Distributed Computing Systems: an Overview

1977

Haim Schwarzkopf
University of Central Florida

Find similar works at: https://stars.library.ucf.edu/rtd

University of Central Florida Libraries http://library.ucf.edu

Part of the Engineering Commons

STARS Citation

https://stars.library.ucf.edu/rtd/376

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of STARS. For more information, please contact lee.dotson@ucf.edu.
DISTRIBUTED COMPUTING SYSTEMS: AN OVERVIEW

BY

HAIM (JIMMY) SCHWARZKOPF
B.S.E., Florida Technological University, 1976

RESEARCH REPORT

Submitted in partial fulfillment for the requirements for the degree of Master of Science in Engineering in the Graduate Studies Program of the College of Engineering, Florida Technological University

Orlando, Florida
1977
DEDICATED TO MY PARENTS

Dr. and Mrs. Bedrich Schwarzkopf

who made it all possible
DISTRIBUTED COMPUTING SYSTEMS: AN OVERVIEW

ABSTRACT

by
Haim (Jimmy) Schwarzkopf

Associative processors, parallel processors, content addressable parallel processors, networks, and other architectures have been around the computing scene as "Distributed Processing", for some time now. Several hundred papers have been written discussing their use and design but so far no academic work has tried to summarize the field called "Distributed Processing" using a systems approach.

This research report attempts to remedy this lack. It attempts to gather into one place information that existed as of late 1976 in a format easily understandable by managers and systems engineers. The report deals also with certain issues of centralization and decentralization of EDP (Electronic Data Processing) facilities, created by the introduction of distributed computing systems into industries and businesses.
ACKNOWLEDGEMENTS

I would like to thank the members of my committee, Dr. C. S. Bauer, Dr. H. I. Klee, and Dr. B. W. Lin, for their kind attention and scholarly advice. I am especially indebted to Dr. Christian Bauer for the many hours spent advising me on the development and presentation of this report. Not only did he give freely of his time in helping me on this report, but he was responsible for getting me interested in computers in particular, and systems science in general.

Finally, I would like to thank the entire C.O.E. faculty for the excellent program of studies that has been offered to me. Special thanks to Dr. George F. Schrader, Chairman of the IEMS Department, who put up with me for the last four years and provided teaching assistantships which helped to support me during this period.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>vii</td>
</tr>
<tr>
<td>1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2. DISTRIBUTED SYSTEMS</td>
<td>3</td>
</tr>
<tr>
<td>3. HORIZONTALLY DISTRIBUTED COMPUTERS</td>
<td>6</td>
</tr>
<tr>
<td>(a) Parallel Processor</td>
<td>6</td>
</tr>
<tr>
<td>(b) Associative Processors</td>
<td>9</td>
</tr>
<tr>
<td>4. VON-NEWMAN MACHINE</td>
<td>10</td>
</tr>
<tr>
<td>5. VERTICALLY DISTRIBUTED COMPUTERS</td>
<td>12</td>
</tr>
<tr>
<td>6. BASIC NETWORK TYPES</td>
<td>14</td>
</tr>
<tr>
<td>(a) Point-to-Point</td>
<td>14</td>
</tr>
<tr>
<td>(b) Multipoint</td>
<td>14</td>
</tr>
<tr>
<td>(c) Centralized or Star</td>
<td>15</td>
</tr>
<tr>
<td>(d) Tree</td>
<td>16</td>
</tr>
<tr>
<td>(e) Loop or Ring</td>
<td>17</td>
</tr>
<tr>
<td>(f) Multistar</td>
<td>18</td>
</tr>
<tr>
<td>7. DISTRIBUTED PROCESSING</td>
<td>19</td>
</tr>
<tr>
<td>8. HARDWARE COMPONENTS</td>
<td>21</td>
</tr>
<tr>
<td>9. DISTRIBUTIVE DATA BASES</td>
<td>27</td>
</tr>
<tr>
<td>10. A REPRESENTATIVE DISTRIBUTED PROCESSING APPLICATION</td>
<td>31</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>11. CENTRALIZATION vs. DECENTRALIZATION OF</td>
<td>36</td>
</tr>
<tr>
<td>COMPUTER SYSTEMS</td>
<td></td>
</tr>
<tr>
<td>12. CONCLUSIONS</td>
<td>46</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>48</td>
</tr>
</tbody>
</table>
LIST OF TABLES

TABLE

1. General and Organizational Considerations; Advantages .......................... 38

2. General and Organizational Considerations; Disadvantages ......................... 39

3. Cost Factors; Advantages ....................................................... 40

4. Cost Factors; Disadvantages ..................................................... 41

5. Personnel Considerations; Advantages ............................................. 42

6. Technical Considerations; Programming; Advantages ...................... 43

7. Technical Considerations; Programming; Disadvantages ..................... 44

8. Technical Considerations; Operations ........................................ 45
LIST OF FIGURES

1. Division by Types of Distributed Processors ........... 5
2. Array Computers ............................................ 7
3. Multiprocessor Computer .................................... 8
4. Multicontrol Computer ....................................... 9
5. von-Neumann Machine ....................................... 10
6. Point to Point Network ...................................... 14
7. Multipoint Network .......................................... 15
8. Centralized or STAR Network ............................... 16
9. Tree or Hierarchical Network ............................... 16
10. Ring or Loop Network ....................................... 17
11. MultiStar Network .......................................... 18
12. Distributive Processing Environment .................... 35
1. INTRODUCTION

The field of Distributed Processing is eliciting a great deal of interest at the current time. The task undertaken by this research report on Distributed Processing was one of searching and culling the literature on three major topics:

* Distributive Systems and Computer Networks
* Distributive Data Bases
* Centralization vs. Decentralization of computing systems

Distributed systems hold a special place among currently feasible computer configurations, simply because they present a new and attractive alternative to totally centralized or decentralized systems. Unfortunately, the term "distributed processing" means different things to different people. In the first sections of the paper a brief explanation of the different hardware configurations are given.

