Report HW#1

Team Member

Pawel Urbanowicz 108015016
Martin Ledl 108012012

Github Repository

Due to the 5MB upload restriction of the online classroom, we weren't able to upload the results and the dataset. Therefore, we made our Github repository publicly accessable. It can be found here: https://github.com/mledl/BDMA_HW

Responsibilities

Pawel Urbanowicz

- Setup local Spark environment for quick testing of code (OSX)
- Setup production environment (cause 16GB RAM MacBook):
  - Docker for Deployment
  - Spark/Hadoop to set off jobs and process data
- Calculation of values using Spark:
  - Minimum, Maximum and Count of requested columns
  - Min-Max Normalization of requested columns
- Run script(s) on production environment
- Take benchmarks and Stats

Martin Ledl

- Setup local Spark environment for quick testing of code (Windows)
- Setup and initialize project on Github
- Data Preprocessing using Spark:
  - Dimension reduction to needed columns
  - Replacement of missing values
  - Conversion of data types
- Calculation of values using Spark:
  - Mean and Standard Deviation of requested Columns
- Output Format of Results:
  - Write to files
  - Merge files due to partitioning of HDFS into 1 file

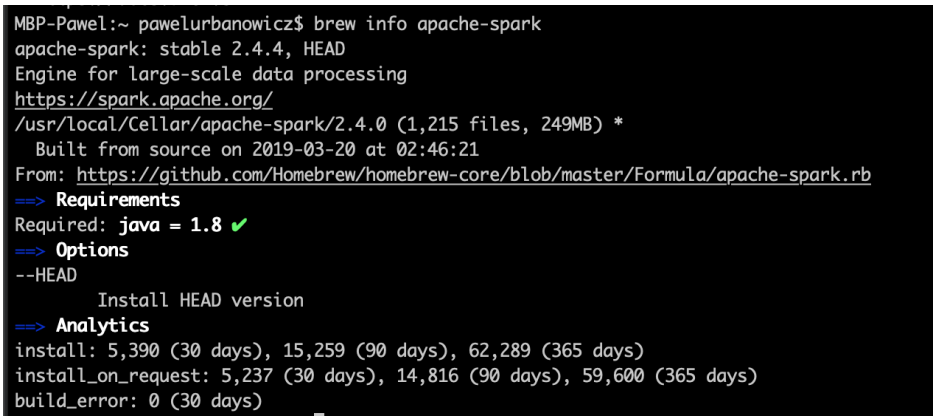
Environment Setup

For local development we tested our code on a locally installed spark instance and for target stage we used Docker technology to wrap spark master instance, 2 spark workers instances and our Python script into separate containers. Software/Frameworks in use:

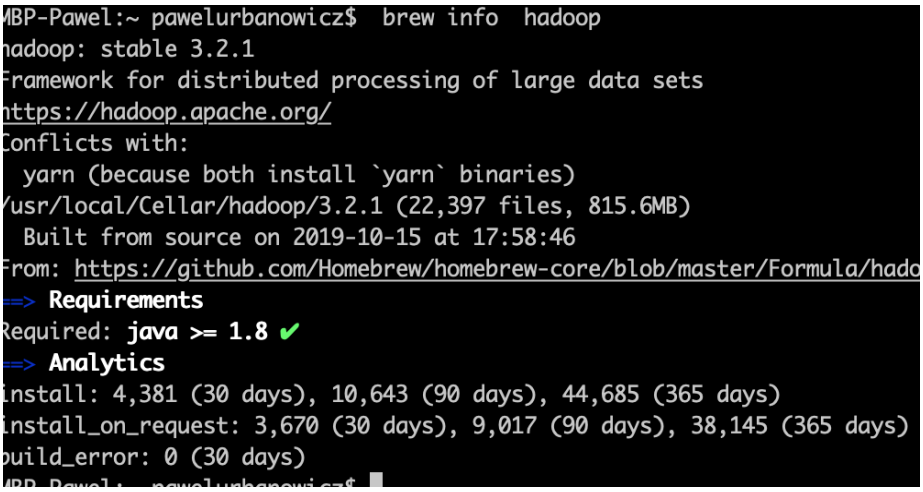
- Python 3.7 to write our code
- Spark version 2.4.4
- Hadoop version 3.2.1
- Docker engine version 18.09.2.

