

Для части по батчевой обработке (sqoop)

1. Создать отдельную БД в Hive
2. Импортировать в нее три любые таблицы из базы pg_db в PostgreSQL используя SQOOP. Для каждой таблицы используйте отдельный формат хранения -- ORC/Parquet/AVRO Рекомендую захватить таблицу sales_large -- там порядка 10 миллионов записей, она будет достаточно репрезентативна для проверки компрессии.
3. Найдите папки на файловой системе куда были сохранены данные. Посмотрите их размер.
4. Сделайте несколько произвольных запросов к этим таблицам.
5. [факультативно] Сделайте п.2 С использованием компрессии. Как включить компрессию см в ссылки в описаниях формата хранения.
6. [факультативно] Посмотрите на размеры файлов и время выполнения запросов аналогично с п4 и п5 и сравните данные с компрессией/без компрессии.
7. [продвинутое задание] Повторите задание предварительно залив в PostgreSQL один датасет размером не менее 500Мб. Обратите внимание на влияние компрессии.

1. Создал БД pg_db_test

2,3,5. Создал таблицу character в формате parquet без компрессии

```
1 SET parquet.compression=none;
2 CREATE TABLE `pg_db_test.character` (
3     `charid` string,
4     `charname` string,
5     `abbrev` string,
6     `description` string,
7     `speechcount` int)
8 STORED AS PARQUET;
```

```
sqoop import --connect jdbc:postgresql://node3.novalocal/pg_db --username exporter -P --table
character --hive-import --hive-database pg_db_test --as-parquetfile
```

```
[student2_11@manager ~]$ hdfs dfs -du -h -s /user/hive/warehouse/pg_db_test.db/character
69.6 K 208.8 K /user/hive/warehouse/pg_db_test.db/character
```

Итоговый размер около 69.6 кб

4. Попробовал различные запросы на таблице character:

```
SELECT SUM(speechcount) FROM pg_db_test.character WHERE abbrev='First Musician'
```

```
INFO : MapReduce Jobs Launched:
INFO : Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 8.87 sec HDFS Read: 40088 HDFS Write: 3 SUCCESS
INFO : Total MapReduce CPU Time Spent: 8 seconds 870 msec
INFO : Completed executing command(queryId=hive_20200311145454_25daf121-983d-4a97-a0f3-92a63288ac24); Time taken: 26.17 seconds
INFO : OK
```

Query History Saved Queries Results (1)

_c0

1 15

```
SELECT count(*) FROM pg_db_test.character WHERE description IS NOT NULL
```

```
INFO : MapReduce Jobs Launched:
INFO : Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 8.07 sec HDFS Read: 35820 HDFS Write: 4 SUCCESS
INFO : Total MapReduce CPU Time Spent: 8 seconds 70 msec
INFO : Completed executing command(queryId=hive_20200311145757_0b85a826-44f8-428e-b7ef-ad5ab05dd6ab); Time taken: 24.219 seconds
INFO : OK
```

Query History Saved Queries Results (1)

_c0

1	967
---	-----

SELECT charname, SUM(speechcount) FROM pg_db_test.character WHERE description IS NOT NULL GROUP BY charname

	charname	_c1
1	Aaron	57
2	Abhorson	13
3	Abraham	5
4	Achilles	74
5	Adam	10
6	Adrian	9

Импорт таблицы sales_large в формате parquet

```
sqoop import --connect jdbc:postgresql://node3.novalocal/pg_db --username exporter -P --table
sales_large --hive-import --hive-database pg_db_test --hive-table sales_large_parquet --as-
parquetfile -m 1
```

```
20/03/13 11:29:17 INFO mapreduce.ImportJobBase: Transferred 458.7065 MB in 266.1453 seconds
(1.7235 MB/sec)
20/03/13 11:29:17 INFO mapreduce.ImportJobBase: Retrieved 12000000 records.
[student2_11@manager ~]$ hdfs dfs -du -h -s /user/hive/warehouse/pg_db_test.db/sales_large_p
arquet
458.7 M 1.3 G /user/hive/warehouse/pg_db_test.db/sales_large_parquet
[student2_11@manager ~]$
```

