

Assignment 10 – Week 14

This assignment is based on lecture 12 (chapter 24 – Distributed Databases)

- Submit your *own work* on time. No credit will be given if the assignment is submitted after the due date.
 - Note that the completed assignment should be submitted in .doc, .docx, .rtf or .pdf format only.
 - In MCQs, if you think that your answer needs more explanation to get credit then please write it down.
 - You are encouraged to discuss these questions in the Sakai forum.
-

- (1) Compare and contrast a DDBMS with distributed processing. Under what circumstances would you choose a DDBMS over distributed processing?

ANS:

A distributed database management system is a system in which data is physically distributed over several sites in a network. Even if other users are accessing the data over the network, we do not consider this to be a distributed DMBS, but simply a distributed processing environment. Consider a company with different divisions all over, and each division manages its own data, so they each have their own databases. It would be more efficient to do this than to have all the divisions access one centralized database and update/change their data.

- (2) Compare and contrast a DDBMS with a parallel DBMS. Under what circumstances would you choose a DDBMS over a parallel DBMS?

ANS:

Parallel DBMS is basically a DBMS running across multiple processors and disks that is designed to execute operations in parallel, whenever possible, to improve performance while there is no parallel execution of database (using multiple processors) in DDBMS. Parallel DBMS is generally used for very large records of databases possibly of the order of terabytes, or database systems that must process thousands of transactions per second.

- (3) Discuss the advantages and disadvantages of a DDBMS.

ANS:

Advantages of DBMS:

- **Reflects organizational structure:** Many organizations & teams are naturally distributed over several locations. For example, a Walmart has many offices in different cities. It is natural for databases used in such an application to be distributed over these locations. A Walmart may keep a database at each branch office containing details such things as the staff that work at that location, the account, product & transaction information of customers & products etc.

- **Improved share ability and local autonomy:** Basically, data can be placed at the database site close to the users who normally use that data. In this way, users can consequently establish and enforce local policies regarding the use of this data.
- **Improved availability:** In a centralized DBMS, a computer failure normally terminates the applications of the DBMS while operating but in case of ddbms failure of at one site or a failure of any communication link making some sites inaccessible, does not make the entire system in inaccessible to operate.
- **Improved reliability**
Usually data is replicated to more than one node and failure of a node or any communication link between or any node does not necessarily make the data inaccessible.

Disadvantages:

- **Complexity:** DDBMS basically hides the distributed nature from the user and provides more acceptable level of performance and reliability is more complex than the centralized DBMS.
- **Cost:** Procurement and maintenance costs for the DDBMS is higher than those for centralized DBMS.
- **Security:** In DDBMS we have to controlled and provide the security to different layer, locations and nodes while accessing data, so it is more complex to secure the distributed dbms than centralize dbms.
- **Lack of experience:** Distributed DBMS have not been mostly accepted so it lacks the experience while designing & developing DDMS system in terms of centralized dbms.
- **Database design more complex:** The design of distributed database is more complex than the centralized dbms because in ddbms we must take the account of fragmentation of data, allocation of fragmentation to specific sites and data replication.

(4) What is the difference between a homogeneous and heterogeneous DDBMS? Under what circumstances would such systems generally arise?

ANS:

Homogeneous System: In this type of system, all sites use the same DBMS product. This type of systems is much easier to design and manage. This type of approach provides incremental growth, making the addition of a new site to the DDBMS easy and allows increased performance by the parallel processing of the multiple sites.

Heterogenous System: This type of system usually results when individual sites have implemented their own database and integration is considered at a later stage. In a heterogeneous system, translations are required to allow communication between different DBMSs.

(5) What functionality do you expect in a distributed DBMS?

