hive project

PART 1

1. create `wdi_gs` table to parse data from `wdi_2016`
2. create `wdi_csv_text` external table and insert data from `wdi_gs` into it
3. notice the cache in worker and master nodes
4. compare the bash approache and hive approache to scan whole table

- Create a table, called `wdi_gs`
- Data is stored in `gs://jarvis_data_eng_haotianzhu/dataset/wdi_2016`

```
%hive


DROP TABLE IF EXISTS wdi_gs
```

```
Query executed successfully. Affected rows : -1
```

```
%hive

CREATE EXTERNAL TABLE wdi_gs
(year INTEGER, countryName STRING, countryCode STRING, indicatorName STRING, indicatorCode STRING,
  indicatorValue FLOAT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n'
LOCATION 'gs://jarvis_data_eng_haotianzhu/dataset/wdi_2016'
TBLPROPERTIES ("skip.header.line.count"="1")
```

```
Query executed successfully. Affected rows : -1
```

- Data information shown below

```
%hive
DESCRIBE FORMATTED wdi_gs
```

settings

col_name	data_type	comment
# col_name	data_type	comment
	null	null
year	int	
countryname	string	
countrycode	string	
indicatorname	string	
indicatorcode	string	
indicatorvalue	float	

scan whole table using query `selct count(*) from wdi_gs`

```
select count(*) from wdi_gs
```

settings

_c0
21759408

- drop table if exists
- create a external table named as `wdi_csv_text`
- `hdfs` location: `hdfs:///user/hive/wdi/wdi_csv_text`
- `comma` delimited format
- after creating the table, insert data from `wdi_gs` into `wdi_csv_text`
- run query to check if all data is inserted successfully

```
DROP TABLE IF EXISTS wdi_csv_text
```

```
Query executed successfully. Affected rows : -1
```

```
CREATE EXTERNAL TABLE wdi_csv_text
(year INTEGER, countryName STRING, countryCode STRING, indicatorName STRING, indicatorCode STRING, indicatorValue FLOAT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n'
LOCATION 'hdfs:///user/hive/wdi/wdi_csv_text'
```

```
Query executed successfully. Affected rows : -1
```

```
INSERT OVERWRITE TABLE wdi_csv_text
SELECT * FROM wdi_gs
```

```
Query executed successfully. Affected rows : -1
```

- we have inserted data into database, and now we want to check if files are stored in hdfs
- to check that we use `hdfs dfs` command
- after that we execute SELECT count(countryName) FROM wdi_csv_text twice and we find that 2nd query run faster

```
%sh
hdfs dfs -ls -h  hdfs:///user/hive/wdi/wdi_csv_text
hdfs dfs -du -s -h hdfs:///user/hive/wdi/wdi_csv_text
```

```
Found 5 items
-rwxrwxrwt   2 anonymous hadoop    386.0 M 2020-11-30 22:35 hdfs:///user/hive/wdi/wdi_csv_text/000000_0
-rwxrwxrwt   2 anonymous hadoop    385.8 M 2020-11-30 22:35 hdfs:///user/hive/wdi/wdi_csv_text/000001_0
-rwxrwxrwt   2 anonymous hadoop    386.0 M 2020-11-30 22:35 hdfs:///user/hive/wdi/wdi_csv_text/000002_0
-rwxrwxrwt   2 anonymous hadoop    385.6 M 2020-11-30 22:35 hdfs:///user/hive/wdi/wdi_csv_text/000003_0
-rwxrwxrwt   2 anonymous hadoop    210.4 M 2020-11-30 22:35 hdfs:///user/hive/wdi/wdi_csv_text/000004_0
1.7 G  hdfs:///user/hive/wdi/wdi_csv_text
```

```
Beeline version 2.3.7 by Apache Hive
0: jdbc:hive2://cluster-c54d-m:10000> select count(*) from wdi_csv_text;
+-----------+
|    _c0    |
+-----------+
| 21759408  |
+-----------+
1 row selected (3.23 seconds)
0: jdbc:hive2://cluster-c54d-m:10000> select count(*) from wdi_csv_text;
+-----------+
|    _c0    |
+-----------+
| 21759408  |
+-----------+
1 row selected (0.39 seconds)
```

As we can see from above, the second query is faster than the first one because of the file system cache.

When we run the same query in the worker node, the speed is as fast as the second query in the master node.

After we clear the cache, we re-run the query, and the query has the same speed as the first one in the master node.

The worker and master nodes share the same cache.

Discuss the performance result between the bash and Hive approaches.

