Solution – Exercise 01

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/01_hadoop/sample_data/Faust_1.txt
/user/hadoop/Faust_1.txt
```

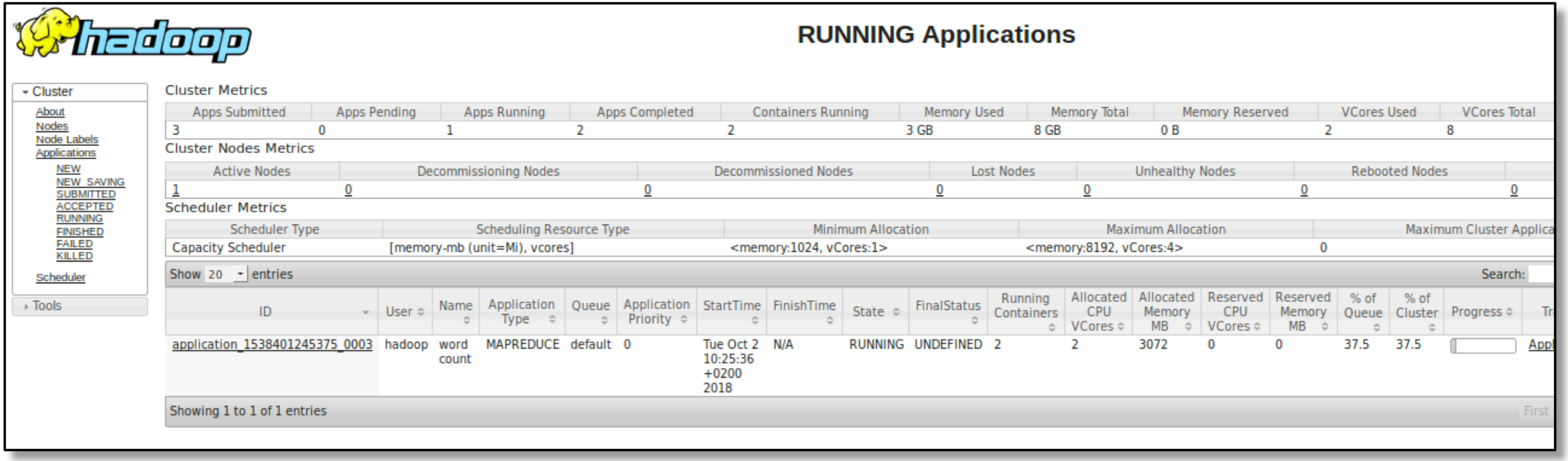
2. Use and run default MapReduce Jar (*hadoop-mapreduce-examples-3.1.1.jar*) to calculate **wordcount** for text file „*Faust_1.txt*“.

```
hadoop jar hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.1.1.jar
wordcount /user/hadoop/Faust_1.txt /user/hadoop/Faust_1_Output
[…]
2018-10-02 10:25:36,818 INFO mapreduce.Job: Running job: job_1538401245375_0003
2018-10-02 10:25:48,156 INFO mapreduce.Job:  map 0% reduce 0%
2018-10-02 10:25:55,276 INFO mapreduce.Job:  map 100% reduce 0%
2018-10-02 10:26:01,339 INFO mapreduce.Job:  map 100% reduce 100%
2018-10-02 10:26:02,360 INFO mapreduce.Job: Job job_1538401245375_0003 completed successfully
[…]
```

Solution

Exercise 2:

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



RUNNING Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total
3	0	1	2	2	3 GB	8 GB	0 B	2	8

Cluster Nodes Metrics

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

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Applica
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	Tr
application_1538401245375_0003	hadoop	word count	MAPREDUCE	default	0	Tue Oct 2 10:25:36 +0200 2018	N/A	RUNNING	UNDEFINED	2	2	3072	0	0	37.5	37.5		Appl

Showing 1 to 1 of 1 entries

First

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
```

