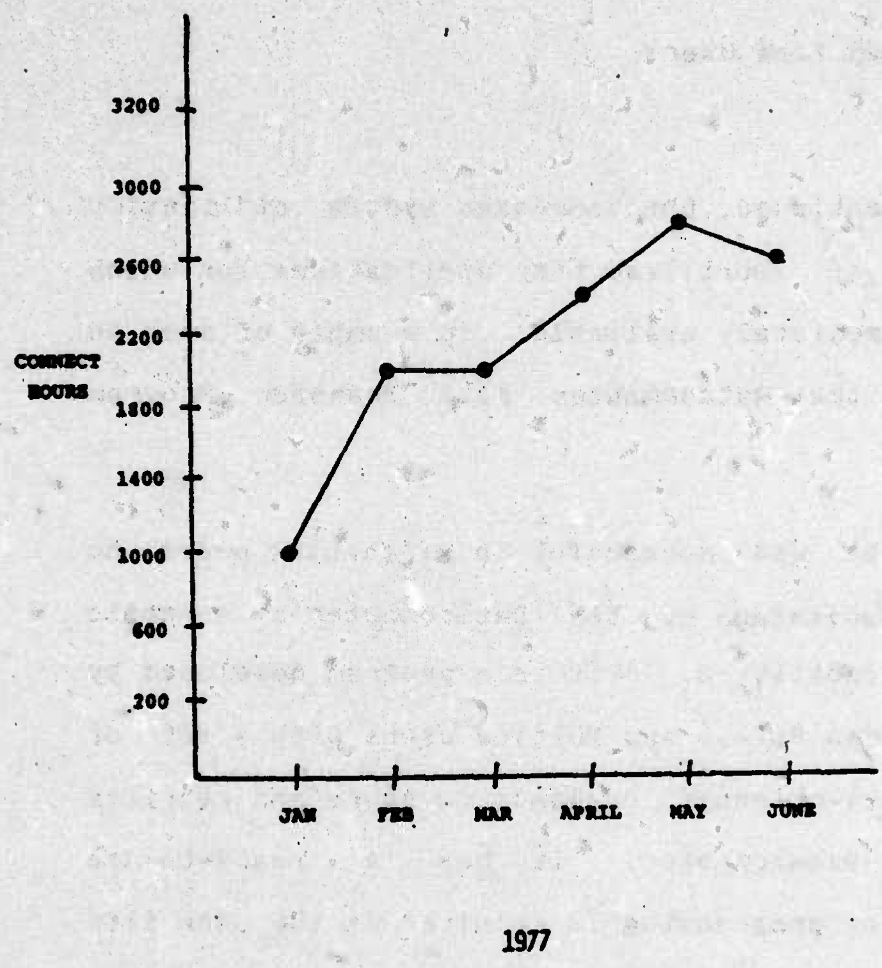
Processor Utilization   Figure 1.2



---

is shown in Figure 1.2. The total connect time from remote users to the Datacomputer increased by 160% from 1025 connect hours in January to 2661 connect hours in June, having reached a high water mark of 2892 hours in May. This utilization increase is shown in Figure 1.3.

Total Connect Time Figure 1.3



1.3.2 Short Lead Time Users

As previously mentioned, the increased system utilization was a result of short lead time applications for which software was immediately available. An example of such an application is the Datacomputer File Transfer Program (DFTP).

The DTT project was successful in soliciting users who wanted to take advantage of the Datacomputer's economic file storage facilities. DFTP is a program developed by CCA which provides PDP-10 and Multics users with a set of simple, terminal-oriented commands to store and retrieve files on the Datacomputer. It has a ready-to-use interface, and no programming is required at the user site (for PDP-10 and Multics systems). This is a big factor in the short term success of DFTP.

DFTP translates the user's commands into Datalanguage, the interface language of the Datacomputer, and handles all data transfers, network connections, and local file I/O operations. DFTP notifies the user of the results of each command.

Six new sites began using DFTP during the DTT project bringing the total number of user sites to 20. These new DFTP users were:

- Rome Air Development Center (Griffiss AFB)
- Carnegie-Mellon University
- the Air Force Armament Development and Test Center (Eglin AFB)
- the Packet Satellite/Speech project at ARPA
- Air Force Avionics Laboratory (Wright-Patterson AFB)
- National Bureau of Standards

The number of distinct DFTP user sessions increased from 237 in January to 835 in June. This 252% increase in use reflected the popularity of DFTP. The number of active users jumped from 42 to 72 in this period. Though DFTP was originally implemented for Tenex systems, it is now operational on several other PDP-10 operating systems including TOPS-10, TOPS-20, ITS and SAIL, as well as on the Multics system.

During the DTT project, interest in a DFTP-like facility was expressed by several Arpanet users running the Unix system on PDP-11's. These included the UCLA-Security research group, Lawrence Livermore Laboratory, Lincoln

Laboratories Applied Seismology Group and the Rand Corporation. In conjunction with the Datacomputer presentation at the Rand Corporation, a meeting was held to arrange a joint effort to develop a DFTP-like facility for Unix systems. This effort has begun, and the facility is expected to be available to Unix users in the fourth quarter, 1977.

1.3.3 Long Lead Time Users

Some applications, which are well-suited for the Datacomputer, will nonetheless involve a long lead time. Seismological and meteorological databases, which have high input rates, may involve the gathering and subsequent analysis of data at dispersed geographic locations. A data utility supporting network access and a mass memory system is ideal for these applications. The process of planning and designing such an application, though, is a fairly long term effort.

During the fourth quarter, 1976, a group at the Division of Energy Conservation at Argonne National Laboratory (ANL) began to experiment with the Datacomputer. This group was planning to build a prototype weather database on the Datacomputer for use by the CAL-ERDA Building

Energy Analysis Project. CAL-ERDA is a family of computer programs, currently under development, which is used for analysis of energy usage in buildings. It is a cooperative effort of ANL, Lawrence Berkeley Laboratory (LBL), Los Alamos Scientific Laboratory (LASL) and the U.S. Army Construction Engineering Research Laboratory (CERL). The weather data to be stored on the Datacomputer would be retrieved by several laboratories.

This application is particularly appropriate for the Datacomputer because it involves both sharing data at widely dispersed sites and the storage of a very large database. However, the complexity of the application leads to a longer start-up period than simple DFTP-like uses. CCA has been actively involved in this start-up effort.

CCA arranged for the group from ANL to meet with a representative from National Oceanographic and Atmospheric Administration (NOAA) in conjunction with the ERDA seminar in January, 1977. NOAA has access to a large amount of weather and solar radiation data (totalling more than 10 billion bits) from the National Climatic Center in Asheville, North Carolina, and will make this available for the CAL-ERDA application. Since this meeting, CCA, ANL, and NOAA have been working on the design of this database. It is expected that this database will be used in several ERDA and NOAA projects, including CAL-ERDA and ATMES (Advanced Technology Mix Energy Systems).

This weather database effort has been underway for several months, and continues to make progress. Such applications will bring a steady and substantial increase in system utilization over the long term.

1.4 Summary

The goal of the Technology Transfer Project was to increase the Datacomputer user community by presenting the benefits available to prospects. The Datacomputer was presented in several different forums, and a great deal of interest was generated. Those users who indicated a desire to use the system were given the assistance they needed, and the overall system utilization increased substantially. CCA will continue to maintain contact with all of the active prospects identified by the DTT project, and the system utilization should increase further as planned projects begin implementation.

2. Message Archiving and Retrieval System

2.1 Introduction

Under this task CCA designed and implemented a prototype version of MARS (Message Archiving and Retrieval System), a system which provides economical storage and convenient retrieval of Arpanet messages.

MARS achieves inexpensive storage through use of the Datacomputer [MARILL and STERN], a network database utility developed by CCA under ARPA contracts, MDA903-74-C-0225 and MDA903-74-C-0227. The Datacomputer offers on-line storage at a cost which is 2 orders of magnitude less than other on-line alternatives.

Messages archived using MARS are heavily indexed and can be retrieved in a variety of ways including Boolean combinations of message recipients, message date and time, any text words in the message subject, and text words in the message body. The MARS facilities are integrated very naturally into the existing collection of message handling tools:

- A message is designated for archiving by sending it to the MARS-filer using one of the usual message mailing tools such as SNDMSG.
- A message is designated for retrieval by sending a request as ordinary mail to the MARS-RETRIEVER.

The prototype MARS has been fully implemented and placed in service at CCA. It has proven to be an effective and popular tool in this community of mail users.

In this section we discuss both the current implementation of MARS and some general archiving concepts which suggest extensions of MARS for the future:

Subsection 2.2 - "Project Overview" presents the general goals of the implementation and its system architecture.

Subsection 2.3 - "System Description" provides details of the implemented prototype.

Subsection 2.4 - "Future Directions" discusses a collection of improvements and extensions appropriate for a full scale implementation of MARS.

2.2 MARS Project Overview

Electronic mail, as typified by the Arpanet mail services, is becoming an important mode of inter-personal and inter-organizational communication. In the Arpanet community, thousands of individuals and hundreds of organizations use these services routinely for the exchange of messages, in much the same way that they would have used telephones or paper mail in the past. In many situations the electronic mail systems are preferable to these older alternatives because:

1. They are much faster than paper mail.

2. They are often cheaper than telephones.

3. They are often more convenient than telephones because they do not require that both parties be available simultaneously.

4. They create a (potentially) permanent written record of the correspondence.

The model of voluminous and enthusiastic use of electronic mail which has been established in the Arpanet community seems likely to be repeated in both civilian and military organizations outside of Arpanet.

If electronic mail is to fulfill its potential for impact on human communication, it is essential that effective tools exist for handling messages throughout their entire life-cycle: from composition, through transmission, through receipt and reading, and finally to archiving. In the Arpanet mail systems today, effective tools are available for composition, transmision, receipt and reading. However, there does not exist a complete, effective and economical mechanism for archiving mail. The MARS project has focussed on the implementation of a prototype of such an archiving facility. In this section we present the design goals of the prototype and its general architecture.

The MARS prototype was designed to meet the following objectives:

1. Storage economy - In a large community of electronic mail users, such as the Arpanet community, the volume of a message archive can grow very large indeed. One active user of an Arpanet mail facility whom we studied has received or sent

25,000 bits of message traffic per day over the last two years. While this user does not represent a random sample in any sense, we felt that his behavior was typical of active users. If there were 1000 such users, 11 IBM-3330 type disk-packs would be required to hold 1 year of their message traffic. At typical on-line storage rates this archiving activity would cost about $350,000. One clear objective of a reasonable archiving facility is to lower this very high element of the system cost.

2. Retrieval quality - The raison-d'etre of a message archiving facility is the requirement to retrieve messages from time to time. Because the total message archive will be very large and the time frame over which messages are held will be fairly long (e.g. several years), it will often be difficult to find a particular message in the archive. This will be especially true for the frequent case in which the message can only be vaguely identified, for example by a specification like "A message sent to Walker early in 1977 regarding complete security." In today's military message archiving schemes a message can only be

retrieved if the message identification number is known. Even in these cases the retrieval usually requires days to complete. This kind of service exhibits a low retrieval quality. A key design objective for MARS was to achieve flexible retrieval requests which exhibit good precision and recall and which can be handled in a timely fashion.

3. Ease of use - As is usual for systems oriented toward end user functions, it is important that MARS be easy to use. Since MARS operates in an environment where there are already well established user tools and procedures, this requirement translates into a need for a clear and natural integration of the MARS facilities into the existing environment.

The MARS prototype meets all of these objectives:

1. Storage economy is achieved through use of the Datacomputer [MARILL and STERN], a network database utility developed by CCA for ARPA. The Datacomputer employs a mass memory system called the Ampex Tera-bit Memory (TBM) which can economically store millions of messages per year.

The use of TBM Mass Memory for storage of messages offers dramatic savings in storage cost compared to alternative on-line devices. For example, a single storage component of a TBM Mass Memory can store as much data as 100 3330-type disks [GREEN], [WILDMANN]. The TBM component cost is $100,000, whereas the disk units would cost more than $2,500,000. Indeed TBM Mass Memory storage costs are comparable to the cost of paper storage in filing cabinets.

2. The MARS prototype supports the retrieval of messages based on a variety of criteria including Boolean combinations of message recipients, message date and time, any text words in the message subject and text words in the message body. It will retrieve messages based on these criteria in a matter of minutes.