Distributed data bases are a very important part of the successful Management Information System. They are created by taking portions or subsets of the overall corporate data base, and putting them out in the remote locations. The remote site corresponds to where the data is created and used, and where decisions are made based upon it. Exception and summary data, which the headquarters location needs for its data base can be retrieved from the remote
sites at appropriate time intervals. A part of the paper describes and explains Distributed Data Bases and its uses.

The organizational authority structure is very important in determining the chosen information system configuration. Those organizations accustomed to central control move earliest and most strongly to centralized data processing; those most devoted to decentralization move slowly, carefully and with maximum compromise. The last section of the report covers the EDP related issues in centralization or decentralization of companies affecting Distributed Processing.
2. DISTRIBUTED SYSTEMS

During the research upon which this report is based, it was found that highly parallel processors, computers-on-a-chip, networks, intelligent terminals and others have all been described as "distributed" systems. For the purposes of discussion and ease of understanding, the systems covered by the term distributive processing will be divided in two main groups: (a) horizontally distributed systems, and (b) vertically distributed systems.

Before defining the two main groups of distributed machines, the difference between a real DISTRIBUTIVE SYSTEM and one that is only DISTRIBUTED should be noted.

The phrase "DISTRIBUTED PROCESSING" stands for the use of a DISTRIBUTIVE SYSTEM, which entails the segmenting of its data bases and distributing its processing among smaller modules. This system can be either distributed geographically or the data base segments and the processor modules could be clustered in only one location.

There are many ways of accomplishing the distribution of processing. The easiest and most clear division is in the amount of processing that takes place simultaneously on the "same" application or program. This concept was used when dividing distributed processing in two main groups.

Horizontally distributed systems are those systems which can process data blocks of the same application simultaneously, while
vertically distributed systems process different parts of the data, at different levels, sending results for further computation from one to another of the processors.

Figure 1 depicts how these processing concepts have been classified for the purposes of this report.
Vertically Distributed Processing
- Point to Point
- Multipoint
- Star or Centralized

Hierarchical or tree

Loop or Ring

Multi-star

Horizontally Distributed Processing

Non-Associative Parallel
- Array
- Multiprocessor
- Multi-control

Associative Parallel

Non-Parallel

Fig. 1. Division by Types of Distributed Processors
3. HORIZONTALLY DISTRIBUTED COMPUTERS

This family of computers are the newest and the most misunderstood architectures. As is seen in Figure 1, it is made up mainly of parallel and associative processors with some overlapping.

(a) Parallel Processors

A very rough definition of parallelism is that of putting together N computers to form a supercomputer. This can be done in terms of either parallelism within the instruction stream or parallelism within the data stream or both. This combinations of parallelism produces 3 types of systems: multicontrol, array, multiprocessor systems, respectively (Kuck, 1977).

Array computers: this type of a system operates on vectors as basic units of information. Its parallelism derives from a parallel data-stream with a single instruction stream. The processing power is distributed in that while it executes a single instruction stream (with loads, adds, stores and branches of a serial von-Neumann computer), it manipulates whole vectors of data simultaneously, as shown in Figure 2.

An example of this kind of computer is the ILLIAC IV computer (Thurber and Wald, 1975).
Multiprocessor Computer: this type of a system consists of N complete computers plus interconnections for passing data and control information among the computers. It derives its parallelism from having both a parallel data stream and a parallel instruction stream. This computer is shown in Figure 3.

An example of this type of system is a machine developed at Carnegie Mellon University (Enslow, 1977).
Multicontrol Computers: In this type of computers each operand is operated upon simultaneously by several instructions. It derives its parallelism from having a multiple instruction stream but a single data stream. This computer is shown in Figure 4.

An example of this type of system is the CDC STAR computer (Bell, 1971).
(b) Associative Processors

A very rough definition of ASSOCIATIVE PROCESSORS, also called Content Addressable Computers, is any machine in which the processing units (or processing memory) are addressed by a property of the data contents instead of the memory address itself (Yau and Fung, 1977). This type of computer will be important in the next generation of business machines (Foster, 1976). Associative processors can be either parallel or serial. Examples of associative-parallel computers include the ILLIAN IV and the Goodyear STARAN machines (Higbie, 1976).
4. THE VON-NEUMANN MACHINE

In 1946, John von Neumann, a mathematician from the Institute for Advanced Study at Princeton University described a basic philosophy of computer design (Stone, 1975). Almost everything concerning computer design that von-Neumann discussed in his paper has been incorporated in modern computers. Thus, it is often stated that the basis for the design of computers is the "von Neumann Machine".