Environment setup for OSX

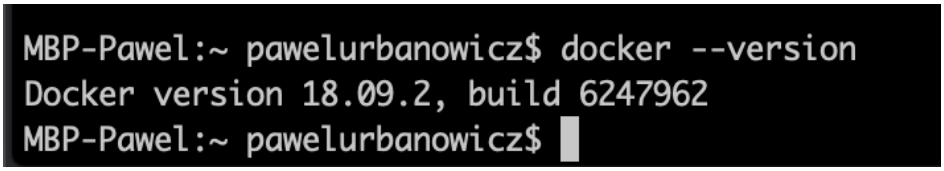
a. Install Spark
brew install apache-spark



b. Install Hadoop



c. Install Docker Desktop for Mac
https://docs.docker.com/docker-for-mac/install/



d. Install Docker Compose
brew install docker-compose

```
MBP-Pawel:~ pawelurbanowicz$ brew info docker-compose
docker-compose: stable 1.24.1 (bottled), HEAD
Isolated development environments using Docker
https://docs.docker.com/compose/
/usr/local/Cellar/docker-compose/1.24.0 (1,635 files, 17.3MB) *
  Poured from bottle on 2019-06-22 at 23:31:43
From: https://github.com/Homebrew/homebrew-core/blob/master/Formula/docker-compose.rb
==> Dependencies
Required: libyaml ✔, python ✔
==> Options
--HEAD
        Install HEAD version
==> Caveats
Bash completion has been installed to:
  /usr/local/etc/bash_completion.d

zsh completions have been installed to:
  /usr/local/share/zsh/site-functions
==> Analytics
install: 11,264 (30 days), 31,818 (90 days), 125,097 (365 days)
install_on_request: 11,024 (30 days), 31,116 (90 days), 120,355 (365 days)
build_error: 0 (30 days)
MBP-Pawel:~ pawelurbanowicz$
```

e. Install python 3
brew install python

```
MBP-Pawel:~ pawelurbanowicz$ python3 --version
Python 3.7.4
MBP-Pawel:~ pawelurbanowicz$
```

f. Clone code from repository
git clone https://github.com/mledl/BDMA_HW

g. Open terminal and go to HW1/docker-spark directory and run:
docker-compose up

```
MBP-Pawel:docker_spark_hadoop pawelurbanowicz$ docker-compose up
Creating network "docker_spark_hadoop_default" with the default driver
Creating namenode     ... done
Creating spark-master ... done
Creating spark-worker-2                ... done
Creating spark-worker-1                ... done
Creating docker_spark_hadoop_datanode_1 ... done
Attaching to spark-master, namenode, spark-worker-1, spark-worker-2, docker_spark_hadoop_datanode_1
namenode          | Configuring core
spark-master      | Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
```

h. Go to HW1 directory to build image for python script
docker build --rm -t hpc-app .

```
MBP-Pawel:HW1 pawelurbanowicz$ docker build --rm -t hpc-app .
Sending build context to Docker daemon  96.87MB
Step 1/11 : FROM bde2020/spark-submit:2.4.4-hadoop2.7
 ---> dac823dd609e
Step 2/11 : COPY /app /app
 ---> 16b126a91da3
Step 3/11 : COPY /preprocessed /preprocessed
 ---> a8de2603ec68
Step 4/11 : COPY docker-spark/template.sh /
 ---> ed53462056b7
Step 5/11 : RUN apk add --update alpine-sdk
 ---> Running in a83f5ae0b58e
```