3. The MARS capabilities are integrated into the existing message environment by having all communications with the archiver take the form of passed messages. A message is archived by including MARS-FILER in the CC: field of the message. A retrieval request is submitted by sending it as a message to the MARS-RETRIEVER.

With this understanding of the MARS objectives and the general scheme for achieving them, we are in a position to consider the system architecture in more detail. Figure 2.1 illustrates the basic MARS architecture. As depicted here, the architecture involves four computers communicating over the Arpanet. These are:

1. A User Host running a message system;

2. The Archiver;

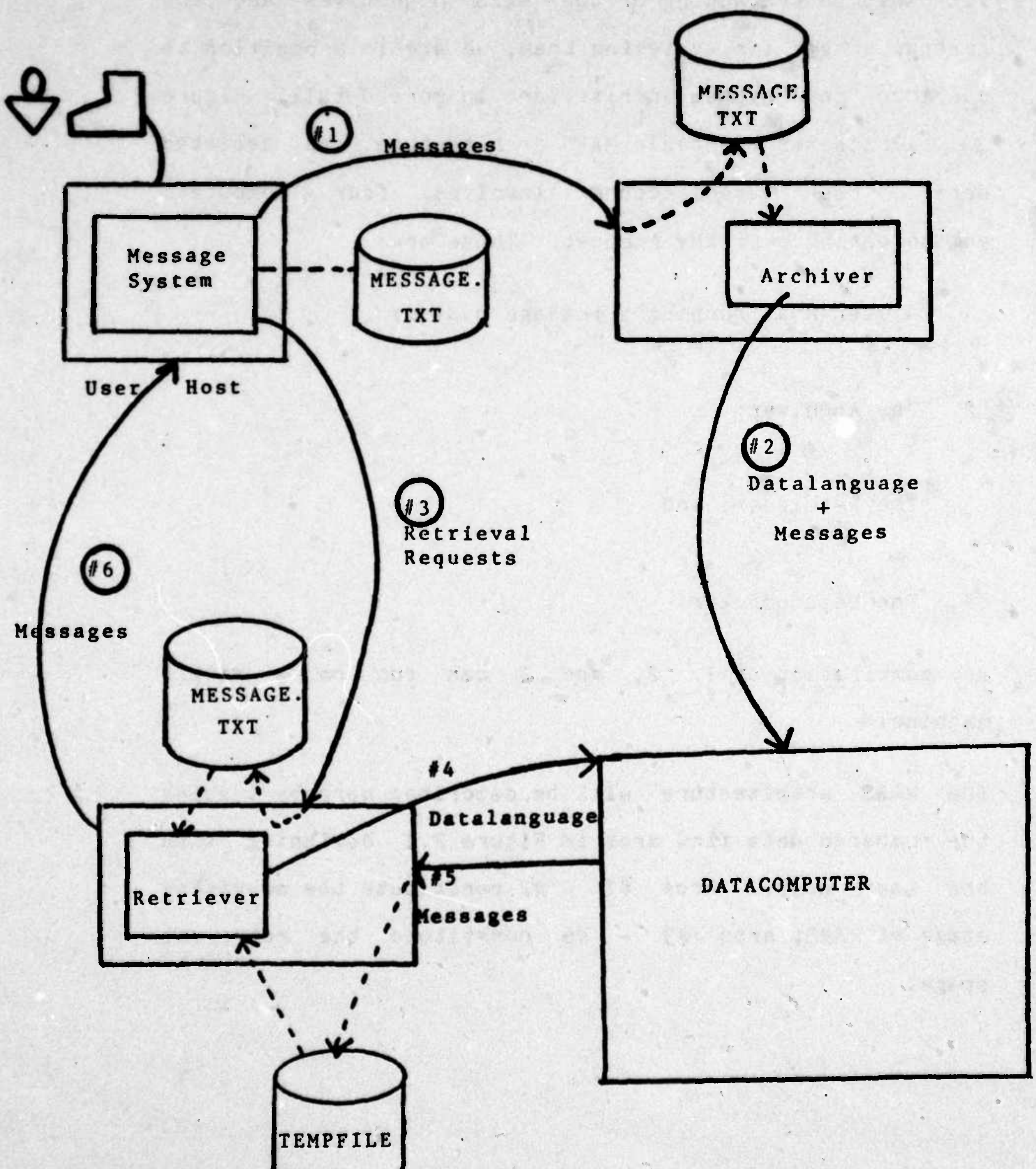
3. The Retriever; and

4. The Datacomputer.

Any combination of 1, 2, and 3 can run on a single machine.

The MARS architecture will be described here by tracing the numbered data flow arcs in Figure 2.1 beginning with the user host. Arcs #1 - #2 constitute the archiving stage of MARS; arcs #3 - #6 constitute the retrieval stage.

MARS Architecture   Figure 2.1



2.3 Archiving

Data flow arc #1: A user employing an ordinary message system designates a message for archiving in one of two ways:

1. When the message is being sent, the user may include the addressee "Archiver@<archiver-site>" in the TO: or CC: fields of the message, or

2. after the message has been received he may forward it to the Archiver. (In the long run message systems could be modified to do this automatically.)

In either case, the result is that the message is sent to the Archiver site and is stored in the MESSAGE.TXT file of the Archiver directory. It is important to note that this interface is compatible with all standard Arpanet message systems and requires no reprogramming of existing message systems.

Data flow arc #2: Periodically the Archiver program wakes up and reads the messages which have been stored in its

MESSAGE.TXT file. It indexes each message on several fields including: TO, FROM, CC, DATE, every word of subject, and every word of text. Then it sends each message along with the appropriate control information (expressed in Datalanguage) to the Datacomputer for

---

Retrieval Request   Figure 2.2

MARS-A Retrieval Request (in Arpanet Message Format):

```
TO:      WALKER
FROM:    ROTHNIE or MARILL

DATE:    BEFORE MAY 1, 1977
TEXT:    MARS or MESSAGE ARCHIVING , MEETING
```

---

permanent storage. <u>Retrieval</u>: The process of retrieving messages from the Datacomputer involves four steps:

1. first a user specifies to the Retriever which messages he desires;

2. This specification is translated into datalanguage and passed on to the Datacomputer;

3. The Datacomputer selects the specified messages and returns them to the Retriever; and

4. The Retriever passes the messages on to the user.

These steps are described further below.

Data flow arc #3: A user specifies messages to be retrieved by preparing a retrieval request (RR) and sending it to the Retriever as a normal message. Two example retrieval requests are shown in Figure 2.2. The format of an RR mimics the format of the messages it is trying to retrieve. The example shows an RR requesting messages sent to Walker by Rothnie or Marill before May 1, 1977, in which the word groups

- MARS or
- message archiving

appear together with

- meeting.

Within this framework, MARS will permit users to specify arbitrary Boolean combinations of indexed fields as the basis for message retrievals.

Data flow arc #4: Periodically the Retriever wakes up and reads its MESSAGE.TXT file. For each RR it finds there, it formulates a Datalanguage request and sends it to the Datacomputer to retrieve the specified messages. (This process is subject to security controls which are built into the Datacomputer.) The Retriever may send the

Datalanguage retrieval request to any the Datacomputer component and the Datacomputer itself will determine which of its distributed modules contains the requested message.

The message activated form of retrieving which is described here is appropriate for most retrievals. However, the complete form of MARS will permit interactive access to the Retriever to accommodate human search behavior and to provide more rapid access.

<u>Data flow arc #5</u>: the Datacomputer retrieves the requested messages from all necessary distributed components and returns them to the Retriever for temporary buffering at the Retriever site.

<u>Data flow arc #6</u>: As the final step in retrieval, the Retriever passes messages from its temporary buffer to a locally resident message system (e.g., SNDMSG) and "mails" them back to the requesting site.

In summary then: The MARS capabilities are conveniently coupled to existing message systems. MARS receives messages to be archived as ordinary network mail and deposits them in its large storage resource, the Datacomputer. Requests for message retrievals are delivered as network mail and mimic the format of the messages to be retrieved. The desired messages are extracted from the Datacomputer and delivered as mail to the requesting user.

2.4 System Description

The description of the MARS system is factored here into two segments. The first segment is a programmer's eye view of the implementation. The second segment describes the format of messages in the MARS database.

2.4.1 The Programs

MARS is composed of three operationally independent programs, a 'Filer', a 'Retriever' and a 'Retrieval Request Preparer'. This segment describes the key components of each program and the current status of their development.

All of the MARS programs are written in the BCPL language except for the pre-existing MACRO-coded lowest-level Datacomputer input/output interface package.

The Filer and Retriever programs, as implemented on CCA-Tenex, are intended to be activated automatically and periodically by a program demon.

The MARS Filer Program key components are: Initialize, Get-Next-Message, Index-Message, Pack-Up-Message and Transmit-To-Datacomputer.

The Initialize component is encoded in a way that facilitates the compilation of both a "production" system as well as a "debugging" version which accepts operating instructions from an on-line terminal. (The technique used relies upon the BCPL compile-time debugging switch.) It is this component which houses most of the operating-system-dependent functions.

While these functions do, typically, concern themselves with generating the programming environment, creating work files, interfacing the program's input/output requirements to the system's facilities, and the like, a specific MARS problem was rather handily solved for us by the Tenex System.

In order to allow for the delivery of new mail during the archiving process, we adopted the technique of renaming the Tenex MESSAGE.TXT;1 file. This file has unique Tenex-based attributes which prevent its deletion whether deliberately or accidentally ordered. Thus, the renaming operation, which would ordinarily result in the loss of the original file name, here causes the simultaneous creation of a fresh MESSAGE.TXT;1 file ready to receive new mail.

The Get-Next-Message component is responsible for scanning the ARCHIVE input file, nee MESSAGE.TXT;1, for isolating individual messages, and for determining the message's filing mode.

There are three modes of filing currently supported in the MARS prototype:

-- single-message mode, wherein the MARS-Filer mailbox appears in the message as an addressee, but not as the primary recipient.

-- forwarded-message mode, wherein the MARS-Filer mailbox appears as the only primary recipient; and

-- batch mode, wherein the mailing envelope is addressed to MARS-Filer and the subject-field contains the keyword "batch".

Since there does not yet exist an ARPANET standard for the format of messages, the variability amongst formats is still greater than the Filer can handle as it stands. Nonetheless, a user can successfully file any message in a "foreign" format by forwarding it to the Filer under the aegis of a mail-handling program which does produce good formats. Admittedly, the correct header-field indexing, as described below, will not be done on the enclosed message; but at least, the words in its unreadable header fields will appear as "text" words in the indexing.

In the case of single-message-mode filing, the entire message is scanned for indexing terms, commencing with the DATE field of the message-header and terminated either by the message's byte count or by the end-of-message indicator supplied by the message-composing program. (The Tenex SNDMSG program, for example, standardly appends seven (7) hyphens followed by the codes for a carriage-return and a line-feed to each message it generates.)

In the case of forwarded-message-mode filing, all interesting indexing information is extracted from the message-header prior to discarding it. The name of the archiver, the date and time the message was forwarded, and the subject-line information are recorded. The remainder is handled as though it were a non-forwarded message which had been CC'd to the Filer.

In the case of batch-mode filing, only the archiver's name and the date and time she/he sent the package are retained. The message-body portion is treated as a series of individual messages.

The <u>Index-Message</u> component is designed to be capable of isolating every parsable token in a message. Needless to say, the adoption of standardized message formats would facilitate this process.

The Filer "indexes", in effect, on everything without analysis, except for the following:

-- Each distinguishable section of the message is indexed separately; each header line is a separate inversion domain, as is the body of the message.

-- The header lines which contain ARPANET addresses are analyzed in order to index separately on mailbox and host.

-- The DATE: field is parsed and converted to a numeric value, to allow retrievals of the form, 'BEFORE<date>', 'SINCE<date>', etc.

A scanner control was designed to comply with the message syntax standards proposed by ARPA's Committee on Computer-Aided Human Communication (CAHCOM) in RFC724 (which, in turn, retains the minimum formatting characteristics proposed in its predecessors RFC561 and RFC680). One requirement is that the DATE field be unique. We further require that it be the first message-header item. Another proposed standard, which is a MARS standard as well, is that the message-header fields be separated from the message-text by a double carriage-return/line-feed (i.e., a blank line).

For each message, a vector of parsed tokens is created. The parsed tokens are collected by the message-field in which they occurred - to be used as "indexes", i.e., values of inverted fields, by the Datacomputer.

