Instructions for Assessed Exercises

Dr. Nikos Ntarmos <nikos.ntarmos@glasgow.ac.uk>

Introduction

The following describe how to connect to one of the computers in the SoCS network, and how to use said computers to access the BD4 clusters. Users logged on to one of the lab PCs in the Boyd Orr or Sir Alwyn Williams buildings may skip the next section and go directly to "Programmatic access to the BD4 clusters".

External access to the SoCS network

All computers in the SoCS network (including lab PCs and course servers) are hidden to the outside world by one or more firewalls, and the computing nodes comprising the cluster for the BD4 course are no exception. This holds for all computers not directly connected to the School's network, including computers connected to the *eduroam* wireless network. In order for students and staff members to be able to bypass these firewalls, the School is using a relatively low capacity server, called "sibu.dcs.gla.ac.uk" ("sibu" for short). **Note:** Sibu is meant to be used **only** as an intermediate step and not as a general purpose server/host, as its processing capacity is rather limited.

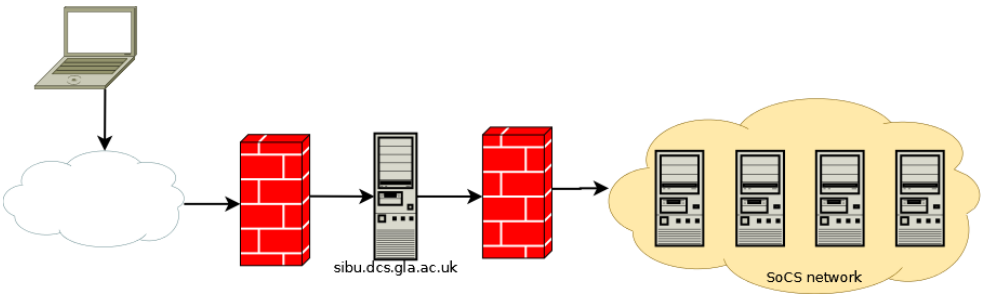

Figure 1 Sibu as an intermediate SSH hop to the SoCS network

Sibu is accessible from most networks outside the SoCS network block; however, there are several cases where connections are being dropped/rejected by Sibu. This may happen intermittently if, for example, you are connecting from a mobile network (i.e., over 3G/4G) or from a remote network yet unknown to Sibu's firewall. In these cases, you'd need to connect over VPN to the University, and then to ssh to Sibu. See [1] for instructions on how to configure your computer for VPN access.

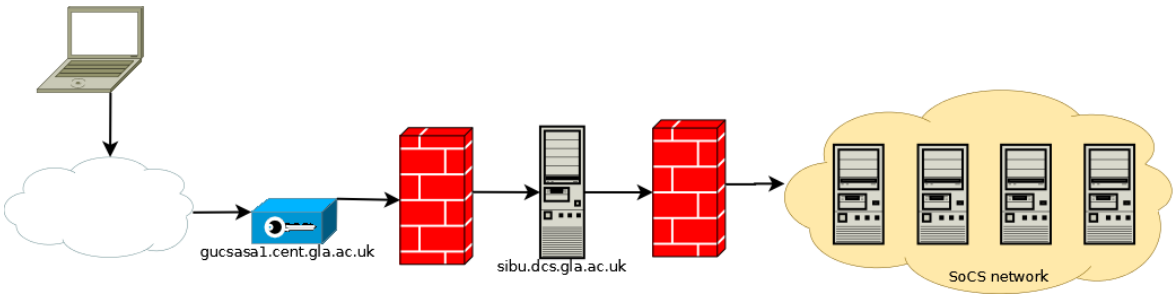

Figure 2 VPN+Sibu

[1] http://www.gla.ac.uk/services/it/flexaccess/vpn/offcampusaccess/

Sibu auto-mounts the users' home directories over the network (using NIS+/NFS); this means that all of your files are directly accessible from Sibu, and any file you upload to your home directory on Sibu is also directly accessible by any other Linux/*nix host in the SoCS network. This allows you to copy files to and from the SoCS network by just using Sibu as a mediator. You can use *scp* on Linux/*nix/MacOSX computers, or *WinSCP* on Windows hosts.

```
user@home:~$ scp myfiles.zip user@sibu.dcs.gla.ac.uk:
user@home:~$ scp -r myfiles/ user@sibu.dcs.gla.ac.uk:
```

