SEIS 736 Final Project Report

Examining the Correlation between Wikipedia page views and News events   
Princewill Eneh

**Abstract**

*In this paper I explore the correlation between the number of page views on wikipedia pages to news events that happen on the specific day of the views. I intend to use knowledge from Mapreduce, partitions, counters, and other key concepts we learnt in the class.*

**Table of Content**

1. Introduction 3
2. Data source 3
3. Data description and schema 3
4. Data pre-processing required (parsing, filtering, etc.) 4
5. Bad data 5
6. Map Reduce algorithm 6
7. Output description 9
8. Analysis (Verifying output) 12
9. Performance/scale characteristics 14
10. Future Research 16
11. Conclusions 16

**Introduction**

Wikipedia is at the forefront of information lookup and most times when we search for information about an entity on search engines like Google and Bing, more often than not the first result that is returned is usually an article from Wikipedia about the searched term. Knowing this, We will try to see if the number of times an entity is looked up on Wikipedia correlates with the current event or news happening at the time of the lookup. We will capture the number of times a page is viewed as page counts. We will do this by using MapReduce to analyze and sort these pages. After which we will compare the results to trending topics and events in the news that happened on that particular day.

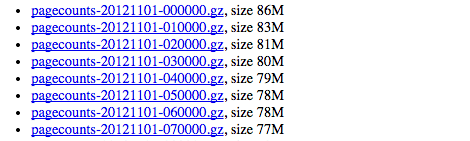
**Data source  
Wikipedia Traffic Statistics Dataset**

This dataset contains a 320 GB sample of the data used to power [trendingtopics.org](http://www.trendingtopics.org/). It includes 7 months of hourly page traffic statistics for over 2.5 Million wikipedia articles (~ 1 TB uncompressed) along with the associated wikipedia content, linkgraph, & metadata.[1]

**Data description and schema  
Wikidata/wikistats (260G)**

This contains hourly wikipedia article traffic statistics dataset covering 7 month period from October 01 2008 to Nov 2014, this data is regularly [logged from the wikipedia squid proxy](http://dammit.lt/2007/12/10/wikipedia-page-counters/) by [Domas Mituzas](http://dammit.lt/). This is a large data and will be problematic if all 260GB of data is used because of the scope of the hardware available to me. Therefore I evaluated the page views for just a single day ( **November 4th 2012** ). This was reduced the total size of the data to evaluate to 2.6GB.[2]

Each log file is named with the date and time of collection: pagecounts-20090430-230000.gz



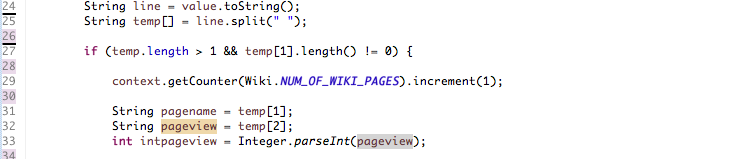
Each request of a page, whether for editing or reading, whether a "special page" such as a log of actions generated on the fly, or an article from Wikipedia or one of the other projects, reaches one of Wikipedia squid caching hosts and the request is sent via udp to a filter which tosses requests from the internal hosts, as well as requests for wikis that aren't among the general projects. This filter writes out the project name, the size of the page requested, and the title of the page requested.[3]

Here are a few sample lines from one file:   
Each line has 4 fields: **projectcode, pagename, pageviews, bytes**

**en Barack\_Obama 997 123091092  
 en Barack\_Obama%27s\_first\_100\_days 8 850127  
 en Barack\_Obama,\_Jr 1 144103  
 en Barack\_Obama,\_Sr. 37 938821  
 en Barack\_Obama\_%22HOPE%22\_poster 4 81005**

**Data pre-processing required (parsing, filtering, etc.)**

Like mentioned above, each line in the file has 4 fields **projectcode, pagename, pageviews, bytes**. We are only interested in the pagenames and the pageviews. Therefore we have to eliminate the other two . We do this by using the below code in our map function in the mapper class.



