COMPARATIVE MERITS OF A CENTRAL DATA BASE AND REPLICATED DATA BASES FOR AN INTEGRATED INVENTORY CONTROL SYSTEM

CLEARING THE FOREST

The guiding principle governing distributed information systems is optimum functional distribution. With functional distribution, only selected processing functions take place within a single element of a total system, while other functions are distributed to other system elements. The goal is to locate specific elements physically close to the users of the functions being distributed. In significant contrast, the concept of centralized functionality dictates that all computing and data storage resources are physically located in one processing element. Centralized functionality presupposes that users must come to the computer or, at best, will be remotely attached to the almighty source.

The last decade has rendered a variety of related buzz words with overlapping and confused meanings. These include terms like "distributed processing," "distributed systems," "distributed networks," and "distributed data bases." The sad part of all of this is that these terms now have no universally understood definitions. They are used by authors and industry professionals differently. There has, however, been a fortunate recent trend to standardize some of these terms,[1] and in this paper I will attempt to use terminology which reflects this move toward standardization.

A distributed information system can be defined as "a coordinated set of information processing capabilities implemented in two or more relatively independent resource centers such as computer sites, terminal locations, and so on."[2] In my view, there are three substructures which combine to form the total structure of a truly distributed information system. These are:

1. A distributed processing substructure that reflects "a technique for implementing one logically related set of information processing (application-related) functions within multiple physical devices."[3]

2. A distributed data base substructure which implies "a single logical data base which has been implemented in more than one physical segment, attached to more than one information processor."[4]

3. A networking substructure consisting of "the hardware and software functions which support the definition, establishment, and use of facilities for data movement among (usually physically separated) information system components."[5]

Theoretical consideration of one or any combination of these substructures would result in a work of great magnitude. A practical consideration (i.e., related to a user application within a specific hardware/software environment) could be similarly monumental. Therefore, my intention is to present a practical aid to help designers of automated inventory control systems make key decisions related to data base distribution. The impetus of this paper will focus on substructure 2 above, applicable whenever a solution involves a distributed data base. All cases will deal only with solutions which have been (or could be) implemented on HP-3000s using IMAGE and DSN/DS3000. All of the processors configured in this paper are of the same type (HP-3000 series), with all of the CPUs comprising any one configuration running under the same MPE operating system version, etc. In these examples, then, we will encounter no compatibility problems related to distribution of the work load. Many of the cases considered here are now operating using the solutions described. No companies will be referred to by name, and some configurations have been altered slightly (in order to heighten the effect of the solution).
Specifically, then, I will present cases involving wholesale distributors operating from multiple locations using HP-3000s to specifically control all of the automated Inventory applications, which include Order Entry, Stock Allocation, Order Filling/Shipping, and Warehouse Transfers, as well as other general considerations specifically related to supply and demand in the wholesale distribution business. The solutions I will examine are:

I. Distributed Data Bases

 1. Partitioned Data Bases

  a. Geographically Partitioned

  b. Hierarchically Partitioned

 2. Replicated Data Bases

  a. Hierarchically Replicated

  b. Fully Replicated

  c. Horizontally Replicated

 3. Combined Partitioned and Replicated Data Bases

II. Centralized (Nonreplicated) Data Bases

I. Distributed Data Bases

The two fundamental structures for data base distribution are partitioned and replicated. Complex applications may warrant the combination of both partitioned and replicated structures.

1. Partitioned Data Bases

All data base design involves the development of a conceptual data base, which should parallel a logical schema of some sort. The conceptual data base will include all data of interest to an organization. Whenever a conceptual data base is separated into nonredundant sections (i.e., partitions) and spread across multiple information processors, a partitioned data base is formed. Once all of these partitions are attached to a single information processor, together they form a single logical data base which defines the sum of the partitions. The generic partitioning of a conceptual data base often results in the configuration depicted in Figure 1.

Despite the physical representation of this configuration, there is, at this point, no determination of where the partitioned data bases and replicated hardware should reside geographically. For example, data volumes and performance considerations alone could require such a partitioning, although all three partitions occupy the same location, with all hardware in the same computer room. The manner in which a data base is partitioned very closely correlates to the information processing structure of the distributed system. In one case, partitioning may be used to insure privacy and improve security by partitioning public information from sensitive data. In another, data may be partitioned in order to physically locate it close to where access requests for that data originate.

1.a. Partitioning Data Bases Geographically

One organization in the wholesale distribution industry operates two HP-3000/64s in a horizontally distributed system (i.e., where both CPUs cooperate as equals). Their particular configuration exemplifies the classic geographically partitioned data base. The company operates on a nation-wide basis within the continental United States. Demand for the items distributed is affected both seasonally and geographically. The conceptual data base for this company locates data elements logically, so that data base elements accessed heavily by East Coast users are partitioned with the Eastern Host, while the Western Host partitioning reflects the same mentality. This configuration is depicted in Figure 2.

