Proof Of Concept project
On
Hadoop and Big Data

Project Title

Wikipedia – Daily top 10 trending topics

Submitted by **Balasubramaniam Muthusamy** (balarm@yahoo.com) on 07-Aug-2014

This project was done as part of Edureka (www.edureka.in) Big Data and Hadoop training Batch 24
May - 22 June, 8:30pm to 11:30pm IST

Table of Contents

1. Project Objective

To demonstrate the ability and usability of the Hadoop framework for analyzing large volume of data (Big Data). This project was done as part of the "Haoop and Big Data training" curriculum from www.edureka.in , using publically available data sets.

2. Work Statement

Wikipedia may not need any introduction. This is very popular online encyclopedia covering several languages.

- **Wikipedia dumps page view counts data at http://dumps.wikimedia.org/other/pagecounts-raw/.**
- **The page counts file is published every hour for the entire Wikipedia DB.**
- **Each hourly file is about 340+ MB (uncompressed) contains 6.5-7 million records.**
- **This data file contains page view counts information for all language pages, project pages, image pages, etc.**
- **Need to extract only the topic pages of English Wikipedia pages.**
- **Sample page url will be like** http://en.wikipedia.org/wiki/Lionel_messi.
- **'Lionel_messi' is the topic/entity. The goal is to extract top 10 trending topics on daily basis from English Wikipedia pages.**

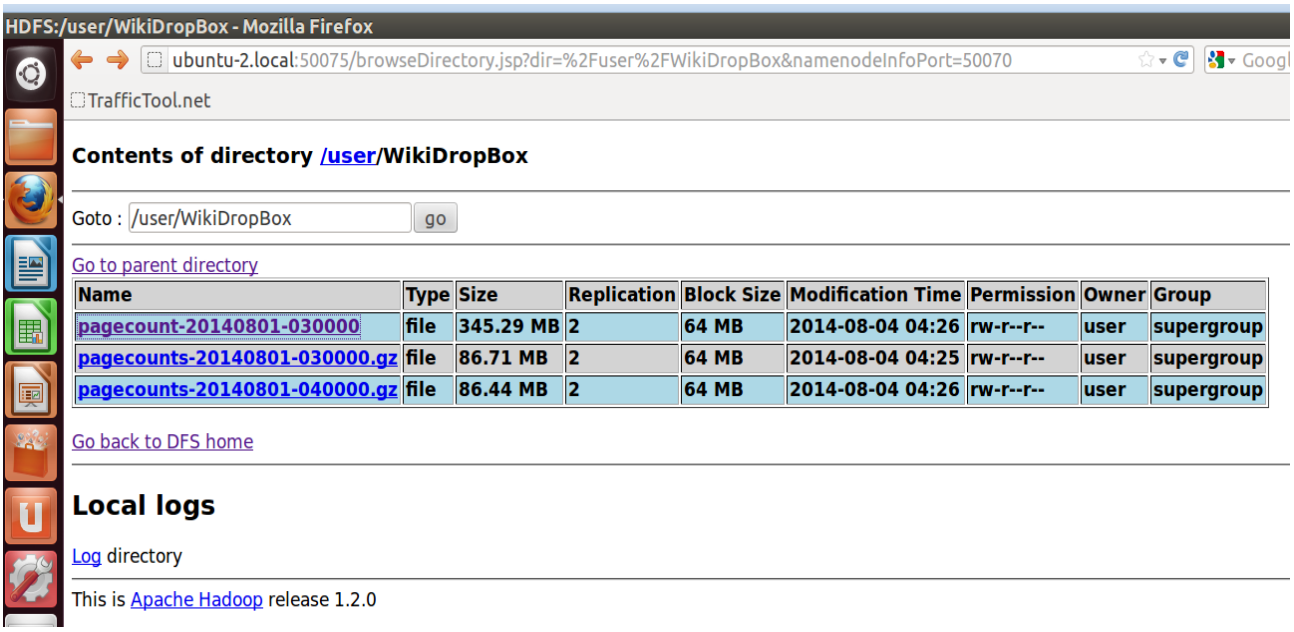
3. Infrastructure overview

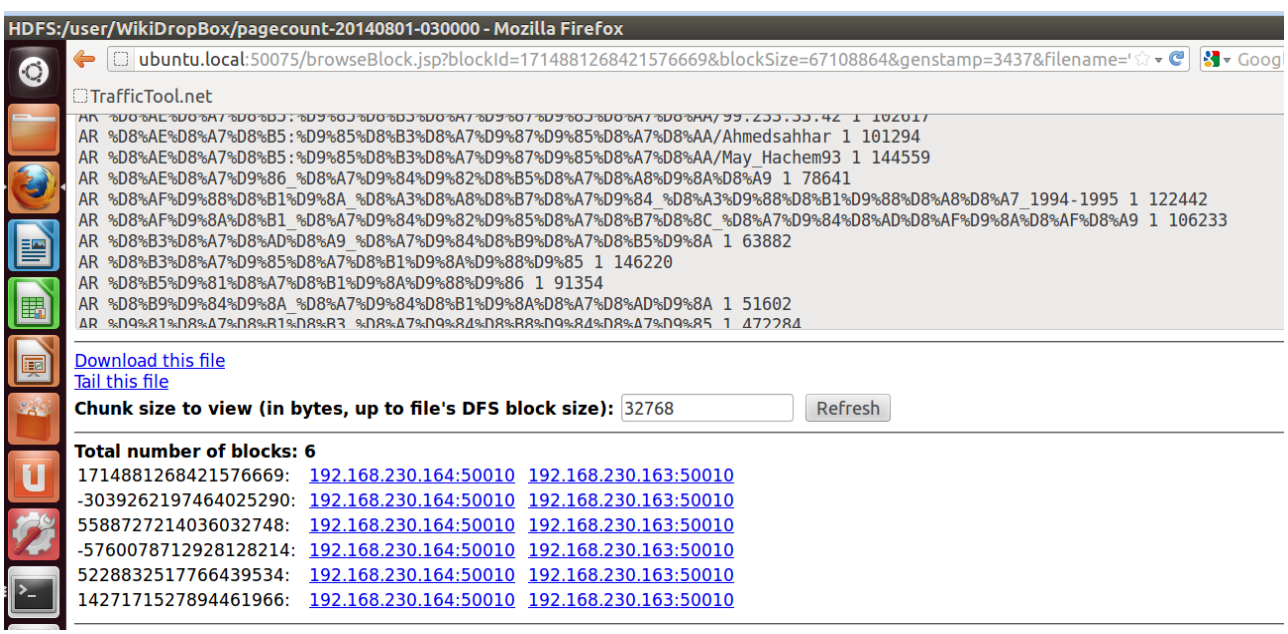
HW/SW components	Description
Multi node cluster	Ubuntu 12.04 LTS VM images, running on Windows 7, 64 bit Set up multi node cluster with 2 Ubuntu VM images as below. - 192.168.230.164 (Master) NameNode, DataNode - 192.168.230.164 (Slave) DataNode
RAM/Physical memory	2 GB RAM, 40 GB for each of Ubuntu images.
Java	1.6
IDE & other tools	Eclipse KEPLER , Putty, FileZilla
Hadoop	1.2.0
Apache Pig	Version 0.11.0 : for ETL
Apache Hive	Version 0.9.0 : for ETL. For this project, ETL can be accomplished using either Pig or Hive. Have demonstrated ETL using both Pig and Hive.