Figure 3 Copying files/directories with scp

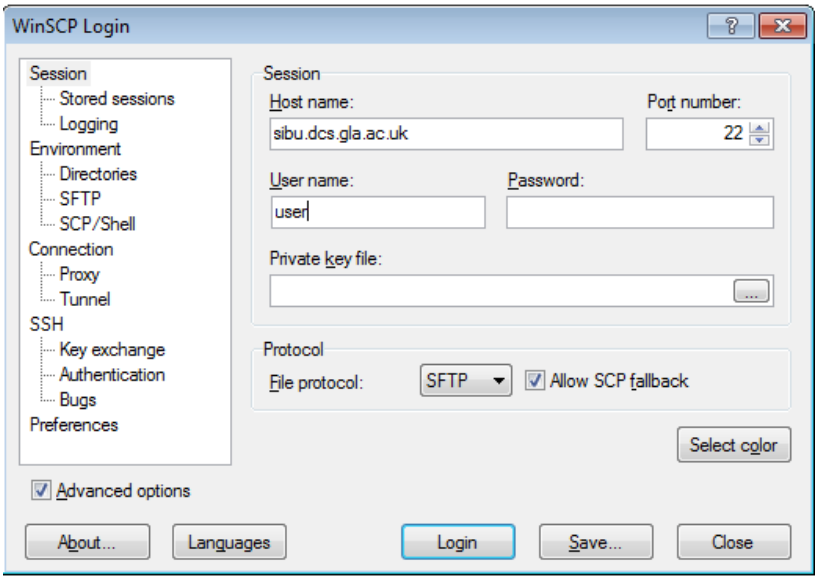

Figure 4 WinSCP

For your assessed exercises, you will need to be able to (compile and) run Java 7 code which will be connecting to the BD4 clusters in order to submit MapReduce jobs and access data stored on HDFS and/or HBase. Access to the cluster nodes is only allowed for computers within the SoCS network, as these nodes are behind the firewalls mentioned earlier. Thus, in order for your code to access these clusters, it will have to be executed from a host within the School's network (e.g., any of the lab PCs in Boyd Orr). Currently the L4 lab PCs are configured to dual-boot Windows and Linux. However, for the following to work you will need to locate a PC booted into Linux, as Windows doesn't provide SSH and/or shell sessions by default.

```
1. user@home:~$ ssh user@sibu.dcs.gla.ac.uk

2. user@sibu:~$ ssh bo620-XX
   (where: XX ϵ [01, 02, 03, …, 30])

3. user@bo620-XX:~$ java …
```

Figure 5 SSH hop through Sibu to L4 lab PCs

Finding such a PC may be an exercise in patience, as other users may reboot the PCs to Windows without first asking for permission. To this end, the School has provided us with a separate server,

named “karkar.dcs.gla.ac.uk” (or “karkar” for short). This is a Linux host with a much higher processing capacity than Sibu, thus you may use it to compile and run your code accessing the BD4 clusters. Note that you will still have to hop through Sibu (and/or the VPN concentrator), as Karkar is not accessible from outside the School’s network. In any case, Glasgow University (GU) students are urged to use lab PCs whenever possible, as Karkar will also be used by fellow students from the Singapore Institute of Technology (SIT) enrolled in this course.

