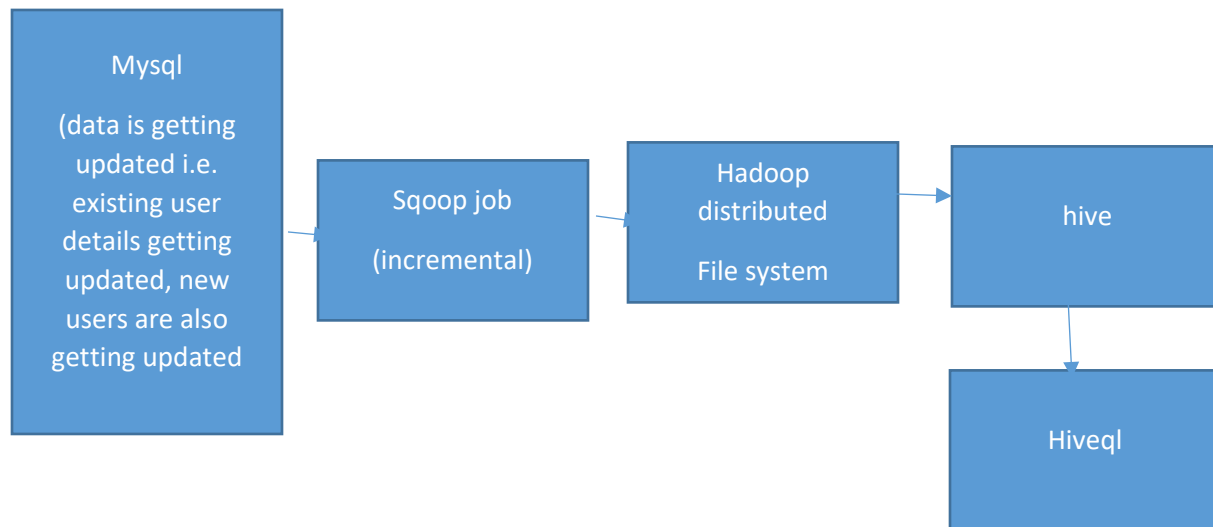


Sources

1. Hadoop documentation
2. Apache spark documentation

Hive incremental dataload



Practical

1. Nano user_details.csv

```
hadoop@hadoop-VirtualBox:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.41-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database userdetails;
Query OK, 1 row affected (0.00 sec)

mysql> use userdetails
Database changed
mysql> create table if not exists users(userid int, username varchar(60), contact varchar(60), date_of_birth varchar(60)
-> ,gender varchar(60), user_city varchar(60), user_email varchar(80), modified_date varchar(80));
Query OK, 0 rows affected (0.09 sec)

mysql> load data local infile '/home/hadoop/user_details.csv'
-> into table users
-> fields terminated by ','
-> lines terminated by '\n'
```

```
mysql> load data local infile '/home/hadoop/user_details.csv'
-> into table users
-> fields terminated by ','
-> lines terminated by '\n'
-> ignore 1 lines;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'bu '\n'
ignore 1 lines' at line 4
mysql> load data local infile '/home/hadoop/user_details.csv'
-> into table users
-> fields terminated by ','
-> lines terminated by '\n'
-> ignore 1 lines;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'bu '\n'
ignore 1 lines' at line 4
mysql> load data local infile '/home/hadoop/user_details.csv'
-> into table users
-> fields terminated by ','
-> lines terminated by '\n'
-> ignore 1 lines;
Query OK, 4 rows affected, 3 warnings (0.02 sec)
Records: 4 Deleted: 0 Skipped: 0 Warnings: 3

mysql> select * from users;
+-----+-----+-----+-----+-----+-----+-----+-----+

