

# BDA – Hands-on with Spark

Arnav Tandon (2018278) | Garvita Jain (2018034)

## Methodology and Approach

- Apache Spark and all the relational database systems – Postgres, MongoDB, and Hadoop were installed and successfully set-up.
- The next step involved downloading all connectors that are required to connect all these database systems to Spark.
- All queries are formatted in a python file and run. Execution times are observed for all queries to analyze the efficiency and speed of both the methods.
- The queries used were same as those used in Assignment-1 and therefore follow the assumptions mentioned during its submission.

## Task 1 – Postgres with Apache Spark

Observation of Execution Time:

Tasks	Frontend	Backend
1a   1b	2.3964   1.4340	1.5218   0.2962
2	4.3875	0.4778
3	5.6038	0.5290
4	0.7604	0.2440
5	0.6633	0.1643
6	0.9006	0.3892
7	0.6708	0.1372
Total	12.4856	3.7599

## Screenshots of OUTPUT

```
File Edit View Search Terminal Help
arnav@arnav-lenovo-ideaPad-S540-15iML-0:~/Desktop/tut/spark$ python3 test-spark-connection.py
21/03/11 21:09:56 WARN Utils: Your hostname, arnav-lenovo-ideaPad-S540-15iML-0 resolves to a loopback address: 127.0.1.1; using 192.168.1.82 instead (on interface wlp6s20f3)
21/03/11 21:09:56 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/home/arnav/.local/lib/python3.8/site-packages/pyspark/jars/spark-unsafe_2.12-3.1.1.jar) to constructor java.nio.DirectByteBuffer(long,int)
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
21/03/11 21:09:57 WARN NativeCodeLoader: Unable to load native-heapoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
+-----+
| event | date | count |
+-----+
+-----+
|opened|2010-09-02| 2 |
|opened|2010-09-06| 1 |
|opened|2010-09-08| 1 |
|opened|2010-09-09| 4 |
|opened|2010-09-10| 3 |
|opened|2010-09-11| 3 |
|opened|2010-09-12| 3 |
|opened|2010-09-13| 3 |
|opened|2010-09-15| 2 |
|opened|2010-09-16| 2 |
|opened|2010-09-18| 6 |
|opened|2010-09-19| 4 |
|opened|2010-09-20| 2 |
|opened|2010-09-22| 1 |
|opened|2010-09-23| 4 |
|opened|2010-09-24| 5 |
|opened|2010-09-25| 5 |
|opened|2010-09-27| 4 |
|opened|2010-09-28| 2 |
|opened|2010-09-29| 2 |
+-----+
only showing top 20 rows

Time for 1a Backend 1.5218369960784912
+-----+
| event | date | count |
+-----+
+-----+
|discussed|2010-09-09| 6 |
|discussed|2010-09-10| 10 |
|discussed|2010-09-11| 13 |
|discussed|2010-09-12| 5 |
|discussed|2010-09-13| 7 |
|discussed|2010-09-14| 1 |
|discussed|2010-09-15| 3 |
|discussed|2010-09-16| 2 |
|discussed|2010-09-17| 1 |
```

```

File Edit View Search Terminal Help
[discussed|2010-10-01| 2|
[discussed|2010-10-04| 8|
[discussed|2010-10-06| 15|
-----+-----+
only showing top 20 rows

Time for 1b Backend 0.296252489899658
-----+-----+
| name|monthnumber|countscore|
|-----+-----+
| [rafaelfranca| 1.0| 878|
| [rafaelfranca| 2.0| 555|
| [rafaelfranca| 3.0| 580|
| [rafaelfranca| 4.0| 758|
| [rafaelfranca| 5.0| 915|
| [rafaelfranca| 6.0| 582|
| [rafaelfranca| 7.0| 579|
| [rafaelfranca| 8.0| 655|
| [rafaelfranca| 9.0| 585|
| [rafaelfranca| 10.0| 667|
| [rafaelfranca| 11.0| 596|
| [ralls-bot| 12.0| 546|
|-----+-----+

Time for 2 Backend 0.47786736488342285
-----+-----+
| name| week|countscore|
|-----+-----+
| [mikel|2010-09-06 00:00:00| 16|
| [mikel|2010-09-13 00:00:00| 6|
| [josevalin|2010-09-20 00:00:00| 9|
| [josevalin|2010-09-27 00:00:00| 6|
| [josevalin|2010-10-04 00:00:00| 12|
| [josevalin|2010-10-11 00:00:00| 6|
| [krakoten|2010-10-18 00:00:00| 4|
| [fxn|2010-10-25 00:00:00| 1|
| [rsin|2010-11-01 00:00:00| 2|
| [spastorino|2010-11-01 00:00:00| 2|
| [josevalin|2010-11-08 00:00:00| 6|
| [frackverrot|2010-11-15 00:00:00| 3|
| [josevalin|2010-11-15 00:00:00| 3|
| [tenderlove|2010-11-22 00:00:00| 6|
| [josevalin|2010-11-29 00:00:00| 4|
| [drogus|2010-12-06 00:00:00| 2|
| [josevalin|2010-12-13 00:00:00| 7|
| [dhh|2010-12-20 00:00:00| 5|
| [jeremy|2011-01-03 00:00:00| 21|
| [tenderlove|2011-01-10 00:00:00| 2|
|-----+-----+
only showing top 20 rows

Time for 3 Backend 0.5290446281433185

