Comp3100 Group Project

## **Project title**

Distribution operation system of intelligent cloud storage

## **Introduction**

Since traditional SAN (Storage Area Network) or NAS (Network Attached Storage) Storage technologies have bottlenecks in storage capacity and scalability. Moreover, as limitations within the variety of hardware devices deployed, the cost for users to upgrade the system is greatly increased.

However, cloud storage adopts a protractible distributed system. Cloud storage provides external data storage in the network. The system integrates various types of ultra-large storage devices through cluster applications, grid technology, and distributed file systems.

Distribution operation system of intelligent cloud storage is subverting the traditional storage system architecture, which is widely recognized by the industry and users for its advantages of good scalability, cost performance and fault tolerance.

## **Aims**

Cloud storage is a new concept in the storage field. Compare with traditional storage technologies, cloud storage provides better scalability. whereas the requirement to extend storage capability, solely necessitate adding a server to implement it. However, not the requirement to revamp the structure of the storage system.

For the Client, users do not need to care about how the cloud storage system is implemented internally, nor do they need to understand the storage provision method and underlying foundation.

The main goal is that any authorized user can pass Network to utilize data storage and business access services provided by cloud storage distribution systems.

## **Background**

## **4.1 System Architecture & Client-server Communication model**

DOSICS has good scalability, fault tolerance, and internal implementation transparency to users. According to Figure 1, this model is a distributed file system that works by separating multiple types of services. The advantage is that it will not cause the entire operation capacity to be reduced by the occurrence of a specific error. DOSICS provides customers with efficient verification, data storage and data cloud synchronization services.

The communication model of this system is persistent and synchronous. According to Figure 1, the user needs to maintain the network during the entire operation. When the data is completely or not transferred to the cloud database, the system will give a feedback message immediately.

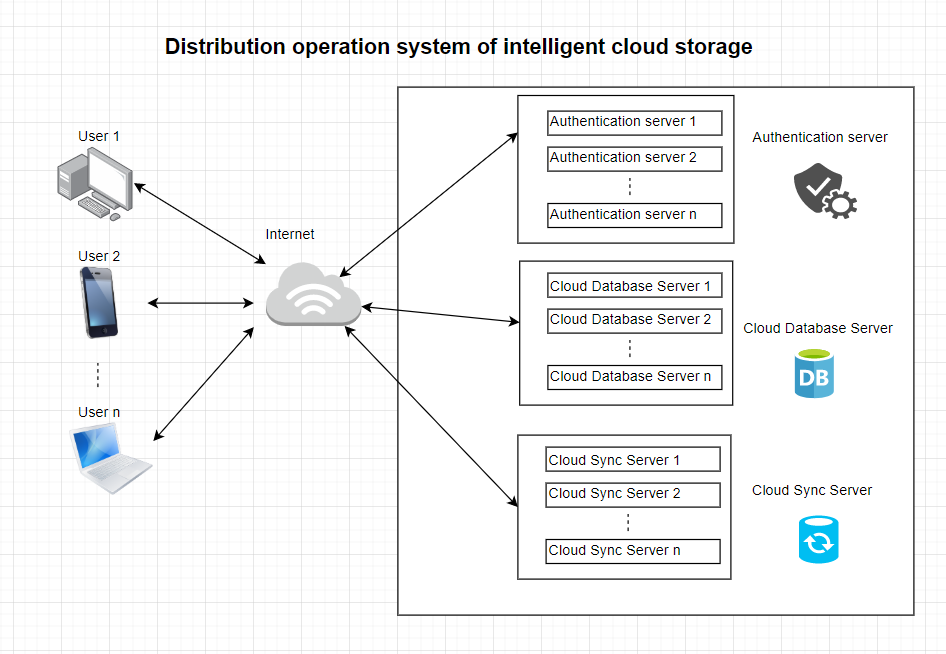


Figure 1. Distribution operation system of intelligent cloud storage (DOSICS) architecture diagram