This machine works on the basis of a single data/single-instruction stream sequential organization (Bell, 1971). The processor has a control unit that identifies input information either as data or instruction, both of which have to be stored in a sequential order. These today are the most common machines and are
the building block for the vertically distributed computers. Vertically distributed computers are the real force behind distributed processing (March, 1976), that is why the rest of the research focused nearly in its entirety in this kind of system. From this point on, in the paper, the terms Distributed Processing and Vertically Distributed Processing will be used interchangibly.
5. VERTICALLY DISTRIBUTED COMPUTERS

This family of distributive computers, better known as computer networks can be defined as an interconnected group of computers, where each either acts as a processing system or as a communications control system. Computers are generally of the single data stream/single instruction stream.

The computers in operational control of the network can be divided in two groups (Nielsen, 1974):

(a) main-site, or host computers
(b) remote computing systems

The host processors in the network perform major computation, control data bases, and generally supervise operation of the network. They can share such resources as programs, data bases, and memory space.

Remote computing systems are systems with access to the host processor, that perform a minor part of the processing before sending the "edited" data to the host processor.

The communications control computers are devoted exclusively to network control functions. These functions include line control, error checking, message formatting, message switching, and data concentration.

In addition to the processing and control computers, a typical network might consist of a wide variety of remote terminals, each with some processing capability.
To understand vertically distributed processing systems, first some common network terminology should be defined (Kimbleton and Sneider, 1975):

**TOPOLOGY** = refers to the geometric arrangement of links and nodes of a network.

**LINK** = is the communications path between two nodes

**NODE** = is the end point of any branch of a network.

In designing a network, many factors must be evaluated in choosing the most suitable topology. However, one major factor exerts a pronounced influence on this choice: the type of participation by each of the nodes. Any node can be a provider of resources exclusively, a user of resources exclusively, or some combination of resource provider and resource user (Lynch, 1976).
6. BASIC NETWORK TYPES

(a) Point to Point: This is the simplest network (Figure 6) made up of a host processor connected to one communications input/output device per line. The communication input/output device may be a terminal or another processor (Nutt, 1975).

![Diagram of Point to Point Network]

Fig. 6. Point to Point Network

SOURCE: Moore, 1977

(b) Multipoint: In multipoint operation, one station in the network (normally the host processor) is always designed as the control station (Figure 7) (Nutt, 1975).

The remaining stations are designated as tributary stations. The control station controls network traffic by means of polling; that is, it polls the tributary stations (which may be terminals or computers) to send messages. Messages can either go only between the control station and tributary stations or between all stations.
(c) Centralized or STAR: In this type of system all users communicate with a central point that may have supervisory control over the system. Data movement is outward or inward toward the host (Figure 8). If communication becomes necessary between the remote processors or terminals, the host acts as a central message switcher to pass data between them (DEC, 1974).
(d) Tree: When unlike components are connected in a vertical (Hierarchical) distribution of functions, especially control, a system like that of Figure 9 results (DEC, 1974).
(e) Loop or Ring: In this arrangement, the communication bus is in a ring configuration shared by all stations. The advantage of this architecture lies in the high reliability of the bus, as for example, technical difficulties in any one point in the ring will not cause total communications failure. The system is shown in Figure 10 (Acree, 1976).

Fig. 10. Ring or Loop Network

SOURCE: Moore, 1977
(f) Multistar: In this configuration there are several supervisory or exchange points, each with its own set of host and remote processors and a means for direct communication between the points (Figure 12) (DEC, 1974).

![Multistar Network Diagram](image)

Fig. 11. Multistar Network

SOURCE: Moore, 1977
7. DISTRIBUTED PROCESSING

For the great majority of computer professionals, the different processing architectures described in the proceeding sections are only concepts whose importance lies in the understanding of possibilities in the future of data processing.

Vertically distributed processing has caught on very fast and with the proliferation of this type of system, it has grown to the point where literature, applications and developments are ignoring the horizontally distributed processors. For the reason mentioned above, the words distributive processing will be taken to mean vertically distributed processing in the rest of the report.

Distributed computing is characterized by two distinct but (Black, 1976) closely related forms of processing; communications processing and dispersed data processing. Communications processing provides an intelligent pipeline that permits the effective transfer of data and control of the machine/machine interfaces. Dispersed data processing supports the man/machine interfaces that interact directly with the user.

Both are important, but most of the emphasis and support to date has been for the communications function because it must be well (Kimbleton and Sneider, 1975) developed and integrated for a distributed processing network to work even reasonably well. For this reason, network definitions and implementations are undergoing
rapid evolutionary development.

A network structure is chosen simply to support the goals and functions that an organization wishes to implement; it is not an end result in itself. Most network organizations are generally reliable with existing hardware and software. They are like social organizations in that they are organized either hierarchically, with a powerful central computer acting as a feudal overload of the system, or anarchically, with the system composed of an association of independent computers.

It is important to note that all kinds of networks can be constructed using the same heterogeneous components, although some manufacturer's network philosophies are predisposed to a specific structure (the multistar structure may require "some" not standard off-the-shelf hardware and software products) (Hovey, 1976).
8. HARDWARE COMPONENTS

When analyzing the hardware components peculiar to distributive processing as opposed to more traditional approaches, two types of components have to be examined (Cooper, 1977).

