JOHNS HOPKINS
CAREY BUSINESS SCHOOL

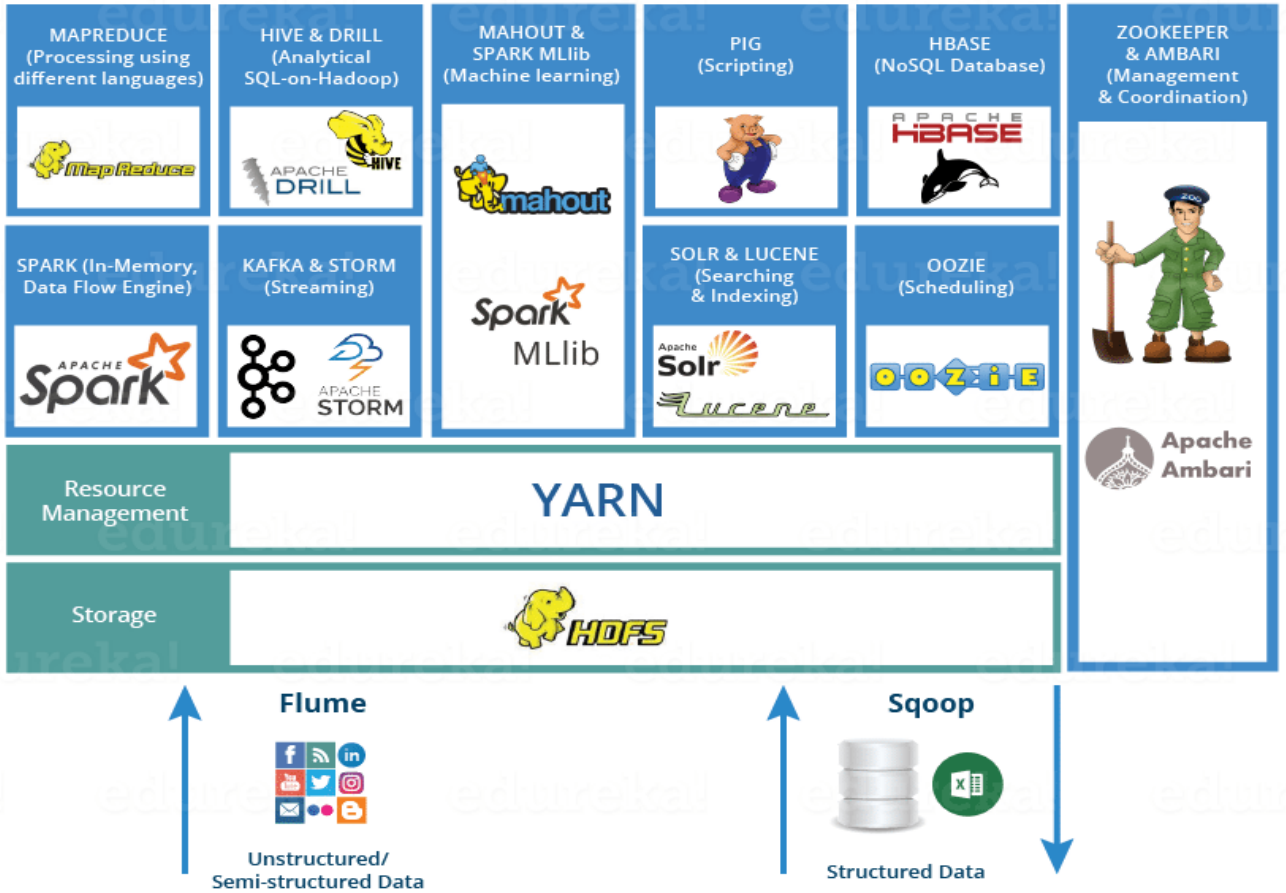
Hive and Dictionary-Based Sentiment

BU.330.740 Large Scale Computing on the Cloud

Minghong Xu, PhD.
Associate Professor

Hadoop Overview

- An open source framework for writing and running distributed applications that process large amount of data
- Google: first to publicize MapReduce for scaled data processing
- Doug Cutting: develop the first version of Hadoop



Key Components in this Course

- Distributed File System: HDFS
- Operating System: YARN
- Original Distributed Processing Engine: MapReduce
- Improved MapReduce: Spark
- Distributed Query Language: Hive
- Distributed Scripting Language: Apache Pig

JOHNS HOPKINS
CAREY BUSINESS SCHOOL

Hive: Distributed Query Language

What is Hive

- Provide an SQL (structured query language) dialect for querying data stored in HDFS, and other filesystems that integrate with Hadoop, such as S3, HBase
- Translates most queries to MapReduce jobs
- Explore the scalability of Hadoop, while presenting a familiar SQL abstraction