```

```
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'bu '\n'
ignore 1 lines' at line 4
mysql> load data local infile '/home/hadoop/user_details.csv'
-> into table users
-> fields terminated by ','
-> lines terminated by '\n'
-> ignore 1 lines;
Query OK, 4 rows affected, 3 warnings (0.02 sec)
Records: 4 Deleted: 0 Skipped: 0 Warnings: 3

mysql> select * from users;
+-----+-----+-----+-----+-----+-----+-----+-----+
| userid | username | contact | date_of_birth | gender | user_city | user_email | modified_date |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 0 | username | contact | date_of_birth | gender | user_city | NULL | NULL |
| 1 | parth | 9999999999 | 2001-01-13 | male | hyderabad | parth@teksystems.com | 2023-01-13 |
| 2 | jirti | 9999999999 | 2002-01-13 | female | hyderabad | kirti@teksystems.com | 2022-01-13 |
| 3 | bhanu | 9999999999 | 2001-02-13 | male | hyderabad | bhanusir@teksystems.com | 2020-02-13 |
+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> █
```

```
hadoop@hadoop-VirtualBox:~$ sqoop import --connect jdbc:mysql://localhost/userdetails --username root --password hadoop@123 --table users -m 1 --target-dir '/userdetails/'
```

```
use hive;

drop table if exists users;

create table if not exists users(userid int, username varchar(60), contact varchar(60), date_of_birth date, gender varchar(60), user_city varchar(70), user_email varchar(80), modified_date date)
row format delimited
fields terminated by ','
lines terminated by '\n'
stored as textfile
location '/userdetails'
;

select * from users;
```

```
mysql> use userdetails;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> load data local infile '/home/hadoop/userdetails1.csv' into table users fields terminated by ',' lines terminated by '\n';
Query OK, 4 rows affected, 8 warnings (0.01 sec)
Records: 4 Deleted: 0 Skipped: 0 Warnings: 8

mysql> select * from users;
```

userid	username	contact	date_of_birth	gender	user_city	user_email	modified_date
0	username	contact	date_of_birth	gender	user_city	NULL	NULL
1	parth	9999999999	2001-01-13	male	hyderabad	parth@teksystems.com	2023-01-13
2	jirti	9999999999	2002-01-13	female	hyderabad	kirti@teksystems.com	2022-01-13
3	bhanu	9999999999	2001-02-13	male	hyderabad	bhanusir@teksystems.com	2020-02-13
0	NULL	NULL	NULL	NULL	NULL	NULL	NULL
4	bhanu_superman	3332323232	2001-06-13	male	tokyo	bhanusir@teksystems.com	2023-03-29
6	somanna	21212121	2001-06-13	male	kolkata	som@teksystems.com	2023-03-30
7	tiyanshi	3332323232	2001-06-13	female	paris	tiyanshi@teksystems.com	2023-03-29

```
8 rows in set (0.00 sec)
```

Python mapper program

1. Gedit mapper.py



```
#!/usr/bin/python3
"""mapper.py"""
import sys

for line in sys.stdin:
    line=line.strip()
    words = line.split()
    for word in words:
        print('%s\t%s' % (word,1))
```

2. Cat /home/Hadoop/Downloads/harry potter.txt | python3 mapper.py

```

that      1
was       1
spreading      1
over      1
his       1
face.     1
"They    1
don't     1
know     1
we're    1
not       1
allowed  1
to        1
use       1
magic     1
at        1
home.     1
I'm       1
going     1
to        1
have      1
a         1
lot       1
of        1
fun       1
with      1
Dudley   1
this      1
summer...."      1
THE       1
END       1

```

hadoop@hadoop-VirtualBox:~\$ gedit mapper.py

```

File  Text Editor  Wed 14:59
Open  reducer.py

#!/usr/bin/python3

from operator import itemgetter
import sys

current_word=None
current_count=0
word=None

for line in sys.stdin:
    line=line.strip()
    #word and count are separated by <tab> delimited. 1 -> refers to no of splits
    word,count= line.split('\t',1)

    #converting count string into int
    try:
        count=int(count)
    except ValueError:
        continue

    if current_word ==word:
        current_count+=count
    else:
        if current_word:
            print('%s\t%s'%(current_word,current_count))
        current_count=count
        current_word=word

#to output the last word
if current_word==word:
    print('%s\t%s'%(current_word,current_count))

