GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN
INSTITUT FÜR INFORMATIK

Practical Course on Parallel Computing · **SoSe 2018**

Prof. Yahyapour · Prof. Grabowski · Prof. Quadt

Assignment Sheet 11 · Submission Deadline: 03.07.2018, 3 pm

Organizational Issues

Organizational hints were already given in the lecture and can also be found as slides in StudIP. You can either put your solution in the post box of Johannes Erbel (Institute of Computer Science, 0. Floor) or you can hand it in directly in the tutorials/lectures. Source code and also other solutions can be send via E-Mail to the following adress:
johannes [dot] erbel [at] cs [dot] uni-goettingen [dot] de.
Contact this address if you did not recieve your credentials for the cluster during the lecture.

MapReduce and Hadoop - Hints for the Development

Before starting, please first read the following prerequisites:

- The programming language is Java. **Please use version 8.**
- You have to use a `hadoopuserX` user to log in to the gateway nodes in the cloud to be able to submit jobs to the Hadoop cluster. The groups with `hadoopuser[1-7]` should use `gateway1` (141.5.109.125) and the groups with `hadoopuser[8-15]` should use `gateway2` (141.5.109.124). You can connect to the gateways using an ssh connection:

```
$ ssh hadoopuserX@gatewayIP
```

- The Web interfaces for the Hadoop cluster in the cloud are available at `http://hadoop-master:50070` (NameNode), `http://hadoop-master:8088` (ResourceManager) and `http://hadoop-master:19888` (job history information). Note that some of the links on these webpages link to local IPs in the cloud that can not be resolved from outside the cloud. If you want to be able to resolve them, you can find an explanation on how to configure a proxy at the end of this assignment sheet.
- You have a home directory inside HDFS. When you use the Hadoop command line interface to connect to HDFS, the default working directory is this home directory. The following command lists the contents of your home directory (on `gateway[1-2]`):

```
$ hdfs dfs -ls
```

  Your home directory is empty by default.
- For the development, use either the Hadoop cluster in the cloud or a pseudo-distributed Hadoop cluster on your own machine. A tutorial for installing Hadoop on your local machine can be found at: `http://hadoop.apache.org/docs/current/`

`hadoop-project-dist/hadoop-common/SingleCluster.html`. We recommend to use the existing infrastructure.

- Please note that your data on the gateway nodes and also inside HDFS is **not** backed up by us automatically and we give no warranty for data loss. You have to maintain a backup yourself!

Assignment 1 – Scheduling in YARN (*2 Points*)

Do some research and answer the following questions:

a) Which schedulers are implemented in the YARN framework? Use the ResoureManager Web Interface to determine which of them is used in our cluster!

b) Explain the basic functionality of the *FairScheduler*!

Assignment 2 – Analyzing Twitter data (*5 Points*)

The social graph formed by Twitter users is different to the one formed by other social networks: The relations between Twitter users are unidirectional: that means if user A follows user B, user B does not necessarily follow user A. In this assignment, we are going to use real Twitter data that was crawled and published by Kwak et al. in 2010 [1]. This data consists of a single file where each line represents a single "is-followed by"-relationship [1]:

```
TWITTERID \t FOLLOWER \n
```

Example:

```
12  13
12  14
12  15
16  17
```

Users 13, 14 and 15 are followers of user 12. User 17 is a follower of user 16.
We are going to use a subset of the data collected by Kwak et al. The data consists of 50 million "is-followed by"-relationships and is available in HDFS at `/twitter/twitterSlice`.
Use your newly acquired MapReduce skills to implement MapReduce analysis jobs, that answer the following questions!

a) Which Twitter ID has the most followers and how many people follow this Twitter ID?

b) Which Twitter ID follows the most other Twitter IDs?

c) Are there any bidirectional following relationships, that means are there any pairs of users A and B, such that A is a follower of B and B is a follower of A. If there are such pairs, how many of them exist in the given data?

Implement each of the tasks as a separate analysis job! You are allowed to do simple post-processing of your output data via command line tools if necessary. Before you start, think about suitable key/value pairs!

[1] `http://an.kaist.ac.kr/traces/WWW2010.html`

Assignment 3 – Friend Finder (*4 Points*)

In this assignment, we use MapReduce to implement the “Friend Finder” Algorithm discussed in the lecture. The data for this assignment consists of the social graph formed by 63,731 Facebook users with a total of 817,035 friendships taken from the KONECT project [2]. The data format is similar to the one of Assignment 2, except that here each line represents a Facebook friendship and hence a bidirectional relationship:

```
FACEBOOKUSER \t FACEBOOKUSER \n
```

Implement the “Friend Finder” Algorithm discussed in Lecture 10! Your algorithm might need to do more than one MapReduce iteration. You find the input data inside HDFS at `/facebook/fbLinks`. Which pair of Facebook friends has the most friends in common and how many friends do they have in common? What is the average number of common friends between two Facebook friends in the given data?

Assignment 4 – Hadoop Streaming (*2 Points*)

Hadoop Streaming enables users to implement Hadoop jobs in other languages than Java. Hereby, the map and reduce functions are implemented as single executables that communicate over standard UNIX input and output streams. The map and reduce jobs read from standard input line by line, whereby each line has the following format:

```
KEY\tVALUE
```

Use your favorite programming language to reimplement the WordCount example using Hadoop Streaming and test it on `ulysses.txt` from Assignment Sheet 4! You can test your program locally using UNIX pipes:

```
$ cat ulysses.txt | <mapexecutable> | sort -k1,1 | <reduceexecutable>
```

After you verified that it is running correctly, submit it as a job on the Hadoop cluster:

```
$ hadoop jar /opt/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.9.1.jar \
-mapper <mapexecutable> -reducer <reduceexecutable> \
-file <path to mapexecutable> -file <path to reduceexecutable> \
-input <input dir in HDFS> -output <output dir in HDFS>
```

Hereby the option `-file` let you specify files that are copied to a cache in the cluster before the the functions are executed.

