DATA MANAGEMENT AND TRANSFER IN DISTRIBUTED E-COMMERCE SYSTEM

Soe Moe Myint[1], Moe Moe Myint[2], Aye Aye Cho[3]

[1]*Lecturer, Faculty of Computer Systems and Technologies, University of Computer Studies, Pathein, Myanmar*
[2]*Lecturer, Information Technology Support and Maintenance, University of Computer Studies, Pathein, Myanmar*
[3]*Associate Professor, Faculty of Computer Science, University of Computer Studies, Hinthada, Myanmar*

ABSTRACT

Nowadays almost every organization has mobile workers or distributed stations. These workers could include but are not limited to service personal telecommuters, executives, sale personnel, field engineers, and others who are rear in the main office. These workers require the successful deployment of applications; data is vital and can yield benefits such as cost saving, better and more timely decision making, customer report, and improved productivity. These benefits can transform normal business processes into a corporate strategic advantage. This paper is intended to allow the teams of people to work together over a network as if they are in the same physical location. It will be implemented as two portions: HOST (main office) which will perform data management based on Business Rules and sharing rules, and CLIENT which will synchronize with HOST to send and receive data. It will also provide two services-reliable and high-speed transports.

Keywords: *Host, Client, synchronization, data compression method*

1. INTRODUCTION

Data-intensive, high-performance computing applications require the efficient management and transfer of terabytes or petabytes of information in wide area, distributed computing environments. In such applications, massive datasets must be shared by community of hundreds or thousands of employee (workers) distributed worldwide. These workers need to be able to transfer large subsets of these datasets to local sites or other remote resources for processing. They may create local copies or replicas to overcome long wide-area data transfer latencies. The data management environment must provide security services such as authentication of users and control over who is allowed to access the data. In addition, once multiple locations, researchers need to be able to locate copies and determine whether to access an existing copy or create a new one to meet the performance needs to their application.

This system is intended to implement the data management system and synchronization of field users to the HOST system. The system and its contents reside in their entirety on multiple endpoints. The synchronization ensures that each endpoint has a complete and consistent copy. It synchronous distributed repositories, that is, collections of replicas stored in geographically dispersed nodes. A repository may be a sub-tree of local file system or a selection thereof, specified by regular expression.

2. BACKGROUND THEORY

E-Commerce applications operate on and generate large amounts of data. The data generated by E-Commerce application is two types: experimental data, or information collected from Sales, Production, sent by field users or brought from data provider company; and metadata, or information about the experiment data ,such as the number of events or the results of analysis.

File sizes and numbers of files are determined to some extent by the type of software used to store experimental data and metadata. Access patterns vary for experimental data files and metadata. Experimental data files typically have a single creator. During an initial production period lasting several weeks, there files are modified as new objects are added. After data production is complete, files are not modified. In contrast, metadata files may be created by multiple individuals and may be modified or augmented over time, even after the initial period of production.

The consumers of E-Commerce data and metadata will be number in the hundreds or thousands. These users are distributed at many sites worldwide. Hence, it is often desirable to make copies or replicas of the data being analyzed to minimize access time and network load.

