International Journal of Academic Research in Business and Social Sciences
August 2012, Vol. 2, No. 8
ISSN: 2222-6990

Framework for Deploying Client/Server Distributed Database System for effective Human Resource Information Management Systems in Imo State Civil Service of Nigeria

AHAIWE Josiah
Information Management Technology Department, Federal University of Technology, Owerri - Nigeria
E-mail jahaiwe@yahoo.com

Nwaokonkwo Obi
Information Management Technology Department, Federal University of Technology, Owerri - Nigeria

Abstract

The information system is an integrated system that holds financial and personnel records of persons working in various branches of Imo state civil service. The purpose is to harmonize operations, reduce or if possible eliminate redundancy and control the introduction of “ghost workers” and fraud in pension management. In this research work, an attempt is made to design a frame work for deploying a client/server distributed database system for a human resource information management system with a scope on Imo state civil service in Nigeria. The system consists of a relational database of personnel variables which could be shared by various levels of management in all the ministries’ and their branches located all over the state. The server is expected to be hosted in the accountant general’s office. The system is capable of handling recruitment and promotions issues, training, monthly remunerations, pension and gratuity issues, and employment history, etc.

Keywords: personnel and payroll, Imo state, distributed systems, management information systems, distributed database systems, ministries, civil service.

Introduction

Many technologies have had significant impact on people’s life but none as effective as information technology since the twentieth century Nworuh G, and Ahaiwe J (2003). Information technology is the pivot on which process engineering is hinged. If properly strategized, it can improve information use and context that can enhance performance and coordinate activities across functional units as well as interact with external entities. The non application of information technology in Imo state civil service has given rise to: inconsistent personnel data; non uniformity in the management of personnel records in the ministries/departments; problems in recruitment, training, postings, transfers, seniority lists, promotions and retirements, salary related issues, enforcing statutory deductions, processing of request for loans, advances and ensuring their recovery. This situation has given rise to "Ghost workers". Ghost workers are individuals fraudulently placed on payroll without their adding value to the operations of the civil service.

It is in the light of these problems that this research work attempts to design a framework for deploying a client-server distributed database system for human resource management in Imo state of Nigeria. The present system does not share data among the ministries due to lack of appropriate network. Consequently, duplication of activities in data-gathering and entry took place, occasioning time loss, inconsistency, unreliability of the services delivered.

Human Resources Information Management Systems

According to Ralph, S George R (2003), "Human resource management information system also called personnel management information system is concerned with activities related to employees and potential employees of organization. Because the personnel function relates to all other functional areas in business, the human resource management information system plays a valuable role in ensuring organizational success."

The Imo state human resources function is responsible for attracting, developing, and maintaining the ministries' workforce. It supports activities such as identifying potential employees, maintaining complete records on existing employees, and creating programs to develop employees' talents and skills. The human resource management systems will help senior management identify the human resources requirements (skills, educational level, types of positions, number of positions, and cost) for meeting the ministries' long-term plans. Middle management will use human resource management systems to monitor and analyze the recruitment, allocation, and compensation of employees and operational managers will use the human resource systems to track the recruitment and placement of the ministries' employees. The system also helps to track those leaving the Imo state service, categorizes all the employees of the organization, and allows viewing of salary details as and when required by the top-level management in the organization. It helps to establish entry dates and appraisal dates of the employees, and their retirement and benefits. It generates various important reports that provide valuable information to top level management. Outputs from the human resource system includes: benefit reports, salary surveys, scheduling reports, training test scores, job application profiles, and needs planning reports.

