Report on Perspectives on the CAP Theorem

This paper starts discussing by highlighting the fundamental trade-off between partition tolerance, consistency, and availability. The CAP Theorem addresses these issues alongside with the CAP theorem practical implication. Generally, the safety and liveness cannot be guaranteed in an unreliable distributed system, as safety, consistency, liveliness, and unreliable are some of the factors. It is impossible to achieve both consistency and availability simultaneously therefore to do so, one must compromise these two and therefore a trade-off. The CAP theorem in the context of a web service, consistency usually means that each server returns the right response to each request, hence weak consistency services are useful services while avoid sacrificing availability. It supports different partitioning such as Data partitioning, operation partitioning, functional partitioning, user partitioning and hierarchical partitioning. The simple services follow atomicity as it per forms simple services. The complicated services on the cannot be specified by sequential specification but are highly focused in complicated services. The availability in web services usually means that for every request should eventually have a response and for the partition tolerance, since servers are not reliable, communications are treated as faulty. Therefore, the CAP Theorem summarizes the states that in a network subjected to communication failures, it is impossible for any web services to implement an atomic read/write shared memory that guarantees a response to every requests. In practicality, the best effort availability, the network, and the services are optimized to provide the best effort availability, example the use of Chubby servers are proved to consistent. It should also be responsive in all the situations. One of the applications that uses this concept is web cache where the consistency is almost guaranteed. There may be small trade-offs such as content could be out of date, but this can be balanced by setting the desired level of consistency that can be updated dynamically. The CAP Theorem helps in redesigning the system and segmenting it into components to provide different levels of guarantees. In near future, the CAP Theorem will provide fundamental base to design systems which are safe and liveness of the systems including key factors such as trade-off between scalability and consistency, tolerating attacks such denial-of-service attacks and compactible for Mobile wireless networks prioritizing geography and proximity as critical, social interactions as primary and privacy as top priority.

Report on Column-Stores vs. Row-Stores: How Different Are They Really?

This paper discussing the column-stores database systems have consistently performed much better than the traditional row-based database systems. The authors also highlighted that this may lead to pre-conceived notion that one can obtain the benefits of column-store using row-store, but this assumption is false. The paper discusses these by demonstrating the impact on performance of a variety of column-oriented query execution techniques such as vectorized query processing, compression, and a new join algorithm. As disk space is cheap and getting cheaper drastically which helps in reducing required number of disks indirectly reduce power consumptions which is a major cost factor that is becoming increasingly important. There is a fundamental difference between a from-scratch column-store and a row-store using column-oriented physical design without exploring alternate physical designs for the row-store system. The column-store DBMS should be optimized for late materialization, block iteration, column-specific compression techniques, use of new optimization called invisible joins and the measure of different variants of C-store database by removing column-specific one by one. The compression can offer the most benefits and less substantial in other cases, but the late materialization offers performance about a factor of 3 across the board. The other performance optimizations include Block Iteration and Invisible join that gives about a factor of 1.5 times in terms of performance. The authors concluded by showing that not only the column-store in a row-store is impossible rather also emphasis that the simulation of their setup column store performed poorly on today's row-store systems release.

I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/examination. I further pledge that I have not copied any material from a book, article, the Internet or any other source except where I have expressly cited the source.

Signature: Janmejay Mohanty Date: 4th March 2023