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*“.

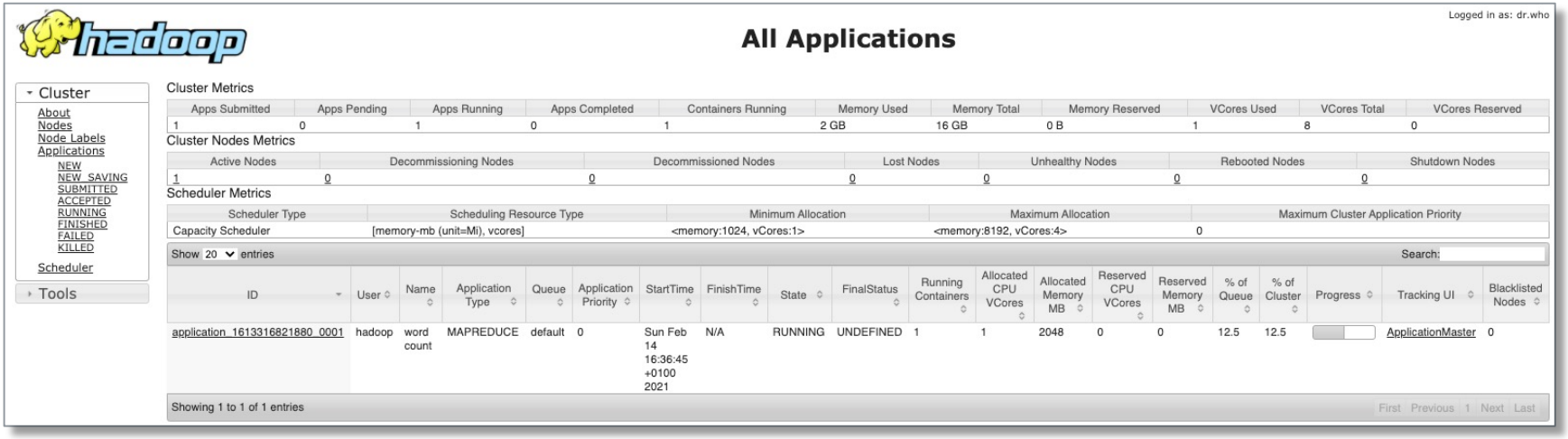
```
hadoop jar hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.1.2.jar wordcount /user/hadoop/Faust_1.txt /user
/hadoop/Faust_1_Output

[…]
2021-02-14 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/
2021-02-14 15:36:46,164 INFO mapreduce.Job: Running job: job_1613316821880_0001
2021-02-14 15:36:54,318 INFO mapreduce.Job: Job job_1613316821880_0001 running in uber mode : false
2021-02-14 15:36:54,319 INFO mapreduce.Job:  map 0% reduce 0%
2021-02-14 15:37:00,404 INFO mapreduce.Job:  map 100% reduce 0%
2021-02-14 15:37:05,448 INFO mapreduce.Job:  map 100% reduce 100%
2021-02-14 15:37:06,462 INFO mapreduce.Job: Job job_1613316821880_0001 completed successfully
[…]
```

Solution

Exercise 2:

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



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
```

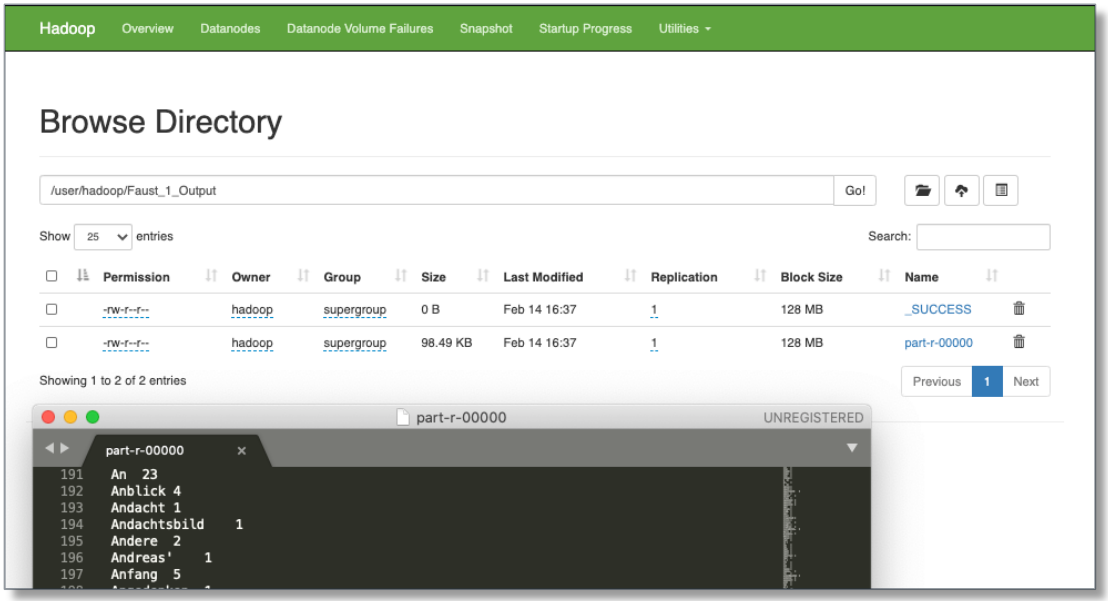
```
head -10 Faust_1_Output.csv