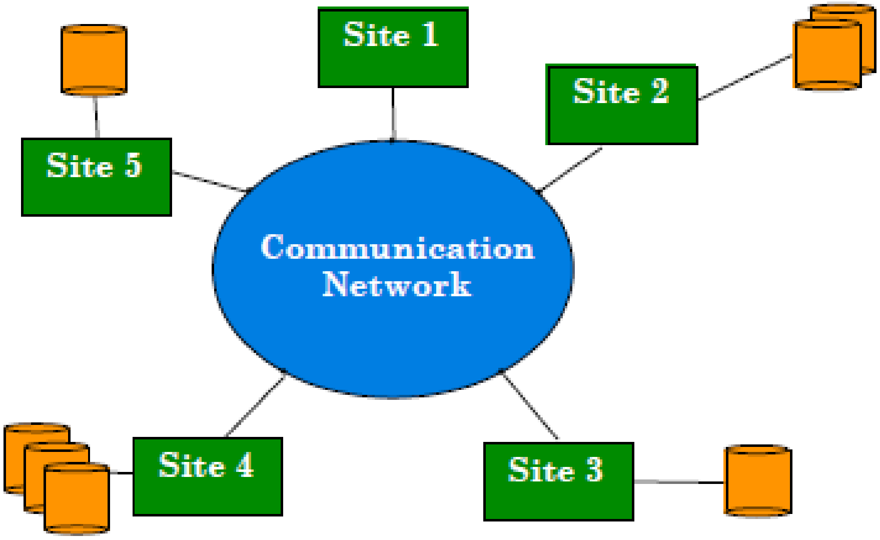
Theory of Distributed Database (DDB) and Distributed Database Management Systems (DDBMS)

Distributed Database (DDB) is a database in which portions of the database are dispersed or stored on multiple computers within a network. Users have access to the portion of the database at their location so that they can access the data relevant to their tasks without interfering with the work of others. It is a collection of multiple logically interrelated databases distributed over a computer network.

Distributed Database Management System (DDBMS) is a software system that manages the distributed database as if it were all stored on the same computer while making the distribution transparent to the user. The DDBMS synchronizes all the data periodically and, in cases where multiple users must access the same data, ensures that updates and deletes performed on the data at one location will be automatically reflected in the data stored elsewhere. The computers in distributed networks shown in Figure 1 are autonomous in the way they carry out their actions.

Distributed database management has been proposed for this work for various reasons ranging from organizational decentralization and economical processing to greater autonomy. Navathe (2000) highlighted some of these advantages as follows:

Figure 1: A Distributed Database Architecture.



a. *Management of distributed data with different levels of transparency such as*

- ***Distribution or network transparency**:* This refers to freedom for the user from the operational details of the network. It may be divided into location transparency and naming transparency. Location transparency refers to the fact that the command used to perform a task is independent of the location of data and the location of the system where the command was issued. Naming transparency implies that once a name is specified, the named objects can be accessed unambiguously without additional specification.

- ***Replication transparency**:* In this case, copies of data may be stored at multiple sites for better availability, performance, and reliability. Replication transparency makes the user unaware of the existence of copies.

- ***Fragmentation transparency**:* Two types of fragmentation are possible. Horizontal fragmentation distributes a relation into sets of rows. Vertical fragmentation distributes a relation into sub-relations where each sub-relation is defined by a subset of the columns of the original relation. A global query by the user must be transformed into several fragment queries. Fragmentation transparency makes the user unaware of the existence of fragments.

b. ***Increased reliability and availability**:* These are two of the most common potential advantages cited for distributed databases. Reliability is broadly defined as the probability that a system is running (not down) at a certain time point, whereas availability is the probability that the system is continuously available during a time interval. When the data and DBMS software are distributed over several sites, one site may fail while other site continues to operate. Only the data and software that exists at the failed site cannot be accessed. This improves both reliability and availability. Further improvement is achieved by judiciously *replicating* data and software at more than one site. In a centralized system, failure at a single site makes the whole system unavailable to all users.

c. ***Improved performance**:* A distributed DBMS fragments the database by keeping the data closer to where it is most often needed. Data localization reduces the contention for CPU and I/O services and simultaneously reduces access delays involved in wide area networks. When a large database is distributed over multiple sites, smaller databases exist at each site. As a result, local queries and transactions accessing data at a single site have better performance because of the smaller local databases. In addition, each site has a smaller number of transactions executing than if all transactions are submitted to a single centralized database. Moreover, inter-query and intra-query parallelism can be achieved by executing multiple queries at different sites, or by breaking up a query into a number of sub-queries that execute in parallel. This contributes to improved performance.

d. ***Easier expansion:*** In a distributed environment, expansion of the system in terms of adding more data, increasing database sizes, or adding more processors is much easier.