The first category consists of the processing elements being linked together to form the network and where defined earlier as Dispersed Data Processing Equipment. When surveying this category, the question to keep in mind is: "Are there particular types of equipment more apt to be linked together in distributed networks as opposed to centralized networks?" (Lynch, 1976)

The second category consists of the communication link itself, previously defined as Communications Processing Equipment. Focusing on the question of whether there is communication hardware peculiar to distributed networks because of the distinctive characteristics of this type of networking. Devices common to all networks are:

(a) Dispersed Data Processing equipment:

- Teletypewriters
- CRTs or video display units
- Hand-held terminals
- Remote batch terminals (high-speed input and output devices)
- Intelligent terminals (CRTs equipped with cassettes, stored program capacity and even disk storage)
Workstations (clustered terminals around a terminal controller)

Industry-application-oriented terminals

Graphics terminals and plotters

Office computers (designed for magnetic ledger operations)

Small business computers (often minicomputer-based)

Minicomputers

General-purpose computers

(b) Communications Processing Equipment:

- Common carriers
- Multiplexers
- Concentrators (often part of clustered terminal packages)
- Private branch exchanges (for line switching)
- Message switchers
- Communications controllers
- Communications processors
- Communications software

Most distributed processing systems will be built using minicomputers and small business computers (Kallis, 1977). The philosophy behind the use of minis - give a user only what he needs to do a particular job - is similar to the philosophy behind distributed processing - make the data processing facilities fit the applications.

Two newer types of business minis, the personal computers and
the word processors, particularly emphasize this trend toward local processing and the distributed networks. Both are designed for nonspecialized, nonprogramming office personnel, and both consequently present interesting management and control problems (Burns, 1977).

Although many factors have combined to make distributed processing a reality, none has contributed more than minicomputers. The minicomputer proved to users that dispersing computer power resulted in quicker turnaround and lower costs. Experience has shown that while super hosts were burying user programs in long batch processing job queues, the minicomputer users were getting responses in a matter of seconds, usually interactively.

Bureaucracy was not only eliminated at the computer site, but also during the purchase of the system. The cost of a typical minicomputer system ($8000 to $150,000) is low enough that the system can be considered a tool rather than a major capital expenditure. Thus, fewer management levels need to be consulted when making a buy decision.

If minicomputers and intelligent terminals of all sorts are particularly characteristic of distributed systems elements, communications processors are particularly characteristic of network management hardware. There are three main types: front ends, concentrators and message switchers.

A front-end processor, by definition, is located with and
attached to one or two specific host computers. Its primary function is to conserve the host computer's resources by assuming some or all of its communications management responsibilities. As a result, data received from a network can be presented to a host in a constant format and from a single defined source. The host can be essentially free of any direct involvement with network requirements.

In a distributed network, the front-end processor is used in much the same capacity, but it can interface its host computer(s) with the network's message-switching element instead of directly with terminals. For this function, the front-end processor intercepts and controls all traffic between its host(s) and the network. In addition, the front-end processor can act as a controller for terminals and peripherals that are direct subordinates of the host(s).

The remote concentrator, or line concentrator, can be linked to a front-end processor located away from its host computer rather than adjacent to it. In fact, many mini-based communication processing devices can be configured either as front-end processors or as remote concentrators.

In a distributed network, the remote concentrator can be used either to offload the host computer and to improve line utilization or as a network node. On the other hand, a general-purpose mini-computer or small business computer used as a network node can be configured to perform remote concentrator functions.

In networks where a large number of terminals exist at the remote site, the remote concentrator may be required to perform polling
functions, in which case it would be involved in two different levels of polling: polling terminals and being polled by the host computer.

Message switchers have been variously called traffic directors, message routers and dispatchers. Unlike that of the remote concentrator, the message switcher's output is not necessarily to a host computer, but to a terminal, remote concentrator or another message switcher.

A message switcher can be based on either a general-purpose minicomputer or a special-purpose processor.

Circuit switching by the host computer in a distributed environment is strictly limited and is usually alone only as an emergency backup. In networks where a number of remote terminals have to communicate with one another, as well as with the host computer, the use of a message switcher is cost-effective.

One major feature of a message switcher which a remote concentrator generally lacks is the ability to detect and act on a priority indicator. Also, the message switcher frequently acts on administrative data and text messages, whereas the remote concentrator typically acts on data intended to be processed in a computer.

Most users do not buy a whole distributed system. Instead, they take an existing network, perhaps add more intelligence to certain elements, change the software in the host and make other changes necessary to disperse processing. The impetus for these changes probably came from a need to expand. Instead of upgrading the host, a remote mini is added and the network architecture is
Selection of distributed processing system components should proceed from performing a study to determine functions and tasks appropriate or desirable for distribution, to a consideration of what classes of devices can perform those tasks and functions, to an evaluation of the various manufacturer's product offerings.

The operational organization of the business is the key to determining if distributed processing is suitable. The business structure also holds the answer to component selection once distributed processing has been decided upon (Doll, 1977).
9. DISTRIBUTED DATA BASE

The term distributed data base can be taken to mean either distributing the data base management function (the control and manipulation of data) or distributing the content of a data base (the data itself). These are two very different techniques with two very different realizations. The data base management function throughout a network is still in the initial implementation stage (Champine, 1977). Today the majority of distributed data bases simply mean redundant data. Local copies of data are maintained at user sites, with most of the same data still retained in a centralized data base.