## **4.2 discrete even simulation**

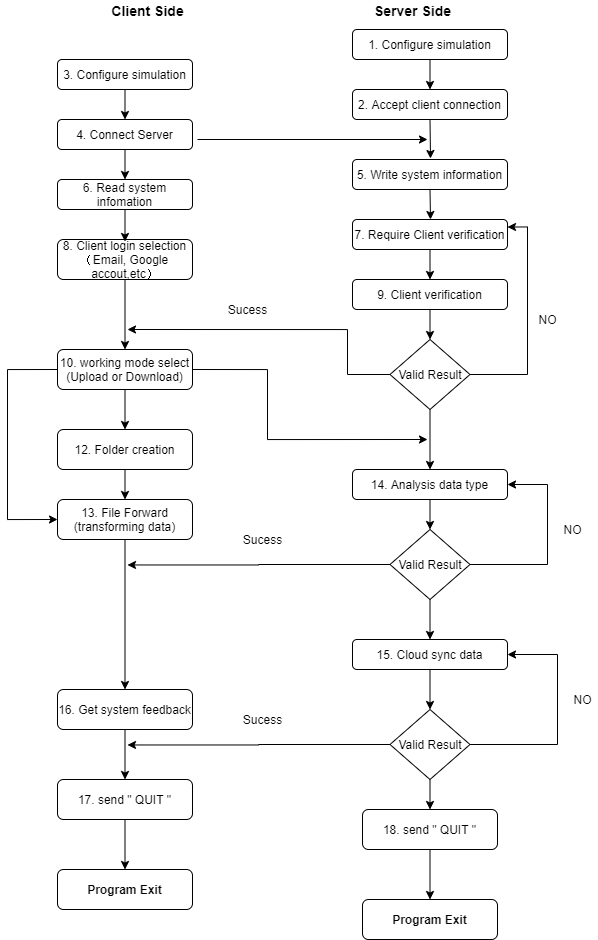
Figure 2 shows the discrete event simulation of this model 

Figure 2. Simulation step

## **Project plan**

## **5.1 Project roles**

|  |  |  |
| --- | --- | --- |
| **Name** | **Student-ID** | **Project Roles** |
| Xinglin Chen | 44089333 | Project Management & Testing |
| Fei Huang | 44129866 | Design Algorithm & Programming |
| Jiahui Lin | 45141916 | Design Algorithm & Programming |

## **5.2 Schedule**

|  |  |  |
| --- | --- | --- |
| **Tasks** | **Details** | **Due Date** |
| Stage 0 | Design document including describing your understanding of the project and project plan | 20/03/2020 |
| Stage 1 | Design and Implement a ‘vanilla’ client-side simulator with a simple job dispatcher | Week 6 |
| Stage 2 | Implement three baseline scheduling algorithms: First-Fit (FF), Best-Fit (BF) and Worst-Fit (WF) | Week 9 |
| Stage 3 | Design and Implement a new scheduling algorithm | Week 12 |
| Demo | presentation, maintenance of data repository (GitHub/Bitbucket) and wiki pages | Week 13 |

## **5.3 Arrangements**

Communication is vital to the success of the project. Team members ought to apprehend the status of the project and the way they're affected. The primary methods that communication for group members is thru Facebook messenger and includes face-to-face conferences twice a week.

## **5.4 Data Management**

## Data management through GitHub to know the status of the project data, this way will greatly improve the efficiency of our cooperation

GitHub Link ：<https://github.com/SnakeCN21/COMP3100-Group-Project>

## **5.5 Programming language**

## In this group project, we agreed that the programming language is developing by java.

## Advantages:

## The core library has network feature packs that support TCP / IP UDP, etc.

## Supports vast data structures in the core library

## Java has many APIs and is very extensive

## Etc

## **References**

* Adler, R. M. (1995). Distributed coordination models for client/server computing. Computer, 28(4), 14-22.
* Kossmann, D., Kraska, T., Loesing, S., Merkli, S., Mittal, R., & Pfaffhauser, F. (2010). Cloudy: A modular cloud storage system. Proceedings of the VLDB Endowment, 3(1-2), 1533-1536.
* Oluwatosin, H. S. (2014). Client-server model. IOSRJ Comput. Eng, 16(1), 2278-8727.
* Sprague, P. J., & Lipscomb, T. H. (1993). U.S. Patent No. 5,247,575. Washington, DC: U.S. Patent and Trademark Office.
* Van der Mei, R. D., Hariharan, R., & Reeser, P. K. (2001). Web server performance modeling. Telecommunication Systems, 16(3-4), 361-378.