Project 1: Hive Demo

Chengen Xie
Yebin Han
You Wu
Yuan Meng

Hive Introduction

- Apache Hive
  - Hive is a software running on HDFS
  - Hive resembles a traditional relational database
  - Hive translates HiveQL (SQL-like) language into map/reduce job to visit data file on HDFS

Hive Introduction - HiveQL

- HiveQL
  - A SQL-like language.
  - “Heavily influenced by MySQL”
  - Provides operations (such as multitable inserts) inspired by MapReduce.

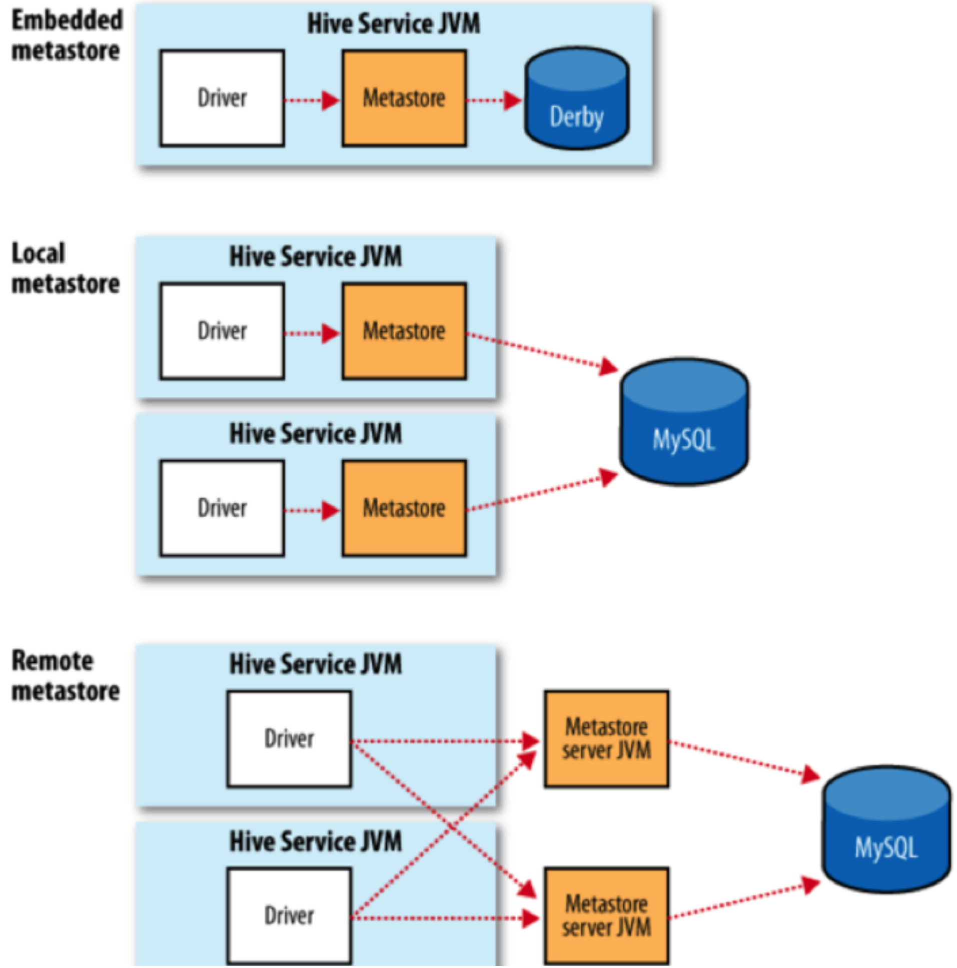
(White, Tom. *Hadoop: The definitive guide*. " O'Reilly Media, Inc.", 2012.)

Hive Introduction - Metastore

- Hive Metastore
  - A traditional relational database to store the Hive metadata (database ID, Table ID, Table InputFormat and etc.)
  - Three ways of Metastore
    - Derby database (default)
    - local standalone database
    - remote database

(http://stackoverflow.com/questions/17065672/what-does-the-hive-metastore-and-name-node-do-in-a-cluster)

