

DEMOCRATIZE DATA INTENSIVE COMPUTING USING NEBULA

DR. SIDDARAJU

*Prof & Head, Dept of CSE, Dr Ambedkar Institute of Technology, Bangalore ,India
siddaraju.ait@gmail.com*

LAVANYA S

*Dept of CSE, Dr Ambedkar Institute of Technology, Bangalore ,India
lavanya.s01996@gmail.com*

LEKHANI KELAM

*Dept of CSE, Dr Ambedkar Institute of Technology, Bangalore ,India
lekhanikelam@gmail.com*

MANJUSHREE R

*Dept of CSE, Dr Ambedkar Institute of Technology, Bangalore ,India
manjushreecs073@gmail.com*

MEGHA B U

*Dept of CSE, Dr Ambedkar Institute of Technology, Bangalore ,India
meghabumesh14@gmail.com*

ABSTRACT

Cloud platform is extensively used for data intensive computing. Data intensive computing is a parallel computing that deals with large data sets. Main drawback of current cloud infrastructure is, there is no flexible data mobility which is result of centralization of resources. Restricted data mobility of cloud has made existing cloud infrastructure unsuitable for distributed applications with large datasets. Here we present Nebula-a server which democratizes data intensive computing by incorporating edge computing, and it also avoids replication and provides secure storage by means of data encryption.

KEYWORDS: Democratize, Edge Computing, Data Intensive computing, Data Encryption

INTRODUCTION

Centralized cloud storage data-centre topology is the topology that contains one or few data centers managed by the cloud service provider in a small geographical area. An advantage of this topology is the less costs of operational expenses. A disadvantage can be, since the data is located in single central server, there will be high data traffic which leads to input or output bottlenecks and there can be single point of failure also. High costs are incurred in data migration.

The ultimate goal of distributed computing is to upgrade performance by connecting users and IT resources in a cost-effective and reliable manner. Edge computing is an optimized cloud computing where processing of data occurs at the network edge, very near to the source of data. This in turn can help to reduce data transmission traffic and real time analyzing of data. Due to the lock in period in cloud migrating from one service provider to other service provider is difficult and also incurs cost. Thus, user might maintain accounts in multiple clouds which results in redundancy in storage. Nebula acts as an interface for accessing multiple accounts of single user. Every user can open multiple account in cloud using nebula. It keeps track of the files or data stored by the user. When a user tries to upload a file, the nebula checks for the existence of file in the cloud. It checks for the data in every cloud. If it was not present then file will be uploaded successfully. Else if data or file is present in any cloud the nebula sends that file is already present and then a link to download the file will be provided. File is downloaded only if user enters the key associated with that file. A key is generated when user uploads the file in the cloud. It is stored along with the file. Both the key generation and Encryption of file strengthens the security of the nebula.

DESIGN AND METHODOLOGY

The basic idea of project is to create a common interface to multiple clouds enabling user to maintain data in multiple clouds with minimal management effort. The project also avoids replication of files by checking for duplicates prior to uploading. User needs to first login to nebula server to access any of the cloud in which he holds an account. If user tries to upload file nebula checks if the file already exists in any cloud in which user possess an account, if it is found nebula simply alerts user that file exists in particular cloud and returns a download link to particular file facilitating user to use the existing file. User can download that file only if he would be able to authenticate by entering the key which was generated by nebula at time of uploading the file. If there is no file with same name then nebula will upload the file to cloud selected by the user.

The nebula architecture is implemented using the following modules:

USER INTERFACE DESIGN

This module comprises the functionalities related to user interaction with nebula interface such as registration of new user, authentication of the existing user.

IMPLEMENTING MULTI CLOUDS IN SINGLE APPLICATION

This module is responsible for creating and managing multiple accounts in single application of nebula manager. User can use all accounts via common interface.

GENERATING KEYS

For secure storage purpose, a key is generated when a file is uploaded into the cloud. This key must be stored along with the file.

UPLOAD FILES IN MULTI CLOUD WITH KEYS

The files should be stored in cloud with generating keys. This key is generated to secure user file from cloud providers and other attackers.

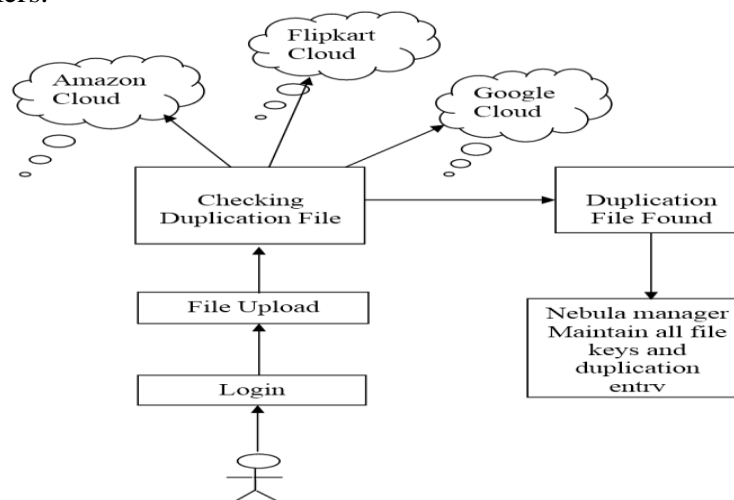


Figure 1. System Architecture of Nebula

DUPLICATION CHECKING IN MULTI-CLOUD

This module comprises functionalities of checking for the duplicate file is different accounts of user before uploading the file. Suppose if file exists nebula manager gives the information to the user about the location of the file.

