Title
DISTRIBUTED DATA MANAGEMENT IN A MINICOMPUTER NETWORK: THE SEEDIS EXPERIENCE

Author
Merrill, D.

Publication Date
1982-10-01

Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

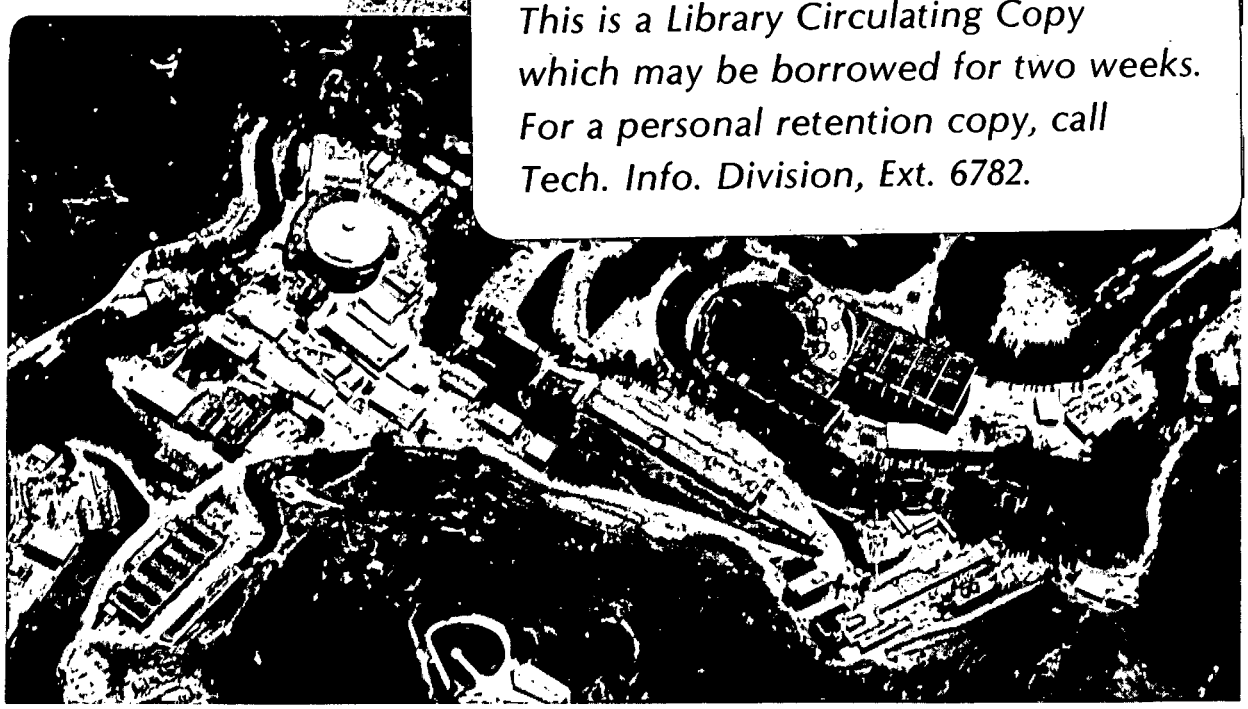
Computing Division

Presented at the 1982 Integrated Data Users Workshop, USGS National Center, Reston, VA, October 13, 1982; presented at the Second International Workshop on Statistical Database Management, Los Altos, CA, September 27-29, 1983; and published in the Proceedings

DISTRIBUTED DATA MANAGEMENT IN A MINICOMPUTER NETWORK: THE SEEDIS EXPERIENCE

D. Merrill, J. McCarthy, F. Gey, and H. Holmes

October 1982



Prepared for the U.S. Department of Energy under Contract DE-AC03-76SF00098

DISCLAIMER

LBL-15075

DISTRIBUTED DATA MANAGEMENT IN A MINICOMPUTER NETWORK: THE SEEDIS EXPERIENCE

Deane Merrill, John McCarthy, Fred Gey, Harvard Holmes

University of California
Lawrence Berkeley Laboratory
Berkeley, California 94720

October 7, 1982

Presented in the Proceedings of the Second International Workshop on Statistical Database Management, September 27-29, 1983 Los Altos, CA.