It can take some time as some libraries must be built from sources

i. Add data to hadoop
docker cp household_power_consumption.csv namenode:household_power_consumption.csv
docker exec -it namenode bash
hadoop fs -mkdir -p household_power_consumption.csv /data/household_power_consumption.csv

j. Run previously build image
docker run -it --name hpc-app -e ENABLE_INIT_DAEMON=false --link spark-master:spark-master --net docker_spark_hadoop_default -d hpc-app

```
MBP-Pawel:HW1 pawelurbanowicz$ docker run -it --name hpc-app -e ENABLE_INIT_DAEMON=false --link spark-master:spark-master  --net
3f349f1d0aae29ba7228402abd62f8e8fe1d2dfb6623e173d588cdbf4e3d1aeb
MBP-Pawel:HW1 pawelurbanowicz$ docker ps
CONTAINER ID        IMAGE                                COMMAND                  CREATED             STATUS              PORT
3f349f1d0aae        hpc-app                              "/bin/bash /template…"   7 seconds ago       Up 5 seconds
e217b0571e1e        bde2020/spark-worker:2.4.4-hadoop2.7   "/bin/bash /worker.sh"   About an hour ago   Up About an hour    8081
95e8265d6db7        bde2020/spark-worker:2.4.4-hadoop2.7   "/bin/bash /worker.sh"   About an hour ago   Up About an hour    8081
2b841a2810b3        bde2020/spark-master:2.4.4-hadoop2.7   "/bin/bash /master.sh"   About an hour ago   Up About an hour    6066
MBP-Pawel:HW1 pawelurbanowicz$
```

Result of setup:

http://localhost:8089/

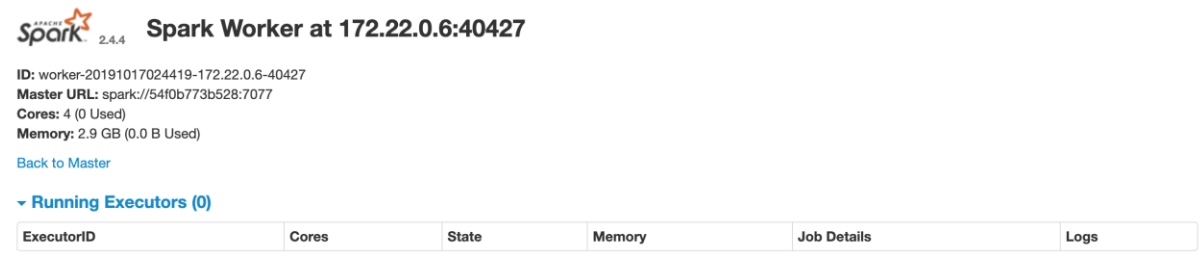
Spark Master at spark://248fe853406e:7077

URL: spark://248fe853406e:7077
Alive Workers: 2
Cores in use: 8 Total, 8 Used
Memory in use: 2.0 GB Total, 2.0 GB Used
Applications: 1 Running, 2 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (2)

Worker Id	Address	State	Cores	Memory
worker-20191015132226-172.19.0.4-38783	172.19.0.4:38783	ALIVE	4 (4 Used)	1024.0 MB (1024.0 MB Used)
worker-20191015132226-172.19.0.6-45749	172.19.0.6:45749	ALIVE	4 (4 Used)	1024.0 MB (1024.0 MB Used)

