School of Engineering and Applied Science
Ahmedabad University
Cloud Computing (CSC440)
Project Definition

Distributed Secure and Highly Available Cloud Object Storage

Bhavya Patwa, Kaivalya Shah, Karan Patel, Maitrey Mehta, Riddhesh Sanghvi

Introduction:

Cloud storage relies almost exclusively on large storage providers acting as trusted third parties to transfer and store data. This system is vulnerable to a variety of security threats and is also dependent on a single system which if fails could result in unavailability of data.

A peer-to-peer cloud storage network implementing encryption in transit and at rest would allow users to transfer and share data in a highly secure and reliable manner. A decentralized cloud storage network offers many advantages compared to datacenter-based cloud storage. Data security can be maintained using client-side encryption, while data integrity will be maintained via a proof of retrievability. The impact of infrastructure failures and security breaches will be greatly reduced. Furthermore, the currently unused pool of storage within several computers and mobile devices can be utilized.

Services:

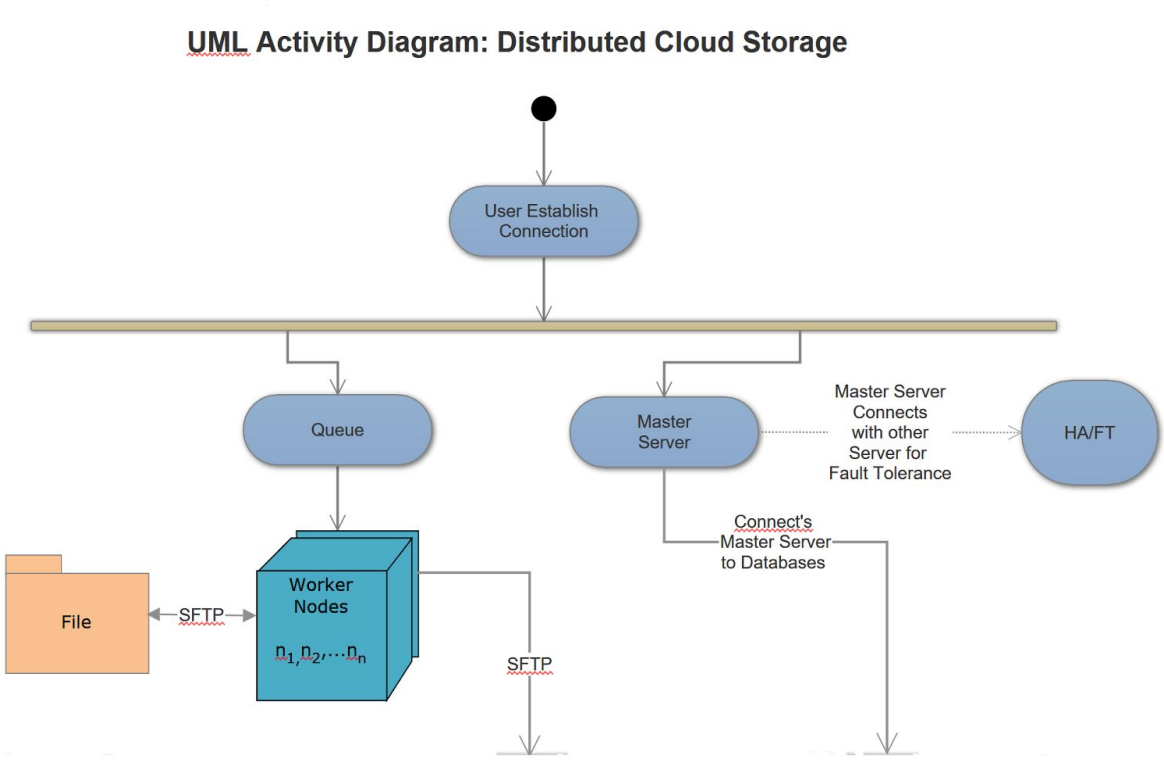
1. Upload Object
    a. Creating Shards
        i. Splitting File into fixed sized chunks
        ii. Adding numerical order metadata
        iii. Encrypting individual chunks
    b. Uploading protocol to upload data to peers