DISTRIBUTED DATA MANAGEMENT IN A MINICOMPUTER NETWORK: THE SEEDIS EXPERIENCE

Deane Merrill, John McCarthy, Fred Gey, Harvard Holmes

Computer Science and Mathematics Department
Lawrence Berkeley Laboratory
Berkeley, California 94720

Abstract

This paper describes distributed data management aspects of SEEDIS (Socio-Economic Environmental Demographic Information System). SEEDIS is an experimental system for the retrieval, analysis, and display of geographically linked data. SEEDIS operates on nine computers in a nationwide network. Users at any location select and retrieve all data in the same way, regardless of whether they are stored locally or at a remote location.

The network implementation has been substantially modified during 1983. New enhancements include: local caching of data files to improve efficiency; linking to an automatic tape library (ATL) to make larger volumes of data accessible; node independence to facilitate automatic sharing of data among autonomous SEEDIS installations without the need for central control; improvements providing robust operation despite unreliable network connections; and automatic recording of all cache transactions for subsequent statistical analysis.

1. History and Background

SEEDIS (Socio-Economic Environmental Demographic Information System) is an experimental integrated computer system for the retrieval, analysis and display of geographically linked data [1]. SEEDIS embodies 60 person-years of cumulative integrated development, supported since the early 1970's by the Department of Energy, Department of Labor, Environmental Protection Agency, and other government agencies. SEEDIS is used both as a development testbed for computer science research, and in selected applications.

A major task of SEEDIS is the integration and organization of data from diverse sources. Used primarily by universities and government agencies, SEEDIS fills a need not met by two other kinds of systems available in the private sector: time series financial systems used for modeling and predicting economic trends, and small-area demographic systems used to access census data for market site analysis [2].

On the average, SEEDIS is used about 500 times per month. Usage is equally divided between development and applications. The Populations at Risk to Environmental Pollution (PAREP) project, which is concerned with relationships between human health and environmental pollution, provides and uses data on mortality, cancer incidence, socio-economic characteristics, and air quality [3]. 1980 Census reports being produced for the Department of Labor will require incorporation of most of the 1980 Census of Population and Housing, bringing the size of the SEEDIS database to about 50 gigabytes (500 tapes at 6250 bpi) [4].

SEEDIS data currently available to the interactive user include 350 million individual data values on disk and over 5 billion data values on a tape-based mass storage system. Data are available for about a million distinct geographic areas. These include eighty different types of geographic entities (e.g., states, counties, census tracts, enumeration district/block groups, etc.).

The size of SEEDIS databases, financial constraints, and the need for local control over data stored at dispersed geographic locations prompted development of techniques for data retrieval and display in a distributed computing environment. SEEDIS meets the needs and resources of small groups in the research community who can afford a small computer but not the resources required for on-line storage of large databases, nor the costs of timesharing on a large mainframe computer. SEEDIS software is in the public domain; it runs in the standard DEC (Digital Equipment Corporation) VMS operating system on a VAX 11/780 computer. To access SEEDIS databases at LBL (Lawrence Berkeley Laboratory), DECNET hardware and software are required.

2. Initial Network Implementation

SEEDIS operates in a homogeneous network of DEC VAX computers and uses standard DECNET facilities. The network presently comprises some 50 minicomputers. There are currently nine VAX-11/780's running SEEDIS. These are located in the San Francisco Bay area, the state of Washington, Washington DC, and North Carolina. Program modules, area and data definition files, and geographic base map files (about 75 megabytes) are stored at each SEEDIS site, or "node." Selecting (i.e., specifying for retrieval) or displaying data (e.g., mapping) does not involve network access, so response time depends only on the local system load and the speed of the user's terminal connection.

This work was supported by the Office of Health and Environmental Research and the Office of Basic Energy Sciences of the U.S. Department of Energy under Contract DE-AC03-763F00098; and the Department of Labor, Employment and Training Administration under Interagency Agreement No. 06-2063-36.