Hive

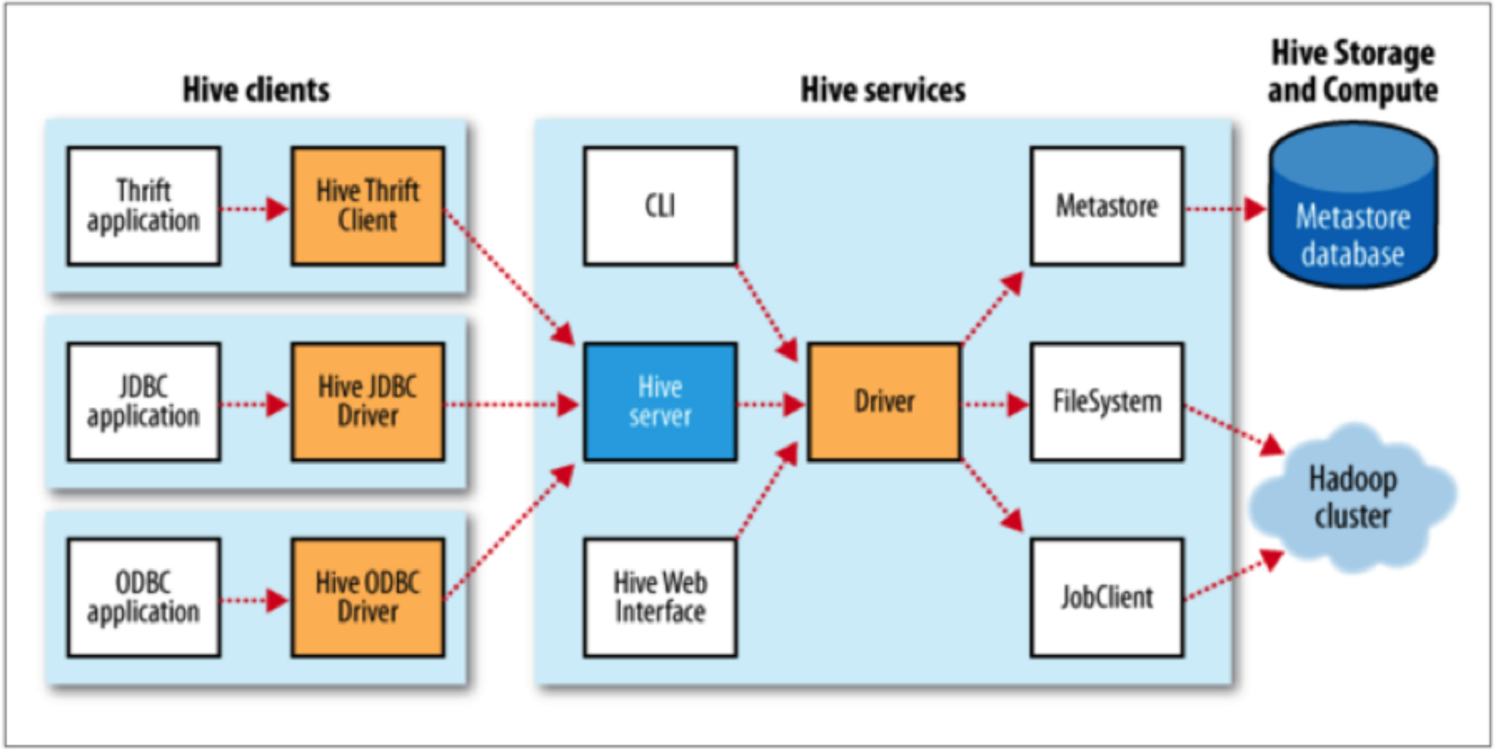


(White, Tom. *Hadoop: The definitive guide*. " O'Reilly Media, Inc.", 2012.)

Hive Introduction - Ways to use Hive

- Hive shell (CLI, command line interface) or Hive script
- Hive services
  - Thrift, JDBC, ODBC interface
- Hive Web Interface (HWI)
  - Operate Hive through web browser

Hive Introduction - Ways to use Hive



(White, Tom. *Hadoop: The definitive guide*. " O'Reilly Media, Inc.", 2012.)

Hive Introduction - UDF

- UDF (User Defined Function)
  - Single row input and single row output
    - e.g. stripping characters from the end of the strings
- UDAF (User Defined Aggregate Function)
  - Multiple rows input and single row output
    - e.g. maximum of a collection of integers
- UDTF (User Defined Table Generating Function)
  - Single or multiple rows as input, a table as output

Application Workload Example

- Data source: NYSE Apple stock financials
- Example operations:
  - Create table
  - Load data
  - Calculate average close price for each year
    - HiveQL
    - User Defined Function

Application Workload Example

- Create table
  - CREATE TABLE APPLE (YEAR INT, MONTH INT, DAY INT,)
  - ROW FORMAT DELIMITED
  - FIELDS TERMINATE BY ‘,’;

Application Workload Example

- Load Data
  - LOAD DATA LOCAL INPATH ‘apple.csv’
  - OVERWRITE TABLE APPLE;

Application Workload Example

- Calculate average close price for each year
  - SELECT YEAR, AVG(CLOSE)
  - FROM APPLE
  - GROUP BY YEAR;