Apache Sqoop	Version 1.4.4: For data transfer from HDFS (Pig/Hive output files) to RDBMS (MySQL)
MySQL	Version 5.5.38 : for storing the daily top trending data.

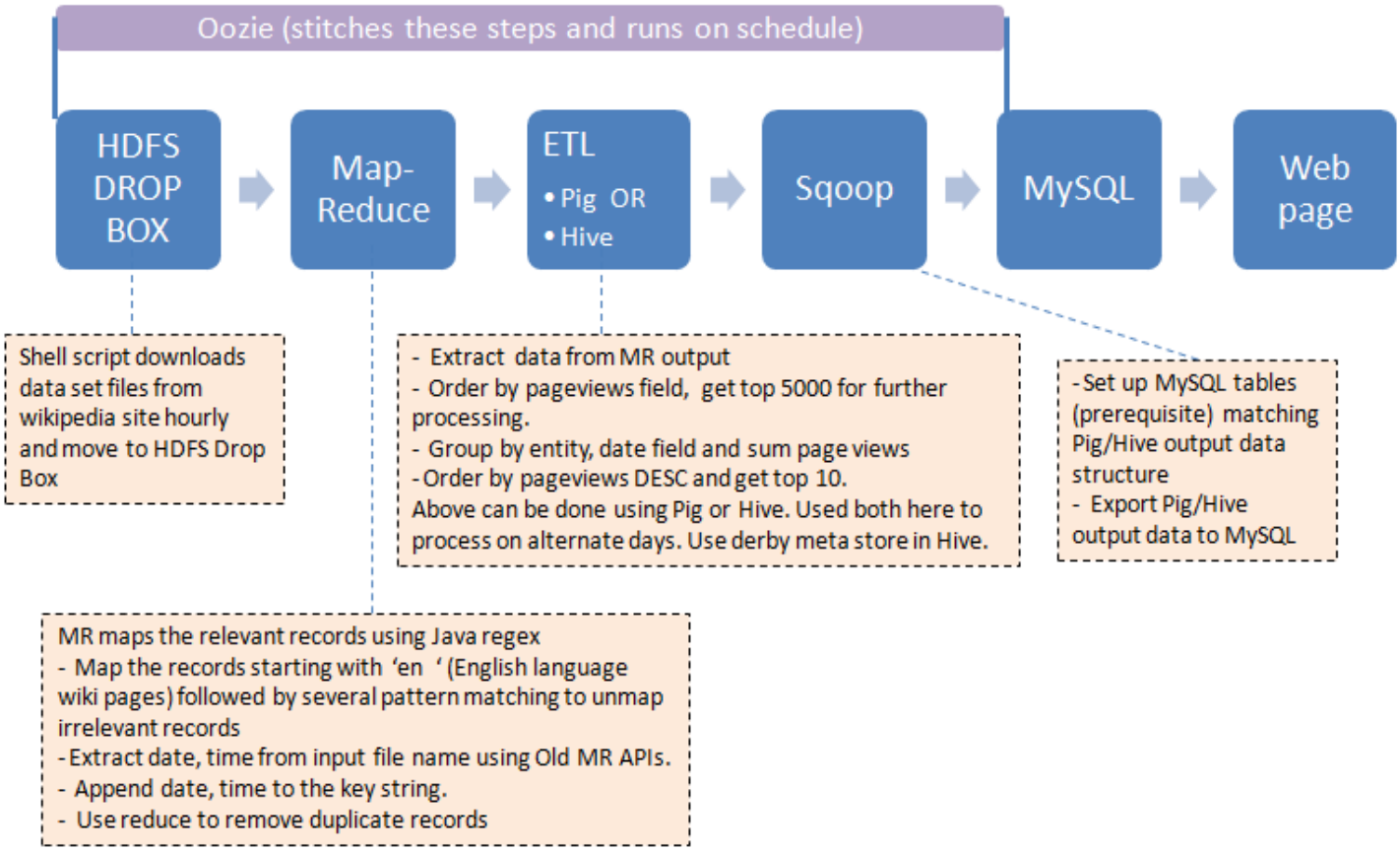
HDFS – Hadoop Distributed File System

- **Multi-node cluster was set up with two data nodes.**
- **Set the replication = 2. Retained default block size = 64 MB**
- **Each hourly data set of Wikipedia page counts file is about 340+ MB.**
- **Shown below hourly data sets stored in HDFS Drop Box with replication and file splits.**





4. Architecture/Solution overview



Oozie is not covered as part of this POC project. Made several attempts to install Oozie – which turned out to be complex, messy and limited resource documentation on the net.