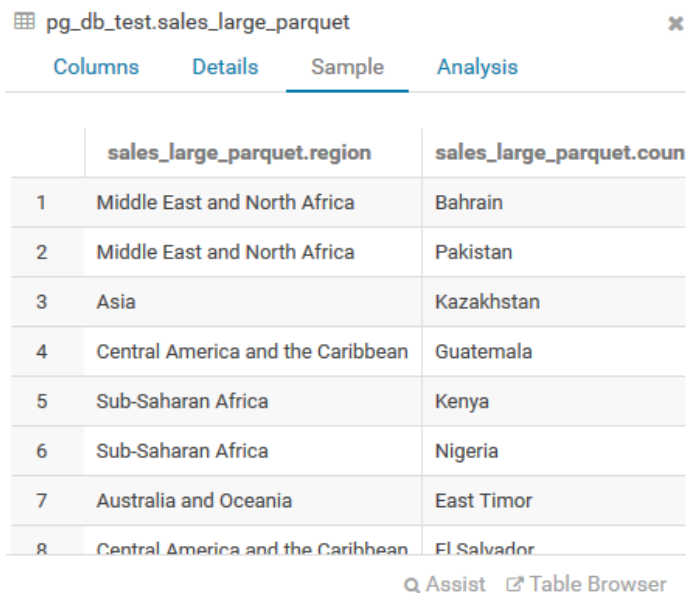
Размер таблицы 458.7 Мб.

pg_db_test.sales_large_parquet

Columns	Details	Sample	Analysis
COLUMN_STATS_ACCURATE false			
avro.schema.url	hdfs://manager.novalocal:8020 /user/hive/warehouse /pg_db_test.db/sales_large_parquet /metadata/schemas/1.avsc		
kite.compression.type	snappy		
numFiles	0		
numRows	-1		
rawDataSize	-1 B		
totalSize	0 B		

Как видно на скриншоте по умолчанию для parquet установлена компрессия snappy. Если пробовать импорт с ключом -z компрессия все равно остается snappy.

Проверим читабельность импортируемых данных:



	sales_large_parquet.region	sales_large_parquet.coun
1	Middle East and North Africa	Bahrain
2	Middle East and North Africa	Pakistan
3	Asia	Kazakhstan
4	Central America and the Caribbean	Guatemala
5	Sub-Saharan Africa	Kenya
6	Sub-Saharan Africa	Nigeria
7	Australia and Oceania	East Timor
8	Central America and the Caribbean	El Salvador

Импорт таблицы sales_large в формате avro

```
sqoop import --connect jdbc:postgresql://node3.novalocal/pg_db --username exporter -P --table sales_large --as-avrodatafile --target-dir /user/hive/warehouse/pg_db_test.db/sales_large_avro -m 1
```

```
20/03/13 11:47:39 INFO mapreduce.ImportJobBase: Transferred 1.4841 GB in 251.6029 seconds (6.0401 MB/sec)
20/03/13 11:47:39 INFO mapreduce.ImportJobBase: Retrieved 12000000 records.
[student2_ll@manager ~]$ hdfs dfs -du -h -s /user/hive/warehouse/pg_db_test.db/sales_large_avro
1.5 G    4.5 G    /user/hive/warehouse/pg_db_test.db/sales_large_avro
```

Размер таблицы 1.5 Gb.

