COMP472: Parallel and Distributed Systems

myCloud

Term Project

ÖĞRENMEYE SINIRSIZ ÖZGÜRLÜK

The aim of this project was to demonstrate a simple cloud file system using FUSE and RPYC in python.

Setting

The program was developed on a Oracle VM VirtualBox environment with ubuntu 18. Python 3.6 was used for coding.

Requirements

The following packages are required for running this application;

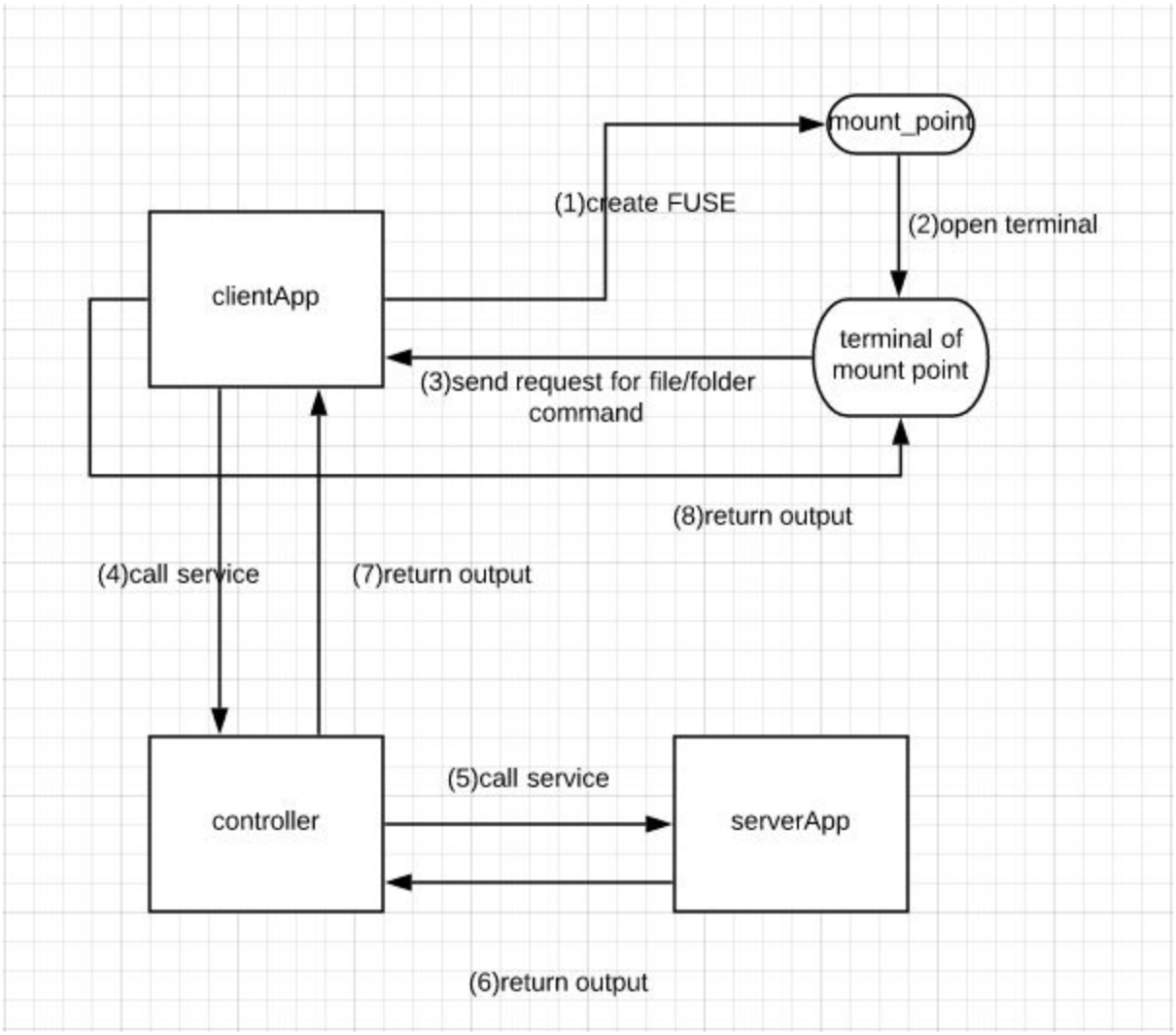
- RPyC
- FUSE
- FusePy
- netifaces (which is default for ubuntu18)

Before Running The Code

There are some configurations that needs to be done before running the code on your computer.

1. Learn your IP address. (Type <hostname -I> in terminal)
2. Put your IP address in the HOST part of serverApp/init.py and clientApp/clientApp.py.
3. Check if the used ports are available in your system.
4. If you are running the code in a virtualbox environment, make sure the machine uses bridged network.
5. Create a folder in your serverApp directory and name it ‘data’.
6. Go to serverApp/services.py and change the ‘data_path’ variable to serverApp/data’s full path.

Code Structure



This is a simple diagram showing the main flow of our application. You start the application by running controller/in.py first. Then, you need to run serverApp/init.py and finally, run clientApp/clientApp.py. If the connections are established successfully, the terminals will indicate that.

Controller

The controller manages the communication between the clientApp and the servers. The controller knows which file paths belong to which servers but it does know the content of the file. It also knows the connected servers and their port numbers and IP addresses. Controller stores all these data in controller/data/data_table and controller/data/server_table.

Controller can handle multiple connections. It may choose randomly or it may also choose a specific server (based on the type of command) to execute a command.

Flow

Let's say we run the command 'mkdir" at the terminal of the mount_point. First, the request will be handled by clientApp and it will call **conn.root.mkdir()**, which will be handled at controller side. At the controller side, the controller will call **conn.root.mkdir()** which will be handled at server side and server will create a folder in their data folder.

Problems We Faced

1. At first, we were unable to use FUSE. Then we found out that fusepy needed a more up-to-date version of FUSE so we developed the code on a ubuntu18 environment.(ubuntu16 will also work)
2. We were able to implement 'ls' and 'mkdir' commands to out application.However, other commands such as 'touch', 'rmdir' or 'open' does not work properly even though the services and methods are included. We still don't know why they don't work.

GitHub

https://github.com/ersinguven/PADS-myCloud