DCS – A Distributed Cloud Storage Network

Channel Islands
CALIFORNIA STATE UNIVERSITY

Author: Richard Zins • Advisor: Dr. Micheal Soltys • Course: COMP 499

About The Project

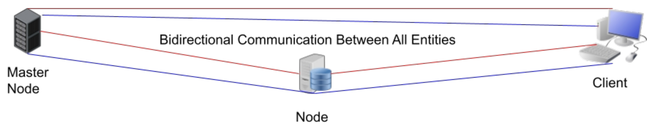
DCS is an open source distributed cloud storage solution for users who don't want to depend on large corporations to keep their data safe and secure. The Master Node stores information about how much storage each Node has available and nothing else. This means that this central point should not be a point of interest for attackers. All communications are bidirectional using a custom protocol that implements requests and responses between all entities. Clients store each data file that they want to upload on individual Nodes across the network. This allows the user's data to be distributed all over the world. Also all communications are encrypted over TLS so the user's data is safe from attackers monitoring their network traffic. Most importantly since this project is open source development is community driven, and it can be easily forked/adapted to meet that communities specific needs.

Tool and Technologies Used

Golang: https://golang.org/
Openssl: https://www.openssl.org/
KeyCDN Rest API: https://tools.keycdn.com/geo

Architecture

The architecture of this project consist of a Master Node, a Node, and a Client. The Master Node stores information about all the other Nodes on the network. All the other Nodes store the end users' data. While the Client interacts with both the Master Node and Node entities in order to find and transmit that user's data.



Example DCS Network Topology

Entity Structure

Below you will find an object representation of the different components of this project.

«entity» **Master Node**
HashMap[NodeUUID] -> NodeData NodeData { NodeIp, NodeKey, Storage } tls keys

«entity» **Node**
HashMap[ClientUUID] -> ClientData ClientData { ClientKey, LocalPathToData } NodeKey NodeUUID tls keys

«entity» **Client**
HashMap[FileName] -> NodeData NodeData { NodeIp, NodeUUID } ClientKey ClientUUID tls keys

Custom Protocol

To better understand how the different entities interact with each other please view the protocol messages below.

Requests:
- ATL storageInBytes (LNode -> MasterNode)
- RFL nodeId key (LNode -> MasterNode)
- NODE storageNeedInBytes (CNode -> MasterNode)
- UPDATE nodeId key (LNode -> MasterNode)
- CHECK nodeId oldIp(CNode -> MasterNode)
- STORE nodeId key2 dataSizeInBytes (CNode -> LNode)
- RETRIEVE nodeId key2 (CNode -> LNode)
- REMOVE nodeId key2 (CNode -> LNode)

Responses:
- ATLR yesOrNo nodeId key (MasterNode -> LNode)
- RFLR yesOrNo (MasterNode -> LNode)
- NODER ipOfNewNode nodeId (MasterNode -> CNode)
- UPDATER yesOrNO (MasterNode -> LNode)
- STORER yesOrNo (LNode -> CNode)
- RETRIEVER yesOrNo dataSizeInBytes (LNode -> CNode)
- REMOVER yesOrNo (LNode -> CNode)
- CHECKR yesOrNo ip (MasterNode -> CNode)

Intuitive CLI

The DCS client implements a simple and robust command line interface. From the CLI the user can store a file on the network; retrieve a file from the network; remove a file from the network; get publicly available information on the host of the Master Node and other Nodes on the network; and review all the commands with the help function.

```
rizins@pop-os:~/CapstoneDemo$ go run CNode.go
Welcome to the DCS Client CLI!
Please type HELP for a list of commands.
> HELP
STORE <filename>: Stores the file you specify on the network.
RETRIEVE <filename>: Retrieves the file you specify from the network.
REMOVE <filename>: Removes that file from the network.
FILES: Lists all your files stored on the network.
INFO: Shows information about the nodes that store your files on the network.
HELP: Shows this list of commands.
>
```

To Summarize

- The end users' data is distributed throughout multiple Nodes on the network
- The project is free and open source hence it has the benefits of being community driven
- Communication between all parties is encrypted over TLS
- All Clients and Nodes have a universally unique identifier
- A Client's secret key is required to retrieve or remove its file from the Node
- A Nodes secret key is required to remove it from the Master Node's list of available Nodes
- A Nodes secret key is required to update it's ip in the Master Node's list of available Nodes

Acknowledgments/Links

I would like to thank Dr. Soltys for his help, guidance, and support on this project. For more information on this project please visit http://dcsrichardzins.cikeys.com/ and https://github.com/RIZY101/DCS