Computing Cluster

The course has available Linux virtual machines on university infrastructure that we will use in programming assignments.

About the cluster

The cluster has multiple VMs. Some capacity is reserved for running system resources and is not available for compute. The VMs are accessed through SSH, and services on the main node are accessed through SSH tunneling.

SSH key

Access to the cluster will be restricted to enrolled students and teaching assistants (exceptions to this rule can be considered, contact the professor about them). Students need to submit a public key on the course website, which they will use to log into the cluster.

For students who do not yet have an SSH key, instructions for generating one can be found under the Generating a new SSH key section of GitHub's guide. Additional information about SSH keys is available on ssh.com's tutorial.

If you are running Windows, an OpenSSH client may be preinstalled and the commands in the guides above will just work. If an OpenSSH client is not installed, you can install it directly from Microsoft following these instructions; you need only the OpenSSH *client*; do not install the OpenSSH *server* package.

Submit *only* the public key (i.e., the `.pub` file) created in the key generation process. Keep the private key on (or copy it to) the device(s) you will use to work on the assignments. Never share private keys.

A user will be created for each student. Private files must be stored in your `home` directory. The system has some disaster recovery features, but each student is responsible for backing up their data, just in case.

Accessing the cluster

Student usernames will be the same as their university ID. After your account has been generated, you will be able to log into the cluster using:

```
ssh <id>@<hostname>
```

A more convenient way to access the cluster is by creating a `.ssh/config` file, and configuring an entry for the cluster's VM. The following configuration will forward ports 8081 and 8088 on your machine to ports 8081 and 8088 on the main VM, which run the submission front-ends. You can then open the front-ends by navigating to `http://localhost:8081` and `http://localhost:8088` on your browser after SSH'ing into the main VM.

```
Host cluster
  User <username>
  Hostname <hostname>
  Port <port>
  LocalForward 8088 <hostname>:8088
  LocalForward 8081 <hostname>:8081
```

NOTE: The VM's `<hostname>` is `timbersaw.winet.dcc.ufmg.br`, but the `<port>` is not defined yet. We're working on setting it up.

Workspaces

You have two main areas to store data in the cluster. The first is your `home` directory, located at `/home/<username>`, as in any Linux machine. You have full control of your home directory.

The second storage area is on HDFS (Hadoop's Distributed File System). HDFS is where Hadoop and Spark applications will read data from, and where results will be stored. As we will see in the course, HDFS distributes data in the cluster, providing redundancy and improving performance of applications by allowing computation to run on the same server where data is stored. The default path for users is in a directory given by the username, i.e., `/user/<username>`, which will be empty if you have not done anything on the cluster yet.

To check what files are in your HDFS directory, you can run the following on the main VM:

```
~$ hdfs dfs -ls /user/<login>
```

We have not configured disk quotas on the cluster. Again, please be considerate with colleagues and avoid unnecessarily grabbing disk space. In particular, check output and log files generated by your programs to ensure they are not unnecessarily large.

Sharing the cluster

The cluster has sufficient capacity for the programming assignments, but ill-engineered or inefficient solutions may put a strain on the available resources. As the cluster is shared across all students, please be considerate to your colleagues and avoid hogging compute and memory resources. Expect high cluster utilization close to assignment submission dates. Plan to work on assignments early and throughout the development period to avoid high-utilization periods.

Always specify the optimal amount of resource for your applications and avoid opening interactive terminals for long periods of time. Do not try to change anything that is not on your workspaces.