HandsOn – **MapReduce**

Quick Introduction To Java and MapReduce

WordCount – Mapper

```
import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCount {

  public static class TokenizerMapper
       extends Mapper<Object, Text, Text, IntWritable>{

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(Object key, Text value, Context context
                    ) throws IOException, InterruptedException {
      StringTokenizer itr = new StringTokenizer(value.toString());
      while (itr.hasMoreTokens()) {
        word.set(itr.nextToken());
        context.write(word, one);
      }
    }
  }
```

WordCount – Reducer

```
  public static class IntSumReducer
       extends Reducer<Text,IntWritable,Text,IntWritable> {
    private IntWritable result = new IntWritable();

    public void reduce(Text key, Iterable<IntWritable> values,
                       Context context
                       ) throws IOException, InterruptedException {
      int sum = 0;
      for (IntWritable val : values) {
        sum += val.get();
      }
      result.set(sum);
      context.write(key, result);
    }
  }

  public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "word count");
    job.setJarByClass(WordCount.class);
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntSumReducer.class);
    job.setReducerClass(IntSumReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```


Run Java MapReduce Job

1. Compile Java file:

```
hadoop/bin/hadoop com.sun.tools.javac.Main WordCount.java
```

2. Create Jar File:

```
jar cf WordCount.jar WordCount*.class
```

3. Execute MapRedcue Job:

```
hadoop jar WordCount.jar WordCount /user/hadoop/Faust_1.txt
/user/hadoop/wordcount_output
```


Run Java MapReduce Job

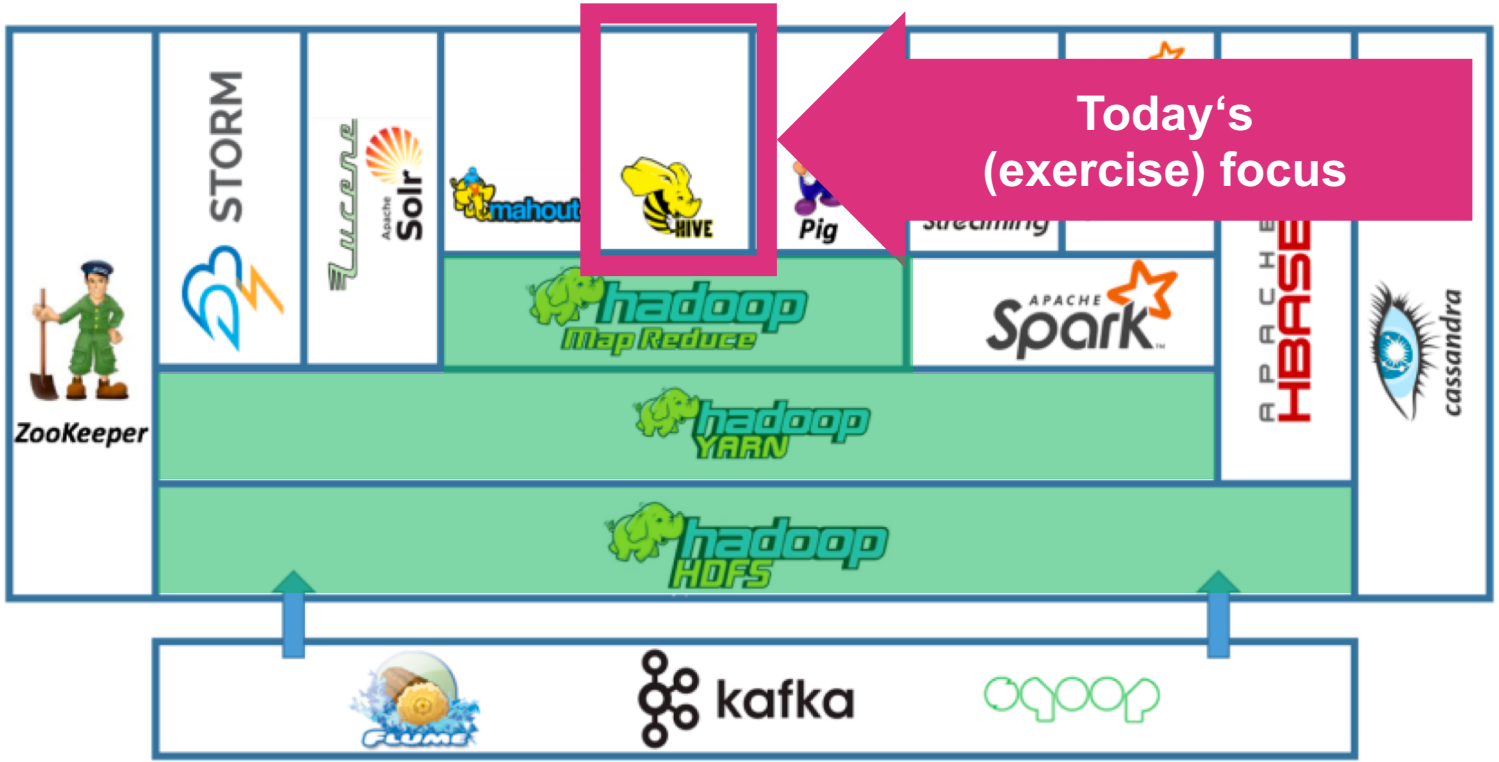
4. Take a look at the results:

```
hadoop fs -cat /user/hadoop/wordcount_output/part-r-00000 | head -10
"Allein,  1
"Alles    1
"Als      1
"Der      1
"Die      2
"Er       2
"Ich      4
"Im       1
"Mein     1
"Nur      1
[…]
```


HandsOn – **Hive and HiveQL**

Quick Introduction To Hive and HiveQL

The Hadoop Ecosystem



Apache Hive

Apache Hive

- Initial Release in October 2010
- written in Java
- current Version: 3.1.0

http://hive.apache.org

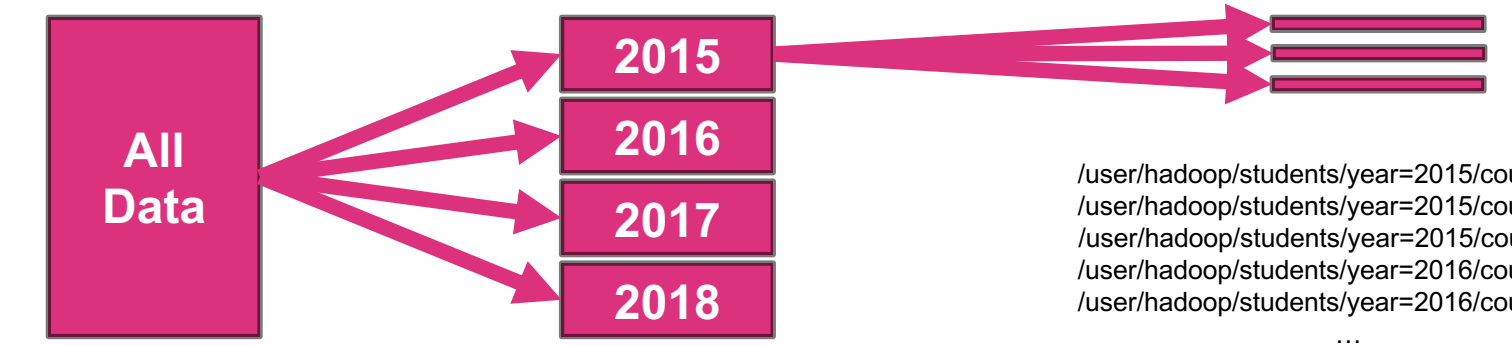
- **What is Hive:**
  - **easy to use** (HiveQL)
  - **based on Hadoop** (HDFS, YARN)
  - **good scale-out capabilities** (e.g. by partitioning and underlying HDFS and YARN)
  - **best used for:**
    - (BigData) **datawarehousing tasks**
    - a **DataLake**
    - **interface** for Analysts, DataScientists, Develo pers
    - **adhoc** (batch) **querying, aggregation** and **a nalysis** of large amounts of data (**PB!**) and hundreds of nodes
- **What Hive is NOT**:
  - **transactional database**
  - **highly responsive**

HiveQL

- SQL like **query language**
- Supported by: Hive CLI (*deprecated*), Beeline CLI and most JDBC clients
- Hive supported HDFS file formats:
  - **TextFile** (even compressed by gzip or bzip2)
  - **SequenceFile**
  - **RCFile**
  - **ORC**
  - **Parquet**
  - **Avro**

HDFS/Hive - **Partitioning**

- Partitioning of data distributes load and speeds up data processing
- A table can have one or more partition columns, defined by the time of creating a table (`CREATE TABLE student(id Int, name STRING) PARTITIONED BY (year STRING)… STORED AS TEXTFILE LOCATION '/user/hadoop/students';`)
- partitioning can be done either **static** or **dynamic**
- each distinct value of a partition column is represented by a **HDFS directory**



HDFS/Hive - **Wordcount**

- Previous WordCount example achieved by using Hive and HiveQL:

```
CREATE TABLE faust (line STRING);

LOAD DATA INPATH '/user/hadoop/faust' OVERWRITE INTO TABLE faust;

CREATE TABLE word_counts AS
SELECT word, count(1) AS count FROM
(SELECT explode(split(line, '\\s')) AS word FROM faust) temp
 GROUP BY word ORDER BY word;
```

```
SELECT * from word_counts ORDER BY count DESC LIMIT 10;
```

word	count
und	509
die	463
der	440
ich	435
Und	400
nicht	346
zu	319
[…]	

Exercises Preparation A

Install and Setup Hive

Install and Setup Hive

1. Get Hive:

```
wget http://ftp.halifax.rwth-aachen.de/apache/hive/hive-3.1.0/apache-hive-3.1.0-bin.tar.gz
```

2. Uncompress Hive:

```
tar -xzf apache-hive-3.1.0-bin.tar.gz
```

3. Move Binaries:

```
mv apache-hive-3.1.0-bin /home/hadoop/hive
```


Install and Setup Hive

4. Setup **UNIX** Environment Parameters:

```
vi .bashrc
```

```
export HIVE_HOME=/home/hadoop/hive
export PATH=$PATH:/home/hadoop/hive/bin
```

```
source .bashrc
```

Install and Setup Hive

5. Verify **Hive** Installation:

```
hive --version
```

```
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hadoop/hive/lib/log4j-slf4j-impl-2.10.0
.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/common/lib/s
lf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation
.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive 3.1.0
Git git://vgargwork.local/Users/vgarg/repos/hive.apache.master -r bcc7df9582483
1a8d2f1524e4048dfc23ab98c19
Compiled by vgarg on Mon Jul 23 16:02:03 PDT 2018
From source with checksum 147d8070369f8c672407753089777fd
```

Install and Setup Hive

6. Create Hive directories within *HDFS*:

```
hadoop fs -mkdir -p /user/hive/warehouse
hadoop fs -chmod g+w /user/hive/warehouse
```

```
hadoop fs -mkdir -p /tmp
hadoop fs -chmod g+w /tmp
```

7. Create Hive Config SH:

```
cp hive/conf/hive-env.sh.template hive/conf/hive-env.sh
vi hive/conf/hive-env.sh
```

```
export HADOOP_HOME=/home/hadoop/hadoop
export HIVE_CONF_DIR=/home/hadoop/hive/conf
```

Install and Setup Hive

8. Create Hive Config (*hive-site.xml*):

```
vi hive/conf/hive-site.xml
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?><!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<configuration>
                <property>
                                <name>javax.jdo.option.ConnectionURL</name>
                                <value>jdbc:derby:;databaseName=/home/hadoop/hive/metastore_db;create=true</value>
                                <description>
                                JDBC connect string for a JDBC metastore. To use SSL to encrypt/authenticate the connectio
                                n, provide database-specific SSL flag in the connection URL.
                                For example, jdbc:postgresql://myhost/db?ssl=true for postgres database.
                                </description>
                                </property>
```

Install and Setup Hive

```
        <property>
                <name>hive.metastore.warehouse.dir</name>
                <value>/user/hive/warehouse</value>
                <description>location of default database for the warehouse
                </description>
        </property>
        <property>
                <name>hive.metastore.uris</name>
                <value/>
                <description>Thrift URI for the remote metastore. Used by metastore cli
                ent to connect to remote metastore.
                </description>
        </property>
        <property>
                <name>javax.jdo.option.ConnectionDriverName</name>
                <value>org.apache.derby.jdbc.EmbeddedDriver</value>
                <description>Driver class name for a JDBC metastore</description>
        </property>
        <property>
                <name>javax.jdo.PersistenceManagerFactoryClass</name>
                <value>org.datanucleus.api.jdo.JDOPersistenceManagerFactory</value>
                <description>class implementing the jdo persistence</description>
        </property>
</configuration>
```


Install and Setup Hive

9. Edit YARN MapRed Config (*mapred-site.xml*) for higher memory usage:

```
vi hadoop/etc/hadoop/mapred-site.xml
```

```
<property>
        <name>mapreduce.map.memory.mb</name>
        <value>3072</value>
    </property>
    <property>
        <name>mapreduce.reduce.memory.mb</name>
        <value>3072</value>
    </property>
```

10. Restart DFS and YARN:

```
stop-all.sh
start-dfs.sh
Start-yarn.sh
```


Install and Setup Hive

11. By default Hive makes use of *Derby* as a metastore database, so we need to initialize *Derby*:

```
hive/bin/schematool -initSchema -dbType derby
```

```
[…]
Starting metastore schema initialization to 3.1.0
Initialization script hive-schema-3.1.0.derby.sql
[…]
Initialization script completed
schemaTool completed
```

Install and Setup Hive

12. Test if Hive Installation and Configuration is successful. Start Hive:

```
hive
```

```
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hadoop/hive/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/
impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7
.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = c120d0b1-9025-43db-96e4-48ccfb875f1a

Logging initialized using configuration in jar:file:/home/hadoop/hive/lib/hive-common-3.1.0.jar
!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider us
ing a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Hive Session ID = fdd6f06a-d4e4-48e6-8971-997e8a0a8e2c
hive>
```


Install and Setup Hive

13. Execute First SQL Query:

```
hive> show databases;
OK
default
Time taken: 0.083 seconds, Fetched: 1 row(s)
hive>
```


Hive: Create and Work with External Tables

Using public dataset of IMDb.com

Get IMDb Data And Move It To HDFS

1. Get **IMDb Data** (*https://www.imdb.com/interfaces/*):

```
wget https://datasets.imdbws.com/title.basics.tsv.gz
wget https://datasets.imdbws.com/title.ratings.tsv.gz
```

2. Uncompress IMDb Data:

```
gunzip title.basics.tsv.gz
gunzip title.ratings.tsv.gz
```

3. Create HDFS Directories for IMDb Data:

```
hadoop fs -mkdir /user/hadoop/imdb
hadoop fs -mkdir /user/hadoop/imdb/name
hadoop fs -mkdir /user/hadoop/imdb/ratings
```

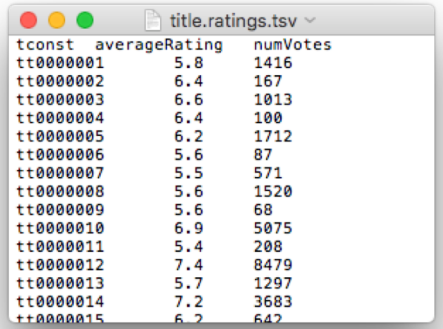

Create External Tables In Hive

4. Transfer IMDb data files to HDFS:

```
hadoop fs -put title.ratings.tsv /user/hadoop/imdb/ratings/ratings.tsv
hadoop fs -put  name.basics.tsv /user/hadoop/imdb/name/title.tsv
```

5. Create External Table **`imdb_ratings`** (file *title.ratings.tsv*) in Hive:

```
hive > CREATE EXTERNAL TABLE IF NOT EXISTS imdb_ratings(
          tconst STRING,
          average_rating DECIMAL(2,1),
          num_votes BIGINT
) COMMENT ‚IMDb Ratings‘
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' STORED AS
TEXTFILE LOCATION '/user/hadoop/imdb/ratings';
```



Create External Tables In Hive

6. Create External Table `imdb_movies` for file *title.basics.tsv* in Hive:

title.basics.tsv