http://localhost:8084/ and http://localhost:8085/



Spark Worker at 172.22.0.6:40427

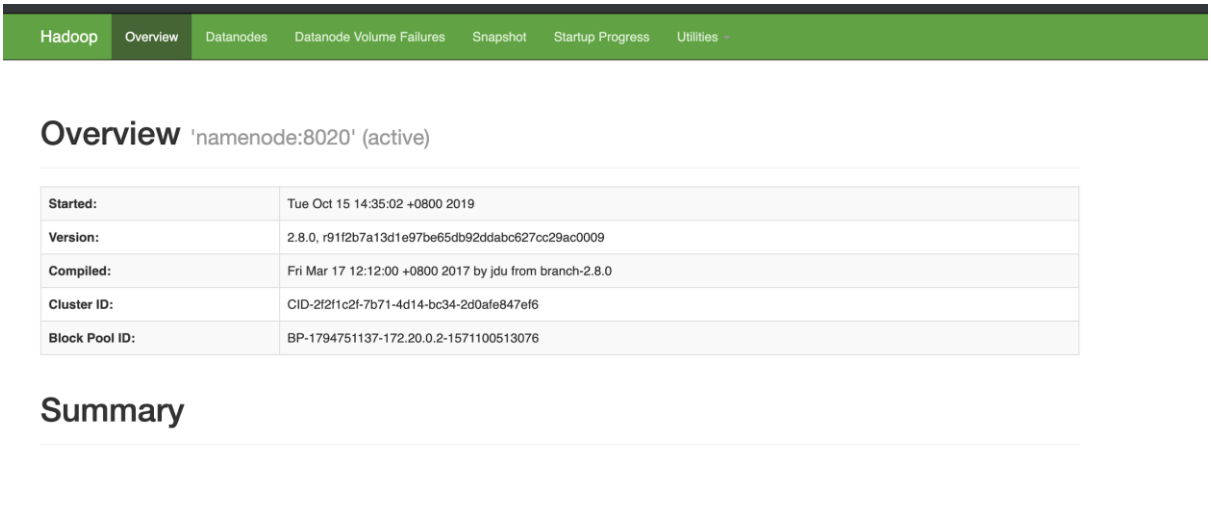
ID: worker-20191017024419-172.22.0.6-40427
Master URL: spark://54f0b773b528:7077
Cores: 4 (0 Used)
Memory: 2.9 GB (0.0 B Used)

Back to Master

Running Executors (0)

ExecutorID	Cores	State	Memory	Job Details	Logs

http://localhost:9870/



docker ps

Data Preprocessing

In order to process the requested tasks on the data and calculate the statistics and min-max-normalization the data needs to be preprocessed to get right results. In this case, data preprocessing has been done in 3 steps:

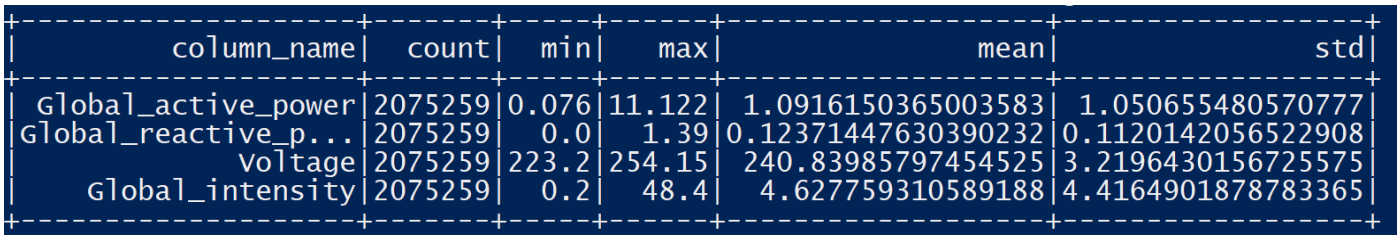
1. **Dimension Reduction**
   Deleted all columns from the raw dataframe except of global_active_power, global_reactive_power, voltage and global_intensity.
2. **Replacement of Missing Values**
   Missing values are represented by “?” in the dataset. We need to tell Spark that “?” should be marked as null value and then replace all null values by the mean of the specific column.
3. **Conversion of data types**
   After replacing all missing values by the specific column mean all values of the specific columns are the same (double due to Spark). Therefore, no more actions regarding the conversion of data types are required.

Output Format

This section describes in which way the results are presented and where the files which contain the results can be found.

