DESIGNING AN ATM NETWORK BASED ON DISTRIBUTED DATABASE SYSTEMS

Md. Rahat Hossain*, Abdullah Azfar*, Golam Shagadul Amin Talukder, S M Faisal*, Abdullah Al Hasib***

ABSTRACT

Automated teller machines (ATM) have become a part and parcel of new generation banking. The ATM network is fully based on centralized database environment. Time of transaction varies according to the distance between the ATM and centralized database. A more convenient approach would be to make a distributed database environment for ATM networks. This will make life easier, tasks secured and efficient utilization of resources. In this paper we have proposed a design of an ATM network based on distributed database environment. We have also discussed the costs and benefits of our design by allocating fragments on different sites.

KEYWORDS

Automated Teller Machine (ATM), Host Processor, Distributed Database Management System, Fragments

1. INTRODUCTION

Automated teller machines (ATM) can now be found at most supermarkets, convenience stores and travel centers. When anybody is short on cash, he can walk over to the ATM, insert his card into the card reader, respond to the prompts on the screen, and within a minute he walks away with his money and a receipt. All the ATM machines working around the world are based on the concept of centralized database system. In this paper we discuss a distributed approach for the ATM networks.

2. PRESENT WORKING PRINCIPLE

A detailed survey on ATM network can be found in [1]. An ATM is simply a data terminal with two input and four output devices. Like any other data terminal, the ATM has to connect to, and communicate through, a host processor. The host processor is analogous to an Internet service provider (ISP) in that it is the gateway through which all the various ATM networks become available to the cardholder (the person wanting the cash).

* Department of Computer Science and Information Technology, Islamic University of Technology, Board Bazar, Gaziopur-1704, Bangladesh
** Metropolitan University, Sylhet, Bangladesh

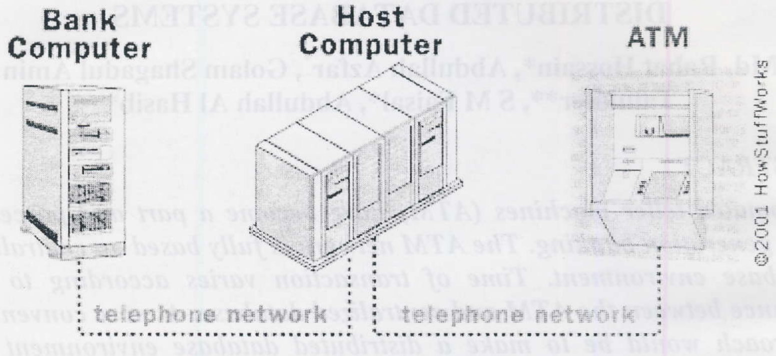

Figure 1: ATM connections [1]

When a cardholder wants to do an ATM transaction, he or she provides the necessary information by means of the card reader and keypad. The ATM forwards this information to the host processor, which routes the transaction request to the cardholder's bank or the institution that issued the card. If the cardholder is requesting cash, the host processor causes electronic funds transfer to take place from the customer's bank account to the host processor's account. Once the funds are transferred to the host processor's bank account, the processor sends an approval code to the ATM authorizing the machine to dispense the cash. The processor then ACHs (Automated Clearing House - a bank terminology means that a person or business is authorizing another person or business to draft on an account) the cardholder's funds into the merchant's bank account, usually the next bank business day. In this way, the merchant is reimbursed for all funds dispensed by the ATM.

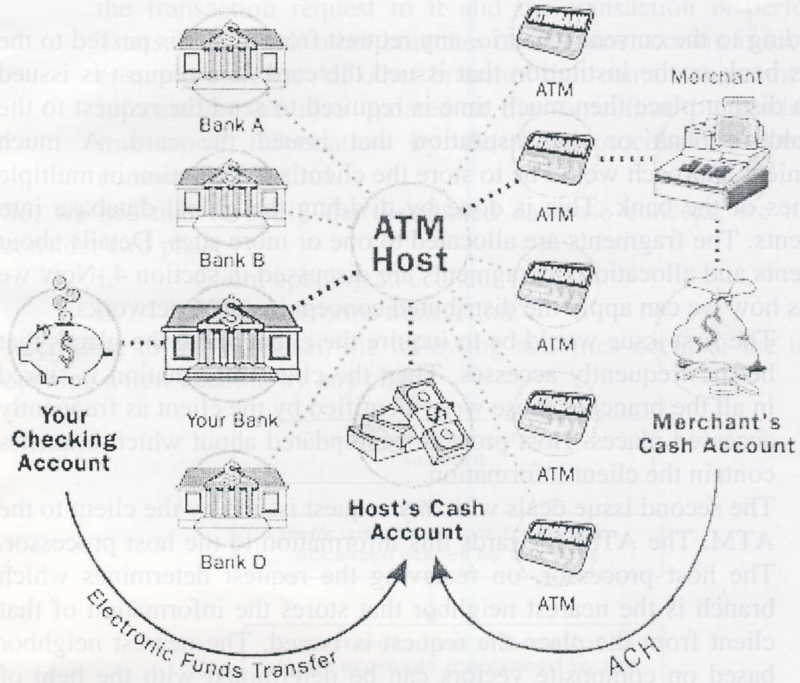

Figure 2: Architecture of ATM network [1]

