

Урок 5. Поточковая обработка данных. Flume/Flink/SparkStreaming

Выполнил Колеганов Н.Д.

Задание

Для части по SQOOP

Провести импорт таблицы из вашего сервера БД в Hadoop с использованием SQOOP в любых двух вариантах из перечисленных ниже.

- в Hive-таблицу (--hive-import)
- в HDFS в формате avro (--as-avrodatafile)
- в HDFS в формате sequencefile (--as-sequencefile)

Если у вас нет своего сервера то можно использовать тот Postgres, который я показал на лекции. Пароль expoter_pass

Для части по потоковой обработке (Flume)

- Создать Flume-агент с именем, соответствующим имени своего пользователя (например Flume4_20)
- Создать любой Flume поток используя Flume сервис соответствующего номера.
 - Тип источника источник – exes
 - Тип канала – memory
 - Тип слива – hdfs
- Убедиться что данные поступают в слив.
- Создать поверх данных в hdfs таблицу через которую можно просмотреть полученные данные.
- [Продвинутый вариант] Сделать то-же самое используя несколько сливов в разные места, например в HDFS и в Hive одновременно
- [Продвинутый вариант] Повторить стандартный пример с выборкой сообщений из Twitter.

Создал БД pg_db_test1

Создал таблицу character в формате parquet без компрессии

```
1.37s pg_db_test1 text ?
1 CREATE DATABASE pg_db_test1;
2
3 SET parquet.compression=none;
4
5 CREATE TABLE pg_db_test1.character (
6     charid string,
7     charname string,
8     abbrev string,
9     description string,
10    speechcount int)
11 STORED AS PARQUET;
```

```
sqoop import --connect jdbc:postgresql://node3.novalocal/pg_db --username exporter -P --table character --hive-import --hive-database pg_db_test1 --as-parquetfile
```

```
[student4_2@manager ~]$ hdfs dfs -du -h -s /user/hive/warehouse/pg_db_test1.db/character
69.6 K  208.8 K /user/hive/warehouse/pg_db_test1.db/character
```

Итоговый размер около 69.6 кб

Попробовал различные запросы на таблице character:

```
SELECT SUM(speechcount) FROM pg_db_test1.character WHERE abbrev='First Musician'
```

The screenshot shows two Hive queries and their results. The first query is `SELECT SUM(speechcount) FROM pg_db_test1.character WHERE abbrev='First Musician'`, which returns a single value of 15. The second query is `SELECT count(*) FROM pg_db_test1.character WHERE description IS NOT NULL`, which returns a single value of 967. Both queries are shown with their execution logs and the results are displayed in a table format.

id	1	15
1	15	

id	1	967
1	967	

```
SELECT count(*) FROM pg_db_test1.character WHERE description IS NOT NULL
```

```
SELECT charname, SUM(speechcount) FROM pg_db_test1.character WHERE description IS NOT
```

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

```
NULL GROUP BY charname
```

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_test1 --hive-table sales_large_parquet --as-parquetfile -m 1
```

```
[student4_2@manager ~]$ sqoop import --connect jdbc:postgresql://node3.novalocal/pg_db --username exporter -P --table sales_large --hive-import --hive-database pg_db_test1 --hive-table sales_large_parquet --as-parquetfile -m 1
```

```
[student4_2@manager ~]$ hdfs dfs -du -h -s /user/hive/warehouse/pg_db_test1.db/sales_large_parquet  
458.7 M 1.3 G /user/hive/warehouse/pg_db_test1.db/sales_large_parquet
```

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:

pg_db_test.sales_large_parquet		✕	
Columns		Details	
Sample		Analysis	
	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
```

```
[student4_2@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;
```

pg_db_test.sales_large_avro

Проверим читабельность
✖ импортируемых данных:

	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.

Для части по потоковой обработке (Flume)

Создал Flum

The screenshot shows the Flume web interface for a cluster named 'Flume4_2' (GeekBrains Cluster). The 'Status' tab is selected, displaying 'Health Tests'. A 'Create Trigger' button is visible. The 'Agent Health' section shows 'Healthy Agent: 1. Concerning Agent: 0. Total Agent: 1. Percent healthy: 100.00%. Percent healthy or concerning: 100.00%.' with a 'Suppress...' link. On the right, a 'Charts Lib' section is partially visible, showing 'Critic' and 'VENTS'.

