Mathokoza Mntambo

ID: UB76859SCO86054

Bachelors in Computer Science

DDT 033: Distributed Database

Atlantic International University

Honolulu, Hawaii

Date: 22nd August 2022

TABLE OF CONTENTS

Introduction.

A distributed database is fundamentally a database that is not restricted to one system, it is spans over different locations, i.e., on several computers or over a grid of computers. A distributed database system is situated on various places that do not share physical components. This may be needed when a specific database requires to be opened by various users worldwide. It requires to be controlled such that for the users it seems like one single database. Distributed databases let local users to access and manage the information in the local databases while offering some sort of global information management which offers global users with a universal view of the data.

The distributed idea permits more availability and reliability. Additional advantage that raised the growth of DDBs is enhanced performance. When a huge database is distributed, it turns out to be a group of smaller ones. Each reduced database, while maintaining the overall structure and characteristics of the other components, executes in its own environment and on its own software, hardware and transaction load. This results in the local databases to have significantly better performance for accesses and queries run on the local database unlike if they were in a huge database. Moreover, transactions that include access to more than one location can take advantage and execute the process streams in parallel, therefore reducing the through-put time of the transaction. The separate outcomes are then combined to give the final response.

Distributed Database.

Distributed databases let local users to manage and access the information in the local databases while offering some sort of universal data management which offers universal users with a universal view of the information. Such universal views let users to combine information from the different sources that may not have been previously integrated, hence offering the possible for new knowledge to be uncovered. The basic local databases may be *homogeneous* and make part of a design which seek out to dispense data processing and storage to attain greater efficiency, or they be *heterogeneous* and make part of a legacy

system whereby the original databases may have been created using different information models.

Applications of Distributed Database include:

- It is utilized in Corporate Management Information System.
- It is utilized in multimedia applications.
- Utilized in Hotel chains, Military's control system etc.
- It is utilized in creating control system.

Advantages of Distributed database.

Management of information with different degree of transparency – In an ideal world, a database must be distributed transparent in the perception of masking the particulars of where each file is really kept in the system. Types of transparencies include network transparency, replication transparency and fragmentation transparency

Increased availability and reliability – Availability is explained as the possibility that the system is uninterruptedly accessible during a time interval while reliability is explained as the possibility that a system is operational at a certain time.

Easier Expansion – In a distributed setting expansion of the system in terms of increasing database sizes, adding more data or adding more processor is far easier.

Improved Performance – Intraquery and interquery parallelism can be achieved by running many queries at different locations by breaking down a query into subqueries that run in parallel which leads to enhanced performance.

Distributed database design.

General aims of the distributed database design include:

- to deliver high performance
- to offer reliability
- to deliver functionality
- to fit into the existing environment
- to deliver cost-saving results

Stages of Distributed Database Design

There are in general numerous design alternatives.

Top-down approach: First the general ideas and the global framework are explained then the details after.

Down-top approach: The detail modules are clarified first then the global framework defined after. Usually heterogeneous and existing databases are combined into a common distributed system.

If the system is created from a scratch, the top-down approach is more welcomed. If the system is to match to already available systems or some modules are so far ready, then the down-top approach is usually utilized. During the requirement analysis stage, the distribution requirements and fragmentation are also considered.

Goals of the Distribution Design and Fragmentation

Local processing

It is necessary to execute as much jobs as possible at the local level. The local processing offers a more effective execution and an easier management.

Reliability and availability of the DDB

It is required to dispense the data over different locations to deliver higher availability and reliability.

Distribution of processing load

It is necessary to dispense the processing power over the different locations to deliver a higher throughput.

Reduction of storage costs

It can be cost effective if not every location is furnished with the same costly high-performance elements. It is sufficient to put in only some of such dedicated locations, the others can utilize it in shared means.

Data and Access Control.

Data access control is a method utilized to control user admittance to data in an organization. It includes leveraging the code of least privilege (POLP), i.e., handling users' access rights

grounded on their responsibilities in the organization, and limiting and defining what information they can access.

Types of data access control

Organizations must select an information access control rule that best meets their needs. There are four kinds of access control systems put aside by how the authorizations are allocated to users.

Mandatory access control (MAC)

This access model uses a central authority to allocate access rights to all user. The administrator categorizes system users and resources based on their access requirements and risk level. The access to data is based on the rights that the user holds. This model delivers a high level of information protection and is utilized by government agencies to protected highly classified data.

Discretionary access control (DAC)

In this model, the information owner chooses who is suitable to access their information. The owner sets rules that regulate who is approved to access the data, making this model extra flexibility and ideal for small to medium-sized businesses.

Role-based access control (RBAC)

This model is the most broadly utilized control mechanism, since it positions with the needs and role of every user in the organization. Anyone trying to access information outside their space is restricted.

Upcoming access control method

The attribute-based access control (ABAC) method is a next generation approval model that delivers dynamic access control. In this mechanism, the resources and users are allocated a set of variables, and admission is reliant on on the value allocated to the variable. The variables vary from geographical location to time of access.

Use of data access control

Access control in information security is vital to make sure that information does not wind up with wrong individuals or leave the business. Many organizations save personal information related to their customers or clients, files containing classified data and far more. It is authoritative that this information is protected, and executing an access control system assists lessen the chance of information leaks.

