Trend Analysis based on Access Pattern over Web Logs using Hadoop

[1]Snehal S. Sonawane, [2]Sonal S. Patil, [3]Supriya D. Sonawane, [4]Saurabh Garud

SSBT's College of Engineering & Technology
Bambhori, Jalgaon - 425 001 (MS)

Abstract- With growing advancement in the electronic commerce field, trend analysis is spreading all over the world. The invariable progress of World Wide Web leads to the generation of Log file. Log file contain parameter like user names, IP addresses, URLs, Page links, Browsers, Operating system etc. we use Hadoop as platform. Hadoop is use for Distributed File System. This Paper can analyze the trend based On User behavior for this Purpose we used the Hadoop MapReduce function. The key terms and algorithm uses are Hadoop Map Reduce Function,Word count, AlltransactionReducer etc.In this Paper we give the input as Web Log File we can preprocess that Log file using Data Preprocessing tools then apply Hadoop Mapreduce function on log file. The output should be in the form of Graphs, Piechart or any graphical representation. The primary Aim of the paper is to construct log analysis system which depicts trends based on the users behaviour mode using Hadoop MapReduce which facilitates handling of heterogeneous query execution on log file.

***Keywords* - Cloudera, Hadoop, MapReduce, Log Files, Web Mining, MySql Database, Hadoop Distributed File System, Trend Analysis.**

I. INTRODUCTION

In trend analysis to find a current trend based on their access pattern over the set of log files .These log files provide important information about the functioning and response of applications and devices. Web logs are store a click script data which can be useful for mining purpose. Web Log is the outcome of web usage mining which contains information of web access of the users. For systematic development in the field of technology in sectors such as business, public and private has been observed for deploying a large amount of data over the web. Hadoop MapReduce is the one function which facilitates handling of multiple query execution on log file, which helps for reduce the query execution time, analyzing such data is necessary to take accuracy where in log file analysis is an effective solution. Log files are the files that list the actions or holds the information about trends that have been occurred and reside in web server. Analysis involves to discover the meaningful and understandable patterns from the various types of log files. A need to process and store log files using traditional techniques and using platform of Hadoop.

II. PROBLEM DEFINITION

For achieving trend analysis using access pattern over a weblog there prevails a need to process and store log file so that there should be a system whose approach to the higher processing capacity and competent for gathering information for processing. These log files are stored and processed using traditional techniques such as virtual database is one of the executive technique. But since log files over the web are outsized storage become a constraint where in executive techniques such as virtual database prove to be inexectual for the same. Hadoop over a large scale distributed batch processing infrastructures that provide adequate data storage distributive and analogous processing, isolation of process and fault tolerant on occurrence of data loss. Hadoop mapreduce facilitate handling of heterogeneous query execution on log file. proposed system is about to managing the large chunk of weblog using hadoop mapreduce which reduces the response time for throughput generation loads the log data executively and ensures reliability.

III. SYSTEM DESIGN

Web log analysis for determining the trends is carried out using Hadoop approach. Hadoop makes use of Map and Reduce technique to fulfill the purpose with the help of java.util.regex package for pattern matching with regular expressions. The common log file or combined log format is expressed using regular expression, so as to obtain different fields from the log file and grouping them on the basis different categories such as IP address, username, request type, requested URL, status codes, size of page, referrer and user agent. The access log files which are available in the form of text file format are input files to MapReduce.

Implementation of Mapper

- Map is a special function that applies the function f to each element in the list.
- Map[f,(1,2,3,4,5)] = (f[1],f[2],f[3],f[4],f[5])
- Input:
- The Entire Data Set
- Maps all the input values to a key
- map() is called once for each line of input

