

Ex.2

1.

The 5Gb file is too big to upload it on canvas, you can refer to the screen shot to check the size of this file, and the python file that generates the 5Gb csvfile is also uploaded on canvas.

2.

a)

stu_name	id	score
Aaron Aaberg	4790329952	0

The screen shot of the result:

```
C:\WINDOWS\system32\cmd. x + v
WARN: JAVA_HOME not found in your environment.
Please set the JAVA_HOME variable in your environment to match the
location of your Java installation

Apache Drill 1.21.1
"A little SQL for your NoSQL."
apache drill> select columns[0] as stu_name, columns[1] as id, cast (columns[2] as int) as grade
2..semicolon> from dfs.`C:/Users/oscar meng/Desktop/data_process/l2/students.csv`
3..semicolon> order by grade
4..semicolon> desc
5..semicolon> limit 2;
+-----+-----+-----+
| stu_name | id      | grade |
+-----+-----+-----+
| Aaron Aaberg | 4790329952 | 100   |
| Aaron Aaberg | 4790329952 | 100   |
+-----+-----+-----+
2 rows selected (60.785 seconds)
apache drill> select columns[0] as stu_name, columns[1] as id, cast (columns[2] as int) as grade
2..semicolon> from dfs.`C:/Users/oscar meng/Desktop/data_process/l2/students.csv`
3..semicolon> order by grade
4..semicolon> asc
5..semicolon> limit 2;
+-----+-----+-----+
| stu_name | id      | grade |
+-----+-----+-----+
| Aaron Aaberg | 4790329952 | 0      |
| Aaron Aaberg | 4790329952 | 0      |
+-----+-----+-----+
```

b)

name	avgScore
Latrish Auel	50.574

The screen shot of the result:

```
Apache Drill 1.21.1
"Got Drill?"
apache drill> select data.name,avg(data.score) as avgScore
2..semicolon> from (
3..semicolon>   select name,score
4..semicolon>   from (
5..semicolon>     select
6..semicolon>       columns[0] as name,
7..semicolon>       columns[1] as id,
8..semicolon>       columns[2] as score
9..semicolon>     from dfs.`C:/Users/oscar meng/Desktop/data_process/l2/students.csv`
10..semicolon>   )
11..semicolon> ) as data
12..semicolon> group by data.name
13..semicolon> order by avgScore desc
14..semicolon> limit 10;
```

```

C:\WINDOWS\system32\cmd. x + v
14.....)>
15.....)>          columns[2] as score
16.....)>
17.....)>          from dfs.`C:/Users/oscar meng/Desktop/data_process/l2/students.csv`
18.....)>
19.....)>      )
20.....)>
21.....)> ) as data
22.semicolon>
23.semicolon> group by data.name
24.semicolon>
25.semicolon> order by avgScore desc
26.semicolon>
27.semicolon> limit 10;
+-----+-----+
| name | avgScore |
+-----+-----+
| Latrisha Auel | 50.574 |
| Sol Battistone | 50.56640625 |
| Dewitt Alterman | 50.54984375 |
| Kenya Askari | 50.5325 |
| Christa Alcon | 50.512125 |
| Joy Arrants | 50.50734375 |
| Myung Ballon | 50.50690625 |
| Latonya Audi | 50.50415625 |
| Man Backfisch | 50.49496875 |
| Kisha Assing | 50.49475 |
+-----+-----+
10 rows selected (54.909 seconds)
apache drill>

```

3.

stu_name	id	score	idAsc
Emanuel Amarin	8457454041	50	87904000
Emelda Amormino	1480730870	50	87903999

