Using Apache Spark in the Analysis of Wikipedia Page View Data in Correlation to the Real World Events

Sebastian Cousins, Debzani Deb
Department of Computer Science
Winston-Salem State University

Purpose

- Apache Spark [1] has emerged as the next generation big data processing engine, and businesses are adopting it at lightning speed.
- However, as a new framework, there are not enough resources available that explicitly illustrates the process of analyzing and querying a NOSQL (schema-less) big data store using Spark and visualizing the results in an efficient way.
- The objective of this study is to gain knowledge in the use and application of Spark and other newer big data tools and to identify a process of analyzing, querying and visualizing big data.
- More specifically, we utilized tools such as Apache Spark, Spark SQL, Scala Programming Framework [2], and Apache Zeppelin [3] to gain insight on how strongly Wikipedia page view data [4] correlates with real-world trending events.

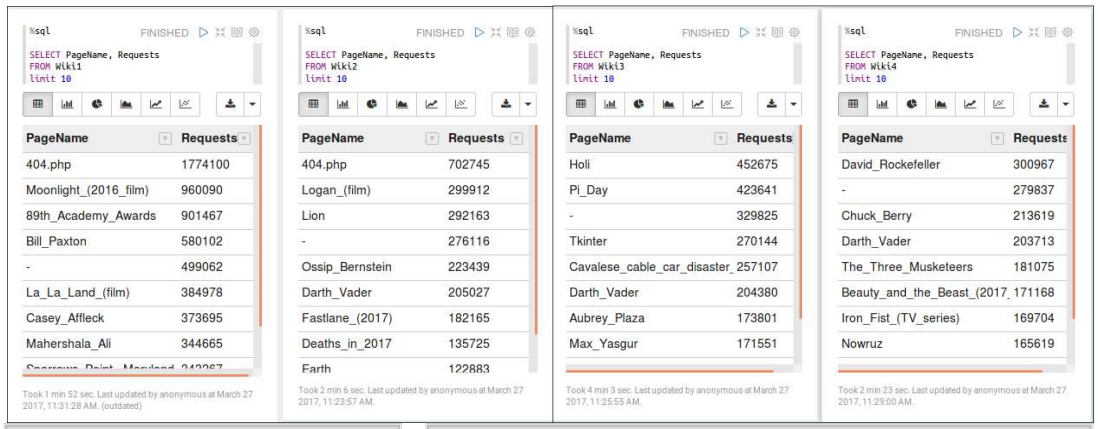
Methodology

- Wikipedia page view data (space delimited text files) were preprocessed and aggregated to analyze recent four weeks' worth of page view data (40GB).
- We utilized Spark-supported Scala programming framework to write standalone program where Spark SQL queries can be embedded.
- We then linked our queries to Apache Zeppelin framework that allows us to instantly visualize the query results.

Feedback @ http://compsci.wssu.edu/iBigCloud

Scala Standalone Program where SQL Queries are Embedded

```
val rawWiki = sc.textFile("/home/ra1/Desktop/Spark_Temp/MT1") // Loads wiki pageview data and creates RDD
val enPages = rawWiki.filter(_.split(" ")(0) == "en") // create new RDD after pulling only english "en" wiki pages
//Perform series of transformations to create RDD with key(PageName)Value(Requests) pair
val enTuples = enPages.map(line => line.split(" "))
val enKeyValuePairs = enTuples.map(line => (line(1), line(2).toInt))
//Reduce the rows to consolidate "Requests" of each "PageName"
val reducedWiki = enKeyValuePairs.reduceByKey(_+_, 2)
// Create DataFrame from the RDD and register it
val wikiDataframe = redcedWiki.toDF("PageName","Requests")
wikiDataframe.registerTempTable("WikiDataframe")
//wikiDataframe can now be used as a relational table and can be queried
//Remove wikipedia "Main_Page and other irrelevant pages
val Wiki1 = sqlContext.sql("SELECT * FROM WikiDataframe where PageName NOT LIKE " AND PageName NOT
LIKE '%en%' AND PageName NOT LIKE '%Main_Page%' AND ……..)
Wiki1.registerTempTable("Wiki1")
// Spark SQL query to find 10 most requested Wikipedia pages
val Trend10 = sqlContext.sql("SELECT PageName, Requests FROM Wiki group by PageName,     Requests order
by Requests desc limit 10")
Trend10.show() //Show the results
```