2.1 Overview of the System Design

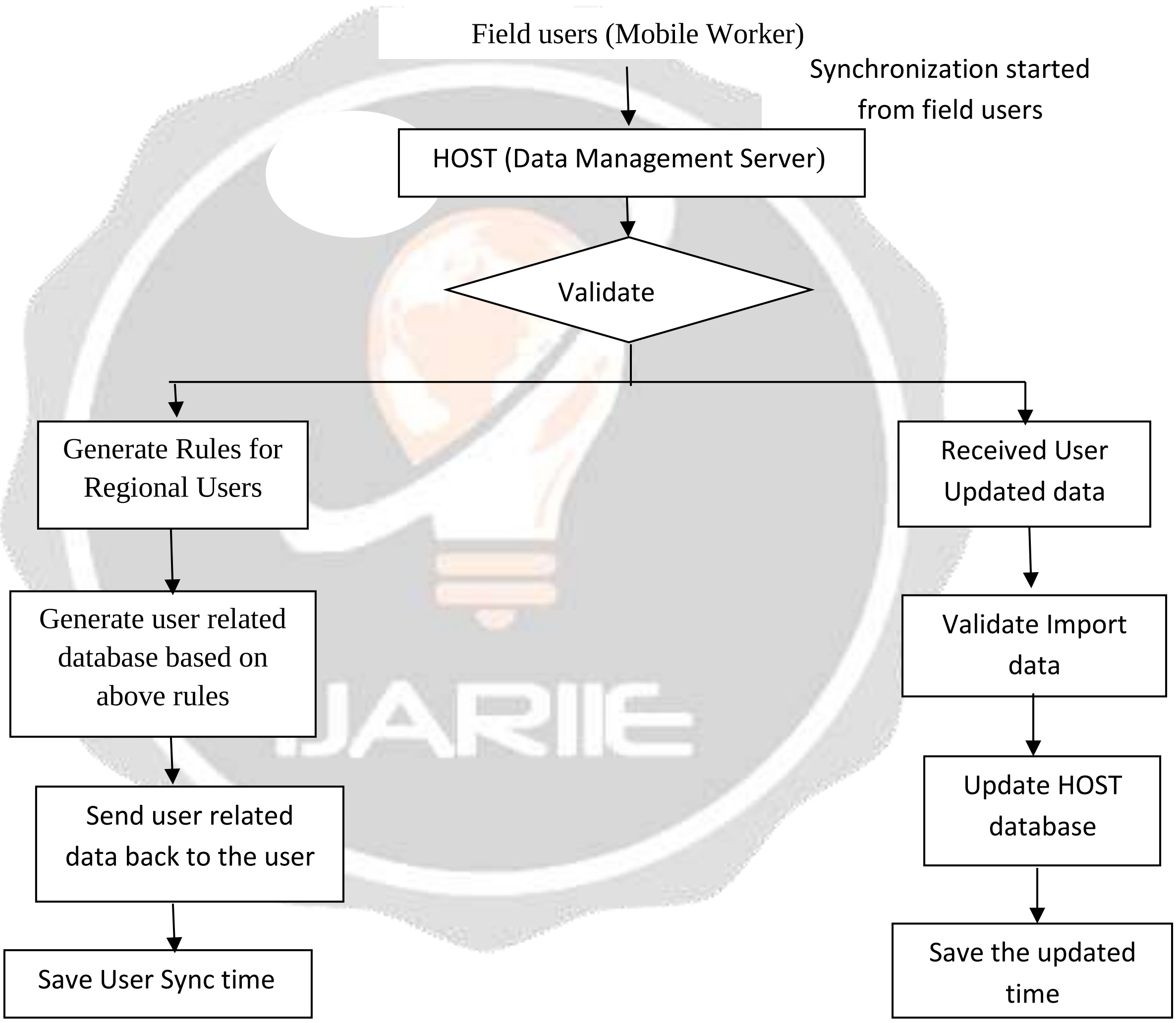

Figure1. Overall system flow diagram

2.2 Data Management Architecture

The four principles derive the design of data management architecture. These principles derive from the fact that data management applications must frequently operate in wide area, multi-institutional, heterogeneous environments. The four principles are (1) Mechanism neutrality (2) Policy neutrality (3) Compatibility with Data Management infrastructure (4) Uniformity of information infrastructure. These four principles lead to develop a layered architecture. Figure 2 in which the lowest layers provide high performance access to an orthogonal set of basic mechanisms, but do not enforce specific usage policies. Rather, such policies are implemented in higher layers of the architecture, which build on the mechanisms provided by the basic components.