Stats (1) (2)

The first two tasks are described together, because they are about the calculation of statistical values (Minimum, Maximum, Count, Mean, Standard Deviation) of the columns global_active_power, global_reactive_power, voltage and global_intensity.
The results of those calculation are gathered in a Spark Dataframe and printed to the standard output of the application. Furthermore, those results are written to a CSV file named “HW1/results/hw1_stats.csv”. The following graphic shows a screen capture of the generated output.



column_name	count	min	max	mean	std
Global_active_power	2075259	0.076	11.122	1.0916150365003583	1.050655480570777
Global_reactive_p...	2075259	0.0	1.39	0.12371447630390232	0.1120142056522908
Voltage	2075259	223.2	254.15	240.83985797454525	3.2196430156725575
Global_intensity	2075259	0.2	48.4	4.627759310589188	4.4164901878783365

Min-Max-Normalization (3)

In the third task we were required to perform min-max-normalization on the columns global_active_power, global_reactive_power, voltage and global_intensity.
The data of those 4 columns is written to 1 single CSV file. Spark creates a new CSV file per partition of HDFS that is used to process the data. This results in multiple CSV files that need to be merged together to one single CSV file holding the min-max-normalized data for the 4 given columns.
To accomplish one single output file there are 2 options.

1. Using coalesce(1) or repartition(1) on the Dataframe instance to write to the file like df.repartition(1).write.format("com.databricks.spark.csv").option("header", "true").mode('overwrite').save("../results/tmp_stats").
   This has the drawback that it is slow for large files, because only 1 partition is used.
2. Use given/default amount of partitions and merge the resulting files as done in “HW1/app/app.py#merge_files” function.

The result of the min-max-normalization is stored in the file “HW1/results/hw1_min_max_normalization.csv”.

Running application in production with 2 spark workers

```
spark-worker-2    | 19/10/17 03:00:51 INFO ExecutorRunner: Launch command: "/usr/lib/jvm/java-1.8-openjdk/jre/bin/java" "-cp" "///conf:/spark/jars/*" "-Xmx1024M" "-Dspark.driver.port=4232
he.spark.executor.CoarseGrainedExecutorBackend" "--driver-url" "spark://CoarseGrainedScheduler@8cd7583ce972:42327" "--executor-id" "1" "--hostname" "172.23.0.6" "--cores" "4" "--app-id" "
030051-0000" "--worker-url" "spark://Worker@172.23.0.6:42557"
spark-worker-1    | 19/10/17 03:00:51 INFO ExecutorRunner: Launch command: "/usr/lib/jvm/java-1.8-openjdk/jre/bin/java" "-cp" "///conf:/spark/jars/*" "-Xmx1024M" "-Dspark.driver.port=4232
he.spark.executor.CoarseGrainedExecutorBackend" "--driver-url" "spark://CoarseGrainedScheduler@8cd7583ce972:42327" "--executor-id" "0" "--hostname" "172.23.0.5" "--cores" "4" "--app-id" "
