**Dynamo: Amazon’s Highly Available Key-value Store**

Amazon is the most prominent online shopping website in the world also issues the problem of reliability of data. During the peak hours Amazon try to maintain the customers by increasing the network servers to fulfill the requirement. The ideology behind the Amazon development platform is that the reliability and scalability of a system is dependent on the application status. Dynamo is highly available and scalable distributed data store built for Amazon platform, this provides primary-key only interface to meet the requirements of these applications. Traditionally, production systems store their state in relational databases, but with the advent of the new technology which is a highly available data storage technology that fulfills the requirements of important classes of services. Dynamo has simple key/value interface which uses the resources efficiently and pertain simple scale out scheme to address growth in data set size.

Dynamo’s storage system has following requirements as in

* Query model (read and write operations)
* ACID (properties that ensures database transactions are reliable or not)
* Efficiency (to constantly achieve the latency and throughput requirements)

Service Level Agreement is a formally negotiated contract where a client and service agree on several system-related characteristics. For systems that are prone to network and server failures the availability can be increased by optimistic replication techniques. The work that stands related to Dynamo can be referred as peer-peer systems, Distributed file systems and Databases, Discussion. The system architecture for the Dynamo consists of System Interface, partitioning algorithm, Replication, Data versioning, Execution of get() and put() operations and Handling failures. Dynamo has provided with desired levels of availability and performance has been successful in handling server failures.

**Data management in cloud environments: NoSQL and NewSQL data stores**

In the recent advent of the revolutionary digitalization there is immense flow and generation of data from numerous systems and devices, the technology of loud computing uses the service as pay-per-use model where the consumer pays only for the resources utilized.  Cloud environments pose new requirement to data management as in Scalability and high performance, Elasticity, Fault tolerance, security and privacy features. Storing of huge amount and processing of massive data, a common employed strategy is to partition the data and store across different nodes. The CAP theorem is comprised of consistency, availability and tolerance. The term NoSql stands for open-source, distributed and non-relational databases, which also means "NOT only SQL". The main concurrency-control schemes can be categorized as pessimistic or optimistic. Pessimistic concurrency control, or pessimistic locking, assumes that two or more concurrent users will try to update the same record or object at the same time. To prevent this situation, a lock is placed onto the accessed entity so that exclusive access is guaranteed to a single operation

The two most common horizontal-partitioning strategies are as follows:

* Range partitioning
* Consistent hashing

Range partitioning assigns data to partitions residing in different servers based on ranges of a partition key. Partitioning graph databases is significantly more challenging than partitioning other NoSQL stores. In master–slave replication node is designated as a master and is the only node that processes write requests. Changes are propagated from the master to the slave nodes. Basically, two consistency models can be distinguished: strong and eventual consistency. Strong or immediate consistency ensures that when write requests are confirmed, the same (updated) data are visible to all subsequent read requests.