Distributed Databases and Concurrent Programming

A distributed database is where the storage mediums of which the database is stored are not attached to the same processor indicating they are part of different sets of hardware. This means that the database can be all across the world but have to work together through organized network servers, decentralized independent computers on the Internet or on corporate intranets.

The distribution of the data is spread across the network of database servers at different physical locations. Distributed databases can improve performance for users by allowing transactions to be processed on many machines, instead of being limited to one which decreases the load a single machine and processing time for a single request is faster as less resource allocation is required when dealing with different requests concurrently.

Features -

Distributed databases allow for a larger volume of traffic as previously mentioned but over a much wider audience from all over the world.

With Distributed databases, keeping data up to date is important as data integrity can be lost. There are multiple methods in keeping the network up to date including replication methods and duplication methods.

Replication is specialised software created for the distributed database to identify changes across the network and then proceeds to update all other databases so they are identical.

Duplication is where a master database is identified and all other servers are a copy of this and any new updates are made to the master copy.

Further work with distributed databases can lead to distributed SQL.

Technical Challenges -

In Transactional systems, concurrent programming is used and a level of assurance is required so that database transactions are performed without violating the data integrity of the respective databases from multiple users across multiple databases accessing the same data because otherwise the system will fail.

A database transaction is a unit of work normally encapsulating a number of operations over a database like reading a database object, writing, acquiring lock and querying.

This control is very important in the implementation for airport and airline websites when booking flights.

In a case example; two people cannot book the same seat as a seat only sits one. This means that the person booking that seat needs some sort of assurance that that seat won’t be double booked. This is where in concurrent distributed databases; one user must acquire lock on certain data of sets of data relating to their seat.

A common distributed database concept is ACID and this is conforming for transactions as mentioned above within that concept. This concept deals with issues and regulation of the database across a network in the aim to provide a flawless interaction with the databases for all users.

The ‘I’ in ACID stands for Isolation and the concept of isolation is what “guarantees that no transaction sees premature operations of other running transactions.” In essence, this prevents conflicts between concurrent transactions which solves the double booking problem seen in the airline website through code implementation conforming to this concept.

The ‘D’ in ACID is for Durability which is for the “persistence of executed transactions.” Once a transaction has been committed the outcome of the transaction such as state changes are kept, even in case of system failures.

In the Netflix case, the durability would allow users to view a video from where they left off even if their computer crashes, the servers go down or they close the browser.

It is also a part of being able to record which users watched what and allow as many as possible to watch the same show as possible (avoiding locking the file for a single user) using the servers they have.

Legal, Social and Environmental Implications –

Implications of the distributed database service are the increased energy consumption, site hire and maintenance staff required to run the services unless the servers used are hired. This process can cost a company a lot more money as not only will the tax on energy consumption and wages be applied but environmental implications may require the company to take action on how their systems operate to be more efficient causing less (CO2 or Heat) pollution to the environment. Hired servers are a common way to reduce costs while still maintaining the service required.

For instance, Netflix hires servers from amazon to host the digital movie streaming service for across the globe and use a distributed database of movies and TV shows for customers to use. The distributed servers amazon provide Netflix allow for millions of users to be able to watch shows and many of them streaming the same movies.

Legal issues with hired servers might include the unknown extent to which the hired servers’ security covers the web service being hosted and when the company is required to conform to the data protection act or prevent computer misuse, – i.e. against attacks or malware. Data must be held securely and must be relevant to the operations of the systems. In some cases, the level of security provided by a hired server may invoke additional cost to the original or standard contract and for companies not willing to pay this cost and compromising security are in breach of Data Protection and Computer Misuse Act. In 2015, it was found that the names, addresses and Social Security numbers of students and recent graduates at seven Indiana University campuses who attended from 2011 to 2014 were stored for nearly a year on a sub-contracted site unprotected by security authentication to which the university had found three automated computer data mining applications had accessed and downloaded the data files.

Overall, on a distributed database service for an airline, data integrity and assurance of service for a customer can be guaranteed and serviced quickly by following guidelines such as ACID.

References;

<https://en.wikipedia.org/wiki/Distributed_database>

<https://en.wikipedia.org/wiki/Concurrency_control>

<https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/transactions-and-concurrency>

<http://berb.github.io/diploma-thesis/original/061_challenge.html>