2.1. Distributed Data Operations

After the user has specified data selections, SEEDIS automatically extracts the requested data values from local and remote files, copying them into a self-describing file in the user's working space. Standard DECNET facilities automatically provide shared access to archived data files (about 1 gigabyte) on disk packs mounted on two of the nodes in Berkeley. Except for response time, the difference between retrieval of locally-stored and remotely-stored data is not apparent to users.

DECNET naming conventions automatically permit transparent access to remote files without additional programming effort. For example, lblg::dba0:[mydir]xyz.dat is a file in directory "mydir" on disk drive dba0 on node LBLG. Since data are stored on a particular disk pack and not a particular drive, SEEDIS maintains tables specifying the name of the disk pack on which database is installed, (e.g., SEEDIS005). The VMS operating system automatically assigns logical names to locally mounted disk packs, so data can be directly accessed by disk pack location, (e.g., disk$seedis005:[mydir]xyz.dat).

Special software was written to extend the standard DEC capabilities to remotely mounted SEEDIS packs. For example, whenever SEEDIS is invoked at any node, a background process searches the network for disk pack SEEDIS005; if it is found on drive dba0 at remote node lblg, a local system logical name assignment is established to translate disk$seedis005 to lblg::dba0:. If the pack is not found, any previous assignment for disk$seedis005 is canceled.

Disk packs SEEDIS001 through SEEDIS005, containing on-line SEEDIS databases, are located at LBL and can be mounted by an operator on either of the nodes LBLG or LBLH. A program DSCHED, which can be invoked from any node, allows remote users to easily determine when a disk pack will be mounted, or to request future mounting.

2.2. Initial Implementation Limitations

The initial 1979 SEEDIS network implementation had several limitations. First, even for small requests, data extraction took 20 to 30 minutes for remotely stored data, as compared to 2 or 3 minutes when data were stored locally. The difference was due to overhead in underlying DECNET remote file access protocols, which were not well understood at the time the SEEDIS data extraction module was written. Second, only a small fraction (about seven percent) of all SEEDIS databases could be stored on disk packs and an even smaller fraction could be on line at any given time. Data which had originally been stored on an IBM photodigital mass storage device now reside only on tape. In the absence of another low cost mass storage device, new mechanisms were necessary to access the large amount of archival data. Finally, the original implementation did not provide for automatic updating of SEEDIS system tables on data locations across the network, so changes required intervention by a central database administrator. While this was tolerable initially, it was clearly preferable to give each node independent responsibility to alter physical storage locations of its own individual data sets in a way that could be automatically communicated to other SEEDIS nodes.

3. Distributed DBMS Enhancements

During the past year, a number of improvements have been made to overcome limitations of the initial network implementation. They include mechanisms for: local caching of data files to improve efficiency; linking to an automatic tape library (ATL) to make larger volumes of data accessible; node independence to facilitate automatic sharing of data among autonomous SEEDIS installations without the need for central control; improvements providing robust operation despite unreliable network connections; and automatic recording of all cache transactions for subsequent statistical analysis.

3.1. Caching

In order to speed up access to frequently-used data and to provide an automatic mechanism for allocating scarce on-line storage space to the most frequently-used data, a simple system of caching was introduced. Archived files containing data required by the user are temporarily copied in their entirety to a disk cache at the user's local node. Archived data are partitioned so that no single file occupies more than a small fraction of the total cache. Files remain in the local cache for shared use until the space is needed for a more recent request. Every file is marked with the date and time of last access; least recently used files are removed first. Each file's lifetime depends on its utilization and the size of the cache, which is set by the local system manager.

Precautions are taken to prevent deadlock or thrashing: (1) a user request is immediately rejected with a message if the data request will exceed the total available space in the cache; (2) user requests are completely processed one at a time; (3) recently requested or used data have a guaranteed minimum lifetime of several hours in the cache, regardless of the number of pending cache requests; (4) a safety margin of about 2000 blocks (one megabyte) is maintained for necessary housekeeping functions.