- The performance result with the Hive approach is much better than that with the bash approach.
- The bash approach is around 30 sec while hive approach (sql) only costs 1 sec

```
%sh
cd ~
hdfs  dfs -get  hdfs:///user/hive/wdi/wdi_csv_text .
cd wdi_csv_text
#calculate current directory size
du -ch .
#1.8G   total

#clear fs cache
echo 3 | sudo tee /proc/sys/vm/drop_caches
#bash row count
date +%s && cat * | wc && date +%s
```

```
1.8G    .
1.8G    total

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

sudo: no tty present and no askpass program specified
1606778355
21759408 179709942 1838962843
1606778369
```

PART2

- find the bug that the `indicatorcode` col does not display properly
- the bug happends because of the parsing issue, to solve it, we choose to use a different parsing way.
- drop table if exists
- create a new table called `wdi_opencsv_gs` using `OpenCSVSerde`
- create `wdi_opencsv_text` destination table (output table with hdfs location)
- insert data from `wdi_opencsv_gs` into `wdi_opencsv_text`
- verify data

```
%hive

SELECT distinct(indicatorcode)
FROM wdi_csv_text
ORDER BY indicatorcode
LIMIT 10
```

```
DROP TABLE IF EXISTS wdi_opencsv_gs
```

```
CREATE EXTERNAL TABLE wdi_opencsv_gs
(year INTEGER, countryName STRING, countryCode STRING, indicatorName STRING, indicatorCode STRING, indicatorValue FLOAT)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
LOCATION 'gs://jarvis_data_eng_haotianzhu/dataset/wdi_2016'
TBLPROPERTIES ("skip.header.line.count"="1")
```

```
DROP TABLE IF EXISTS wdi_opencsv_text
```

```
CREATE EXTERNAL TABLE wdi_opencsv_text
(year INTEGER, countryName STRING, countryCode STRING, indicatorName STRING, indicatorCode STRING, indicatorValue FLOAT)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
LOCATION 'hdfs:///user/hive/wdi/wdi_opencsv_text'
```

```
INSERT OVERWRITE TABLE wdi_opencsv_text
SELECT * FROM wdi_opencsv_gs
```

```
Query executed successfully. Affected rows : -1
```

```
SELECT distinct(indicatorcode)
FROM wdi_opencsv_text
ORDER BY indicatorcode
LIMIT 10
```

settings

indicatorcode
AG.AGR.TRAC.NO
AG.CON.FERT.PT.ZS

- Now we run `SELECT count(countryName)` query to wdi_opencsv_text and wdi_csv_text
- We find that wdi_opencsv_text is pretty slow
- it is because that SerDe do extra work to parse csv, and it is much slower than conventional delimiter definition.

```
%hive

SELECT count(countryName) FROM wdi_opencsv_text
```

settings

_c0
21759408

```
%hive
SELECT count(countryName) FROM wdi_csv_text
```

settings

_c0
21759408

- we find that `indicatorCode` becomes string in wdi_opencsv_text
- it is because that SerDe treats all columns to be of type String. Even if you create a table with non-string column types using this SerDe, the DESCRIBE TABLE output would show string column type. The type information is retrieved from the SerDe.
- To solve such problem, we can use View table

```
DROP VIEW IF EXISTS wdi_opencsv_text_view
```

```
Query executed successfully. Affected rows : -1
```

```
CREATE VIEW IF NOT EXISTS wdi_opencsv_text_view
AS
SELECT year, countryName, countryCode, indicatorName, indicatorCode, Cast(indicatorValue AS FLOAT) AS indicatorValue
FROM wdi_csv_text
```

```
Query executed successfully. Affected rows : -1
```

PART3

write queries to solve business problem

Write a HiveQL to find 2015 `GDP growth (annual %)` for Canada.

Output columns: `GDP_growth_value, year, countryName`

```
select  indicatorValue AS GDP_growth_value, year, countryName
from wdi_opencsv_text_view
where indicatorname like "GDP growth (annual %)" and year = 2015 and countryName="Canada"
```

settings

gdp_growth_value	year	countryname
1.0782688	2015	Canada

- the query is slow. it costs 30 sec to fetch the result, the main reason is that data is not pre-sorted and there is no index to help us quickly fetch result by keyword
- to solve it, we use the partition

```
DROP TABLE IF EXISTS wdi_opencsv_text_partitions
```

Query executed successfully. Affected rows : -1

```
%hive

CREATE EXTERNAL TABLE wdi_opencsv_text_partitions
(year INTEGER, countryName STRING, countryCode STRING, indicatorName STRING, indicatorCode STRING, indicatorValue FLOAT)
PARTITIONED by(y STRING)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
LOCATION 'hdfs:///user/hive/wdi/wdi_opencsv_text'
```

Query executed successfully. Affected rows : -1

```
set hive.exec.dynamic.partition.mode=nonstrict
```

Query executed successfully. Affected rows : -1

```
set hive.exec.dynamic.partition=true
```

Query executed successfully. Affected rows : -1

```
INSERT OVERWRITE TABLE wdi_opencsv_text_partitions PARTITION(y)
SELECT *, year as y FROM wdi_opencsv_text
```

Query executed successfully. Affected rows : -1

```
testfreegcp@cluster-c54d-m:~$ hdfs dfs -ls /user/hive/wdi/wdi_opencsv_text
Found 62 items
-drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=1977
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=1978
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=1979
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=1980
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=1981
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=1982
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=1983
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=1984
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=1985
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=1986
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=1987
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=1988
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=1989
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=1990
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=1991
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=1992
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=1993
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=1994
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=1995
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=1996
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=1997
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=1998
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=1999
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=2000
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=2001
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=2002
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=2003
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=2004
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=2005
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=2006
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=2007
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=2008
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=2009
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=2010
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=2011
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=2012
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=2013
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=2014
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=2015
drwxrwxrwt   - gpadmin hadoop          0 2020-12-03 07:20 /user/hive/wdi/wdi_opencsv_text/y=2016
```