```

```

File Edit View Search Terminal Help
-----+-----+
| [2010-08-30 00:00:00| 2|
| [2010-09-06 00:00:00| 15|
| [2010-09-13 00:00:00| 17|
| [2010-09-20 00:00:00| 17|
| [2010-09-27 00:00:00| 13|
| [2010-10-04 00:00:00| 10|
| [2010-10-11 00:00:00| 5|
| [2010-10-18 00:00:00| 5|
| [2010-10-25 00:00:00| 3|
| [2010-11-01 00:00:00| 4|
| [2010-11-08 00:00:00| 9|
| [2010-11-15 00:00:00| 8|
| [2010-11-22 00:00:00| 9|
| [2010-11-29 00:00:00| 6|
| [2010-12-06 00:00:00| 5|
| [2010-12-13 00:00:00| 6|
| [2010-12-20 00:00:00| 7|
| [2010-12-27 00:00:00| 4|
| [2011-01-03 00:00:00| 9|
| [2011-01-10 00:00:00| 7|
|-----+-----+
only showing top 20 rows

Time for 4 Backend 0.2440032958984375
-----+-----+
| [non|count|
|-----+-----+

Time for 5 Backend 0.16439294815863477
-----+-----+
| date|count|
|-----+-----+
| [2010-09-02| 2|
| [2010-09-06| 11|
| [2010-09-08| 11|
| [2010-09-09| 10|
| [2010-09-10| 13|
| [2010-09-11| 16|
| [2010-09-12| 8|
| [2010-09-13| 10|
| [2010-09-14| 11|
| [2010-09-15| 5|
| [2010-09-16| 4|
| [2010-09-17| 1|
| [2010-09-18| 6|
| [2010-09-19| 4|
| [2010-09-20| 2|
| [2010-09-21| 6|
| [2010-09-22| 4|
| [2010-09-23| 9|

```

```

File Edit View Search Terminal Help
[2010-09-20] 2|
[2010-09-21] 6|
[2010-09-22] 4|
[2010-09-23] 9|
[2010-09-24] 8|
[2010-09-25] 10|
+-----+
only showing top 20 rows

Time for 6 Backend 0.38921594619750977
+-----+
| author|events|
+-----+
[arunagw] 228|
+-----+

Time for 7 Backend 0.13725733757019043
Total Backend time 3.75995222091675
Ans Frontend 1a
+-----+
| event| timestamp|count(1)|
+-----+
[opened|2010-09-02| 2|
[opened|2010-09-06| 1|
[opened|2010-09-08| 1|
[opened|2010-09-09| 4|
[opened|2010-09-10| 3|
[opened|2010-09-11| 3|
[opened|2010-09-12| 3|
[opened|2010-09-13| 3|
[opened|2010-09-15| 2|
[opened|2010-09-16| 2|
[opened|2010-09-18| 6|
[opened|2010-09-19| 4|
[opened|2010-09-20| 2|
[opened|2010-09-22| 1|
[opened|2010-09-23| 4|
[opened|2010-09-24| 5|
[opened|2010-09-25| 5|
[opened|2010-09-27| 4|
[opened|2010-09-28| 2|
[opened|2010-09-29| 2|
+-----+
only showing top 20 rows

Frontend 1a time 2.396444797515869
Ans Frontend 1b
+-----+
| event| timestamp|count(1)|
+-----+
[discussed|2010-09-09| 6|
[discussed|2010-09-10| 10|

