Quickstart to installing tools used in TDT4305

(c) Kjetil Nørvåg

We will in this document only cover standalone installations, since usage of (pseudo-)distributed installations will be similar, but installation can be much more difficult (and those that want to try will probably not need this document anyway. :) On Mac, it is more or less similar to Linux.

Important: This Quickstart is just a temporary version and might be extended. Bugs will be fixed, but if you run into problems try first to find out of the problem yourself or ask others (the quickstart is not aimed at solving all issues that might come up during installation, the aim is just to get you started). The forum on It's learning should also be very suitable for getting help from others. As soon as the teaching assistant in the course is ready he will also be available for help.

Java and Python

For all systems, JDK will be needed, and for some also Python. Paths to these installations containing space (as will be default in Windows) can create problems, so you might want to install it on the root of the file system (e.g., C:\Java), or create a separate folder like, e.g., ProgramFiles.

Java download: http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html

Python 2.7 download: https://www.python.org/downloads/

It might be necessary to have Java and Python in the path.

Linux: Add path to PATH variable.

Windows: right-click on "Computer", Properties, Advanced system settings, and then Environment variables, and variable Path.

It might also be necessary to have the variable JAVA_HOME set (for example, in my case it is C:\Java\jdk1.8.0_66), for Windows this is added under environment variables. For Linux, something similar to this (depending on actual path, this one is for Oracle Java on Ubuntu)):

```
export JAVA_HOME=/usr
```

Important: Version numbers in examples might be different from what you have downloaded, so adapt according to what you have downloaded/installed.

Hadoop

For all OS, the first step is to download Hadoop, get the binary release from here: http://hadoop.apache.org/releases.html (for example http://www.eu.apache.org/dist/hadoop/common/hadoop-2.6.3/hadoop-2.6.3.tar.gz)

Unpack the file (tar xfz file.tar.gz from bash, Winzip or similar in Windows). It might be convienient to add the path to the bin directory to the path, typically one creates one variable called HADOOP_HOME with path to Hadoop, and then add $HADOOP_HOME/bin to the path.

Default installation is standalone, you can verify this in the files under $HADOOP_HOME/etc/hadoop. The files core-site.xml, hdfs-site.xml and mapred-site.xml

If not, edit these file, they should have the same contents as the file mapred-site.xml.template (for some distributions, mapred-site.xml does not exist, this is OK).

If you use Windows, you also need the files hadoop.dll and winutils.exe, these files should be placed in the bin directory. For some reason pre-compiled versions of these are not included in the distributions, so you either have to build them yourself (you'll find instructions if you search for it on the web, but can be a bit tricky if you're not used to the tools involved), or you can use a pre-compiled version from somebody that has already done the work. The precompiled versions on It's Learning come from the Hortonworks Hadoop distribution (http://hortonworks.com/) so they should be safe, but as always, use at own risk. :)

To check everything is OK, run

```
$HADOOP_HOME/bin/hadoop
```

As a second test, download the data files (ncdc directory) and the MaxTemperature application, and run

```
$HADOOP_HOME/bin/hadoop jar MaxTemperature.jar ncdc output
```

Keep in mind that the destination directory should not exist before you run it. If everything works, you have now run your (maybe) first MapReduce job! The result is in the output folder, essentially one file for each reducer (which is always one for standalone). If you look at the contents of the file part-r-00000 it should be:

```
1901	317
1902	244
```

In order to make your own MapReduce applications, just compile the java-files in, e.g., Eclipse, and export as runnable jar. You will need some of the Hadoop libraries (jars) during build, they are located under share\hadoop
For the MaxTemperature application, the following files are needed: hadoop-mapreduce-client-core-2.7.1.2.3.4.0-3485 and hadoop-common-2.7.1.2.3.4.0-3485

Other options for installation

Another option which will also give you more tools to try in an easy way, is to either download distributions from, e.g., Cloudera and install it yourself (does not work for Windows), or install VirtualBox from https://www.virtualbox.org/wiki/Downloads and use appliances with the relevant software installed, e.g.:

http://www.oracle.com/technetwork/community/developer-vm/index.html#bdl
http://www.cloudera.com/content/www/en-us/downloads/quickstart_vms/5-5.html

(Note: I had problems getting the one from Cloudera to work last time I tried)

Pig

I have not found any easy way to make Pig work under Windows (at least with the releases I tried with), so for trying out Pig I advise to use VirtualBox. The appliances mention above might require more memory than what is installed in your PC, if that is the case it might be better just downloading the Ubuntu Desktop iso-file and install this on your own VM, and then just use this one and follow the Linux guidelines.

Download Pig (e.g., pig-0.15.0.tar.gz) from here: http://www.eu.apache.org/dist/pig/
and unpack. Start grunt with

```
bin/pig  –x local
```

(The parameter –x local means standalone)

Spark

Quick install and testing of Spark (Scala shell is used in examples, for Python shell to work, Python will have to be installed as well).

Instructions for running Spark on Windows:

1) Download from http://spark.apache.org/downloads.html (Choose most recent release), "Pre-built for Hadoop 2.6 and later")
2) Unpack the downloaded file (You might have to change file extension to ".tar.gz", depending on which zip tool you use)
3) Copy winutil.exe (same as for Hadoop) (winutils.exe can also be found here, or at http://public-repo-1.hortonworks.com/hdp-win-alpha/winutils.exe , as said before, use at own risk. :)
4) set HADOOP_HOME=X where X is the full path to the directory over bin, e.g., HADOOP_HOME=C:\Users\noervaag\Desktop\spark-1.5.1-bin-hadoop2.6
5) (Might be necessary to give all users "full access control" to /tmp)
6) Go to the bin directory, start Spark with spark-shell.cmd or pyspark.cmd

You will get some errors relating to sqlContext when you start spark-shell under windows, this is probably a Hive bug, so as long as you don’t use the sqlContext object there are probably no problems. There are other issues (for example, saveAsTextFile will not work), so if you want to do something more than try easy examples you’ll have to use pyspark (or another OS if wanting to use Scala).

On Linux, and similar on Mac (if not having your own, log into login.stud.ntnu.no):

1) Download Spark as above and unpack ("wget http://www.eu.apache.org/dist/spark/spark-1.6.0/spark-1.6.0-bin-hadoop2.6.tgz " and then tar xfz on the file)
2) Unpack, go to the bin directory of the installation
3) ./spark-shell (use ./pyspark if you want to use the python shell)

Don't care about all the info that on the screen when Spark is starting, if success you will now be at the "scala>" command line. Quick test if everything is OK:

```
scala> val f = sc.textFile("spark-shell")

f: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[1] at textFile at <console>:21

 scala> f.count()

res0: Long = 92

scala>
```

R

Installing R is easy, simply go to the home page of the R project at https://www.r-project.org/ and download and install. R should work without problems on “all platforms”.