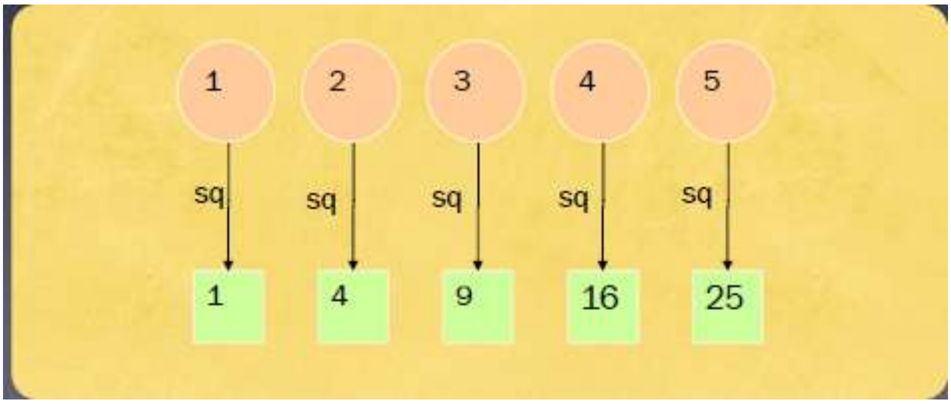


Fig: Mapper

- Output:
- Collects (key, value) pairs
- Passes to reducer as hashmap.

Implementation of Reducer

- Reduce[f,x,list]
- Sets an accumulator
- Initial value is x
- Applies f to each element of the list plus the accumulator
- Result is the final value of the accumulator
- Reduce[f,x,a,b,c] = f[f[f[x,a],b],c]

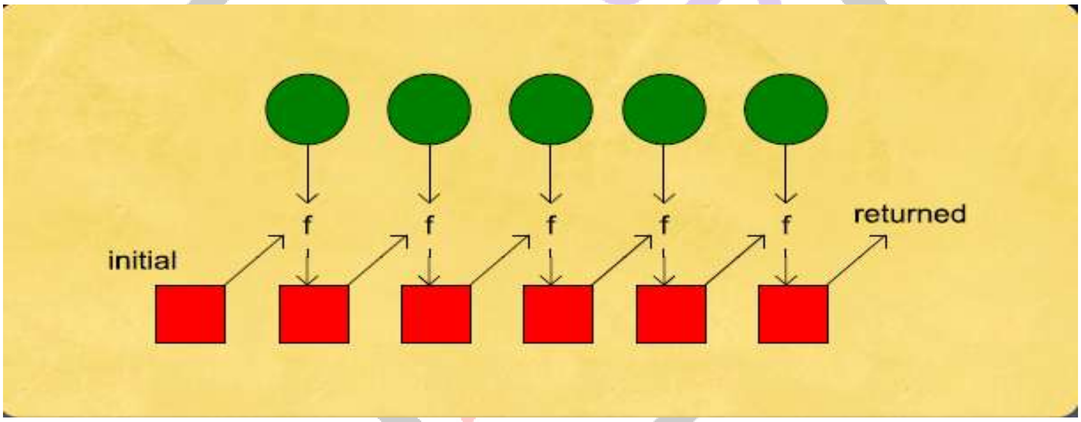


Fig: Reducer

Implementation of Word count

The mapper emits an intermediate key-value pair for each word in a document. The reducer sums up all counts for each word.

```
1. class Mapper
2. method Map(docid a, doc d)
3. for all term t 2 doc d do
4. Emit(term t, count 1)
1. class Reducer
2. method Reduce(term t, counts [c1, c2])
3. sum=0
4. for all count c 2 counts [c1,c2] do
5. sum = sum + c
6. Emit(term t, count sum)
```

Implementation Environment

This section specifies the various software names and their versions which are required for the development of this project. This section shows the implementation environment of the project. Thus the softwares mandatory for the implementation of this project are as follows