```

Python

3.

zooming 2

```

hadoop@hadoop-VirtualBox:~$ cat /home/hadoop/Downloads/HarryPotter.txt | python3 mapper.py | sort | python3 reducer.py

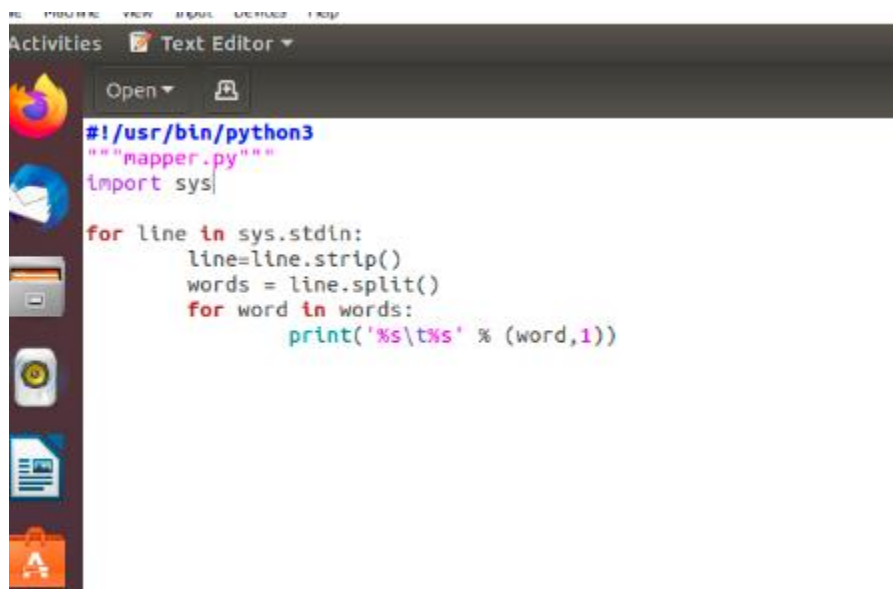
```

```

Your      9
YOUR      1
you're    41
"You're   14
You're    2
yours,"   2
yours,    3
yours?"   1
yours.    1
Yours     1
yourself," 2
yourself,  4
yourself"  1
yourself   7
yourselves, 1
yourselves." 1
yourselves. 2
yourselves 1
youth     1
you've    22
"You've    8
You've     2
Yvonne?"   1
"Zabini,   1
zigzagging 2
zombie,    2
zoo,"      1
zoo.       2
Zoo        4
Zoom       1
Zoomed     1
Zooming    2

```

Suppose we want Hadoop to do the map reduce job




```

#!/usr/bin/python3
"""mapper.py"""
import sys

for line in sys.stdin:
    line=line.strip()
    words = line.split()
    for word in words:
        print('%s\t%s' % (word,1))

```

```
Open ▾ 
#!/usr/bin/python3
"""reducer.py"""
from operator import itemgetter
import sys

current_word=None
current_count=0
word=None

for line in sys.stdin:
    line=line.strip()
    #word and count are separated by <tab> delimited. 1 -> refers to no of splits
    word,count= line.split('\t',1)

    #converting count string into int
    try:
        count=int(count)
    except ValueError:
        continue

    if current_word ==word:
        current_count+=count
    else:
        if current_word:
            print('%s\t%s'%(current_word,current_count))
            current_count=count
            current_word=word
#to output the last word
if current_word==word:
    print('%s\t%s'%(current_word,current_count))
```

Chmod +x mapper.py

Chmod +x reducer.py

```
hadoop@hadoop-VirtualBox:~$ hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.7.6.jar -file mapper.py -mapper /home/hadoop/mapper.py -file reducer.py -reducer /home/hadoop/reducer.py -input /wordcount/* -output /word_count
```

```
hadoop@hadoop-VirtualBox:~$ hdfs dfs -cat /word_count/part-00000
```