Distributed data bases
A summary of research

Mark E. Deppe and James P. Fry
Data Translation Project, Graduate School of Business Administration, University of Michigan, Ann Arbor, MI 48109, U.S.A.

The overall objective for distributed data bases is the sharing of data among several distinct but inter-connected computing facilities through an integrating mechanism. A review of the literature indicates that little progress has been made in this area due to the large number of technological problems involved. Some researchers have obtained analytical/theoretical results in the area of physical data allocation under the restrictive assumptions of static and known access patterns and independence between programs and data.

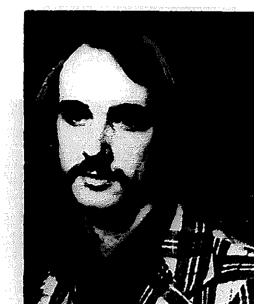
The remaining unsolved technological and operational problems include measurement and evaluation techniques, maintenance of multiple image files, and security/privacy. In the next five to ten years, significant benefits would accrue if data translation techniques, an integrated data base control system, and the integrated data base schema and physically distributed data issues were investigated.

Keywords: Distributed data bases, data distribution, directory allocation, data base synchronization, data base integrity, data base security, file allocation, data translation, update synchronization, data control system, data base deadlock, access control, data model, integrating schema.

Computer Networks 1 (1976) 130–138

1. Introduction

The advent of computer networks as defined by Roberts [30] brought about a great potential for sharing and distributing resources (data, programs, hardware, etc.) among heterogeneous computing facilities. Utilizing high capacity communications channels, the goal is to inter-connect dissimilar computer facilities to exchange information and to share resources. Unfortunately, the trend towards sharing resources has yet to be fully realized within today's technology. This paper focuses on one of the resource sharing goals of computer networking: the sharability of data, or more specifically, the ability for a number of data base management systems to co-operate with one another and thereby share the data under their control. The term distributed data base has been used to describe this area. Each researcher, ranging from Chu [14] to Johnson [25], however, ascribes different semantics to the area objectives. Following the work of Aschim [3] and Levin [27], we define a *distributed data base* to be a physical partitioning of a data base over possibly different computing facilities while providing an integrated access to the data. It appears that the main reason that the trend towards



Mark E. Deppe is a Research Associate on the Data Translation Project at The University of Michigan, investigating technologies and new applications for data base restructuring. He received a BS in computer engineering from The University of Michigan and an MS in computer, information, and control engineering in 1976.

James P. Fry is a Research Scientist associated with the Data Translation Project at The University of Michigan performing research in data base and program translation, restructuring and data base design and evaluation. Presently, he is national chairman of ACM Special Interest Group on Management of Data (SIGMOD) and active in CODASYL (Systems Committee and past chairman of Stored-Data Definition and Translation Task Group). Mr. Fry received his bachelor's and master's degrees from The University of Michigan.

distributed data bases has not yet been realized is due to a lack of an underlying technology for integration. Facilities such as language and data translation to provide communication among different computing environments are necessary to achieve this objective. Furthermore, the operational issues of multiple data base synchronization, update and deadlock, have yet to be solved.

The purpose of this paper is to explore the area of distributed data base technology and to summarize the state-of-the-art. The goal is to indicate the accomplishments, recommend future research areas, and identify researchers in this field.

The central goal of distributed data base technology is the sharability of data among diverse users using different computing facilities. Specific goals which are, in some cases, analogous to the decentralization concept in data processing include reliability and backup, efficient use of existing resources, distribution of data volume, partitioning to increase response time, and localization of data base controls.

It is interesting to observe that the basic issues to be resolved in the area of distributed data bases are similar in form and substance to the ones that have been facing data base management system researchers for the past 10 years. The myriad of distributed data base issues tend to fall into two categories: technological and operational. Covered in the technological category are the necessary mechanisms required to achieve integrated access and processing functions which include data translation, description and information translation, language translation including manipulation and interrogation, and data directory. The operational issues address the software design considerations and the performance implications of the distributed system. Included in this category are the issues of multiple data base synchronization, concurrency problems (such as update and deadlock), and distribution/partitioning problems.