2.1 DRAWBACKS OF THE PRESENT SYSTEM

i) Every transaction is a centralized operation. No concept of any local operation.

ii) Transaction processing time varies on the distance between the client's machine and the bank. For long distance it requires much longer time.

iii) For concurrent transaction requests by multiple clients the requests are stored in a queue. Clients who place their request later may have to wait until; previous clients are served.

iv) If there is a link failure between the client's machine and the central database then no transaction can be performed. Clients have to wait until the link is reestablished.

3. APPLYING DISTRIBUTED CONCEPTS

According to the current scenario, any request from a user is passed to the client's bank or the institution that issued the card. If a request is issued from a distant place then much time is required to send the request to the cardholder's bank or the institution that issued the card. A much convenient approach would be to store the client's information in multiple branches of the bank. This is done by dividing the global database into fragments. The fragments are allocated to one or more sites. Details about fragments and allocation of fragments are discussed in section 4. Now we discuss how we can apply the distributed concept in ATM networks.

- The first issue would be to inquire the client about the places that he/she frequently accesses. Then the client information is stored in all the branches those were identified by the client as frequently accessed places. Host processor is updated about which branches contain the client information.
- The second issue deals with any request issued by the client to the ATM. The ATM forwards this information to the host processor. The host processor, on receiving the request determines which branch is the nearest neighbor that stores the information of that client from the place the request is issued. The nearest neighbor based on composite vectors can be determined with the help of the following equation [2]:

Metric = K1 * Bandwidth + (K2 * Bandwidth) / (256 –Load) + K3 * Delay

Where

Metric = factor that store a host processor in its Database for its every neighbor.

K1, K2, K3 = constant used to weight the effect of the routing metrics. The default values for K1 and K3 are = 1, and the default value for K2 is 0. The network's administrator can modify all these constants.

If we consider the default constant values, the basic metric calculation reduces to just this:

Metric = Bandwidth + Delay

Regardless of whether we use the default values, the result of the metric calculation is a composite metric that can fairly evaluate potential routes through a highly diverse network. The value of this composite metric is frequently described as the cost of a route. The route with the lowest cost is the best route.

- After determining the nearest neighbor, the host processor routes the transaction request to it and the transaction is performed. Transaction in all other branches at the same account is locked to ensure atomicity. All other branches that contain the same client information are also updated at the same time by sending the transaction information by the host processor to maintain concurrency control.

So, we see that to apply the distributed database concept, we have to consider two phases :

i. Client opening an account.
ii. Client performs transaction.

According to our proposal, the following activities occur at the time of opening a new account by any client:

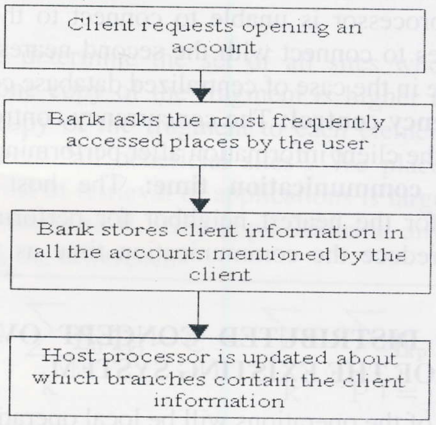

Figure 3: Initial Account opening procedure

During the phase of transaction, the following activities occur:

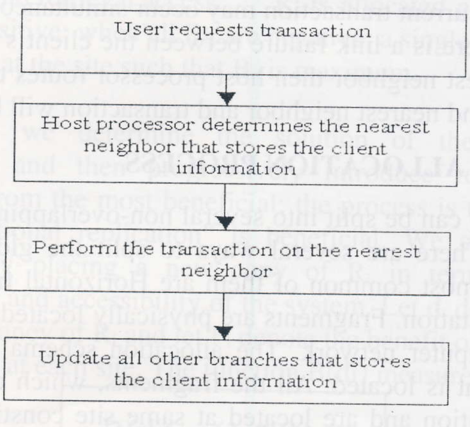

Figure 4: Transaction Processing

3.1 PERFORMANCE ISSUES

By applying the distributed concept to the existing ATM system, the performance is increased in following ways.

