ODC Server Guide

For 2ID70 and 2AMD15

This document is a general guide to interacting with the cluster of servers at odc-09.win.tue.nl. This cluster is available for courses such as *2ID70: Data-intensive systems and applications*, and *2AMD15: Big Data Management*. First, the common steps for interacting with the server are described. Secondly, the steps that are specific to each course and/or assignment (milestone) are described.

CONNECTING

Connecting to the cluster will be done through the SSH File Transfer Protocol, or SFTP for short. An account will be created per student group and you will receive the necessary credentials in the first days or weeks of the course. Assuming that your username is `group-n`, you can connect to the cluster using SFTP with the following command:

```
sftp -P 222 group-n@odc-09.win.tue.nl
```

You will be prompted for your password. A connection to the TU\e VPN or network should not be necessary to reach the cluster.

UPLOADING AND DOWNLOADING FILES

Since access to the cluster is restricted to SFTP, the possible interactions with the server are limited. You can upload- and download a file called `file.txt` with the following two commands, respectively:

```
put file.txt
get file.txt
```

For the `put` command to work, the `file.txt` file must be in your current local directory. While connected via SFTP, you can change your local directory using the `lcd` command. For example, you can use the following commands to 1) navigate to the parent directory of your current local directory, 2) navigate into a child directory called dir of your current local directory and 3) navigate via an absolute path to a directory called dir located on your local D: drive (assuming a local Windows system):

```
1. lcd ..
2. lcd dir
3. lcd D:\dir
```

The `cd` command can be used in the same way to navigate remotely (i.e., on the server). Note that the remote system is a Linux-based system.

You can also use the * symbol with the put- and get commands to match any sequence of characters in a file name. For example, you can write “put *.sql” or “get *.log” (without the quotes) to upload- or download all files with .sql or .log extensions, respectively.

You can list the contents of the current remote directory using the `ls` command.

You can create directories inside (sub-directories of) your `home` directory using the `mkdir` command.

Typically, one or more files must be uploaded inside (a sub-directory of) your groups `home` directory. These files are detected automatically by the server, and an attempt is made to execute one or more of them.

UPLOADING FILES FOR A POSTGRESQL SUBMISSION

Submitting SQL files to a PostgreSQL instance on the cluster only requires uploading the necessary `.sql` files into a (sub-directory of) your group's `home` directory. Which `.sql` files are necessary is explained in the Milestone document for the relevant assignment.

No data needs to be uploaded for a PostgreSQL submission.

UPLOADING FILES FOR A SPARK SUBMISSION

Submitting a Spark program to a Spark instance on the cluster can be done in several ways. Firstly, we distinguish between a program written in Java/Scala and one written in Python.

Java/Scala

For a Spark program written in Java or Scala, you can upload a file called `app.jar` or `app.zip`. This file must contain a file called `Main.class` (case-sensitive), or a manifest which explicitly defines another class as the main class. The program will be started from this main class.

Python

For a Spark program written in Python, you can upload a file called `app.zip`. This file must contain a file called `main.py` (case-sensitive), from which the program will be started.

Data

Depending on the assignment (Milestone), it may be necessary to upload your own data to the cluster. Regardless of the language that the program you upload is written in, all necessary data files must be contained in a file called `data.zip` which is uploaded alongside `app.zip` or `app.jar`.

OBTAINING AUTOMATED FEEDBACK

Once a new set of files is detected by the server, it will attempt to execute them whenever the necessary resources are available. Note that the cluster as a whole is a shared resource, and so it may take a while before the necessary resources are available.

For both PostgreSQL and Spark submissions, a log file will be created and written to by the server. This log file will contain automated feedback on the submission in general (e.g., which files were detected, which steps were taken to execute it, did it execute successfully, etc.). The name of this log file will be:

```
submission-x.handler.log
```

where `x` is a submission identifier.

For Spark submissions, a second log file will be created and written to by the server. This log file will contain the output of Spark itself. The name of this log file will be:

```
submission-x.spark-submit.log
```

You can retrieve these files using the `get` command.