Query Processing.

This is to change a query in a high-level declaratory language (like SQL) into an efficient and correct execution strategy. Query processing is pulling out data from a huge amount of data without really altering the basic database where the data are stored.

The primary objective of query processing in a distributed setting is to create a high-level query on a distributed database, that is seen as a one database by the operators, into an effective execution approach presented in a low-level language in local databases.

Once a query evaluation plan is chosen, the system assesses the produced low-level query and produces the output. Although the query experiences different processes prior to finally being run, these processes take little time in compared to the time it would actually take to run an un-validated and un-optimized query. The flow of a query processing includes two stages:

Compile time

Parsing and Translation - split the query into tokens then inspect for the accuracy of the query.

Query Optimisation - Assess multiple query implementation plans then choose the best out of them.

Query Generation - Create a low-level database executable code

Runtime time (evaluate/execute the hence created query)

Transaction Management.

Transactions are a set of actions used to execute a logical set of job. A transaction generally means that the information in the database has been altered. One of the main uses of DBMS is to guard the user's information from system failures. It is done by making sure that all the information is reinstated to a reliable form when the computer is started again after a crash. The transaction is any one running of the operator program in a DBMS. Running the same program many times will create many transactions.

The following actions can be executed in a transaction:

Access/Read data (R).

Change/Write data (W).

Commit.

Uses of Transaction Management

The DBMS is utilized to plan the access of information concurrently. This means that the operator can have access to multiple information from the database without being affected with each other. Transactions are utilized to manage concurrency.

It is utilized to satisfy ACID properties.

It is utilized to solve Write/Read Conflict.

It is utilized to implement Serializability, Recoverability and Cascading.

Distributed Concurrency Control.

Concurrency control has to do with the consistency and isolation properties of transactions. The distributed concurrency control method of a distributed DBMS makes sure that the uniformity of the database is kept in a multiuser distributed situation. If transactions do not break any consistency constraints, the easiest method of achieving this goal is to run each transaction unaccompanied, one after another.

Concurrency control is offered in a database to:

- implement isolation between transactions.

- preserve database uniformity through consistency preserving implementation of transactions.
- resolve write-read and read-write conflicts.

Concurrency control techniques include:

Two-phase locking Protocol - an action which safeguards authorization to read or authorization to write a data item.

Time stamp ordering Protocol - a tag that is attached to a transaction or any data, which represents a precise time on which the transaction has been utilized in any way.

Multi version concurrency control – this method keeps old versions of data to raise concurrency

Validation concurrency control - The approach is built on the assumption that the widely held database actions do not conflict.

Data Replication.

Data replication is the operation by which information exist in on a virtual/physical server(s) or cloud instance (principal instance) is endlessly copied or replicated to a cloud instance (standby instance) or secondary server(s). Organizations duplicate data to assist backup, high availability or/and disaster recovery. Data is either asynchronously or synchronously replicated, dependent on the site of the secondary instance. How the data is duplicated influences Recovery Point Objectives (RPO) and Recovery Time Objectives (RTOs).

Benefits of data replication

Even though data replication can be challenging in terms of computational and storage requirements, cost, businesses broadly utilize this database management method to attain the following objectives:

- Enhance the availability of information
- Increase the speed of information access
- Improve server performance
- Achieve disaster recovery

Kinds of data replication

There are many kinds of replication done by businesses nowadays, depending on information replication tools used. Some of the common replication styles are:

- Full table replication
- Transactional replication
- Snapshot replication
- Replication
- Key-based incremental replication

Data Stream Management.

This is a computer software system to control nonstop data streams. It is almost the same as a database management system (DBMS), that is, though, developed for still information in conventional databases. DSMS runs a nonstop query that is not only executed once, but is eternally installed. Hence, the query is continuously run until it is clearly uninstalled. Because most DSMS are information-driven, a nonstop query generates new outcomes as long as new information come to the system.

Stream Data Models

Data stream is an add-only system of time stamped things that come in some order.

Cloud Data Management

The latest tendency in distributed computing is cloud computing and has been the topic of much hype. The vision incorporates on demand, dependable services offered via the Internet with simple access to virtually limitless computing, networking and storage resources. What makes cloud computing exclusive is its capability to offer various levels of functionality like platform, infrastructure and application as services which can be united to best fit the clients' requirements.

Conclusion.

Distributed databases fundamentally offer us the benefits of distributed computing to the database management sphere. In the present scenario of the fast-changing world, distribution of information is a necessity. Distribution of information has its own disadvantages and advantages. A database is a more effective method to organize and store data than worksheets, it permits for a centralized capability that can simply be changed and quickly shared amongst many users.

Bibliography

https://people.eecs.berkeley.edu/~brewer/cs262/concurrency-distributed-databases.pdf
https://phoenixnap.com/blog/software-development-life-cycle. Goran Jevtic. Date: 15 May 2019.
https://us.sios.com/what-we-do/data-replication/
https://www.geeksforgeeks.org/distributed-database-system. Date: 17 February 2022
https://www.manageengine.com/data-security/what-is/data-access-control.html. Date: 14 August 2018
https://www.manageengine.com/device-control/data-replication.html
https://www.sciencedirect.com/topics/computer-science/distributed-databases