This specific geographical partitioning is the most practical solution for the company because of its natural geographic grouping of data base access requirements. If elements accessed from intermediate locations had been equitably distributed between the two centers, this partitioning would have caused entirely too much traffic between centers to be practical. Perhaps the best rationale for this solution is the high cost of cross-country data transmission. However, even in those instances where cost of transmission is not dependent upon distance (e.g., packet network services), transmission volume remains a crucial factor. The system illustrated in Figure 2 reflects an application where the majority of accesses to each geographic data base partition will originate locally. However, provision was still made for access of

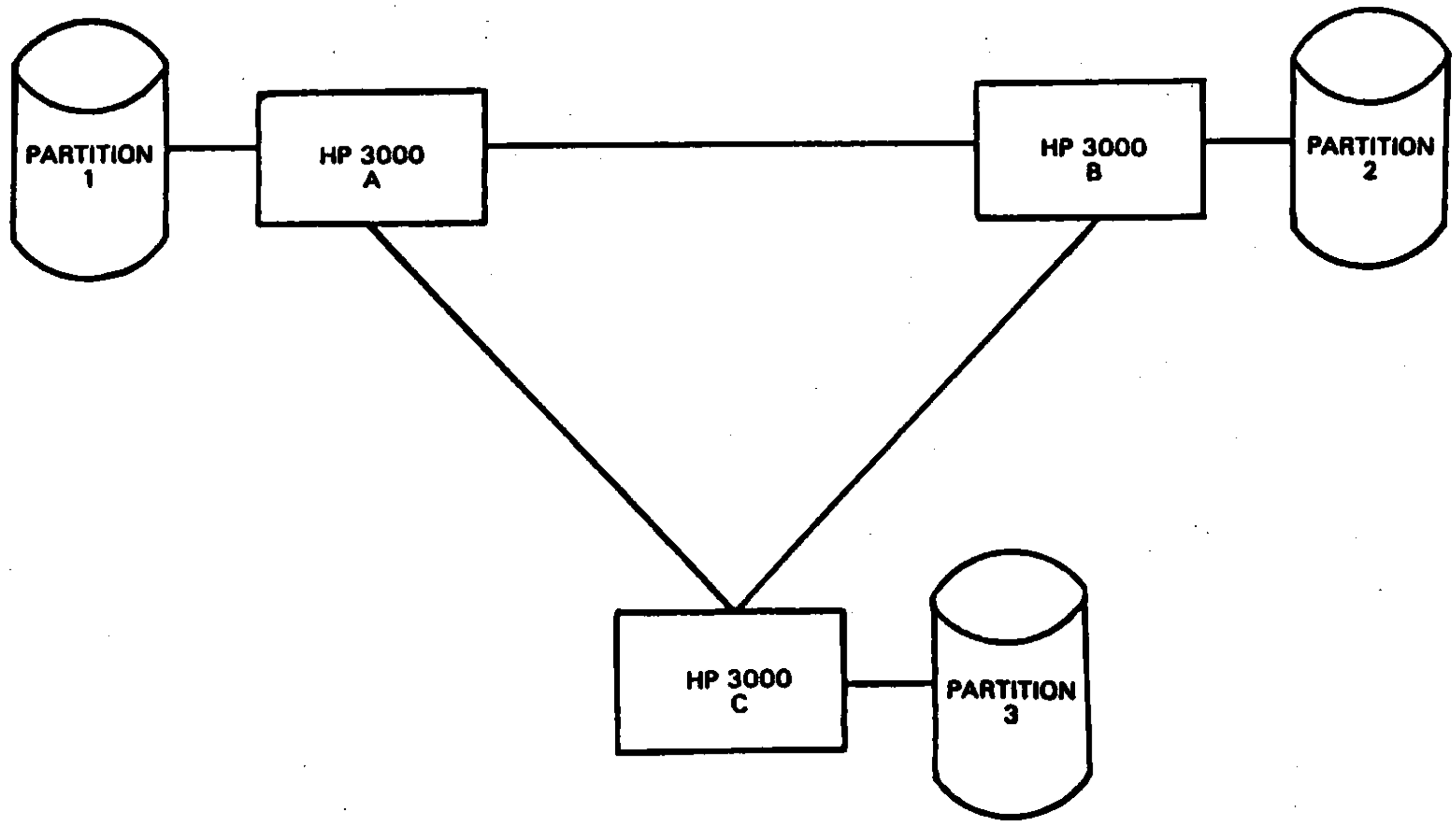

Figure 1: Generically Partitioned Conceptual Database

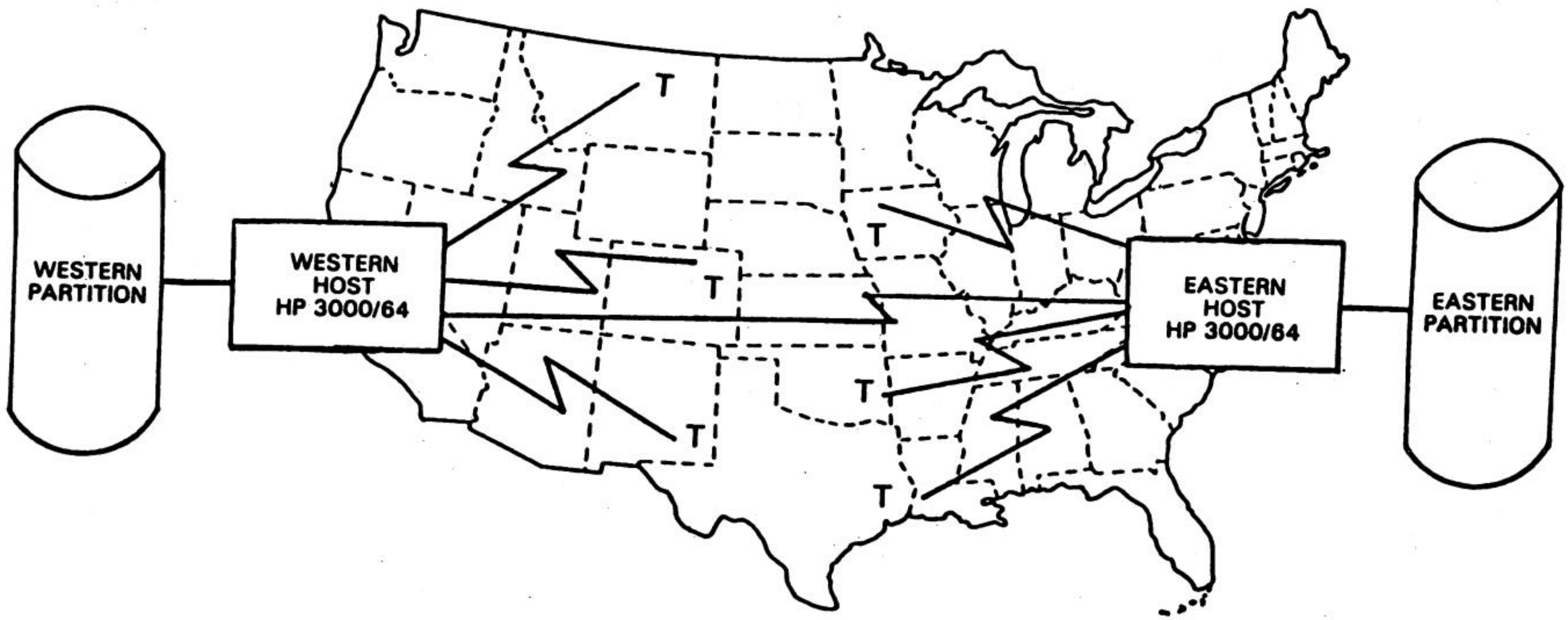

Figure 2: Geographically Partitioned Databases

the Eastern Host by the Western Division, and vice versa. For despite the excellence of the design of the partitioning, there are still a few inventory items which both locations stock commonly. Thus, there is some limited transferring of goods between divisions which had to be accommodated. Still, this design remains best for the way this company operates.

1.b. Hierarchically Partitioned Data Bases

Another partitioning methodology advocates the hierarchical distribution of data elements. Most examples of hierarchical distribution bring to mind applications in which the processing hierarchy corresponds to the organizational hierarchy (e.g., the highest-level host handles corporate-level processing and each subordinate division consolidation handles its own needs using other hosts or satellite CPUs). Organizations attempting to automate inventory control functions are discovering more and more instances ideally suited to the solutions afforded by hierarchically partitioned data bases. These instances are relatively new, coming with the advent of a breakthrough in modern automated inventory control systems called Distribution Resource Planning (DRP).