- **Interconnection among existing databases:** Client information is distributed in multiple branches. Our concept provides the interconnection among existing databases by maintaining a physical correlation with all the information of the client of different branches.
- **Incremental growth:** It becomes easier for the organization to add new clients on different branches by applying the distributed approach.
- **Reliability:** Another valuable benefit of our concept is reliability of data. The client information is stored in multiple branches. If the host processor is unable to connect to the nearest neighbor then it tries to connect with the second nearest neighbor. This is impossible in the case of centralized database concept.
- **Concurrency control:** The concurrency control is maintained by updating the client information after performing any transaction.
- **Reduced communication time:** The host processor always searches for the nearest neighbor for performing transaction. It helps to reduce the communication time as well as transaction time.

3.2 HOW THE DISTRIBUTED CONCEPT OVERCOMES THE LIMITATIONS OF THE EXISTING SYSTEM

i) Most of the operations will be local operations.
ii) Transaction is performed in nearest neighbor. So the processing time is optimal.
iii) Concurrent transaction may occur simultaneously.
iv) If there is a link failure between the client's machine and the nearest neighbor then host processor routes the request to the second nearest neighbor and transaction will be performed.

4. FRAGMENT ALLOCATION PROCESS

A global relation can be split into several non-overlapping portions called fragments [3]. There are several ways to split the global relation into fragments. The most common of them are Horizontal fragmentation and Vertical fragmentation. Fragments are physically located at one or several sites of the computer network. The allocation schema defines at which site(s) a fragment is located. All the fragments, which correspond to the same global relation and are located at same site constitute the physical image of that global relation.

4.1 MEASURING COST AND BENEFITS OF FRAGMENT ALLOCATION

For our proposed architecture we use horizontal fragmentation. We discuss the costs and benefits of the allocation of fragments of a global relation R. We consider the following definitions [3]:

i is the fragment index

j is the site index

k is the application index

f_{kj} is the frequency of application k at site j

r_{ki} is the number of retrieval references of application k to fragment i

u_{ki} is the number of update references of application k to fragment i

$n_{ki} = r_{ki} + u_{ki}$

1) At first we determine the set of all sites where the benefit of allocating one copy of the fragment is higher than the cost, and allocate a copy of the fragment to each element of this set; this method selects "all beneficial sites". We place R_i at all sites j where the cost of retrieval of applications is larger than the cost of update references to R_i from applications at any other site. B_{ij} is evaluated as the differences:

$$\boxed{B_{ij} = \sum_{k} f_{kj}\, r_{ki} - C * \sum_{k} \sum_{P \neq j} f_{kp}\, u_{ki}}$$

C is a constant, which measures the ratio between the cost of an update and retrieval access. R_i is allocated at site j such that Bij is positive; when all Bij are negative, a single copy of R_i is placed at the site such that B_{ij} is maximum.

2) Secondly we determine the solution of the non-replicated problem, and then progressively introduce replicated copies starting from the most beneficial; the process is terminated when no "additional replication" is beneficial. We can measure the benefit by placing a new copy of R_i in terms of increased reliability and accessibility of the system. Let d_i denote the degree of redundancy of R_i and let F_i denote the benefit of having R_i fully replicated at each site. The function B(di) measures this benefit.

$$\boxed{B(di) = (1-2^{i-di})\, Fi}$$

Then we evaluate the benefit of introducing a new copy of R_i at each site j by modifying the previous formula as follows:

$$B_{ij} = \sum_{k} f_{kj}\, r_{ki} - C * \sum_{k} \sum_{P \neq j} f_{kp}\, u_{ki} + B(di)$$

This formula takes into account the degree of replication.

4.2 WHY AVOIDING VERTICAL FRAGMENTATION

For a banking operation, we need to store particular client's information together. It is unusual to store client's name in one place and account number in another place. Rather all the information is kept together for easier access. For this reason we have avoided vertical fragmentation. If necessary then vertical fragmentation can also be applied. It just depends on the designer of the database how he/she wants to design the database.

5. CONCLUSIONS

ATM networks are the wave of new generation banking. Today, ATM networks have conquered normal banking procedures by rendering 'smarter' services. Our proposed distributed approach for ATM networks can be a way to properly utilize and different resources. This utilization is essentially required and its' necessity will increase rapidly with time. It will not only make the life easier but also will make the tasks secure and resource efficient. We expect that the proposed distributed approach for ATM networks will become feasible and pervasive for the next generation banking. The actual contributions made by the proposed concept are:

- Most of the operations will be local operations.
- Transaction is performed in nearest neighbor. So the processing time is optimal.
- Concurrent transaction may occur simultaneously.
- If there is a link failure between the client's machine and the nearest neighbor then host processor routes the request to the second nearest neighbor and transaction will be performed.

6. REFERENCES

[1] http://www.howstuffworks.com

[2] Mark A. Sportack, "*IP Routing Fundamentals*", Cisco Press, First edition, February 19, 1999, ISBN: 1-57870-071-x.

[3] Stefano Ceri, Giuseppe Pelagatti, "*Distributed Database Principles and Systems*", McGraw-Hill Book Computer Science Series, International Edition, 1987. pp. 84-85.