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):



Exercise 3

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

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

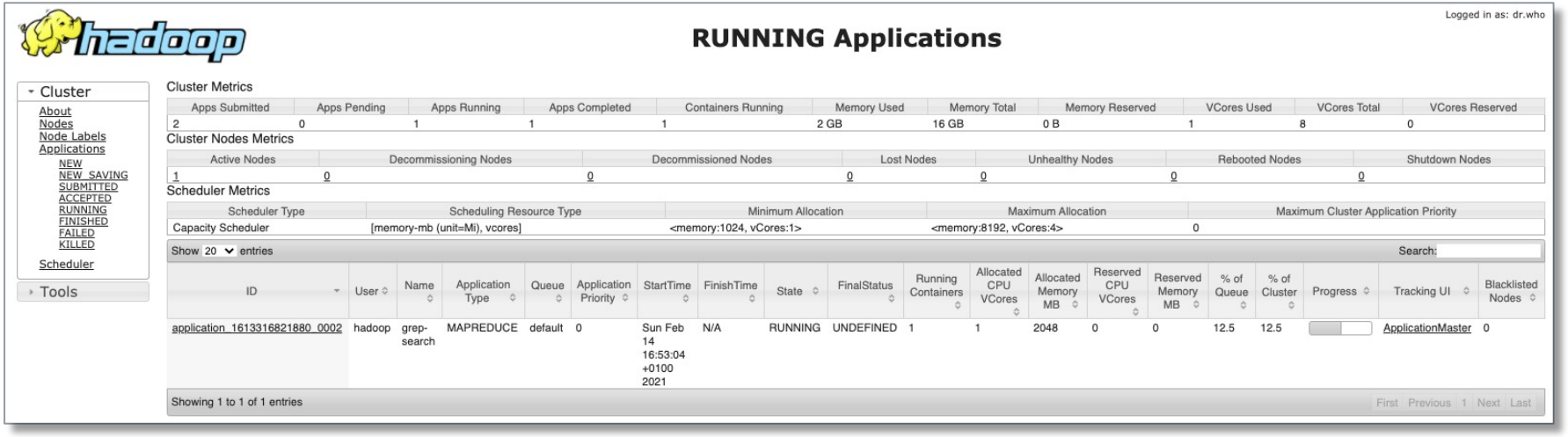
```
hadoop jar hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.1.2.jar grep /user/hadoop/Faust_1.txt /user/hadoop/Faust_1_Count_Output 'Faust'

[…]
2019-10-12 16:44:16,517 INFO mapreduce.Job: Running job: job_1570893575375_0008
2019-10-12 16:44:27,637 INFO mapreduce.Job: Job job_1570893575375_0008 running in uber mode : false
2019-10-12 16:44:27,638 INFO mapreduce.Job:  map 0% reduce 0%
2019-10-12 16:44:31,678 INFO mapreduce.Job:  map 100% reduce 0%
2019-10-12 16:44:36,717 INFO mapreduce.Job:  map 100% reduce 100%
2019-10-12 16:44:37,735 INFO mapreduce.Job: Job job_1570893575375_0008 completed successfully
[…]
```

Solution

Exercise 3:

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



hadoop

Logged in as: dr.who

RUNNING Applications

Cluster
- About
- Nodes
- Node Labels
- Applications
  - NEW
  - NEW_SAVING
  - SUBMITTED
  - ACCEPTED
  - RUNNING
  - FINISHED
  - FAILED
  - KILLED
- 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 Search:

ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	Reserved CPU VCores	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		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_Count_Output/part-r-00000 Faust_1_Count_O
utput.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):