2. Current research in data sharing

The large volume of technical literature on resource sharing in computer networks has been assembled in the bibliographies of Blanc [7] and Bernard [5]. Even a cursory review of the literature indicates the extreme pervasiveness of the data sharing goal on the network, the data manager, and the user interfacing technologies. Current published and unpublished research will be reviewed in terms of the approaches to data sharing, the design issues, and the control system implications.

2.1. Current approaches to data sharing

The benefits of a data sharing system may well be offset by the problems involved with its access and use thereby discouraging user participation. A primary goal of such a system is, therefore, to provide a common user interface which will guarantee ease of access to all users. There are two fundamental approaches to providing a common user interface: the centralized DBMS and the distributed DBMS.

The first approach to data sharing, the centralized DBMS, consolidates all files and DBMS functions into a single network node and generally assumes a single model of data and query language. The common user interface is therefore achieved through standardization. Furthermore the centralized nature of the approach eliminates many design problems inherent in distributed data sharing systems and requires only a modest level of network technology. However, a centralized DBMS is based upon a total system design approach to distributed data systems (Comba [17]). Consequently, this approach is not well suited to the evolutionary environment of a large, general purpose computing network, since the management of data and computing hardware typically is not controlled by a single enterprise.

The distributed DBMS is the second approach to data sharing. In a distributed system, data files and data base systems reside on different network nodes. In a distributed system, commonality may be achieved either through standardization or integration.

Standardization is similar to the centralized DBMS approach in that a single DBMS is used, thereby permitting only a single data model and query language. It differs from the centralized approach in that data files and management systems are distributed among the network nodes. This approach also relies in part on the concept of a total system design since homogeneous DBMS's are fundamental to its architecture. Nevertheless, it is more suited towards an evolutionary environment since the same DBM systems may already reside on several network host computers. A higher degree of network technology is required since provision must be made for the transmission of queries and data files.

The integrated approach to distributed data sharing differs from standardization in that different data base management systems are allowed. This approach permits different query languages and data models to be used but requires translation mechanisms to achieve a semblance of commonality. This approach is best suited to the evolutionary environment of a large general purpose computing network, but demands the highest level of network technology and data base control systems.

In an unpublished paper, Cashin [9] refers to the integrated approach as 'data base interworking'. Network file transfer and file access are given as current examples of interworking, since users express requests in terms of a local host system and are protected from the access languages and conventions of the remote system. The major focus of discussion concerns the problems of translating between both query languages and data models. Cashin proposes the formulation of a common query language data model to be used as a communication mechanism between data base systems. The relational model of data is suggested as a possible common data model.

Aschim [3] classified organizational structures of data base networks by two methods according to the geographical location of files and directories, and also according to the distribution and types of data management systems used in the network. Important issues inherent to each classification are discussed and some solutions advanced. Translation problems resulting from a distributed integrated DBMS approach are investigated in light of standard and non-standard interfaces between systems. Several other issues involving translation between host computers are discussed.

Shoshani and Spiegler [34] investigate three approaches to data sharing; the centralized data management system (CDMS), the standardized (SDMS), and the integrated (IDMS). Advantages and disadvantages of each configuration are discussed. One important advantage of the CDMS is that translation of data and languages is not required and therefore this simplistic approach should provide good access time; however, the approach inhibits the development of new data management systems and would be impractical to implement since different DBMS's are often designed with different goals in mind. With the SDMS, failure of one node could not cause failure of the entire system. Like the CDMS, the SDMS approach inhibits DBMS development and does not permit existing information to be shared. IDMS is the most evolutionary approach but incurs the additional burden of translation. Development of a common data management language is proposed as a viable solution to the translation problem.

The area of data file translation has been investigated by Merten and Fry [28] and a general methodology based upon a high level data description language developed. An implementation of a prototype translator within this framework has demonstrated the feasibility of the approach and has provided further insights into the area (Birss and Fry [6]).

2.2. Program and data distribution

Program and data distribution involves both the logical and physical partitioning of software resources. In the Ph.D. dissertation by Levin [27], the state-of-the-art in distributed data bases was extended in the program and data allocation area. A model was formulated that considered the effect of the dependencies between programs and data and their optimal allocation in a network. The allocation problem was partitioned into three levels for solution. The first level assumed that the access patterns were static and known over time. A zero-one linear programming solution was developed for the optimal allocation of files. In the second level, the assumption that the access request patterns were known was relaxed and a dynamic programming solution approach was formulated. The third and final level addresses the situation where the access request patterns are unknown initially. A statistical procedure for the estimation of these patterns was developed and incorporated into the file allocation model. This would be useful in the adaptive reassignment of files in the network.

