Distributed System Presentation

Gluster FS Clone

•••

Group 5

Dang Vinh Bao
Nguyen Thanh Long

Vu Hoang Linh
Phung Duc Tuan

Contents:

I. Introduction

II. Aim

III. How Gluster works

IV. Pseudo code

I. Introduction:

GlusterFS (Gluster File System) is an open source distributed file system that can scale out in building-block fashion to store multiple petabytes of data.

The clustered file system pools storage servers over TCP/IP or InfiniBand Remote Direct Memory Access (RDMA), aggregating disk and memory and facilitating the centralized management of data through a unified global namespace. The software works with low-cost commodity computers and is based on Linux.

<http://searchstorage.techtarget.com/definition/GlusterFS-Gluster-File-System>

I. Introduction

- A GlusterFS clone
- Replicated volumes: Bricks in the same volume share the same files

II. Aim

- Fully functional “1 server - 1 client”
- Actual files sent and received

III. How GlusterFS works?

Use: WatchService API, an file change modification API.

Reason: low level easy to customize.

III. How GlusterFS works?

Detect: Create, Modify and Delete event.

Problem: because Modify event in Ubuntu actually;

Create a temporary file: .goutputstream(Text Editor) , .<name file>.swp (nano), ...

Modify that file

Delete that file

Recreate/Modify the file that we change

III. How GlusterFS works?

Solution:

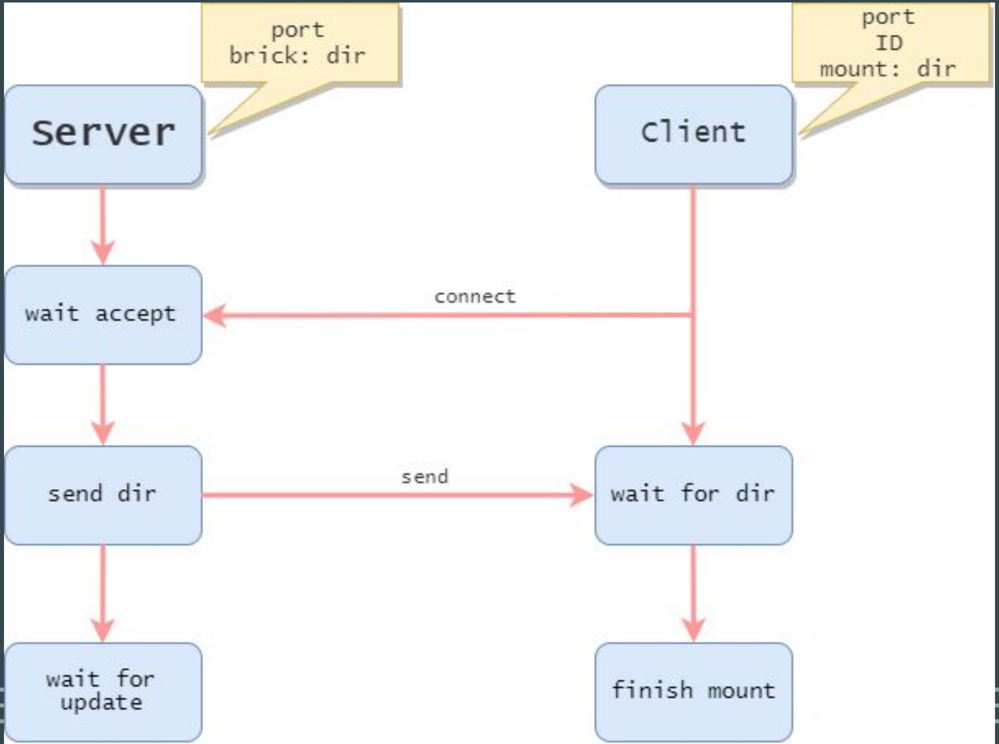
all temporary files seem start with “.” in the beginning

User cannot create file start with “.” in the name using conventional methods

=> ignore change of files with “.” as the beginning of the file name.