The screen shot of the result:

```

C:\WINDOWS\system32\cmd. x + v
Error: VALIDATION ERROR: Missing columns column type not compatible with projection specification
Projected column: column[0, 1, 2]
Missing columns column: 'column' VARCHAR NOT NULL
Fragment: 3:0
[Error Id: 342a545c-cedb-40a5-9334-c7ef2277e90a on DESKTOP-SK0EEVJ:31010] (state=,code=0)
apache drill> select stu_name, id, score, idAsc
2..semicolon>
3..semicolon> from (
4.....)>
5.....)>     select columns[0] as stu_name, columns[1] as id, columns[2] as score,
6.....)>     row_number() over (order by cast(columns[2] as int)) as idAsc,
7.....)>     (select count(*) from dfs.`C:/Users/oscar meng/Desktop/data_process/l2/students.csv`) as cnt
8.....)>     from dfs.`C:/Users/oscar meng/Desktop/data_process/l2/students.csv`
9.....)>
10.....)>
11.....)>
12.....)>
13.....)> )
14.semicolon> where (mod(cnt, 2) = 1 and idAsc = trunc((cnt + 1) / 2)) or (mod(cnt, 2) = 0 and idAsc in (trunc(cnt / 2),
trunc(cnt / 2) - 1));
+-----+-----+-----+-----+
| stu_name | id | score | idAsc |
+-----+-----+-----+-----+
| Emelda Amormino | 1480730870 | 50 | 87903999 |
| Emanuel Amarin | 8457454041 | 50 | 87904000 |
+-----+-----+-----+-----+
2 rows selected (198.931 seconds)
apache drill>

```

Ex.3

We prepare data of 7.76MB, 175.44MB, 7.4GB.

Here is our code:

```
from pyspark import sql, SparkConf, SparkContext

spark = sql.session.SparkSession.builder.master("yarn").appName("14e3").getOrCreate()
rdd = spark.read.text("hdfs://hadoop-master:9000/user/root/input/students_100.csv").rdd

pairs = rdd.flatMap(lambda r: [r[0].split(",")[1:3]])
max_grade = pairs.reduceByKey(lambda x, y: max([x, y]))

for entry in max_grade.collect():
    print("{}\t{}".format(entry[0], entry[1]))
```

The results of the maximum grade and the speed are as follows:

For 7.76MB file:



Logs for container_1718969116675_0003_01_000001

▼ ResourceManager	Showing 4096 bytes. Click here for full log
RM Home	2042 99
	5698780648 99
	1704972580 99
	2613805785 99
	3079118982 99
	7719693690 99
► NodeManager	7570202457 99
Tools	4479393409 99
	7505380952 99
	7752841197 97
	7328155585 99
	2493939168 99
	5647212618 99
	4956967352 99
	2228389283 99
	1482860712 98
	3383607347 98
	5548385484 96
	1243470210 98
	8534877313 99
	0115171641 98
	7546222371 99
	8180759754 97
	4741362574 99
	6353386528 99
	1479325877 99
	5347540429 94
	7397329924 98
	1515478394 99
	0045999805 99
	5272921968 99
	6934382406 99
	3177622924 98
	3758517592 99
	1321578086 99
	3085084417 99
	9989452664 98
	4391831070 98
	3426462013 99
	7892297780 99
	4891242399 99
	4964449354 99
	5840687267 99
	6262391581 99

Cluster

About
Nodes
Node Labels
Applications
NEW
NEW CLUSTERING
SUBMITTED
ACCEPTED
SUCCEEDED
FAILED
KILLED
Scheduler

Tools

hadoop-master:8088/cluster/app/application_1718969116675_0003

80%

Application application_1718969116675_0003

Application Overview

User: hadoopuser

Name: spark.py

Application Type: SPARK

Application Tags:

Application Priority: 0 (Higher integer value indicates higher priority)

YarnApplicationState: FINISHED

Queue: default

FinalStatus Reported by AM: SUCCEEDED

Started: Fri Jun 21 17:00:11 +0500 2024

Launched: Fri Jun 21 17:00:11 +0500 2024

Finished: Fri Jun 21 17:01:56 +0500 2024

Elapsed: 1min, 45sec

Tracking URL: History

Log Aggregation Status: DISABLED

Application Timeout (Remaining Time): Unlimited

Diagnostics:

Unmanaged Application: false

Application Node Label expression: <Not set>

AM container Node Label expression: <DEFAULT_PARTITION>

Total Resource Preempted: <memory-0, vCores-0>

Total Number of Non-AM Containers Preempted: 0

Total Number of AM Containers Preempted: 0

Resource Preempted from Current Attempt: <memory-0, vCores-0>

Number of Non-AM Containers Preempted from Current Attempt: 0

Aggregate Resource Allocation: 434537 MB-seconds, 208 vcore-seconds

Aggregate Preempted Resource Allocation: 0 MB-seconds, 0 vcore-seconds

Show 20 entries

Attempt ID

Started

Node

Logs

Nodes blacklisted by the app

Nodes blacklisted by the system

appattempt_1718969116675_0003_000001

Fri Jun 21 20:00:11 +0800 2024

http://hadoop-master:8088/cluster/app/application_1718969116675_0003_000001

Logs

0

0

0

Showing 1 to 1 of 1 entries

First Previous 1 Next

For 175.44MB file:



