ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 4, April 2014

Analysis of Log Data and Statistics Report Generation Using Hadoop

Siddharth Adhikari[1], Devesh Saraf[2], Mahesh Revanwar[3], Nikhil Ankam[4]

B.E Student, Department of Computer Engineering. of Computer, Vishwakarma Institute of Information Technology, Pune, India[1]

B.E Student, Department of Computer Engineering. of Computer, Vishwakarma Institute of Information Technology, Pune, India[2]

B.E Student, Department of Computer Engineering. of Computer, Vishwakarma Institute of Information Technology, Pune, India[3]

B.E Student, Department of Computer Engineering. of Computer, Vishwakarma Institute of Information Technology, Pune, India[4]

ABSTRACT: Web Log analyser is a tool used for finding the statics of web sites. Through Web Log analyzer the web log files are uploaded into the Hadoop Distributed Framework where parallel procession on log files is carried in the form of master and slave structure. Pig scripts are written on the classified log files to satisfy certain query. The log files are maintained by the web servers. By analysing these log files gives an idea about the user in the way like which IP address have generated the most errors, which user is visiting a web page frequently.. This paper discuss about these log files, their formats, access procedures, their uses, the additional parameters that can be used in the log files which in turn gives way to an effective mining and the tools used to process the log files. It also provides the idea of creating an extended log file and learning the user behaviour. Analysing the user activities is particularly useful for studying user behaviour when using highly interactive systems. This paper presents the details of the methodology used, in which the focus is on studying the information-seeking process and on finding log errors and exceptions. The next part of the paper describes the working and techniques used by web log analyzer.

KEYWORDS: Hadoop, MapReduce, Pig, Web log files.

I. INTRODUCTION

Web Log Analyzer is a fast and powerful log analyzer [5]. It gives you information about site's visitors: activity statistics, accessed files, paths in the sites, information about referred pages, browsers, operating systems etc.[4]. The program produces easy-to-read reports that include both text information (tables) and charts. View the Web Log Analyzer sample report to get the general idea of the variety of information about your site's usage it can provide. The log analyzer can create reports in in HTML, PDF and CSV formats. It also includes a web server that supports dynamic HTML reports [5]. Web Log Analyzer can analyse logs of Apache and IIS web servers log files. It can even read ZIP compressed log files. Log files are files that list the actions that have been occurred. These log files reside in the web server. The Web server stores all of the files necessary to display the Web pages on the user's computer [3]. All the individual web pages combines together to form the completeness of a Web site.

The browser requests the data from Web server, and then by using HTTP, the server delivers the data back to the browser that had requested the web page [4]. The browser then converts, or formats, the files into a page which is viewable by the user. This gets displayed in the browser [3]. In the same way the server can send the files to many client computers simultaneously, which allows multiple clients to view the same page simultaneously [6].the next section of the paper focuses on the contents of web log files.

II. RELATED WORK

A. Tableau:

Tableau Desktop is based on breakthrough technology from Stanford University that lets you drag & drop to analyse data. You can connect to data in a few clicks, then visualize and create interactive dashboards with a few more [9]. Years of research has been done to build a system that supports people's natural ability to think visually. Shift fluidly between views, following your natural train of thought. You're not stuck in wizards or bogged down writing scripts.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 4, April 2014

You just create beautiful, rich data visualizations [9]. It's so easy to use that any Excel user can learn it. Get more results for less effort. And its 10–100 times faster than existing solutions. Tableau Desktop is an intuitive, drag-and-drop tool that lets you see every change as you make it. Work at the speed of thought, without ever taking your eyes off the data. Anyone comfortable with Excel can get up to speed on Tableau quickly. Business leaders use it to see and understand many facets of their business at a glance. Scientists use it to create sophisticated trend analyses. Marketers use it to make data-driven decisions that drive ROI through the roof [9]. Tableau's user interface is easy to use; Tableau lets you create rich visualizations and dashboards in minutes. Working with massive data? Tableau's Data Engine is blazing fast. With Tableau you don't have to use anything less than all the data you need. Tableau's architecture-aware approach means you can bring all your data right onto your laptop and still have interactive response time. Work with hundreds of millions of rows of data right on your own computer. Get answers in seconds. It's real-time business analytics for real-time business data [9].

B. Contents of Log Files:

The Log files in different web servers maintain different types of information [2]. The basic information present in the log file is as follows.