These tokens are packed up, along with the message as it will be retrieved, for storage on the Datacomputer.

This method provides all the flexibility we need in order to scan all the various message-header fields that current message-composing programs can produce and to decide which of them may subsequently be used for retrievals.

The following list defines the message-header fields and affiliated field-names which are identified as such by Index-Message, and sketches their interpretation. The terminology is adopted from RFC724.

<date-field> The <date-time> information is converted to the standard Tenex internal date/time format, which is better adapted for less-than/greater-than comparisons, as in retrievals which specify a date range.

<originator> The <mach-from-field> is the only option adequately identified so

as to permit its selection by <field-name>, i.e., "From", on the Retrieval Request. The syntax following the <field-name> is limited to a <host-phrase> which is defined to be an <atom> followed by a <host-indicator>. The mailbox scanner breaks down the <host-phrase> so that retrievals may be performed with or without a <host-name>.

<addressee-field> All of the <address-field> <field-name>s are recognized. The syntax of the field following the field-name (To,cc,bcc,Fcc) is limited to a <host-phrase>.

<extension-field> We recognize the "Subject" and "Message-Id" field-names and handle them separately. The other field-names ("In-Reply-To", "Keywords", "References", "Comments",<user-defined-field>) are collected in a default bin of two-element terms, the first of

which is the field-name, and the second of which is the remainder of the input line exactly as scanned.

The "Subject" field merits special attention because of its generally- accepted purpose of tersely conveying the essence of a message. One-character words are arbitrarily discarded though they need not be; hyphenated phrases, i.e., words bound together by hyphens, are retained intact. (There is even the practice of sending "ethereal" mail via bodyless messages! The entire message is incorporated in the "Subject" field.)

The "Message-Id" field also receives special attention. In fact, if a message is received without one, a unique Message-Id is created for it. It is the contents of this field tested in conjunction with the archiver's name which would enable us to isolate and eliminate duplicates when retrieving.

The composition of the Message-Id field is as follows:

<[From-Host]Sent-Date Time . From-Name>

<message-text> The start of this field is recognized by the blank line which is expected to follow the message-header portion of the message, and it runs until the message byte-count is exhausted.

"Words" and hyphenated phrases are collected, eliminating duplicates and "bubble" sorting in the process so that the resultant list reflects the position of the original occurrence of the term, modified by its frequency of appearance in the message.

Two-character words are discarded though they need not be. And, for now, only the first thousand unique terms are processed. Of these, only the first 127 (the counting limit of a 7-bit byte) are actually used for indexing. Full-text indexing would not require reprogramming here, but see Pack-Up-Message constraints.

The Pack-Up-Message component resolves all conflicts between the scanned message and the format of the data expected to be transmitted to the Datacomputer.

The data sent is constrained to match exactly the format specified in a Datalanguage PORT description. This component needs "re-tuning" whenever FILE and PORT descriptions are modified. The Datalanguage description of a recent MARS model file is appended to the end of this section.

The Transmit-To-Datacomputer component is the Datacomputer interface. It contains the calls to open a file, send 'records' of information, and to close a MARS file during the course of a filing session. Messages are sent, one at a time after indexing.

In the event of network, Tenex or Datacomputer failure, MARS crashes; it is impossible to continue filing once the processing has been interrupted. We do provide for generating a back-up disk file of indexed messages and have succeeded in accomplishing delayed (later) filing by direct call on the RDC program.

The MARS Retriever Program key components are: Initialize, Get-Next-RR, Scan-RR, Perform-Retrieval and Transmit-To-User.

The Initialize component is similar to the Filer program's Initialize; it contains all the Tenex-dependent input/output interface routines.

The Get-Next-RR component is comparable to the Filer program's Get-Next-Message; a strictly-formatted message is expected.

The current implementation requires that MARS-Retriever appear as the primary recipient of the message. The CC: and SUBJECT: fields are ignored.

The Scan-RR component scans the message-body portion of the message to construct a table of tokens which will be used to compose the Datalanguage for performing the retrieval.

The following list defines the message-header field names which are recognized, and some notes on their interpretation. The scanning of each field is terminated by a carriage-return.

DATE: The format of the date field is day-month-year. Use of hyphens is optional. This field will cause only those messages composed on the specified date to be retrieved.

AFTER: Use of this field will retrieve messages composed after the specified date.

SINCE: This field is interpreted like the AFTER: field.

BEFORE: Use of this field will retrieve messages composed before the specified date.

UNTIL: This field is interpreted like the BEFORE: field.

FROM: This field is expected to contain a valid mailbox name. The host specification is optional. If more than one name is specified, ORing of the names is implicit. Retrieval based upon host specification alone has not been implemented.

TO: This field is expected to contain one or more valid mailbox names. The host specification is optional. Spaces and commas between the names imply AND.

SUBJECT: Use of this field will retrieve all messages whose indexed subject-field contents match the specified word(s). Spaces and commas imply AND. The use of OR must be explicit.

TEXT: Use of this field will retrieve all messages whose indexed message-body contents match the specified word(s). Spaces and commas imply AND. The use of OR must be explicit.

The Perform-Retrieval component contains the Datacomputer interface routines.

The Transmit-To-User component mails the retrieved messages one at a time. They will appear as new mail appended to the requester's MESSAGE.TXT file.

The MARS Retrieval Request Preparer is an interactive program which was derived straightforwardly from the Retriever program.

The input stream scanner was lifted from the Retriever's Get-Next-RR component and a rudimentary interactive controller was applied to organize the program flow.

The Retriever's Transmit-to-User component was adapted to transmit the Retrieval Request message to MARS-Retriever and to send a copy of it to the user (as more new mail).

Utility print routines and debugging aids were included in toto and used as the basis for programming the Retrieval Request formatter.

2.4.2 The Message Format

Two file models have been developed for filing indexed messages, the major difference being the anticipated message length. Messages of less than 1000 lines use one model; longer messages (up to 10,000 lines) use the other. This latter definition has sufficed to file all of the ARPANET Special Interest Group MsgGroup correspondence, a collection which is well-noted for its prolixity. We may, at a later time, define more models tailored to specific applications.

The Datalanguage description of the current message format is given below.

Datalanguage Description of MARS Model File

```
CREATE IX FILE LIST (,1000)
MESSAGE STRUCTURE
DATE-FIELD      STRUCTURE
        DATE    BYTE, B=18
        TIME    BYTE, B=18
        END     /* DATE-FIELD */
NET-INFO        STRUCTURE
        DATE    BYTE, B=18
        TIME    BYTE, B=18
        END     /* NET-INFO */
MESSAGE-ID      STRING (,79), C=1, I=D
FROM-NAME       STRING (,79), C=1, I=D
FROM-HOST       STRING (,79), C=1, I=D
SUBJECT-LIST-COUNT      BYTE, B=7
SUBJECT-LIST    LIST (,127), C=SUBJECT-LIST-COUNT
        SUBJECT-WORD    STRING (,79), C=1, I=I
ARCHIVER-NAME   STRING (,79), C=1
ARCHIVER-HOST   STRING (,79), C=1
RECIPIENTS-LIST-COUNT   BYTE, B=7
RECIPIENTS-LIST LIST (,127), C=RECIPIENTS-LIST-COUNT
RECIPIENT       STRUCTURE
        RECIPIENT-NAME  STRING (,79), C=1, I=I
        RECIPIENT-HOST  STRING (,79), C=1, I=I
        END     /* RECIPIENT */
KEYWORDS-LIST-COUNT     BYTE, B=7
KEYWORDS-LIST   LIST (,127), C=KEYWORDS-LIST-COUNT
        KEYWORD STRING (,79), C=1
TEXTWORDS-COUNT BYTE, B=7
TEXTWORDS       LIST (,127), C=TEXTWORDS-COUNT
        TEXTWORD        STRING (,79), C=1, I=I
MESSAGE-BODY    LIST (,1000)
        LINE    STRING (,511), D=31
END;    /* MESSAGE */
```

2.5 Future Directions

In the form described in this report, MARS already offers a unique though rudimentary service to ARPANET correspondents. Its adaptability in processing various message formats while keeping pace with the most advanced offerings in the field augurs well for the life expectancy of the system.

This part of the report explores several avenues of potential development using MARS as a base, and consolidates our opinions on these matters as best we can formulate them at this time.

The ideas are gathered into three major parts; first, filing enhancements; second, retrieval enhancements; and, third, security issues.

2.5.1 On Filing

Improved indexing would result from incorporating a facility for using synonyms and root analysis in message indexing. The indexing scheme of the Filer is intended to maximize the liklihood that a message regarding a given topic will be retrieved in response to a request from a user interested in that topic. To accomplish this end, the Filer must use more information than the exact text of the message itself.

Specifically, MARS could be enhanced by using synonym dictionaries so that, for example, "DOD" and "Department of Defense" and "Defense Department" would be indexed the same way.

The addition of root analysis would ensure that, for example, system and systems are indexed identically.

The most frequent words in message-texts tend, of course, to be the particles -- 'this', 'which', 'with', etc., which are unlikely to be of much use in retrieval. When we undertake the preparation of pre-loaded dictionaries for synonyms and root analysis, we shall include in them a standard list of proscribed words to be ignored for indexing purposes.

Paper-clipping characterizes a facility which would enable a user to archive a message in such a way that it would be retrieved as part of a set of messages with which it might not otherwise have been associated.

One concept explored is meant to resemble that of a physical paper clip: something which binds messages together so that the retrieval of one implies the retrieval of all.

Forward chaining is another linking technique worth considering. If a message refers to an earlier message in a manner detectable by the Filer - e.g., in a WITH REFERENCE TO: or IN REPLY TO: header line, or in an annotation added by the archiver - then a standard-form identification of the earlier message would be placed on a "Refers to" indexing (inverted) list in the current message together with the contents of the Refers-to list of the earlier message itself.

Later, when someone comes across an important message, s/he can, with a simple additional request, retrieve all later messages stemming from the earlier one. (For reasons of documentary integrity, the MARS files have been designed so that a message, once filed, cannot easily be modified; hence the original message cannot conveniently be updated to point to later messages which refer to it.)

2.5.2 On Retrieving

Interactive retrieval in the MARS context connotes a facility for composing retrieval requests and retrieving messages on-line. While most retrievals are expected to be performed off-line, it would occasionally be useful to perform this operation interactively.

Other techniques which were considered, and might still be candidates for future development are:

-- collecting all the retrieved messages into a single file, readable by mail-handling programs, and then FTPing that file directly into the requester's directory; but this requires having access (password or other privilege) to that directory on the foreign host.

-- collecting the retrieved messages as above, and mailing the file as a single long message;

the disadvantage to this is that many mail-handling programs are unable to "go down a level" and explore the structure of a "message" which is really a file of messages. The user would be unable to survey headers, etc., of the contained messages.

-- collecting the retrieved messages into a single file as above, but then placing that file on the Datacomputer with DFTP, and sending the requester just a short message telling him where to find it; this is quite feasible and may yet be done. It does require that the user be on a host which is equipped with a DFTP program, and that s/he be able and willing to use it.

<u>Tolerant retrieval</u> implies the existence of a facility for dealing with common errors so that the value of a retrieval request is seldom completely lost due to user error. Such a capability is important since most retrieval takes place in an off-line fashion with the user unavailable to correct an error quickly.

If the system is too unforgiving, a frustrating sequence of unsuccessful requests could easily ensue. Tolerance would involve actions such as:

-- correcting common misspellings in headers;

-- limiting the size of excessive retrievals by retrieving only message headers or truncated messages; and

-- automatically weakening the restrictiveness of a retrieval request if no messages are found. (This could involve, for example, dropping conjunctive clauses or expanding a range of dates.)

<u>Flexible retrieval</u> denotes a facility for retrieving just portions of messages rather than the complete text. This would include surveying header information or simply counting the records satisfying some condition.

<u>Automatic retrieval</u> might be of greater interest in a more strictly commercial environment. It entails a facility for generating a retrieval request on behalf of a user in response to some event.

This mechanism could be used in conjunction with a "tickler file" to remind a user of something on a given date. For every user employing this feature the Retriever would prepare a Retrieval Request each day and retrieve those messages stored in the tickler file indexed under that date.

2.5.3 Security Issues

Specification of retrieval authority involves providing a facility by which the user who archives a message may specify who may retrieve it. Default authority would be granted to all users listed in the TO:, CC:, or FROM: fields.