```

```

File Edit View Search Terminal Help
[discussed|2010-09-12| 5|
[discussed|2010-09-13| 7|
[discussed|2010-09-14| 1|
[discussed|2010-09-15| 3|
[discussed|2010-09-16| 2|
[discussed|2010-09-17| 1|
[discussed|2010-09-21| 6|
[discussed|2010-09-22| 3|
[discussed|2010-09-23| 5|
[discussed|2010-09-24| 3|
[discussed|2010-09-25| 5|
[discussed|2010-09-27| 3|
[discussed|2010-09-29| 2|
[discussed|2010-09-30| 3|
[discussed|2010-10-01| 2|
[discussed|2010-10-04| 8|
[discussed|2010-10-06| 15|
+-----+
only showing top 20 rows

Frontend 1b time 1.3715429306030273
Ans Frontend 2
+-----+
| name|monthnumber|countscore|
+-----+
[rafaelfranca| 1| 828|
[rafaelfranca| 10| 667|
[rafaelfranca| 7| 579|
[rafaelfranca| 5| 915|
[rafaelfranca| 4| 758|
[rafaelfranca| 8| 651|
[rafaelfranca| 3| 580|
[rafaelfranca| 6| 592|
| rails-bot| 12| 546|
[rafaelfranca| 2| 555|
[rafaelfranca| 9| 585|
[rafaelfranca| 11| 590|
+-----+

Frontend 2 time 4.387509822845459
Ans Frontend 3
+-----+
| name| week|countscore|
+-----+
| rafaelfranca|2013-04-01 00:00:00| 33|
| pftg|2013-06-12 00:00:00| 33|
| rails-bot|2010-06-18 00:00:00| 33|
| tenderlove|2011-04-18 00:00:00| 12|
| jeremy|2011-01-03 00:00:00| 21|
| rafaelfranca|2015-04-06 00:00:00| 52|
| rails-bot|2017-04-03 00:00:00| 27|
| carlosantoniodasilva|2012-03-26 00:00:00| 39|

```

```

Ans Frontend 3
+-----+-----+
| name | week | countscore |
+-----+-----+
| rafaelfranca|2013-04-01 00:00:00| 33|
| pftg|2013-08-12 00:00:00| 33|
| rails-bot|2016-06-18 00:00:00| 33|
| tenderlove|2011-04-18 00:00:00| 12|
| jeremy|2011-01-03 00:00:00| 21|
| rafaelfranca|2015-04-06 00:00:00| 52|
| rails-bot|2017-04-03 00:00:00| 27|
| carlosantoniodasilva|2012-03-26 00:00:00| 39|
| dhh|2011-04-25 00:00:00| 26|
| steveklabnik|2013-03-11 00:00:00| 50|
| rafaelfranca|2015-01-26 00:00:00| 30|
| mikel|2010-09-13 00:00:00| 6|
| josevalim|2011-11-07 00:00:00| 15|
| rafaelfranca|2015-05-18 00:00:00| 22|
| pixeltrix|2017-03-06 00:00:00| 27|
| rafaelfranca|2013-07-29 00:00:00| 39|
| sgrif|2014-12-22 00:00:00| 60|
| rafaelfranca|2015-05-25 00:00:00| 40|
| rails-bot|2018-05-28 00:00:00| 25|
| spastorino|2011-07-11 00:00:00| 32|
+-----+-----+
only showing top 20 rows

```

Protend 3 time 5.603874921798706

```

Ans Frontend 4
+-----+-----+
| weekstamp|count(pull_requestid)|
+-----+-----+
|2010-08-30 00:00:00| 2|
|2010-09-06 00:00:00| 15|
|2010-09-13 00:00:00| 17|
|2010-09-20 00:00:00| 17|
|2010-09-27 00:00:00| 13|
|2010-10-04 00:00:00| 10|
|2010-10-11 00:00:00| 5|
|2010-10-18 00:00:00| 5|
|2010-10-25 00:00:00| 3|
|2010-11-01 00:00:00| 4|
|2010-11-08 00:00:00| 9|
|2010-11-15 00:00:00| 8|
|2010-11-22 00:00:00| 9|
|2010-11-29 00:00:00| 6|
|2010-12-06 00:00:00| 5|
|2010-12-13 00:00:00| 0|
+-----+-----+