The term distributed database management system can describe various systems that differ from one another in many respects Navathe (2000).

The main thing that all such systems have in common is the fact that data and software are distributed over multiple sites connected by some form of communication network. Some of the factors that make these DDBMSs different are: the degree of homogeneity of the DDBMS software. If all servers (or individual local DBMSs) use identical software and all users (clients) use identical software, the DDBMS is called homogeneous; otherwise, it is called heterogeneous. Another factor related to the degree of homogeneity is the degree of local autonomy. If there is no provision for the local site to function as a stand-alone DBMS, then the system has no local autonomy. On the other hand, if *direct access* by local transactions to a server is permitted, the system has some degree of local autonomy.

At one extreme of the autonomy spectrum, we have a DDBMS that "looks like" a centralized DBMS to the user. A single conceptual schema exists, and all access to the system is obtained through a site that is part of the DDBMS—which means that no local autonomy exists. At the other extreme we encounter a type of DDBMS called a *federated DDBMS* (or a *multi-database system*). In such a system, each server is an independent and autonomous centralized DBMS that has its own local users, local transactions, and DBA and hence has a very high degree of *local autonomy.*

There are three alternative approaches to separating functionality across different DBMS-related processes; these alternative distributed DBMS architectures are called Client-Server, Collaborating Server, and Middleware.

Distributed Database Systems

Distributed database is a set of logically related databases that are stored on computers at several geographically sites and linked by means of computer network. These interrelated databases work together to perform certain specific tasks. Distributed computer systems work by splitting a larger task into a number of smaller ones that can then be solved in a coordinate fashion. Each processing unit can be managed independently and can develop its application.

Navathe (2000) described a distributed (DDB) as a collection of multiple logically interrelated databases distributed over a computer network, and a distributed database management system (DDBMS) as a software system that manages a distributed database while making the distribution transparent to the user. From the definition of Navathe (2000), the elementary unit of a distributed database is a computer that is networked with other computers; the computer being autonomous in the way it carries out its functions. Computers are linked to one another over a communications network that enables an exchange of message between them. The objective of this message exchange is to achieve cooperation between computers for the purpose of attaining a common goal. In this research an attempt is made to design a framework for client server distributed database system for deploying human resource management information systems for Imo state civil service. The database is hosted by the office of the Accountant General of Imo state who is the custodian of the system and it supervises the account of the state ministries and extra ministerial department. The concept of data replication and fragmentation is adopted by other sites to have access to data/records in the database. The operation done by the system includes:- processing of staff emoluments on a monthly basis, retrieval of personnel information, monitoring of unutilized funds derived from unpaid staff salaries, monitoring of ‘ghost workers’ , update of records in the database, collection of existing personnel records, capturing of new personnel records including photograph and biometric data (fingerprints) etc.

The client-server architecture is used as a platform for database application development and is an approach which has successfully been used to solve some lingering problems in society today. A distributed system is an information-processing system that contains a number of independent computers that cooperate with one another over a communications network in order to achieve a specific objective (Kay Romer et al., 2006).

Distributed database bring the advantages of distributed computing to the database management domain. A distributed computing system consists of a number of processing elements, not necessarily homogeneous, that are interconnected by a computer network, and that cooperate in performing certain assigned tasks. A physical view of a distributed system includes computers as nodes of the communications network along with details about the communications network itself. In contrast, a logical view of a distributed system highlights the applications aspects.

Figure 2 shows the communication network of some ministries and their communication networks in the Office of Accountant General of Imo state. Each of these ministries is regarded as a site. The diagram can be interpreted as a set of cooperating processes that describes how each ministry is given allocation funds during budget and planning. The data in the ministries are replicated to all local regional databases.