Variations on default authority may be specified by the user who archives a message through the use of a RETR(IEVERS): special handling field. The specifications in this field are sequences of 3 types of terms:

user names;
distribution list names; and
distribution list names followed
by an EXCEPT clause.

An EXCEPT clause could be employed to specify names to be deleted from the distribution list.

Audit trails would provide a facility for recording all retrievers of designated messages. If the special-handling-designator AUDIT were specified for a message, MARS would keep a record of each instance of retrieval of that message. The record would include the identity of the retriever, ARPANET host and socket number, and date of retrieval. The audit trail is intended to aid in the analysis of potential security breaches and as a deterrent to unauthorized access.

Record traffic issues require a facility for designating an archived message as an "official" copy of a sent message. This capability requires that MARS be able to justify substantial user confidence that the message in the archives is in fact a true copy of the originally sent message with the same message ID. This could be accomplished with relatively great promise of success in several ways.

Basically, a message would only be archived as "record traffic" if the sender of the message were the same as the user who is archiving the message, and the message is being sent to MARS-Filer as part of the original TO: or CC: list. This would ensure that the archived message is a true copy unless the mailer program was somehow subverted.

A second means of ensuring true copies is a feedback loop in which the message received by MARS-Filer is sent to some human authority for approval. This authority may be the message sender or some official (like a notary) who would contact the sender and verify the legitimacy of the message. Only after the loop had been closed by a message (password protected) from this authority, would the message be recorded as a true copy and marked with a "MARS-AUTHENTICATED" STAMP.

A third approach would employ the cryptosystem which has been devised by Rivest, Shamir and Adleman of MIT in their paper (MIT/LCS/TM-82).

This scheme would work by having the sender of an "official" message apply h(is/er) private encryption key to code the message prior to transmission to MARS-Filer. (Appropriate message-header conventions would need to be adopted to flag this case.) The Filer would look up the public decryption keys for the archiver's name and, using them, run the decryption algorithm, and file the result. As for the previous approach, a copy of the cleartext message would be sent back to the archiver for confirmation. As a security measure against tampering, a copy of the encrypted message would also need to be filed.

Appendix A

Seismic Symposium Paper

USE OF THE DATACOMPUTER IN THE VELA SEISMOLOGICAL NETWORK

Robert H. Dorin and Donald E. Eastlake III
Computer Corporation of America
575 Technology Square
Cambridge, Massachusetts 02139

Abstract

The Datacomputer is a very large capacity, centralized database management facility for distributed networks. It was developed for the Advanced Research Projects Agency (ARPA) by CCA, and is currently available on the Arpanet. The Datacomputer is the primary storage and retrieval resource for the Vela Seismological Network. Several different types of seismic data are stored on the Datacomputer. The size of the database is growing at a rate of about 30 billion bits per month. File organizations attempt to make the most significant data rapidly available to seismologists on Velanet. Special communications protocols were designed and a dedicated processor employed to accomodate high bandwidth, real-time seismic array data.

1. Introduction

The Datacomputar is a network data utility developed by CCA and designed to handle large files and communicate with multiple remote programs on the Arpanet. This system is the result of a research and development effort sponsored by the Advanced Research Projects Agency (ARPA) of the Department of Defense. The Nuclear Monitoring Research Office (NMRO) of ARPA has established a seismic data network, and the Datacomputer is the primary storage and retrieval resource being utilized by the effort.

This seismic data activity involves the collection, storage and processing of seismic waveform information (seismograms) as measured by seismometers installed throughout the world. The data will assist seismologists in exploring techniques for detecting seismic events, pinpointing their location, and recognizing the causes of these events. A major application of the work is the detection of underground nuclear tests in preparation for future Strategic Arms Limitation Treaties. By establishing an on-lina, real-time database of seismic information from a world-wide network of monitoring sites, a graat deal of data can be made easily available to computers in the network for seismic analysis and othar purposes.

This paper describes the storage and retrieval of data in support of this objective. Section 2 outlines the communications facilitias which support the collection and distribution of the seismic data. Section 3 dascribes the storage fscility, the Datacomputer. In Section 4, tha nature of the data stored in the system is explained, whila in section 5, its physical organization is described. Finally, section 6 considers certain detailed communications issues which have arisen in handling thia application.

2. The Vela Seismological Network

The computers at varioua sites involved in the gathering and aubsequent analysis of seismic data ara known as the Vela Seismological Network or Velanat. Since many of the computars are on the Arpanat, the Arpanet was chosen as the most appropriate communications medium available for the entire system. Tha Arpanet is a geographically distributed computer communications network which was designad to provida for the sharing of data and computing resources among its users. Currently thera are over 100 computers (called "hosts") connected to Arpanet, as shown in Figure 1.

ARPANET GEOGRAPHIC MAP, AUGUST 1976

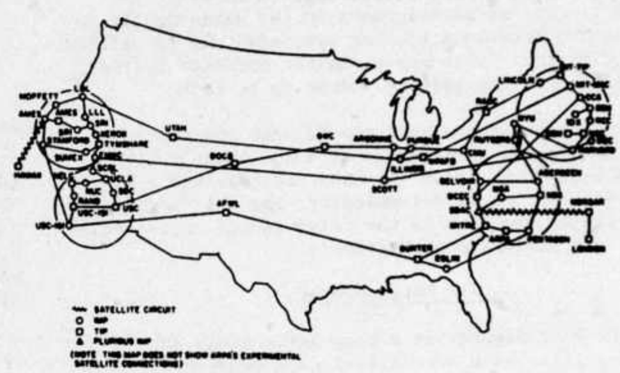


Figure 1.

Arpanet utilizes a technique called "packet switching" to transmit digital information from host to host.[5] Packet switching is a communications discipline in which data is dividad into small (usually ona to two thousand bit) segments called packets. Each packet is then augmented with additional information including its destination, sequencing, and an error checking code. The packet is then launched into a network connecting the communicating points. The network is composed of communications lines and active computer "switching" nodes. Tha switching nodes check for correct transmission, and will retransmit a packet if transmission errors are detected. The nodes dynamically determine the best routing in order to avoid congestion and lines which are out of service. The transmission is nearly as rapid as a dedicated channel but with superior error characteristics.

The structure of the Velanet appears in Figure 2. The Velsnet consists of two sites sending seismic waveform information in real-time, the Large Aperture Seismic Array (LASA) in Montana, and the Norwegian Seiamic Array (NORSAR). LASA data is transmitted via

Seismic Data Network

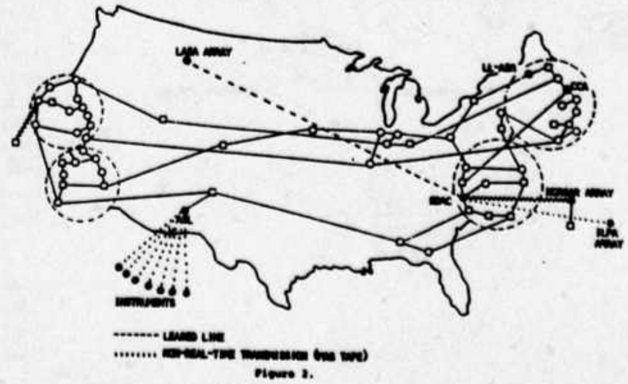


Figure 2.

leased telephone lines to an intermediate processor at the Seismic Data Analysis Center (SDAC) in Alexandria, Virginia. NORSAR data arrives at SDAC via a satellite communications link of Arpanet.

Data which is not transmitted in real-time arrives at SDAC on magnetic tapes from the Iranian Long Period Array (ILPA), as well as from other instrument clusters throughout the world. Non-array seismic data is sent by magnetic tape from various locations around the world to the Albuquerque Seismological Laboratory (ASL) of the U.S. Geological Survey. Both the real-time and non-real-time data arriving at SDAC, as well as the data concentrated at ASL, are forwarded through the Arpanet to CCA. The current traffic of seismic data to the Datacomputer is 12 kb around the clock storing approximately 30 billion bits per month. There are plans for additional sites which may boost the traffic volume up to 35 kb.

Processors throughout the Valanet can retrieve the seismic data. Processors at SDAC, Lincoln Labs Applied Seismology Group (LL-ASG) and possibly elsewhere will be used by seismologists for this purpose. The nature of the data in the files stored on the Datacomputer is discussed in Section 4.

3. The Datacomputer

The Datacomputer is a very large scale data storage utility with substantial data management capabilities.[4] Its design is heavily influenced by its use as a resource in a network and its support of a trillion-bit mass memory device. Though database management systems have been in wide use in the past decade, the network orientation and built-in interface to a mass memory make the Datacomputer a unique resource.

By providing service as a dedicated, special-purpose database "machine" in a network, the Datacomputer is sharable by all computers having access to the network. User programs running on network hosts communicate with the Datacomputer across the Arpanet. These programs, may, in turn, communicate with terminal users, as shown in Figure 3. Of particular importance to the Velanet computers is the support of data sharing among dissimilar machines. The Datacomputer software includes data conversion facilities, and will perform translation between various hardware representations and data structuring concepts.

Datacomputer User's View

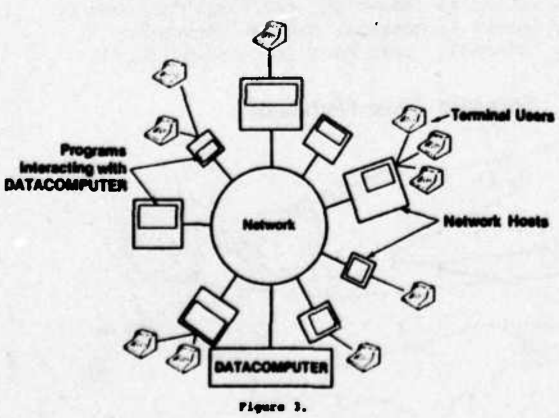


Figure 3.

The architecture of the system is shown in Figure 4. The system processor is a DEC System-10 (PDP-10), running the Tenex operating system. The Datacomputer is interfaced to the Arpanet Interface Message Processor (IMP), which in turn interfaces to two 50 kilobit/second telephone lines into the network.

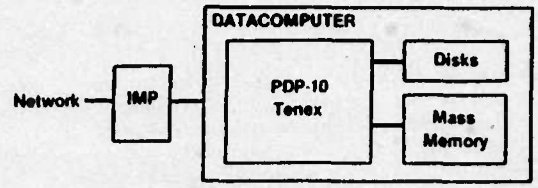


Figure 4.

One of the requirements of the seismic data analysis project is very large on-line storage capacity. This requirement was met by the acquisition and integration of an Ampex Terabit Memory System (TBM)[6], the first public installation of this device. The TBM has a maximum on-line capacity of 3.2 trillion bits. A significant advantage of this technology is an extremely low per-bit cost, about $1/megabit. This is on the order of 20 times cheaper than other on-line alternatives[3]. The mass storage devices currently on the market all have high price tags, with configurations ranging from several hundred thousand to several million dollars, but because of their enormous capacities, their cost per bit is low.

Access time to the TBM is substantially slower than disk storage. The average access to any portion of the three trillion bit mass memory is 15 seconds. In order to improve the effective speed of the device, the Datacomputer software includes extensive facilities to "stage" data from tertiary TBM storage to secondary disk storage. These routines attempt to minimize the overhead involved in using a relatively slow access memory device. When data which has not already been staged is referenced by a user's request, a portion of the database surrounding the referenced data is staged to the faster disks. Some files are staged in their entirety, while others are staged in extents whose size may vary among files. Once staged, data will remain on disk for repeated accesses, from one or more users, until the staging area becomes overcrowded. The preferential de-staging of least recently used and of unmodified data is part of a strategy to optimize use of the staging area.

The Datacomputer offers a complete set of database management facilities for retrieval and maintenance, data description, and access control. Since the files stored on Velanet are rather large (single files as large as 4 billion bits), the capability to select small portions of the database efficiently is very significant. The use of multiple, user-selected inversions provides rapid access to qualified records based on their content. These inversion tables are maintained by the system, and the user need not be aware of their existence. Such efficient retrieval mechanisms will assist seismologists in selecting seismic data containing certain attributes, such as location and magnitude, and then verifying theories about the wsveform signals associated with those attributes.