All cached files copied from archive locations reside in a "temporary" cache subdirectory. All cache updates are accomplished in batch mode by a pseudo-user CACHE. The date of last access of each file (plus a constant increment) is automatically maintained by the VMS operating system. Another portion of the cache consists of small "permanent" files which are periodically updated but never deleted. These files contain pointers to information at other nodes.

This caching scheme is largely transparent to SEEDIS users, but it has involved an important enhancement to the user interface. Following standard SEEDIS procedures for data selection, the user defines a geographic scope and level (for example California by county) and then selects desired data elements from one or more on-line data dictionaries. After data selection is complete, the user types "extract" to append the data values to

his/her working data set.

In the new caching implementation, the "extract" command first automatically copies entire archived files to the cache if they are not there already, and then extracts selected data from the cached files to the user's working directory. If the required data are not already in the cache, the user is warned to expect a delay. The user may choose either to wait or to put the process into the background by typing an interrupt character (control-Y). The user then can type "show" to check the status of the cache request, "cancel" to cancel the request and begin another unrelated SEEDIS task, "continue" to complete the requested extraction as soon as caching is completed, or "quit" to leave SEEDIS. In all cases the background caching process proceeds to completion. Re-entering SEEDIS an hour or two later, the user can extract the requested data without delay.

If the requested data reside on the automatic tape library (ATL) at LBL (see below), the cache request requires access to BKY, the Lawrence Berkeley Laboratory computer center operating system. The user is prompted for a BKY account number and password, if not already specified in the user's login command procedure. Interactive help is available for the new user who needs to open a new BKY computer account.

3.2. Automatic Tape Library Mass Storage

The initial implementation of SEEDIS on CDC computers in the mid-1970's made use of an IBM photodigital storage device, the "chipstore." When IBM discontinued support for that product in 1979, SEEDIS databases were moved to a tape-based mass storage "gettape-stotape" system (GSS). This system, developed at LBL, implemented a self-describing UNIX-like directory structure for tapes and optionally makes use of an Automatic Tape Library (ATL) connected to the CDC machines.

When SEEDIS was initially reimplemented on the Distributed Computer Network VAX's, there was no link to the ATL. Selected databases were installed on disk for the initial implementation. At present, installed data occupy 1 gigabyte on five disk packs. The 1979 network implementation accessed only files on disk packs mounted at nodes in the network. In order to access data on tape, the tape had to be manually mounted, copied to disk, and installed in SEEDIS, a time-consuming and labor-intensive process. Although SEEDIS tapes contained much useful data (including most of the 1970 U.S. Census), they were virtually inaccessible. SEEDIS use did not justify the number of disk packs, let alone disk drives, required to keep the data on line.

With anticipated arrival of 1980 census data, there was a need for low-cost, moderately quick access to mass storage. Although optical disks had seemed a likely answer in the late 1970's, that technology was still too costly and unreliable. In order to fill this need, the SEEDIS project proposed a network link from the VAX machines to the Computer Center's CDC computers, in order to access and make use of the Automatic Tape Library and its GSS mass storage tape file system.

Two-way communication with the ATL is accomplished by programs BKYSUBMIT and BKYCLAIM, which are installed on every SEEDIS node. BKYSUBMIT and BKYCLAIM use DECNET to talk to a special network node DGATE, which in turn communicates with the ATL over a high-speed hyperchannel link. The 1983 SEEDIS network implementation includes an interface to BKYSUBMIT and BKYCLAIM, including proper handling of the various errors that can occur. As a result, low-priority SEEDIS data are now gradually being moved to tapes on the ATL, freeing valuable disk space for more important files and caching.

3.3. Node Independence

One of the most serious drawbacks of the 1979 implementation was the difficulty of modifying archived data files. Every node had an identical copy of program modules, database lists and data dictionary files. Files at every node had to be modified if any changes were made to publicly installed data files. Obsolete data files could not be removed until new software and data dictionaries were installed on every SEEDIS node, a time-consuming process even with only nine nodes. With additional SEEDIS nodes planned for 1983 and beyond, a better solution was required.