The code steps through each line of the file and assigns the items in column 2 and 3 to temp variables which are then passed to the **context.write(text, intWritable).**

**Bad data**

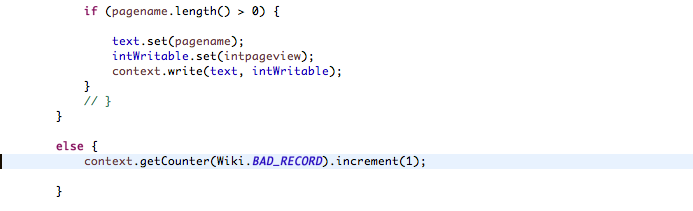
I anticipated the occurrence of bad data. For the mapper code to work the parsing should always pick the 2nd and 3rd values in the line. This is not always the case as some of the wikipedia pages had spaces in them and others had indecipherable letters. An example is shown below:

I%Q .(fx@'"K];=m{з[}U\_윾~[z[|KwGVoܟ~D/\_\_쯽!ۦx#B^x}\_y`E/nU :c/vOl~\_}ƇO$x~=2QNq`A~f-9h~ef<^8bs(f/%A("

To check for this I declared an ENUM in the mapper class.

Screen Shot 2014-12-01 at 10.46.21 AM.png

Every time this situation is encounter the BAD\_RECORD counter is incremented and the line is not considered in the final result.

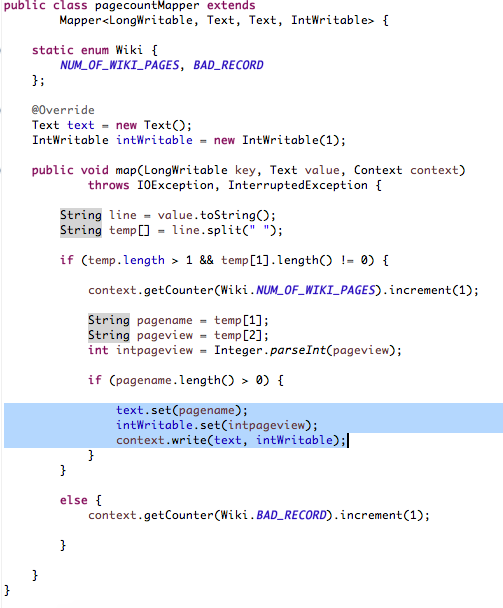


**Map Reduce Algorithm**

We want the final result to be the pages and the number of views with the result sorted in descending order. This cannot be achieved with one single MapReduce job. A second mapReduce (**MAPPER 2 , REDUCER 2**) is required to sort the initial results that we get from the first MapReduce.

**MAPREDUCE 1**

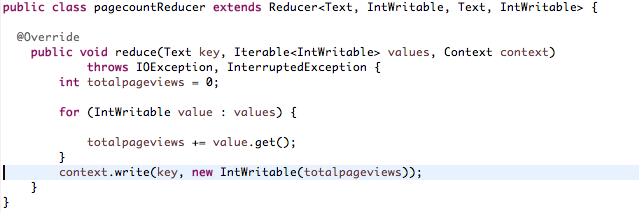
Below is a the map function for **MAPPER 1**:

*figure 6: MAPPER 1*

After our lines are parsed by eliminating by picking the 2nd and 3rd words in the line which represent the page (**pagename**) and the pageviews(**intpageviews**) for that hour of the day, we assign the values to the keys and values of the the mapper class. But first we have to convert the page views from strings to int.

We also include counters for bad records (*BAD\_RECORD)* and total number of articles (*NUM\_OF\_WIKI\_PAGES*) which are incremented for every bad record and line in the file respectively. This is important because it helps us check for bad records and also helps us evaluate the percentage of pageviews of the top pages that account for all the pages viewed on that day.

**REDUCER 1**



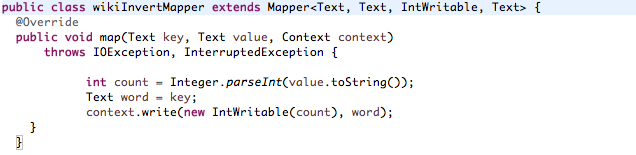
At this point we have the pagename and the page views for that hour. We wish to add them up for all the other hours that will be returned.  
The reduce method runs once for each key which is the pagename received from the shuffle and sort phase of the MapReduce framework. The method receives a key of type Text, a set of values (pageviews) of type **IntWritable**, and a **Context** object. For each value in the set of values passed to us by the mapper we add the value to the word count counter for this key. So in this case we add up all the views from all the hours. Then call the write method on the Context object to emit a key and a value from the reduce method.