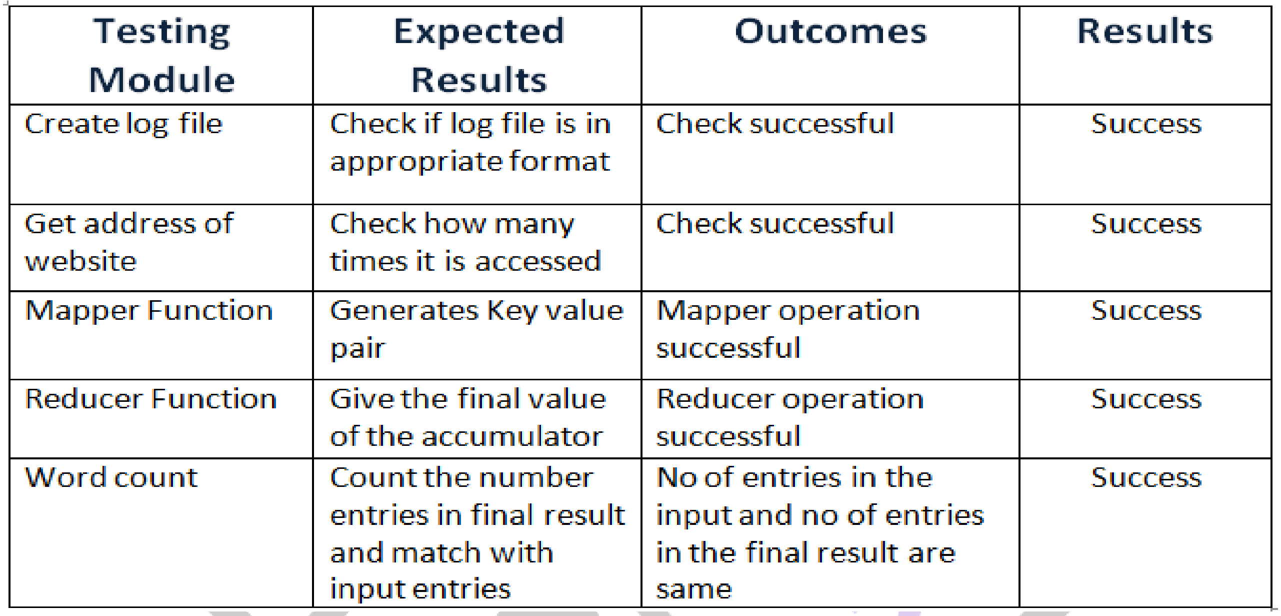
- Java java version 7.0 is the basic requirement for this project to execute.
- Apache Tomcat Apache Tomcat is an open-source web server and servlet container developed by the Apache Software Foundation (ASF). Tomcat implements several JavaEE specifications including Java Servlet, JavaServer PagesJSP, Java EL, and

WebSocket, and provides a "pure Java" HTTP web server environment for Java code to run in Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation, released under the Apache License 2.0 license, and is open-source software. In this project, the apache Tomcat version 7.0 is used.

IV. SYSTEM TESTING

Test case is an object for execution for other modules in the architecture does not represent any interaction by itself. A test case is a set of sequential steps to execute a test operating on a set of predefined inputs to produce certain expected outputs. There are two types of test cases manual and automated. A manual test case is executed manually while an automated test case is executed using automation. In system testing, test data should cover the possible values of each parameter based on the requirements. Since testing every value is impractical, a few values should be chosen from each equivalence class. An equivalence class is a set of values that should all be treated the same. Testing for the entire project is described in the following table.

Table: Testing carried for trend analysis



Testing Module	Expected Results	Outcomes	Results
Create log file	Check if log file is in appropriate format	Check successful	Success
Get address of website	Check how many times it is accessed	Check successful	Success
Mapper Function	Generates Key value pair	Mapper operation successful	Success
Reducer Function	Give the final value of the accumulator	Reducer operation successful	Success
Word count	Count the number entries in final result and match with input entries	No of entries in the input and no of entries in the final result are same	Success

V. RESULT

Results and Analysis for Dataset creator file, which includes the modules as read IP address of the device that is been used for trend analysis processing, read page URLs, read dates. The log creator is used for generating a large data set so we require an array list of users, IP's, pages, dates etc. The log creator has to take this modules as an input and create a large data which has to be feed to the Hadoop to create a log file. This log file generated by log creator is used as an input by Hadoop.

- The expected Trend analysis portrays the users browsing pattern and summarizes the outcome into a graphical report which depicts most visited web pages, browsing session and trending keywords. Hadoop MapReduce framework provides parallel distributed processing and reliable data storage for large volumes of log files.

- Hadoop MapReduce plays a key role by proficient management of data and decreases the response time. The proposed system with the help of Hadoop MapReduce analyzes the log files and segregates the fields of the log files using regular expression mechanism. The segregated and structured fields are stored in the database in accordance with Hadoop thereby enabling ease of data retrieval. Module and error generated in matching the overall count of URLs of output file to the URLs of the input weblog file.