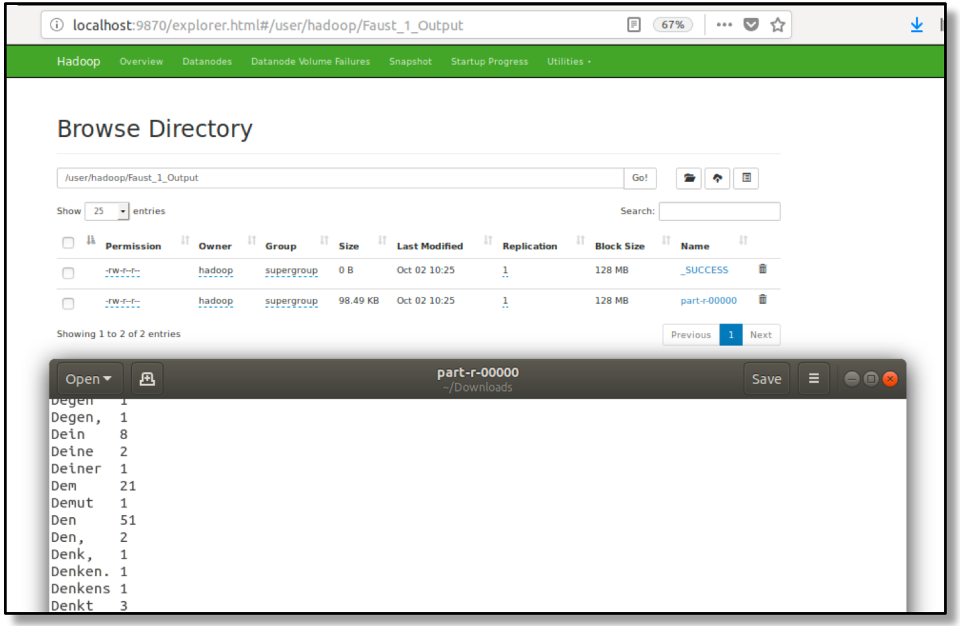
```
shuf -n 10 Faust_1_Output.csv

Phantasie,          1
unwanden.  1
winden,    1
Offenbarung,        2
Undene!    1
Winternächte        1
derweil    1
wiederholten        1
tun        3
Gestalten.          1
```

Solution

Exercise 2:

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



Browse Directory

/user/hadoop/Faust_1_Output

Show 25 entries

Search:

	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
□	-rw-r--r--	hadoop	supergroup	0 B	Oct 02 10:25	1	128 MB	_SUCCESS
□	-rw-r--r--	hadoop	supergroup	98.49 KB	Oct 02 10:25	1	128 MB	part-r-00000

Showing 1 to 2 of 2 entries

Previous 1 Next

```
Degen    1
Degen,   1
Dein     8
Deine    2
Deiner   1
Dem      21
Demut    1
Den      51
Den,     2
Denk,    1
Denken.  1
Denkens  1
Denkt    3
```

Solution

Exercise 3:

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

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

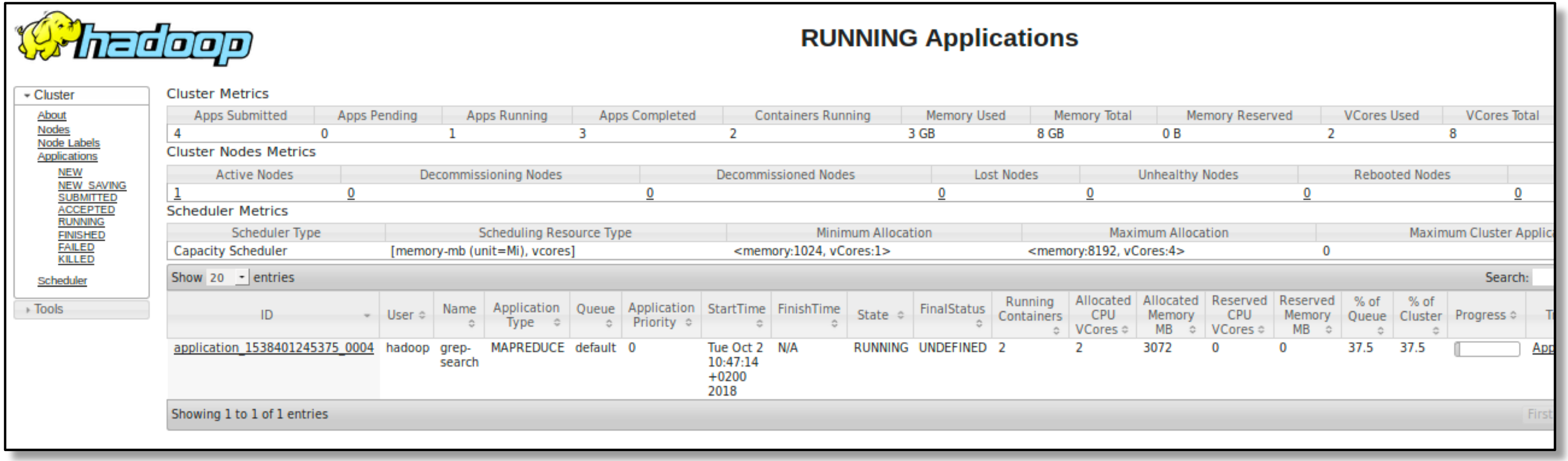
2. Use and run default MapReduce Jar (*hadoop-mapreduce-examples-3.1.1.jar*) to **grep** for string „Faust“ in text file „*Faust_1.txt*“ and count appearances of string.

```
hadoop jar hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.1.1.jar
grep /user/hadoop/Faust_1.txt /user/hadoop/Faust_1_Count_Output 'Faust`
[…]
2018-10-02 10:47:39,680 INFO mapreduce.Job: Running job: job_1538401245375_0005
2018-10-02 10:47:51,905 INFO mapreduce.Job:  map 0% reduce 0%
2018-10-02 10:47:57,966 INFO mapreduce.Job:  map 100% reduce 0%
2018-10-02 10:48:04,023 INFO mapreduce.Job:  map 100% reduce 100%
2018-10-02 10:48:04,031 INFO mapreduce.Job: Job job_1538401245375_0005 completed successfully
[…]
```