```

File Edit View Search Terminal Help

```

|2010-10-11 00:00:00| 5|
|2010-10-18 00:00:00| 5|
|2010-10-25 00:00:00| 3|
|2010-11-01 00:00:00| 4|
|2010-11-08 00:00:00| 9|
|2010-11-15 00:00:00| 8|
|2010-11-22 00:00:00| 9|
|2010-11-29 00:00:00| 6|
|2010-12-06 00:00:00| 5|
|2010-12-13 00:00:00| 6|
|2010-12-20 00:00:00| 7|
|2010-12-27 00:00:00| 4|
|2011-01-03 00:00:00| 9|
|2011-01-10 00:00:00| 7|
+-----+-----+
only showing top 20 rows

```

Protend 4 time 0.8553266525268555

Ans 5

```

+-----+-----+
| non|count(event)|
+-----+-----+

```

Protend 5 time 0.6633701324462891

Ans Frontend 6

```

+-----+-----+
| timestamp|count(pull_requestid)|
+-----+-----+
|2010-09-02| 2|
|2010-09-06| 1|
|2010-09-08| 1|
|2010-09-09| 10|
|2010-09-10| 13|
|2010-09-11| 16|
|2010-09-12| 8|
|2010-09-13| 10|
|2010-09-14| 11|
|2010-09-15| 5|
|2010-09-16| 4|
|2010-09-17| 1|
|2010-09-18| 6|
|2010-09-19| 4|
|2010-09-20| 2|
|2010-09-21| 6|
|2010-09-22| 4|
|2010-09-23| 9|
|2010-09-24| 8|
|2010-09-25| 10|
+-----+-----+
only showing top 20 rows

```

```

File Edit View Search Terminal Help
[2010-12-27 00:00:00] 4|
[2011-01-03 00:00:00] 9|
[2011-01-10 00:00:00] 7|
+-----+
only showing top 20 rows
Frontend 4 time 0.7004289054876605
Ans 5
+-----+
|mon|count(event)|
+-----+
+-----+
Ans Frontend 6
+-----+
| timestamp|count(pull_requestId)|
+-----+
[2010-09-02] 2|
[2010-09-06] 1|
[2010-09-08] 1|
[2010-09-09] 10|
[2010-09-10] 13|
[2010-09-11] 16|
[2010-09-12] 8|
[2010-09-13] 10|
[2010-09-14] 1|
[2010-09-15] 5|
[2010-09-16] 4|
[2010-09-17] 1|
[2010-09-18] 6|
[2010-09-19] 4|
[2010-09-20] 2|
[2010-09-21] 6|
[2010-09-22] 4|
[2010-09-23] 9|
[2010-09-24] 8|
[2010-09-25] 10|
+-----+
only showing top 20 rows
Frontend 6 time 0.900690739918889
Ans Frontend 7
+-----+
| author|events|
+-----+
[arunag] 228|
+-----+
Frontend 7 time 0.6708910465248479
Total time for frontend 12.485618591308594
arnav@arnav-lenovo-ideapad-5540-151M-01:~/Desktop/tut/spark$

```

## Inferences:

- The queries were answered relatively faster when directly sent to the backend instead of working through the pgadmin service.
- Due to the smaller size of the database, the difference in execution times is not significant. But still comparable to find out the better approach.

## Task 2 – MongoDB

### Importing database on mongodb using Terminal

```

garvita — mongo — 133x35
garvita@GARVITAS-MacBook-Air-934 ~ % mongoimport --db PullRequests --collection Events --type csv --headerline --ignoreBlanks --file
/Users/garvita/Documents/SEMESTER6/BDA/pullreq_events.csv
connected to: mongodb://localhost/
2021-03-11T23:26:42.491+0530 [#####] PullRequests.Events 9.59MB/11.8MB (81.5%)
2021-03-11T23:26:45.491+0530 [#####] PullRequests.Events 11.8MB/11.8MB (100.0%)
2021-03-11T23:26:46.160+0530 273088 document(s) imported successfully. 0 document(s) failed to import.
garvita@GARVITAS-MacBook-Air-934 ~ % mongo
MongoDB shell version v4.4.3
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("03a22394-eafb-47be-b3aa-e4b1b3b7f992") }
MongoDB server version: 4.4.3

The server generated these startup warnings when booting:
  2021-03-10T20:20:24.408+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
  2021-03-10T20:20:24.409+05:30: Soft rlimits too low
    2021-03-10T20:20:24.409+05:30:          currentValue: 256
    2021-03-10T20:20:24.409+05:30:          recommendedMinimum: 64000

Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()

> show dbs
PullRequests 0.000GB
admin         0.000GB
config        0.000GB
local         0.000GB

```

The database is imported in MongoDB and all queries are requested using the shell.