В Hue таблицу не видно, поэтому создаем там EXTERNAL TABLE:

```
1 CREATE EXTERNAL TABLE pg_db_test.sales_large_avro (
2   region string,
3   country string,
4   itemtype string,
5   saleschannel string,
6   orderpriority string,
7   orderdate string,
8   orderid int,
9   shipdate string,
10  unitssold decimal(10,0),
11  unitprice decimal(10,0),
12  unitcost decimal(10,0),
13  totalrevenue decimal(10,0),
14  totalcost decimal(10,0),
15  totalprofit decimal(10,0))
16 STORED AS AVRO;
```

Проверим читабельность импортируемых данных:

pg_db_test.sales_large_avro

Columns Details Sample Analysis

	sales_large_avro.region	sales_large_avro.country
1	Middle East and North Africa	Bahrain
2	Middle East and North Africa	Pakistan
3	Asia	Kazakhstan
4	Central America and the Caribbean	Guatemala
5	Sub-Saharan Africa	Kenya
6	Sub-Saharan Africa	Nigeria
7	Australia and Oceania	East Timor
8	Central America and the Caribbean	El Salvador

Импорт таблицы sales_large в формате avro с компрессией gzip

```
sqoop import --connect jdbc:postgresql://node3.novalocal/pg_db --username exporter -P --table sales_large --as-avrodatafile --target-dir /user/hive/warehouse/pg_db_test.db/sales_large_avro -z -m 1
```

Размер таблицы получается около 450 Мб.

Как сделать импорт в формате orc не разобрался.

Для части по потоковой обработке (Flume)

1. Посмотреть что не так в конфигурации NetCat Flume agent которого я сделал. Описать и аргументировать.

2. Создать любой Flume поток используя Flume сервис соответствующего номера.

Тип источника источник -- exec

Тип канала -- file

Тип слива -- hdfs

3. [Продвинутый вариант] Сделать то-же самое используя несколько сливов в разные места, например в HDFS и в HIVE одновременно

4. [Продвинутый вариант] Повторить стандартный пример с выборкой сообщений из Twitter. Перед этим связаться со мной :)

Flume конфиг:

```
# Naming the components on the current agent
```

```
Flume11.sources = Exec
```

```
Flume11.channels = FileChannel
```

```
Flume11.sinks = HDFSSink
```

```
# source exec
```

```
Flume11.sources.Exec.type = exec
```

```
Flume11.sources.Exec.channels = FileChannel
```

```
Flume11.sources.Exec.command = cat /var/log/cron
```

```
# channel
```

```
Flume11.channels.FileChannel.type = file
```

```
Flume11.channels.FileChannel.checkpointDir = /flume/flume11/file-channel/checkpoint
```

```
Flume11.channels.FileChannel.dataDirs = /flume/flume11/file-channel/data
```

```
Flume11.channels.FileChannel.capacity = 1000
```

```
Flume11.channels.FileChannel.transactionCapacity = 100
```

```
# sink
```

```
# HDFS
```

```
Flume11.sinks.HDFSSink.type = hdfs
```

```
Flume11.sinks.HDFSSink.channel = FileChannel
```

```
Flume11.sinks.HDFSSink.hdfs.path = /user/student2_11/flume/%y-%m-%d
```

```
Flume11.sinks.HDFSSink.hdfs.filePrefix = hdfs-audit-
```

```
# File size to trigger roll, in bytes (256Mb)
```

```
Flume11.sinks.HDFSSink.hdfs.useLocalTimeStamp = true
```

```
Flume11.sinks.HDFSSink.hdfs.rollSize = 268435456
```

```
Flume11.sinks.HDFSSink.hdfs.rollInterval = 0
```

```
Flume11.sinks.HDFSSink.hdfs.rollCount = 0
```

```
Flume11.sinks.HDFSSink.hdfs.fileType = SequenceFile
```

```
Flume11.sinks.HDFSSink.hdfs.codec = gzip
```

Проверил записи в указанной папке слива:

Команда cat /var/log/cron работает и записи создаются, а tailf почему-то работать отказалась.

```
[student2_11@manager ~]$ hdfs dfs -ls /user/student2_11/flume
Found 3 items
drwxrwxrwx - flume student2_11 0 2020-03-13 11:47 /user/student2_11/flume/20-03-12
drwxrwxrwx - flume student2_11 0 2020-03-13 12:36 /user/student2_11/flume/20-03-13
drwxrwxrwx - flume student2_11 0 2020-03-15 12:51 /user/student2_11/flume/20-03-15
```