**MAPREDUCE 2**

The second MapReduce job transposes key and value, so the count (as key now) is sorted. We now add the sort comparator class to sort the counts in descending, instead of the default ascending order.

**MAPPER 2**

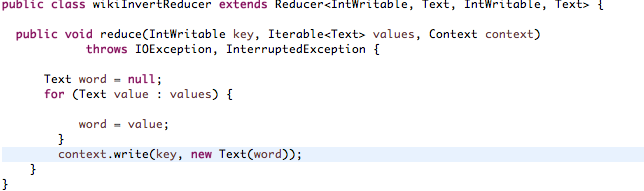
The results from the first MapReduce is a page and the page total number of views for that day. But we are interested in knowing the pages that have the highest number of views sorted in descending order.



Here we parse the result from **MapReduce 1** to **MAPPER 2**. This will be the pagename and the pageview. The page views are made the keys. This will ensure that our results are sorted.

**REDUCER 2**

Here the page names are made the values.



**Output description**

We will perform the output analysis by first showing the outputs from **MapReduce 1** and **MapReduce 2** as well as various new sources and archives of top stories on November 4th, 2012.

**Result of MapReduce 1**

As mentioned previously the output of the MapReduce 1 is a list of all the pagenames and the number of pageviews. These are however not sorted. The run created 48 output files because the number of tasks was not specified. There were **24 Launched Map tasks** (representing the 1 hour/file of a total of 24 hours in a day)and **48 Launched Reduce tasks** .

A small sample of the output is shown below:

|  |  |
| --- | --- |
| Rich\_Owens 3  Rich\_Site\_Summary 9  Rich\_Thompson\_(pitcher,\_born\_1984) 4  Rich\_Zvosec 5  Rich\_in\_London 1  Rich\_list 3  Rich\_user\_interaction 7  Richa\_Gangopadhyay%23Personal\_life 1  Richard%20Avedon 1  Richard%20Blackwell 1  Richard%20Cornu 1  Richard%20Haier 1  Richard%20Harris 3  Richard%20Hill%20(footballer%20born%201963) 2  Richard%20Howly 1  Richard%20Monckton%20Milnes,%201st%20Baron%20Houghton 1  Richard%20Norris%20(field%20hockey) 1  Richard%20Piaty 1  Richard%20Toepffer 1  Richard%20W.\_Pfaff 1  Richard's\_paradox 3  Richard+Diebenkorn 1  Richard,\_hertug\_av\_Gloucester 1  Richard\_1.\_L\xC3\xB8vehjerte 1 | Richard\_Alves 1  Richard\_Austen\_Ferries 2  Richard\_B.\_Bernstein 4  Richard\_B.\_Fisher\_Center\_for\_the\_Performing\_Arts 11  Richard\_B.\_Spencer 1  Richard\_Balam 2  Richard\_Bangs\_Adventures 2  Richard\_Barham\_(disambiguation) 1  Richard\_Barrett\_(Irish\_republican) 2  Richard\_Barrett\_(irish\_Republican) 1  Richard\_Barthram 2  Richard\_Basil\_Brandram\_Jones 3  Richard\_Baum 1  Richard\_Beckinsale 438  Richard\_Bell\_(Virginia\_politician) 1  Richard\_Bertram 2  Richard\_Birdsall 1  Richard\_Blackwell%23Ten\_Worst\_Dressed\_Women\_List 1  Richard\_Bohlman 1  Richard\_Boyle,\_1st\_Earl\_of\_Cork 8  Richard\_Boys 1  Richard\_Brandram 2  Richard\_Bright\_(actor) 62  Richard\_Broghamer 1 |