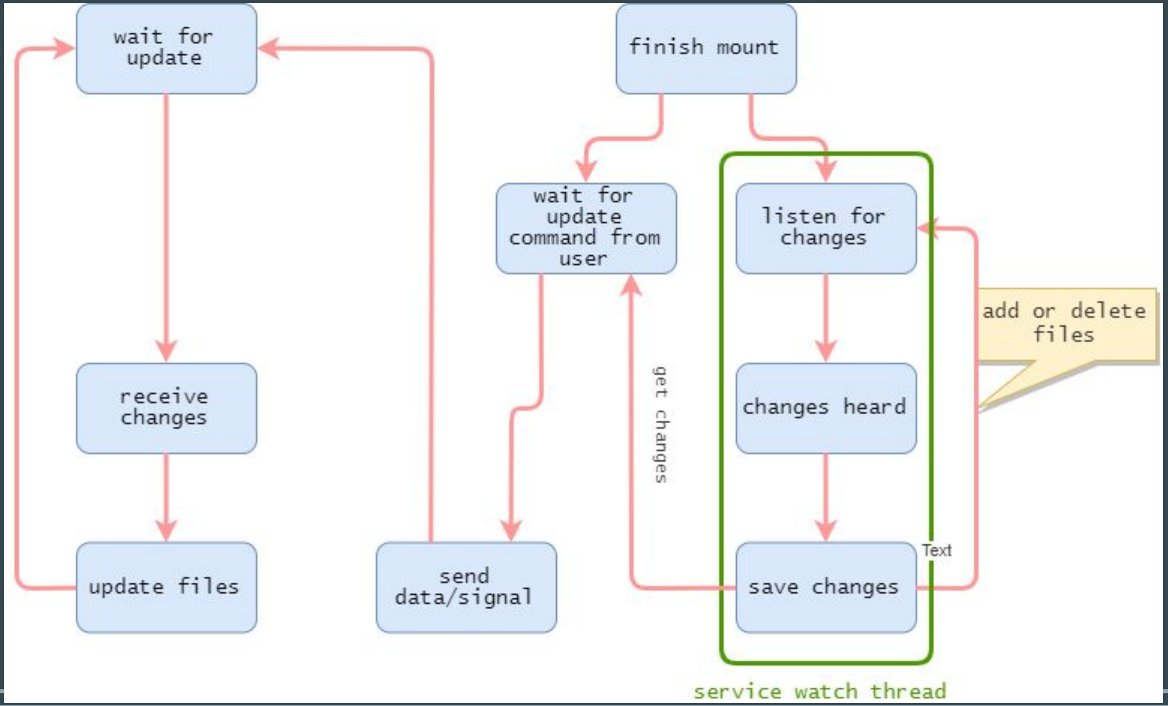
III. How GlusterFS works?