[6] Traditional inventory control systems treat each distribution center in a network separately, so that each center must fend for itself and determine its own inventory requirements based upon its own isolated history of supply and demand. In the case of Regional Distribution Centers (RDCs) which supply satellite Distribution Centers (DCs), each DC traditionally places its own orders with an RDC. Items are often transferred between DCs and RDCs if inventory requirements necessitate such activity, which is expensive. With DRP, however, the requirements of all of the DCs supplied by any one RDC are considered cumulatively by that RDC, so that goods may be centrally dispersed in a more controlled, time-phased manner. Costly warehouse transfers are eliminated. In addition, DRP has proven that RDCs can get the appropriate goods to the appropriate DC within the required time frame. Consider the distribution hierarchy of Figure 3.

In this example, the individual inventory requirements for San Francisco and San Diego are summarized and added to the requirements for Los Angeles. Similarly, the individual requirements for Houston and New Orleans are summarized and considered to be part of the demand requirements for Dallas. At another level within this hierarchy, the individual (summarized) demand requirements for Los Angeles and Dallas are added together and planned for by the Chicago central supply facility, which must also plan for its own demand requirements. This arrangement is known as a "push" system, in which a common supply facility determines replenishment requirements for the distribution centers it supplies. This overall relationship lends itself very well to an automated solution involving hierarchically partitioned data bases, as seen in Figure 4.

In this configuration, the Chicago Host handles system-wide data, and Los Angeles handles data for itself as well as for San Francisco and San Diego. Dallas handles data for itself as well as for Houston and New Orleans. At the bottom (i.e., DC) level of the hierarchy, there will be no duplication of data. Each DC will have access to its appropriate RDC, yet no DC will have

