

# Solution – Exercise I

Hadoop, HDFS, YARN, MapReduce Example



# Solution

## Prerequisites:

- install Ubuntu 18.04
- Install Java JDK 1.8.0
- Create Hadoop user
- Install and Setup SSH (*public/private key authentication, authorized\_keys, ...*)
- Install and Configure Hadoop 3.1.3 (*pseudo-distributed mode*)
- Start HDFS and YARN
- Clone Git Repo:

```
git clone https://github.com/marcelmittelstaedt/BigData.git
```



# Solution

## Exercise 2:

1. Copy sample file from GIT repo to **HDFS** user directory:

```
hadoop fs -put BigData/exercises/winter_semester_2022-2023/01_hadoop/sample_data/Faust_1.txt /user/hadoop/Faust_1.txt
```

2. Use and run default MapReduce Jar (*hadoop-mapreduce-examples-3.1.2.jar*) to calculate **wordcount** for text file „*Faust\_1.txt*“.

```
hadoop jar hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.1.2.jar wordcount /user/hadoop/Faust_1.txt /user/hadoop/Faust_1_Output
```

```
[...]
2022-02-14 15:36:46,163 INFO mapreduce.Job: The url to track the job: http://big-data.c.dhbw-254309.internal:8088/proxy/application_1613316821880_0001/
2022-02-14 15:36:46,164 INFO mapreduce.Job: Running job: job_1613316821880_0001
2022-02-14 15:36:54,318 INFO mapreduce.Job: Job job_1613316821880_0001 running in uber mode : false
2022-02-14 15:36:54,319 INFO mapreduce.Job: map 0% reduce 0%
2022-02-14 15:37:00,404 INFO mapreduce.Job: map 100% reduce 0%
2022-02-14 15:37:05,448 INFO mapreduce.Job: map 100% reduce 100%
2022-02-14 15:37:06,462 INFO mapreduce.Job: Job job_1613316821880_0001 completed successfully
[...]
```

## Exercise 2:

3. Take a look at Ressource Manager for Job Execution  
(<http://XXX.XXX.XXX.XXX:8088/cluster/apps/RUNNING>):

**hadoop**

Cluster Metrics

Apps Submitted	0	Apps Pending	1	Apps Running	2	Apps Completed	1	Containers Running	2 GB	Memory Used	16 GB	Memory Total	0 B	Memory Reserved	1	VCores Used	8	VCores Total	0	VCores Reserved
----------------	---	--------------	---	--------------	---	----------------	---	--------------------	------	-------------	-------	--------------	-----	-----------------	---	-------------	---	--------------	---	-----------------

Cluster Nodes Metrics

Active Nodes	0	Decommissioning Nodes	0	Decommissioned Nodes	0	Lost Nodes	0	Unhealthy Nodes	0	Rebooted Nodes	0	Shutdown Nodes
--------------	---	-----------------------	---	----------------------	---	------------	---	-----------------	---	----------------	---	----------------

Scheduler Metrics

Scheduler Type	Capacity Scheduler	Scheduling Resource Type	[memory-mb (unit-M), vcores]	Minimum Allocation	<memory:1024, vCores:1>	Maximum Allocation	<memory:8192, vCores:4>	Maximum Cluster Application Priority	0
----------------	--------------------	--------------------------	------------------------------	--------------------	-------------------------	--------------------	-------------------------	--------------------------------------	---

Show: 20 entries

ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCoers	Allocated Memory MB	Reserved CPU VCoers	Reserved Memory MB	% of Queue	% of Cluster	Progress	Tracking UI	Blacklisted Nodes
application_1696087941529_0003	hadoop	word count	MAPREDUCE	default	0	Sat Sep 30 17:51:12 +0200 2023	N/A	RUNNING	UNDEFINED	1	1	2048	0	0	12.5	12.5	<div></div>	ApplicationMaster	0

Showing 1 to 1 of 1 entries

First Previous 1 Next Last

# Solution

## Exercise 2:

4. a) Copy MapReduce output file back to ubuntu local filesystem (using bash):

```
hadoop fs -get /user/hadoop/Faust_1_Output/part-r-00000 Faust_1_Output.csv
```

```
head -10 Faust_1_Output.csv
```

```
Allein 1
```

```
Alles 1
```

```
Als 1
```

```
Der 1
```

```
Die 2
```

```
Er 2
```

```
Ich 4
```

```
Im 1
```

