NASA 2017 Final Project
SA#2 NFS Server Static Load Balance

B03901018 電機三 楊程皓
B03901078 電機三 蔡承佑
B05902053 資工一 陳奕均

June 25, 2017

1 Introduction

Currently, we have only one NFS server supplying all the workstation. Now suppose we have N NFS server, providing M users to access their home directory, we have to support following functions:

1. If a new grade of students joins, we have to use scripts to create their corresponding home directory on NFS.
2. If new NFS servers joins, we have to adjust the home directories to balance the load.
3. If there are old NFS servers discarded, we have to move the home directories to other NFS servers.

In this project, all VMs are under CentOS 7

2 Workstation Environment Simulation

2.1 Structure

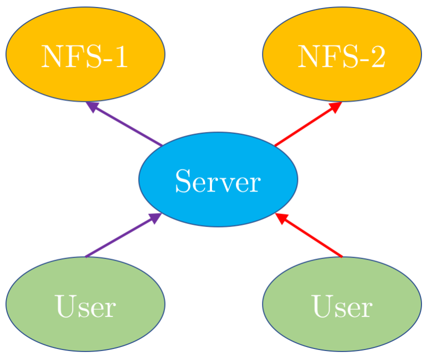

Figure 1: Structure of the workstation

2.2 NFS Setup

In NFS server, we must write the IP of the server in `/etc/exports` to share directories to the workstation server. It is important that the IP address of NFS server and workstation server must be constant. In `/etc/systemd/network/25-wired.network`, we add

```
[Match]
Name=enp0s8

[Network]
Address=192.168.100.3/24
```

So that the IP addresses of NFS server and workstation server will be constant, which is the value given manually. And the mount points are divided by grades of users. i.e. b03, b04, b05, etc. For example, in NFS-1, mount points are `/etc/nfsshare/{b03902,b04902,b05902}`

2.3 Server Setup

First, to decrease system loading, we use `autofs` to mount the NFS directories. Only when the mounted points are used will they be mounted. If a mount point is not used for a given time, it will be unmounted.

3 Creating One Single User

There are several steps we must follow to create a user:

1. Give a user name and password
2. Set the home directory
3. Link the home directory to NFS
4. Change the owner of the home directory of the user.

Part of our script to do this is shown as following:

```
useradd $username -d /home/$grade/$username
echo -e "$username\n$username\n" | passwd $username >& /dev/null
mv /home/$grade/$username /autofs/$nfs/$grade/$username
ln -s /autofs/$nfs/$grade/$username /home/$grade/$username
chown -R $username /autofs/$nfs/$grade/$username
```

After execution of this script, we create a new user whose home directory is a symbolic link to the directory lies in NFS server.

4 Creating or Deleting A Group of Users

Suppose now we want to create accounts for freshmen, e.g. b06902{001-120}. Or more generally, we want to create or remove tens of or hundres of users. The only data now we have is the current user_list and the current distribution. All we need to do is shown in Figure 2. There are two main changes we have to manage: on **NFS server** side (distribution) and **workstation server** side (add/remove home directories). To be more clearly, following are steps we should do:

1. Backup the old user_list and generate a new user_list base on the old one upon need.
2. Call `distribute.out` and input the modified user_list and old distribution. This program will compare those files to redistribute the newly created user or remove users to balance the distribution.
3. Call `group_user_change` to compare the previous user_list and modified user_list to figure out the home directories links we have to modify.
4. Call `update_users.sh` and input distribution/changes and group_user_change to complete the modifications on workserver server and NFS server.

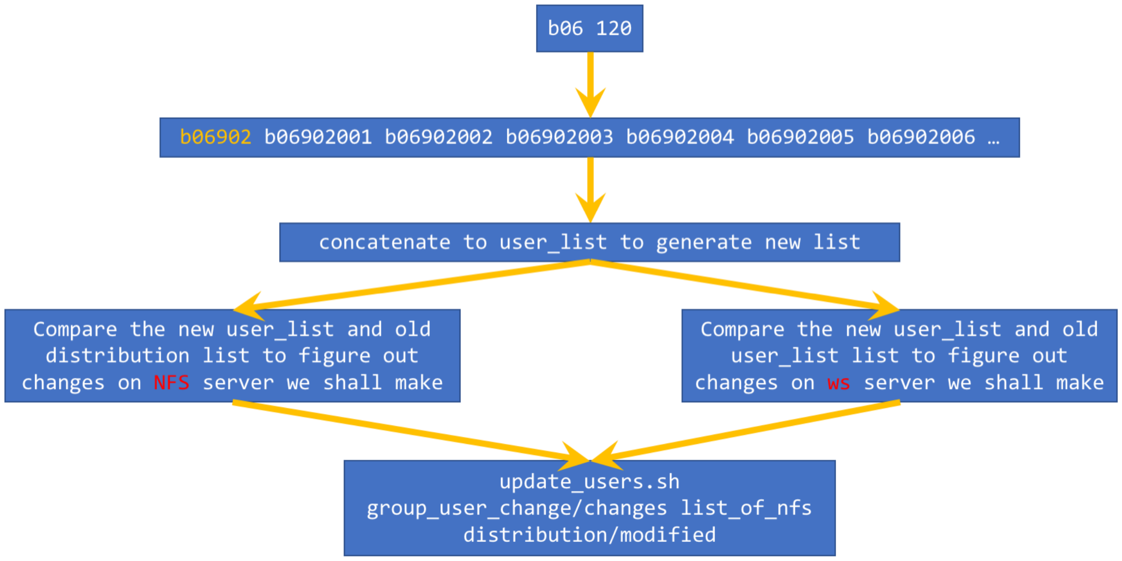
b06 120
b06902 b06902001 b06902002 b06902003 b06902004 b06902005 b06902006 …
concatenate to user_list to generate new list
Compare the new user_list and old distribution list to figure out changes on NFS server we shall make
Compare the new user_list and old user_list list to figure out changes on ws server we shall make
update_users.sh
group_user_change/changes list_of_nfs
distribution/modified