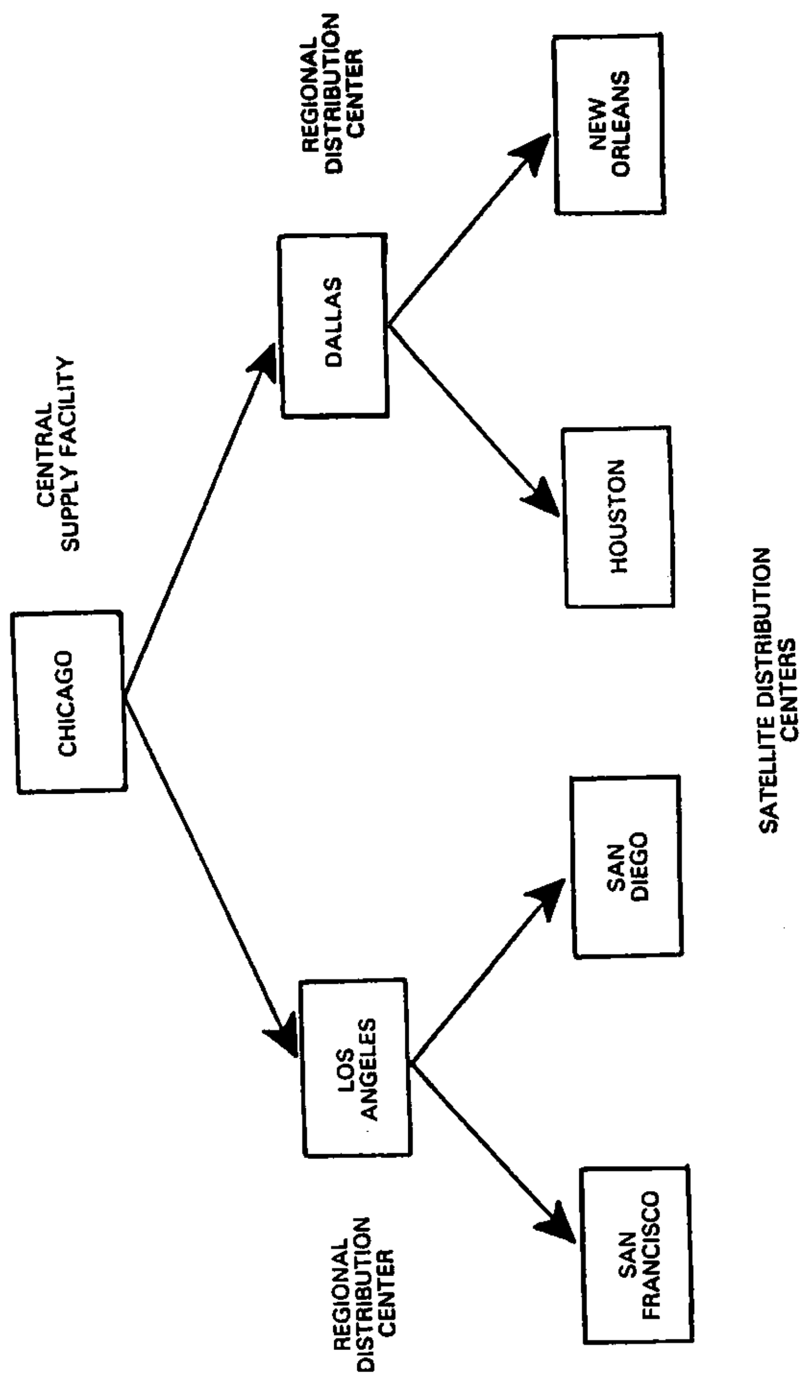

Figure 3: Hierarchically Organized Wholesale Distributor

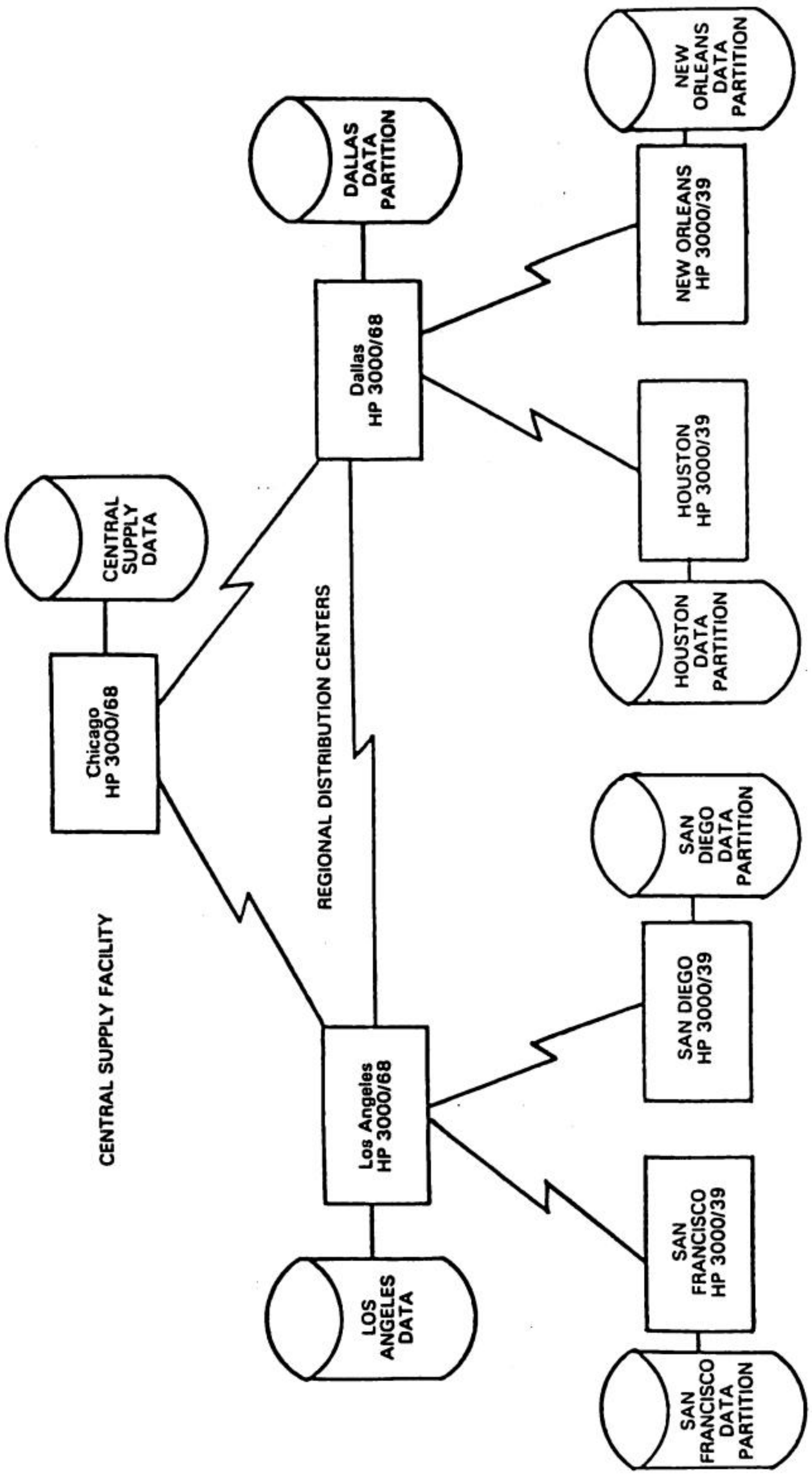

Figure 4: Hierarchically Partitioned Data Bases

data on site other than that pertaining to items distributed by it. Only at higher levels is there significant duplication of data. In effect, the data elements at all locations parallel the organizational hierarchy which has been successfully partitioned across the CPUs within the hierarchy. Because of this design, information can be exchanged both up and down the hierarchy. The central supply facility in Chicago must consolidate inventory demand requirements for all of the lower levels, while data must be distributed from each level to the next in a downward direction as delivery schedules are maintained. Because of this interrelationship of data traveling in both directions within the hierarchy, the overall system exhibits a structure which is truly partitioned and distributed hierarchically, rather than merely equaling the sum of separate independent data bases.

2. Replicated Data Bases

Another way to address distributed data bases is to place copies of all or parts of a whole company's data at multiple locations, resulting in a "replicated data base." Like partitioning, replication puts the data in the location where the access requests originate. The difference is that replication uses duplicate (i.e., copied) information, while partitioning enables the user to access original data elements. In instances where frequent and comprehensive data availability is required, replicated data bases provide a generally good solution. Essentially, there are two ways in which a data base can be replicated: hierarchically and horizontally.

2.a. Hierarchical Data Base Replication

A wholesale distributor operates with one central supply facility, two RDCs, four satellite DCs, and six nonstocking sales offices nation-wide. The central supply facility and both RDCs stock all of the items distributed by the company. Each satellite DC stocks approximately 50% of the items carried by the RDC which supplies it. In addition, certain sales offices may order only from specific DCs. The overall relationship is depicted by Figure 5.

Notice that New York supplies both Los Angeles and Pittsburgh, and all three carry identical items. Half of the items stocked by Pittsburgh are used to supply Cleveland, the rest are for Louisville. Half of the items Denver stocks are intended specifically for replenishment of Phoenix, with the remainder intended to supply Los Angeles. The Detroit sales office may order only from Cleveland, while the Tulsa sales office may order from either Cleveland or Louisville. The San Francisco sales office may order only from Los Angeles, but San Diego may order from Phoenix or Los Angeles, while the Las Vegas sales office must order only from Phoenix. This arrangement is known as a "pull" system, in which each distribution center determines its replenishment requirements and orders from a common supply facility. This odd, but real situation lends itself ideally to a solution involving hierarchically replicated data bases. This solution in depicted in Figure 6.