5. Map Reduce

5.1. Objective and Data set structure

- **WikiDropBox : curl hourly data set from http://dumps.wikimedia.org/other/pagecounts-raw/ and stored it in hdfs '/user/WikiDropBox'. Currently this was done as manual step. It can be automated with shell script action in Oozie.**
- **Downloaded 3 hourly files per day for 2 days (Aug 1st and 2nd) for this POC project.**
- **All files under the folder : MapReduce program file input argument is pointed to the folder there by mapred picks up all the files under this folder (/user/WikiDropBox/).**
- **Downloaded files are compressed with .gz extension and these compressed files are handled by hadoop automatically.**

<u>MapReduce input data structure:</u>

- **Not all the records are delimited into four fields.**
- **Records needed for this analysis are the ones start with 'en '.**
- **On these records, series of pattern matching applied to map out irrelevant ones.**
- **Good records are delimited by space ' ' in the order as below: lang, entity, page view count, total number of bytes transfered**
- **Following are few sample records:**
    - fr.b Special:Recherche/Achille_Baraguey_d%5C%27Hilliers 1 624
    - en Main_Page 242332 4737756101
    - %22//commons.wikimedia.org/wiki/:A11v+1092338.ogg.en.srt 1 146
    - en Great_Goliath_Memorial_Battle_Royal 1 18626 **(good record)**
    - en Lionel_messi 5423 4737752 **(good record)**

- **Date and time fields are important data which is not part of the input file record. So, these need to be extracted from input file name during mapping process using old mapred APIs and append the key string.**

- **<u>MapReduce output data structure:</u>**
    - Lionel_messi 5423 4737752 20140801 020000 (starting 'en ' truncated and date, time data data appended)

5.2. Approach

Local development environment and proper unit testing are critical to make sure the code gets tested before promoting to production.

Local development environment can set up with hadoop 1.2.0 distribution and relevant jar files in Eclipse to perform local testing.

MRUnit is a framework to perform unit testing. This is not covered as part of this POC project.

5.3. MapReduce program code

WikiScrub.java

```java
package in.edureka;
import java.io.IOException;
import java.util.regex.Pattern;
import java.util.Iterator;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobClient;
import org.apache.hadoop.mapred.JobConf;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;
import org.apache.hadoop.mapred.TextInputFormat;
import org.apache.hadoop.mapred.TextOutputFormat;

public class WikiScrub {

    public static class Map extends MapReduceBase implements
    Mapper<LongWritable, Text, Text, IntWritable> {

        Text k2 = new Text();
        Pattern rx_lower_first_letter = Pattern.compile("([a-z])(.*)");
        Pattern rx_image = Pattern.compile("(.*).(jpg|gif|svg|jpeg|png|JPG|GIF|PNG|txt|ico|php|mp3).*");
        Pattern rx_namespace_titles = Pattern.compile("(.*)(Media|Special" +
                "|Talk|User|User_talk|User%20talk|Project|Project_talk|Project%20talk|File" +
                "|File_talk|File%20talk|MediaWiki|MediaWiki_talk|MediaWiki%20talk|Template" +
                "|Template_talk|Template%20talk|Template~|Help|Help_talk|Help%20talk|Help~|Category|Wikipedia~" +
                "|Category_talk|Category%20talk|Portal|Wikipedia|Wikipedia_talk|Wikipedia%20talk|Image~|Wiki//)(\\:|~)(.*)");

        String modLine = null;
        String fileName = new String();
        String date = new String();
        String time = new String();
        String[] dateTime;
          // fetch the input file name and split date , time
        public void configure(JobConf job1)
         {
               fileName = job1.get("map.input.file");
               dateTime = fileName.split("-");
                   date = dateTime[1];
                   time = dateTime[2].substring(0,6);
         }

@Override
public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output, Reporter reporter)
        throws IOException {
    String keyText = new String();
    String line = value.toString();
    if(line.startsWith("en "))
     { modLine = line.replaceFirst("en ", "");
        if(rx_lower_first_letter.matcher(modLine).matches() || rx_image.matcher(modLine).matches() || rx_namespace_titles.matcher(modLine).matches() ||
           modLine.startsWith("_") || modLine.contains("<br>") || modLine.startsWith("Main_Page") || modLine.startsWith("Undefined") || modLine.startsWith("Wiki"))
           {//these are unwanted records, hence don't map.
           }
        else {
        keyText = modLine+" "+date+" "+time;
        k2.set(keyText);
        output.collect(k2, new IntWritable(1));
              }
     }
}
} //end of mapper

        public static class Reduce extends MapReduceBase implements
                Reducer<Text, IntWritable, Text, Text> {
            @Override
            public void reduce(Text key, Iterator<IntWritable> values,
                    OutputCollector<Text, Text> output, Reporter reporter)
                    throws IOException {
                //Dedupe the records - get the key only which is already grouped, value is not required for this use case
                output.collect(key, new Text(" "));
               }
          } // end of reducer
```

```java
    // driver method
    public static void main(String[] args) throws Exception {
        //Creating a JobConf object and assigning a job name for identification purposes
        JobConf conf = new JobConf(WikiScrub.class);
        conf.setJobName("wikiscrub");
        //Setting configuration object with the Data Type of output Key and Value
        conf.setOutputKeyClass(Text.class);
        conf.setOutputValueClass(Text.class);
        //Setting configuration object with the Data Type of output Key and Value of mapper
        conf.setMapOutputKeyClass(Text.class);
        conf.setMapOutputValueClass(IntWritable.class);
        //Providing the mapper and reducer class names
        conf.setMapperClass(Map.class);
        conf.setReducerClass(Reduce.class);
        //Setting format of input and output
        conf.setInputFormat(TextInputFormat.class);
        conf.setOutputFormat(TextOutputFormat.class);
        //The hdfs input and output directory to be fetched from the command line
        FileInputFormat.setInputPaths(conf, new Path(args[0]));
        FileOutputFormat.setOutputPath(conf, new Path(args[1]));
        //Running the job
        JobClient.runJob(conf);
    }

}
```

