Dynamo: Amazon's Highly Available Key-value Store

Kartikeya Upasani (kuu2101)

Paper Review, 8 October 2016

1 Motivation

Today's internet scale applications cater to a large number of users. Failure even for a short amount of time can lead to significant financial losses, as well as a bad user experience.

2 Goal

Developing a storage that can manage large volumes of structured data with high availability for large number of clients attempting concurrent access.

3 Key Idea

Sacrificing the relational schema for a simple key-value store that uses only a single primary key. Trading data consistency for data availability.

4 Approach

Dynamo is designed specially for internal use by Amazon's services that need only a non-relational key-value storage. Conflict resolution during concurrency is pushed to reads instead of writes. It can either take place at the application end by the developer, or at the data storage end by a chosen policy. Nodes store different key ranges in a distributed fashion, and a ring arrangement of nodes is utilized where each node stores a particular range of keys. A node has enough information to route requests to the node containing the data in just one hop. Since there is redundancy of data, multiple nodes may store overlapping key ranges. Sloppy quorum may put a key on a node that is not supposed to hold the key if a node that is supposed to hold the key is down, thereby increasing write availability further. The key partition algorithm is incrementally scalable.

5 Results

The biggest downfall of the database is it's consistency during reads, however, according to the analysis done, divergent versions during reads are avoided 99.94% of the times. The implementation is also able to meet the required 99.9^{th} percentile latency for reads and writes.

6 Conclusion

Dynamo is an excellent choice for write-oriented applications such as managing the shopping cart. It is being deployed at production at Amazon.

7 Comments

Dynamo only supports single key updates, does not guarantee isolation of transactions, and does not allow for structure of data beyond key-value pairs. However, these compromises are necessary to achieve 99.9% reliable latency, which is a very stringent requirement.