Database in Hive

- A *catalog* or *namespace* of tables
- Organize tables into logical group
- Operations
  - CREATE
  - SHOW
  - USE
  - DROP

```
hive> CREATE DATABASE financials;
```

```
hive> CREATE DATABASE IF NOT EXISTS financials;
```

```
hive> SHOW DATABASES;
default
financials
```

```
hive> SHOW DATABASES LIKE 'h.*';
human_resources
hive> ...
```

```
hive> USE financials;
```

```
hive> DROP DATABASE IF EXISTS financials;
```

Table

- CREATE TABLE statement follows SQL conventions with more flexibility

```
CREATE TABLE IF NOT EXISTS mydb.employees (
  name         STRING COMMENT 'Employee name',
  salary       FLOAT  COMMENT 'Employee salary',
  subordinates ARRAY<STRING> COMMENT 'Names of subordinates',
  deductions   MAP<STRING, FLOAT>
               COMMENT 'Keys are deductions names, values are percentages',
  address      STRUCT<street:STRING, city:STRING, state:STRING, zip:INT>
               COMMENT 'Home address')
COMMENT 'Description of the table'
TBLPROPERTIES ('creator'='me', 'created_at'='2012-01-02 10:00:00', ...)
LOCATION '/user/hive/warehouse/mydb.db/employees';
```

```
CREATE TABLE IF NOT EXISTS mydb.employees2
LIKE mydb.employees;
```

```
hive> SHOW TABLES IN mydb;
```

```
hive> SHOW TABLES 'empl.*';
employees
```

```
DROP TABLE IF EXISTS employees;
```

External Table

- Tables are “managed” or “internal” by default
  - Dropping a table removes its data in HDFS
- Use EXTERNAL when creating the table avoids this behavior
  - Dropping an external table removes only its metadata

```
CREATE EXTERNAL TABLE IF NOT EXISTS stocks (
  exchange        STRING,
  symbol          STRING,
  ymd             STRING,
  price_open      FLOAT,
  price_high      FLOAT,
  price_low       FLOAT,
  price_close     FLOAT,
  volume          INT,
  price_adj_close FLOAT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LOCATION '/data/stocks';
```

Load Data and Export Data

- Load from file

```
LOAD DATA LOCAL INPATH '${env:HOME}/california-employees'
OVERWRITE INTO TABLE employees
```

- Create table and load in 1 query

```
CREATE TABLE ca_employees
AS SELECT name, salary, address
FROM employees
WHERE se.state = 'CA';
```

- INSERT … DIRECTORY …

```
INSERT OVERWRITE LOCAL DIRECTORY '/tmp/ca_employees'
SELECT name, salary, address
FROM employees
WHERE se.state = 'CA';
```

Select…From…Where

- SELECT…FROM
  - Retrieve the data from a table
- WHERE
  - Condition/criteria

```
hive> SELECT name, deductions FROM employees;
John Doe    {"Federal Taxes":0.2,"State Taxes":0.05,"Insurance":0.1}
Mary Smith  {"Federal Taxes":0.2,"State Taxes":0.05,"Insurance":0.1}
Todd Jones  {"Federal Taxes":0.15,"State Taxes":0.03,"Insurance":0.1}
Bill King   {"Federal Taxes":0.15,"State Taxes":0.03,"Insurance":0.1}
```

```
SELECT * FROM employees
WHERE country = 'US' AND state = 'CA';
```

```
hive> SELECT name, salary, deductions["Federal Taxes"],
    >   salary * (1 - deductions["Federal Taxes"])
    > FROM employees
    > WHERE round(salary * (1 - deductions["Federal Taxes"])) > 70000;
John Doe    100000.0  0.2   80000.0
```

Group By…Having

- GROUP BY & HAVING
  - Group data from the multiple records
  - Generally used in conjunction with the aggregate functions: avg(), count(), sum()

```
hive> SELECT year(ymd), avg(price_close) FROM stocks
    > WHERE exchange = 'NASDAQ' AND symbol = 'AAPL'
    > GROUP BY year(ymd)
        > HAVING avg(price_close) > 50.0;
1987    53.88968399108163
1991    52.49553383386182
1992    54.80338610251119
1999    57.77071460844979
2000    71.74892876261757
2005    52.401745992993554
...
```

Join

JOIN

- Inner JOIN: matching records in every table

```
hive> SELECT a.ymd, a.price_close, b.price_close
    > FROM stocks a JOIN stocks b ON a.ymd = b.ymd
    > WHERE a.symbol = 'AAPL' AND b.symbol = 'IBM';
2010-01-04      214.01  132.45
2010-01-05      214.38  130.85
2010-01-06      210.97  130.0
2010-01-07      210.58  129.55
2010-01-08      211.98  130.85
2010-01-11      210.11  129.48
...
```