**Result of MapReduce 2**

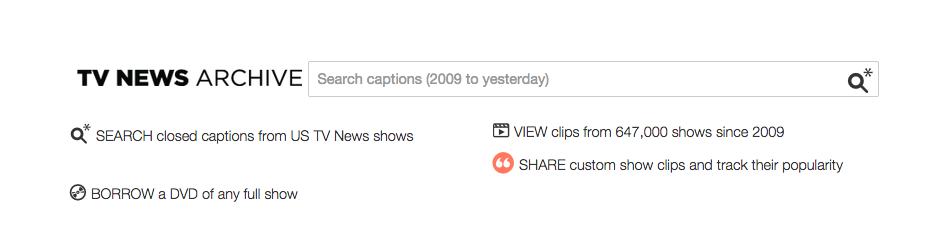
The output from **MapReduce 1** becomes the input for **MapReduce 2**. The sorting is performed and a single output is produced. There were **48 Launched Map tasks** (representing the 48 output files from **MapReduce 1** )and **1 Launched Reduce task**.

Output from **MapReduce 2:**

|  |  |
| --- | --- |
| 19054936 en  7526066 ja  3802643 Main\_Page  2765650 de  167365 Spezial:Search  159528 /Wikipedia:Your\_first\_article  156983 he  141458 index.html  **129038 Skyfall**  124840 Ottoman\_Empire  **117858 James\_Bond**  **92952 One\_Direction**  86532 Facebook  84683 YouTube  82870 Main\_Page/  81939 500.shtml  77248 Rio%2B20  75476 Hauptseite  **75117 Gangnam\_Style**  **72986 Daniel\_Craig**  71837 Category:Petroleum  69343 Portal:Huvudsida  65129 Halloween  **59753 Barack\_Obama**  **58838 Mitt\_Romney**  57872 Cat\_anatomy  53728 Jerzy\_Janowicz  53641 Leonard\_Matlovich  53041 File:Revival\_sivaji.png  51661 w/index.php  51282 Abu\_Dhabi  48281 Wiki  44987 Assassin%27s\_Creed\_III  44895 Ana\_Sayfa  44201 German\_in\_the\_United\_States  **43894 Daylight\_saving\_time**  43304 Google | 42582 United\_States\_presidential\_election,\_2012  41384 Windows\_8  40873 Fran\_pes  39600 encyclopedie/Accueil  39277 Hollow\_Body\_Guitar  **37996 Hurricane\_Sandy**  36409 Harry\_Potter  35960 404\_error  35615 Special:Watchlist  **35046 Cristiano\_Ronaldo**  **34707 Lionel\_Messi**  34571 Special:Recherche  33425 David\_Ferrer  **33332 Justin\_Bieber**  33274 Guy\_Fawkes  **32242 Nicki\_Minaj**  31460 bg  31178 How\_I\_Met\_Your\_Mother  31028 Brunost  30876 Abraham\_Lincoln  30434 India  30343 Halo\_4  29720 The\_Big\_Bang\_Theory  29259 Cline\_Dion  29156 Khas:Perubahan\_terkini  28584 Star\_Wars  28449 Javier\_Bardem  28410 simple  28386 Search.php  28270 Fun\_(band)  28199 Sex  **28120 Electoral\_College\_(United\_States)**  27831 Adolf\_Hitler  27769 Image:Dilbert\_time\_wasting\_morons\_book.jpg  27659 Assassin%27s\_Creed  **27362 Rihanna**  27118 Sebastian\_Vettel  27117 Specjalna:Szukaj  27040 Mario\_Maurer |

**Top News and Events of the Day** (Nov 4th, 2012)

To get the top news of the day for Nov 4th, 2012, I explored the archives of various news sources. **TVnewsArchive.org**, a web based news archiving system for the top trending topics of the day on US television. Samples of the Top news of the day for **Nov 4th, 2012** are listed below:



|  |  |  |
| --- | --- | --- |
| SHOWS | STATIONS | TOPIC |
| [Book TV](https://archive.org/details/tv?q=obama&time=20121104&fq=program:%22Book+TV%22) 10  [CNN Newsroom](https://archive.org/details/tv?q=obama&time=20121104&fq=program:%22CNN+Newsroom%22) 10  [The Daily Rundown](https://archive.org/details/tv?q=obama&time=20121104&fq=program:%22The+Daily+Rundown%22) 8  [America's Election Headquarters](https://archive.org/details/tv?q=obama&time=20121104&fq=program:%22America%27s+Election+Headquarters%22) 6  [Geraldo at Large](https://archive.org/details/tv?q=obama&time=20121104&fq=program:%22Geraldo+at+Large%22) 6  [The Ed Show](https://archive.org/details/tv?q=obama&time=20121104&fq=program:%22The+Ed+Show%22) 6  [Meet the Press](https://archive.org/details/tv?q=obama&time=20121104&fq=program:%22Meet+the+Press%22) 5  [Road to the White House](https://archive.org/details/tv?q=obama&time=20121104&fq=program:%22Road+to+the+White+House%22) 5  [ABC News Good Morning America](https://archive.org/details/tv?q=obama&time=20121104&fq=program:%22ABC+News+Good+Morning+America%22) 4  [FOX News Sunday With Chris Wallace](https://archive.org/details/tv?q=obama&time=20121104&fq=program:%22FOX+News+Sunday+With+Chris+Wallace%22) 4 | [MSNBC](https://archive.org/details/tv?q=obama&time=20121104&fq=channel:%22MSNBC%22) 19  [MSNBCW](https://archive.org/details/tv?q=obama&time=20121104&fq=channel:%22MSNBCW%22) 19  [CNN](https://archive.org/details/tv?q=obama&time=20121104&fq=channel:%22CNN%22) 18  [CNNW](https://archive.org/details/tv?q=obama&time=20121104&fq=channel:%22CNNW%22) 18  [CSPAN2](https://archive.org/details/tv?q=obama&time=20121104&fq=channel:%22CSPAN2%22) 13  [FOXNEWS](https://archive.org/details/tv?q=obama&time=20121104&fq=channel:%22FOXNEWS%22) 13  [FOXNEWSW](https://archive.org/details/tv?q=obama&time=20121104&fq=channel:%22FOXNEWSW%22) 13  [CSPAN](https://archive.org/details/tv?q=obama&time=20121104&fq=channel:%22CSPAN%22) 12  [FBC](https://archive.org/details/tv?q=obama&time=20121104&fq=channel:%22FBC%22) 11  [KNTV](https://archive.org/details/tv?q=obama&time=20121104&fq=channel:%22KNTV%22) (NBC) 9  [WJLA](https://archive.org/details/tv?q=obama&time=20121104&fq=channel:%22WJLA%22) (ABC) 9  [WRC](https://archive.org/details/tv?q=obama&time=20121104&fq=channel:%22WRC%22) (NBC) 9 | [obama](https://archive.org/details/tv?q=news&time=20121103&fq=topic:%22obama%22) 205  [sandy](https://archive.org/details/tv?q=news&time=20121103&fq=topic:%22sandy%22) 182  [romney](https://archive.org/details/tv?q=news&time=20121103&fq=topic:%22romney%22) 181  [ohio](https://archive.org/details/tv?q=news&time=20121103&fq=topic:%22ohio%22) 144  [florida](https://archive.org/details/tv?q=news&time=20121103&fq=topic:%22florida%22) 127  [new york city](https://archive.org/details/tv?q=news&time=20121103&fq=topic:%22new+york+city%22) 123  [pennsylvania](https://archive.org/details/tv?q=news&time=20121103&fq=topic:%22pennsylvania%22) 114  [fema](https://archive.org/details/tv?q=news&time=20121103&fq=topic:%22fema%22) 100  [mitt romney](https://archive.org/details/tv?q=news&time=20121103&fq=topic:%22mitt+romney%22) 98  [new jersey](https://archive.org/details/tv?q=news&time=20121103&fq=topic:%22new+jersey%22) 93 |

**Google Trends** for November 2012 also has a summary of the most trending news items . Below is a shot of the trending news items categories.

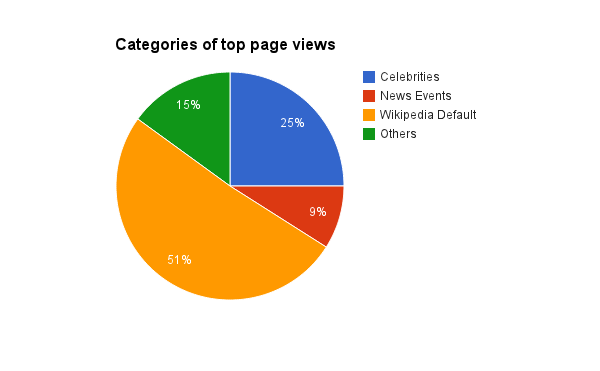
|  |  |  |
| --- | --- | --- |
| Screen Shot 2014-12-01 at 3.04.11 PM.png | Screen Shot 2014-12-01 at 3.05.42 PM.png | Screen Shot 2014-12-01 at 3.04.22 PM.png |
| Screen Shot 2014-12-01 at 3.04.49 PM.png | Screen Shot 2014-12-01 at 3.05.19 PM.png | Screen Shot 2014-12-01 at 3.04.58 PM.png |