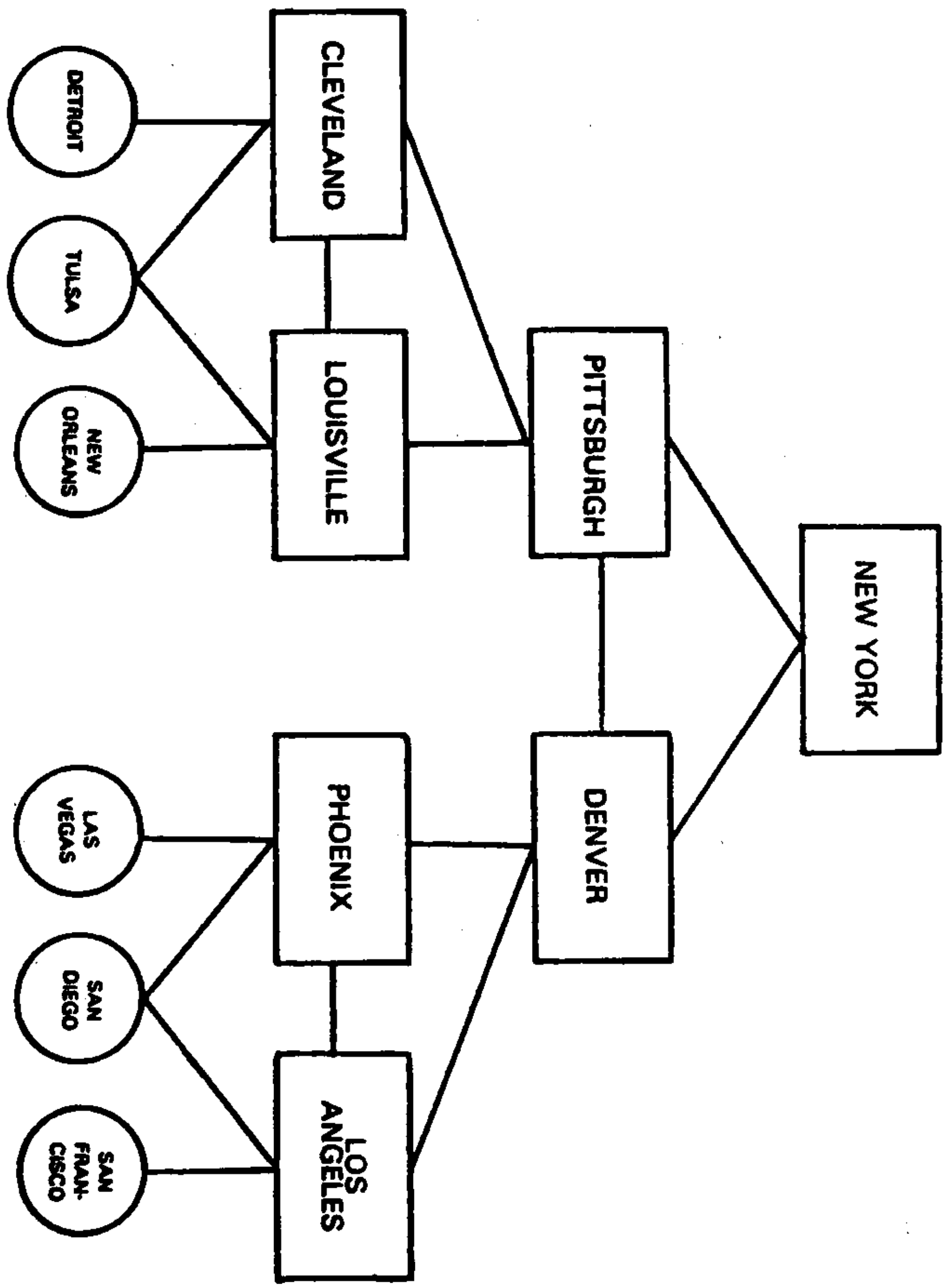

Figure 5: Hierarchically Organized Wholesale Distributor

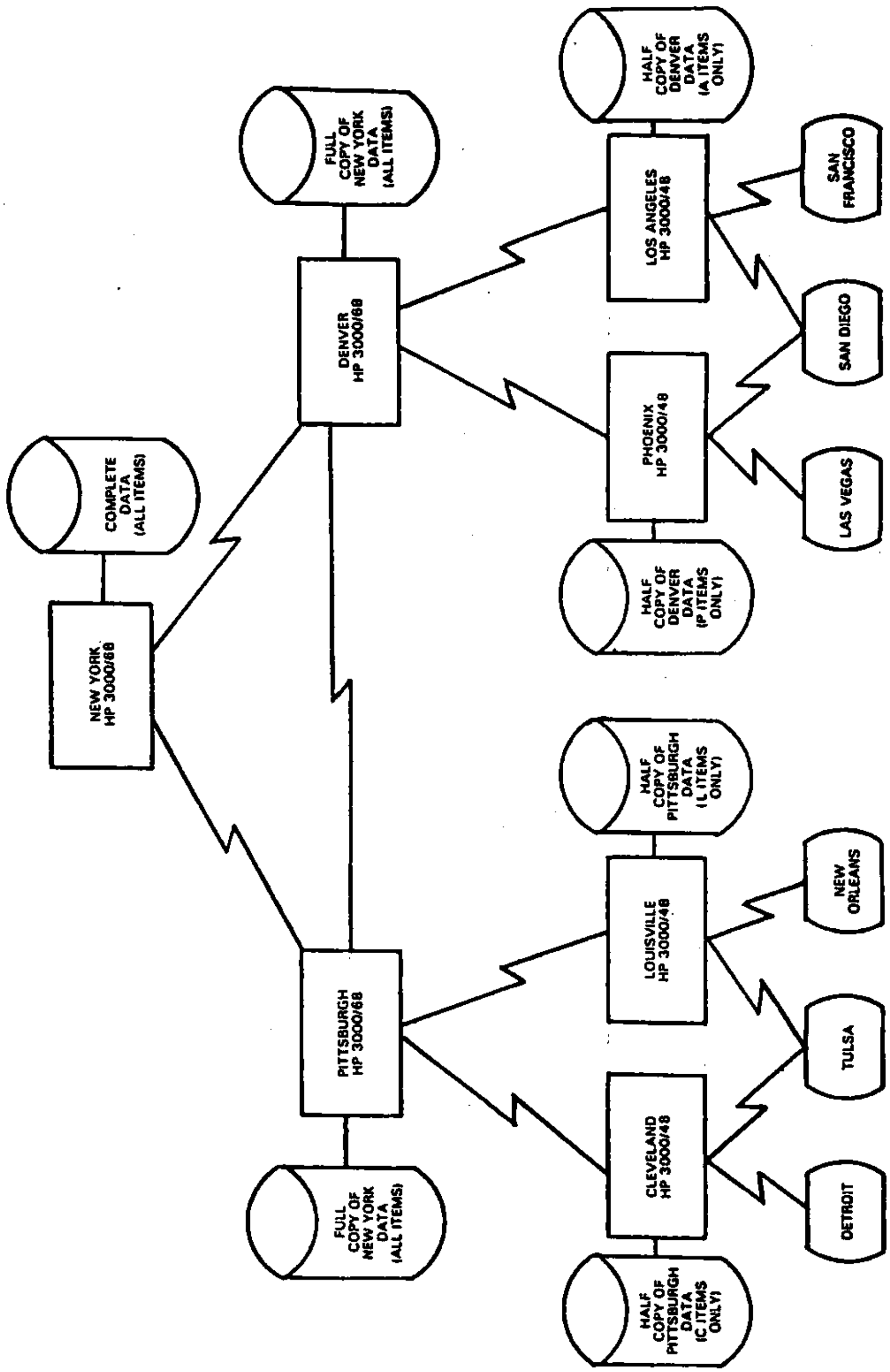

Figure 6: Hierarchically Replicated Database (Full and Partial Replication)