Sutherland [35] in a BBN report recognized two fundamentally different approaches to the problem of physical data allocation in a distributed data base system. The first provides for the maintenance of identical copies of the data base at each network node, while the second allows for distributed, nonoverlapping segments. These two approaches represent the extremes of a spectrum of possible allocation schemes which involve varying degrees of redundancy. The goals of a data allocation scheme involve several issues. Distribution should provide protection against individual component failure thereby increasing the reliability of the system and providing for backup data files. Furthermore, data should be distributed in a natural fashion (geographically near its greatest use) such that efficient access of the data is realized.

Sutherland also addresses program allocation as a dynamic process which is independent of a particular machine. Program allocation occurs at execution time, such that processing occurs on a machine most suited to handle the task under current load configurations. Two approaches to the problem are discussed. The first requires the constant collection of host status information to provide global knowledge of the network to a selection and routing process. The second approach requires the local host to broadcast a 'request for service' to all appropriate machines and choose among those responding.

Pioneering work in the area of physical data distribution has been provided by Chu [14] in which he investigated a linear programming solution to optimize the allocation of files in a network. Under the assumptions that 1) the number of file copies is known, 2) queries are routed to all files, 3) query patterns are known, and 4) a Poisson queuing discipline is assumed, he formulated a zero-one linear programming solution. It is well known that such a solution approach is only valid for small problems since the number of variables and constraints increase rapidly as the number of nodes in the problem increase. Whitney's Ph.D. dissertation [36] addressed the broader problem of computer communication system design but applied Chu's approach to the optimal allocation of files. Casey [10] relaxed limitations set by Chu's model and showed that the proportions of update traffic to query traffic determine an upper bound for the number of file copies which are maintained in the network.

One aspect of data distribution, the location of the file location directory or catalog, is the topic of current research by Wesley Chu at the University of California at Los Angeles [1]. This research is directed towards the analysis of directory location schemes under various situations. Steve Kimbleton at the University of Southern California is currently involved with the problem of network file allocation [1]. Kimbleton has recently developed an 'optimal' assignment policy based on the minimization of process execution costs. The resulting assignment algorithm he claims is cheap, easy to implement, as well as being intuitively acceptable.

2.3. Distributed file system

A distributed file system is dependent upon functions provided by operating systems, data base management systems, and computer network systems. Research in any of these areas therefore is applicable to the area of distributed data bases. The research by Sutherland [35] for BBN responded to several important design issues. One issue addressed network transparency as a means to provide a uniform degree of access to all network resources. This would be accomplished using a network control system with a high degree of automation to reduce or remove explicit network interactions by a user. A concept based on the trapping mechanism of the TENEX system was proposed as one solution for implementation. Several other issues dealt with the maintenance and update of multi-image files on the system. One solution proposed involved the collection of global system update information. Persistent processes (which insured the eventual completion of tasks assigned to temporarily non-functioning machines) and time stamping of data were also investigated as possible solutions.

In a recent paper, Johnson and Thomas [25] developed a method for synchronizing multiple updates in a multiple copy data base environment. Under the assumption that each data base manager is responsible for updating his copy of the data base and that complete communication exists among the data base managers, a time stamp solution is employed. The key feature of their approach to maintaining consistency and synchronization of the data base is to append a quintuple to each element which contains such information as time created, time last modified, site originating change, etc. This approach allows several sites to maintain multiple copies of a data base in a consistent state. It should be noted, however, that the time used is in reference to the originating site (the network has no time synchronization of clocks as yet) which could potentially cause several updates to be lost. Additionally, Johnson places some severe restrictions on the types of modifications that are allowed.

Current research at the University of Illinois is investigating the problem of maintaining multiple data base copies in a network environment [1]. Three primitive operations are proposed and several algorithms are provided as solutions. Algorithms involving clock synchronization or modification constraints have been avoided as much as possible. Problems concerning network failure recovery are also under study. Causes of failure are classified and several solutions proposed.