## Task 3 – HDFS (Hadoop File System) with Apache Spark

Observations of Execution Time:

Tasks	Frontend
1a   1b	2.85   3.31
2	18.8
3	27.5
4	2.98
5	3.01
6	5.87
7	9.7
Total	74.02

## Screenshots of OUTPUT

```
In [20]: %%time
sqlContext.sql("select author, count(event) events from pr where event='opened' and extract(year from time)=2011 group by author o
+-----+-----+
| author|events|
+-----+-----+
|arunagw|  228|
+-----+-----+

CPU times: user 1.44 ms, sys: 11.2 ms, total: 12.6 ms
Wall time: 9.7 s
```

```
In [21]: %%time
sqlContext.sql("select date(time), count(pull_requestId) from pr group by date(time) order by date(time)").show()

+-----+-----+
|      time|count(pull_requestId)|
+-----+-----+
|2010-09-02|          2|
|2010-09-06|          1|
|2010-09-08|          1|
|2010-09-09|         10|
|2010-09-10|         13|
|2010-09-11|         16|
|2010-09-12|          8|
|2010-09-13|         10|
|2010-09-14|          1|
|2010-09-15|          5|
|2010-09-16|          4|
|2010-09-17|          1|
|2010-09-18|          6|
|2010-09-19|          4|
|2010-09-20|          2|
|2010-09-21|          6|
|2010-09-22|          4|
|2010-09-23|          9|
|2010-09-24|          8|
|2010-09-25|         10|
+-----+-----+
only showing top 20 rows

CPU times: user 7.75 ms, sys: 2.6 ms, total: 10.4 ms
Wall time: 5.87 s
```

```
%%time
sqlContext.sql("select a.name,b.monthnumber,b.countscore from(select hello.name, max(events) even, hello.monthnumber  from (selec
```

name	monthnumber	countscore
rafaelfranca	1	828
rafaelfranca	2	555
rafaelfranca	3	580
rafaelfranca	4	758
rafaelfranca	5	915
rafaelfranca	6	582
rafaelfranca	7	579
rafaelfranca	8	651
rafaelfranca	9	585
rafaelfranca	10	667
rafaelfranca	11	590
rails-bot	12	546

CPU times: user 2.42 ms, sys: 23.4 ms, total: 25.8 ms  
Wall time: 18.8 s

```
: %%time
sqlContext.sql("select event,date(time),count(*) from pr group by event,date(time) having ev
```

event	time	count(1)
opened	2010-09-02	2
opened	2010-09-06	1
opened	2010-09-08	1
opened	2010-09-09	4
opened	2010-09-10	3
opened	2010-09-11	3
opened	2010-09-12	3
opened	2010-09-13	3
opened	2010-09-15	2
opened	2010-09-16	2
opened	2010-09-18	6
opened	2010-09-19	4
opened	2010-09-20	2
opened	2010-09-22	1
opened	2010-09-23	4
opened	2010-09-24	5
opened	2010-09-25	5
opened	2010-09-27	4
opened	2010-09-28	2
opened	2010-09-29	2

only showing top 20 rows

CPU times: user 0 ns, sys: 23.6 ms, total: 23.6 ms  
Wall time: 4.27 s



```
%%time|
sqlContext.sql("select event,date(time),count(*) from pr group by event,date(time) having event='opened' order by date(tim
```

```
+-----+-----+-----+
| event|      time|count(1)|
+-----+-----+-----+
|opened|2010-09-02|      2|
|opened|2010-09-06|      1|
|opened|2010-09-08|      1|
|opened|2010-09-09|      4|
|opened|2010-09-10|      3|
|opened|2010-09-11|      3|
|opened|2010-09-12|      3|
|opened|2010-09-13|      3|
|opened|2010-09-15|      2|
|opened|2010-09-16|      2|
|opened|2010-09-18|      6|
|opened|2010-09-19|      4|
|opened|2010-09-20|      2|
|opened|2010-09-22|      1|
|opened|2010-09-23|      4|
|opened|2010-09-24|      5|
|opened|2010-09-25|      5|
|opened|2010-09-27|      4|
|opened|2010-09-28|      2|
|opened|2010-09-29|      2|
+-----+-----+-----+
```