030051-0000" "--worker-url" "spark://Worker@172.23.0.5:43997"
spark-master      | 19/10/17 03:07:13 INFO Master: Received unregister request from application app-20191017030051-0000
spark-master      | 19/10/17 03:07:13 INFO Master: Removing app app-20191017030051-0000
spark-worker-2    | 19/10/17 03:07:13 INFO Worker: Asked to kill executor app-20191017030051-0000/1
spark-worker-2    | 19/10/17 03:07:13 INFO ExecutorRunner: Runner thread for executor app-20191017030051-0000/1 interrupted
spark-worker-2    | 19/10/17 03:07:13 INFO ExecutorRunner: Killing process!
spark-worker-1    | 19/10/17 03:07:13 INFO Worker: Asked to kill executor app-20191017030051-0000/0
spark-worker-1    | 19/10/17 03:07:13 INFO ExecutorRunner: Runner thread for executor app-20191017030051-0000/0 interrupted
spark-worker-1    | 19/10/17 03:07:13 INFO ExecutorRunner: Killing process!
spark-master      | 19/10/17 03:07:13 INFO Master: 172.23.0.7:57902 got disassociated, removing it.
spark-master      | 19/10/17 03:07:13 INFO Master: 8cd7583ce972:42327 got disassociated, removing it.
spark-worker-2    | 19/10/17 03:07:14 INFO Worker: Executor app-20191017030051-0000/1 finished with state KILLED exitStatus 143
spark-worker-2    | 19/10/17 03:07:14 INFO ExternalShuffleBlockResolver: Clean up non-shuffle files associated with the finished executor 1
spark-worker-2    | 19/10/17 03:07:14 INFO ExternalShuffleBlockResolver: Executor is not registered (appId=app-20191017030051-0000, execId=1)
spark-master      | 19/10/17 03:07:14 WARN Master: Got status update for unknown executor app-20191017030051-0000/0
spark-master      | 19/10/17 03:07:14 WARN Master: Got status update for unknown executor app-20191017030051-0000/1
spark-worker-2    | 19/10/17 03:07:14 INFO ExternalShuffleBlockResolver: Application app-20191017030051-0000 removed, cleanupLocalDirs = true
spark-worker-2    | 19/10/17 03:07:14 INFO Worker: Cleaning up local directories for application app-20191017030051-0000
spark-worker-1    | 19/10/17 03:07:14 INFO Worker: Executor app-20191017030051-0000/0 finished with state KILLED exitStatus 0
spark-worker-1    | 19/10/17 03:07:14 INFO ExternalShuffleBlockResolver: Clean up non-shuffle files associated with the finished executor 0
spark-worker-1    | 19/10/17 03:07:14 INFO ExternalShuffleBlockResolver: Executor is not registered (appId=app-20191017030051-0000, execId=0)
spark-worker-1    | 19/10/17 03:07:14 INFO ExternalShuffleBlockResolver: Application app-20191017030051-0000 removed, cleanupLocalDirs = true
spark-worker-1    | 19/10/17 03:07:14 INFO Worker: Cleaning up local directories for application app-20191017030051-0000
```

