EEL 6935 Spring 11 (Distributed Computing) Homework 5

Addendum (Optional)

To ease your development/uploading process, we are providing this tutorial. Though it is NOT required it would be very helpful. The following will be discussed.

- NFS setup
    - Managing users
    - Installing NFS client
- Passwordless login
- Packages installed on the machines
- Note on runaway procecesses

Please replace "*login*" with your username on the course website.

NFS (Network File System) setup

The goal is to connect all the machines you are using by means of the NFS protocol. In other words, the file systems of faraday, cyberg, nova, and your three VMs (virtual machines) are to be connected in a transparent manner. The most obvious and significant benefit of doing this is that you may save considerable amount of time and effort to move files across the machines compared to when using low-level commands such as scp.

The six machines will form a NFS network. The server is the nova machine (192.168.135.12) and the rest are the clients. You may have found that faraday and cyberg are already coupled to the server. The client directory name is "nova" under your home directory. Here, you will connect your three VMs to the server.

The first step is setting up the NFS clients on the virtual machines. In fact, the NFS server part is already done for you.

On faraday or cyberg, identify your user id by typing 'id'. This is different from your username. The user id is a number assigned each username and is used for NFS.

```
$id
```

You should see a number like '30xx'.

Now, do the following steps for each of your VM.

Managing users

Login to one of your VM and switch to root account to change password. For the following example, assume that the VM's IP address is 192.168.0.5. Please replace it with your own VM's.

```
$ssh -i login-key ubuntu@192.168.0.5
$sudo -i
$passwd
```

We will also remove other unnecessary things. For example, you may want to ssh the VMs without –i option. In order to do that, edit the line 50 of the file /etc/sshd/sshd_conf as below.

```
PasswordAuthentication yes
```

Restart ssh daemon.

```
$/etc/init.d/ssh restart
```

Now, add a new user same as your username with the *userid* obtained from faraday or cyberg in the above steps

```
$adduser -uid userid login
```

Set the password when prompted. Exit the VM.

Installing NFS client

On the same VM, login as root without –i option and run the following commands to copy the installation files from nova machine. Note that there is a space before the last period (.) in the last scp command below.

```
$ssh root@192.168.0.5
$mkdir nfs
$cd nfs/
$scp login@192.168.135.12:/home/sp11dc/nfs/*.deb .
```

Just answer “yes” when prompted for authentication.

Now, install the following packages in the order as shown.

```
$dpkg -i portmap_6.0.0-2ubuntu1_amd64.deb
$dpkg -i libgssglue1_0.1-4_amd64.deb
$dpkg -i librpcsecgss3_0.19-2_amd64.deb
$dpkg -i libnfsidmap2_0.23-2_amd64.deb
$dpkg -i nfs-common_1.2.2-1ubuntu1_amd64.deb
```

Once NFS client is installed, edit the configuration file /etc/default/nfs-common as follows.

```
NEED_IDMAPD=yes
NEED_GSSD=no
```

Then, mount the remote directory at the NFS server which is nova.

```
$modprobe nfs
$mount -t nfs4 192.168.135.12:/home/login /home/login
```

To unmount the directory,

```
$sudo umount -f /home/login
```

Exit the VM.

```
$exit
```

You can check the mounted directory set as your home directory.

```
$ssh 192.168.0.5
```

After you've done this procedure for each VM, you can compile your program once on one machine and run it on any machine. For example, you would upload a source file from your local machine to the mounted directory (~/nova) on faraday or cyberg, compile it on nova and run it on the 3 VMs.

However, you might encounter an error message when you exit vi.

```
E137: Viminfo file is not writable: /home/username /.viminfo.
```

Then just change the permission of the file .viminfo.

```
chmod 666 ~/.viminfo
```

Passwordless login
You may want to login without password across the machines.

On the client machine, run the following command and press enter whenever prompted.

```
$ssh-keygen -t rsa
```

Then run the following command where "remote_server" is the server you want to access.

```
$cat ~/.ssh/id_rsa.pub | ssh remote_server "cat - >> ~/.ssh/authorized_keys"
```

For example, if you want to access faraday without password on your local machine,

```
$cat ~/.ssh/id_rsa.pub | ssh login@faraday.acis.ufl.edu -p2222 "cat - >> ~/.ssh/authorized_keys"
```

Packages installed on the machines

On a VM

- Java: 1.6.0_24
- C/C++: gcc 4.4.5
- Python: 2.6.6
- Perl: 5.10.1

On nova

- Java: 1.6.0_24
- C/C++: gcc 4.4.3
- Python: 2.6.5
- Perl: 5.10.1

On faraday and cyberg

- Java: 1.6.0_04
- C/C++: gcc 4.1.1
- Python: 2.4.3
- Perl: 5.8.8

Though a program can be compiled on faraday or cyberg, you'd better do it on nova since the package versions of nova are closer to the VMs than those of faraday or cyberg.

Note on Runaway Processes

Your threads should terminate gracefully. While testing your programs run-away processes might exist. However, these should be killed frequently to avoid wasting of resources and potential memory leaks.

To check your processes running on a local machine:	ps -u <your-username>
To kill all the processes with the same name, type:	skill <process-name>
To check runaway processes on remote hosts:	ssh <host-name> ps -u <your-username>
And to clean:	ssh <host-name> skill <process-name>