Database Management System – cs422 DE

Assignment 10 – Week 14

---

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

ANS:

Both DDBMS and distributed processing are for managing data across the network but with different purposes. DDBMS is a database management system where data are physically distributed over several sites, and it makes distribution transparent to users. Distributed processing is breaking down the large computational tasks into smaller sub-tasks to speed up the process. If we have a system for every branch and each branch will process their own data, DDBMS will be more efficient than accessing to one centralized database through distributed processing.

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

ANS:

DDBMS is a database management system where data are physically distributed over several sites across the networks, and it makes distribution transparent to users. Parallel DBMS is also a database management system running across multiple processors in parallel to improve performance while there is no parallel execution over multiple processors in DDBMS. Parallel DBMS will be better to process very large amount of data efficiently.

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

ANS:

Advantages of DDBMS

- Reflects organizational structure – If an organization is naturally distributed over several locations, it is natural for database used in the application to be distributed over those locations. It may keep its database at each branch office containing details related to that branch. The staff at the specific branch can enquire data locally and the company headquarter can enquire data globally, accessing data at all or a few branches.
- Improved shareability and local autonomy – As data are physically distributed, users at one site can access data stored at other sites. A global DBM is responsible for the entire system. With DDBMS, the local DBM can manage the local DBMS.
- Improved availability – The system can reroute the failed note's requests to another site if a single node fails while a computer failure terminates the operations of the DBMS in a centralized DBMS.

- Improved reliability – Because data may be replicated at more than one site, a failure of a not does not make the data inaccessible.
- Improved performance – Each site handles only a part of the entire database; speed of database access may be better than that achievable from a remote centralized database.
- Economics – It is much more economical to partition the application and performe the processing locally at each site.
- Modular growth – It is much easier to handle expansion with DDBM, adding a new sited to the network without affecting the operations of other sites.
- Integration – Distributed data processing can make organizations to be able to integrate software components from different vendors to meet their specific requirements.
- Remaining competitive – Many enterprises have had to reorganize their businesses and use distributed database technology to remain competitive.

Disadvantages of DDBMS

- Complexity – DDBMS can me more complex by hiding the distributed nature from the users and providing the acceptable level of performance, reliability, and availability and by replicating data.
- Cost - Increased complexity can make more maintenances costs.
- Security – In DDBMS, user can access data both locally and globally across different sites over network whose itself needs to be secured.
- Integrity control more difficult – In DDBMS, the communication and processing costs that are required to enforce integrity constraints may be prohibitive.
- Lack of standards – Although DDBM depends on effective communication, we now see only standard communications. Because of that lack of standards has significantly limited the potentials of DDBMS.
- Lack of experience – General-purpose DDBM has not been widely accepted so that we do not have the same level of experience as centralized DBMS yet.
- Database design more complex - Because DDBMS split into a number of fragments and each fragment is stored on one or more computers across the network, design is more complex.

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

ANS:

In a homogeneous DDBMS, all sites use the same DBMS product. Homogeneous systems are much easier to design and mange, provide incremental growth, and allow increased performance by exploiting the parallel processing capability of multiple sites.

In a heterogeneous system, sites may run different DBMS products. Heterogeneous stems generally arise when individuals have implemented their own databases and integration is considered at a later stage, so translations are required to allow communication between different DBMSs.

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

ANS:

We expect DDBMS to have at least the functionality of a DBMS and to have the following functionalities:

- Extended communication services
- Extended data dictionary
- Distributed query processing
- Extended concurrency control
- Extended recovery services

(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:

Distributed database design is complex and has three key issues which are

- fragmentation – dividing the database into smaller pieces
- allocation – assigning the fragments to different sites
- replication – creating and maintaining copies of the fragments for data availability and reliability

Those issues apply to the global system catalog which is very important to manage the distributed database system effectively. It holds information specific to the distributed nature of the system, such as the fragmentation, replication, and allocation schemas. The birth-site of each global relation is recorded in each local global system catalog. Whenever a fragment or replica is moved to a different location, the local catalog at the corresponding relation's birth-site must be updated so that it can locate a fragment or replica of a relation.

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

ANS:

The strategic objectives for the definition and allocation of fragments:

- locality of reference
- improved reliability and availability
- improved performance
- balanced storage capacities and costs
- minimal communication costs

(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:

Alternative schemes for fragmenting a global relation:

- horizontal fragmentation – works with tuples

- vertical fragmentation – works with attributes
- mixed fragmentation – mixes of horizontal and vertical
- derived horizontal fragmentation – based on the horizontal fragmentation of a parent relation

We would check for correctness to ensure that that database does not undergo semantic change during fragmentation by satisfying the correctness rules which are completeness reconstruction and disjointness.

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

ANS:

Layers or transparency which should be provided with a DDBMS, with examples:

- distribution transparency – fragmentation, location, replication, local mapping, and naming transparencies
- transaction transparency – concurrency and failure transparencies
- performance transparency – distributed query processing, load balancing, caching
- DBMS transparency - schema and security transparencies

(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:

The disadvantages of the approach with the central name server:

- loss of some local autonomy
- performance problems if the central site becomes a bottleneck
- low availability: if the central site fails, the remaining sites cannot create any new database objects

An alternative approach that overcomes these disadvantages is to prefix an object with the identifier of the site that created it. e.g., the relation Employee created at site S1 might be named S1.Employee.