All the data management facilities of the Datacomputer are handled through a uniform, high-level notation called Datalanguage. Datalanguage is the communicationa vehicle between the Datacomputer and the remote processors uaing this data utility on the network. Further details concerning these database facilities are described in working papers and technical reports.[1,2]

4. Seismic Data Files

There are two basic categories of files stored on the Datacomputer for the seismic project. First, the raw data and status files provide complete seismic readings at various instruments and arrays around the world, as well as associated information on the status of these instruments so that the raw readings can be properly interpreted. Second, there are the derived event summary and associated seismic waveform files. These consist of a processed distillation of the first category into likely seismic events, and the time segments of raw data that are associated with these events. The derivation of the second category of files from the first is performed by processors at the Seismic Data Analysis Center in Alexandria, Virginia.

Raw data files are organized by: type of sensor (array or non-array), frequency of measurement (long and short period), location of sensor, and time interval (day or month of measurement).

Array data originates from the Large Aperture Seismic Array (LASA) in Montana, the Norwegian Seismic Array (NORSAR), and the Iranian Long Period Array (ILPA). LASA and NORSAR are real-time on-line sites. The data is sent through the Arpanet to a central collection point at SDAC. A processor there re-transmits the data to CCA. ILPA data is sent to SDAC on magnetic tape, and then stored into the Datacomputer via the Arpanet. Each of these arrays performs measurements at two frequencies: long period (1 sample per second) and short period (10-20 samples per second). Long period data is stored in monthly files (files containing all observations for a month) while short period data is stored in daily files.

Non-array data is collected from single instruments at Seismic Research Observatories (SRO's) around the world and recorded on magnetic tape. It is then gathered at the Albuquerque Seismological Laboratory (ASL) and forwarded to the Datacomputer over the Arpanet. This data is also divided into long and short period files having one and twenty samples per second respectively. All non-array (SRO) data is stored in multi-site day files. This organization makes it most convenient to locate all of the data related to a single seismic event.

Most of the status information stored in the Datacomputer concerns real-time changes in the status of instruments. A single stream of monthly status files is maintained for the on-line LASA and NORSAR arrays. This file will include status for the instruments of real-time arrays to be added in the future, as well as real-time status entries concerning the communications lines and processors involved in the on-line network. Separate status streams are maintained for the ILPA non-real-time array instruments and for the SRO sites.

Each entry in the status files includes the time of the entry, information on the station and instrument involved, and data relating to the status of the instrument. This data includes indications of communications errors, data being absent, data marked invalid by operator, or calibration in progress. There is also a flag to indicate that the status entry is the result of a change in status or the result of a dump of status for many instruments regardless of change. Status dumps currently exist only for the on-line instruments, and are automatically produced once a day and after communications disruptions in order to recover from any change in status that might have been lost. Determination of the status of an instrument at a particular time requires searching the Datacomputer files for the last entry concerning the instrument before that time. The daily status dumps into the status file limit the interval needed to be searched. As mentioned above, entries are also made for the status of the processors involved in the on-line Velanet.

The culmination of the raw data and status files are the event summary files and their parallel seismic waveform files. These each come in preliminary and final forms.

The event summary files are, at the highest level, a list of arrivals. Each event has a unique event number, assigned by SDAC, and information about the location, depth, and magnitude of the event, as well as further information on the source and confidence level of these conclusions. Finally, for each event, there is a second level list of arrivals of the seismic signals at instruments resulting from the event. For each arrival, this includes information identifying the station, channel, time of arrival, and considerable information about the station and its relation to the event.

No actual seismic readings are included in the event summary files but sufficient information is present to locate the relevant areas in the raw data files or, as explained below, in the seismic waveform files.

The seismic waveform files contain actual segments of seismic data that are associated with arrivals in event files. Each waveform file entry carries the event number from the event summary file entry with which the data is believed to be associated, as well as information about the station, channel, relation between the station and associated event, and the start time and duration of the data segment.

The preliminary event summary and seismic waveform files are derived by SDAC from the on-line arrays whose data is available in real-time. A latter cycle is planned which will take data from the non-array SROs, the off-line arrays, and the detections from the short period SRO data and other sources to produce the final event summary and final seismic waveform files.

5. Seismic Database Organization

The nature of the data and its use varies among the different seismic files, as does their size and structure. The organization of the seismic data attempts to enhance the practical access to this database in which individual files may be as large as 4 billion bits.

Chronological Organization

One uniform characteristic of all of the present seismic files is their organization into chronological streams. Each item represents some event or state at a point in time. Thus, in some sense, a sequence of data of a particular type represents a segment out of a potentially infinite stream of that type of data.

These potentially infinite streams must be broken up into physical files for several reasons. The seismic data will fill many physical volumes (TBM tapes), and the Datacomputer requires that physical files reside on a single volume. The current Datacomputer implementation also puts a limit of slightly greater than 4 billion bits on the size of single filea. The subdivision of the seismic data streama into sections allows the Datacomputer directory to rapidly accesa the chronological area of interest and determine the TBM tape on which the data resides. Sectiona of the data atreams are more managable, and direct indicea to such aections can be built more easily. Finally, in the case of some atatus and calibration filea, each file contains the state of the inatrumenta at the start of ita time period. Thia makes reference to earlier files in that stream unnecesaary.

A uniform scheme haa been employed, uaing the hierarchial directory available in the Datacomputer. Beneath aeveral directory levels apecifying the exact type of data, the data stream ia divided into yeara, in moat casea, the yeara subdivided into montha, and, in aome caaea, the months divided into daya.

File Groupa

The seismic data atreams are divided into a aequence of phyaical files to facilitate their creation and manipulation. However, it ia sometimea convenient to view a atream of one type as a single logical entity and ignore these physical boundariea. This ia done with the file groups feature of the Datacomputer which providea multiple file data aeta.

File groups allows an application to treat a group of files aa a aingle file, from which retrievals may be done in the aame way aa from a normal file. Operationa are available for adding and deleting phyaical file entries in a group, and a file may appear in the list of constituents for any number of groups. So, a file group ia a logical aet of files which facilitatea retrievala from a seiamic data atream which has been divided into aeveral physical files.

Some seiamic file groupa may include thouaands of day filea, and it would be impractical to retrieve from that group if the Datacomputer had to accesa each of ita conatituent files. In order to optimize references to the phyaical files, a general facility ia available whereby a logical conatraint ia placed on the data occurring in each file in the file group. When a request ia iasued on a file group, the Datacomputer can determine which individual filea must be accessed by checking the retrieval condition of the requeat againat the logical conatrainta on the filea in the group. For the aeiamic filea, such conatrainta normally include limits on the values of date fielda. By uaing theae conatrainta, a requeat on a file group for a one hour interval of data will actually acceaa only one or two physical files (two if the hour falls acroaa the boundary between filea).

6. Communicationa Consideratlona

Real-Time Communicationa

The real-time data from the LASA and NORSAR arraya is aent through the Seiamic Data Analyaia Center in Alexandria, Virginia, before arriving at CCA. At present, it conaiata of two Arpanet messagea per aecond, one for each array (this will change when new protocols are implemented aa explained in the following section). The real-time procesaing and tranamiaaion elements leading to CCA provide a ahort interval (currently less than 20 aeconds) of elaaticity in the aystem. A facility muat exist to accept the data at CCA rapidly and continuoualy.

The Datacomputer aystem ia implemented within a general purpoae time-aharing system with a varying load depending on time of day and other factors. It cannot guarantee the required reaponaiveness. In addition, both the basic computer syatem on which it is implemented, a Digital Equipment Corporation PDP-10 with the TENEX operating syatem, and ita peripherala including the TBM, require regular preventative maintenance, and hence cannot be operated continuously.

To provide the required round-the-clock responaivenesa, a amall, dedicated, reliable syatem known aa the Seismic Input Processor (SIP) has been implemented. The SIP ia a Digital Equipment Corporation PDP-11/40 with two RP-04 (i.e., IBM 3330-like) diaka and an Arpanet interface. It accepta the data stream from the Communication and Control Processor (CCP) at SDAC, buffera it on ita diak, reformata the data, and periodically tranamits it to the Datacomputer. At the current bandwidth, 26 houra of buffering are provided per diak pack.

Thua the SIP completely isolatea the real-time data stream from Datacomputer downtime or delay. The SIP alao providea a convenient real-time operator meaaage facility whereby operationa personnel at the Seiamic Data Analyaia Center and at CCA can keep each other appraiaed of ayatem status.

The real-time nature of these communicationa alao made it adviaable not to uae the standard Arpanet hoat-to-hoat protocol. Thia standard protocol ia impoaed on top of the basic mesaage tranamission facilitiea of the Arpanet to allow hoata to dynamically utilize multiple "virtual" connectiona. However, for the Velanet real-time links, all of the connectiona are known in advance, and the standard initial connection and atatua logic is not required. Furthermore, the atandard protocol providea no sequence numbering and is conatrained to not more than one meaaage in flight on a given connection if the connection ia to aupport automatic hoat level retransmisaion on errora. The protocol uaed on the aeismic linka providea ita own aequence numbera, acknowledgementa, and checkauma. It can allow multiple meaaagea in flight up to the limit impoaed by the baaic Arpanet (eight at present) without loaa of other capabilitiea.

High Bandwidth

Some problema have been encountered due to the relatively high bandwidtha of data tranamitted over the Arpanet in thia project. The SIP periodically bursta its accumulated data to the Datacomputer, and utilizea very high bandwidtha frequently exceeding the 50 kilobita per aecond rate at which the Arpanet communicationa linea operate. However, the SIP and the Datacomputer are phyaically adjacent, and are locally connected to the aame network node, the CCA Interface Mesaage Proceaaor (IMP). The data between them need not traverse any outaide communicationa line.

Some problems have ariaen due to congeation in the Arpanet. To alleviate theae problems, the initial Velanet protocola have been redeaigned for greater efficiency. Formerly, each logical mesaage, be it data, an acknowledgment, an operator message, or whatever, was aent as a phyaical mesaage over the Arpanet. In the new veraion of the protocol, several logical messagea are normally packed into phyaical messagea of two typea. One type containa any logical meaaage except acknowledgmenta of correct receipt. The second type containa acknowledgmenta of the firat type of meaaage.

This packing scheme minimizes the Arpanet overhead per message by maximizing the information per message and minimizing the number of messages. Furthermore, inside the Arpanet and invisible to its users, messages, which can be as large as 8000 bits, are divided into packets of 1008 data bits. By accumulating the data and packing it into full size messages, full packets are assured, as well as the minimal packet overhead per data bit.

7. Summary

The Datacomputer provides the Vela Seismological Network shared access to a large capscity, data management utility. Careful file design and special communications protocols were necessitated by the real-time volume of seismic array dats. This spproach makes one of the largest scientific databases in the world available on-line to processors anywhere on the Arpanet.

References

1. Computer Corporation of America, "Further Datalanguage Design Concepts", Datacomputer Project Working Paper No. 8, December 15, 1973.

2. Computer Corporation of America, "Datacomputer Version 1 User's Manual", Datacomputer Project Working Paper No. 11, August 1, 1975.

3. Green, D.E., "A Survey of Mass Memory Technology, Products and Projections", Technical Report No. SAI-74-614WA, Science Applications Inc., January 1975.

4. Marill, T. and Stern, D.H., "The Datacomputer: A Network Data Utility", Proceedings AFIPS National Computer Conference, AFIPS Press, Vol. 44, 1975, pp. 389-395.

5. Metcalfe, R.M., "Packet Communication", Technical Report No. MAC TR-114, Massachusetts Institute of Technology, December 1973.

6. Wildmann, M., "Terabit Memory Systems: A Design History", Proceedings of the IEEE, Vol. 63, No. 8, August 1975, pp. 1160-1165.

Appendix B

Computerworld Article

Page 30 COMPUTERWORLD May 9, 1977

Arpanet DBMS Uses Decsystem-10, Mass Memory

CAMBRIDGE, Mass. — The Datacomputer, said to be a new type of data base management system (DBMS), has been developed by Computer Corp. of America (CCA) under contract to the Advanced Research Projects Agency (Arpa) of the Department of Defense.

The system is dedicated to the data management needs of a network of computers and offers access to a mass memory.

Files in the trillion-bit range may be accessed within seconds, a previously impossible or impractical task using standard disk or tape technology, CCA claimed.

The Datacomputer caps a research and development effort at CCA involving approximately 70 man-years. Service has been available on experimental releases of the system since August 1973, and the full system with a mass memory has been operational since October 1976.