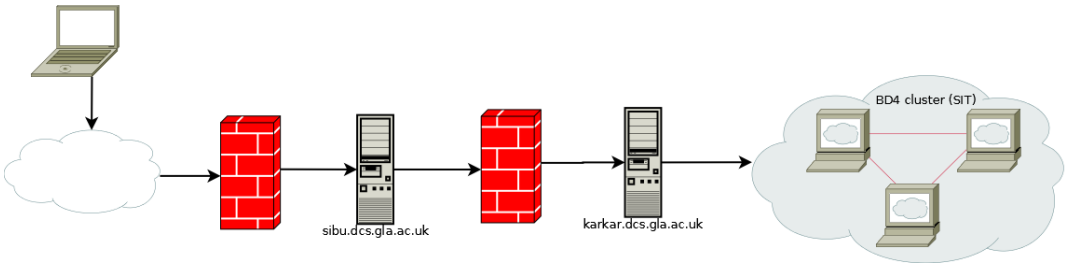

Figure 6 Sibu+Karkar

Last, you may want to access the web UIs of the BD4 clusters (e.g., to get statistics and logs of your jobs, list files on HDFS, examine your HBase tables, etc.). Since the servers for these UIs are running on the cluster nodes, you will have to tunnel your HTTP connections through your (VPN+) SSH connection. The easiest way to do this is via SSH’s built-in support for SOCKS proxying. The first step in using this is enabling dynamic port forwarding on the SSH connection. This is accomplished via the “*-D*” command line option, or via the corresponding UI knobs (e.g., in Putty for Windows hosts).

```
1. user@home:~$ ssh -CD 1080 user@sibu.dcs.gla.ac.uk

2. user@sibu:~$ ssh karkar

3. user@karkar:~$ java …
```

Figure 7 SSH hop through Sibu to Karkar, with SOCKS proxying

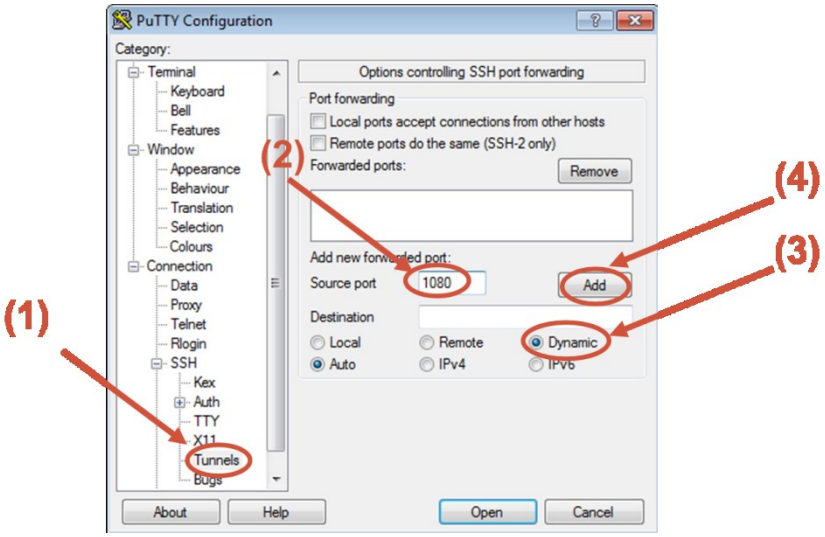

Figure 8 SOCKS proxying in Putty

Once the SOCKS proxy is in place, you then need to configure your web browser of choice to connect through a SOCKS v5 proxy, listening at IP 127.0.0.1 and port 1080. Please refer to your browser's documentation about how to best accomplish this, or contact us for further instructions. The URLs for the NameNode, JobTracker and HMaster UIs will then be:

- For UG students:
  - NameNode: http://bigdata-01:50070/
  - JobTracker: http://bigdata-01:50030/
  - HBase Master: http://bigdata-01:60010/
- For SIT students:
  - NameNode: http://eysturoy:50070/
  - JobTracker: http://eysturoy:50030/
  - HBase Master: http://eysturoy:60010/

