CS542000 Cloud Programming Lab2: Hbase, Hive & Pig

Josh Kao

National Tsing Hua University

2015/06/01

Objective

- To get familiar with
    1. Using HBase with API
    2. Querying data by Hive, Pig

Outline I

What is Hbase?

"*HBase is an open source, non-relational, distributed database modeled after Google's BigTable and written in Java. It is developed as part of Apache Software Foundation's Apache Hadoop project and runs on top of HDFS (Hadoop Distributed Filesystem), providing BigTable-like capabilities for Hadoop.*"

— Wikipedia

What is Hive?

"*Apache Hive is a data warehouse infrastructure built on top of Hadoop for providing data summarization, query, and analysis.*"

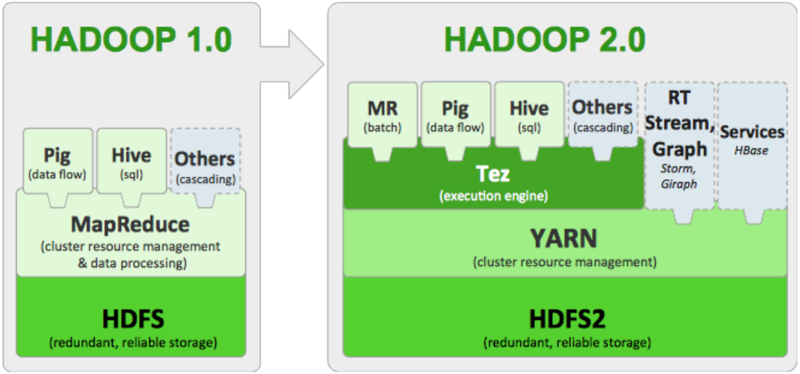
— Wikipedia

What is Pig?

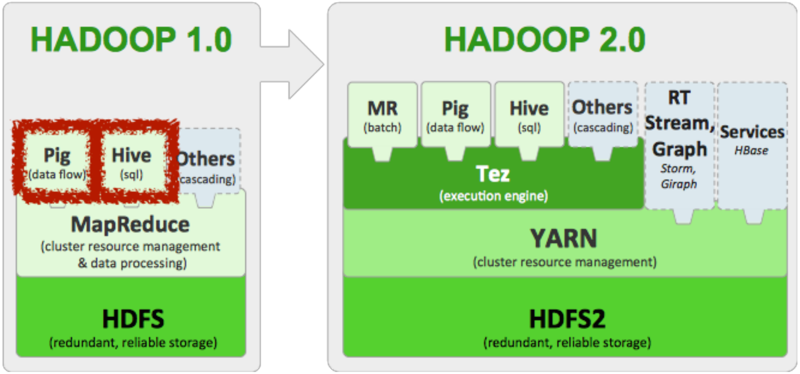
"*Pig is a high-level platform for creating MapReduce programs used with Hadoop. The language for this platform is called Pig Latin. Pig Latin abstracts the programming from the Java MapReduce idiom into a notation which makes MapReduce programming high level, similar to that of SQL for RDBMS systems. Pig Latin can be extended using UDF (User Defined Functions).*"

— Wikipedia

Hadoop ecosystem evolution


HADOOP 1.0
Pig (data flow)
Hive (sql)
Others (cascading)
MapReduce (cluster resource management & data processing)
HDFS (redundant, reliable storage)
HADOOP 2.0
MR (batch)
Pig (data flow)
Hive (sql)
Others (cascading)
Tez (execution engine)
RT Stream, Graph Storm, Giraph
Services HBase
YARN (cluster resource management)
HDFS2 (redundant, reliable storage)

Today's mission

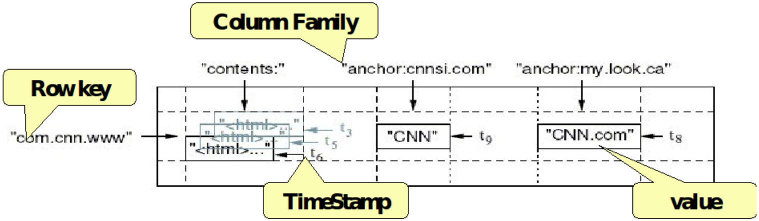

HADOOP 1.0
Pig (data flow)
Hive (sql)
Others (cascading)
MapReduce (cluster resource management & data processing)
HDFS (redundant, reliable storage)
HADOOP 2.0
MR (batch)
Pig (data flow)
Hive (sql)
Others (cascading)
RT Stream, Graph Storm, Giraph
Services HBase
Tez (execution engine)
YARN (cluster resource management)
HDFS2 (redundant, reliable storage)

Outline I

Logical Data Model

- Tables are sorted by **row key**
- Table schema only defines its column families
  - Each family consists of any number of columns
  - Value of each column consists of any number of version (timestamp)
  - Columns can be added dynamically
  - #Columns of each row can be different with other rows
  - Only one data type: byte[]

Logical Data Model



Physical Data Model

- Columns in the same column family would be group in the same physical storage, and would be sorted by column

How to access data?