Most centralized data bases operate under the integrated corporate data base (ICDB) concept. Before considering the distributed data base, it might be well to review the elements of a centralized data base, the ICDB.

An ICDB environment can be defined as the consideration of the collection, storage and dissemination of data as a logical, centrally controlled and standardized utility function (Curtis, 1977).

It should be emphasized that ICDB is not a system; rather it is a concept under which the information system structure should be implemented. This implementation, using the ICDB concept, requires development of the four functional elements (Yasaki, 1977). The four elements of the ICDB are:
1. Data Bank - the logically centralized repository for all the data utilized in a corporation.

2. Data Base Administrator (DBA) - a person responsible for coordination of all data-related activities.

3. Data Base Management System (DBMS) - a software function performing the storage, retrieval and maintenance of data.

4. User/System Interface (USI) - the subsystems necessary to permit multiple classes and types of users to direct the system to structure the available data effectively into information and thus communicate with and fully utilize the resources at their disposal.

Since a centralized data base is a collection of logically related files at one location, then a distributed data base is a logical integration of related data bases at a number of locations. There are various reasons for the necessity of this integration, two of which are paramount: it permits users to produce reports summarizing information from different locations and it provides a means of employing data stored in another data base. Neither of these reasons precludes a distributed data base with no redundant data. And although such a setup is technically feasible, it is operationally undesirable for reasons of backup, security and integrity (Korns, 1976).

There are some design considerations for a sound, shared, distributed data base. First, the design must be compatible with all systems within the network. The data base organization, to be easily implemented and maintained, should be standardized, and the accessing
languages and data access methods should be compatible with all systems.

To accomplish this, some major problems will have to be solved. For example, the DBA must specify the correct data content and logical organization for each user node; a method must be devised to direct all users to all data located around the network; and safeguards must be developed to ensure that all those requesting data are qualified (Rodriguez, 1976).

The integrated corporate data base concept could be applied to support an organizationally distributed data base, as well as a centralized one. Furthermore, DP installations with a multivendor hardware policy may also take full advantage of the ICDB concept by logically centralizing the data bases of different vendors' processors into a single data base.

Storage facilities housing the data base should be designed to support a full range of storage and accessing requirements. This can be facilitated by using hierarchical secondary storage.

Different types of secondary storage media offering alternative access techniques and speeds at correspondingly adjusted costs would allow the DBA to specify and design the optimum physical storage configuration for the data base. By providing multiple levels of hierarchy, an installation can capitalize on the unique requirements of individual users and save both time and money in the process of storing and transferring data.

Total system efficiency depends largely on the specific organization of the data base. There can be only one physical
representation of the data (random or sequential), and the DBA's choice as to which representation to employ is important if not critical.

The applications view of the data (sometimes referred to as the logical representation) is equally important, inasmuch as the application modules of the user systems will be designed to utilize these representations. However, the amount of flexibility allowed in logical representation for applications depends almost entirely on the specific implementation of the DBMS (Tsichritzis and Cockouski, 1976).
A laboratory computer complex can be used to illustrate some of the principles mentioned in the previous sections (Karp, 1976). In this example, the computers are attached to analytical instruments, used to control laboratory experiments and perform data acquisition. One small computer is dedicated to a nuclear magnetic residence spectrometer, another to an infrared spectometer, and a third to a mass spectometer. An application program assigned to the computer system in each instrument manages the experimental apparatus.

The differences between a set-up of stand-alone computers that partition a data processing workload and a computer network that integrates these same tasks will now be shown.

In a stand-alone system each instrument's readings are independent of each other, and either a manual set-up or another independent computer system would be necessary to merge the analytical data to determine, let us say, the molecular structure of a complex organic compound.

Tie the computers together into a network, however, and they can then do the analysis online. Moreover, the central computer by maintaining data files on all experimental results can prevent a researcher from inadvertently repeating an experiment. But even more important, the research team would have instant access to all the cumulative data and analyses on a sample because this information
could all reside in a comprehensive network file.

In this laboratory example, the system designer also has an option to use one of two methods to transmit the data around the system. The dedicated computers could store all the analytical data generated in an experiment and then use high-speed synchronous communications to transmit the information to the central computer. Alternatively, the data could be transmitted piecemeal as determined by the program using a synchronous data communications techniques.

Either data transmission method requires that the computers be compatible, and this constraint is typically achieved by means of software designed to follow a so-called communications protocol. In other words, all processors linked in a network must "understand" all transmitted and received messages. Such compatibility should be beneficial to users. Networks that are constructed from a diversity of computer brands tend to require more than one protocol, and this can cause transmissions delay. Before re-transmitting a communication, a processor will have to reformat the received message so that it can match the protocol required by the non-compatible computer.

Networks can perform other functions besides distributing a data processing workload. Many networks are used to provide redundancy and back-up as a security measure. And even when setting up stand-alone computers, designers should consider creating a system that could be integrated into a network to meet unforeseen or expansion needs. The designer can build a network in increments according to need and cash availability.
As was explained a network distributes computer functions among its elements according to the organizational structure, locations, and tries to get the most cost-effective arrangement in the specific application (Benson, 1976).