ANS:

Functionality of Distributed Database System:

- **Replicated Data Management:** The basic function of DDBMS is basically to decide which copy of a replicated data item to access and to maintain the consistency of copies of replicated data items.
- **Distributed Database Recovery:** The ability to recover from the individual site crashes and from new types of failures such as failure of communication links.
- **Keeping track of data:** The basic function of DDBMS is to keep track of the data distribution, fragmentation, and replication by expanding the DDBMS catalog.
- **Distributed Query Processing:** The basic function of DDBMS is basically its ability to access remote sites and to transmits queries and data among the various sites via a communication network.
- **Security:** The basic function of DDBMS is to execute Distributed Transaction with proper management of the security of the data and the authorization/access privilege of users.
- **Distributed Directory Management:** A directory basically contains information about data in the database. The directory may be global for the entire DDB, or local for each site. The placement and distribution of the directory may have design and policy issues.
- **Distributed Transaction Management:** The basic function of DDBMS is its ability to devise execution strategies for queries and transaction that access data from more than one site and to synchronize the access to distributed data and basically to maintain the integrity of the complete database

(6) One problem area with DDBMSs is that of distributed database design. Discuss the issues that have to be addressed with distributed database design. Discuss how these issues apply to the global system catalog.

ANS:

The main thing in DDBMSs is that how to place the database and application that run on DDBMS across the sites. Basically, two mostly used approaches are partitioned and replication. In partitioned scheme database is basically divided into several disjoint partitions and each of which is placed at a different site. In the case of replicated design, this can be fully or partially replicated. Mostly involves mathematical programming to minimize combined cost of storing the database, processing transactions against it, and communication

(7) What are the strategic objectives for the definition and allocation of fragments?

ANS:

The following are the strategic objectives for the definition and allocation of fragments:

- **Locality of reference:** Data should be stored close to where it is used based on the user's location
- **Improved reliability and availability** are improved by replication. There is another copy of the fragment available at another site in the event of one site failing.
- **Storage capacity and cost considerations** should be maintained based on the balance of availability and cost of storage at each site so that cheap mass storage can be used where possible.
- **Balanced storage capacities and costs** Consideration should be given to the availability and cost of storage at each site so that cheap mass storage can be used, where possible.

(8) Describe alternative schemes for fragmenting a global relation. State how you would check for correctness to ensure that the database does not undergo semantic change during fragmentation.

ANS:

Basically, there are two types of fragmentation horizontal and vertical. Horizontal fragments are subsets of tuples and vertical fragments are subsets of attributes. We can also find the mixed and derived partitions as well

- **Horizontal Fragmentation:** It is a subset of tuples. It is done using the condition to filter the data during the selection.
- **Vertical fragmentation:** This is defined using the *Projection* operation of the relational algebra.
- **Mixed fragmentation:** It is mixed of both horizontal and vertical fragmentation
- **Derived horizontal fragmentation:** This fragmentation achieves the relation of a join of two or more relation at the same location.
- **Correctness of fragmentation:** This type of fragmentation can be checked by ensuring the completeness, reconstruction and disjoint of the data.

(9) What layers of transparency should be provided with a DDBMS? Give examples to illustrate your answer. Justify your answer.

ANS:

Four layers of transparency should be provided with a DBMS.

- a. Distribution Transparency: Users should perceive as a single database and should not realize about the fragmentation even if there are various distributed databases.
- b. Transaction Transparency: Every transaction in DDBMS must ensure that the database integrity of the distributed databases involve there.
- c. Performance Transparency: The performance of the DDBMS in case of transaction should be as smooth and effective as centralized system without any lag or performance degradation due to overhead communications across the distributed databases in a network.
- d. DBMS transparency: It hides the fact that different databases are being used locally in heterogenous DDBMS.

(10) A DDBMS must ensure that no two sites create a database object with the same name. One solution to this problem is to create a central name server. What are the disadvantages with this approach? Propose an alternative approach that overcomes these disadvantages.

ANS:

Following are the disadvantages with central name server approach

- Performance problems occurs if the central site become a bottleneck.
- Loss of some local autonomy.
- If the central site fails, the remaining sites cannot create any new database.

An alternative solution is to prefix an object with the identifier of the site that created it.