USER DOWNLOADS WITH KEY CHECKING

This module encompasses the functionality of downloading the available file in the cloud. The user first needs to login to respective cloud where the file is located and later should authenticate using keys for that file to download the file.

IMPLEMENTATION

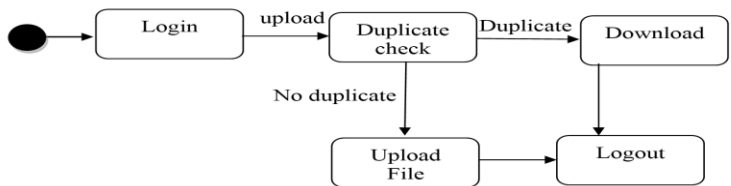


Fig 2 State Chart Diagram

State diagrams give an abstract view of the behaviour of the system. In our state diagram first, the user logs into server and tries to upload a file, if the file already exists a download link will be provided else file will be uploaded successfully.

SEQUENCE DIAGRAM

It shows interactions within a system and the messages exchanged among a set of objects over time. The below diagrams depicts sequence of actions which take place when there is no duplicate files and when duplicate file exists respectively.

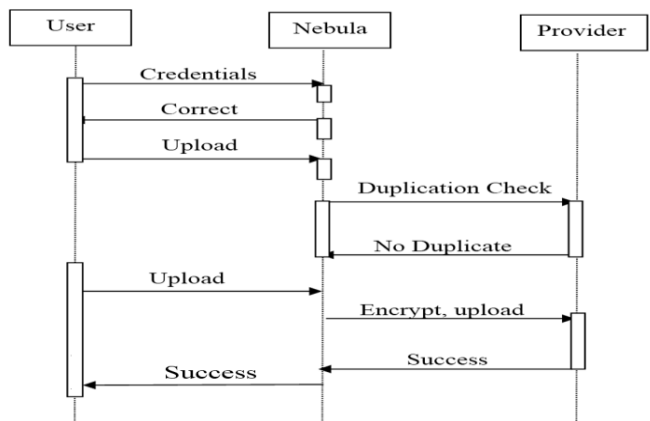


Fig 3. Sequence Diagram for uploading file if no duplicates found

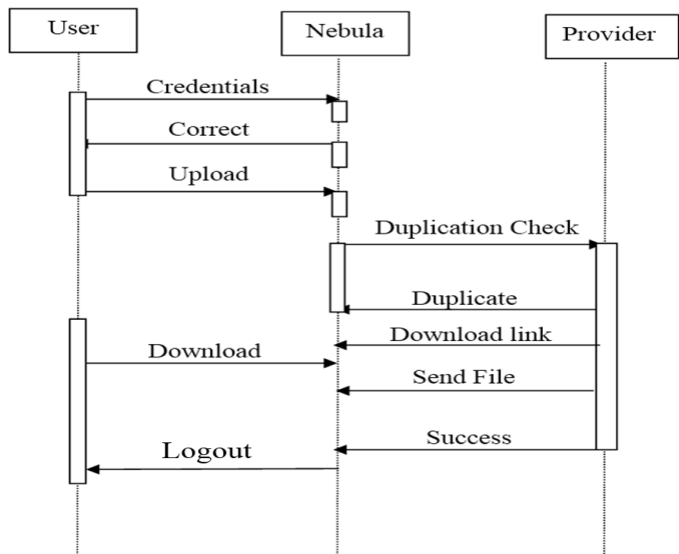


Fig 4. Sequence Diagram for downloading file if duplicates found

RESULTS

With the help of nebula, the system is able to detect duplication, provide data encryption and privacy in cloud.

CONCLUSION

We presented the design and implementation of Nebula, an edge-based platform that enables distributed in-situ data intensive computing. It effectively overcomes problems of centralized cloud infrastructure by using distributed architecture. It provides secure storage by encrypting the data at nebula server before uploading to cloud. It reduces the replication of files stored on cloud and thereby avoids wastage of cloud storage. It can be further enhanced by using digital image processing to compare images in the file.

REFERENCES

- 1) Abhishek Chandra, Jon B. Weissman, Benjamin Heintz. (2013).IEEE Internet Computing, Decentralized Edge Clouds.
- 2) B.Cohen. (2003). Incentives build robustness in BitTorrent, In Proceedings of the First Workshop on the Economics of Peer-to-Peer Systems.
- 3) Fernando Costa, Luís Moura Silva, Michael Dahlin. (2011). IEEE International Symposium on Parallel and, Volunteer Cloud Computing: MapReduce over the Internet.
- 4) Heintz, Chenyu Wang, Abhishek Chandra, Jon B. Weissman. (2013). IEEE International Conference on Cloud ,Cross-Phase Optimization in MapReduce.
- 5) I. Foster, C. Kesselman, J. Nick, and S. Tuecke. (2002). The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration .In Proceedings of the Global Grid Forum.