5.4. Execution

```
user@ubuntu:~/codeMR$ hadoop jar wiki.jar in.edureka.WikiScrub /user/WikiDropBox/ /user/WikiMrOut
Warning: $HADOOP_HOME is deprecated.

14/08/04 07:54:51 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applic
e.
14/08/04 07:54:52 INFO util.NativeCodeLoader: Loaded the native-hadoop library
14/08/04 07:54:52 WARN snappy.LoadSnappy: Snappy native library not loaded
14/08/04 07:54:52 INFO mapred.FileInputFormat: Total input paths to process : 3
14/08/04 07:54:52 INFO net.NetworkTopology: Adding a new node: /default-rack/192.168.230.164:50010
14/08/04 07:54:52 INFO net.NetworkTopology: Adding a new node: /default-rack/192.168.230.163:50010
14/08/04 07:54:52 INFO mapred.JobClient: Running job: job_201408012222_0069
14/08/04 07:54:53 INFO mapred.JobClient:  map 0% reduce 0%
14/08/04 07:55:07 INFO mapred.JobClient:  map 8% reduce 0%
14/08/04 07:55:10 INFO mapred.JobClient:  map 10% reduce 0%
14/08/04 07:55:13 INFO mapred.JobClient:  map 11% reduce 0%
14/08/04 07:55:19 INFO mapred.JobClient:  map 12% reduce 0%
14/08/04 07:55:26 INFO mapred.JobClient:  map 13% reduce 0%
14/08/04 07:55:29 INFO mapred.JobClient:  map 14% reduce 0%
14/08/04 07:55:32 INFO mapred.JobClient:  map 15% reduce 0%
14/08/04 07:55:38 INFO mapred.JobClient:  map 16% reduce 0%
14/08/04 07:55:50 INFO mapred.JobClient:  map 17% reduce 0%
14/08/04 07:55:52 INFO mapred.JobClient:  map 25% reduce 0%
14/08/04 07:55:54 INFO mapred.JobClient:  map 29% reduce 0%
14/08/04 07:55:56 INFO mapred.JobClient:  map 29% reduce 4%
14/08/04 07:55:57 INFO mapred.JobClient:  map 30% reduce 4%
14/08/04 07:55:59 INFO mapred.JobClient:  map 30% reduce 8%
14/08/04 07:56:05 INFO mapred.JobClient:  map 39% reduce 8%
14/08/04 07:56:06 INFO mapred.JobClient:  map 40% reduce 8%
14/08/04 07:56:07 INFO mapred.JobClient:  map 43% reduce 8%
14/08/04 07:56:08 INFO mapred.JobClient:  map 43% reduce 12%
```

```
14/08/04 08:01:16 INFO mapred.JobClient:     Launched map tasks=8
14/08/04 08:01:16 INFO mapred.JobClient:     Data-local map tasks=6
14/08/04 08:01:16 INFO mapred.JobClient:     SLOTS_MILLIS_REDUCES=334906
14/08/04 08:01:16 INFO mapred.JobClient:   File Input Format Counters
14/08/04 08:01:16 INFO mapred.JobClient:     Bytes Read=543637610
14/08/04 08:01:16 INFO mapred.JobClient:   File Output Format Counters
14/08/04 08:01:16 INFO mapred.JobClient:     Bytes Written=179273719
14/08/04 08:01:16 INFO mapred.JobClient:   FileSystemCounters
14/08/04 08:01:16 INFO mapred.JobClient:     FILE_BYTES_READ=583971631
14/08/04 08:01:16 INFO mapred.JobClient:     HDFS_BYTES_READ=543638602
14/08/04 08:01:16 INFO mapred.JobClient:     FILE_BYTES_WRITTEN=876458462
14/08/04 08:01:16 INFO mapred.JobClient:     HDFS_BYTES_WRITTEN=179273719
14/08/04 08:01:16 INFO mapred.JobClient:   Map-Reduce Framework
14/08/04 08:01:16 INFO mapred.JobClient:     Map output materialized bytes=291995215
14/08/04 08:01:16 INFO mapred.JobClient:     Map input records=20709186
14/08/04 08:01:16 INFO mapred.JobClient:     Reduce shuffle bytes=291995215
14/08/04 08:01:16 INFO mapred.JobClient:     Spilled Records=17186357
14/08/04 08:01:16 INFO mapred.JobClient:     Map output bytes=280524484
14/08/04 08:01:16 INFO mapred.JobClient:     Total committed heap usage (bytes)=1320861696
14/08/04 08:01:16 INFO mapred.JobClient:     CPU time spent (ms)=289680
14/08/04 08:01:16 INFO mapred.JobClient:     Map input bytes=1084936119
14/08/04 08:01:16 INFO mapred.JobClient:     SPLIT_RAW_BYTES=992
14/08/04 08:01:16 INFO mapred.JobClient:     Combine input records=0
14/08/04 08:01:16 INFO mapred.JobClient:     Reduce input records=0
14/08/04 08:01:16 INFO mapred.JobClient:     Reduce input groups=3814459
14/08/04 08:01:16 INFO mapred.JobClient:     Combine output records=0
14/08/04 08:01:16 INFO mapred.JobClient:     Physical memory (bytes) snapshot=1549799424
14/08/04 08:01:16 INFO mapred.JobClient:     Reduce output records=3814459
14/08/04 08:01:16 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=3401936896
14/08/04 08:01:16 INFO mapred.JobClient:     Map output records=5728930
```

5.5. Output

6. Pig for ETL

6.1. Objective

Pig Latin scripting is used to perform ETL on the output data from Map Reduce. Transform the data from Map Reduce to arrive at top 10 records and store it in HDFS.