%sql FINISHED

```
SELECT PageName, Requests
FROM Wiki1
limit 10
```

PageName	Requests
404.php	1774100
Moonlight_(2016_film)	960090
89th_Academy_Awards	901467
Bill_Paxton	580102
-	499062
La_La_Land_(film)	384978
Casey_Affleck	373695
Mahershala_Ali	344665

Took 1 min 52 sec. Last updated by anonymous at March 27 2017, 11:31:28 AM. (outdated)

%sql FINISHED

```
SELECT PageName, Requests
FROM Wiki2
limit 10
```

PageName	Requests
404.php	702745
Logan_(film)	299912
Lion	292163
-	276116
Ossip_Bernstein	223439
Darth_Vader	205027
Fastlane_(2017)	182165
Deaths_in_2017	135725
Earth	122883

Took 2 min 6 sec. Last updated by anonymous at March 27 2017, 11:23:57 AM.

%sql FINISHED

```
SELECT PageName, Requests
FROM Wiki3
limit 10
```

PageName	Requests
Holi	452675
Pi_Day	423641
-	329825
Tkinter	270144
Cavalese_cable_car_disaster	257107
Darth_Vader	204380
Aubrey_Plaza	173801
Max_Yasgur	171551

Took 4 min 3 sec. Last updated by anonymous at March 27 2017, 11:25:55 AM.

%sql FINISHED

```
SELECT PageName, Requests
FROM Wiki4
limit 10
```

PageName	Requests
David_Rockefeller	300967
-	279837
Chuck_Berry	213619
Darth_Vader	203713
The_Three_Musketeers	181075
Beauty_and_the_Beast_(2017_	171168
Iron_Fist_(TV_series)	169704
Nowruz	165619

Took 2 min 23 sec. Last updated by anonymous at March 27 2017, 11:29:00 AM.

Acknowledgment

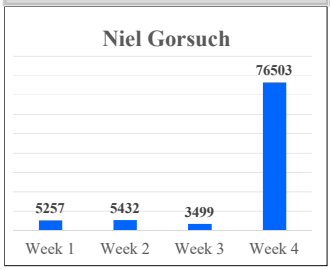
NSF RIA Award #1600864

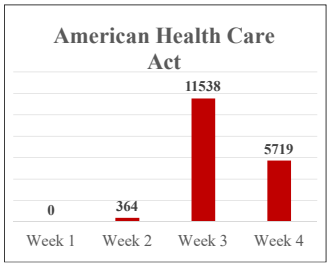
WSSU RIP Award

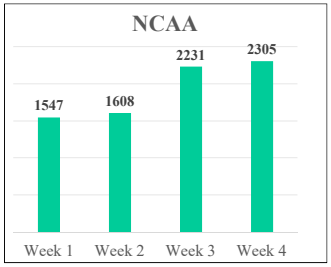
Conclusions

- The resultant visualizations show strong correlation between Wikipedia page search and the real-world trending topics.
- We also found Spark to be dramatically faster than Apache Hadoop and Hive [5] when it comes to big data analytics.

Weekly Topic Trends







References

1. Apache Spark, http://spark.apache.org/.
2. Scala, https://www.scala-lang.org/
3. Apache Zeppelin, https://zeppelin.apache.org/
4. Page view statistics for Wikimedia projects, https://dumps.wikimedia.org/other/pagecounts-raw/.
5. Apache Hive, https://hive.apache.org/.