- LEFT OUTER JOIN / RIGHT OUTER JOIN / FULL OUTER JOIN
  - NULL used for missing records

Order By and Sort By

- ORDER BY
  - Total ordering, all data passes through a single reducer
  - May take an unacceptably long time to execute for large data sets

```
SELECT s.ymd, s.symbol, s.price_close
FROM stocks s
ORDER BY s.ymd ASC, s.symbol DESC;
```

- SORT BY
  - Local ordering, order the data only within each reducer

```
SELECT s.ymd, s.symbol, s.price_close
FROM stocks s
SORT BY s.ymd ASC, s.symbol DESC;
```

Distribute By and Cluster By

- DISTRIBUTE BY with SORT BY v.s. CLUSTER BY
  - Controls how map output is divided among reducers
  - Exploit parallelism of SORT BY, yet achieve a total ordering

```
hive> SELECT s.ymd, s.symbol, s.price_close
    > FROM stocks s
    > DISTRIBUTE BY s.symbol
    > SORT BY  s.symbol ASC, s.ymd ASC;
1984-09-07  AAPL  26.5
1984-09-10  AAPL  26.37
1984-09-11  AAPL  26.87
1984-09-12  AAPL  26.12
1984-09-13  AAPL  27.5
1984-09-14  AAPL  27.87
1984-09-17  AAPL  28.62
1984-09-18  AAPL  27.62
1984-09-19  AAPL  27.0
1984-09-20  AAPL  27.12
...
```

```
hive> SELECT s.ymd, s.symbol, s.price_close
    > FROM stocks s
    > CLUSTER BY s.symbol;
2010-02-08  AAPL  194.12
2010-02-05  AAPL  195.46
2010-02-04  AAPL  192.05
2010-02-03  AAPL  199.23
2010-02-02  AAPL  195.86
2010-02-01  AAPL  194.73
2010-01-29  AAPL  192.06
2010-01-28  AAPL  199.29
```

Dictionary-based Sentiment

- A dictionary is prepared to store the polarity values of each lexicons
- For each word of the text present in the dictionary, polarity score calculating by adding to get an overall polarity score
- This method relies heavily on a pre-defined list (or dictionary) of sentiment-laden words
- Also called **lexical approach**, as dictionary also called **lexicon**

Build Sentiment Lexicons

- Manual approach
  - Manually construct a lexicon and tag words in it as positive or negative
- Supervised learning approach
  - Use a few labeled examples
  - Regression, support vector machine, deep learning, etc.
- Unsupervised learning approach
  - Clustering, group words by similarity
  - E.g. “boring” is more likely used together with “tedious”, “didn’t (like)”
  - “tedious” is less likely used together with “exciting”

Sentiment Lexicons

- Harvard General Inquirer:
  - http://www.wjh.harvard.edu/~inquirer/homecat.htm
  - Spreadsheet: http://www.wjh.harvard.edu/~inquirer/inquirerbasic.xls
  - Free for research use
- LIWC (Linguistic Inquiry and Word Count):
  - http://www.liwc.net/
  - With a fee
- Bing Liu Opinion Lexicon
  - https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html
- SentiWordNet
  - https://github.com/aesuli/SentiWordNet

Issues of Sentiment Lexicons

- Disagreements between polarity lexicons

Table 5 Disagreement levels for the sentiment lexicons reviewed above.

	MPQA	Opinion Lexicon	Inquirer	SentiWordNet	LIWC
MPQA	–	33/5402 (0.6%)	49/2867 (2%)	1127/4214 (27%)	12/363 (3%)
Opinion Lexicon		–	32/2411 (1%)	1004/3994 (25%)	9/403 (2%)
Inquirer			–	520/2306 (23%)	1/204 (0.5%)
SentiWordNet				–	174/694 (25%)
LIWC					–

- Can be domain-specific

Issues: Sentence with Aspects

- "The food was great but the service was awful"

- Identify aspects (supervised)
  - Hand-label a small corpus of review sentences with aspect
    - food, décor, service, value, NONE
  - Train a classifier to assign an aspect to a sentence
    - Given this sentence, is the aspect food, décor, service, value, or NONE

Issues: Subtlety and Ordering Effects

- Subtlety:
  - "If you are reading this because it is your darling fragrance, please wear it at home exclusively, and tape the windows shut."
- Thwarted expectations and ordering effects:
  - "This film should be brilliant. It sounds like a great plot, the actors are first grade, and the supporting cast is good as well, and Stallone is attempting to deliver a good performance. However, it can't hold up."
  - "Well as usual Keanu Reeves is nothing special, but surprisingly, the very talented Laurence Fishbourne is not so good either, I was surprised."