6.2. Dataset

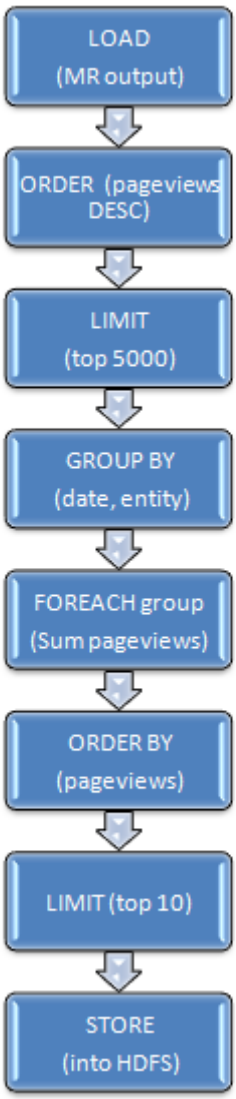
Pig input file = MR output file stored at '/user/WikiMrOut' (as in the above section).

Pig output data structure: entity date pageviews size

6.3. Approach

Pig is a data flow language, where each step of the data flow was executed separately in grunt shell and test the desired outcome on a sample data set using features like DESCRIBE, ILLUSTRATE. This helped develop a robust Pig script that can be executed reliably in larger data sets.

Data Flow diagram:



6.4. PIG program code

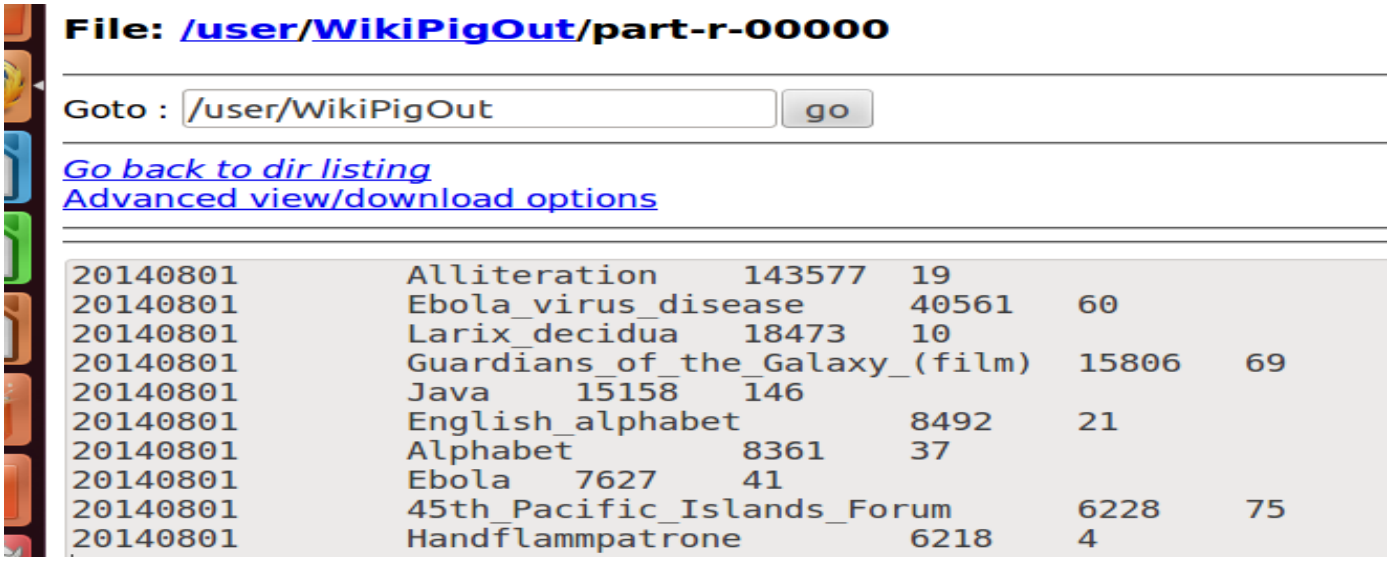
```
user@ubuntu: ~
user@ubuntu:~$ cat wiki.pig
hd = LOAD 'hdfs://192.168.230.164:8020/user/WikiMrOut/part-00000' using PigStorage(' ')
AS (entity:chararray,
    pageviews:long,
    totalsize:long,
    date:chararray,
    time:chararray);

hd_ord = ORDER hd BY pageviews DESC;
hd_ord_t5k = LIMIT hd_ord 5000;
hd_group = GROUP hd_ord_t5k by (entity,date);
dd = FOREACH hd_group
     GENERATE group.date, group.entity,SUM(hd_ord_t5k.pageviews),
     (SUM(hd_ord_t5k.totalsize)/SUM(hd_ord_t5k.pageviews))/1024;

dd_ord = ORDER dd BY $2 DESC;
dd_top10 = LIMIT dd_ord 10;

STORE dd_top10 INTO 'hdfs://192.168.230.164:8020/user/WikiPigOut';

user@ubuntu:~$
```