Initialize the server



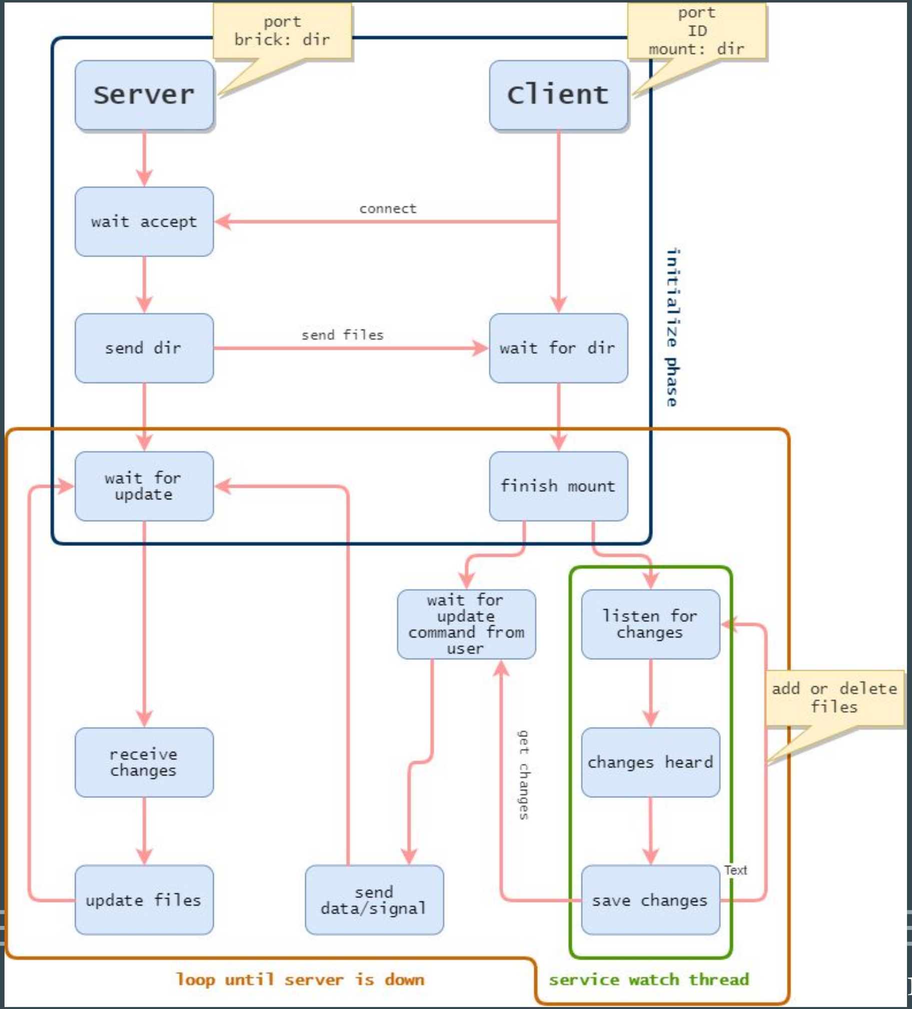
III. How GlusterFS works?

Listening Loop



III. How GlusterFS works?

The complete diagram

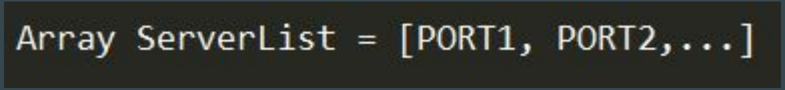


III. How GlusterFS works?

DEMO TIME!