Currently the utilized portion of the mass memory is approximately 120G bits or the equivalent of 150 IBM 3330 disks, CCA said.

The Datacomputer is a network data utility: it is a node dedicated to providing data base management services to other computers on the network.

All communication between the Datacomputer and user programs running on these distant machines is handled through the high-level Datalanguage, according to the firm.

The Datacomputer processor is a Digital Equipment Corp. Decsystem-10 which currently interfaces to an Ampex Corp. Terabit memory with a maximum on-line capacity of 3.2 trillion bits.

The use of mass memory allows very large data bases to be kept on-line and also offers very inexpensive storage for small- and large users alike, CCA said. Effective access to very large data bases is achieved through the use of inverted files and staging strategies.

The Datacomputer will access only the selected portions of large data bases on the mass memory and stage them to faster disk devices for repeated reference.

The Datacomputer is particularly useful for sharing of data among dissimilar machines in a network environment, CCA noted. The system converts data types to match the convention of each machine in the net.

Major Applications

Applications using the Datacomputer include the collection, storage and processing of seismic waveform information as measured by instruments installed throughout the world.

Approximately 20G bits of seismic data are being stored on the Datacomputer per month. This is one of the largest scientific data bases ever assembled, CCA said.

Straightforward bulk data storage and retrieval is another major application of the Datacomputer. Many users alleviate local storage problems by archiving files to the Datacomputer. CCA has developed a simple, terminal-oriented Datacomputer File Transfer Program to facilitate archiving.

One large version of the Datacomputer offers service on Arpanet. The Datacomputer is available to government installations and other organizations with access to the Arpanet.

Other versions of the Datacomputer can be installed on standard Decsystem-10s and -20s, CCA said.

For additional information contact Dr. James Rothnie, CCA, 575 Technology Square, Cambridge, Mass. 02139.

APPENDIX B

Appendix C

Datacomputer Presentation Text

1. Outline

Good morning! I'm Bob Dorin of the Sponsored Research Division at Computer Corporation of America, and I'm going to be talking about the Datacomputer, an advanced system which has recently completed its Research and Development cycle at CCA.

The presentation will be structured in this way. First I'd like to say a few words about Computer Corporation of America, for those of you not familiar with CCA. Next, the management overview will answer those most obvious questions which come to mind concerning the DC, such as what it is and why one might care about it. Then we'll look at the system in more detail beginning with the user's view and then moving to the most significant details of the hardware and software. Next, a few applications currently on the DC will be discussed, and finally, I'll summarize the major points of the presentation.

2. CCA

First, a quick look at CCA: Computer Corporation of America, which was founded in 1965 has its headquarters in Cambridge, Massachusetts, with an office in the Washington area. CCA specializes in advanced database management and communication systems.

More specifically CCA undertakes Research and Development contracts in computer science on behalf of the U.S. government. Much of this work is sponsored by the Advanced Research Projects Agency (ARPA) of the Department of Defense. Some of the project areas we've been involved in are the development of data management access methods, data structures, user-interaction languages, teleprocessing, we've also done work in natural language query systems, CCA is probably the industry leader in distributed database research. Ultra-large on-line storage media, and computer networks are major aspects of the DC, and you'll be hearing more about them today.

CCA's internal R&D efforts have resulted in two proprietary software products: a top-of-the-line database management system called Model 204 and a minicomputer based message system called TDA. Model 204 is implemented for IBM 360 and 370 computers. The system is very widely used, particularly in the Federal government. TDA is a new product which offers facilities for message preparation, editing, communication, filing, and retrieval, and is suitable for private or common carrier networks.

Today we are going to discuss the Datacomputer, one of the projects CCA has undertaken for ARPA. The Datacomputer was a large research effort, involving approximately 70 man-years in conceptualization, design and implementation. The system completed its development cycle in December 1976 and is now entering an extended operational period in which it will provide service to government users. It is my intention today to tell what this system is and how it can be made available for your use.

3. Management Overview

We'll begin with a management overview structured around the following questions which come to mind for those unfamiliar with the DC. What is the DC? Why is it of interest? What does this thing look like physically? Where are there DC installations? What applications use the DC? And finally how might I use the system?

3.1 What is DataC

Well, what is the DC? It is a very large capacity, centralized database management facility for distributed networks. This picture represents a distributed network. Several different computers, at different geographic locations, are connected to the network. Each of these machines, called "hosts", can communicate with other hosts on the network. Networks are proliferating today because of the potential they offer for sharing data and computing resources. The most widely known computer network is the Arpanet. One large version of the DC is a host on Arpanet.

The DC runs as s host on the network, and it provides service as a data management utility. This means that other hosts can create files, store data and retrieve that data using the database facilities of the DC. In order to perform effectively in this role the DC needs a very large storage capacity. The DC achieves this by employing a mass memory with a potential for storing three trillion bits, that is the equivalent of three thousand 3330-type disk packs and it is all available on-line. So, in a nutshell, the DC is a database management system which provides service to computers in a network and employs a mass memory for capacity and economy.

3.2 Why of interest

Well, this sort of facility is certainly unique but why is it of interest - what benefits csn it provide? This is perhaps the most important question we hope to answer today. First, the DC is a convenient place to put data to be shared among several systems, be they geographically dispersed processors or multiple processors at a particular large installation. Nearly all large spread out organizations have databases of interest to more than one installation. Having a centrally managed copy of that database is the only technically feasible approach to sharing such databases available today. So the DC is designed to support this sort of sharing even among dissimilar hardware types. The hosts accessing the DC may be large machines, minicomputers, and may represent different manufacturers.

The second major benefit of the DC is its support of mass memories. Several mass memories are now on the market, offering capacity in the trillion bit range. These systems offer two advantages. First larger databases may be stored on-line than ever before possible.

Second, the storage cost is an order of magnitude less than other on-line alternatives. This, of course, is significant both to large and small databases. Mass memories have a high initial cost with configurations ranging from several hundred thousand to several million dollars. However, the cost per bit is on the order of $1/megabit. Disk storage is typically 20 times that. In order to take advantage of the low cost per bit, one needs a great deal of data. By handling the data needs of several systems with one facility, one gets large collections of data and heavy use of the facility. We estimate an average on-line storage charge of $5-10/megabit per year on the DC. This includes all processing and I/O charges, whereas typical time-sharing system charges are 25 times that for storage alone.

One of the key benefits of the DC is that it really does support a mass memory. It is the only complete system offering convenient, ready-to-use access to a mass memory, since all interface software needed to effectively use the mass memory is built into the DC. The mass memory manufacturers have not developed sophisticated interfaces. Attempts to interface software to mass memories have been difficult and time-consuming efforts. We are aware of such attempts to efficiently access very large databases on a mass memory. In some cases, as much as two man-years of systems programming resources were invested, without the desired results being achieved. Efficient selective access to a mass memory is by no means a trivial problem.

Let's go on to another point.

The database services are available without duplication of effort. The DC is a specialized data management installation with elaborate facilities for that purpose.

All the network hosts can use the data management facilities of the DC without a database system running locally at the site. It would be difficult to duplicate and maintain the data management services at each computer in the net. Personnel resources, as well as additional hardware/software costs, would be required to keep a database system running smoothly at each site.

There are certainly other interesting aspects of the DC, but I've concentrated on those benefits which are of particular interest, and make it unique from standard database management systems running locally.

3.3 What like physically

What is it like physically? The DC is realized as software that runs on standard DEC equipment. It supports trillion-bit storage and network interfaces. Just to give you an idea of what we're talking about, this picture is the DC at CCA. On the far left is the PDP-10 with the mass memory to the right.

3.4 Where are they

Where are DC installations today? Currently a version of the DC is running at CCA, and is operated by CCA as a service to users of the Arpanet. It supports several hundred users from about 20-30 hosts on the Arpanet.

This DC on the Arpanet has been a complete working system since Aug./75, and since mid-1976, has included a mass memory. The mass memory currently installed is an Ampex TBM, the first to be commercially installed. The current capacity is 200 billion bits, but the potential capacity is 16 times that or 3.2 trillion bits. Currently the utilized space is the equivalent of approximately 150 3330 disks.

Another DC has been installed at the Naval Ocean Systems Center (formerly the Naval Electronics Laboratory Center) in conjunction with the Advanced Command and Control Architectural Testbed project (ACCAT). This particular DC is being accessed by hosts on a special secure network. This installation is an example of a DC configuration without mass memory. A second DC will be installed on this network in June.

3.5 What applications

The DC is suitable for a wide range of applications ranging from sophisticated database management of formatted data to straightforward bulk data storage and retrieval.

One particularly interesting application which involves a complex shared scientific database is the seismic data analysis project which uses the service at CCA. Data is stored into and retrieved from the DC via the Arpanet. Data is gathered and analyzed at dispersed locations on the network. This is a very large database, and I'll be saying more about it later.

An application at the other extreme is bulk data storage and retrieval. The DC can simply be used as a large storage device to archive files for later use. For example, students at a university computing center can store files very cheaply on the DC. Students at Harvard have been doing this for several years. So the DC can be used in quite a range of applications, both complex database applications and simple, yet very economical, file storage.

3.6 How to use

How can we use the DC?

3.6.1 DC Service

The quickest way to gain access to the DC is to use the DC service available on the Arpanet. Those of you with access to the Arpanet who wish to use the DC should let us know. Presently there is no charge to government users for small-to-medium size DC applications on the Arpanet. In the context of the Arpanet DC, a small-to-medium application is, say, 1 billion bits. That is a great deal of free storage. It would cost between $100 and $200 K per year on a typical time-sharing system.

3.6.2 DC Software

The DC software can be installed on any PDP-10 or PDP-20, and a network interface installed to fit other environments. The DC at NOSC has a special network interface, and since the system is modular, new interfaces can be developed and incorporated into the system in a straightforward manner.

3.6.3 Customized Applications

CCA is also available to assist users with complicated applications in using the DC. We will undertake development work, and can customize DC applications to meet the database requirements of a particular organization.

This ends the overview portion of the presentation.

4. User's View

4.1 Programs on remote hosts

We now have a general picture of what the DC is and how it might be used. Let's now consider things in a little more detail. First, the external view of the DC, the user's view. In looking at the DC from a user's point of view, it is important to realize that the real user of the DC is not a person; it is a program running on a remote host. These programs may or may not in turn support terminal users. In general their function is to handle database management needs by communicating with the DC over the net. The means of communicating needs to the DC is to send messages containing a high-level database notation called Datalanguage.

4.2 Datalanguage

In Datalanguage, a user can perform the functions expected of a database managment system: storage, manipulation, and retrieval of information. The purpose of the programs running on the host machines communicating with the DC is to send DL across the network. The DC then can send messages back to the user program. Data transfers to or from the DC are also accomplished through this uniform interface.

This is an example of a user program which takes natural language questions concerning a database of ships and translates them into DL. The user asks the question "Where is the Enterprise", and the application translates the question into the appropriate DL. The DL is sent by the program across the network to the DC.

The DC then accesses the appropriate data, in this case a single record for the Enterprise. This data is, in turn, converted by the DC into the form specified by the application program (called a "port") and passed back across the network. Finally the application program can respond to the user.

This particular natural language application was developed by a DC user for their database.

4.3 General data management

DL offers a complete set of retrieval and maintenance facilities. Boolean expressions may be used to qualify a portion of the database. That qualified set of records may be modified, deleted, or retrieved either as complete records or in a format of selected fields only. Computation and conditional processing may be performed on the data using DL facilities.

The DC offers extensive data description facilities in order to specify the form of the stored data. The file description given at file creation time indicates the logical structure of the file, as well as certain physical attributes. Correspondingly each time data is sent to the DC or retrieved from the DC, a format must be specified for the bit stream going across the network. This port description may also be stored for later re-use. Conversions will occur in order to transform the data from its file description to the port description. Or vice versa.

Keep in mind that a variety of systems may be connected through this network. Different HW types may be retrieving data, and must receive data in the format of that particular machine. The DC contains conversion facilities to match appropriate character set, numeric representation and byte sizes.

Finally a hierarchical directory offers facilities for flexible access controls. Passwords can be used to control read, write and file creation privileges on the DC.

4.4 Summary

The key user view ideas are that the DC offers a complete set of database management facilities, all these functions are accomplished through DL, and the DL is sent to the DC by user programs running on remote hosts.