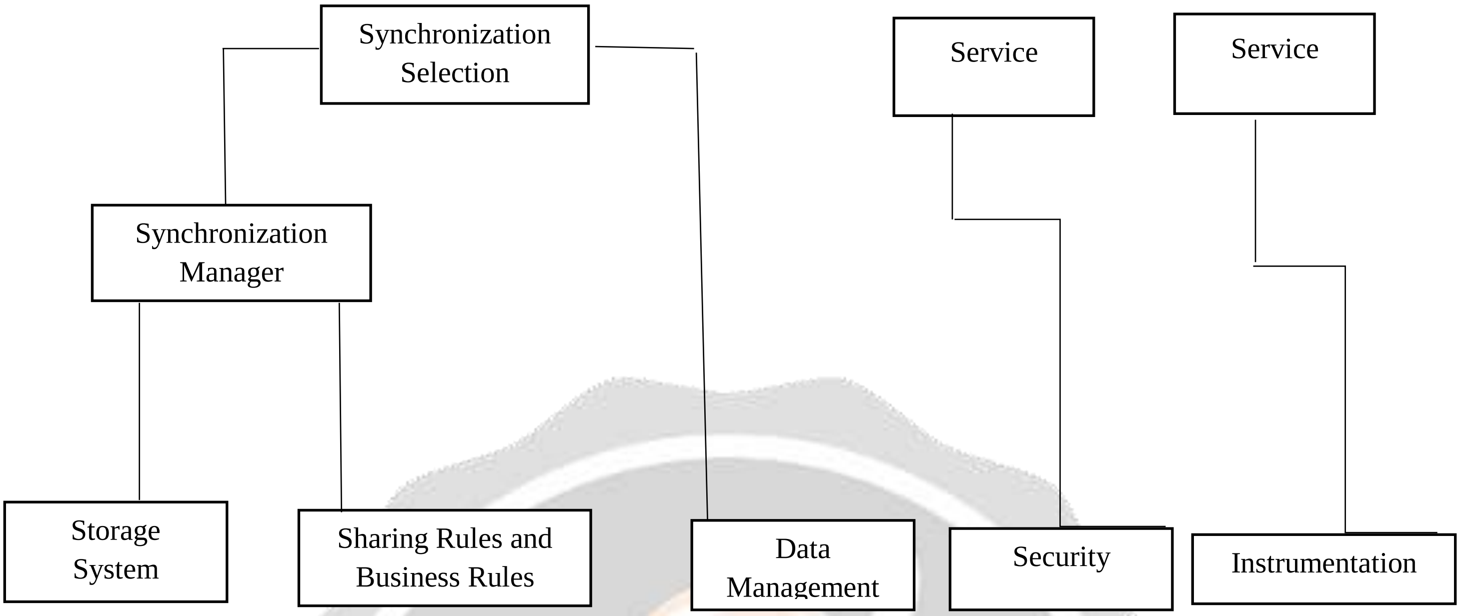

Figure2. Major Components and Services of Data Management System

2.3 Methods used to gain high performance

-Synchronizing large amount of replicated data sets in distributed system.
-Parallelism in the planning and the synchronizing phase
-Instead of the whole file, synchronizing only the updated parts
-Using sophisticated compression methods

2.3.1 Data Compression Methods

```
Begin
    If CheckUpdate( )=TRUE then
    If(CheckRelatedUser(userid)=FALSE)then break;
        Get Updated records from the database
        Prepare flat file// in order to decrease the size
        Write updated records to the flat file//
        eg.10001/H20001/500//that means sales information of hospital ID
        H20001 for product ID 10001 for product ID10001 and quantity is 500
        Send this flat file to user repository
    End if
End
```

2.3.2 Parallel Processing

```
Begin
    While(ReceivingRequest)
    Create a thread for a user//so that host can perform synchronization for many users in parallel
    End while
End
```

2.3.3 In Each Thread

```
Begin
    Do Synchronizing for specific user
```

End

2.4 Efficiency of the System

Offline Usage: A direct result of that synchronization is the ability to use any application while disconnected from the network .Any changes made to the application by any of the participants are propagated across the network. When the offline user returns to the network, all changes are forwarded and synchronized, even if other users have gone offline in the meantime.