Programmatic access to the BD4 clusters

Once logged on to a computer within the SoCS network, you will be able to access the BD4 clusters programmatically by using the appropriate configuration files supplied to you through Moodle (under "Supporting material"). These configuration files should be parsed and added to your Java client code configuration object, by using the *addResource()* method of the *org.apache.hadoop.conf.Configuration* class (or one of its appropriate subclasses).

```
import org.apache.hadoop.conf.Configuration;
[…]
Configuration conf = new Configuration();
conf.addResource("client-conf.xml");
[…]
```

Figure 9 Configuration for Hadoop jobs over HDFS files

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.HBaseConfiguration;
[…]
Configuration conf = HBaseConfiguration.create();
conf.addResource("client-conf.xml");
[…]
```

Figure 10 Configuration for Hadoop jobs over HBase

There will be two such configuration files on Moodle: one for GU students ("*client-conf-ug.xml*") and one for SIT students ("*client-conf-sit.xml*"). Please take care to use the correct one!

In order for you to compile and run your code at home (or in the lab), you will need a basic set of Jar files. These will be supplied to you through Moodle (under "Supporting Material"). Alongside these files there will also be a set of shell scripts which you can use to execute the "*mapred*", "*hdfs*" and "*hbase*" commands mentioned in the tutorials. Please do **not** include these or the jar files in your submissions on Moodle; as a matter of fact, do **not** submit any binary (.class, .jar) files, as these are of no use to us and only bloat the size of the submissions.

You will find all of the above in a file named "*bd4-hadoop.zip*" under "Supporting Material". To use it, uncompress the zip file in your home directory and copy/rename the appropriate client configuration file from "*bd4-hadoop/conf/…*" to "*bd4-hadoop/conf/core-site.xml*". The ".sh" files should already have the executable bit on; if not, then execute "*chmod u+x bd4-hadoop/bin/*.sh*" to enable it. The "*bd4-hadoop/bin/hdfs-dfs.sh*" script replicates the functionality of "*hdfs dfs …*" (or "*hadoop dfs …*"). Similarly, "*bd4-hadoop/bin/mapred-job.sh*" replicates the functionality of "*mapred job …*" (or "*hadoop job …*"). That is, for example, instead of executing "*hdfs dfs -ls /user/bd4-ae1*" you should now execute "*hdfs-dfs.sh -ls /user/bd4-ae1*", instead of "*mapred job -list*" you should use "*mapred-job.sh -list*", etc. Last, "bd4-hadoop/bin/hbase.sh" provides a subset of the functionality of the "hbase" command – namely, "*hbase shell*", "*hbase classpath*" and "*hbase CLASSNAME*".

Note: compile your code with a Java7 compiler, or with the source level compatibility set to Java 7 (or 1.7). Failing to do so may result in errors caused by incompatibility with the JDK version used in the clusters.

When your code is ready for execution, you should bundle it up into a jar file. You can do that through your favourite IDE or by using command-line tools. It is this file that will be submitted to the cluster by the JobClient instance. In order for Hadoop's client-side code to locate said jar file, you should include the full path to it in your job's *Configuration* instance. Please note that "*mapred.jar*" is not the name of your jar file but the predefined name of the relevant configuration variable; you should only change the second argument to this method to point to your jar file.

```
[…]
conf.set("mapred.jar", "file:///home/user/myjobs.jar");
[…]
```

Figure 11 Adding the path to your job's jar file in the job's Configuration

To run your code, just execute it through your IDE of preference (or from command line), making sure to add all jar files under "*bd4-hadoop/lib*", as well as the directory "*bd4-hadoop/conf*", to your classpath. Alternatively, you can use the "bd4-hadoop/java-run.sh" script from the command line.

Last, for your 1st AE, you will be working with a dump of the Wikipedia edit history, downloaded and processed in 2008. The format of the file is defined in the AE1 spec sheet. Said file is already stored on HDFS as "*/user/bd4-ae1/enwiki-20080103-full.txt*". Under the same directory you will also find a cut-down version of this file, named "*/user/bd4-ae1/enwiki-20080103-sample.txt*". You can use the latter as a toy dataset to test your code against; however, keep in mind that the measurements you will supply in the end in your report, should be taken over the full dataset.