Logs for container_1718969116675_0004_02_000001

▼ ResourceManager	Showing 4096 bytes. Click here for full log
RM Home	1195 99
	7888883310 99
	3834272389 99
	1728619390 99
	7818150671 99
	8167795462 99
	8283237222 99
	7667753651 99
	3893504879 99
	1798814550 99
	8192917469 99
	4025889764 99
	6783816208 99
	3792131957 99
	1540004704 99
	8791995916 99
	6912471651 99
	2882434131 99
	3832518657 99
	6222640309 99
	7591920531 99
	6703044806 99
	1801575480 99
	3251296679 99
	7166215111 99
	4387285886 99
	6523758085 99
	3600190345 99
	8141452321 99
	2905394150 99
	7253756173 99
	1617126854 99
	2070336938 99
	6211547057 99
	7637621502 99
	6874313976 99
	6980681998 99
	8889785563 99
	3247539988 99
	9253282845 99
	7490144919 99
	9226516428 99
	8989809791 99
	2293610465 99
	5795484296 99
	7511895039 99
	7139622958 99



Application application_1718969116675_0004

▼ Cluster	Application Overview
About	User: hadoopuser
Nodes	Name: spark.py
Node Labels	Application Type: SPARK
Applications	Application Tags:
NEW	Application Priority: 0 (Higher Integer value indicates higher priority)
NEW SAVING	YarnApplicationState: FINISHED
SUBMITTED	Queue: default
ACCEPTED	FinalStatus Reported by AM: SUCCEEDED
RUNNING	Started: Fri Jun 21 17:06:58 +0500 2024
FINISHED	Launched: Fri Jun 21 17:06:58 +0500 2024
FAILED	Finished: Fri Jun 21 17:10:14 +0500 2024
KILLED	Elapsed: 3mins, 15sec
Scheduler	Tracking URL: History
	Log Aggregation Status: DISABLED
	Application Timeout (Remaining Time): Unlimited
	Diagnostics:
	Unmanaged Application: false
	Application Node Label expression: <Not set>
	AM container Node Label expression: <DEFAULT_PARTITION>
Tools	Application Metrics
	Total Resource Preempted: <memory0, vCores0>
	Total Number of Non-AM Containers Preempted: 0
	Total Number of AM Containers Preempted: 0
	Resource Preempted from Current Attempt: <memory0, vCores0>
	Number of Non-AM Containers Preempted from Current Attempt: 0
	Aggregate Resource Allocation: 918747 MB-seconds, 439 vcore-seconds
	Aggregate Preempted Resource Allocation: 0 MB-seconds, 0 vcore-seconds
Show 20 ▼ entries	Search
Attempt ID	Started
appattemp_1718969116675_0004_000002	Fri Jun 21 20:08:06 +0800 2024
	http://hadoop-slave-3:8042
	Logs
	0
	Nodes blacklisted by the app
	0
	Nodes blacklisted by the system

For 7.4GB file:



Logs for container_1718969116675_0013_02_000001

Showing 4096 bytes. Click [here](#) for full log

▼ ResourceManager	5246	99
RM Home	6576487300	99
	0161089773	99
	8824434771	99
	3818624205	99
► NodeManager	4954898686	99
Tools	7429264998	99
	3893727394	99
	6228821768	99
	5887747776	99
	0518439735	99
	1676611085	99
	3451251894	99
	4829665932	99
	1589916893	99
	4778610836	99
	6674991369	99
	6030528251	99
	0477733765	99
	1348433858	99
	9054802351	99
	7146087724	99
	0346286864	99
	0478214011	99
	7704821494	99
	4040679319	99
	7132998761	99
	0045355831	99
	0300288518	99
	0959075677	99
	6916600331	99
	5966548451	99
	5200745224	99
	2097128415	99
	6179228352	99
	0626513203	99
	9387450617	99
	3796421301	99
	9828966267	99
	8424867691	99
	6867433582	99
	2209876490	99
	7486496070	99
	0910519146	99
	4148467917	99
	7225993618	99
	7228749084	99



