Modern Database Concepts
Prof. Dr. Florian Heinz
florian.heinz@oth-regensburg.de
Exercise Sheet 7

OSTBAYERISCHE TECHNISCHE HOCHSCHULE REGENSBURG

Exercise 1: Local install of Hadoop

On the CIP pool hosts, boot Ubuntu, start a shell and change into the directory **/data/temp** (your home directory is on a network share and does not provide enough space)

Download the Hadoop tar.gz-archive from the Apache Hadoop homepage to /data/temp

```
$ wget https://dlcdn.apache.org/hadoop/common/hadoop-3.2.3/hadoop-3.2.3.tar.gz
```

Untar the archive with the following command:

```
$ tar xvfz hadoop-3.2.3.tar.gz
```

Download the config files **hdfs-site.xml** and **core-site.xml** from GRIPS and put these into the subdirectory etc/hadoop/ of the unpacked archive

Edit the file **etc/hadoop/hadoop-env.sh** in the hadoop directory and add the following line to specify the base of your local Java installation:

```
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
```

Set up passwordless ssh to localhost:

```
$ ssh-keygen
# press several times enter here
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

Initialize the local namenode by issuing the following command:

```
$ bin/hadoop namenode -format
```

Start the local name- and datanode with the following script:

```
$ sbin/start-dfs.sh
```

Verify, that everything is up and running with:

```
$ bin/hdfs dfs -df
```

This should give an output similar to:

```
Filesystem                    Size     Used     Available  Use%
hdfs://localhost:9000  626102980608  1155072  505217871872    0%
```

After that, the local hdfs is up and running.

Modern Database Concepts
Prof. Dr. Florian Heinz
florian.heinz@oth-regensburg.de
Exercise Sheet 7

OSTBAYERISCHE TECHNISCHE HOCHSCHULE REGENSBURG

Exercise 2: Upload book files

Download and extract (with "unzip") the file books.zip from GRIPS **into the hadoop directory**. This file contains 10 books from Project Gutenberg. The filename is the book id.

Create a directory /ex2/books on the hdfs:

```
$ bin/hdfs dfs -mkdir -p /ex2/books
```

Upload the downloaded book files:

```
$ bin/hdfs dfs -put books/* /ex2/books
```

Check if the upload worked:

```
$ bin/hdfs dfs -ls /ex2/books
```

Find out how to display the contents of book id 2032 from the hdfs and determine its title. Sounds familiar?

Exercise 3: Create a GIN on the book files

Create a directory **gin** in the hadoop directory and change into it

Download the Java files **IntSetWritable.java** and **GenerateGIN.java** from GRIPS into this new directory

Open the source files and try to understand what they do

In the **gin** directory compile the Java source files to class files:

```
$ javac -cp $(../bin/hadoop classpath) GenerateGIN.java IntSetWritable.java
```

After that, create a jar file from the resulting class files:

```
$ jar cf gin.jar *.class
```

Now run the MapReduce process on the book files. Closely look at the output and try to understand, what happens (logging of the map, combine and reduce processes)

```
$ ../bin/hadoop jar gin.jar GenerateGIN /ex2/books /ex2/gin
```

Download the resulting files from /ex2/gin in the hdfs and have a look at it.

OSTBAYERISCHE
TECHNISCHE HOCHSCHULE
REGENSBURG

Exercise 4: Modify the Map process

Change the map function in the java file to improve word preprocessing. For example, try to omit stop words like **i**, **and**, **or** and so on. Create an array with the stop words and do not emit a key/value pair from the map process if such a word is seen. You can also try to determine word stems, remove plural forms, etc., whatever comes to your mind.

Recompile and re-run the Map-Reduce process. Mind, that you have to specify an output directory that does not exist. Either delete the old one or choose a different name