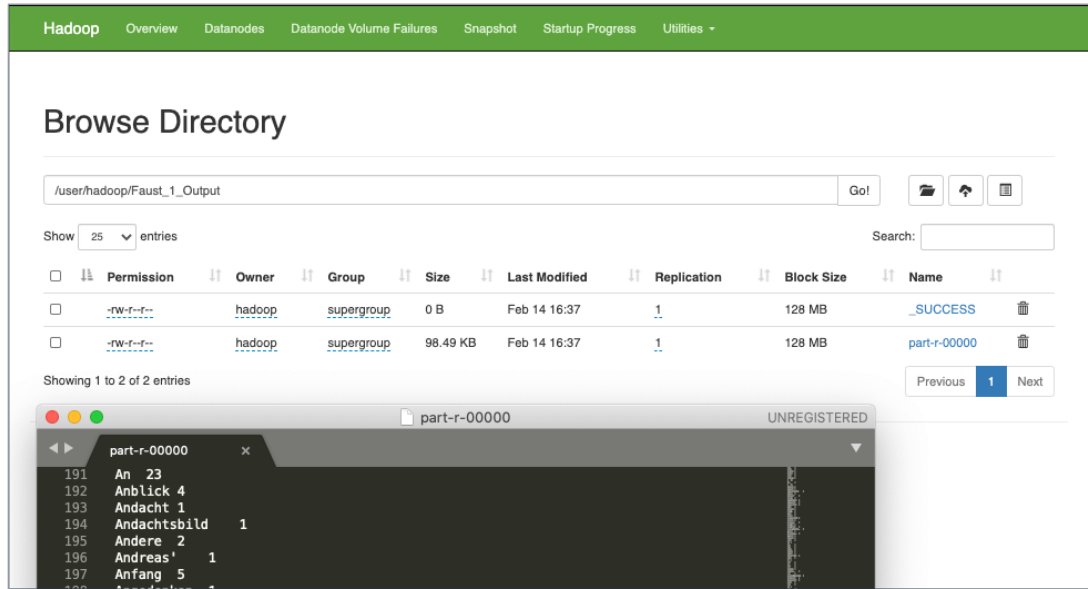
```
Mein 1
```

```
Nur 1
```

# Solution

## Exercise 2:

4. **b)** Copy MapReduce output file back to ubuntu local filesystem (using Web Filebrowser):



The screenshot shows the Hadoop Web File Browser interface. The top navigation bar includes links for Hadoop, Overview, Datanodes, Datanode Volume Failures, Snapshot, Startup Progress, and Utilities. The main heading is "Browse Directory". Below it, the path "/user/hadoop/Faust\_1\_Output" is entered in the search bar. The "Show" dropdown is set to "25" entries. A search bar is also present. The table below lists the files in the directory:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	hadoop	supergroup	0 B	Feb 14 16:37	1	128 MB	<a href="#">_SUCCESS</a>
-rw-r--r--	hadoop	supergroup	98.49 KB	Feb 14 16:37	1	128 MB	<a href="#">part-r-00000</a>

Showing 1 to 2 of 2 entries

Previous 1 Next

A file explorer window is open showing the contents of the file "part-r-00000". The file is a text file with lyrics:

```
191 An 23
192 Anblick 4
193 Andacht 1
194 Andachtsbild 1
195 Andere 2
196 Andreas' 1
197 Anfang 5
```

# Exercise 3

## MapReduce Examples within *hadoop-mapreduce-examples-3.1.1.jar*:

<b>aggregatewordcount:</b>	An Aggregate based mapreduce program that counts the words in the input files.
<b>aggregatewordhist:</b>	An Aggregate based mapreduce program that computes the histogram of the words in the input files.
<b>bbp:</b>	A mapreduce program that uses Bailey-Borwein-Plouffe to compute exact digits of Pi.
<b>dbcount:</b>	An example job that counts the pageview logs stored in a database.
<b>distbbp:</b>	A mapreduce program that uses a BBP-type formula to compute exact bits of Pi.
<b>grep:</b>	A mapreduce program that counts the matches of a regex in the input.
<b>join:</b>	A job that performs a join over sorted, equally partitioned datasets.
<b>multifilewc:</b>	A job that counts words from several files.
<b>pentomino:</b>	A mapreduce tile laying program to find solutions to pentomino problems.
<b>pi:</b>	A mapreduce program that estimates Pi using a quasi-Monte Carlo method.
<b>randomtextwriter:</b>	A mapreduce program that writes 10 GB of random textual data per node.
<b>randomwriter:</b>	A mapreduce program that writes 10 GB of random data per node.
<b>secondarysort:</b>	An example defining a secondary sort to the reduce phase.
<b>sort:</b>	A mapreduce program that sorts the data written by the random writer.
<b>sudoku:</b>	A sudoku solver.
<b>teragen:</b>	Generate data for the terasort.
<b>terasort:</b>	Run the terasort.
<b>teravalidate:</b>	Checking results of terasort.
<b>wordcount:</b>	A mapreduce program that counts the words in the input files.
<b>wordmean:</b>	A mapreduce program that counts the average length of the words in the input files.
<b>wordmedian:</b>	A mapreduce program that counts the median length of the words in the input files.
<b>wordstandarddeviation:</b>	A mapreduce program that counts the standard deviation of the length of the words in the input files.

