Performance Issues In Distributed Real - Time Database Systems

Ms.Bhanu Bhardwaj

Computer Science Department Dronacharya Group of Institution G. Noida
Bhanubhardwaj10@gmail.com

Abstract - Database Management System (DBMS) uses external magnetic device(disks) for the storage of mass data. They offer low cost per bit and non volatility, which makes them indispensible in today's DBMS technology. However, under commercially available operating systems, data can only be manipulated (i.e., Compared, inserted, modified and deleted) in the main storage of computer. Therefore part of the database has to loaded in to main storage area before manipulated and written back to the disk after modification, A database buffer had to be maintained for purpose of interfacing main memory and disk.

KEYWORDS : Database management System, Distributed system, DDBS, Real time System, Query Processing

I. INTRODUCTION

Modern technologies have provided portable computers with wire-less interfaces that allow networked communication even while a user is mobile. Wireless networking greatly enhances the utility of a portable computing device. The combination of networking and mobility will engender new technical challenges that mobile computing must surmount to achieve under investigation and also consider their limitations. In the last few years, the use of portable computers and wireless networks has been widespread. The combination of both opens the door to a new technology: mobile computing. Although the wireless communication networks were designed for the transport of voice signals, their use for data transport is growing. Mobile computing allows users to access from anywhere and at anytime the data stored in repositories of their organizations (i.e. the DBs of the company in which they work) and also for available data in a global information system through Internet. Computers are typically configured for use in a single location. The shift toward mobility and wireless communication is testing the abilities of designers to adapt traditional system structures. Many professionals use mobile computers for their work: sales personnel, emergency services etc. in order to obtain and send information in the place where and at the moment when they actually need it. Moreover, there exist applications in this new framework where the location is an important aspect, such as those that provide information about the nearest hotels, restaurants, etc. That is, mobile computing mainly depends upon the stored database and their transaction, adds a crucial challenges for the researchers. [Alfredo (2000)]

Real-time database systems support applications which have severe performance constraints such as fast response time and continued operation in the face of catastrophic failures. Real-time database systems are still in the state of infancy, and issues and alternatives in their design are not very well explored. In this paper, we discuss issues in the design of real-time database systems and discuss different alternatives for resolving these issues. We discuss the aspects in which requirements and design issues of real-time database systems differ from those of conventional database systems. We discuss two approaches to design real-time database systems, viz., main memory resident databases and design by trading a feature (like serializability). We also discuss requirements in the design of real-time distributed database systems, and specifically discuss issues in the design of concurrency control and crash recovery. It is felt that long communication delays may be a factor in limiting the performance of real-time distributed database systems. We present a concurrency control algorithm for real-time distributed database systems whose performance is not limited by communication delays.

II. THEORY

There is a growing need for real time data services in distributed environment. Modern electronics services and electronic commerce applications characterized by high volume of transactions, can't survived without an online support of computer system and database technology. Many application such as Aircraft control, Shipboard control, Military tracking, Stock arbitrage system, Sensory system and Traffic control, transaction should be processed within their deadlines using the fresh data reflect the control real world status. Existing and emerging application requires distributed database system (DDBS) consist of a collection of sites, connected together via some communication network, in which each site has a data base system site in its own right but the site is agreed to work together. So that a user at any site can access data from anywhere in the network, exactly as if the data are all stored at the user's own site.

The presence of multiple sites in distributed environment raise issue that is not present in centralized system. The Real Time System (RTS) are those system for which correctness depends not only on logical properties of the produced result, but also on the temporal properties of these result. Typically RTS are associated with critical application in which human lives or expensive machineries may be at stake. Hence in such system an action performed too late (or too early) or a computation which uses temporal invalid data may be useless and same time harmful even if such an action or computation is functionally correct. As RTS continue to evolve their application, become more and more complex and often required timely access and predictable processing of massive amount of real time data. Many of these application providing real time data services in distributed environment are essential. The issues involved in providing predictable real time data services in centralized data base system have researched as Distributed Real Time Database System (DRTDBS). DRTDBS are collection of multiple, logically interrelated database distributed over a computer networks where transaction have explicit timing constraints usually in the form of deadline. In such a system data items must be controlled in order to maintain databases logically consistency satisfying timing constraints of various real time activities. In Distributed System may be difficult due to the distributed nature of the transaction and database consistency required.

The implementation of Distributed Real Time Database Systems (DRTDBS) is difficult due the conflicting requirements of maintaining data consistency and meeting transactions deadline. The difficulty comes from the unpredictability of the transactions' response times. Each distributed transaction processing a data item takes a variable amount of time due to concurrency control, I/O and communication delays. While maintaining the consistency of

Underlying database, scheduling and management of the system resources in DRTDBS should also take into account the timing constraints. Access to CPU, main memory, I/O devices and shared data should be managed to make the best effort to satisfy the transaction deadlines.

