Presentation Report

- Gaurav Vaidya

Overview:

This is summarized report of the study of two papers describing two distributed data storage systems:- **PNUTS: Yahoo!'s Hosted Data Serving Platform** and **Dynamo: Amazon's Highly Available Key-value Store**. Both PNUTS and Dynamo are distributed data storage systems and are used as scalable back-ends for the various online applications and services of Yahoo and Amazon respectively. This study reveals that both of these systems have some common features and design considerations and are yet distinct with regards to their architectures and implementations.

The authors of the first paper (PNUTS) begin with the explanation of the need to develop a scalable, highly responsive and reliable backend for web applications with a high amount of concurrent users such as social networking applications. They state that the basis of their design of a distributed data store is relaxation of the consistency model offered by traditional database management systems. Instead, importance is given to the availability of data that is scattered across distant geographical locations. They describe the system architecture with a brief overview of a single geographically separated region and its various components. They also explain how their partitioning scheme divides database tables into smaller tablets that are replicated across many regions. The authors then proceed to describe their timeline based consistency model and record level mastering scheme to explain how versioning is implemented at the record level granularity. This model ensures that no application works on a stale replica of the data. This is further explained by studying some PNUTS API calls. They also briefly discuss some failure and recovery mechanisms. Although the authors do not discuss the system in all its intricate detail, this paper does give the reader a fair idea about the working of PNUTS.

The second paper Dynamo describes a distributed hash table used as a data store for Amazon's web services and applications. Dynamo is a key-value store and its design is also based on relaxation of consistency constraints. However, the eventual consistency guaranteed by Dynamo is different from the timeline consistency that PNUTS speaks of. Dynamo focuses on being constantly available for write operations. All replicas eventually become consistent through reconciliation measures taken at the time of read operations. The authors of Dynamo explain the concepts of Dynamo rings which consist of various nodes and are used to replicate data. Nodes in a Dynamo ring are homogenous in their functional responsibilities. Dynamo is also incrementally scalable. Dynamo also uses version control based on vector clocks which are used to determine the latest version of data. The most significant and distinct feature of Dynamo is its configurability. The authors also describe some failure detection and recovery techniques used by Dynamo such as Hinted Handoff.

Both papers conclude by showing results of experimental evaluations of their respective systems. Evaluation is done based on measures of efficiency and latency of operations in the presence of varying numbers of requests and nodes. Each paper also briefly compares its respective system with other distributed data storage solutions.

Comments:

The presentations were concluded with a comparison between PNUTS and Dynamo. It was noteworthy that PNUTS is a hosted service and work on both hashed and ordered tables, while Dynamo is strictly a distributed hash table and is internal to Amazon. Also, there are significant differences between the inter-node communication protocols and consistency models.

Questions asked during the first presentation (PNUTS):

1. In PNUTS would there be a bottle-neck at every region because every region has only one replica?

   No. A region serves requests originating proximally to that region and therefore all requests directed to a particular record are divided among the various regions that have its replica

2. Isn't the moving of tablets between various storage units inefficient? Is there a specific policy that is observed?

   The Yahoo Message Broker provides a high speed and reliable communication backend for PNUTS. Tablets are only moved asynchronously when they are split or a node is performing recovery, therefore it is a rare event. The authors do not speak of any specific policy here.

3. Can the timeline consistency fail under any circumstances?

   The timeline consistency model always ensures that no application ever reads a stale version of the data because each record knows the location of its master copy and its own version number.

Questions asked during the second presentation (Dynamo):

1. Can this system be used effectively as a distributed data base management system?

   If availability is the key concern then yes it can be used. However it does not enforce referential integrity constraints and has a relaxed eventual consistency model which may at times result in operations being performed on stale versions of data. Therefore it is subjective to usage.