- Doesn't provide SQL Language
- Access data by
  1. **getRow()** : get one row range data, besides, user can get specified timestamp data
  2. **scan()**: get data from whole table

Using HBase

- `$ hbase shell [YOUR_SCRIPT]`
- Cause we share the same database, please name your table with your ID as the prefix. e.g. 103065566_helloTable

Scenario

- **List** all tables in HBase
  - `> list`
- **Create** a new table
  - `> create '{TABLE_NAME}', '{COLUMN_FAMILY1}', ···`
- **Put** value
  - `> put '{TABLE_NAME}', '{ROW}', '{COLUMN_FAMILY[:QUALIFIER]}', '{VALUE}'`

Scenario

- **Get** value
  - `> get '{TABLE_NAME}', '{ROW}'[,'{COLUMN1}',···]`
- **Scan** table
  - `> scan '{TABLE_NAME}'[, optional settting]`

Scenario

- **Delete** value

  -
    ```
    > delete '{TABLE_NAME}', '{ROW}',
            '{COLUMN_FAMILY[:QUALIFIER]}'
    ```
- **Remove** table
  - `> disable '{TABLE_NAME}'`
  - `> drop '{TABLE_NAME}'`
- Learn more from help :D
  - `> help`

Why we need Hive/Pig?

- Need high-level languages
  - Writing java programs for everythign is verbose and slow
  - Not everyone wants to (or can) write java

Outline I

Hive

- Query language is **Hive QL**, variant of SQL
- Tables stored on HDFS as flat files
- Not designed for online transaction processing
- Does not offer real-time queries and row level updates.

Primitive Types

- Integers
  - **TINYINT** - 1 byte integer
  - **SMALLINT** - 2 byte integer
  - **INT** - 4 byte integer
  - **BIGINT** - 8 byte integer
- Boolean type
  - **BOOLEAN** - TRUE/FALSE
- Floating point numbers
  - **FLOAT** - single precision
  - **DOUBLE** - double precision
- String type
  - **STRING** - sequence of characters in a specified character set

Complex Types

- Structs
- Maps
- Arrays

Using Hive

-
  ```
  $ hive -hiveconf hhase.master={HBase_MASTER}:{PORT}
      --auxpath /opt/hive/lib/hive-hbase-handler-0.11.0.jar,
      /opt/hive/lib/hbase-0.94.27.jar,
      /opt/hive/lib/zookeeper-3.4.6.jar
      [-f {YOUR_SCRIPT}.q]
  ```
- Learn basic usage of SQL by yourself
  - W3C School
  - Hive Tutorial https://cwiki.apache.org/confluence/display/Hive/Tutorial

Load HBase tables to Hive

- *hive* > CREATE EXTERNAL TABLE {TABLE_NAME}
  > ({SCHEMA}) STORE BY
  > 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
  > WITH SERDEPROPERTIES("hbase.columns.mapping" =
  > "{CF1}:{Q1},{CF1}:{Q2},{CF2}: · · · "
  > TBLPROPERTIES("hbase.table.name" = "{TABLE_NAME}")

Outline I

Using Pig

- `$ pig [-x local {YOUR_SCRIPT}.pig]`
  `-Dpig.additional.jars=/opt/hbase/lib/protobuf-java-2.4.0a.jar`
- Pig Tutorial https://pig.apache.org/docs/r0.7.0/tutorial.html

Load HBase tables to Pig

- $grunt >$
  ```
  {VAR}= LOAD '{HBASE_PATH}'
      USING org.apache.pig.backend.hbase.HBaseStorage
      ('{CF1}:{Q1} {CF2}:{Q2} ···','-loadKey true')
      AS ({FIELD1}:{TYPE},···);
  ```
- $grunt >$
  ```
  math = LOAD 'hbase://103065566_math'
      USING org.apache.pig.backend.hbase.HBaseStorage
      ('grade:name grade:math','-loadKey true')
      AS (id:CHARARRAY, name:CHARARRAY, math:DOUBLE);
  ```

Pig Example

```
> visits = LOAD '/data/visits' AS (user, url, time);
> visits = FOREACH visits GENERATE user, Canonicalize(url),
              time;
> pages = LOAD '/data/pages' AS (url, pagerank);
> vp = JOIN visits BY url, pages BY url;
> userVisits = GROUP vp BY user;
```

Pig Example (Continue)

```
> userPageranks = FOREACH userVisits GENERATE user,
                  AVG(vp.pagerank) AS avgpr;
> goodUsers = FILTER userPageranks BY avgpr > '0.5';
> DUMP goodUsers;
> STORE goodUsers INTO '/data/good_users';
```

Outline I

Problem Description

- Part 1 - Load data to HBase by API
    1. Modify sample code to load file math to HBase
    2. This table named {STUDENT_ID}_math with column-family **grade** and qualifier **name**,**math**
    3. Load file eng to HBase
    4. This table named {STUDENT_ID}_eng with column-family **grade** and qualifier **name**,**eng**

Problem Description

- Part 2 - Query data by Hive or Pig from HBase
  Write a script to satisfy the following statement
  1. Load table from HBase to Hive(Pig)
  2. JOIN these two tables to a new table named {STUDENT_ID}_score
  3. CREATE a new column named **avg** $(= (eng + math)/2\ )$
  4. Find #students failed ($avg < 60$)
  5. Show the name of top 5 students

Outline I

Login to server

- **Host**: 140.114.91.199 (with 1 master, 8 slaves)
- **Account**: {YOUR_STUDENT_ID}
- **Password**: cloud5566 (default)

It's recommended to use ***passwd*** to change your password.

Grading

- Part 1 - Load data to HBase by API

20%	Load data to table {STUDENT_ID}_math
20%	Load data to table {STUDENT_ID}_eng

- Part 2 - Query data by Hive or Pig from HBase

30%	Count #students failed
30%	Find top 5 students

Outline I

Reference

- http://hortonworks.com/blog/apache-hadoop-2-is-ga/
- http://contest.trendmicro.com/2014/cn/material/hbase.pdf
- http://www.datascience-labs.com/hbase/
- http://gethue.com/hadoop-tutorial-use-pig-and-hive-with-hbase/