In this configuration, the New York host maintains a master data base containing the inventory control data for the entire organization. Pittsburgh and Denver also have complete physical copies of the New York data base. Cleveland has a partial replication of the master data base, consisting of the data related only to the items stocked by Cleveland. Louisville has a different part of the same master data base replicated, with only the items it ships. Phoenix and Los Angeles similarly have differing partial copies of the same master data base (i.e., the one shared by New York, Pittsburgh, and Denver). All of the replications at the lowest level are mutually exclusive, with no redundancy between them. Each local data copy, then, only replicates that data needed for its own operation. In contrast, the data bases for New York, Pittsburgh, and Denver are fully replicated. This turned out to be a big plus for this organization, because full replication served as back-up for the whole organization when there were problems.

2.b. Horizontally Replicated Data Bases

Another wholesale distributor of durable goods uses three HP-3000/68s as horizontally distributed hosts. Each distributed host maintains stock balances only for items available from its own physical location. In this sense, the data base is partitioned. However, in addition to this data, each host maintains product data consisting of product code, product description, and the location(s) stocking each product. Thus, the data for every host reflects a partial replication of the data of every other host. This configuration is depicted in Figure 7.

This arrangement allows orders for items not locally stocked to be validated and priced (for credit checking) before orders for them are sent to the appropriate stocking location. Items that do not exist anywhere within the inventory control system can be identified and rejected immediately. This instance of partial data base replication typifies a global directory or catalog of stocked items. A data base such as this, that is both horizontally replicated and partially partitioned, is generally rare. Most often data base replication is used exclusively in systems which are distributed hierarchically. Most horizontal system structures employ strict partitioning, with no overlapping data elements, or use separate, entirely independent data bases at each host location. Combination partitioned and replicated data bases are more common in other types of applications (e.g., airline reservation systems) than they are in inventory control applications.

II. The Centralized Data Base Approach

Generally, a centralized approach locates all computing and data storage in one processing element. Remember, though, that in one configuration, a centralized approach can involve multiple CPUs and even separated data bases in one location (e.g., the generic separation effected in Figure 1 could reflect three CPUs under one roof functioning together as one processing element). Also, the fact that all users of a system are served by the same data center supports the claim that linked CPUs and distributed data bases in one location add up to a centralized solution (although most industry professionals would call this a distributed solution).

Advantages

An approach involving one centralized data base is appropriate as long as its performance adequately satisfies the requirements of an inventory control system. In fact, any of the cases studied here could have used a centralized solution. Notably, there is no other alternative in instances where the conceptual (i.e., logical) data base cannot be partitioned. Most smaller distribution companies currently employ the centralized solution, even when shipping from multiple locations.

There are some advantages to a centralized approach. One is that you may avoid all of the disadvantages associated with the many alternative distributed approaches. Even in the centralized approach involving multiple CPUs and separate data bases in one location, you will eliminate many potential problems encountered when distributing data bases and processing resources across different locations.

Users who attempt to distribute data bases and processing resources across different locations may face a variety of problems. Many system designers tend to underestimate the difficulty of controlling access to multiple data bases. This problem is encountered whether or not linked CPUs occupy the same computer room. Although centralized data bases also require coordination and access control procedures, these can be handled with far less difficulty simply because everything happens on one computer. Security is similarly problematic, i.e., far less of a problem for one data base on one computer. System performance, back-up and recovery procedures, and data integrity maintenance across multiple copies which are partitioned and replicated in any number of ways are also requirements which can kill any efforts toward distribution. Associated costs most often end dreams of functional distribution. Initial costs required to replicate hardware, software, and personnel are only part of the picture. Facilities must be properly prepared (e.g., upgrades in electrical and air conditioning systems are just the tip of the iceberg). These issues have little to do with data base design. However, they may end the design of distributed data bases before it begins.