only showing top 20 rows

CPU times: user 0 ns, sys: 23.6 ms, total: 23.6 ms  
Wall time: 4.27 s

```
+-----+-----+-----+
|      name|      week|countscore|
+-----+-----+-----+
|      mikel|2010-09-06 00:00:00|      16|
|      mikel|2010-09-13 00:00:00|       6|
|   josevalim|2010-09-20 00:00:00|       9|
|   josevalim|2010-09-27 00:00:00|       6|
|   josevalim|2010-10-04 00:00:00|      12|
|   josevalim|2010-10-11 00:00:00|       6|
|   krekoten|2010-10-18 00:00:00|       4|
|      fxn|2010-10-25 00:00:00|       1|
| spastorino|2010-11-01 00:00:00|       2|
|      rsim|2010-11-01 00:00:00|       2|
|   josevalim|2010-11-08 00:00:00|       6|
|franckverrot|2010-11-15 00:00:00|       3|
|   josevalim|2010-11-15 00:00:00|       3|
| tenderlove|2010-11-22 00:00:00|       6|
|   josevalim|2010-11-29 00:00:00|       4|
|      drogus|2010-12-06 00:00:00|       2|
|   josevalim|2010-12-13 00:00:00|       7|
|      dhh|2010-12-20 00:00:00|       5|
|      jeremy|2011-01-03 00:00:00|      21|
|   josevalim|2011-01-10 00:00:00|       2|
+-----+-----+-----+
```

only showing top 20 rows

CPU times: user 19.7 ms, sys: 14.1 ms, total: 33.8 ms  
Wall time: 27.5 s



```
%%time
sqlContext.sql("select EXTRACT(MONTH FROM time) mon, count(event) from pr where event='merged' and extract(year from time)=2010 gr")
```

mon	count(event)
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	1

```
CPU times: user 7.32 ms, sys: 2.62 ms, total: 9.95 ms
Wall time: 3.01 s
```

```
In [43]: %%time
sqlContext.sql(" select date_trunc('week',time) weekstamp, count(pull_requestid) from pr where event='opened' group by event,weekstamp")
```

weekstamp	count(pull_requestid)
2010-08-30 00:00:00	2
2010-09-06 00:00:00	15
2010-09-13 00:00:00	17
2010-09-20 00:00:00	17
2010-09-27 00:00:00	13
2010-10-04 00:00:00	10
2010-10-11 00:00:00	5
2010-10-18 00:00:00	5
2010-10-25 00:00:00	3
2010-11-01 00:00:00	4
2010-11-08 00:00:00	9
2010-11-15 00:00:00	8
2010-11-22 00:00:00	9
2010-11-29 00:00:00	6
2010-12-06 00:00:00	5
2010-12-13 00:00:00	6
2010-12-20 00:00:00	7
2010-12-27 00:00:00	4
2011-01-03 00:00:00	9
2011-01-10 00:00:00	7

only showing top 20 rows

```
CPU times: user 4.63 ms, sys: 1.77 ms, total: 6.4 ms
Wall time: 2.98 s
```

## 2 Executor

name	week	countscore
mikel	2010-09-06 00:00:00	16
mikel	2010-09-13 00:00:00	6
josevalim	2010-09-20 00:00:00	9
josevalim	2010-09-27 00:00:00	6
josevalim	2010-10-04 00:00:00	12
josevalim	2010-10-11 00:00:00	6
krekoten	2010-10-18 00:00:00	4
fxn	2010-10-25 00:00:00	1
spastorino	2010-11-01 00:00:00	2
rsim	2010-11-01 00:00:00	2
josevalim	2010-11-08 00:00:00	6
franckverrot	2010-11-15 00:00:00	3
josevalim	2010-11-15 00:00:00	3
tenderlove	2010-11-22 00:00:00	6
josevalim	2010-11-29 00:00:00	4
drogus	2010-12-06 00:00:00	2
josevalim	2010-12-13 00:00:00	7
dhh	2010-12-20 00:00:00	5
jeremy	2011-01-03 00:00:00	21
josevalim	2011-01-10 00:00:00	2

only showing top 20 rows

CPU times: user 8.35 ms, sys: 0 ns, total: 8.35 ms  
Wall time: 16.6 s

```
%%time
my_spark.sql("select a.name,b.monthnumber,b.countscore from(select hello.name, max(events) even, hello.monthnumber from (select
```

name	monthnumber	countscore
rafaelfranca	1	828
rafaelfranca	2	555
rafaelfranca	3	580
rafaelfranca	4	758
rafaelfranca	5	915
rafaelfranca	6	582
rafaelfranca	7	579
rafaelfranca	8	651
rafaelfranca	9	585
rafaelfranca	10	667
rafaelfranca	11	590
rails-bot	12	546