Distributed computing allows different users or computers to share information. Distributed computing can allow an application on one machine to leverage processing power, memory, or storage on another machine. It is possible that distributed computing could enhance performance of a stand-alone application, but this is often not the reason to distribute an application. Some applications, such as word processing, might not benefit from distribution at all. In many cases, a particular problem might demand distribution. If a company wishes to collect information across locations, distribution is a natural fit. In other cases, distribution can allow performance or availability to be enhanced. If an application must run on a PC and the application needs to perform lengthy calculations, distributing these calculations to faster machines might allow performance to be enhanced. Database performance is an important aspect of database usability. Distributed real time database systems (DRTDBS) must be designed on all levels of database architecture to support timely execution of requests. The primary performance objective in DRTDBS is to minimize the number of missed deadlines. Due to the demanding nature of this objective, traditional approaches are inadequate. However, the research in DRTDBS has been mostly devoted to extending traditional transaction processing techniques to solve the issues important for the design of DRTDBS. In this environment, new policies and protocols must be designed to efficiently handle the transactions execution. [Udai Shanker (2006)]. Distributed database systems (DDBS) pose different problems when accessing distributed and replicated databases. Particularly, access control and transaction management in DDBS require different mechanism to monitor data retrieval and update to databases. Current trends in multi-tier client/server networks make DDBS an appropriated solution to provide access to and control over localized databases. Oracle, as a leading Database Management System (DBMS) vendor employs the two-phase commit technique to maintain consistent state for the database. The objective of this paper is to explain transaction management in DDBS and how Oracle implements this technique. An example is given to demonstrate the step involved in executing the two-phase commit. By using this feature of Oracle, organizations will benefit from the use of DDBS to successfully manage the enterprise data resource. Performance and cost to handle the mobile distributed database management in a business not only affect the business it also raises many problems. Performance of mobile distributed database system is heavily dependent on allocation of data among different sites, because major cost is executing queries in a mobile distributed database system is the data transfer cost incurred in moving data accessed by a query from different sites to site where the query is initiated. The static location provides only limited response to changing workload. Many algorithms is proposed for the distributed database system for non-replicated distributed database. But still more to do for replicated distributed database. The mobility is also a constraint for mobile distributed database, limited bandwidth , frequent dis-connection and failure increase the complication in available methods [Arjan Singh (2009)].

III. ISSUES IN THE DISTRIBUTED REAL-TIME DATABASE SYSTEM

The design and implementation of DRTDBS introduce several other interesting problems. Among these problems, predictability and consistency are fundamental to real time transaction processing, but sometimes these require conflicting actions. To ensure consistency, we may have to block certain transactions, however may cause several unpredictable transaction

execution and may lead to the violation of timing constraints. There are number of other sources of unpredictability such as communication delays, site failure and transactions interaction with the underlying operating system and I/O subsystems. Other design issues of DRTDBS are data access mechanism and invariance, new metrics for database correctness and performance, maintain global system information, security, fault tolerance, failure recovery, etc.

Although a lot of research has been done on these issues, there still exist many challenging and unresolved issues. Many real-time application need to share data that are distributed among multiple site. In different application remote data access consist of multi-hop network operation and take substantially more time then the local data access. Another problem is that due to the long remote data access time, by the time a transaction gets all the data it need, some of the data item may have already become stakes. The time expressed in the form of dead line is a critical factor to be considered in distributed real time transaction . The completion of transaction on or before its deadline is one of most important performance objective of DRTDBS. One of the most significant factors is the data conflict among transactions . The data conflict that occurs among executing conflict. Scheduling of distributed transactions. Optimizing the use of memory. Management of distributed transactions. Deadline assignment strategies. Possibilities of distributed deadlocks.

IV. PROBLEMS RELATED TO DBMS BUFFER MANAGEMENT

4.1 DBMS BUFFER MANAGEMENT UNDER A VIRTUAL OS

Embedding a DBMS and a OS environment, in which it is usually treated like a normal application program, can result in aggregating effects on buffer management. If DBMS runs in a virtual address space, program code as well as the DBMS buffer is paged by OS memory management, unless they are made resident in main memory. While replacement of buffer pages is done by the DBMS according to the logical references, paging of main memory frames is performed by independent OD algorithms based on the addressing behaviour within the main memory frames. In such an environment, the following kinds of faults can occur:

1. Page faults
2. Buffer faults
3. Double-page faults

V. ROUTING AND QUERY PROCESSING

The mobile computing poses typical problems from the point of view of routing and query processing. For example, as per the mobile-computing model, the route between a pair of hosts may change over time, if one of the two hosts is mobile. This simple fact may have a dramatic effect at the network level, since location-based network addresses are no longer constants within the system. However, these networking issues are beyond the scope of this course.

The mobile-computing model also directly affects database query processing. In the case of distributed query processing, the communication costs play important role in query optimization process while selecting the best method of query evaluation strategy [Madria (1998)]. Mobility results in dynamically changing communication costs, therefore, complicate the optimization process. User time is a highly valuable commodity in most of the business applications. Connection time is the unit of monetary charges is assigned in most cellular systems; therefore, it should be minimum. One of the basic principles of radio communication is that it requires less energy to receive than to transmit radio signals [Dunham (1998)] thus, transmission and reception of data impose different power demands on the mobile host.

