2020

CSCI 49378: Lecture 4: Distributed File Systems

Bonan Liu
CUNY Hunter College

NYC Tech-in-Residence Corps

Distributed File Systems

Bonan Liu
Tech-In-Residence Member, Hunter College, CUNY
Spring 2020

Disclaimer

The content of this presentation is being provided for educational and informational purposes only. The views, thoughts, and opinions expressed in this presentation belong solely to the author, and not necessarily to the author's employer.

The content of this presentation is not endorsed by the author's employer.

Agenda

- Distributed File Systems (Blob Storage)
- Distributed Key-Value Stores
- Distributed Relational Databases

Distributed File Systems

Distributed file systems (DFS) is a distributed data management system which provides an interface to store files and to access them for read/write later.

- GPFS: IBM General Parallel File System
- GFS: Google File System
- S3: Amazon Simple Storage Service
- HDFS: Hadoop Distributed File System

Hardware Comparation[1]

Type	I/O per seconds	USD/GB	Good for Read	Good for Write
Memory	> 10M	21	Good	Good
SSD	35k	3	Good	Medium
SAS	180	0.5	Medium	Medium
SATA	90	0.07	Medium	Medium

[1] Large-Scale Distributed Storage System: Principles and Architectures. Chuanhui Yang.

Portable Operation System Interface (POSIX):

- Open/Close a file
- Read/Write a file
- Open/Close a directory
- Read a directory

Common File Operations in DFS:

- Create/Delete a file
- Read/Write file content
- Set/Read a file attribute

An example: Google File System

- System Elements
  - GFS Master
  - ChunkServer
  - GFS Client
- System Architecture

An example: Google File System

- A life of the write request
  - Leasing
  - LRU: Least Recent Used Cache
- Design of Master
  - Metadata
  - Master Fault Tolerance
  - Chunk Load Balance
  - Garbage Collection

Distributed Key-Value Stores

Distributed Key-Value store is a system which supports read and write of individual structured objects.

- Amazon Dynamo DB
- Google Bigtable
- Alibaba Tair

Most distributed key-value stores rely on distributed file systems as underlying storage system.

Distributed Key-Value Stores

Common Operations:

- Put: Save a key-value pair
- Get: Get a value by a key
- Delete: Delete a key-value pair

Unsupported Operations:

- Cross-Row/Cross-Table transaction
- Secondary index

Distributed Key-Value Stores

Before dive into the architecture of distributed key-value stores, let’s catch up the missed content in lecture 2.

- Sharding
- Consistent Hashing

Sharding

Sharding is a technique to break up the large amount of data or large amount of computation requests into smaller pieces and store or process them separately.

Consistent Hashing

- Map the hash space into a loop (ring)
- Hash the target data/request
- Hash the node
- Always allocate the target data/request to the nearest node

An example: Amazon Dynamo

- Key-Value Distribution
  - Consistent Hashing
- Nodes Management
  - Gossip Protocol
- Data Replication
- NWR
- The life of a write request
- The life of a read request

Distributed Relational Databases

Distributed relational database is a RDBM system which manage data distributed across multiple nodes.

- Oracal
- Microsoft SQL Server
- MySQL
- Google Spanner

Most distributed relational databases rely on distributed file systems as underlying storage system.

Distributed Relational Databases

Common Operations:

- SELECT
- INSERT
- UPDATE
- DELETE

a.k.a. CRUD operations. (Create/Read/Update/Delete)

Distributed Key-Value Stores

Before dive into the architecture of distributed key-value stores, let’s catch up the missed content in lecture 2.

- Two-Phase Commit

Two-Phase Commit

Assumption:

- Each node can stabelly write logs in local storage
- A node serves as the coordinator of the system

Commit-Request Phase:

- Coordinator send queries to each participants
- Participants vote

Commit Phase:

- Participants act per Coordinator’s second request

An example: MySQL Sharding

- System Elements
  - MySQL Client
  - DB Proxy
  - DB Group
  - Metadata Server
- System Architecture
- Major Challenges

Readings

The Google File System. Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung

Dynamo: Amazon's Highly Available Key-value Store. Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall and Werner Vogels