Application application_1718969116675_0013

Logged in as: hadoopuser

Cluster

About

Nodes

Node Labels

Applications

NEW

NEW SAVING

SUBMITTING

ACCEPTED

RUNNING

FINISHED

PAUSED

FAILED

Scheduler

Tools

User: hadoopuser

Name: spark.py

Application Type: SPARK

Application Tags:

Application Priority: 0 (Higher integer value indicates higher priority)

YarnApplicationState: FINISHED

Queue: default

FinalStatus Reported by AM: SUCCEEDED

Started: Fri Jun 21 18:23:23 +0500 2024

Launched: Fri Jun 21 18:23:23 +0500 2024

Finished: Fri Jun 21 18:31:15 +0500 2024

Elapsed: 7mins, 52sec

Tracking URL: History

Log Aggregation Status: DISABLED

Application Timeout (Remaining Time): Unlimited

Diagnostics:

Unmanaged Application: false

Application Node Label expression: <not set>

AM container Node Label expression: <DEFAULT_PARTITION>

Application Metrics

Total Resource Preempted: <memory:0, vCores:0>

Total Number of Non-AM Containers Preempted: 0

Total Number of AM Containers Preempted: 0

Resource Preempted from Current Attempt: <memory:0, vCores:0>

Number of Non-AM Containers Preempted from Current Attempt: 0

Aggregate Resource Allocation: 11149939 MB-seconds, 1337 vcore-seconds

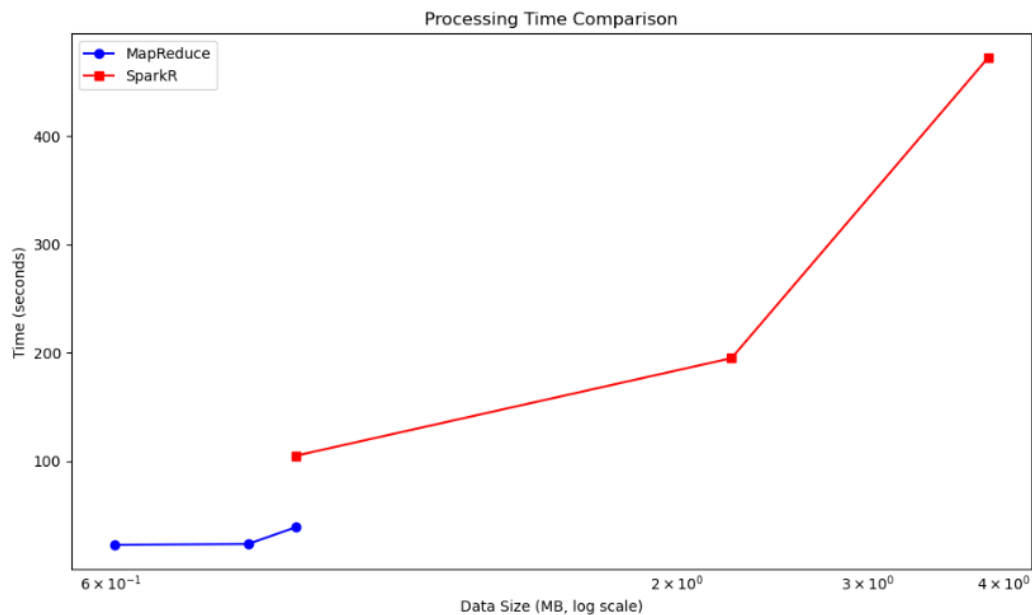
Aggregate Preempted Resource Allocation: 0 MB-seconds, 0 vcore-seconds

Show	20	▼	entries	Attempt ID	Started	Node	Logs	Nodes blacklisted by the app	Nodes blacklisted by the system
attempt_1718969116675_0013_000002				Fri Jun 21 21:23:36 +0800 2024	http://hadoop-storm-3-8042	Logs	0	0	
attempt_1718969116675_0013_000001				Fri Jun 21 21:23:23 +0800 2024	http://hadoop-storm-2-8042	Logs	0	0	

Showing 1 to 2 of 2 entries

First Previous 1 Next Last

Comparing the result between MapReduce and SparkR:



From the graph, we can see that for smaller files, MapReduce is faster than SparkR. For bigger files, we believe that MapReduce will be slower than SparkR. This is because SparkR is more efficient in handling large datasets due to its in-memory processing capabilities.

file size(MB) / time(s)	4.03	6.36	7.76	175.44	7577.6
MapReduce	22.614	23.488	39.084	None	None
SparkR	None	None	105	195	472