An example of the above statement can be seen in the mock-up of a distributed processing network within a single company depicted in Figure 12. This company is made up of Research & Development, Manufacturing and Sales (Karp, 1976).

Scientists in the research and development division work on laboratory experiments and gather data for analysis. Each lab uses a dedicated minicomputer having disk storage and a printer; but the cost of individual computers are high. So, a resource sharing computer network is employed. The large, expensive peripherals reside at a central host computer. Each lab has a relatively small satellite system that includes a console terminal. Scientists use the terminal to edit, compile and transmit data and programs for storage and execution.

The manufacturing division, aiming to automate plant operations, wants to control raw material input, the manufacturing machinery, and also operation of an automated warehouse for finished goods storage. The management also wants access to current inventories and stock levels. A hierarchical distributed computer network can distribute the tasks to individual computers, each specifically designed to handle a function and each in communications with all other systems. For example, realtime process control systems control
the actual manufacturing operations, and are in turn linked to supervisory systems that control overall parts flow. The supervisory system computers pass data to a transaction-processing or other large system for management control.

The sales offices that are widely dispersed need data to access current inventory information and shipping dates, not individual telephone lines to a central computer are expensive. A communications network that significantly reduces line costs is installed. It contains terminal concentrators at local regional centers to switch data traffic over high speed lines to the central system. These data can be terminal input or data-pre-input on disk or magnetic tape. Processed information can be shipped back to the offices, to terminals or disk storage for later printing offline.

In the example, the mechanism for data communications between programs and devices on different systems and on computers running under different operating systems would be the relatively high levels of software sophistication and expense.
The computers here should be HORIZONTALLY DISTRIBUTED for faster response times.

**Fig. 12. Distributed Processing**

**SOURCE:** Karp, 1976

Environment: with Horizontal & Vertical Computers
The issue of centralization of information systems has been unleashed by the increasing number of Distributed Computing Systems. Furthermore, the centralization-decentralization problem is complex and important enough, so that any paper dealing with Distributed Processing should cover at least the most important aspect of the problem: the centralization-decentralization decision.

Articles on the advantages and disadvantages of centralization and/or decentralization abound in the literature (Fleming, 1976; Reynolds, 1977 et al.). Since different authors have different assumptions and approach the problem somewhat differently, the arguments are not strictly comparable. For this reason, a table summarizing most of the pro and con arguments advance in the literature is condensed in Tables 1-8.

The problems of pure EDP centralization and decentralization have produced the "common alternatives" where distributive processing fits best (Doll, 1977; Atrick, 1976). Some of them are:

1. **Operations centralized and system development left to division** - this is an alternative most often adopted in large organizations producing highly technical products, such as aerospace manufacturers, with large amounts of scientific and engineering processing.
2. **System development centralized and operations dispersed** - this is an alternative usually found in large business organizations with geographically dispersed divisions performing identical functions, none of them of such a nature that very large computers are required.

3. **Central control of equipment acquisitions and central development of applications common to an entire functional area** - this compromise is generally found in large, geographically dispersed companies whose divisions and subsidiaries have products representing a compromise between diversity and commonality.

4. **One larger centralized computer plus smaller satellite computers and remote job entry terminals, and centralized development augmented by small development groups for unique local needs** - a compromise somewhat simpler than the one above, more appropriate in smaller and less diversified companies.

5. **Centralization of policies for equipment acquisition and personnel training, some centralized standards, and common systems for management reporting** - this alternative is the most appropriate to multi-national corporations, where multi-lingual and multi-cultural factors exist, and different equipment is superior in different countries.
### TABLE 1
GENERAL AND ORGANIZATIONAL CONSIDERATIONS