Spark Master at spark://a36c08df1e14:7077

URL: spark://a36c08df1e14:7077
Alive Workers: 2
Cores in use: 8 Total, 0 Used
Memory in use: 5.7 GB Total, 0.0 B Used
Applications: 0 Running, 1 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (2)

Worker Id	Address	State	Cores	Memory
worker-20191017025301-172.23.0.5-43997	172.23.0.5:43997	ALIVE	4 (0 Used)	2.9 GB (0.0 B Used)
worker-20191017025301-172.23.0.6-42557	172.23.0.6:42557	ALIVE	4 (0 Used)	2.9 GB (0.0 B Used)

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration

Completed Applications (1)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration
app-20191017030051-0000	app	8	1024.0 MB	2019/10/17 03:00:51	root	FINISHED	6.4 min

Application: app

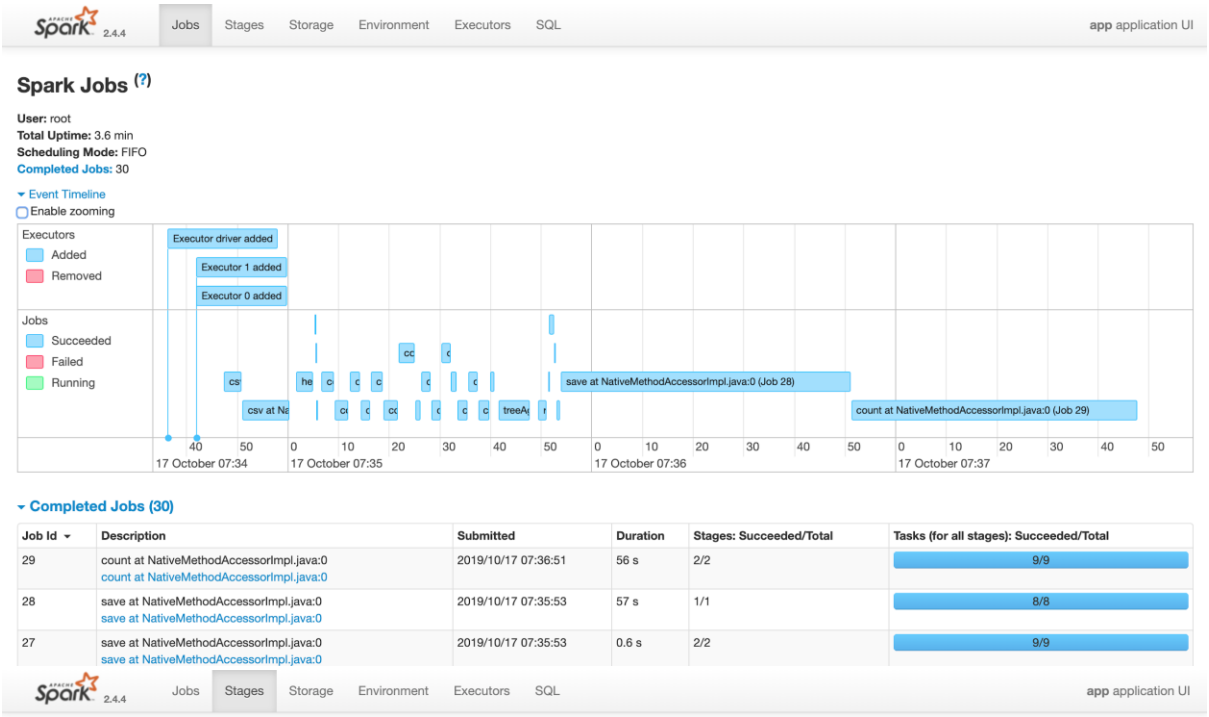
ID: app-20191017030051-0000
Name: app
User: root
Cores: Unlimited (8 granted)
Executor Limit: Unlimited (2 granted)
Executor Memory: 1024.0 MB
Submit Date: 2019/10/17 03:00:51
State: FINISHED

Executor Summary (2)

ExecutorID	Worker	Cores	Memory	State	Logs

Removed Executors (2)

ExecutorID	Worker	Cores	Memory	State	Logs
1	worker-20191017025301-172.23.0.6-42557	4	1024	KILLED	stdout stderr
0	worker-20191017025301-172.23.0.5-43997	4	1024	KILLED	stdout stderr



Completed Jobs (30)

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
29	count at NativeMethodAccessorImpl.java:0 count at NativeMethodAccessorImpl.java:0	2019/10/17 07:36:51	56 s	2/2	9/9
28	save at NativeMethodAccessorImpl.java:0 save at NativeMethodAccessorImpl.java:0	2019/10/17 07:35:53	57 s	1/1	8/8
27	save at NativeMethodAccessorImpl.java:0 save at NativeMethodAccessorImpl.java:0	2019/10/17 07:35:53	0.6 s	2/2	9/9

Stages for All Jobs

Completed Stages: 50

Completed Stages (50)

