Diskless Data Analytics on Distributed Co-ordination Systems

Dayal Dilli[1] / Kenneth B. Kent[1] / Robin Bate Boerop[2]

Faculty Of Computer Science, University of New Brunswick[1] / UserEvents Inc[2]

ddilli@unb.ca / ken@unb.ca / robin.bateboerop@userevents.com

Outline

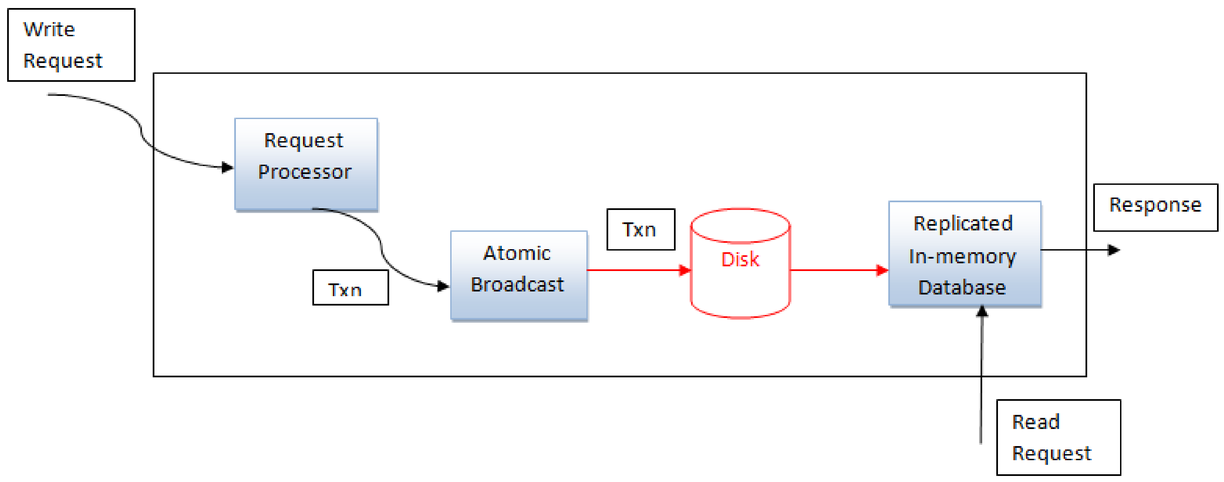
- Benchmark the performance of widely used distributed co-ordination systems.
- Compare and analyze the benchmark results to identify the scope of diskless operation improving performance. Also, identify the most efficient distributed co-ordination system to implement the research.
- Implement a diskless solution to the chosen efficient distributed co-ordination system.
- Analyze the durability, availability, consistency and performance guarantee of the resulting system.

Motivation

The challenge in the area of distributed applications is to perform the analysis quickly while preserving the consistency, availability and durability of its data. The focus of this research is to review the performance of various distributed co-ordination systems and analyze them to arrive at a diskless solution that can offer the best case performance.

Background

Apache Zookeeper, a file-system-like store, is a highly available, distributed service. It provides facilities for building distributed synchronization, group services, message queue services, naming and configuration management. Zookeeper utilizes the Zookeeper Atomic Broadcast (ZAB) protocol to provide distributed co-ordination. It uses fuzzy snapshot with transaction logging to provide persistence. This is one of the most efficient and widely used distributed co-ordination systems identified in our study. We use Zookeeper as the primary focus of our research.



Problem

Benchmarking the fundamental operations in Zookeeper, we found the write performance of Zookeeper to be very low. The reason behind this is the transaction logging scheme operating behind every write operation on Zookeeper. Since the transaction logs are written to disk, it introduces a disk overhead to operations. Many use cases of Apache Zookeeper like message queues, distributed barriers and locks have write intensive workloads. These applications using Zookeeper suffer from disk-based performance bottlenecks.

Solution

The transaction logging on to the disk is the major bottleneck in Zookeeper. By removing the dependency of the hard drive to log the data, the performance will be increased. Since data is in memory, it expires at some point causing durability problems. We take advantage of ZAB replication along with fuzzy snapshot to restore the data of Zookeeper during failures without affecting the performance.

Results

The performance evaluation after implementing the diskless scenario, results in approximately a 30x write speed up. We also generate optimal Performance Vs Durability trade-off levels for various use-cases of distributed applications.