<table>
<thead>
<tr>
<th>Centralized Systems</th>
<th>Decentralized Systems</th>
<th>Distributed Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>easier consolidation of company-wide operating results</em></td>
<td><em>profit &amp; loss responsibility</em></td>
<td><em>feeling of exclusive use by user organization</em></td>
</tr>
<tr>
<td><em>ease of control &amp; coordination by corporation management</em></td>
<td><em>familiarity with local problems</em></td>
<td><em>when using standard equipment; the development of applications &amp; transfer of personnel between divisions easier.</em></td>
</tr>
<tr>
<td><em>enhances corporate consolidation</em></td>
<td><em>rapid response to local needs (also less formal)</em></td>
<td><em>permits the moving of &quot;extra&quot; computer equipment to divisions with an extra load.</em></td>
</tr>
<tr>
<td><em>can lead to integration of other administrative functions</em></td>
<td><em>special programs and services can be tailored to division needs</em></td>
<td><em>reduces the number of separate equipment while allowing decentralized DP advantages</em></td>
</tr>
<tr>
<td><em>easier to implement and maintain standards</em></td>
<td><em>easier communication between DP and user (more involvement)</em></td>
<td></td>
</tr>
<tr>
<td><em>shared development costs</em></td>
<td>&quot;hands-on&quot; experience for users possible</td>
<td></td>
</tr>
<tr>
<td><em>small user access to large CPU</em></td>
<td><em>more flexibility in coping with crises and changes in plan</em></td>
<td></td>
</tr>
<tr>
<td><em>a greater variety of services and programs can be offered</em></td>
<td><em>better service—under user control</em></td>
<td></td>
</tr>
<tr>
<td><em>user relieved of mgt. &amp; operation of computer facility</em></td>
<td><em>flexibility in aligning EDP with organiz. philosophy</em></td>
<td></td>
</tr>
<tr>
<td><em>easier to direct overall use of computing</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Centralized Systems</td>
<td>Decentralized Systems</td>
<td>Distributed Systems</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>*management problems associated with large staffs</td>
<td>*strict controls and standards required to prevent duplication of software development</td>
<td>*additional controls &amp; standards for:</td>
</tr>
<tr>
<td>*prone to cause barriers to acceptance</td>
<td>*no professional EDP management</td>
<td>- ensuring communication between sites</td>
</tr>
<tr>
<td>*more likely to cause political problems</td>
<td>*separate equipment acquisition studies &amp; interchangeability</td>
<td>- safeguard access to distributed data base</td>
</tr>
<tr>
<td>*higher risk of failure</td>
<td>*strict controls and standards required to prevent duplication of software development</td>
<td>- people in DP required to have two managers.</td>
</tr>
<tr>
<td>*more rigid: any change may have serious ramifications</td>
<td>*strict controls and standards required to prevent duplication of software development</td>
<td>*problems of network management:</td>
</tr>
<tr>
<td>*requires top mgmt. involvement</td>
<td>*strict controls and standards required to prevent duplication of software development</td>
<td>- income allocation</td>
</tr>
<tr>
<td>*more vulnerable to corporate overhead reduction</td>
<td>*additional controls &amp; standards for:</td>
<td>- expense allocation</td>
</tr>
<tr>
<td>*management problems associated with centralized organizations:</td>
<td>- ensuring communication between sites</td>
<td>- assigning performance responsibility</td>
</tr>
<tr>
<td>- standardization</td>
<td>- safeguard access to distributed data base</td>
<td>- agreement on priorities</td>
</tr>
<tr>
<td>- assigning performance responsibility</td>
<td>- people in DP required to have two managers.</td>
<td></td>
</tr>
<tr>
<td>- agreement on priorities</td>
<td>- income allocation</td>
<td></td>
</tr>
<tr>
<td>- scheduling problems</td>
<td>- expense allocation</td>
<td></td>
</tr>
<tr>
<td>- expense allocation &amp; pricing</td>
<td>- assigning performance responsibility</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- agreement on priorities</td>
<td></td>
</tr>
<tr>
<td>Centralized Systems</td>
<td>Decentralized Systems</td>
<td>Distributed Systems</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>*economies of scale in main frames</td>
<td>*lower communication costs</td>
<td>*depends on the amount of centralization or decentralization of the system. Also depends on the actual network and data base structures</td>
</tr>
<tr>
<td>*economies of scale in mass storage devices</td>
<td>*modest start-up costs</td>
<td></td>
</tr>
<tr>
<td>*reduced record storage duplication</td>
<td>*low incremental expansion costs</td>
<td></td>
</tr>
<tr>
<td>*reduced site preparation and protection costs</td>
<td>*higher share of raw computing power available to user</td>
<td></td>
</tr>
<tr>
<td>*fewer operators required</td>
<td>*avoids certain user-computer communication costs related more to administration than to operations</td>
<td></td>
</tr>
<tr>
<td>*fuller utilization of processing capability.</td>
<td>*better cost/performance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>*faster reaction to new technological advances</td>
<td></td>
</tr>
</tbody>
</table>
### TABLE 4

COST FACTORS: DISADVANTAGES

<table>
<thead>
<tr>
<th>Centralized Systems</th>
<th>Decentralized Systems</th>
<th>Distributed Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>*may require costly controls</td>
<td>*some idle resources</td>
<td>*high costs for extensive conversion</td>
</tr>
<tr>
<td>*danger of expensive overhead</td>
<td>*possible duplication of software costs</td>
<td>*high communication costs</td>
</tr>
</tbody>
</table>
TABLE 5
PERSONNEL CONSIDERATIONS: ADVANTAGES

<table>
<thead>
<tr>
<th>Centralized Systems</th>
<th>Decentralized Systems</th>
<th>Distributed Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>*general shortage of competent D.P. personnel</td>
<td>*greater interest and motivation at local level</td>
<td>*depends on the amount of centralization or decentralization of the system. Also depends on the actual network and data base structures.</td>
</tr>
<tr>
<td>*better availability in metropolitan centers</td>
<td>*identification with the mission of the sub-organization</td>
<td></td>
</tr>
<tr>
<td>*more efficient use of personnel talents (specialization)</td>
<td>*less risk of personnel turnover</td>
<td></td>
</tr>
<tr>
<td>*larger and more expert pool of consultants</td>
<td>*more opportunities to communicate with (and transfer into) line management</td>
<td></td>
</tr>
<tr>
<td>*broader career opportunities - more attractive position</td>
<td>*less skilled personnel required</td>
<td></td>
</tr>
<tr>
<td>*higher standards due to more competitive salary levels</td>
<td></td>
<td></td>
</tr>
<tr>
<td>*personnel turnover less critical</td>
<td></td>
<td></td>
</tr>
<tr>
<td>*fertilization</td>
<td></td>
<td></td>
</tr>
<tr>
<td>*rotation of personnel more natural</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## TABLE 6