Flume конфиг:

Agent Name	Agent Default Group 
	Flume4_2
Configuration File	Agent Default Group 
	<pre> # Naming the components on the current agent Flume4_2.sources = ExecSource Flume4_2.channels = MemChannel Flume4_2.sinks = HdfsSink # Describing/Configuring the source Flume4_2.sources.ExecSource.type = exec Flume4_2.sources.ExecSource.command = tailf /var/log/messages Flume4_2.sources.ExecSource.interceptors = TimestampInterceptor Flume4_2.sources.ExecSource.interceptors.TimestampInterceptor.type = timestamp # Describing/Configuring the HDFS sink Flume4_2.sinks.HdfsSink.type = hdfs Flume4_2.sinks.HdfsSink.hdfs.path= /flume/flume4_2/log/%y-%m-%d/ Flume4_2.sinks.HdfsSink.hdfs.filePrefix = events # Describing/Configuring the channel Flume4_2.channels.MemChannel.type = memory Flume4_2.channels.MemChannel.capacity = 10000 Flume4_2.channels.MemChannel.transactionCapacity = 10 # Bind the source and sink to the channel Flume4_2.sources.ExecSource.channels = MemChannel Flume4_2.sinks.HdfsSink.channel = MemChannel </pre>

Naming the components on the current agent

Flume4_2.sources = ExecSource

Flume4_2.channels = MemChannel

Flume4_2.sinks = HdfsSink

Describing/Configuring the source

Flume4_2.sources.ExecSource.type = exec

Flume4_2.sources.ExecSource.command = tailf /var/log/messages

Flume4_2.sources.ExecSource.interceptors = TimestampInterceptor

Flume4_2.sources.ExecSource.interceptors.TimestampInterceptor.type = timestamp

Describing/Configuring the HDFS sink

Flume4_2.sinks.HdfsSink.type = hdfs

Flume4_2.sinks.HdfsSink.hdfs.path= /flume/flume4_2/log/%y-%m-%d/

Flume4_2.sinks.HdfsSink.hdfs.filePrefix = events

Describing/Configuring the channel

Flume4_2.channels.MemChannel.type = memory

Flume4_2.channels.MemChannel.capacity = 10000

Flume4_2.channels.MemChannel.transactionCapacity = 10

Bind the source and sink to the channel

Flume4_2.sources.ExecSource.channels = MemChannel

Flume4_2.sinks.HdfsSink.channel = MemChannel

Проверил статус:

Restart Command

✓ Starting 1 roles on service Successfully started 1 roles on service.		Jun 26, 1:55:46 PM	22.51s
✓ Execute command Start this Agent on role Agent (node3) Successfully started process.	Agent (node3)	Jun 26, 1:55:46 PM	22.5s
✓ Start a role Supervisor returned RUNNING.	Agent (node3)	Jun 26, 1:55:46 PM	22.5s

\$> flume/flume.sh stdout stderr Role Log

```
Fri Jun 26 13:55:46 UTC 2020
JAVA_HOME=/usr/java/jdk1.8.0_181-amd64
using agent name Flume4_2
using Flume executable /opt/cloudera/parcels/CDH-5.16.2-1.cdh5.16.2.p0.8/lib/flume-ng/bin/flume-ng
using FLUME_HOME_DIR /var/lib/flume-ng
CONF_DIR=/run/cloudera-scm-agent/process/1837-flume-AGENT
CMF_CONF_DIR=/etc/cloudera-scm-agent
Adding hadoop-conf to classpath
Adding hbase-conf to classpath
Adding /usr/lib/flume-ng/plugins.d:/var/lib/flume-ng/plugins.d to plugins path
```

[Full log file](#)

Проверил ведутся ли записи

```
[student4_2@manager ~]$ hdfs dfs -ls /flume/flume4_2
Found 1 items
drwxr-xr-x  - flume flume          0 2020-06-26 13:50 /flume/flume4_2/log
[student4_2@manager ~]$ hdfs dfs -ls /flume/flume4_2/log
Found 1 items
drwxr-xr-x  - flume flume          0 2020-06-26 14:00 /flume/flume4_2/log/20-06-26
[student4_2@manager ~]$ hdfs dfs -cat /flume/flume4_2/log/20-06-26/events.1593179755213
SEQ[org.apache.hadoop.io.LongWritable"org.apache.hadoop.io.BytesWritable
BJun 26 11:01:01 node3 systemd: Starting Session 5717 of user root.
AJun 26 12:01:01 node3 systemd: Started Session 5718 of user root.
BJun 26 12:01:01 node3 systemd: Starting Session 5718 of user root.
AJun 26 13:01:01 node3 systemd: Started Session 5719 of user root.
BJun 26 13:01:01 node3 systemd: Starting Session 5719 of user root.
FJun 26 13:47:42 node3 systemd: Created slice User Slice of student4_2.
AJun 26 13:47:42 node3 systemd: Starting User Slice of student4_2.
JJun 26 13:47:42 node3 systemd-logind: New session 5720 of user student4_2.
GJun 26 13:47:42 node3 systemd: Started Session 5720 of user student4_2.
HJun 26 13:47:42 node3 systemd: Starting Session 5720 of user student4_2.
[student4_2@manager ~]$
```