In a related effort (also at Illinois) resilient protocols for computer networks are being analyzed [1]. The purpose of this study is to determine how network communication protocols can be designed and implemented so that they are resilient to failures and abuse by aberrant or malicious software. Analysis of the ARPANET file transfer protocol has provided an initial indication as to the nature of the problem. To structure the area and decrease complexity, a layered approach to protocol resiliency was employed whereby all underlying protocols involved are assumed to be resilient. One other assumption permits a protocol process to assume all errors are due to the communication media or to the remote process. Initial findings have identified several problem areas concerning protocol resiliency which include connection termination, format inconsistencies, restart facility, and message synchronization.

2.4. Security and privacy

The problems concerning security and privacy in a single host system are severely intensified when extended to networks. In his BBN report, Sutherland [35] favors the implementation of a uniform user identification scheme and investigates the concept of 'authentication sites' which would initially verify a user's identity. Upon successful log-in, an authentication process would be assigned for the duration of the session and would automatically provide access to network resources where permitted.

2.5. Deadlock

Deadlock detection and prevention, like security, is an additional DBMS issue which is intensified when examined in the light of a distributed data base system. Important research by W. Chu and G. Ohlmacher [15] has investigated deadlock prevention and detection mechanisms in distributed data bases. Three methods of deadlock protection are proposed. The first is a simple prevention mechanism which allows a process access to a resource only when all requested resources have become available. The second method, called the process set prevention mechanism, allows a process to proceed and access resources as needed except when the possibility of deadlock is detected. The third method provides a deadlock detection mechanism based on the analysis of process and resource lists. Chu and Ohlmacher conclude that although the simple mechanism is superior in most applications, the process set prevention and detection mechanisms provide for efficient file utilization and greater flexibility and may therefore be preferable in some cases.

In a paper by Aschim [3], deadlock problems associated with exclusive file lockout are investigated. In cases where individual files are not logically interrelated, two solutions are proposed; a two-stage reservation system which requires the assignment of unique priorities to all users or processes, and a fixed sequence reservation system which requires assignment of unique numbers to all resources.

2.6. Management of network name and relation space

Two issues are of prime importance. First, the techniques used to identify and locate files, (such as a directory) and, second, the method by which the internal logical structure of each data file is obtained (similar to the schema concept).

Sutherland envisions the management of the network name space as a binding between names of 'entities' and their location within the network. Two approaches to this issue are discussed: the 'full access name' and the cataloging function. The full access name accomplishes the binding by simply providing a network location field within every file name thereby specifying the location explicitly. The cataloging function provides for a distributed data sharing system to maintain the name-to-location bindings internally such that a user need not know the network location of the file.

2.7. Distributed DBMS control system

The control system of a distributed data base management system may exist as an individual process, or could, in fact, be so integrated into the network systems as to be unrecognizable as a separate entity. The management of distributed data bases on a computer network is designated as a subordinate function of the resource sharing executive network control system in Sutherland's approach. Those network functions which may be directly applied to the management of distributed data bases are separately singled out. Sutherland discusses several control problems associated with two distributed data base designs. The first design provides for the maintenance of identical data base copies at several sites, while the second design consists of non-overlapping segments. The emphasis of work concerning the first type of data base has been the development of control systems which automatically maintain the redundancy. One issue of discussion is the problem of updating multiple copies of the same data. Several features of a control system with such capabilities are discussed.

One such feature involves the collection of network status information to provide information to modification processes. Another feature discussed is the 'persistent' process, a program designed to queue updates to data bases residing on temporarily non-functioning hosts to insure that all updates are delivered only once and in the proper sequence. The use of time stamps as a solution to the update problem is believed to be fundamental to the management of distributed data bases.

Three major control system issues of non-overlapping segmented data bases are discussed:

(1) the maintenance of a global catalog;

(2) ensuring duplicate entries do not occur;

(3) ensuring data base entries are processed only once.

3. Conclusions as to the state-of-the-art

The research reviewed is divided into two categories: 1) theoretical or analytical results, and 2) practical software implementations.