# Solution

## Exercise 3:

1. Copy sample file from GIT repo to **HDFS** user directory:

```
hadoop fs -put BigData/exercises/winter_semester_2022-2023/01_hadoop/sample_data/Faust_1.txt /user/hadoop/Faust_1.txt
```

2. Use and run default MapReduce Jar (*hadoop-mapreduce-examples-3.1.2.jar*) to **grep** for string „Faust“ in text file „*Faust\_1.txt*“ and count appearances of string.

```
hadoop jar hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.1.2.jar grep /user/hadoop/Faust_1.txt /user/hadoop/Faust_1_Count_Output 'Faust'
```


```
[...]  
2022-10-12 16:44:16,517 INFO mapreduce.Job: Running job: job_1570893575375_0008  
2022-10-12 16:44:27,637 INFO mapreduce.Job: Job job_1570893575375_0008 running in uber mode : false  
2022-10-12 16:44:27,638 INFO mapreduce.Job: map 0% reduce 0%  
2022-10-12 16:44:31,678 INFO mapreduce.Job: map 100% reduce 0%  
2022-10-12 16:44:36,717 INFO mapreduce.Job: map 100% reduce 100%  
2022-10-12 16:44:37,735 INFO mapreduce.Job: Job job_1570893575375_0008 completed successfully  
[...]
```



## Exercise 3:

3. Take a look at Ressource Manager for Job Execution

(<http://XXX.XXX.XXX.XXX:8088/cluster/apps/RUNNING>):



### RUNNING Applications

Logged in as: dr.who

Cluster

About Nodes Node Labels ApplicationsNEW SUBMITTEDACCEPTEDRUNNINGFINISHEDFAILEDKILLED Scheduler

Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved
2	0	1	1	1	2 GB	16 GB	0 B	1	8	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown Nodes
1	0	0	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

Show 20 entries

ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCoers	Allocated Memory MB	Reserved CPU VCoers	Reserved Memory MB	% of Queue	% of Cluster	Progress	Tracking UI	Blacklisted Nodes
application_1613316821880_0002	hadoop	grep-search	MAPREDUCE	default	0	Sun Feb 14 16:53:04 +0100 2021	N/A	RUNNING	UNDEFINED	1	1	2048	0	0	12.5	12.5	<div></div>	ApplicationMaster	0

Showing 1 to 1 of 1 entries

# Solution

## Exercise 2:

4. a) Copy MapReduce output file back to ubuntu local filesystem (using bash):

```
hadoop fs -get /user/hadoop/Faust_1_Count_Output/part-r-00000 Faust_1_Count_Output.csv
```

```
cat Faust_1_Count_Output.csv
```

```
50          Faust
```

# Solution

## Exercise 2:

4. **b)** Copy MapReduce output file back to ubuntu local filesystem (using Web Filebrowser):

The screenshot shows the Hadoop Web File Browser interface. The top navigation bar includes links for Hadoop, Overview, Datanodes, Datanode Volume Failures, Snapshot, Startup Progress, and Utilities. The main heading is "Browse Directory". Below it, the current directory path is "/user/hadoop/Faust\_1\_Count\_Output". A search bar is present on the right. The table displays the following entries:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	hadoop	supergroup	0 B	Feb 14 16:53	1	128 MB	._SUCCESS
-rw-r--r--	hadoop	supergroup	9 B	Feb 14 16:53	1	128 MB	part-r-00000

Showing 1 to 2 of 2 entries

Below the table, a terminal window is open, showing the command "part-r-00000" and its output:

```
part-r-00000
1 Faust
2
```