



Assignment Sheet 10 · Submission Deadline: 26.06.2018, 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 – Short Answer Questions (4 Points)

Do some research and answer the following questions:

- Shortly sketch and explain the three different phases of the MapReduce workflow!
- What is the combiner phase and why is it optional?
- Explain the purpose of the *NameNode*, *DataNode*, *ResourceManager*, and *NodeManager* daemons in HDFS and YARN!
- Explain the meaning of the *replication factor* and the *blocksize* in the Hadoop Distributed File System (HDFS). What is controlled by the number of *slots*?

Assignment 2 – Word Count (5 Points)

In this assignment, we learn how to submit a job to the Hadoop cluster and how to implement basic analysis jobs in the Hadoop framework. Execute the following steps! Document the commands you used and their results!

- Download and extract the tar archive `assignment10.tar` from Stud.IP (You can use `scp` to copy it to your home directory on the gateway) The file `wc.jar` contains the compiled word count example discussed in the lecture.
- Count the appearance of words in the file `ulysses.txt`! For this, create a sub-directory in your HDFS home directory and upload `ulysses.txt` to that directory by using the Hadoop command line interface.
- Submit a word count job to the Hadoop cluster. The job can be submitted by

```
$ hadoop jar <path to example jar> WordCount <input dir in HDFS> \
    <output dir in HDFS>
```

Pitfall: Hadoop will create the output directory for you and will throw an exception in case it already exists.

- Use either the Hadoop command line interface or the web interface for HDFS to analyze the output of the job. The output is available in the output directory you specified after the job is finished. How many times does the word *Master* (case-sensitive) appear in the analyzed text?
- The java code for the word count example is given in `WordCountExample/WordCount.java`. Modify this code to implement an analysis job that counts single characters instead of whole words and name it `CharacterCount.java`. You need to create a jar file that contains your analysis code and the required dependencies to be able to submit your job to the cluster.

You can compile your code with

```
$ javac -classpath hadoop-common-2.9.1.jar:\
hadoop-mapreduce-client-core-2.9.1.jar:\
hadoop-annotations-2.9.1.jar:\
commons-cli-1.2.jar CharacterCount.java
```

Now you create a jar file out of the compiled code with

```
$ jar cf cc.jar CharacterCount*.class
```

Submit a character count job to the Hadoop cluster that counts the characters in `ulysses.txt`. Use the outcome of the analysis to determine the relative frequency of the characters in the text. Is the outcome of your analysis typical for an english text?

Assignment 3 – Identify anagrams with MapReduce (6 Points)

“An anagram is a type of word play, the result of rearranging the letters of a word or phrase to produce a new word or phrase, using all the original letters exactly once; for example *anagram* can be rearranged into *nag-a-ram*.”¹.

Implement a Hadoop MapReduce job that identifies the anagrams (only single words) in a given text corpus! The output should be a file that contains a list of anagrams per line e.g.:

```
<key1> orchestra , carthorse , ...
<key2> masters , streams , ...
...
```

You can use the word count code as a basis for your implementation. Your programs should work case-insensitive. That means that e.g. *DoctorWho* is an anagram of *TorchWood* in our definition. Remove any duplicate anagrams in your output values. Before you start programming, identify suitable key/value pairs.

Test your program on the file `ulysses.txt`. After you have verified that your program finished successfully, submit a second analysis that uses the HDFS directory `/books` as input for your job. It contains 250 books in plain text. Use the ResourceManager Web interface to determine the number of map and reduce jobs that are created for your job. Is the result as you expected it to be? If not, explain why! Which anagrams are found by your program for the word “respect”?

Pitfall: The books used for this analysis were scanned and transformed into text files. Thus, a lot of strange “words” occur in your output file.

