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 [?]

With any distributed storage system, such as cloud storage, there are many issues that one needs to be aware of: consistency, fault tolerance and integrity, to name a few. This paper focus especially on the integrity problem. Despite cloud storage systems using the usual mechanisms to avoid data corruption, like RAIDs and ECCs, there are still problems that arise, and usually it is needed to resort to a Third Party Auditor (TPA). This TPA is responsible to check for the integrity of the data to reassure that nothing has been corrupted. While this works, there are privacy, security and confidentiality concerns when using a TPA since they have access to all the data and things can be compromised. To address this problem, the authors propose the usage of smart contracts to perform this data auditing.

4.1 Smart Contracts in Data Auditing

To put it simply, a smart contract is a piece of code that is deployed on a blockchain for the participating nodes to execute in exchange for gas. Since a blockchain is a fully decentralised system, it avoids entirely the chance of a third party tampering with the data, since the blockchain technology on itself is specialised with dealing with this kind of problems.

4.1.1 Proposed System

To implement it, the researchers proposed an architecture which is comprised of three elements: the Cloud Service Provider (CSP), the Data Owner and a blockchain. The blockchain used was Ethereum-based since it is the more popular blockchain to support smart contracts. In this architecture, the user sends the data encrypted to the CSP, hashing it beforehand. This hash is then used in the creation of the smart contract, by the user. The smart contract includes information about the data such as the name of the files, hashes, size and details of the data owner, along with the data itself. This smart contract is then deployed on the blockchain to perform the data auditing periodically. One thing worth noting is that the researchers used an optimised blockchain for data auditing, the DAB (Data Auditing Blockchain), designed for this sole purpose.

4.1.2 Performance Analysis

After the system was designed and implemented, the authors did an evaluation on its performance, with very positive results. They stated that the system provided reliable and confidential auditing, without requiring any effort from the CSP or user after the smart contract is

deployed. The gas costs increased in a linear fashion depending on the number of blocks which is positive.

4.2 Conclusion

In this paper, the researchers presented the problem of data auditing in cloud storage systems and gave a valid solution to it, the usage of smart contracts for data auditing. Its implementation is relatively simple, while having very good results, which is definitely a step in the right direction and could see further developments in the future. While the results achieved by the researchers were good, there are probably optimizations that can be done when developing the system for a real world use.

Chapter 5

Key challenges and research direction in cloud storage [?]

This paper provides a general overview of the issues, challenges and future trends associated with cloud storage and compares the approaches that are being taken by researchers to deal with these problems.

5.1 Issues in cloud storage

5.1.1 Data management

The management of data stored across multiple geo-distributed data centers is associated with certain challenges: scalability, parallelism, recovery, consistency in replication, reliability, and cost.

5.1.2 Data partitioning

Partitioning is meant to improve scalability, optimizes performance, and reduce contention. This process refers to decomposing a table row-wise (horizontally) or column-wise (vertically). Fragmentation is useful for load balancing.

5.1.3 Data replacement

Has an important role in terms of performance in the cloud environment. Some of the issues include: Geographically Distant Data Sharing, Client Mobility, Data Inter-dependencies. One of the proposed solutions consisted of an algorithm that is based on data access patterns and client locations.

5.1.4 Availability through replication

Process of sharing information to provide availability. Consistency between redundant resources is major issue in replication. Data resources can be replicated closer to the application to reduce network latency. Generally, replication is needed to provide fault tolerance and availability for the application. Bandwidth consumption and user's response time are the key factors in providing replication to the application. Systems that provide both reliability and availability are often said to be fault tolerant. The main objective of data replication is to provide availability and to reduce response time and total bandwidth consumption. Advantages of replication: Less energy consumption, network bandwidth, and communication delay, improves reliability, fault tolerance and accessibility. Challenges of replication: Data selection, Consistency in replicated data, sophisticated management, load balancing and recovery.

5.1.5 Data partitioning

Issues related to data security/theft, data unavailability, privacy, integrity.

5.2 Open research challenges and future trends

Multicloud: Prevents downtime, facilitates replication and fragmentation and improves security and costs. Multicloud environments add complexity to application management (greater effort is needed).

DevOps in Cloud: Bridges the gap between developers and operation team. DevOps tools can assist the management and configuration of resources (such as storage, networking, and computing). This approach improves both speed and productivity.

Machine Learning in cloud: Researchers are taking advantage of distributed environments to research parallelization of machine learning and data mining problems.

Big Data in cloud: There are still many challenges to overcome: scalability, availability, data integrity, data transformation, data quality, data heterogeneity, privacy, and legal issues.

IOT in cloud and Fog Computing: Enables new opportunities: complex analysis, data mining and real-time processing. Fog computing is an alternative to cloud computing that is intended for distributed computing where multiple devices (edges) connect to a cloud.