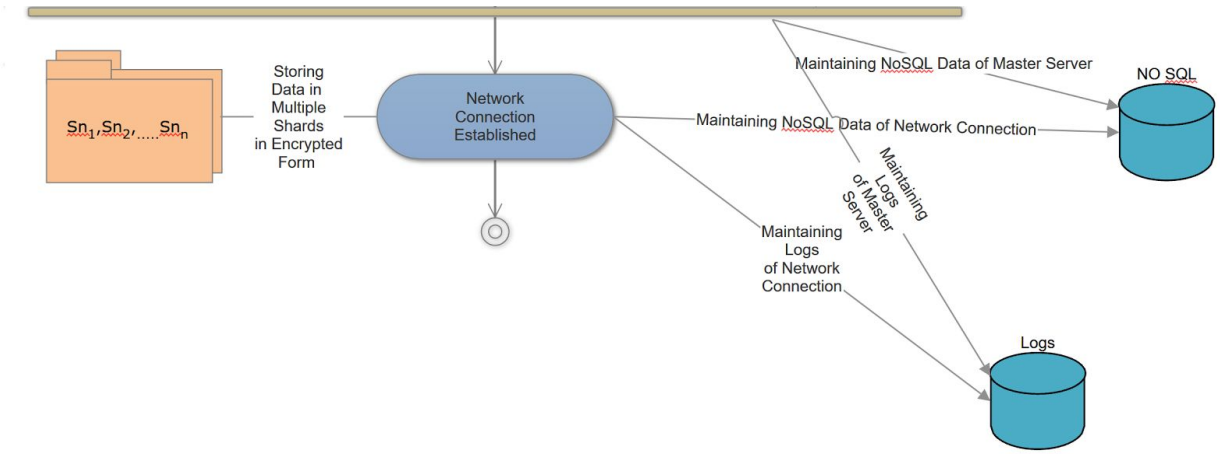
c. Distribution protocol to randomly distribute shards based on availability

2. Download Object
    a. Peer node data availability and data integrity checks
    b. Download protocol to download the data
    c. Merging Shards to retrieve back the downloaded data
        i. Decrypting individual chunks
        ii. Arranging and merging them on the basis of numerical order metadata