6.5. Execution

```
r@ubuntu: ~                                                    8:41 AM  user
user@ubuntu:~$ pig wiki.pig
Warning: $HADOOP_HOME is deprecated.

2014-08-04 08:23:20,208 [main] INFO  org.apache.pig.Main - Apache Pig version 0.11.0 (r1446324) compiled Feb 14 2013, 16:40:57
2014-08-04 08:23:20,213 [main] INFO  org.apache.pig.Main - Logging error messages to: /home/user/pig_1407155000201.log
2014-08-04 08:23:20,810 [main] INFO  org.apache.pig.impl.util.Utils - Default bootup file /home/user/.pigbootup not found
2014-08-04 08:23:21,088 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at:
hdfs://192.168.230.164:8020
2014-08-04 08:23:21,465 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to map-reduce job tracker
at: 192.168.230.164:8021
2014-08-04 08:23:22,771 [main] WARN  org.apache.pig.PigServer - Encountered Warning IMPLICIT_CAST_TO_LONG 1 time(s).
2014-08-04 08:23:22,858 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: GROUP_BY,ORDER_BY,LIMIT
```

```
@ubuntu: ~                                                     8:43 AM  user
2014-08-04 08:30:19,396 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 95% complete
2014-08-04 08:30:30,104 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 100% complete
2014-08-04 08:30:30,113 [main] INFO  org.apache.pig.tools.pigstats.SimplePigStats - Script Statistics:

HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
1.2.0   0.11.0  user    2014-08-04 08:23:24     2014-08-04 08:30:30     GROUP_BY,ORDER_BY,LIMIT

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTIme      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime   AvgReduceTime   M
ianReducetime   Alias   Feature Outputs
job_201408012222_0070   3       0       54      20      43      54      0       0       0       0       hd      MAP_ONLY
job_201408012222_0071   4       1       14      11      12      12      21      21      21      21      hd_ord  SAMPLER
job_201408012222_0072   4       1       37      15      28      31      32      32      32      32      hd_ord  ORDER_BY,COMBINER
job_201408012222_0073   1       1       4       4       4       4       11      11      11      11      hd_ord
job_201408012222_0074   1       1       4       4       4       4       11      11      11      11      dd,hd_group     GROUP_BY,COMBINER
job_201408012222_0075   1       1       4       4       4       4       11      11      11      11      dd_ord  SAMPLER
job_201408012222_0076   1       1       4       4       4       4       11      11      11      11      dd_ord  ORDER_BY,COMBINER
job_201408012222_0077   1       1       4       4       4       4       11      11      11      11      dd_ord          hdfs://192.168.2
164:8020/user/WikiPigOut,

Input(s):
Successfully read 3814459 records (179283027 bytes) from: "hdfs://192.168.230.164:8020/user/WikiMrOut/part-00000"

Output(s):
Successfully stored 10 records (335 bytes) in: "hdfs://192.168.230.164:8020/user/WikiPigOut"

Counters:
Total records written : 10
```

6.6. Output



7. Hive for ETL

7.1. Objective

HiveQL is used to perform ETL on the output data from Map Reduce. Transform the data from Map Reduce to arrive at top 10 records and store it in HDFS.

7.2. Dataset and tables

- **Hive input file = MR output file stored at '/user/WikiMrOut' (as in the above section).**
- **Create HIVE table and load the input data from HDFS for further processing (under 'user/hive/warehouse/…' folder.**

- **Pig output data structure: entity date pageviews.**

- **Insert the output data into another Hive table which stores the daily top 10 data.**

7.3. HiveQL code and execution

Create and load data in staging table 'wiki_daily_f'

```
user@ubuntu:~/codeHIVE$ cat createWikiHiveTables.sql
create database wiki_dbs;
use wiki_dbs;

create table wiki_daily_f
       (entity STRING,
        pageviews INT,
        totalsize INT,
        dt STRING,
        time STRING)
row format delimited
fields terminated by ' '
Stored as textfile;

load data inpath '/user/WikiMrOut/part-00000' into table wiki_daily_f;
```

```
user@ubuntu:~/codeHIVE$ hive -f createWikiHiveTables.sql
WARNING: org.apache.hadoop.metrics.jvm.EventCounter is deprecated. Please use org.apache.hadoop.log.metrics.EventCounter in all ti
.properties files.
Logging initialized using configuration in jar:file:/home/user/hive-0.9.0-bin/lib/hive-common-0.9.0.jar!/hive-log4j.properties
Hive history file=/tmp/user/hive_job_log_user_201408050151_586385259.txt
OK
Time taken: 5.423 seconds
OK
Time taken: 0.044 seconds
OK
Time taken: 1.105 seconds
Loading data to table wiki_dbs.wiki_daily_f
OK
Time taken: 0.869 seconds
```