2.4.1 Process Flow Diagram of the HOST (Main Office)

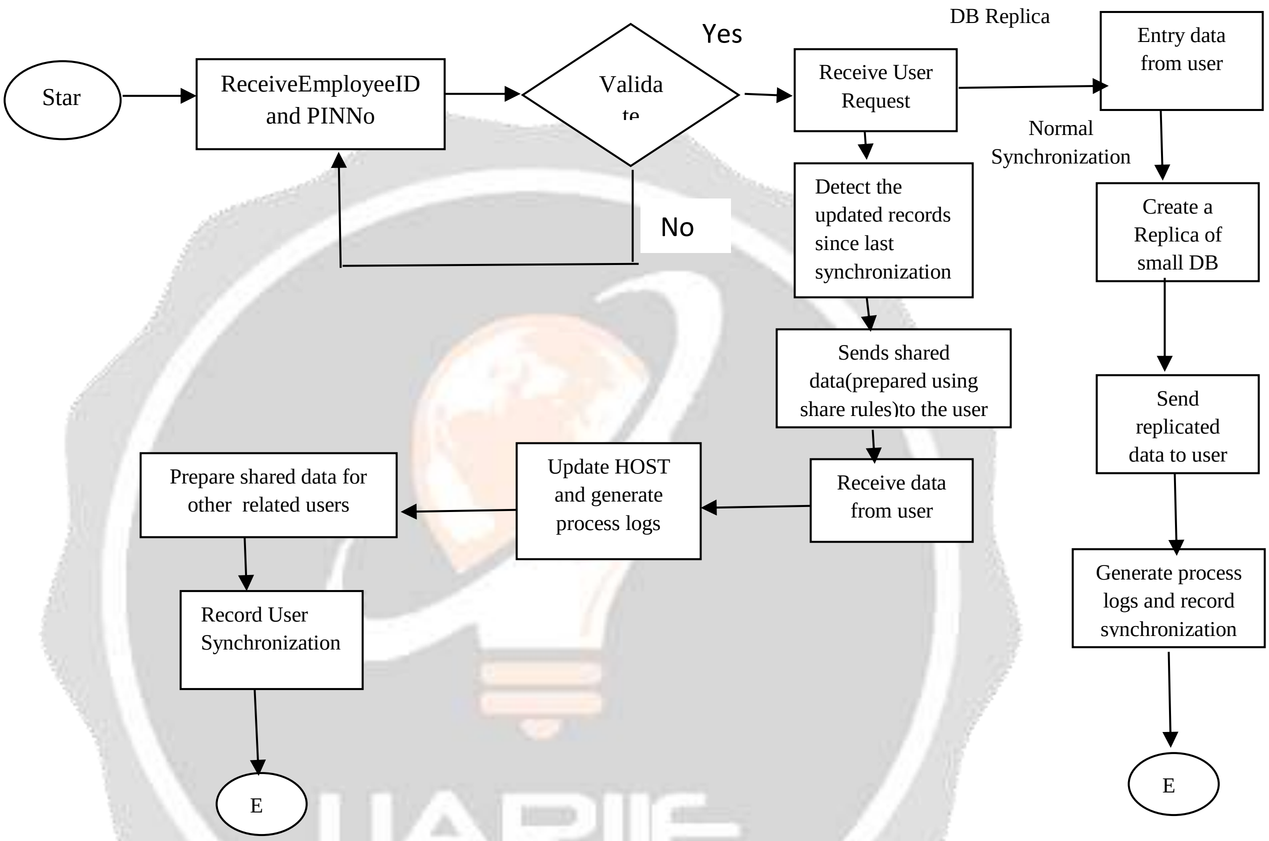

Figure 3.Process Flow of the HOST

2.4.2 Process Flow Diagram of the Clients (Field Users)

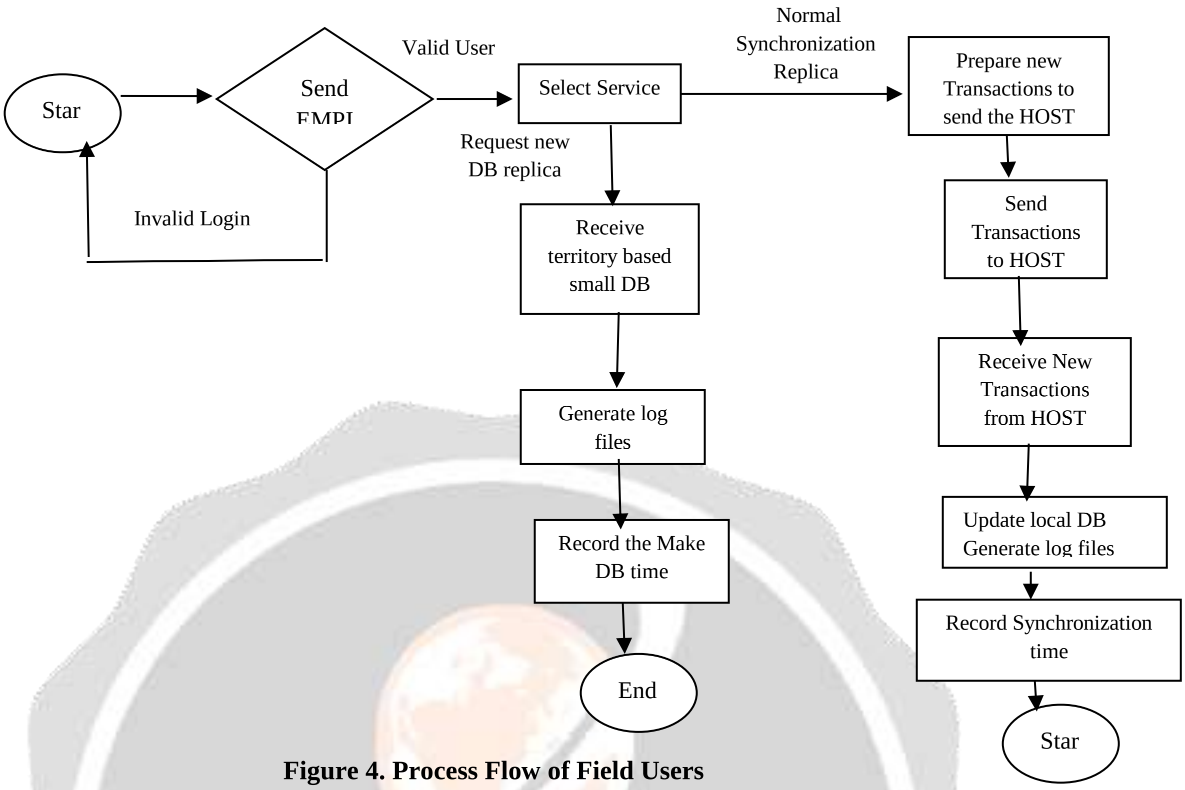

Figure 4. Process Flow of Field Users

When field user starts synchronization process, the system asks the user login to validate the unauthorized usage. Then User Credentials (encrypted) are sent to the HOST. Then if it is an authenticated user, then the user starts requesting services. There are two types of request in synchronization, the first is requesting a new set of data replica and the second is updating data sets. The first one is simply generating a new set of database and sends it to the user. The synchronization is done in two steps, the first for deleting records that have been modified since the last synchronization, and the second for propagating the update data.

4. CONCLUSION

This system is efficient for synchronization and distribution in Distributed System environments. It transmits only modified blocks rather than complete files and it uses efficient data compression algorithm to reduce the communication time. It also gains high performance by omitting the transfer of unnecessary metadata. It will also automatically generate a public/private key pair for encryption ensuring that all activity military-grade encryption without any intervention.

5. REFERENCES

[1] C.J.Date :"An Introduction to Database System "Seven Edition, Addision- Wesley Publishing Company ISBN 0-201-38590-2, 2000
[2]George Coulouris, Jean Dollimore and Tim Kindberg "Distributed Systems Concepts and Design "Third Edition, Addision-Wesley Publishing Company ISBN 0201-61918-0,2001
[3]http://www.umsl.edu/codeanalysis/synchronization.html
[4]http://www.balancedscored.org/appl/mobile-unit.html
[5]http://wwwnustl.com/replica-management-method/replication.html