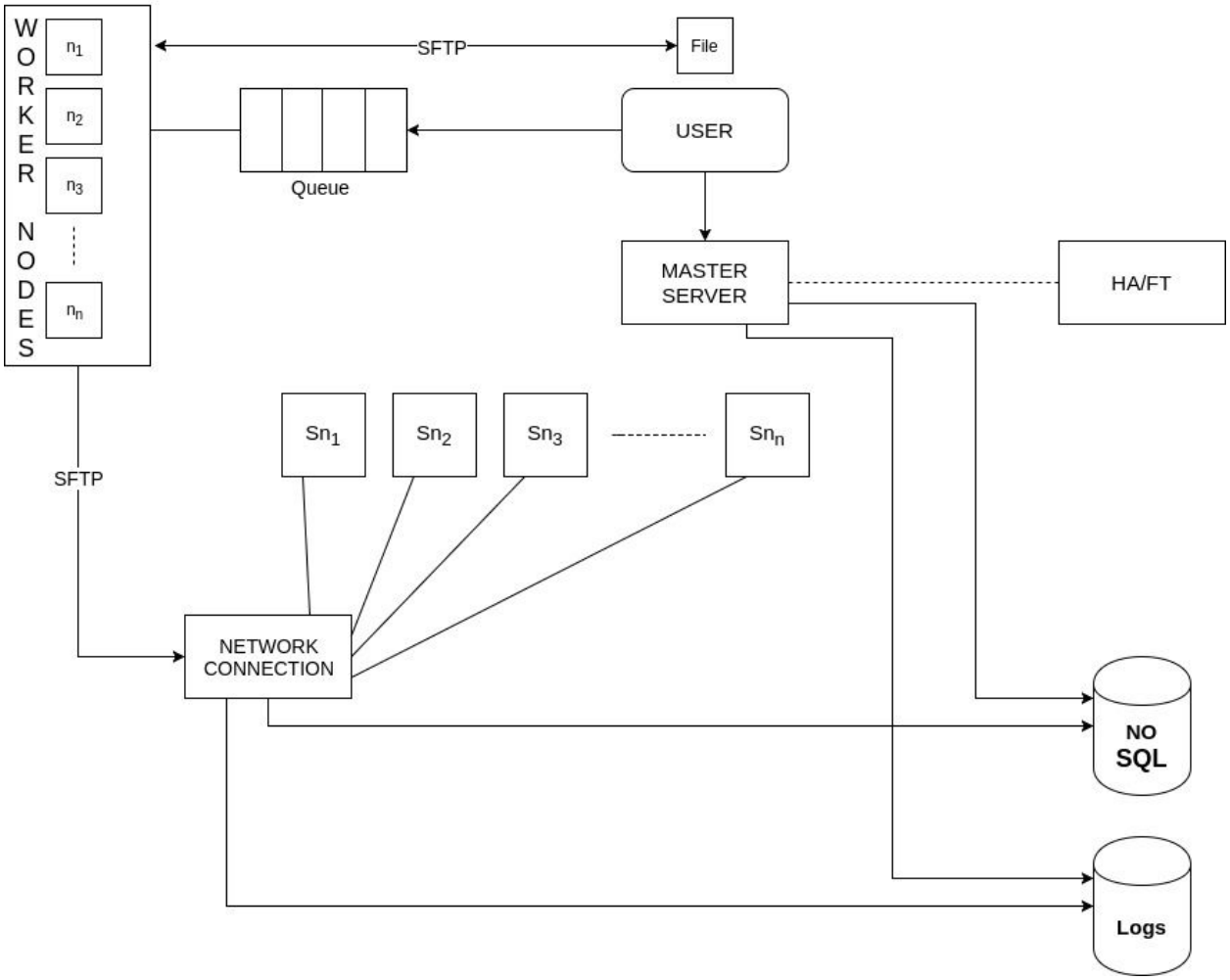
3. Availability Maintenance Service
    a. Scheduled peer node availability and health checks
    b. Data cloning protocol on remote node going offline

Logical Design (UML):





Physical design:



Implementation and Work Division:

The project will require 5 executable codes for the implementation:

1. User (Karan)
2. Master Server (Riddhesh)
3. Queue (Maitrey)
4. Worker Nodes (Kaivalya)
5. Storage Nodes (Bhavya)

Additionally, there will be 2 databases:

1. NoSQL Database to store worker and storage node information
2. Log

The project code will be written in Java as the storage nodes will be computers as well as mobile devices.

Milestone:

Date	**Task**
27th September	● Shard creation and integration ● Creating a master server ● Implementing request queuing ● Creating storage nodes
11th October	● Storing shard and storage node information in databases and logs ● Implementing simple distribution algorithms on worker nodes ● Implementing a test a program to implement an upload operation on the distributed storage system
1st November	● Implementing smarter distribution algorithms on worker nodes based on variability of space in storage nodes. ● Recognizing the best replication factor for shards ● Implementing the list and download operation on the distributed storage system

Literature/References:

[1] A. Razaque, M. Almani and S. Rizvi. “Blackbox: Distributed Peer to Peer File Storage and Backup”. Industrial Electronics, Technology & Automation (CT-IETA), Annual Connecticut Conference, 21-25 May 2017 IEEE.
[2] S. Wilkinson, T. Boshevski, J. Brandoff, J. Prestwich, G. Hall, P. Gerbes, P. Hutchins and C. Pollard. “Storj: A Peer-to-Peer Cloud Storage Network”
[3] B. Chun, F. Dabek, A. Haeberlen, E. Sit, H. Weatherspoon, M. Kaashoek, J. Kubiatowicz and R. Morris. “Efficient Replica Maintenance for Distributed Storage Systems”. NSDI’06: 3rd Symposium on Networked Systems Design and Implementation, 2006 USENIX Association