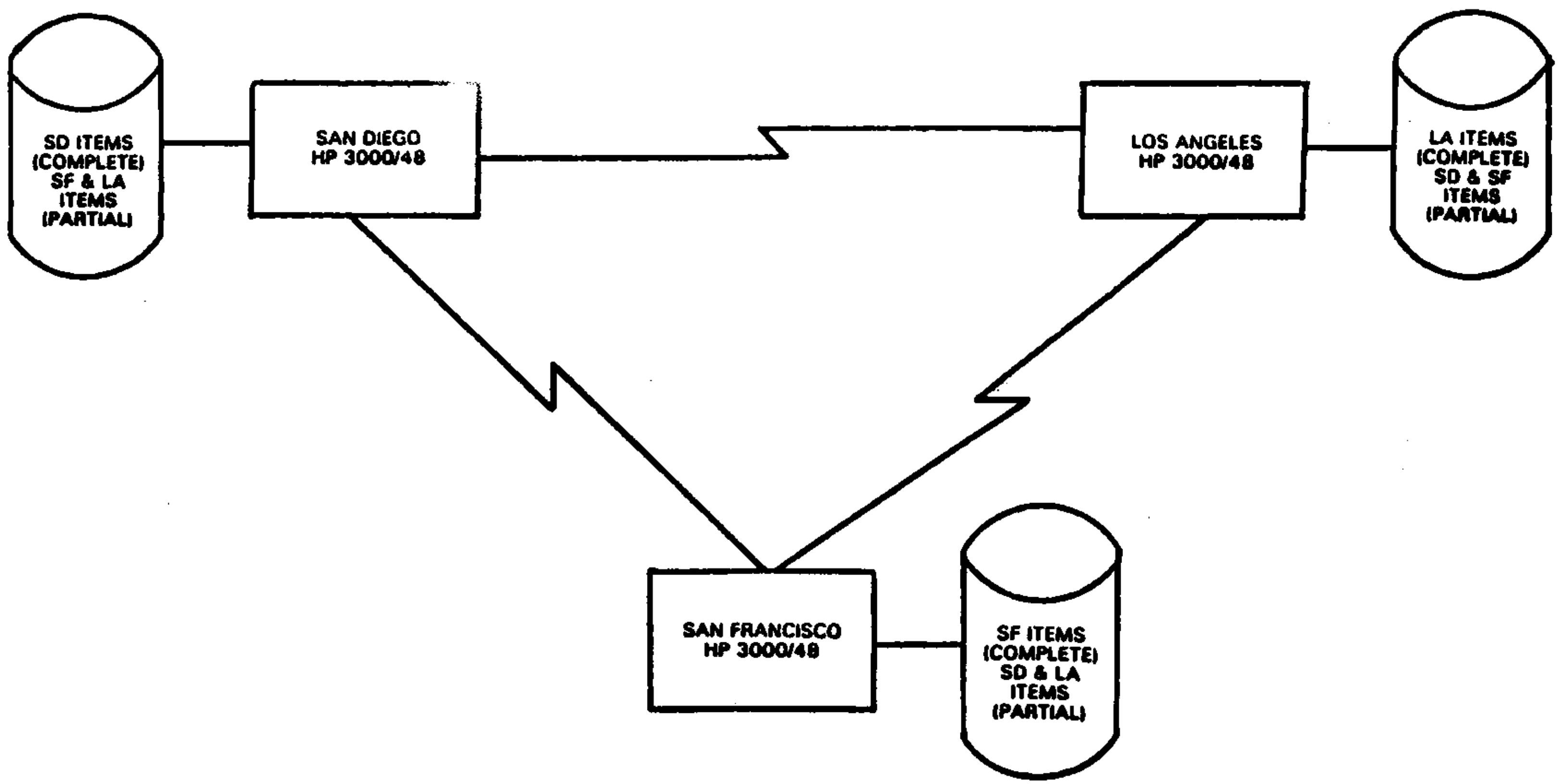

Figure 7: Combined Horizontally Replicated and Partitioned Databases

Disadvantages

On the other side of the fence, centralized data bases suffer from all of the disadvantages that are advantages for distributed data bases. Data base distribution will definitely improve data availability, as well as system survivability, given probable redundancy. Simply replicating data bases between CPUs will help to ensure that data is always accessible from some CPU, given the outstanding up-time reputation of the HP-3000. For organizations which must communicate among branches separated by great geographical distances, communication costs usually can be slashed. Finally, the modularity of distributed data bases means increased flexibility (e.g., certain users only need part of a system and may free-up hardware they would otherwise access by having their own data base and their own CPU).

Concluding Remarks

In its own effort to distribute certain of its organizational functions, Hewlett-Packard Co. has learned a valuable lesson:

> "The most significant lesson we have learned from our experience, however, is that there is no one best way to process data. Information systems must be designed to match the organization they support." [7]

To this, I add that in the next decade, the main breakthrough in information system design will come in the advances which will be made in the configuration of distributed information systems because of technological improvements in data base management systems, hardware, and telecommunications.

Acknowledgment

To Chuck MacKinnon, for persuading me to write this paper and (more importantly) for his masterful editing of it. Thanks, also, to Mark Buehler for executing the diagrams and Nancy Wild for word processing the text.

Notes

1. Although reflected in a number of sources, I feel that it is best exemplified by Grayce Booth in The Distributed System Environment: Some Practical Approaches (New York: McGraw-Hill, 1981).

2. Ibid., p. 255.

3. Ibid.

4. Ibid.

5. Ibid., p. 260.

6. For outstanding coverage of DRP, I strongly recommend Andre J. Martin's Distribution Resource Planning (New Jersey: Prentice-Hall, 1983).

7. Cort Van Rensselaer, "Centralize? Decentralize? Distribute?," Datamation, 25(4):90ff., Technical Publishing, New York, April 1979.

Bibliography

Booth, Grayce M., The Distributed System Environment: Some Practical Approaches (New York: McGraw-Hill, 1981).

Martin, Andre J., Distribution Resource Planning (New Jersey: Prentice-Hall, 1983).

Van Rensselaer, Cort, "Centralize? Decentralize? Distribute?," Datamation, 25(4):90ff., Technical Publishing, New York, April 1979.

JOHN HILL

John Hill is Manager of Distribution Systems for Computer Data Corporation, a nationwide company that develops and markets business software applications for the minicomputer industry. In his current position Hill is responsible for the development and enhancement of Inventory Control, Sales Orders, Purchasing, and Distribution Resource Planning (DRP) software. In addition he provides professional consultation in distributed systems configurations to CDc's clients. He holds an M.A. in Philosophy, as well as certificates in Business Computer Programming and Business Information Systems Design from California State University, Los Angeles. Hill has taught a variety of courses at LA-area community colleges. Prior to joining Computer Data Corporation, he worked for several manufacturing and distribution organizations, including Tandon Corporation, Sunweld Fitting Company, and Kinney Brothers of California. Past positions include Senior Manufacturing Systems Analyst, Database Administrator, Project Manager, and Data Processing Manager.