Figure 2: flow diagram of creating/removing a group of users
.

5 Creating New NFS Server

Steps are as following:

1. Setup NFS server for the workstation server to connect to and mount.
2. Generate a new distribution list and find the users that are to be moved.
3. Move the directories of these users to the new server.
4. Update the symbolic links of the home directories of these users.

5.1 Setup NFS Server

For administrator's convinence, we can `ssh` to the NFS server from workstation server and setup directories to share in `/etc/exports`. Then setup the autofs at the workstation side (`/etc/auto.master`, `/etc/auto.nfs*`)

5.2 Find The Users To Be Moved

We write a C++ program to do this job. The input is the previous distribution list, and output would be a new distribution list and the modification we should make. We evenly take some users from each server, and move them to the new server.

5.3 Move The Directories

According to the modification, move the files of specified users from the previous nfs to the new nfs. This job is done using the same script as adding / deleting users since the input format is set the same.

5.4 Update symbolic links

In the end, we update the symbolic links of the home directories of these users according to the modification. After this, next time the users will login with new nfs servers keeping their home directories.

6 Deleting Old NFS Server

Quite similar to adding new NFS server, but now we need not to setup a new server. Steps are as following:

1. Decide the new distribution.
2. Move the directories to the new distributed NFS server.
3. Update the symbolic link of the home directories of these users.
4. Unmount the NFS servers and update the workstation server's autofs configuration.

6.1 Decide The New Distribution

Similar to adding new NFS server, we use a C++ program to distribute the users on the to-be-deleted server to other servers, and figure out the users that are influenced. We can delete multiple NFS servers at the same time to avoid moving the data more than once by deleting them one by one.

6.2 Move The Directories to New NFS Server

Same as the above. Use the same input format and script to do this job.

6.3 Update Symbolic Link

Same as the above.

6.4 Unmount NFS servers

In the end, the NFS servers we want to shutdown have no critical data now and can be safely removed from the system. Update the workstation server's autofs configuration and the NFS servers are off-duty.

7 Difficulties

Every single job looks easy, but how to integrate them for more convinient use is the main problem of this project. Though each step looks easy, but we nead a clear structure to specify them. For example, I/O format should be accord between programs/scripts, and all programs and scripts should be packeted into a single script for administrator's convenience.

There are still some difficulties.

7.1 Mount Points & NFS Servers Arrangement

In the ideal situation, the load of each mount point should be the same so that autofs can have the highest probability to unmount from time to time. Or in another scheme, we may put those who have the highest loadings in the same mount point so that other mount points can be unmounted longer. The loading of each NFS servers should be the same, and the amount of data should be fairly distributed so that when one of the servers fails, we need the least effort to restore the data and fix the system.

In reality, it's hard to determine the loading of each user in this project, so we assume each user has the same loading. Furthermore, with the requirement that we should change the least amount of user's data during adding/removing users/NFS servers, it's hard to come up with a good algorithm to maintain the optimal distribution when adding/removing users/NFS servers are all possible. Currently we use groups as mount points for simplicity (such as b05902, b06902, others), although number of users in each group may not be the same. Then we distribute the users into NFS servers with the same size. When adding/removing users/NFS servers, we maintain the same number of users in NFS servers so that the expected loading of each NFS server is the same and the effort to move the data is minimized.

7.2 Selinux Context of home directories

When we were trying to login into users we just created, we cannot set selinux context of the symbolic link correctly (seems to be unconfined_u:object_r:user_home_dir_t:s0). The system will show “permission denied” when trying to load the home directory. However, the problem only occurs when trying to login with tty. But somehow with ssh, this problem doesn’t exist. We temporarily use selinux permissive to login with tty.