Please note: Since your executables are executed on the worker nodes in the cluster, they can not use external dependencies that are not installed. The best way to check if your compiled executables run on the cluster nodes is to make a test run (see above) locally on one of the gateway nodes.

Assignment 5 – Benefits and Limitations of MapReduce (*2 Points*)

You have now spent some time working with MapReduce and Hadoop and should be able to answer the following questions:

a) Why do you think MapReduce and Hadoop were such a great success?

b) What are the limitations of MapReduce? Are there problem classes that can not be implemented with help of this computation model?

[2] `http://konect.uni-koblenz.de/networks/facebook-wosn-links`

Proxy Settings

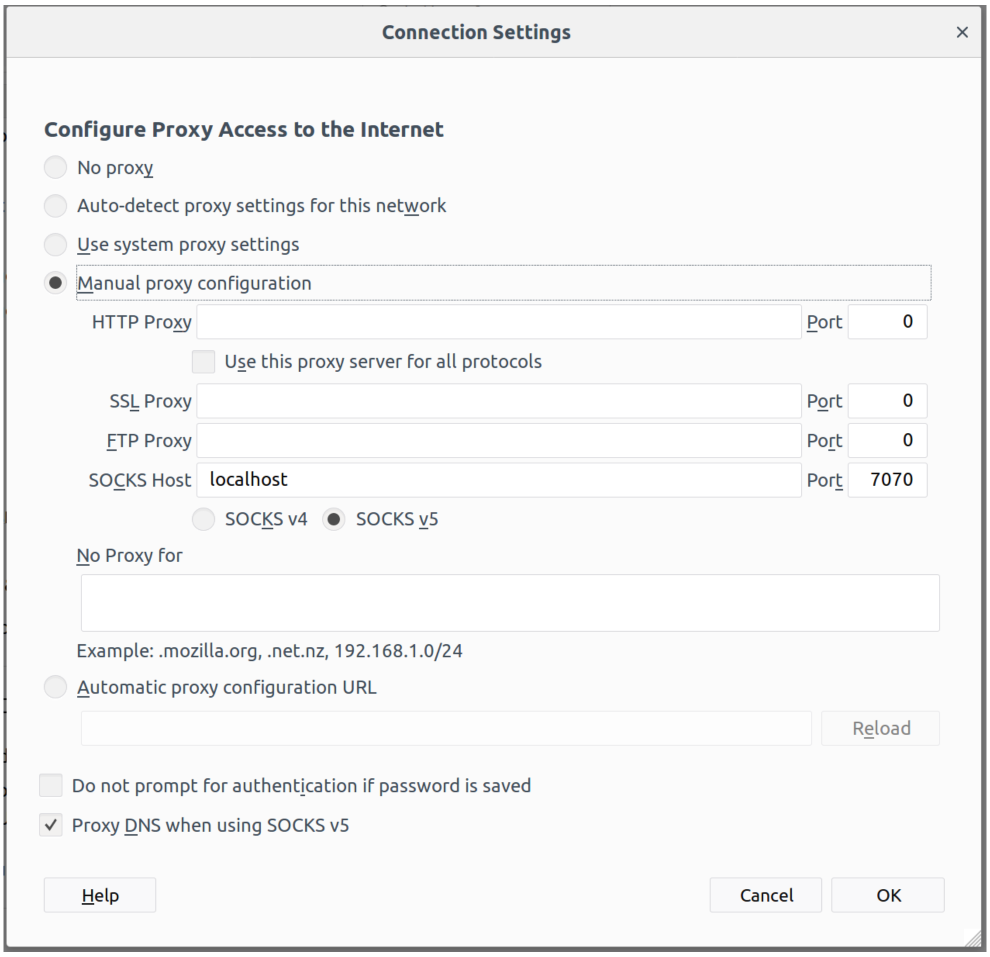
To connect to the hadoop-master web frontent you need to setup an ssh tunnel and specific proxy settings to be used by your browser. To setup the ssh tunnel the following command can be used:

```
$ ssh -f -N -D 7070 hadoopuserX@gatewayIP
```

To kill the tunnel you have to identify the id of the process and kill it:

```
$ ps -ax | grep ssh (to get the pid)
$ kill -9 pid
```

After the tunnel is created you need to adjust your browser settings. For example, in Firefox you can open the preferencees tab and search for proxy. Open the Network Proxy settings and adjust them as showed in the following Figure:



Connection Settings

Configure Proxy Access to the Internet

○ No proxy
○ Auto-detect proxy settings for this network
○ Use system proxy settings
● Manual proxy configuration

HTTP Proxy: Port: 0
☐ Use this proxy server for all protocols
SSL Proxy: Port: 0
FTP Proxy: Port: 0
SOCKS Host: localhost Port: 7070
○ SOCKS v4 ● SOCKS v5

No Proxy for:

Example: .mozilla.org, .net.nz, 192.168.1.0/24

○ Automatic proxy configuration URL
Reload

☐ Do not prompt for authentication if password is saved
✓ Proxy DNS when using SOCKS v5

Help Cancel OK

References

[1] Kwak, Haewoon, et al. “What is Twitter, a social network or a news media?” Proceedings of the 19th international conference on World Wide Web. ACM, 2010.