Cloud Storage Systems

Project - Phase 10 Essay

Tiago Carvalho `fc51034`

Diogo Lopes `fc51058`

Miguel Saldanha `fc51072`

João Roque `fc51080`

João Afonso `fc51111`

Group 14

25/05/2021

Abstract

TODO abstract

Chapter 1

A brief survey on replica consistency in cloud environments

TODO TODO TODO TODO TODO TODO TODO TODO TODO
TODO TODO TODO TODO TODO TODO TODO TODO TODO
TODO TODO TODO TODO TODO TODO TODO TODO TODO
TODO TODO TODO TODO TODO TODO TODO TODO TODO
TODO TODO TODO TODO TODO TODO TODO TODO TODO
TODO TODO TODO TODO TODO TODO TODO TODO TODO
TODO TODO TODO TODO TODO TODO TODO TODO TODO
TODO TODO TODO TODO TODO TODO TODO TODO TODO
TODO TODO TODO TODO TODO TODO TODO TODO TODO
TODO TODO TODO TODO TODO TODO TODO TODO TODO
TODO TODO TODO TODO TODO TODO TODO TODO TODO
TODO TODO TODO TODO TODO TODO TODO TODO TODO
TODO TODO TODO TODO TODO TODO TODO TODO TODO
TODO TODO TODO TODO TODO TODO TODO TODO TODO
TODO TODO TODO TODO TODO TODO TODO TODO TODO

TODO TODO

TODO TODO

Chapter 2

Autoscaling tiered cloud storage in Anna

2.1 Motivation

A wide variety of cloud-storage systems is available today, and developers can select the one that better suits their application's needs, making cost-performance trade-offs. However, these systems are not very dynamic, which is not ideal when most applications deal with a non-uniform distribution of performance requirements.

2.2 Overview of Anna

Anna is an autoscaling, multi-tier, coordination-free, distributed key-value store service for the cloud that allows system operators to specify service-level objectives (SLOs), like fault tolerance or cost-performance. It is built on AWS components.

The performance of a system depends on the volume of a workload

and on whether workloads make a lot of requests to a small subset of the keys or more uniform requests. Anna uses three mechanisms to adapt to these dynamics of workloads:

- **Horizontal elasticity** – Storage tiers are able to scale elastically, increasing/decreasing storage capacity and compute and networking capabilities, depending on the volume of a workload, by adding/removing nodes and repartitioning data among them.
- **Multi-Master Selective Replication** – Hot sets (frequently accessed) are replicated onto many machines, with hot keys being more replicated than cold ones.
- **Vertical Tiering** – Anna is able to promote hot data to the fast, memory-speed tier and demote cold data to cold storage.

2.3 Anna Architecture

TODO INSERIR IMAGENS

Anna has two storage tiers: one that is fast but expensive, providing RAM cost-performance, and another that is slow but cheap, providing flash disk cost-performance.

The **monitoring system** and **policy engine** are responsible for adjusting the system to the workloads' dynamics and meet the SLOs. The **cluster management system** modifies resource allocation based on the decision of the policy engine (with Kubernetes). **Routing service** is a client-facing API that abstracts the internal dynamics of the system.

Each **storage kernel** contains multiple threads that interact with a thread-local storage medium (memory-buffer or disk volume, depending on the tier) and process requests from clients.

Shared-memory coordination and consensus algorithms decrease performance and cause latency and availability issues. Therefore, Anna is coordination-free. Periodically, threads multicast (gossip) updates to other threads that maintain replicas of their keys. Conflicts are resolved asynchronously. As a result, Anna exploits multi-core parallelism within a single machine and smoothly scales out across distributed nodes.

For different key replicas, although a set of gossips may be applied in different orders, **Commutativity**, **Associativity**, and **Idempotence** properties ensure that the state of the replicas eventually converges.

Anna requires maintaining certain metadata to efficiently support the mechanisms initially described. Every tier has two **hash rings**. A global hash ring that determines which nodes in a tier are responsible for storing each key and a local hash ring that determines the set of worker threads within a single node that are responsible for a key. Each individual key K has a **replication vector** that has the number of nodes in each tier storing K, and the number of threads per node in each tier storing K. Anna also tracks **monitoring statistics**, such as the access frequency of each key and the storage consumption of each node.

2.4 Policy Engine

Anna supports three kinds of SLOs: an **average request latency** (ms), a **cost budget** (dollars/h), and a **fault tolerance** (number of replicas that are allowed to fail). If the average storage consumption in a certain tier has violated configurable upper/lower thresholds, nodes are added/removed. Then data is promoted or demoted across tiers. Next, if the latency exceeds a certain fraction of the latency SLO and memory tier's compute consumption exceeds a threshold, nodes are

added to the memory tier. However, if not all nodes are occupied, hot keys are replicated in the memory tier. Finally, if the observed latency is a certain fraction below the objective and the compute occupancy is below a threshold, the system checks if it can remove nodes to save cost.

2.5 Anna API

TODO INSERIR IMAGEM

GetAll allows users to observe the most up-to-date state of a key. Put relies on asynchronous gossip for the update to propagate to other replicas. If the node crashes before gossiping, the update will be lost.

2.6 Conclusion

Integrating the mechanisms described, Anna becomes an efficient, autoscaling system representing a new design point for cloud storage. In many cases, Anna is orders of magnitude more cost-effective than popular cloud storage services and prior research systems. Throughput increases linearly with cost, not plateauing, meaning that it can get better performance out of the same cost when compared to the current available cloud storage solutions. Also, the system is able to adapt to dynamic workloads while not violating the SLOs most of the time. Finally, the system is able to recover from node crashes, while not hurting performance too much, since it does not pause, providing high availability to the users.

Chapter 3

Threats and security issues in cloud storage and content delivery networks: Analysis

TODO

Chapter 4

Data auditing in cloud storage using Smart Contract

TODO

Chapter 5

Key challenges and research direction in cloud storage

TODO