Theoretical and analytical results

Examination of the approaches to distributed data sharing systems has been investigated in several different efforts and classification schemes which have been developed are rather complete. In the area of program and data distribution, theoretical results have indicated several approaches to data and program allocation, however, application to existing systems is difficult due to the problems involved with measuring user/process access patterns and interdependencies of data. A fairly complete analysis of distributed file system designs has been provided, but actual implementations are dependent upon technological advances in the area of computer networks in general. The major problem in this area concerns the update problems resulting from multi-image files. The issue of deadlock is fairly well understood and several algorithms have been proposed to solve this problem in a distributed data sharing system. Few research results have been directed towards the topic of security and privacy other than to acknowledge the serious problems involved in a distributed system. The concept of the network name space has been researched in light of the master file location dictionary or directory. Research has yet to closely examine the aspect of the network relation space, (a data structure which permits relations across file boundaries) and the creation of a master data schema. Several research efforts have touched upon the role of a master distributed data base control system, but as yet do not agree upon the principal functions of such a system.

Software implementations

One of the research results from BBN has been the development and implementation of several distributed data and program sharing systems. RSEXEC, the Resource Sharing Executive system is representative of the state-of-the-art of practical software implementations. RSEXEC is an experimental distributed executive system which functions to integrate the operation of ARPA network TENEX hosts. Several advanced concepts of distributed resource sharing have been implemented. One major feature of RSEXEC is a distributed file system which spans host computer boundaries and supports uniform file access and automatic maintenance of multi-image files (subject to the constraint that all files are operational at the moment of update). It also supports the concept of device binding such that all references to a device are in fact directed to a remote host.

RSEXEC is currently used by TIPS on the ARPA network to provide information services to network users. The design involves a "broadcast" initial connection protocol (for selecting a host for services) and a mechanism whereby multiple file images are consistently maintained. The important aspect of this implementation is that the feasibility of using large hosts to provide extended capabilities to smaller hosts has been demonstrated.

McROSS, another BBN software implementation, is a system capable of running distributed among a number of host computers which simulates and analyzes air traffic situations. McROSS also has the capability to redistribute its operational parts among network hosts without interfering with the simulation. This capability evolved from the application of techniques for dynamic reconfiguration developed in an earlier software effort, CREEPER; a demonstration program which could migrate from computer to computer within the ARPA network.

4. Areas requiring additional research

The goal of data sharing in a multi-computer network intensifies the myriad of existing problems in data management and introduces a new class of distributed management problems. The existing approaches to the single system data management issues of privacy, integrity, concurrent access, user interface, data model, data organization, etc. are challenged by the distributed nature of the data. The complexities and interaction of the numerous components introduce new issues at all levels in the system design process. Although some techniques [35] have been developed for special cases involving a common operating system and computer, research is needed "to develop a coherent methodology for the design, implementation, and management of distributed data bases." An intensive research program is needed to:

(1) Decompose the underlying approaches to distributed data bases. in order to identify the essential components and their interrelationships

(2) Develop a translation technology to include data language and data model translations

(3) Design an integrating schema for the network data base

(4) Design of an integrated data base control system to handle update control, access control, selection, and integrity

(5) Analyze and evaluate alternative data base organizations for their effectiveness for a coherent methodology

Analysis of the essential components of distributed data systems

The area of distributed data sharing lacks a firm logical structure upon which research may be based. Previous attempts to structure the area have succeeded only in revealing the high degree of integration involved. As a result, subsystems are difficult to define and identify further complicating research efforts. Specific investigations should be directed towards providing functional definitions for 1) operating systems, 2) data base management systems, 3) communications networks, 4) resource sharing executive systems, and 5) distributed data base management systems.

These functional definitions will aid in the identification of basic sub-system responsibilities and should provide insights to the natural structure of the area.

Development of translation technologies

Optimization schemes which involve dynamic resource allocation will depend on the ability of the distributed system to store data and execute programs at any network node. Since the integration of different systems is the most evolutionary approach to distributed data sharing systems, the ability to move data and programs from node to node will, in turn, depend on translation technologies in 1) data translation, 2) query translation, 3) data model translation, 4) programming language translation, and 5) operating executive command language translation.

The areas which specifically demand further investigation are data query and model translation mainly since these technologies must be developed to achieve integration of data base management systems, an initial goal of distributed data sharing systems.