CPU times: user 3.2 ms, sys: 12.8 ms, total: 16 ms  
Wall time: 13.1 s

```
#1b
my_spark.sql("select event,date(time),count(*) from pr group by event,date(time) having event='discussed' order by date(time)").show
```

event	time	count(1)
discussed	2010-09-09	6
discussed	2010-09-10	10
discussed	2010-09-11	13
discussed	2010-09-12	5
discussed	2010-09-13	7
discussed	2010-09-14	1
discussed	2010-09-15	3
discussed	2010-09-16	2
discussed	2010-09-17	1
discussed	2010-09-21	6
discussed	2010-09-22	3
discussed	2010-09-23	5
discussed	2010-09-24	3
discussed	2010-09-25	5
discussed	2010-09-27	3
discussed	2010-09-29	2
discussed	2010-09-30	3
discussed	2010-10-01	2
discussed	2010-10-04	8
discussed	2010-10-06	15

only showing top 20 rows

CPU times: user 1.39 ms, sys: 12.1 ms, total: 13.5 ms  
Wall time: 2.03 s

```
#1b
my_spark.sql("select event,date(time),count(*) from pr group by event,date(time) having event='discussed' order by date(time)").show
```

event	time	count(1)
discussed	2010-09-09	6
discussed	2010-09-10	10
discussed	2010-09-11	13
discussed	2010-09-12	5
discussed	2010-09-13	7
discussed	2010-09-14	1
discussed	2010-09-15	3
discussed	2010-09-16	2
discussed	2010-09-17	1
discussed	2010-09-21	6
discussed	2010-09-22	3
discussed	2010-09-23	5
discussed	2010-09-24	3
discussed	2010-09-25	5
discussed	2010-09-27	3
discussed	2010-09-29	2
discussed	2010-09-30	3
discussed	2010-10-01	2
discussed	2010-10-04	8
discussed	2010-10-06	15

only showing top 20 rows

CPU times: user 1.39 ms, sys: 12.1 ms, total: 13.5 ms  
Wall time: 2.03 s

```
: %%time
#5
my_spark.sql("select EXTRACT(MONTH FROM time) mon, count(event) from pr where event='merged' and extract(year from time)=2010 grou
```

mon	count(event)
-----	--------------

CPU times: user 0 ns, sys: 9.8 ms, total: 9.8 ms  
Wall time: 2.02 s

```
my_spark.sql("select event,date(time),count(*) from pr group by event,date(time) having event='opened' order by date(time)").show()
```

```
+-----+-----+
| event|      time|count(1)|
+-----+-----+
|opened|2010-09-02|        2|
|opened|2010-09-06|        1|
|opened|2010-09-08|        1|
|opened|2010-09-09|        4|
|opened|2010-09-10|        3|
|opened|2010-09-11|        3|
|opened|2010-09-12|        3|
|opened|2010-09-13|        3|
|opened|2010-09-15|        2|
|opened|2010-09-16|        2|
|opened|2010-09-18|        6|
|opened|2010-09-19|        4|
|opened|2010-09-20|        2|
|opened|2010-09-22|        1|
|opened|2010-09-23|        4|
|opened|2010-09-24|        5|
|opened|2010-09-25|        5|
|opened|2010-09-27|        4|
|opened|2010-09-28|        2|
|opened|2010-09-29|        2|
+-----+-----+
```

only showing top 20 rows

CPU times: user 5.03 ms, sys: 2.05 ms, total: 7.07 ms  
Wall time: 2.18 s

```
%%time
```

```
#7
```

```
my_spark.sql("select author, count(event) events from pr where event='opened' and extract(year from time)=2011 group by author order by events desc").show()
```

```
+-----+-----+
| author|events|
+-----+-----+
|arunagw|   228|
+-----+-----+
```