- using partition by year, it will create mutiple folders based on the years

- run same query, it only costs few sec to get the result

```
%hive

select  indicatorValue AS GDP_growth_value, year, countryName
from wdi_opencsv_text_partitions
where indicatorname like "GDP growth (annual %)" and y = 2015 and countryName="Canada"
```

settings

gdp_growth_value	year	countryname
1.07826875075381	2015	Canada

- solve the run speed, next thing is that we want to reduce the file size
- we use PARQUET to store data

```
%hive
DROP TABLE IF EXISTS wdi_csv_parquet
```

Query executed successfully. Affected rows : -1

```
CREATE EXTERNAL TABLE wdi_csv_parquet
(year INTEGER, countryName STRING, countryCode STRING, indicatorName STRING, indicatorCode STRING, indicatorValue STRING)
STORED AS PARQUET
LOCATION 'hdfs:///user/hive/wdi/wdi_csv_parquet'
```

Query executed successfully. Affected rows : -1

check if all data is inserted

```
INSERT OVERWRITE TABLE wdi_csv_parquet
SELECT * FROM wdi_opencsv_gs
```

Query executed successfully. Affected rows : -1

```
SELECT count(countryName) FROM wdi_csv_parquet
```

settings

_c0
21759408

```
SELECT count(countryName) FROM wdi_csv_textwdi_csv_text
```

settings

_c0
21759408

```
testfreegcp@cluster-c54d-m:~$ hdfs dfs -du -s -h /user/hive/wdi/wdi_csv_parquet
227.3 M  /user/hive/wdi/wdi_csv_parquet
testfreegcp@cluster-c54d-m:~$ hdfs dfs -du -s -h /user/hive/wdi/wdi_csv_text
1.7 G  /user/hive/wdi/wdi_csv_text
```

you can see that using parquet can significantly reduce the file size

Execute 2015 GDP Growth HQL against wdi_csv_parquet and wdi_opencsv_text tables, and then compare performance.
- Even though, the file size is reduced, the running time is not increased

```
select  indicatorValue AS GDP_growth_value, year, countryName
from wdi_csv_parquet
where indicatorname like "GDP growth (annual %)" and year = 2015 and countryName="Canada"
```

settings

gdp_growth_value	year	countryname
1.07826875075381	2015	Canada

```
select  indicatorValue AS GDP_growth_value, year, countryName
from wdi_csv_text
where indicatorname like "GDP growth (annual %)" and year = 2015 and countryName="Canada"
```

settings

gdp_growth_value	year	countryname
1.0782688	2015	Canada

Second query

- Find the highest GDP growth (NY.GDP.MKTP.KD.ZG) year for each country.
- And then try to use Spark interpreter to run same query.
- The spark.sql run query faster than that in hive tez

```
select v.countryName, v.indicatorValue, min(v.year)
from (select countryName, max(indicatorValue) as indicatorValue
from wdi_opencsv_text_view
where indicatorname like "GDP growth (annual %)"
group by countryName
order by countryName) AS t, wdi_opencsv_text_view v
where v.countryName=t.countryName AND v.indicatorValue=t.indicatorValue
group by v.countryName, v.indicatorValue
order by v.countryName
```

settings

v.countryname	v.indicatorvalue	_c2
Afghanistan	21.020649	2009
Albania	13.501173	1999
Algeria	34.31373	1963
American Samoa	0.0	1960
Andorra	12.02392	2003
Angola	22.593054	2007
Antigua and Barbuda	13.3764	2006
Arab World	12.896334	1976

```
%spark.sql


select v.countryName, v.indicatorValue, min(v.year)
from (select countryName, max(indicatorValue) as indicatorValue
from wdi_opencsv_text_view
where indicatorname like "GDP growth (annual %)"
group by countryName
order by countryName) AS t, wdi_opencsv_text_view v
where v.countryName=t.countryName AND v.indicatorValue=t.indicatorValue
group by v.countryName, v.indicatorValue
order by v.countryName
```

settings

countryName	indicatorValue	min(year)
Afghanistan	21.020649	2009
Albania	13.501173	1999

third query

Write a query that returns GDP Growth for all coutries. Sort by countryName and year.

```
select countryName, year, indicatorValue
from wdi_opencsv_text_view
where indicatorname like "GDP growth (annual %)"
order by countryName ASC, year ASC
```

settings

countryname	year	indicatorvalue
Afghanistan	1960	0.0
Afghanistan	1961	0.0
Afghanistan	1962	0.0
Afghanistan	1963	0.0
Afghanistan	1964	0.0
Afghanistan	1965	0.0
Afghanistan	1966	0.0
Afghanistan	1967	0.0