1980	30.48153846153846
1981	24.38634920634921
1982	19.1397233201581
1983	37.52484126984126
1984	26.869960474308296
1985	20.378814229249013
1986	32.38739130434783
1987	53.82268774703557
1988	41.55588932806324
1989	41.615000000000016
1990	37.50201581027668
1991	52.45154150197629
1992	54.80366141732283
1993	41.06324110671936
1994	34.05222222222221
1995	40.62305555555553
1996	25.048110236220477
1997	18.03237154150197
1998	30.512380952380944
1999	57.65948412698414
2000	71.86388888888888
2001	20.165322580645178
2002	19.12805555555556
2003	18.5217857142857
2004	35.421468253968264
2005	52.34968253968254
2006	70.98760956175303
2007	128.3890836653386
2008	142.31375494071145
2009	146.61908730158729
2010	259.957619047619
2011	364.06142857142896
2012	576.65272
2013	473.1281349206351
2014	295.14261904761906
2015	116.04176470588234

Application Workload Example - UDF

```
package org.apache.hive.cs516;

import java.util.ArrayList;
import org.apache.hadoop.hive.ql.exec.UDAF;
import org.apache.hadoop.hive.ql.exec.UDAFEvaluator;
import org.apache.hadoop.io.DoubleWritable;

public class Average extends UDAF {
   public static class MaximumIntUDAFEvaluator implements
UDAFEvaluator {
            private DoubleWritable result;
            private ArrayList<Double> close;
            public void init() {
               result = new DoubleWritable();
               close = new ArrayList<Double>();
            }

   public boolean iterate(DoubleWritable value) {
            if(value!=null){
               close.add(value.get());
            }

            return true;
   }
```

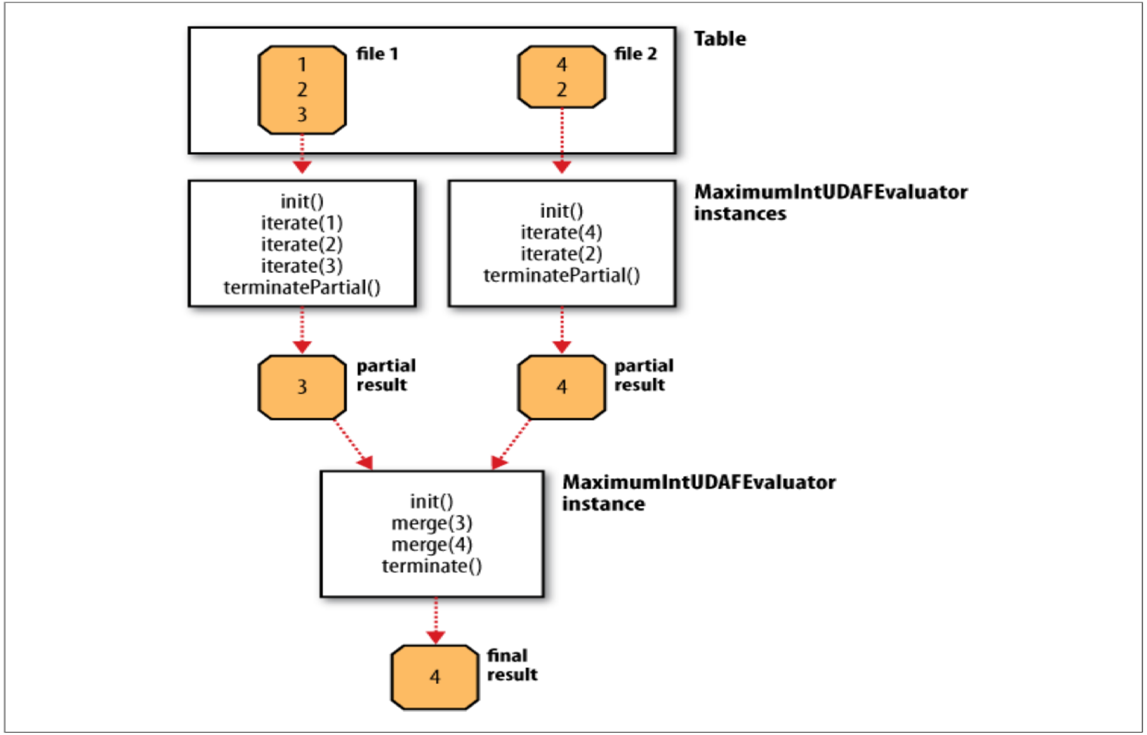
```
public ArrayList<Double> terminatePartial() {

               return close;
            }
            public boolean merge(ArrayList<Double> other) {
               for(Double s : other){
                        close.add(s);
               }
               return true;
            }
            public DoubleWritable terminate() {
               double sum = 0;
               for (Double e : close) {
                        sum += e;
               }
               result.set(sum / close.size());
               return result;
            }
   }
}
```

(based on source code in White, Tom. *Hadoop: The definitive guide*. " O'Reilly Media, Inc.", 2012.)

Application Workload Example - UDF

- Add UDF into Hive:
    - CREATE TEMPORARY FUNCTION average AS 'org.apache.hive.cs516.Average';
    - SELECT average(CLOSE) FROM APPLE;

Application Workload Example - UDF



(White, Tom. *Hadoop: The definitive guide*. " O'Reilly Media, Inc.", 2012.)