Create output table (wiki_daily_t1), insert data from staging table, Transform and load in HIVE output table.

```
user@ubuntu:~/codeHIVE$ cat createAndLoadTop10Table.sql
use wiki_dbs;

create table wiki_daily_t1
       (entity STRING,
        dt STRING,
        pageviews INT
        )
row format delimited
fields terminated by ' '
Stored as textfile;

from wiki_daily_f wda
INSERT OVERWRITE TABLE wiki_daily_t1
select wda.entity, wda.dt, SUM(wda.pageviews) AS pv
where wda.entity != '!%'
group by wda.entity, wda.dt
order by pv DESC
limit 10;


user@ubuntu:~/codeHIVE$
```

```
user@ubuntu:~/codeHIVE$ hive -f createAndLoadTop10Table.sql
WARNING: org.apache.hadoop.metrics.jvm.EventCounter is deprecated. Please use org.apache.hadoop.log.metrics.EventCounter in al
.properties files.
Logging initialized using configuration in jar:file:/home/user/hive-0.9.0-bin/lib/hive-common-0.9.0.jar!/hive-log4j.properties
Hive history file=/tmp/user/hive_job_log_user_201408050155_1988724446.txt
OK
Time taken: 4.282 seconds
OK
Time taken: 1.242 seconds
Total MapReduce jobs = 3
Launching Job 1 out of 3
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapred.reduce.tasks=<number>
Starting Job = job_201408012222_0116, Tracking URL = http://ubuntu:50030/jobdetails.jsp?jobid=job_201408012222_0116
Kill Command = /home/user/hadoop-1.2.0/libexec/../bin/hadoop job  -Dmapred.job.tracker=192.168.230.164:8021 -kill job_201408012
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2014-08-05 01:55:21,738 Stage-1 map = 0%,  reduce = 0%
2014-08-05 01:55:40,062 Stage-1 map = 37%,  reduce = 0%
2014-08-05 01:55:49,208 Stage-1 map = 75%,  reduce = 0%, Cumulative CPU 16.61 sec
```

```
er@ubuntu: ~
2014-08-05 01:58:14,017 Stage-3 map = 100%,  reduce = 0%, Cumulative CPU 1.04 sec
2014-08-05 01:58:15,025 Stage-3 map = 100%,  reduce = 0%, Cumulative CPU 1.04 sec
2014-08-05 01:58:16,031 Stage-3 map = 100%,  reduce = 0%, Cumulative CPU 1.04 sec
2014-08-05 01:58:17,039 Stage-3 map = 100%,  reduce = 0%, Cumulative CPU 1.04 sec
2014-08-05 01:58:18,056 Stage-3 map = 100%,  reduce = 0%, Cumulative CPU 1.04 sec
2014-08-05 01:58:19,064 Stage-3 map = 100%,  reduce = 33%, Cumulative CPU 1.04 sec
2014-08-05 01:58:20,085 Stage-3 map = 100%,  reduce = 33%, Cumulative CPU 1.04 sec
2014-08-05 01:58:21,113 Stage-3 map = 100%,  reduce = 100%, Cumulative CPU 2.34 sec
2014-08-05 01:58:22,128 Stage-3 map = 100%,  reduce = 100%, Cumulative CPU 2.34 sec
2014-08-05 01:58:23,148 Stage-3 map = 100%,  reduce = 100%, Cumulative CPU 2.34 sec
2014-08-05 01:58:24,169 Stage-3 map = 100%,  reduce = 100%, Cumulative CPU 2.34 sec
MapReduce Total cumulative CPU time: 2 seconds 340 msec
Ended Job = job_201408012222_0118
Loading data to table wiki_dbs.wiki_daily_t1
Deleted hdfs://192.168.230.164:8020/user/hive/warehouse/wiki_dbs.db/wiki_daily_t1
Table wiki_dbs.wiki_daily_t1 stats: [num_partitions: 0, num_files: 1, num_rows: 0, total_size: 305, raw_data_size: 0]
10 Rows loaded to wiki_daily_t1
MapReduce Jobs Launched:
Job 0: Map: 1  Reduce: 1   Cumulative CPU: 66.7 sec   HDFS Read: 179282348 HDFS Write: 145128728 SUCCESS
Job 1: Map: 1  Reduce: 1   Cumulative CPU: 33.77 sec   HDFS Read: 145133007 HDFS Write: 546 SUCCESS
Job 2: Map: 1  Reduce: 1   Cumulative CPU: 2.34 sec   HDFS Read: 1005 HDFS Write: 305 SUCCESS
Total MapReduce CPU Time Spent: 1 minutes 42 seconds 810 msec
OK
Time taken: 190.57 seconds
```

7.4. Output

```
hive> use wiki_dbs;
OK
Time taken: 2.161 seconds
hive> describe wiki_daily_t1;
OK
entity  string
dt      string
pageviews       int
Time taken: 0.62 seconds
hive> select * from wiki_daily_t1;
OK
Alliteration    20140801        143577
Ebola_virus_disease     20140801        40561
Larix_decidua   20140801        18473
Guardians_of_the_Galaxy_(film)  20140801        15806
Java    20140801        15158
English_alphabet        20140801        8492
Alphabet        20140801        8361
Ebola   20140801        7627
45th_Pacific_Islands_Forum      20140801        6228
Handflammpatrone        20140801        6218
Time taken: 0.379 seconds
hive>
```

8. Sqoop

8.1. Objective

Use Sqoop export to transfer the HDFS data (Hive or Pig output) to RDBMS (MySQL).

8.2. MySQL set up

Set up MySQL and create table to receive the data via Sqoop from HDFS.

```
mysql> describe wiki_dd;
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| entity    | varchar(100) | NO   |     | NULL    |       |
| dt        | varchar(10)  | NO   |     | NULL    |       |
| pageviews | bigint(20)   | NO   |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> select * from wiki_dd order by pageviews desc;
+-------------------------------+----------+-----------+
| entity                        | dt       | pageviews |
+-------------------------------+----------+-----------+
| Alliteration                  | 20140801 |    143577 |
| Ebola_virus_disease           | 20140801 |     40561 |
| Larix_decidua                 | 20140801 |     18473 |
| Guardians_of_the_Galaxy_(film) | 20140801 |     15806 |
| Java                          | 20140801 |     15158 |
| English_alphabet              | 20140801 |      8492 |
| Alphabet                      | 20140801 |      8361 |
| Ebola                         | 20140801 |      7627 |
| 45th_Pacific_Islands_Forum    | 20140801 |      6228 |
| Handflammpatrone              | 20140801 |      6218 |
+-------------------------------+----------+-----------+
10 rows in set (0.00 sec)

mysql>
```

final output of top10 trending topic in wikipedia , with Aug 1st dataset.

8.3. Dataset

Hive output data in table 'wiki_daily_t1' as given below.

```
hive> use wiki_dbs;
OK
Time taken: 2.161 seconds
hive> describe wiki_daily_t1;
OK
entity  string
dt      string
pageviews       int
Time taken: 0.62 seconds
hive> select * from wiki_daily_t1;
OK
Alliteration    20140801        143577
Ebola_virus_disease     20140801        40561
Larix_decidua   20140801        18473
Guardians_of_the_Galaxy_(film)  20140801        15806
Java    20140801        15158
English_alphabet        20140801        8492
Alphabet        20140801        8361
Ebola   20140801        7627
45th_Pacific_Islands_Forum      20140801        6228
Handflammpatrone        20140801        6218
Time taken: 0.379 seconds
hive>
```

8.4. SQOOP execution

Sqoop export command:

$ sqoop export --connect jdbc:mysql://localhost/wiki_db --table wiki_dd --username root --password root --export-dir /user/hive/warehouse/wiki_dbs.db/wiki_daily_t1/000000_0 --input-fields-terminated-by ' '

```
r@ubuntu: ~                                                                    1:48 AM  user
user@ubuntu:~$ sqoop export --connect jdbc:mysql://localhost/wiki_db --table wiki_dd --username root --password root --export-dir /user/h
ive/warehouse/wiki_dbs.db/wiki_daily_t1/000000_0  --input-fields-terminated-by ' '
Warning: /usr/lib/hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /usr/lib/hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: $HADOOP_HOME is deprecated.

14/08/05 07:03:20 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
14/08/05 07:03:20 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
14/08/05 07:03:20 INFO tool.CodeGenTool: Beginning code generation
14/08/05 07:03:20 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `wiki_dd` AS t LIMIT 1
14/08/05 07:03:20 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `wiki_dd` AS t LIMIT 1
14/08/05 07:03:20 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /home/user/hadoop-1.2.0
Note: /tmp/sqoop-user/compile/a77c1c81fdcf266c40900fd32a66b5fa/wiki_dd.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
14/08/05 07:03:21 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-user/compile/a77c1c81fdcf266c40900fd32a66b5fa/wiki_dd.jar
14/08/05 07:03:21 INFO mapreduce.ExportJobBase: Beginning export of wiki_dd
14/08/05 07:03:24 INFO input.FileInputFormat: Total input paths to process : 1
14/08/05 07:03:24 INFO input.FileInputFormat: Total input paths to process : 1
14/08/05 07:03:24 INFO util.NativeCodeLoader: Loaded the native-hadoop library
14/08/05 07:03:24 WARN snappy.LoadSnappy: Snappy native library not loaded
14/08/05 07:03:24 INFO mapred.JobClient: Running job: job_201408012222_0121
14/08/05 07:03:25 INFO mapred.JobClient:  map 0% reduce 0%
14/08/05 07:03:33 INFO mapred.JobClient:  map 50% reduce 0%
```

```
user@ubuntu: ~
14/08/05 07:03:37 INFO mapred.JobClient:     SLOTS_MILLIS_MAPS=15664
14/08/05 07:03:37 INFO mapred.JobClient:     Total time spent by all reduces waiting after reserving slots (ms)=0
14/08/05 07:03:37 INFO mapred.JobClient:     Total time spent by all maps waiting after reserving slots (ms)=0
14/08/05 07:03:37 INFO mapred.JobClient:     Launched map tasks=4
14/08/05 07:03:37 INFO mapred.JobClient:     Data-local map tasks=4
14/08/05 07:03:37 INFO mapred.JobClient:     SLOTS_MILLIS_REDUCES=0
14/08/05 07:03:37 INFO mapred.JobClient:   File Output Format Counters
14/08/05 07:03:37 INFO mapred.JobClient:     Bytes Written=0
14/08/05 07:03:37 INFO mapred.JobClient:   FileSystemCounters
14/08/05 07:03:37 INFO mapred.JobClient:     HDFS_BYTES_READ=1569
14/08/05 07:03:37 INFO mapred.JobClient:     FILE_BYTES_WRITTEN=255856
14/08/05 07:03:37 INFO mapred.JobClient:   File Input Format Counters
14/08/05 07:03:37 INFO mapred.JobClient:     Bytes Read=0
14/08/05 07:03:37 INFO mapred.JobClient:   Map-Reduce Framework
14/08/05 07:03:37 INFO mapred.JobClient:     Map input records=10
14/08/05 07:03:37 INFO mapred.JobClient:     Physical memory (bytes) snapshot=160735232
14/08/05 07:03:37 INFO mapred.JobClient:     Spilled Records=0
14/08/05 07:03:37 INFO mapred.JobClient:     CPU time spent (ms)=1280
14/08/05 07:03:37 INFO mapred.JobClient:     Total committed heap usage (bytes)=65011712
14/08/05 07:03:37 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=1514405888
14/08/05 07:03:37 INFO mapred.JobClient:     Map output records=10
14/08/05 07:03:37 INFO mapred.JobClient:     SPLIT_RAW_BYTES=751
14/08/05 07:03:37 INFO mapreduce.ExportJobBase: Transferred 1.5322 KB in 14.415 seconds (108.8447 bytes/sec)
14/08/05 07:03:37 INFO mapreduce.ExportJobBase: Exported 10 records.
user@ubuntu:~$
```

8.5. Output

```
mysql> describe wiki_dd;
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| entity    | varchar(100) | NO   |     | NULL    |       |
| dt        | varchar(10)  | NO   |     | NULL    |       |
| pageviews | bigint(20)   | NO   |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> select * from wiki_dd order by pageviews desc;
+--------------------------------+----------+-----------+
| entity                         | dt       | pageviews |
+--------------------------------+----------+-----------+
| Alliteration                   | 20140801 |    143577 |
| Ebola_virus_disease            | 20140801 |     40561 |
| Larix_decidua                  | 20140801 |     18473 |
| Guardians_of_the_Galaxy_(film) | 20140801 |     15806 |
| Java                           | 20140801 |     15158 |
| English_alphabet               | 20140801 |      8492 |
| Alphabet                       | 20140801 |      8361 |
| Ebola                          | 20140801 |      7627 |
| 45th_Pacific_Islands_Forum     | 20140801 |      6228 |
| Handflammpatrone               | 20140801 |      6218 |
+--------------------------------+----------+-----------+
10 rows in set (0.00 sec)

mysql>
```

final output of top10 trending topic in wikipedia , with Aug 1st dataset.

9. Conclusion

The power and usability of Hadoop framework and the solutions in Hadoop eco-system are demonstrated successfully. These codes can scale to handle real time scenarios.

All the steps were done manually for this POC project, as the focus was more on demonstrating the capability of each of the solutions. Implementing Oozie would help stitch these steps and make the end to end solution seamless.