VI. BROADCAST DATA

It is often desirable for frequently requested data to be broadcast in a continuous cycle by mobile support stations, rather than transmitted to mobile hosts on demand. A typical application of broadcast data is stock-market price information. There are two reasons for using broadcast data:

The mobile host does not have to invest on the energy cost for transmitting data requests the broadcast data can be received by a large number of mobile hosts in a single transmission, at no extra cost, thus, ensures effective utilization of the available transmission bandwidth.

Thus, the mobile hosts need to only receive data as and when those data are transmitted, rather than consuming energy by transmitting a request. The mobile host may also have the local non-volatile storage for storing (cache) the broadcast data as and when received, for possible later use. The mobile host may optimize energy costs by determining whether a given query may be processed using only cached data. In case, the cached data is not found to be appropriate for the query, then the mobile host may either wait for the data to be broadcast, or transmit a request for data. However, in order to make this decision, the mobile host must know when the relevant data will be broadcast. The broadcasting of data may be made according to a fixed schedule or a changeable schedule. If the schedule of data transmission is fixed then the mobile host uses the known fixed schedule to determine when the relevant data will be transmitted. In the data transmission, schedule is changeable then even the broadcast schedule may itself be broadcast at a well-known frequency and time intervals. In effect, the broadcast medium can be thought of as a disk with a high latency Requests for data can be thought of as being serviced when the requested data are broadcast. The transmission schedules behave like indices on the disk. This area is still

evolving and research is still being conducted on broadcast data issues.

VII. SOFTWARE MODELS

To deal with the characteristics of mobile computing, especially with wireless connectivity and small devices, various extensions of the client/server model have been proposed. Such extensions advocate the use of proxies or middleware components. Proxies of the mobile host residing at the fixed network, called server-side proxies, perform various optimizations to alleviate the effects of wireless connectivity such as message compression and re-ordering. Server-side proxies may also perform computations in lieu of their mobile client. Proxies at the mobile client undertake the part of the client protocol that relates to mobile computing thus providing transparent adaptation to mobility. They also support client caching and communication optimizations for the messages sent from the client to the fixed server. Finally, mobile agents have been used with client/server models and their extensions. Such agents are initiated at the mobile host, launched at the fixed network to perform a specified task, and return to the mobile host with the results. The emergence of powerful portable computers, along with advances in wireless communication technologies, has made mobile computing a reality. Among the applications that are finding their way to the market of mobile computing—those that involve data management—hold a prominent position. In the past few years, there has been a tremendous surge of research in the area of data management in mobile computing. This research has produced interesting results in areas such as data dissemination over limited bandwidth channels, location-dependent querying of data, and advanced interfaces for mobile computers. This paper is an effort to survey these techniques and to classify this research in a few broad areas [D. Barbara (1999)]. Another concern in terms of software architectures is adaptability. The mobile environment is a dynamically changing one. Connectivity conditions vary from total disconnections to full connectivity. The resources available to mobile computers are not static either, for instance a "docked" mobile computer may have access to a larger display or memory. Furthermore, the location of mobile elements changes and so does the network configuration and the center of computational activity. Thus, a mobile system is presented with resources of varying number and quality. Consequently, a desired property of software systems for mobile computing is their ability to adapt to the constantly changing environmental conditions.

VIII. CONCLUSIONS

Mobile distributed database are being gradually more used in various business, supported by availability of various distributed database management system software's products. This is a big issue that how the data is mange using mobile distributed management system by minimizing the cost and performance of the business. The Mobility brings in a novel dimension to the existing solutions to the problems in distributed databases. We have surveyed a number of the problems and existing solutions. We have identified upcoming research areas that, due to the nature and constraints of mobile computing environment, need rethinking. The upcoming research directions discussed here will be the centres of attractions among the mobile database researchers in future. The key research issues that the research community has tried to tackle are how to cope with mobility, frequent disconnection and energy limitation of the client in a number of DBMS issues, such as transaction management, data caching, data replication and location-aware query processing [P. Serrano-Alvarado, (2004)]. Additional work is also found in the areas of security and privacy [G. Bernard, (2004)], especially for detecting malicious transactions and preserving privacy in location-aware queries. Performance in distributed database is heavily dependent on allocation of data among the sites of the database. The static allocation provides only limited response to workload changes. This situation is even worse when the mobile wireless computers are included in replication schema. Some new methods with their practical approach must be needed for successful mobile computing system.

REFERENCES

[1] DHAWAN, ABHISHEK. "Blue Eyes Technology

[2] Binyamin, M. "Blue Eyes Technology." (2011).

[3] Ekman, P. and Rosenberg, E. (Eds.) (1997). What the Face Reveals: Basic and Applied Studies of Spontaneous Expression Using the Facial Action Coding System (FACS)

[4] Applied Artificial Intelligence

[5] Picard, R. (1997). Affective Computing. MIT Press: Cambridge

[6] G. Bernard, (2004),especially for detecting malicious transactions.

◈◈◈