Client/Server System

A Client-Server system has one or more client processes and one or more server processes, and a client process can send a query i.e. (requests services) to any one of the server process. Clients are responsible for user-interface issues and regarded as Front-end Application. Servers manage data and execute transactions in other word provides services to the clients. They are said to be Back-end Application. Thus, a client process could run on a personal computer and send queries to a server running on a mainframe.

Once this relationship is established, one needs only define the protocols for signaling that a request is being made or serviced, whichever the case may be. The protocol translation is handled by a front-end processor[1] on the server system. The front-end, also called a gateway, translates the request into a query suitable for the server system, executes the request on the server system, and then responds to the network with the results. Ideally, the server system responds with information in the proper format for the client's system. The client system's gateway is responsible for receiving the response from the network. There are several advantages to successful implementation of the client / server architecture. First, the physical separation of the systems allows each local database management system to be truly independent of any other computer hardware and operating system. Because the local DBMS need be compatible only with the communications channel, they can be replaced without modification of the system.

A second benefit is the potential for supporting multiple "host" processors. The gateways between the hosts can handle all of the necessary translations to another host's language (or internal representation). Hence flexibility exists on both sides of the gateway, and major changes in other components of the information management system do not require any modification of the database.

In a **Collaborating Server** system, we can have a collection of database servers, each capable of running transactions against local data, which cooperatively execute transactions spanning multiple servers. When a server receives a query that requires access to data at other servers, it generates appropriate sub-queries to be executed by other servers and puts the results together to compute answers to the original query. Ideally, the decomposition of the query should be done using cost-based optimization, taking into account the costs of network communication as well as local processing costs.

The **Middleware** architecture is designed to allow a single query to span multiple servers, without requiring all database servers to be capable of managing such multi-site execution strategies.

Frame Work for Client-Server Architecture of Imo state Civil Service Human Resource Information Management Systems

The primary types of system architectures for information processing include: Service Oriented Architecture (SOA), distributive (client–server), and centralized information systems processing more commonly associated with mainframe and midrange computers. The Client–Server Architecture is considered in this case. This is network architecture in which each computer or process on the network is either a client or a server. Servers are powerful computers or processes dedicated to managing disk drives (file servers), printers (print servers), or network traffic (network servers). Clients are PCs or workstations on which users run applications. Clients rely on servers for resources, such as files, devices, and even processing power.

In a typical DDBMS, it is customary to divide the software modules into three levels:

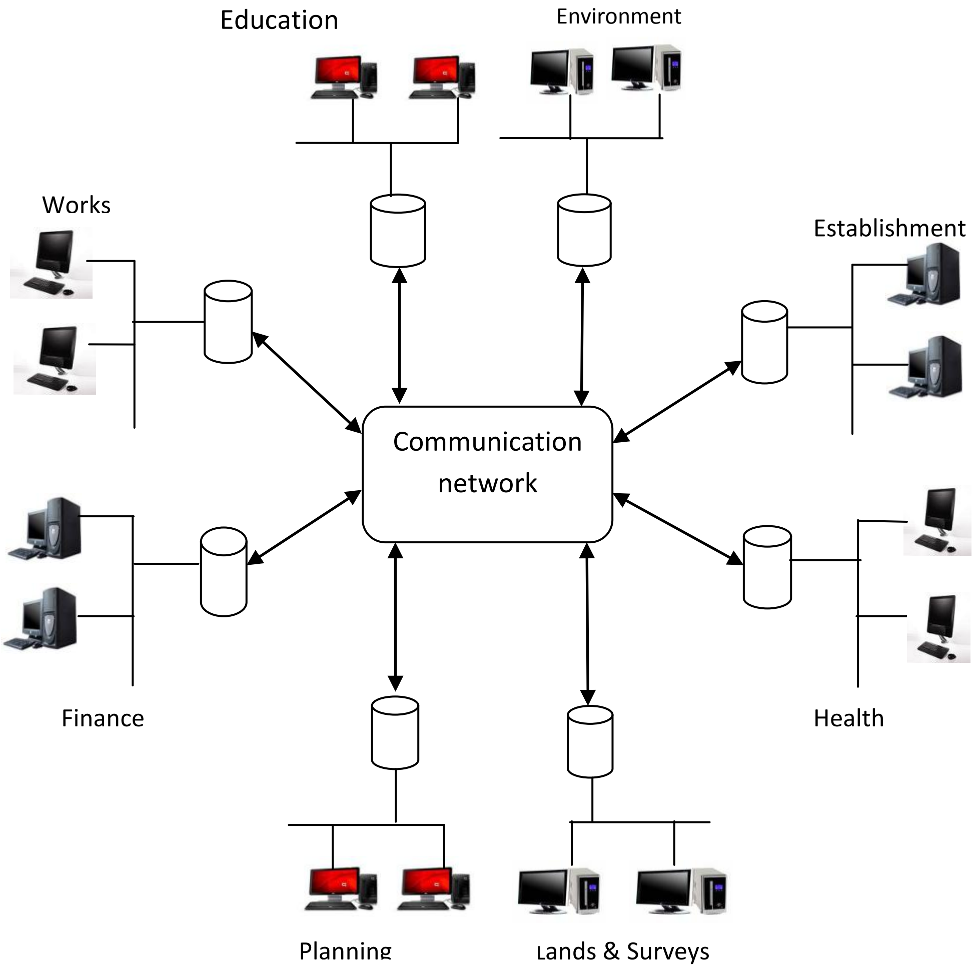
a. The server software is responsible for local data management at a site, much like centralized DBMS software.