CPU times: user 4.3 ms, sys: 1.76 ms, total: 6.06 ms  
Wall time: 2.54 s

```
%%time
```

```
#6
```

```
my_spark.sql("select date(time), count(pull_requestId) from pr group by date(time) order by date(time)").show()
```

```
+-----+-----+
|      time|count(pull_requestId)|
+-----+-----+
|2010-09-02|                    2|
|2010-09-06|                    1|
|2010-09-08|                    1|
|2010-09-09|                   10|
|2010-09-10|                   13|
|2010-09-11|                   16|
|2010-09-12|                    8|
|2010-09-13|                   10|
|2010-09-14|                    1|
|2010-09-15|                    5|
|2010-09-16|                    4|
|2010-09-17|                    1|
|2010-09-18|                    6|
|2010-09-19|                    4|
|2010-09-20|                    2|
|2010-09-21|                    6|
|2010-09-22|                    4|
|2010-09-23|                    9|
|2010-09-24|                    8|
|2010-09-25|                   10|
+-----+-----+
```

only showing top 20 rows

CPU times: user 5.82 ms, sys: 2.36 ms, total: 8.19 ms  
Wall time: 2.13 s

```
#3
my_spark.sql("select a.name,b.week,b.countscore from(select hello.name, max(events) even, hello.week from (select author as nam
```

name	week	countscore
mikel	2010-09-06 00:00:00	16
mikel	2010-09-13 00:00:00	6
josevalim	2010-09-20 00:00:00	9
josevalim	2010-09-27 00:00:00	6
josevalim	2010-10-04 00:00:00	12
josevalim	2010-10-11 00:00:00	6
kreketen	2010-10-18 00:00:00	4
fxn	2010-10-25 00:00:00	1
spastorino	2010-11-01 00:00:00	2
rsim	2010-11-01 00:00:00	2
josevalim	2010-11-08 00:00:00	6
franckverrot	2010-11-15 00:00:00	3
josevalim	2010-11-15 00:00:00	3
tenderlove	2010-11-22 00:00:00	6
josevalim	2010-11-29 00:00:00	4
drogus	2010-12-06 00:00:00	2
josevalim	2010-12-13 00:00:00	7
dhh	2010-12-20 00:00:00	5
jeremy	2011-01-03 00:00:00	21
josevalim	2011-01-10 00:00:00	2

only showing top 20 rows

CPU times: user 10.9 ms, sys: 0 ns, total: 10.9 ms  
Wall time: 15.6 s

#### Inferences:

- Similar trends in the execution time are spotted in both Hadoop and Postgres.
- This verifies the assumption that using the backend service directly using interfaces like Spark greatly improves the query resolution time.

#### Merits and Demerits of Directly Executing commands on Backend of the database System

- Faster speed – Directly executing commands at the backend helps achieve a faster query resolution time. This property is very important while dealing with large databases.
- Easy to Use once configured – Services like Spark gives us the flexibility of options to work in many languages. For example, Java, Scala, Python, R and SQL shells.
- Difficult to set up and configure settings.
- For smaller datasets, the database main memory interfaces provide great ease of use and are better in terms of the tradeoff between speed and comfort.

#### Merits and Demerits of Importing the data in main memory (RDD) to evaluate the queries

- Easy to set up and work on.
- SQL queries reduce chances of erroneous behavior by the system.
- Slower while responding to complex database queries for small data and any SQL query for larger data sizes.
- Does not provide the flexibility and mobility as in the case of backend programming.

#### Learnings

- Working of Apache Spark
- All various functions of Spark and its connectivity and scalability.
- Mechanism, configuration and debugging while installing each software.

- Understood why there is a difference in execution time of queries when requested through different methods.
- Functions and unique points of each Relational Database System included in the scope of this assignment.
- Revised SQL basics and aggregate function queries.
- Understood the meaning of spark connectors and importance of services provided by Apache Spark

### Challenges

- Difficulty connecting Apache Spark with MongoDB system.
- Difficulty installing Hadoop on a Windows system. Had to switch to Ubuntu to successfully work with HDFS.
- Postgres was a relatively simpler task as compared to MongoDB and Hadoop.
- Navigating through and finding relevant information from the documentation.
- Working with unusual directory paths.

### References

- <https://spark.apache.org/docs/latest/sql-data-sources-jdbc.html>
- <https://www.youtube.com/watch?v=snZvQcl2HfQ>
- <https://stackoverflow.com/questions/52390553/org-apache-spark-sql-catalyst-parse-r-parseexception-in-spark-scala-cassandra-ap>
- [https://docs.databricks.com/\\_static/notebooks/mongodb.html](https://docs.databricks.com/_static/notebooks/mongodb.html)
- <https://docs.mongodb.com/spark-connector/current/python-api/>
- <https://community.cloudera.com/t5/Support-Questions/Spark-1-6-How-to-read-and-write-a-csv-file-to-hdfs-without/td-p/222865>