Solution

Exercise 3:

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



RUNNING Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total
4	0	1	3	2	3 GB	8 GB	0 B	2	8

Cluster Nodes Metrics

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

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Applic
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	T
application_1538401245375_0004	hadoop	grep-search	MAPREDUCE	default	0	Tue Oct 2 10:47:14 +0200 2018	N/A	RUNNING	UNDEFINED	2	2	3072	0	0	37.5	37.5		App

Showing 1 to 1 of 1 entries

First

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
```

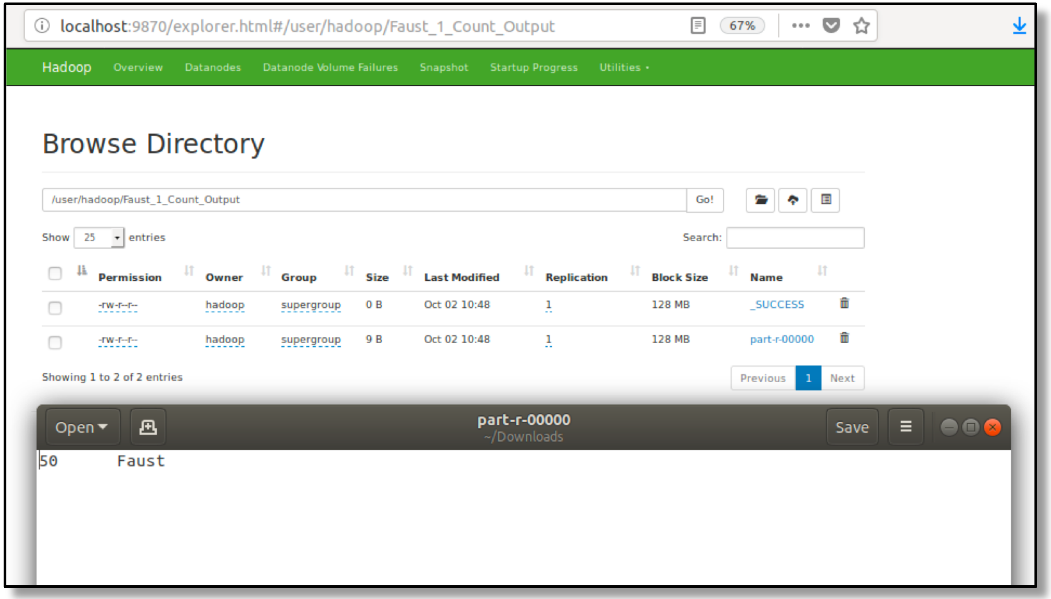
```
cat Faust_1_Count_Output.csv

50          Faust
```

Solution

Exercise 2:

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



Browse Directory

/user/hadoop/Faust_1_Count_Output

Show 25 entries

Search:

	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
□	-rw-r--r--	hadoop	supergroup	0 B	Oct 02 10:48	1	128 MB	_SUCCESS
□	-rw-r--r--	hadoop	supergroup	9 B	Oct 02 10:48	1	128 MB	part-r-00000

Showing 1 to 2 of 2 entries

Previous 1 Next

part-r-00000
~/Downloads

```
50      Faust
```

Help

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.