- IP Address/User name*:* This identifies who had visited the site. The identification of the user is mostly by using the IP address. This may be a temporary address that had allotted. Therefore the unique identification of the user is not achieved. In some web sites the user identification is made by getting the user profile and allows them to access the web site by using a user name and password. In this kind of access the user is being identified uniquely so that the revisit of the user can also be identified.
- Time stamp*:* The time spent by the user in each web page while surfing through the website. This is identified as the session.
- Page visited lastly: The page that was visited by the user before he or she leaves the website [2].
- Success rate*:* The success rate of the web site can be determined by the number of downloads made and the number copying activity under gone by the user [7].
- User Agent: This is nothing but the browser from where the user sends the request to the web server. It's just a string describing the type and version of browser software being used.
- URL: The resource accessed by the user. It may be an HTML page or a script.
- Request type: The method used for information transfer is noted. The methods like GET, POST.

The following sections in the paper describe the tools and techniques used for scripting, displaying, accessing and querying the data.

III. PROPOSED ALGORITHM

1. Map-Reduced Algorithm:

The primary objective of Map/Reduce is to split the input data set into independent chunks that are processed in a completely parallel manner. The Hadoop MapReduce framework sorts the outputs of the maps, which are then input to the reduce tasks. Typically, both the input and the output of the job are stored in a file system.

MapReduce is as a 5-step parallel and distributed computation:

1. Map() input – the "MapReduce system" designates Map processors, assigns the K1 input key value each processor would work on, and provides that processor with all the input data associated with that key value.
2. Map () code – Map () is run exactly once for each K1 key value, generating output organized by key values K2.
3. "Shuffle"– the MapReduce system designates Reduce processors, assigns the K2 key value each processor would work on, and provides that processor with all the Map-generated data associated with that key value.
4. Reduce () code – Reduce () is run exactly once for each K2 key value produced by the Map step.
5. Final output – the MapReduce system collects all the Reduce output, and sorts it by K2 to produce the final outcome.

Logically these 5 steps can be thought of as running in sequence – each step starts only after the previous step is completed – though in practice, of course, they can be intertwined, as long as the final result is not affected.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 4, April 2014

A) Map stage:
Input text is one record each line, we use the Long Writable and Text in MapReduce package as the initial input types of key and value respectively, where the value of key is the offset of each line, and the value of value is the content of the corresponding line. In main function, Map analyses the value, extracts the user ID and his keywords in this searching, and regards the user ID as the middle value of key whose type is Text and the keywords which values are input.

B) Reduce stage:
Firstly, statistics on the keywords of the same user ID. If one same keyword appears many times, plus one every time. At last, put final result into HDFS. Because keys have been sorted before Reduce, all the values of same key are put together, namely have been encapsulated as Iterator <Value Type>. These all can be processed at the same time, and the final output is value which encapsulates the same user's all key words and weights. The middle key is taken as the final output without processing.

IV. PSEUDO CODE

A. **Error count in log files**

Step 1: class Mapper
Step 2: method Map(docid a, doc d)
Step3: for all error_no e belongs doc d do
Step4: Emit(term e; count 1)

Step1: class Reducer
Step 2: method Reduce(term t, counts [c1, c2,…])
Step3: sum = 0
Step 4: for all count c belongs to counts [c1, c2,…] do
Step5: sum = sum + c

B. **Running Pig on Local Mode:**
Step 1: Java –Xmx256m –cp pig.jar org.apache.pig.Main -x local script1- local.pig
Step 2: Java -Xmx256m -cp pig.jar org.apache.pig.Main -x local script2-local.pig
Step 3: Move to the pigtmp directory.
Step 4: Review Pig Script 1 and Pig Script 2.
Step 5: Execute the following command (using either script1-local.pig or script2-local.pig).
$ java –cp $PIGDIR/pig.jar org.apache.pig.Main -x local script1-local.pig
Step 6: Review the result files, located in the part-r-00000 directory.
Step 7: The output may contain a few Hadoop warnings which can be ignored:
2010-04-08 12:55:33,642 [main] INFO
org.apache.hadoop.metrics.jvm.JvmMetrics
- Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized

C. **Running Pig on Map Reduce Mode:**
Pig translates the queries into MapReduce jobs and runs the job on the hadoop cluster. This cluster can be pseudo- or fully distributed cluster.
Step 1: check the compatibility of the Pig and Hadoop versions being used. Export the variable PIG_CLASSPATH to add Hadoop conf directory
Step 2: $ export PIG_CLASSPATH=$HADOOP_HOME/conf/
Step 3: After exporting the PIG_CLASSPATH, run the pig command, as shown below
$ pig
Step 4: INFO org.apache.pig.Main – Apache Pig version 0.10.0 (r1328203) compiled Apr 19 2012, 22:54:12
Step 5: INFO org.apache.pig.Main – Logging error messages to: /Users/varadmeru/pig_1351858332488.log

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 4, April 2014

Step 6: INFO org.apache.pig.backend.hadoop.executionengine.HExecutionEngine – Connecting to Hadoop filesystem at: hdfs: //localhost:9000
Step 7: INFO org.apache.pig.backend.hadoop.executionengine.HExecutionEngine – Connecting to map-reduce job tracker at: localhost: 9001
Grunt>

V. SIMULATION RESULTS

A. Experiments:

In this section, we show the study results from different sources that the feasibility, speedup, validity and efficiency of Map reduce algorithm by two experiments. The experimental Hadoop cluster is composed of one Master machine and two Slave machines with Intel Pentium® Dual Core E5700 3.00GHz CPU and 4.00GB RAM. All the experiments are performed on Ubuntu 12.04 OS with Hadoop 0.20.2, Jdk 1.6.0 and Eclipse 3.7.1. The log data from different web sites are used as the experimental data, which are classified into groups of different datasets. Fig The feasibility, validity, speedup and efficiency are used to evaluate the overall performance of Map Reduce algorithm and compare it with the Pig in the same environment. The data set is processed using pig scripts and statics about the log data are generated. These statics are made available to the user interface where they are displayed on a button click. Following shows the statics generated after processing log data sets. Fig 1. Show the 3 dimensional pie chart generated using Eclipse which shows the different types of error codes occurring number of times. Similarly Fig 2. shows the bar graph plotted for error codes and their occurences.

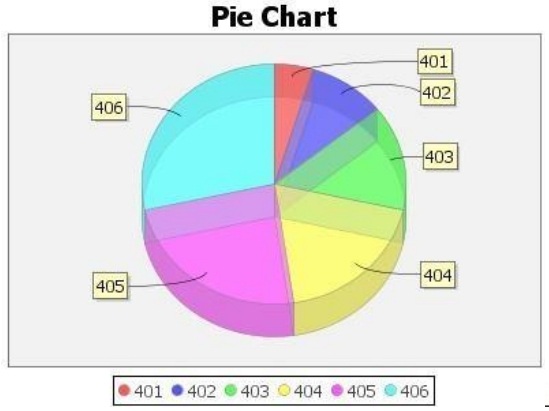

Fig.1.Pie chart generated showing types of errors

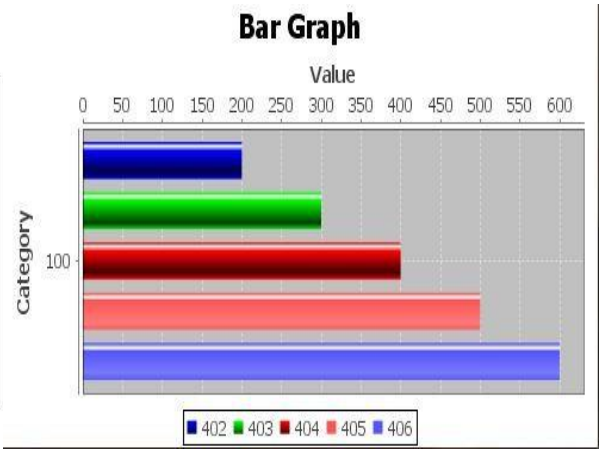

Fig.2. Bar graph showing error rates

VI. CONCLUSION AND FUTURE WORK

This Paper describes a detailed view of Hadoop framework used to process big data .It even gives a description of how the log file is processed for exceptions and errors. This paper describes how a log file is processed using map reduce technique. Hadoop framework is used as it is beneficial for parallel computation of log files. Along with that it also tells about the brief introduction to pig tool which is used for taking a huge data from different sources and places it into HDFS for further processing. As described in the paper, the framework makes use of tableau tool for pictorial representation of log files accessed by the users.