b. The client software is responsible for most of the distribution functions; it accesses data distribution information from the DDBMS catalog and processes all requests that require access to more than one site. It also handles all user interfaces.

c. The communications software (sometimes in conjunction with a distributed operating system) provides the communication primitives that are used by the client to transmit commands and data among the various sites as needed. This is not strictly part of the DDBMS, but it provides essential communication primitives and services.

The client is responsible for generating a distributed execution plan for a multi-site query or transaction and for supervising distributed execution by sending commands to servers.

Figure 2: Client/server DDB for Imo state civil service Ministries



System Design

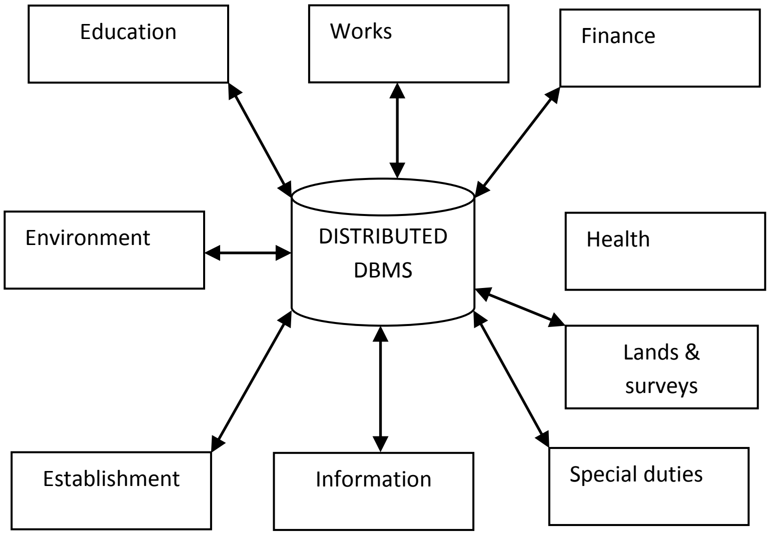
A list of some documents and information requested at the various stages of the design are presented below:

a. Employee Id
b. Employee Name
c. Ministry type
d. Employee Dept
e. Employee Address
f. Employee Age
g. Marital Status

h. Employee Sex
i. Employee Qualifications
j. Employee Work Experience
k. Job Title
l. Date Employed
m. Employee Salary
n. Hours Worked i.e. month, year etc
o. Incentives
p. Awards
q. Leave Allowance
r. Federal Dues
s. State Dues
t. Union Dues
u. Tax i.e. federal, state, union, pension plan etc
v. Gross Pay
w. Net Pay
x. Date of termination
y. Termination reason
z. Employee's Bank Account Number

Application Procedure

In this frame work, each client performs its function at its site but shares data with other sites. Each employee provides valuable data to the human resource system which does the following-maintaining and updating records of existing and left employees and categorize them on the basis of their unit, departments, designation. Reports and queries are also produced. If an employee is transferred from one ministry to the other, his/her data is still accessed.