Design of an integrating schema

Several research efforts have been directed towards the concept of a master catalog system to identify and locate data files in a distributed system, and further investigation is required to develop search strategies and optimal configurations. One additional area which demands investigation concerns the concept of a master or integrating schema which describes the logical structures of all data files existing in the distributed data base. This feature, in conjunction with the master directory, permits the determination of a data file's logical structure as well as its identity and location, and could possibly be essential to the development of query and data model translation schemes. The existence of master schema also permits the logical relation of data across file boundaries. All files in the network could then be considered as areas within a single large data base. The concept of a multi-image file could be extended in such a system to provide for multi-image records which are common to several different data files. Efforts should be initiated to determine the full ramifications of such an approach.

Design of an integrated data base control system

The objective of a distributed data management system is to provide powerful resource sharing capabilities without compromising the basic functions currently provided by single host systems. Current investigations indicate problems in the area of multiple image files. Research is therefore needed to determine the ramifications of multiple image files upon basic DBMS functions.

Functions which require investigation include:

(1) Data definition – specifications must be developed to describe multi-image file structures.

(2) Interrogation – selection schemes to determine which file copy to use for a given application.

(3) Update – problems with synchronization,

deadlock, and data integrity must be investigated. This is a critical area.

(4) Security – ramifications of the network upon security must be evaluated. This is also a critical area.

(5) Creation – specifications must be developed to describe expected access requirements to the system.

Furthermore, the complexity of the distributed data sharing design demands research and development towards an integrating control system. The object of such a system would be to co-ordinate and control the functions of systems subordinate to itself such as operating systems, data base management systems, network communication systems, and translation systems. All functions required of a distributed data sharing system would be accomplished by the individual subordinate processes under control of the integrating system. In some respects research has already discovered the need for such a system. Additional research should be directed towards this area to permit efficient and optimal operation of the distributed system once the underlying technologies become available.

Analysis of design strategies

The advent of distributed data sharing technologies will call for the development of sophisticated techniques to measure and analyze system performance with regard to different optimization strategies. The evolution of these techniques is a critical factor in the evaluation of both static and dynamic distributed system configurations. Evaluations of this type are required by systems designers (as well as operating system procedures) to select resource and job configurations for the distributed data sharing system.

Critical areas

Further investigation into the area of multi-image file maintenance (with special emphasis towards update, data integrity and security/privacy) is critical, as computer network technologies have developed sufficiently to allow practical implementations of distributed file systems. In addition, translation techniques and master schema concepts will play important roles in the development of integrated data base control systems in the next few years and should be investigated. Furthermore integrating data base control system designs should be investigated as a technique to co-ordinate and control the sub-systems within the resource sharing network.

Bibliography

[1] P.A. Alsberg et al., Preliminary Research Study Report, Research in Network Data Management and Resource Sharing, Center for Advanced Computation, University of Illinois at Urbana-Champaign, Urbana, Il. 61801, 19 May, 1975.

[2] P.A. Alsberg et al., Survey Report, Research in Network Data Management and Resource Sharing, Center for Advanced Computation, University of Illinois at Urbana-Champaign, Urbana, Il. 61801, 19 May, 1975.

[3] F. Aschim, Data base networks – An overview, Management informatics, 3 (1) (1974) 12–28.

[4] G G. Belford, Technology Summary, Research in Network Data Management and Resource Sharing, Center for Advanced Computation, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, 19 May, 1975.

[5] D. Bernard, Intercomputer networks – An overview and bibliography, M.S., Thesis, Moore School of Electrical Engineering, University of Pennsylvania, 1973.

[6] E. Birss and J.P. Fry, Generalized software for data translation, Proc. 1976 National Computer Conference, (AFIPS Press, 1976).

[7] R.P. Blanc, Review of computer networking technology, NBS Technical Note 804, January 1974.

[8] G.M. Booth, The use of distributed data bases in information networks, Proc. 1st Int. Conf. on Computer Communication, Washington, DC, 24–26 Oct., 1972. Published: S. Winkler, ed., Computer Communication Impacts and Implications International Conferences on Computer Communication, 1972, pp. 371–376.

[9] P. Cashin, Data base interworking, Unpublished Working Paper, National Physical Laboratory, Teddington, Middlesex, England, 1974.