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
49	count at NativeMethodAccessorImpl.java:0 +details	2019/10/17 07:37:47	0.2 s	1/1			472.0 B	
48	count at NativeMethodAccessorImpl.java:0 +details	2019/10/17 07:36:51	56 s	8/8	126.8 MB			472.0 B
47	save at NativeMethodAccessorImpl.java:0 +details	2019/10/17 07:35:53	57 s	8/8	126.8 MB	149.7 MB		
46	save at NativeMethodAccessorImpl.java:0 +details	2019/10/17 07:35:53	0.3 s	1/1		328.0 B	455.0 B	
45	save at NativeMethodAccessorImpl.java:0 +details	2019/10/17 07:35:53	0.2 s	8/8				455.0 B
44	showString at NativeMethodAccessorImpl.java:0 +details	2019/10/17 07:35:52	0.1 s	3/3				
43	showString at NativeMethodAccessorImpl.java:0 +details	2019/10/17 07:35:51	0.9 s	4/4				
42	showString at NativeMethodAccessorImpl.java:0 +details	2019/10/17 07:35:51	0.1 s	1/1				
41	runJob at PythonRDD.scala:153 +details	2019/10/17 07:35:49	2 s	1/1	896.0 KB			
40	treeAggregate at RowMatrix.scala:433 +details	2019/10/17 07:35:47	0.1 s	2/2			5.0 KB	
39	treeAggregate at RowMatrix.scala:433 +details	2019/10/17 07:35:41	6 s	8/8	127.0 MB			5.0 KB
38	count at NativeMethodAccessorImpl.java:0 +details	2019/10/17 07:35:40	58 ms	1/1			472.0 B	
37	count at NativeMethodAccessorImpl.java:0 +details	2019/10/17 07:35:39	0.8 s	8/8	127.0 MB			472.0 B
36	collect at /app/app.py:56 +details	2019/10/17 07:35:39	70 ms	1/1			704.0 B	
35	collect at /app/app.py:56 +details	2019/10/17 07:35:37	2 s	8/8	127.0 MB			704.0 B
34	collect at /app/app.py:55 +details	2019/10/17 07:35:37	84 ms	1/1			472.0 B	

Docker logs -f hpc-app

```
19/10/17 11:27:54 INFO TaskSchedulerImpl: Removed TaskSet 44.0, whose tasks have all completed, from pool
19/10/17 11:27:54 INFO DAGScheduler: ResultStage 44 (showString at NativeMethodAccessorImpl.java:0) finished in 0.105 s
19/10/17 11:27:54 INFO DAGScheduler: Job 26 finished: showString at NativeMethodAccessorImpl.java:0, took 0.106946 s
+--------------------+-------+-----+------+--------------------+-------------------+
|         column_name|  count|  min|   max|                mean|                std|
+--------------------+-------+-----+------+--------------------+-------------------+
| Global_active_power|2075259|0.076|11.122| 1.0916150365005075| 1.0506554805707766|
|Global_reactive_p...|2075259|  0.0|  1.39|0.12371447630387206|0.11201420565229087|
|             Voltage|2075259|223.2|254.15| 240.83985797449924| 3.2196430156730194|
|    Global_intensity|2075259|  0.2|  48.4|  4.627759310588154|  4.416490187878315|
+--------------------+-------+-----+------+--------------------+-------------------+

19/10/17 11:27:54 INFO FileOutputCommitter: File Output Committer Algorithm version is 1
19/10/17 11:27:54 INFO SQLHadoopMapReduceCommitProtocol: Using output committer class org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
19/10/17 11:27:54 INFO SparkContext: Starting job: save at NativeMethodAccessorImpl.java:0
19/10/17 11:27:54 INFO DAGScheduler: Registering RDD 145 (save at NativeMethodAccessorImpl.java:0)
19/10/17 11:27:54 INFO DAGScheduler: Got job 27 (save at NativeMethodAccessorImpl.java:0) with 1 output partitions
19/10/17 11:27:54 INFO DAGScheduler: Final stage: ResultStage 46 (save at NativeMethodAccessorImpl.java:0)
19/10/17 11:27:54 INFO DAGScheduler: Parents of final stage: List(ShuffleMapStage 45)
19/10/17 11:27:54 INFO DAGScheduler: Missing parents: List(ShuffleMapStage 45)
19/10/17 11:27:54 INFO DAGScheduler: Submitting ShuffleMapStage 45 (MapPartitionsRDD[145] at save at NativeMethodAccessorImpl.java:0), which has no missing p
19/10/17 11:27:54 INFO MemoryStore: Block broadcast_66 stored as values in memory (estimated size 9.6 KB, free 365.0 MB)
19/10/17 11:27:54 INFO MemoryStore: Block broadcast_66_piece0 stored as bytes in memory (estimated size 5.6 KB, free 365.0 MB)
19/10/17 11:27:54 INFO BlockManagerInfo: Added broadcast_66_piece0 in memory on 60dc934ec6f0:43499 (size: 5.6 KB, free: 366.2 MB)
```