¹source: <https://en.wikipedia.org/wiki/Anagram> fetched on 15/06/2018

Proxy Settings

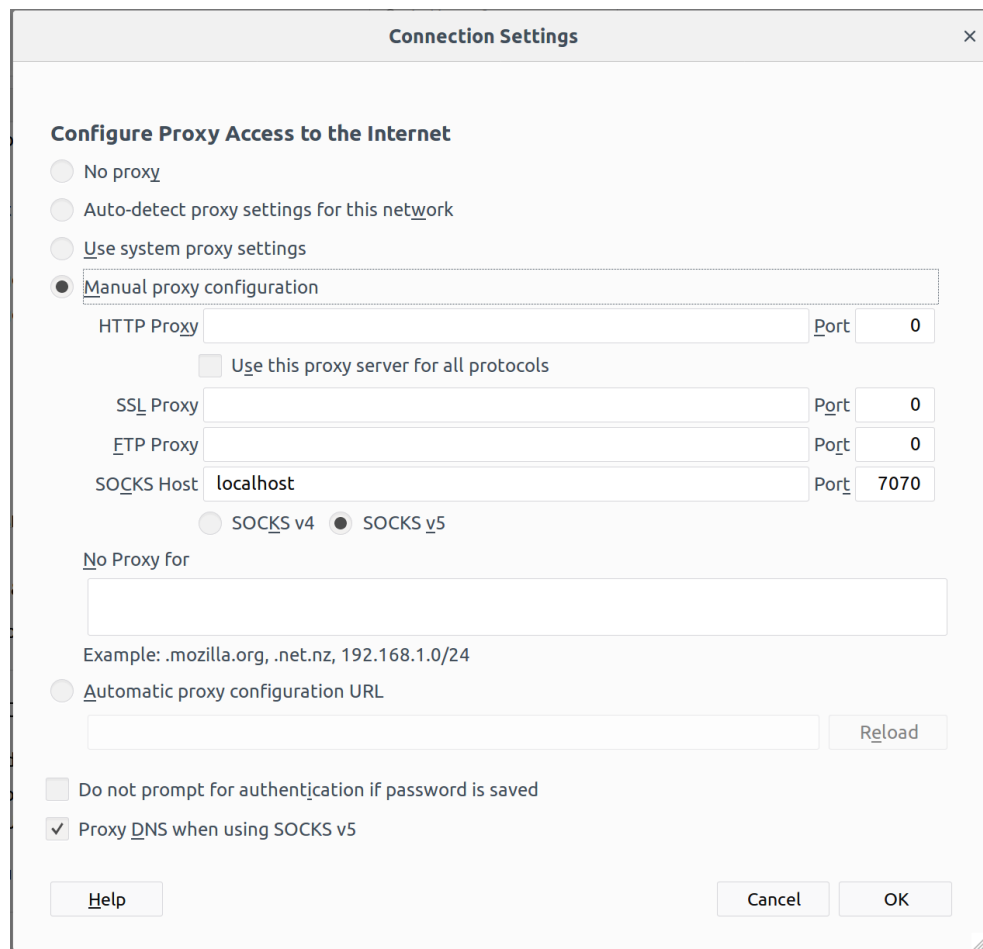
To connect to the hadoop-master web frontend you need to setup an ssh tunnel and specific proxy settings to be used by your browser. To setup the ssh tunnel the following command can be used:

```
$ ssh -f -N -i -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 preferences tab and search for proxy. Open the Network Proxy settings and adjust them as showed in the following Figure:



The screenshot shows the 'Connection Settings' dialog box in Firefox. The 'Configure Proxy Access to the Internet' section has four radio buttons: 'No proxy', 'Auto-detect proxy settings for this network', 'Use system proxy settings', and 'Manual proxy configuration'. The 'Manual proxy configuration' option is selected. Below this, there are input fields for 'HTTP Proxy' and 'Port' (set to 0), a checkbox for 'Use this proxy server for all protocols', 'SSL Proxy' and 'Port' (set to 0), 'FTP Proxy' and 'Port' (set to 0), 'SOCKS Host' (set to localhost) and 'Port' (set to 7070). There are also radio buttons for 'SOCKS v4' and 'SOCKS v5', with 'SOCKS v5' selected. Below these, there is a 'No Proxy for' section with a text input field and an example: '.mozilla.org, .net.nz, 192.168.1.0/24'. There is also an 'Automatic proxy configuration URL' section with a text input field and a 'Reload' button. At the bottom, there are checkboxes for 'Do not prompt for authentication if password is saved' and 'Proxy DNS when using SOCKS v5', with the latter checked. The dialog box has 'Help', 'Cancel', and 'OK' buttons at the bottom.