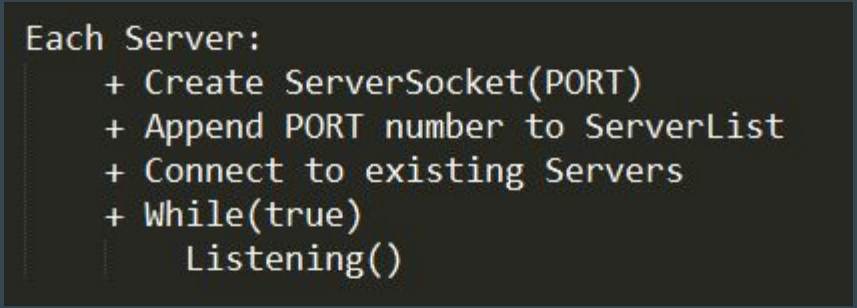
IV. Pseudo code

Global Variable



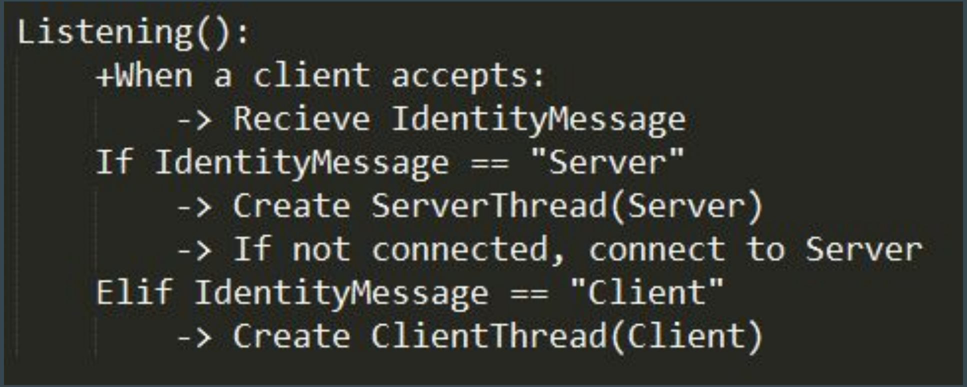
```
Array ServerList = [PORT1, PORT2,...]
```

Intialization



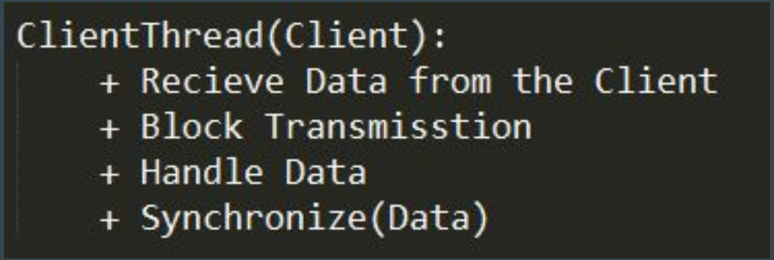
```
Each Server:
    + Create ServerSocket(PORT)
    + Append PORT number to ServerList
    + Connect to existing Servers
    + While(true)
        Listening()
```

Listening Loop

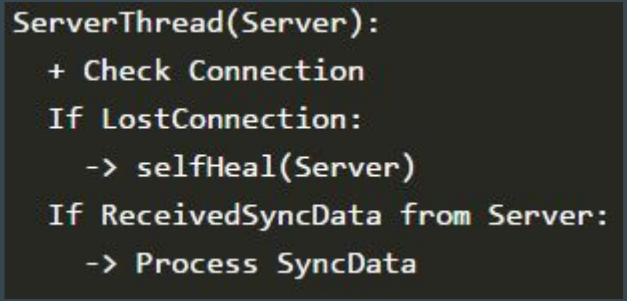


```
Listening():
    +When a client accepts:
        -> Recieve IdentityMessage
    If IdentityMessage == "Server"
        -> Create ServerThread(Server)
        -> If not connected, connect to Server
    Elif IdentityMessage == "Client"
        -> Create ClientThread(Client)
```

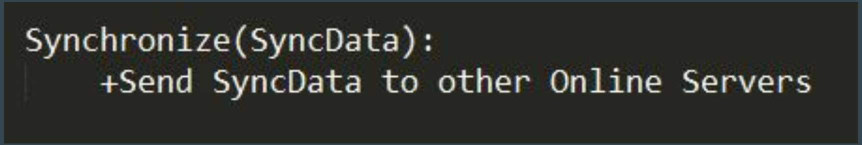
IV. Pseudo code



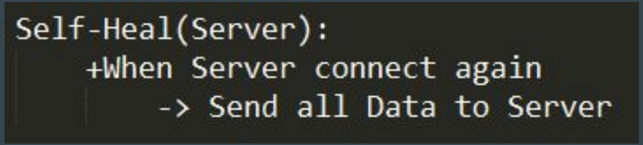
```
ClientThread(Client):
    + Recieve Data from the Client
    + Block Transmisstion
    + Handle Data
    + Synchronize(Data)
```



```
ServerThread(Server):
  + Check Connection
  If LostConnection:
    -> selfHeal(Server)
  If ReceivedSyncData from Server:
    -> Process SyncData
```



```
Synchronize(SyncData):
    +Send SyncData to other Online Servers
```



```
Self-Heal(Server):
    +When Server connect again
        -> Send all Data to Server
```

THANKS FOR LISTENING!