Data Flow of the Distributed System

Components Of The Distributed System

The distributed system has Internet facing Web-based applications that can be accessed remotely by the users either within the confines of the organization or remotely. The following is list of the information technology (IT) infrastructure components of the system

- *Firewall*: A system designed to prevent unauthorized access to or from a private network. Firewalls can be implemented in both hardware and software, or a combination of both. Firewalls are frequently used to prevent unauthorized Internet users from accessing private networks connected to the Internet, especially intranets. All messages entering or leaving the intranet pass through the firewall, which examines each message and blocks those that do not meet the specified security criteria. There are several types of firewall techniques.

- *Packet Filter*: Looks at each packet entering or leaving the network and accepts or rejects it based on user-defined rules. Packet filtering is fairly effective and transparent to users, but it is difficult to configure. In addition, it is susceptible to Internet Protocol (IP) spoofing.

- *Application Gateway*: Applies security mechanisms to specific applications, such as File Transfer Protocol (FTP) and Telnet servers. This is very effective but can impose performance degradation. *Circuit-Level Gateway*: Applies security mechanisms when a Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) connection is established. Once the connection has been made, packets can flow between the hosts without further checking.

- *Proxy Server*: Intercepts all messages entering and leaving the network. The proxy server effectively hides the true network addresses.

- *Router*: A router is a special purpose computer or software device that enables two or more dissimilar networks to communicate. Routers route traffic, which consists of Transmission Control Protocol/Internet Protocol (TCP/IP) packets.

- *Host*: A computer that is connected to a TCP/IP network, including the Internet.

- *Server(s)*: A server is a dedicated computer that allows other computers to connect to it. Various types of servers exist which include the following:
  Domain Name System
  Web servers
  Internet banking servers
  E-mail servers
  Proxy servers

- *PC Workstations*: In networking, a workstation refers to any computer connected to a local area network. It could be a workstation or a personal computer.

- *Intrusion Detection Systems*: Intrusion detection is fundamentally the process of monitoring computer networks and systems for violations of computer policy.

Conclusion

The client-server architecture is used as a platform for the distributed database application development. The essential decision variables were identified. The Database is expected to be hosted by the office of the Accountant General while the concept of data replication and fragmentation is adopted by other sites (ministries) to have access to data/records in the database. The system is intelligent and capable of maintaining and updating records of existing and left employees and categorizes them on the basis of their unit, departments, designations, etc.

References

Andrade, J. M. *(ed.), Dwyer, T., Felts, S.,* and *Carges, M.(* 1996*). The Tuxedo System*: Software for Constructing and Managing Distributed Business Applications. Reading, MA *Addison-Wesley.*

Dates, C. J. (1986). *An Introduction to Database System, Fourth Edition*. Addison-Wesley

Publishing Company: Los Angeles, CA.

Kay-Romer, P., Pilhofer, F. and Arno, P.(2006). *Distributed System Architecture*. Morgan Kaufmann Publishers: San Francisco, CA.

Mowbray, T. J., and *Zahavi*, *R*. (1995). *The Essential CORBA*. New York: John Wiley.

Navathe, E. (2000). *Fundamental of Database Systems, Third Edition.* Teturo Sawada, Exclusive Publisher and Distributor.

Ramakrishnan, R. and Gehrke, J. (2004). *Database Managenment Systems, Second Edition*. McGraw-Hill: New York, NY. http://www.cs.wisc.edu/~dbbook Hicad payroll system

Ralph, S, George R (2003). Fundamentals of Information Systems, Course technologiesThomson place Boston,

Tanenbaum, A. S. (2008). *Computer Networks, Foruth Edition*. Prentice Hall: Princeton, NJ. http://www.cs.vu.nl/~ast/