### TECHNICAL CONSIDERATIONS—PROGRAMMING

### ADVANTAGES

<table>
<thead>
<tr>
<th>Centralized Systems</th>
<th>Decentralized Systems</th>
<th>Distributed Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>more sophisticated software</em></td>
<td><em>smaller programs—need to handle only on local situation</em></td>
<td><em>depends on the amount of centralization or decentralization of the system. Also depends on the actual network and data base structure.</em></td>
</tr>
<tr>
<td><em>better service to programmers and users</em></td>
<td><em>easy to satisfy &quot;hand-on&quot; requirement for testing purposes</em></td>
<td></td>
</tr>
<tr>
<td>- system software can provide help</td>
<td><em>easier to add new applications and services</em></td>
<td></td>
</tr>
<tr>
<td>- greater selection of programming languages, debug aids, etc.</td>
<td><em>forces modular programming; easier to debug and maintain</em></td>
<td></td>
</tr>
<tr>
<td><em>can handle large programs, no need to break up problem</em></td>
<td><em>progressive approach to installing systems (projects break up naturally)</em></td>
<td></td>
</tr>
<tr>
<td><em>easier to implement changes in data base technology</em></td>
<td><em>less specialized support</em></td>
<td></td>
</tr>
<tr>
<td><em>economies of integrated requirements</em></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
TABLE 7
TECHNICAL CONSIDERATIONS: PROGRAMMING
DISADVANTAGES

<table>
<thead>
<tr>
<th>Centralized Systems</th>
<th>Decentralized Systems</th>
<th>Distributed Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>*multiprogramming limits programmers</td>
<td>*forces modular programming which is difficult to implement</td>
<td>*depends on the amount of centralization or decentralization of the system. Also depends on the actual network and data case structures</td>
</tr>
<tr>
<td>*virtual storage conflicts with modular programming</td>
<td>*have the problems with current minis and micros: -little addressable space -non-compatibility (even within brands) -no choices between software vendors</td>
<td></td>
</tr>
<tr>
<td>*mutual interdependence between jobs complicate both development and operations</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### TABLE 8

**TECHNICAL CONSIDERATIONS: OPERATIONS**

<table>
<thead>
<tr>
<th>Centralized Systems</th>
<th>Decentralized Systems</th>
<th>Distributed Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ADVANTAGES</strong></td>
<td><strong>ADVANTAGES</strong></td>
<td><strong>ADVANTAGES</strong></td>
</tr>
</tbody>
</table>
| *reduced mean variance on turn around time, which means better service.  
*a greater variety of services and programs can be offered  
*less disruption when user moves (if both sites use same facilities) | *more fault tolerant design  
*easier to add new service  
*less specialized support necessary  
*newer hardware technology on average | *higher reliability  
*better data communications performance (fewer information errors)  
*flexibility as to location of site  
*chance for better security |
| **DISADVANTAGES**   | **DISADVANTAGES**      | **DISADVANTAGES**   |
| *system software is complex and resource consuming | *user may want to step-up to more elegant system  
*more frequent breakdowns | *system software is complex; and resource consuming |
12. CONCLUSION

To some, distributed systems represent solutions to complex problems; to others, they create problems too complex for solution. However, distributed systems (horizontal and vertical) are coming and both computer professionals and functional management must be prepared.

Navy Captain Grace Hopper is often quoted (Withinton, 1973) about a logging operation that uses oxen. When a log is too big for one ox, they don't send for an elephant, they use two oxen. So too with computers according to some distributed systems advocates; use small or medium scale computers and add another computer as the work load increases.

If compatible systems (oxen) are used, the interface works. Suppose, however, to interface an ox and a horse? An incompatibility would exist. A special interface could be designed and built, but at significant expense. This illustrates the hardware interface problem.

Now suppose two oxen were interfaced correctly, but they are not trained (programmed) to work as a team. A fight could develop or they could refuse to pull at all. This illustrates software incompatibility problems.

If both problems are addressed thoroughly, however, a successful
distributed system can be attained. In a successful distributed system of the 1980's one can expect to find an optimal mixture of horizontal and vertical distributed computers in an on-line total information system.

It is important to mention the fact that several mainframe and minicomputer manufacturers like IBM, Digital Equipment, Texas Instruments and Hewlett-Packard in particular, have started advocating and supporting the concepts of distributed processing (Wang, 1976; HP, 1976) outlined in this report. This enhances the feasibility of successfully implementing such a configuration, and makes the understanding of the choice process even more critical. This choice of a system network is dependent on the characteristics of the applications and how close a network can simulate the organizational structure, speed and information flow desired. Finally, the total cost of a distributed system is still higher than that of a centralized system because of software development and communications costs. But as hardware costs decreased this last decade, so software costs are expected to decrease in the next decade (Eker, 1976; Bremmer, 1976).
REFERENCES


Curtice, Robert M. "Integrity in Data Base Systems." Datamation 23 (May 1977): 64-68.


Patrick, Robert L. "Decentralizing Hardware and Dispersing Responsibility." Datamation 22 (May 1976): 79-90