5. Architecture

5.1 Hardware

5.1.1 Components

Now let's take a look inside the DC, and see how this user view is accomplished. First, the HW. A standard component of the DC is a DECsystem/10 or DECsystem/20. The DC runs under the Tenex or Tops-20 operating system. Though it is not a required part of the DC system, a mass

memory may be used, and it is certainly a significant aspect of the design. When data is being accessed by the DC, it is first "staged" from the mass memory onto its 3330-like disk storage so that the high access speed of the disk can be taken advantage of for currently active data. We'll be talking in more detail about staging shortly. An interface message processor links the DC to the network. In the case of the DC running at CCA on the Arpanet, the interface message processor is a Honeywell 516 computer.

5.1.2 Mass Memory

Now let's talk a little more about the mass memory. The DC at CCA is equipped with a trillion bit store with a potential maximum on-line capacity of 3.2 trillion bits. I previously discussed the benefits of mass memory technology. In summary, the two major advantages are the practical use of very large on-line databases and the economy of storage resources.

As I mentioned, the mass memory at CCA is an Ampex Terabit Memory. A unique feature of the TBM is its use of video tape technology to achieve ultra-dense recording: about one million bits per linear inch of tape, about 100 times denser than ordinary mag tape. This is a reel of TBM tape. A TBM tape is 45,000 inches long, and a little arithmetic will show that it contains the equivalent of 125 standard 1600 bpi mag tapes. TBM tapes are addressable in blocks of 1 million bits so a pseudo-random access behavior is achieved. The mechanical access speeds are remarkably fast. In searching for a block, tape speeds reach 1000 inches per second. At this rate, the equivalent of 2-3 normal mag tapes are being passed every second. The average search time to any bit of the 3 trillion is about 15 seconds.

5.2 Software

Let's take a look at certain software facilities.

5.2.1 Staging

As I said earlier, when data is accessed on the mass memory, it is staged from the tertiary storage to the faster secondary disk storage. The staging software as well as all the interface software required to use the mass memory, is built into the DC.

When data is referenced by a user's request, the DC checks to see if the data has already been staged to disk. After staging a portion of the database surrounding the referenced data, that data will remain on disk (to be accessed by that same and other users) until the staging area becomes overcrowded. Repeatedly referenced data is likely to remain staged, thereby incurring the staging overhead only initially.

Files may be staged in their entirety or in extents whose size may vary from file to file. Furthermore, the nature of a user's activity will affect the staging overhead incurred. Applications which execute successive transactions which access the same file will tend to require less staging than purely random transactions. Similarly the balance of transactions coming from all users will affect the staging activity. The DC will preferentially de-stage data which has not been modified while on the staging disks. This is another strategy taken in an attempt to minimize the overhead incurred.

5.2.2 Inversions

The DC also attempts to achieve rapid access to data that has been staged. Inversions on multiple, user-selected fields is of particular importance in processing large files. This capability of retrieval by content aids in the selection of small portions of large databases. The file description specifies those fields which are selected for inversions.

For those of you who are not familiar with inverted files, they are logical pointers to the data proper based on the value of a field. In other words, were I to request the Enterprise a technique known as hashing would get me directly to the inversion entry which would in turn point to the record. This processing would take essentially a fixed amount of time regardless of file size. The alternative of sequentially searching the file would be far less efficient.

Similarly, multiple inversions could be kept so as to efficiently answer queries concerning the "carriers belonging to the USA", for example. The inversion tables are maintained by the system, and the user need not be aware of their existence. DL requests make no syntactic distinction between inverted and non-inverted fields, but will use whatever inversions are available attempting to optimize the particular request.

6. Applications

To give you a feeling for the way all of this fits together: It might be interesting to take a closer look at a few current DC applications.

6.1 File storage

Bulk data storage and retrieval is a major application of the DC. By using the DC as a big "bit bucket", local computing environments can minimize their local storage requirements. I mentioned earlier that Harvard University takes advantage of the economies available on the DC. They run a program called DFTP for DC File Transfer Program. This program provides the terminal user with a simple set of commmands allowing him to archive and subsequently retrieve files. Many sites on the Arpanet run DFTP. This application does not utilize all of the database management facilities of the DC, but rather it uses the DC as a convenient and economical storage medium.

6.2 Command and control

Another use of the DC is being made in the Advanced Command and Control Architectural Testbed being developed at the Naval Ocean Systems Center in San Diego. Several DC's are being installed there, and being used as a basic vehicle for data management. These command and control applications involve the real-time storage and retrieval of force status information which enables commanders to keep track of equipment, personnel condition and other data about military units. As part of this effort, CCA is building a distributed database system using the DC and its interface as a base. Current users of the DC can take advantage of this new technology since existing DC applications will run on the distributed system.

6.3 Shared Scientific

The DC seems ideally suited for large, shared scientific databases. An example of this is the seismic data analysis project which involves the collection, storage and processing of seismic waveform information as measured by instruments installed throughout the world. This activity will assist seismologists in exploring techniques for the detection of seismic events, as well as assisting in the detection of underground nuclear tests. The DC is the primary storage and retrieval resource being utilized by the effort.

This application of a shared database makes extensive use of the network facilities of the DC. For those not familiar with the Arpanet, it is an international network of over 100 computers at about 40 sites (or "nodes") designed for resource sharing. These sites are significant in the seismic data analysis project. Data is passed into the DC from different machines, in different forms, and some is sent in real-time from collection points in Montana and Norway, the latter over the Arpanet satellite connection. Other data is collected and sent from sites such as the Albuquerque Seismological Laboratory. Some of the files contain data compiled by an intermediate processor, located at the Seismic Data Analysis Center in Alexandria, Virginia. Retrieval and analysis of the data can be performed from any host; much of this work is centered at Lincoln Laboratories of MIT in Massachusetts.

The current traffic of seismic data to the DC is about 12 kb around the clock storing about 30 billion bits per month. There are plans for additonal sites with traffic possibly exceeding 35 kb. This is one of the largest scientific databases ever assembled.

7. Summary

In summary, the DC was developed by CCA for the Advanced Research Projects Agency. There are well over 100 users on the Arpanet. The DC's use of a mass memory offers very large on-line capacity with significant storage economies to large and small users alike. The DC is a convenient system for data sharing among dispersed and dissimilar machines. The service is available on the Arpanet, and can be installed on standard DEC equipment. Finally, CCA stands ready to assist organizations in meeting their application goals.

Appendix D

Datacomputer Presentation Slides

The DATACOMPUTER

A Network Data Utility

Computer Corporation of America
575 Technology Square
Cambridge, Mass. 02139
(617) 491-3670

Outline

- CCA
- Management Overview
- Datacomputer User's View
- Datacomputer Architecture
- Datacomputer Applications
- Summary

Computer Corporation of America

- Founded in 1965
- Offices in Cambridge and Washington
- Specializes in Advanced Database Management and Communication Systems

Research and Development

- Access Methods
- Data Structures
- User-Interaction Languages
- Teleprocessing
- Natural-Language Query Systems
- Distributed Database Management
- Ultra-Large On-Line Storage Media
- Resource-Sharing Computer Networks

Software Products

- Model 204 Database Management System
- TDA Computer Message System

DATACOMPUTER

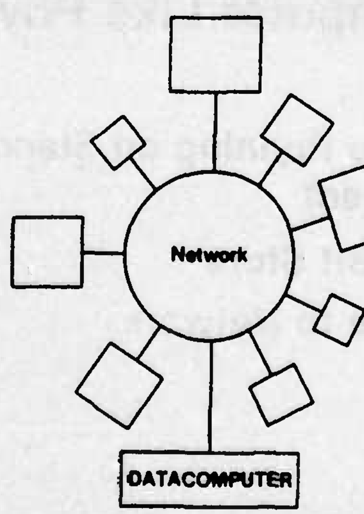
- R&D Project for ARPA
- 70 Man-Year Effort
- Development Completed in December 1976

Management Overview

- What is the Datacomputer?
- Why is the Datacomputer of Interest?
- What is a Datacomputer Like Physically?
- Where are there Datacomputer Installations Today?
- What are Some Applications of the Datacomputer Today?
- How Can One Use the Datacomputer?

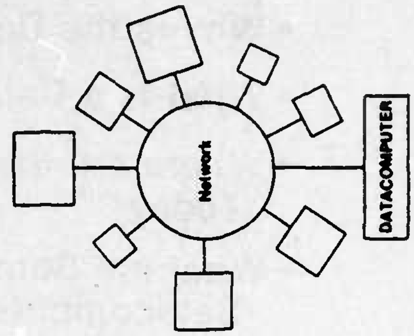
What Is the DATACOMPUTER?

A Very Large Capacity Centralized Database Management Facility for Distributed Networks



Why Is the Datacomputer of Interest?

- Data Sharing Among Dispersed and Dissimilar Computers
- Support of Mass Memory Devices
  - Very Large On-Line Databases
  - Economy of Storage Resources
  - All Interface Software Built Into Datacomputer
- Database Services Available Without Duplication of Effort



What Is a Datacomputer Like Physically?

- Software Running on Standard DEC Equipment
- Trillion-Bit Store
- Interface to Network

Where Are There Datacomputer Installations Today?

- **At CCA**
  - Arpanet
  - Ampex TBM
- **At Naval Electronics Laboratory Center**
  - Special Network
  - No Mass Memory

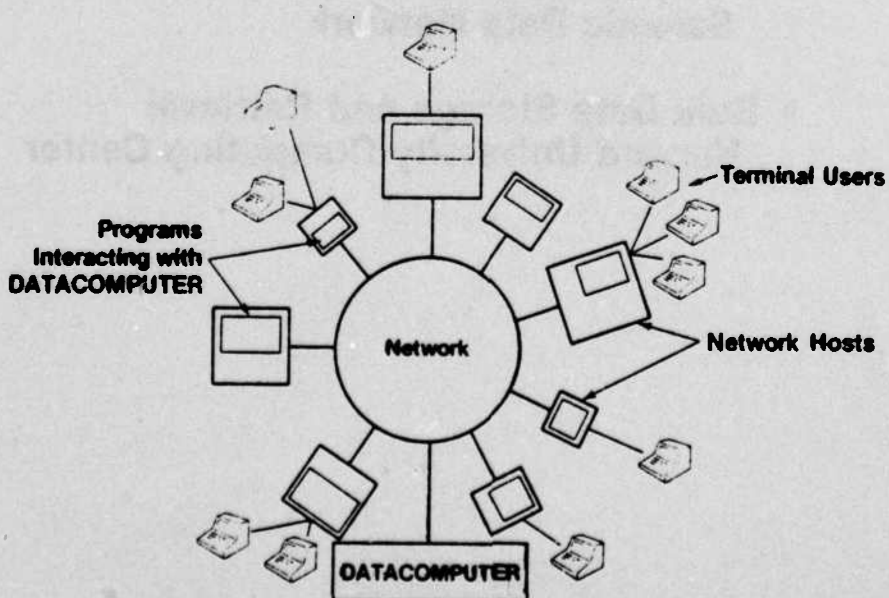
What Are Some Applications of the Datacomputer Today?

- **Complex Shared Scientific Database**
  Seismic Data Network
- **Bulk Data Storage and Retrieval**
  Harvard University Computing Center

How Can One Use the Datacomputer?