One of the guiding principles of the 1983 SEEDIS network implementation has been node independence. Every node should have the ability to install its own data locally, which it may optionally share with other nodes on the network. When a data file is installed, modified, or removed at any node, new information must automatically propagate to every node that has access to that file. The procedures for installing data must be simple and robust enough that only minimal consultation will be required from LBL staff.

The 1983 implementation allows data to be installed at any node, whether or not that node is connected to other SEEDIS nodes on the network. Optionally, the installed data may be flagged for public access, in which case the data become available to remote users as soon a network connection is established. The existence of data is made known to other users through a summary data base directory, which may be printed off line or browsed on line. A copy of the on-line directory is maintained at every node as described in the following example.

Suppose a user at the ETADC node (in Washington, DC) installs or modifies a public-access data file. With the permission of the local system manager, s/he uses documented installation procedures to automatically modify certain files in the local subdirectory seedis/etadc. This portion of the file system contains all ETADC node-specific SEEDIS information. In particular, it contains pointers to permanent archive locations of data and documentation installed by ETADC users. (The data may actually reside elsewhere, for example on the ATL in

Berkeley, California).

The installation procedure invokes a batch process at ETADC, which in turn causes batch processes to be initiated at every other presently connected SEEDIS node. The subordinate processes modify files in the "permanent" portion of their local cache. For example, node RX in Seattle has a subdirectory cache/perm/etadc where it maintains current copies of small files describing ETADC-installed data (i.e., a copy of seedis/etadc from node ETADC). Conversely, node ETADC has a subdirectory cache/perm/rx where it maintains current copies of small files describing RX-installed data (i.e., seedis/rx at node RX).

When the network is down, there is no guarantee that the directory cache/perm/etadc at RX is a correct copy of seedis/etadc at ETADC. If the RX network connection is down at the time ETADC data are installed, that information is kept in a small file at ETADC, and SEEDIS periodically resubmits the same batch update request (once a day until successful). In addition to the broadcast of updates, each node regularly (once a day) checks all other connected SEEDIS nodes to bring information from other nodes in its own "permanent" cache up to date.

Periodically (once a day) at each node, the information in all the subdirectories cache/perm/(anything) is merged and reformatted, to form a global database directory (also in cache/perm). This global database directory is the primary source of information at each node for SEEDIS users and data retrieval software.

The list of known SEEDIS nodes is itself a file which is automatically maintained at every node. For example, a file in seedis/etadc at ETADC identifies ETADC as being a public-access SEEDIS node. When SEEDIS is installed at ETADC, it attempts to broadcast that fact to every node on the network; those which have installed SEEDIS automatically receive and record the information in their directories cache/perm/etadc. Even if ETADC is temporarily disconnected, it is remembered as a SEEDIS node in future broadcast attempts from other nodes. If SEEDIS is deinstalled at ETADC, the information is properly recorded at each node the next time it achieves a network connection with ETADC.

3.4. Robustness Considerations

The caching software needs to be unusually robust to cope with a still unreliable hyperchannel link and DECNET phone connections that may operate only a few hours a month. On several occasions when the hyperchannel was inoperative for an extended period, the requested data were automatically and correctly put in the cache when the link was restored three weeks later. Even such a delayed response is valuable to certain classes of remote users, provided the data are certain to arrive sooner or later without further attention. Users do not have to remain on line waiting for the data to arrive. Once in the local cache, data used with some regularity are likely to remain available for months or longer.

Another aspect of robustness concerns the ability of the system to correctly recover from power failures, system crashes, scheduled and unscheduled shutdowns, VMS system updates, and well-intended but incorrect actions by system managers. In general, the contents of disk files are the only reliable records left by an interrupted job -- batch queues may not survive system updates or system crashes. Each SEEDIS node maintains a record of a pending job it expects to find in the batch queue of every other SEEDIS node, together with the password required to resubmit that job if necessary. This job (which runs once a day at each SEEDIS node) performs routine maintenance operations and keeps alive its "clones" at all other connected nodes. As the job runs only a few minutes a day, the likelihood of its being removed from the batch queue (due to a crash while it is running) is small. The likelihood of such a disaster affecting every node simultaneously is negligible. In other words, the system becomes more robust as more nodes are added (like the brooms of the "Sorcerer's Apprentice!"). Once started at a node, it can be permanently turned off only by a deliberate action of the system manager, for example by deleting critical files or removing the login privilege of the pseudo-user CACHE.