[10] R.G. Casey, Allocation of copies of a file in an information network, Proc. Spring Joint Computer Conference, 1972 (AFIPS Press, Vol. 40, 1972).

[11] R.G. Casey, Design of tree networks for distributed data, AFIPS Conf. Proc. 42 (1973) pp. 251–257.

[12] R.G. Casey, Design of tree structures for efficient querying. CACM 16 (1973) 549–556.

[13] W. Chou and H. Frank, Routing strategies for computer design, Proc. 22nd MRI Symp. on Computer-Communications Networks and Teletraffic, Polytechnic Institute of Brooklyn, New York, 1972 (Polytechnic Press, New York, 1972).

[14] W.W. Chu, Optimal file allocation in a computer network, in: N. Abramson and F. Juo, eds., Computer Communications Networks (Prentice Hall, Englewood Cliffs, 1973) pp. 82–94.

[15] W.W. Chu and G. Ohlmacher, Avoiding deadlock in distributed data bases, Proc. ACM National Conference, Nov. 1975, pp. 156–160.

[16] W.W. Collier, Asynchronous interactions on shared data, IBM SIGOPS, 1974.

[17] P.G. Comba, Needed: distributed control, Proc. Int. Conf. on Very Large Data Bases, Framingham, MA, 22–24 Sept., 1975 (ACM, New York, 1975).

[18] Data computer Project. Datalanguage, Working Paper No. 3 (Computer Corporation of America, Cambridge MA, 1971).

[19] Datacomputer Project. Further datalanguage design concepts, Working Paper No. 8 (Computer Corporation of America, Cambridge, MA, 1973).
[20] Datacomputer Project. Datacomputer version 0/11 User Manual, Working Paper No. 10 (Computer Corporation of America, Cambridge, MA, 1974).
[21] D.J. Farber and F.R. Heinrich, The structure of a distributed computer system – the distributed file system, Proceedings of the 1st International Conference on Computer Communication, Washington, DC, 24–26 October, 1972. Published S. Winkler, ed., Computer Communication – Impacts and Implications International Conferences on Computer Communication, 1972, pp. 364–370.
[22] A.N. Habermann, Prevention of system deadlocks, Comm. ACM, 12 (7) (1960) 364–377.
[23] A.N. Habermann, Synchronization of communication processes, Comm. ACM, 15 (3) (1972) 171–176.
[24] E. Holler, Files in computer networks, first European workshop on computer aries, May 1973, IRIA Frame, pp. 381–396.
[25] P.R. Johnson and R.H. Thomas, The maintenance of duplicate databases, NIC #31507, Jan. 1975.
[26] P.F. King and A.J. Collymeyer, Database sharing – an efficient mechanism for supporting concurrent processes, AFIPS 42 (1973) 271–275.
[27] K.D. Levin, Organizing distributed data bases in computer networks, PhD Dissertation, University of Pennsylvania, 1974.
[28] A.G. Merten, and J.P. Fry, A data description language approach to file translation, Proc. 1974 ACM-SIGFIDET Workshop (ACM, New York, 1974).
[29] R. Peebles, Design considerations for a distributed data access system, PhD. Dissertation, Moore School of Electrical Engineering, University of Pennsylvania, May 1973.
[30] L.G. Roberts, Access control and file directories in computer networks, Proc. 4th Annual IEEE Int. Conf. on Communications, March 1968.
[31] L.G. Roberts and B.D. Wessler, Computer network development to achieve resource sharing, Proc. Spring Joint Computer Conference, 1970 (AFIPS Press, New York, Vol. 35, 1972).
[32] J.E. Shemer, and A.J. Collymeyer, Database sharing: A study of interference, roadblock and deadlock, ACM SIGFIDET Workshop on Data Description, Access, and Control, 1972, pp. 141–163.
[33] A. Shoshani and A.J. Berstein, Synchronization in a parallel-accessed data base, CACM 12 (1969) pp. 604–607.
[34] Shoshani, A. and Spiegler, I., The integration of data management systems on a computer network, NIC #15717, 1973.
[35] W.R. Sutherland, Distributed computation research at BBN, Vol. 111, BBN Technical Report 2976, Dec. 1974.
[36] V.K.M. Whitney, A study of optimal file assignment and communication network configuration, PhD Thesis, The University of Michigan, 1970.