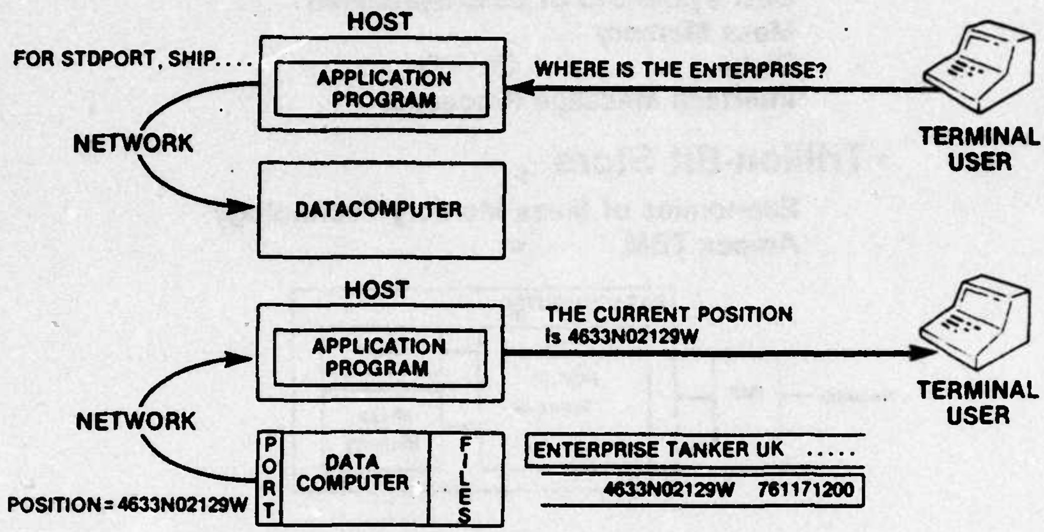
- CCA System Available to Arpanet Users
- Datacomputer Software Installed on Any PDP-10 or PDP-20
- Customized Applications

Datacomputer User's View



Datalanguage

- Uniform, High-Level Interface



Database Management

- Efficient Retrieval and Maintenance Facilities
- Extensive Data Description Facilities
- Flexible Access Controls

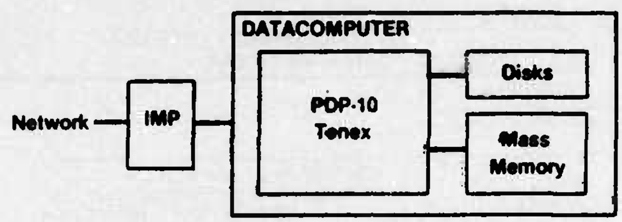
Datacomputer Architecture
Hardware

- **Components**

  DEC System/10 or DEC System/20
  Mass Memory
  Disks
  Interface Message Processor

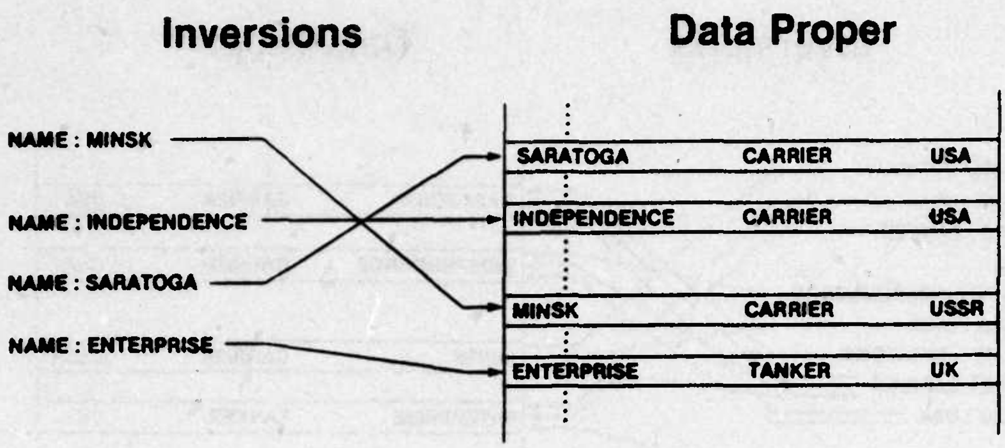
- **Trillion-Bit Store**

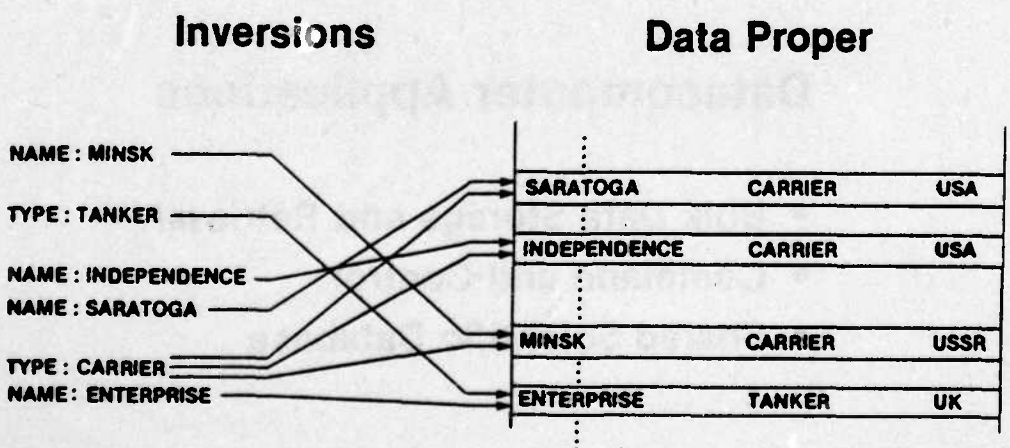
  Economies of Mass Memory Technology
  Ampex TBM

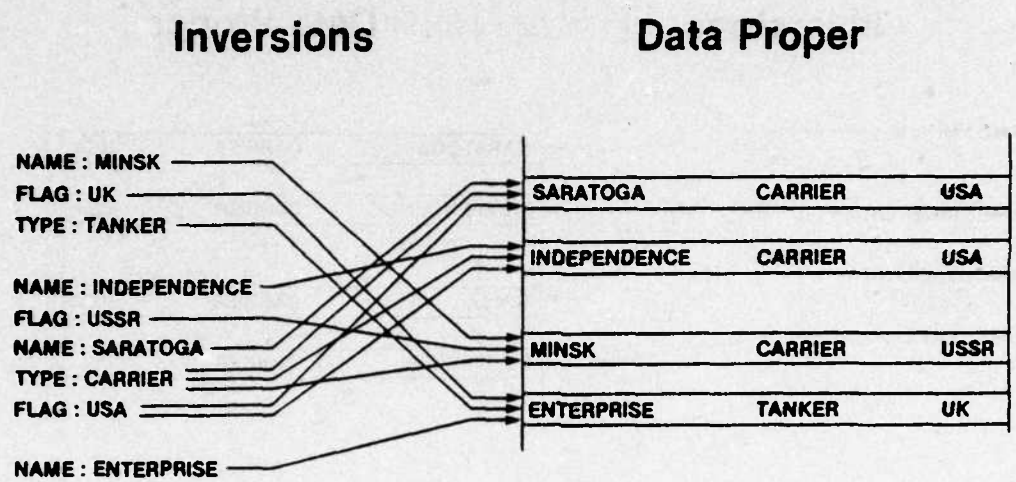


Datacomputer Architecture
Software

- **Staging and Interface to Trillion-Bit Store**
- **Use of Inversions**


Inversions
Data Proper
NAME : MINSK
NAME : INDEPENDENCE
NAME : SARATOGA
NAME : ENTERPRISE
SARATOGA CARRIER USA
INDEPENDENCE CARRIER USA
MINSK CARRIER USSR
ENTERPRISE TANKER UK

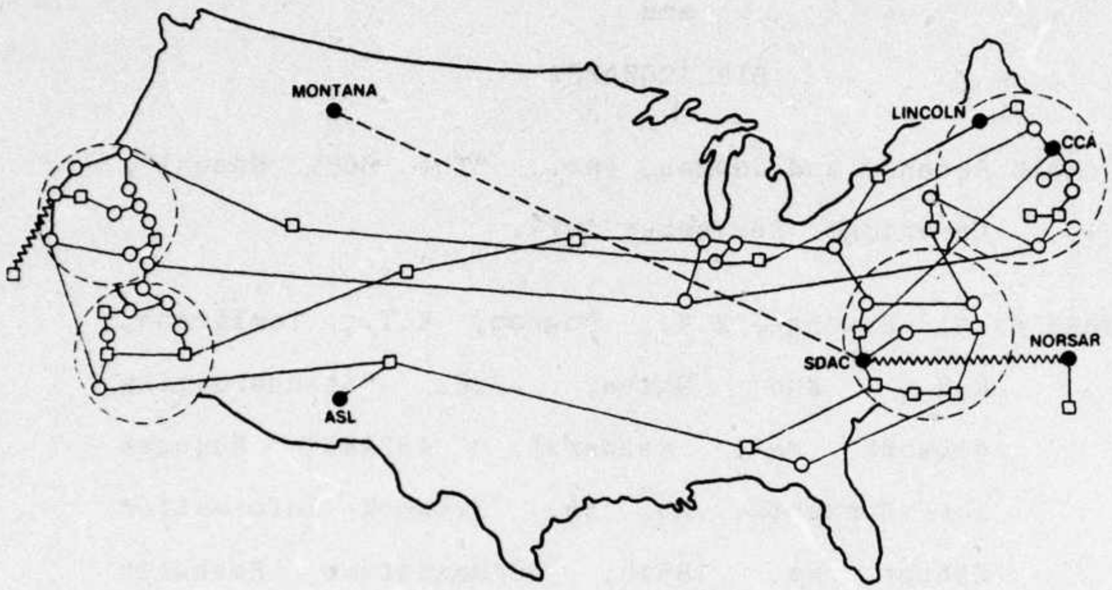

Inversions
Data Proper
NAME : MINSK
TYPE : TANKER
NAME : INDEPENDENCE
NAME : SARATOGA
TYPE : CARRIER
NAME: ENTERPRISE
SARATOGA CARRIER USA
INDEPENDENCE CARRIER USA
MINSK CARRIER USSR
ENTERPRISE TANKER UK



Datacomputer Applications

- Bulk Data Storage and Retrieval
- Command and Control
- Shared Scientific Database

Seismic Data Network



Summary

- CCA's Datacomputer—Developed for ARPA
- Over 100 Users on the Arpanet
- Very Large Capacity, Very Economical On-Line Storage
- Convenient Way of Sharing Data
- Easily Arranged Use of Service Over Arpanet
- Software Installed on Standard DEC Equipment
- CCA Ready to Help Application Development

REFERENCES

and

BIBLIOGRAPHY

[BBN] Bolt Beranek and Newman, Inc., "The BCPL Manual", Cambridge, September 1974.

[BHUSHAN et al] Bhushan, A.K., Pogran, K.T., Tomlinson, R.S., and White, J.E. "Standardizing Network Mail Headers", ARPANET Request for Comments, No. 561, Network Information Center No. 18516; Augmentation Research Center, Stanford Research Institute: Menlo Park, September 1973.

[CCA-a] Computer Corporation of America, "Datacomputer Version 1 User Manual", Datacomputer Project Working Paper No. 11, August 1, 1975.

[CCA-b] Computer Corporation of America, "Datacomputer Version 2 User Guide", Datacomputer Project Working Paper No. 12, November 9, 1976.

[CCA-c] Computer Corporation of America, "Datacomputer Project Semi-Annual Technical Report", July 1, 1976 - December 31, 1976.

AD-A045 623 COMPUTER CORP OF AMERICA CAMBRIDGE MASS F/G 9/2
VERY LARGE DATABASES.(U)
AUG 77 R H DORIN, J Z SATTLEY N00014-76-C-0991
UNCLASSIFIED CCA-77-10 NL

2 OF 2
AD
A045623

END
DATE
FILMED
11-77
DDC

[JOHNSON and KALLAUS] Johnson, M.M., and Kallaus, N.F., "Records Management", Library of Congress #66-12719, 1967.

[MARILL and STERN] Marill, T. and Stern D.H., "The Datacomputer: A Network Data Utility", Proceedings AFIPS National Computer Conference, AFIPS Press, Vol. 44, 1975, pp. 389 - 395.

[MYER and HENDERSON] Myer, T.H. and Henderson, D.A., "Message Transmission Protocol", ARPANET Request for Comments, No. 680, Network Information Center No. 32116; Augmentation Research Center, Stanford Research Institute: Menlo Park, 1975.

[POGRAN et al] Pogran, K., Vittal, J., Crocker, D., Henderson, A. "Proposed Official Standard for the Format of ARPA Network Messages", ARPANET Request for Comments No. 724, Network Information Center No. 37435,Augmentation Research Center, Stanford Research Institute: Menlo Park, May, 1977.

[RIVEST et al] Rivest, R.L., Shamir, A., and Adleman, L.,"On Digital Signatures and Public-Key Cryptosystems", Massachusetts Institute of

Technology Laboratory for Computer Science Paper MIT/LCS/TM-82, April 1977.

[SALTON-a] Salton, G. "The SMART Retrieval System - Experiment Documentation Processing", Prentice-Hall, N.J. 1971.

[SALTON-b] Salton, G. "Automatic Information Organization and Retrieval", McGraw Hill, 1968.

[SATTLEY] ARPANET Special Interest Group MSGGROUP correspondence, from July 1975 to present.

DATE

ILMED

– 7