4. Recording of Cache Transactions

Since January, 1983, transaction records of every cache request have been continuously recorded in a compact machine-readable form. Usage patterns are being statistically analyzed to isolate bugs and improve efficiency.

Between January and June, 1983, the caching mechanism was continuously tested via daily automatic submission of randomly generated requests. Two nodes connected via DECNET shared a common cache on a single disk. Files were routinely and correctly cached from the ATL; delays varied from 20 minutes to 20 days depending on the state of the hyperchannel link. Fewer than 1 percent of the requests failed, in all cases due to hardware error. On only three occasions did the system fail irrecoverably and require intervention -- twice when hyperchannel hardware malfunctions caused the cache to overflow, and once when disk hardware errors caused the batch queues of both nodes to be simultaneously destroyed.

Under normal day time load conditions, a typical small request involving the ATL takes about an hour -- 5 minutes to formulate and submit the request, 20 minutes in batch queues, 10 minutes to read the tape, 20 minutes to put the data in the cache, and 5 minutes to copy the requested data from the cache. Subsequent requests for the same data would require only 10 minutes -- 5 minutes to formulate the request and 5 minutes to extract the data. Some of these times will be reduced in the future by improving the efficiency of the software.

5. Conclusions

Major enhancements have recently been implemented to permit efficient and robust access to distributed data in SEEDIS. Specifically (1) an automatic caching mechanism provides local shared access to user-selected subsets

of SEEDIS databases; (2) automatic access to 50 gigabytes of archived data is achieved through a hyperchannel link to an automatic tape library; (3) data can be independently installed, modified, or removed at any node, with all changes automatically recorded in copies of a global database dictionary at every other node; (4) every node is responsible for initiating periodic housekeeping functions at every other node, so that the whole network is much more robust than any individual node; (5) a continuous log of every cache transaction is being recorded for statistical analysis. So far, caching has been implemented for only one SEEDIS database -- a portion of the 1980 Census that was too large to reside on disk. During late 1983 and early 1984, the mechanism will be implemented for most other major SEEDIS databases including most of the 1980 Census. Distribution of a new version of SEEDIS in 1984 will give remote users automatic access to a vastly increased database, with no increase in local disk storage requirements. At the same time, remote users will be able to install their own SEEDIS databases and make them mutually accessible to other SEEDIS nodes.

References

1. McCarthy, J. L., Merrill, D.W., Marcus, A., Benson, W. H., Gey, F.C., Holmes, H., and Quong, C., "The SEEDIS Project: A Summary Overview of the Socio-Economic, Environmental, Demographic Information System," Lawrence Berkeley Laboratory Report, PUB-424, Rev. May, 1982.

2. Merrill, D. "Overview of Integrated Data Systems: Context, Capabilities and Status," Lawrence Berkeley Laboratory Report, LBL-15074, October, 1982. In Proceedings of the 1982 Integrated Data Users Workshop, Reston, VA, October, 1982.

3. Merrill, D. and Selvin S. "Populations at Risk to Environmental Pollution (PAREP): Project Overview," 1976-1982, Lawrence Berkeley Laboratory Report, LBL-15321, December, 1982. Included in "An LBL Perspective on Statistical Database Management," H. Wong, editor, Lawrence Berkeley Laboratory Report, December, 1982.

4. Department of Labor, Employment and Training Administration and Lawrence Berkeley Laboratory. Report 1: "Population Characteristics: 1980 Census of Population," Lawrence Berkeley Laboratory Report, LBL-14636, April, 1982. Report 2: "Employment and Training Indicators: 1980 Census of Population," Lawrence Berkeley Laboratory Report, LBL-14637, April, 1982. Report 3: "Social Indicators for Planning and Evaluation." Report 5: "Equal Employment Indicators." Three additional reports are in preparation.

TECHNICAL INFORMATION DEPARTMENT
LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720