B. Future Scope:

In computer science, event monitoring is the process of collecting, analyzing, and signalling event occurrences to subscribers such as operating system processes, active database rules as well as human operators. These event occurrences may stem from arbitrary sources in software or hardware such as operating systems, database management systems, application software and processors.Event monitoring makes use of a logical bus to transport event

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 4, April 2014

occurrences from sources to subscribers, where event sources signal event occurrences to all event subscribers and event subscribers receive event occurrences [11]. An event bus can be distributed over a set of physical nodes such as standalone computer systems. Typical examples of event buses are found in graphical systems such as X Window System, Microsoft Windows as well as development tools such as SDT[12].

REFERENCES.

1. L.K. Joshila Grace, V. Maheswari, and Dhinaharan Nagamalai (Jan 2011)" *Web Log Data Analysis and Mining*" in Proc CCSIT-2011, Springer CCIS, Vol 133, pp 459-469
2. R. J. Williams D. E. Rumelhart, G. E. Hinton. Learning representation by back-propagating errors. InNature, volume 323, pages 533–536, 1986.
3. Kommineni, M., & Parvathi, R. (2013). Risk Analysis for Exploring the Opportunities in Cloud Outsourcing.
4. J. Dean, S. Ghemawat, *"MapReduce: Simplified Data Processing on Large Clusters*," In Proc. of the 6th Symposium on Operating SystemsDesign and Implementation, San Francisco CA, Dec. 2004.
5. T. Hastie and R. Tibshirani. Discriminant analysis by gaussian mixtures. Journal of the Royal Statistical Society B, pages 155–176, 1996
   J. Dean and S. Ghemawat. "MapReduce: Simplified Data Processing on Large Clusters". Commune. ACM, 51(1):107–113, 2008.
6. M. Zaharia, D. Borthakur, J. S. Sarma, K. Elmeleegy, S. Shenker, and I. Stoica, "Job scheduling for multi-user map reduce clusters," EECS Department, University of California, Berkeley, Tech. Rep., Apr 2009.
7. Vadivel, R and V. Murali Bhaskaran,'Energy Efficient with Secured Reliable Routing Protocol (EESRRP) for Mobile Ad-Hoc Networks', Procedia Technology 4,pp. 703- 707, 2012.
8. Pavan Reddy, Vaka (2012). Zero-Day Vulnerabilities. International Journal of Innovative Research in Science, Engineering and Technology 1 (2):318-322.
9. Daryl Pregibon. Logistic regression diagnostics. In The Annals of Statistics, volume 9, pages 705–724, 1981.
   R. L¨ammel. Google's MapReduce Programming Model – Revisited. Draft; Online since 2 January, 2006; 26 pages, 22 Jan. 2006.
10. A. C. Arpaci-Dusseau et al. High-Performance Sorting on Networks of Workstations. In SIGMOD 1997, Vol 12, pages 243–254, 1997.
11. F. Chang et al. Bigtable: A distributed storage systemfor structured data. In Proc. OSDI, Vol 1, pages 205–218.USENIX Association, 2006.
12. M. Davis, G. Efstathiou, C. S. Frenk, and S. D. M. White, "The evolutionof large-scale structure in a universe dominated by cold dark matt er,"Astroph J, vol. 292, pp. 371–394, May 1985.
13. D. H. Weinberg, L. Hernquist, and N. Katz, "Photoionization, NumericalResolution, and Galaxy Formation," Astroph. J., vol. 477, pp. 8 +, Mar.1997.
14. Mohit, Mittal (2013). The Rise of Software Defined Networking (SDN): A Paradigm Shift in Cloud Data Centers. International Journal of Innovative Research in Science, Engineering and Technology 2 (8):4150-4160.
15. R. Pike, S. Dorward, R. Griesemer, and S. Quinlan, "Interpreting thedata: Parallel analysis with Sawzall," Scientific Programming, vol. 13,no. 4, 2005.
16. S. R. Knollmann and A. Knebe, "AHF: Amiga's Halo Finder," Astroph.J. Suppl., vol. 182, pp. 608–624, June 2009.
    H. Boral, W. Alexander, L. Clay, G. Copeland, S. Danforth, M. Franklin,B. Hart, M. Smith, and P. Valduriez, "Prototyping Bubba, a highlyparallel database system," IEEE TKDE, vol. 2, no. 1, pp. 4–24, 1990.
17. D. DeWitt and J. Gray, "Parallel database systems: the future of highperformance database systems," Communications of the ACM, vol. 35,no. 6, pp. 85–98, 1992.