**Analysis (Verifying output)**

As expected some of the pages with the hightest views are wikipedia program pages like **en, fr**  that redirects the user to an english and french versions of such that article. Also pages like the **404 error** pages get lots of pageviews. For the purpose of simplicity we will ignore these pages and focus on articles.   
I’ve marked a few of the top pages on Wikipedia for **Nov 4th** with **red** below so we can compare them to the top news of the day both on local US televisions and worldwide google trends.

|  |  |  |
| --- | --- | --- |
| **TOP WIKI PAGES** | **GOOGLE TRENDS** | **TV NEWS TOP TOPICS** |
| **129038 Skyfall**  **117858 James\_Bond**  **72986 Daniel\_Craig**  **75117 Gangnam\_Style**    **59753 Barack\_Obama**  **58838 Mitt\_Romney**  **28120 Electoral\_College\_(United\_States)**  **37996 Hurricane\_Sandy**  **92952 One\_Direction**  **33332 Justin\_Bieber**  **27362 Rihanna** | Screen Shot 2014-12-01 at 3.04.11 PM.png  Screen Shot 2014-12-01 at 3.05.42 PM.png  Screen Shot 2014-12-01 at 3.04.22 PM.png  Screen Shot 2014-12-01 at 3.05.19 PM.png  Screen Shot 2014-12-01 at 3.04.49 PM.png | Skyfall in theaters  O[bama](https://archive.org/details/tv?q=news&time=20121103&fq=topic:%22obama%22) (205) R[omney](https://archive.org/details/tv?q=news&time=20121103&fq=topic:%22romney%22) (181)  Sandy (182) [new jersey](https://archive.org/details/tv?q=news&time=20121103&fq=topic:%22new+jersey%22) (93) fema (100) |

As we can probably tell from the table above, The top trending topics on google and news stations also appear amongst the top read articles on Wikipedia for that day. But an observation that can be made is some data tend to occupy the top pageviews though they have no relevant news value at the time. Celebrities are more likely to be searched for. Below is a graph of the **top 100** results and how they differ by characteristics.



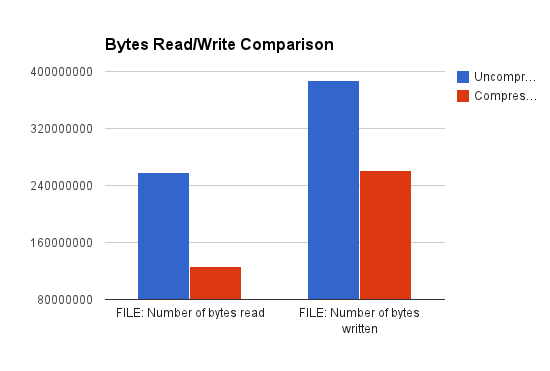
**Performance/scale characteristics**

Below is a the Stats page for the two mapreduce jobs.