```
tconst  titleType       primaryTitle    originalTitle   isAdult startYear       endYear runtimeMinutes  genres
tt0000001       short   Carmencita      Carmencita      0       1894    \N      1       Documentary,Short
tt0000002       short   Le clown et ses chiens  Le clown et ses chiens  0       1892    \N      5       Animation,Short
tt0000003       short   Pauvre Pierrot  Pauvre Pierrot  0       1892    \N      4       Animation,Comedy,Romance
tt0000004       short   Un bon bock     Un bon bock     0       1892    \N      \N      Animation,Short
tt0000005       short   Blacksmith Scene        Blacksmith Scene        0       1893    \N      1       Comedy,Short
tt0000006       short   Chinese Opium Den       Chinese Opium Den       0       1894    \N      1       Short
tt0000007       short   Corbett and Courtney Before the Kinetograph     Corbett and Courtney Before the Kinetograph     0       1894    \N      1       Short,Sport
tt0000008       short   Edison Kinetoscopic Record of a Sneeze  Edison Kinetoscopic Record of a Sneeze  0       1894    \N      1       Documentary,Short
tt0000009       movie   Miss Jerry      Miss Jerry      0       1894    \N      45      Romance
tt0000010       short   Employees Leaving the Lumière Factory   La sortie de l'usine Lumière à Lyon      0       1895    \N      1       Documentary,Short
tt0000011       short   Akrobatisches Potpourri Akrobatisches Potpourri 0       1895    \N      1       Documentary,Short
tt0000012       short   The Arrival of a Train  L'arrivée d'un train à La Ciotat        0       1896    \N      1       Documentary,Short
tt0000013       short   The Photographical Congress Arrives in Lyon     Neuville-sur-Saône: Débarquement du congrès des photographes à Lyon     0       1895    \N      1       Documentary,Short
tt0000014       short   Tables Turned on the Gardener   L'arroseur arrosé       0       1895    \N      1       Comedy,Short
tt0000015       short   Autour d'une cabine     Autour d'une cabine     0       1894    \N      2       Animation,Short
tt0000016       short   Barque sortant du port  Barque sortant du port  0       1895    \N      1       Documentary,Short
tt0000017       short   Italienischer Bauerntanz        Italienischer Bauerntanz        0       1895    \N      1       Documentary,Short
tt0000018       short   Das boxende Känguruh    Das boxende Känguruh    0       1895    \N      1       Short
tt0000019       short   The Clown Barber        The Clown Barber        0       1898    \N      \N      Comedy,Short
tt0000020       short   The Derby 1895  The Derby 1895  0       1895    \N      1       Documentary,Short,Sport
tt0000022       short   Blacksmith Scene        Les forgerons   0       1895    \N      1       Documentary,Short
tt0000023       short   The Sea Baignade en mer 0       1895    \N      1       Documentary,Short
tt0000024       short   Opening of the Kiel Canal       Opening of the Kiel Canal       0       1895    \N      \N      News,Short
tt0000025       short   The Oxford and Cambridge University Boat Race   The Oxford and Cambridge University Boat Race   0       1895    \N      \N      News,Short,Sport
tt0000026       short   The Messers. Lumière at Cards   Partie d'écarté 0       1896    \N      1       Documentary,Short
tt0000027       short   Cordeliers' Square in Lyon      Place des Cordeliers à Lyon     0       1895    \N      1       Documentary,Short
tt0000028       short   Fishing for Goldfish    La pêche aux poissons rouges    0       1895    \N      1       Documentary,Short
tt0000029       short   Baby's Dinner   Repas de bébé   0       1895    \N      1       Documentary,Short
tt0000030       short   Rough Sea at Dover      Rough Sea at Dover      0       1895    \N      1       Documentary,Short
```

Create External Tables In Hive

6. Create External Table **`imdb_movies`** for file *title.basics.tsv* in Hive:

```
hive > CREATE EXTERNAL TABLE IF NOT EXISTS imdb_movies(
          tconst STRING,
          title_type STRING,
          primary_title STRING,
          original_title STRING,
          is_adult DECIMAL(1,0),
          start_year DECIMAL(4,0),
          end_year STRING,
          runtime_minutes INT,
          genres STRING
) COMMENT 'IMDb Movies' ROW FORMAT DELIMITED FIELDS TERMINATED
BY '\t' STORED AS TEXTFILE LOCATION '/user/hadoop/imdb/name';
```


Exercises I

Hive: Create and Work with External Tables

HDFS and HiveQL Exercises - **IMDB**

1. Execute Tasks of previous HandsOn Slides
2. Download https://datasets.imdbws.com/name.basics.tsv.gz
3. Create HDFS Directory **/user/hadoop/imdb/actors/**names.tsv for file name.basics.tsv
4. Create External Hive Table `imdb_actors` for name.tsv
5. Use HiveQL to answer following questions:
   a) How many movies are within the IMDB dataset?
   b) Who is the oldest actor/writer/… within the dataset?
   c) Create a list (`m.tconst, m.original_title, m.start_year, r.average_rating, r.num_votes`) of movies which are:
   - equal or newer than year 2000
   - have an average rating better than 8
   - have been voted more than 100.000 times

   d) How many movies are in list of c)?

HDFS and HiveQL Exercises - **IMDB**

5. Use HiveQL to answer following questions:

e) We want to know which years have been great for cinema.
Create a list with one row per year and a related count of movies which:
- have an average rating better than 8
- have been voted more than 100.000 times

ordered descending by count of movies.