|  |  |
| --- | --- |
| **MapReduce 1** | **MapReduce 2** |
| INFO mapreduce.Job: Counters: 51  **File System Counters**  **FILE: Number of bytes read=2590731186**  **FILE: Number of bytes written=4135284867**  FILE: Number of read operations=0  FILE: Number of large read operations=0  FILE: Number of write operations=0  HDFS: Number of bytes read=995820224  HDFS: Number of bytes written=1369083223  HDFS: Number of read operations=174  HDFS: Number of large read operations=0  HDFS: Number of write operations=96    **Job Counters**  **Launched map tasks=24**  **Launched reduce tasks=48**  Data-local map tasks=9  Rack-local map tasks=1  Total time spent by all maps in occupied slots (ms)=446823  Total time spent by all reduces in occupied slots (ms)=363091  **Total time spent by all map tasks (ms)=446823**  **Total time spent by all reduce tasks (ms)=363091**  Total vcore-seconds taken by all map tasks=446823  Total vcore-seconds taken by all reduce tasks=363091  Total megabyte-seconds taken by all map tasks=457546752  Total megabyte-seconds taken by all reduce tasks=371805184  **Map-Reduce Framework**  **Map input records=78937786**  **Map output records=78937786**  Map output bytes=3024348914  Shuffle Errors  BAD\_ID=0  CONNECTION=0  IO\_ERROR=0  WRONG\_LENGTH=0  WRONG\_MAP=0  WRONG\_REDUCE=0  File Input Format Counters  Bytes Read=995818664  File Output Format Counters  Bytes Written=1369083223  wiki.pagecountMapper$Wiki  **BAD\_RECORDS=86**  **NUM\_OF\_WIKI\_PAGES=78937786** | INFO mapreduce.Job: Counters: 50  **File System Counters**  **FILE: Number of bytes read=781752156**  **FILE: Number of bytes written=1569222140**  FILE: Number of read operations=0  FILE: Number of large read operations=0  FILE: Number of write operations=0  HDFS: Number of bytes read=1369089943  HDFS: Number of bytes written=149362  HDFS: Number of read operations=147  HDFS: Number of large read operations=0  HDFS: Number of write operations=2    **Job Counters**  **Launched map tasks=48**  **Launched reduce tasks=1**  Data-local map tasks=43  Rack-local map tasks=5  Total time spent by all maps in occupied slots (ms)=335483  Total time spent by all reduces in occupied slots (ms)=64843  **Total time spent by all map tasks (ms)=335483**  **Total time spent by all reduce tasks (ms)=64843**  Total vcore-seconds taken by all map tasks=335483  Total vcore-seconds taken by all reduce tasks=64843  Total megabyte-seconds taken by all map tasks=343534592  Total megabyte-seconds taken by all reduce tasks=66399232  **Map-Reduce Framework**  **Map input records=31887860**  **Map output records=31887860**  Map output bytes=1431183272  Shuffle Errors  BAD\_ID=0  CONNECTION=0  IO\_ERROR=0  WRONG\_LENGTH=0  WRONG\_MAP=0  WRONG\_REDUCE=0  File Input Format Counters  Bytes Read=1369083223  File Output Format Counters  Bytes Written=149362 |

Overall the runs of both Mapreduce Jobs were relatively fast with taking a combined **10 minutes** to complete. Another factor that contributed to the speedy processing was putting no restrictions on the number of mapper tasks.

Compression of the files meant that the bytes read and written for both compressed and uncompressed file format shows a big performance gain when the files are compressed and ensured that file movement and processing took less time.



Another metric we have to consider is the computational complexity of the **mapreduce1** and **mapreduce2**. Given that the complexity of the map and reduce tasks are O(map)=f(n) and O(reduce)=g(n) for most cases, this was the case for the two mapper classes and reducer classes. In our case it took 10min to process the one day mapreduce jobs, it will take approximately 20 minutes to process a two day mapreduce job assuming the file sizes for the datasets are similar.

**Future Research**

There are several changes or additions to this project that would improve it performance, scale and it’s conclusiveness. One main improvement to the project would be to use **PIG** so we don’t have to write a different mapper class to do the sorting.

Running a month or year worth of data might have given us a better indication of what articles a truely the top read articles on Wikipedia.

Also, an interesting project would be to try to see if the results of the page views can be predictive of what people will search for in the future.

**Conclusion**

Although celebrities and famous topics tend to occupy the top read articles along with wiki default pages, events like Hurricane Sandy and the Presidential elections and topic related to them which were actual news event appear amongst the top read articles on Wikipedia,   
Knowing this, we can conclude that news events can generate huge number of page traffic on wikipedia to launch them to the top viewed pages on a particular day.

**References**  
1,2 <http://dumps.wikimedia.org/other/pagecounts-raw/> Page view statistics for Wikimedia projects. Retrieved 2014-11-04

3 <https://aws.amazon.com/datasets/2596> Wikipedia Page Traffic Statistics. Retrieved 2014-11-04