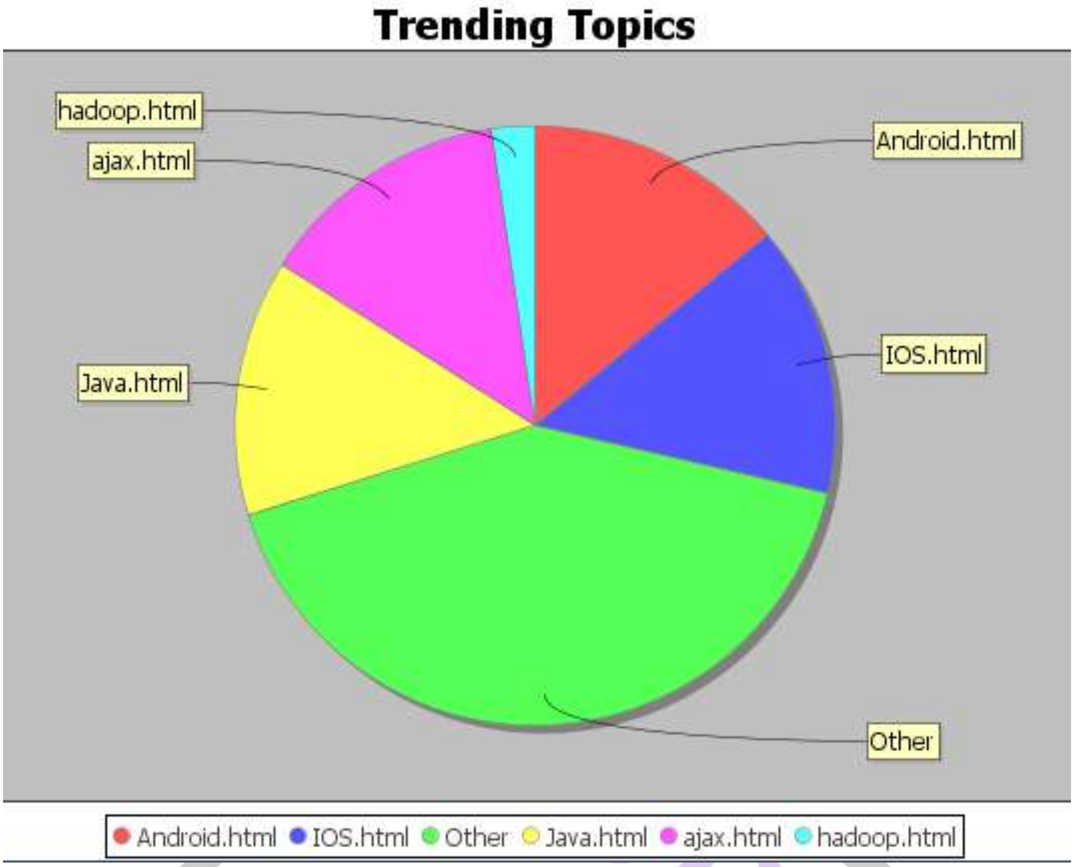


Fig: Piechart for trend analysis

REFERENCES

[1] K.R. Suneetha and Dr. R. Krishnamoorthi, Identifying user behavior by analyzing web server access log file, IJCSNS, Vol.9 No.4, pp. 327-332, April 2009.

[2] Geeta R. B., S. G. Totad and Prasad Reddy, Amalgamation of web usage mining and web structure mining, ACEEE, IJRTE, Vol. 1, No. 2, pp. 279-281, May 2009.

[3] H. Liu and V. Keselj, Combined mining of web server logs and web contents for classifying user navigation patterns and predicting users future requests, In Proc. Elsevier, Data and Knowledge Engineering 61, pp. 304330, 2007.

[4] R. Chourasia and P. Choudhary, An approach for web log preprocessing and evidence preservation for web mining, IJCSE, Vol. 2, Issue 4, pp. 210-215, 2014.

[5] S. O. Fageeri and R. Ahmad, An effcient log file analysis algorithm using binary based data structure, Procedia- Social and Behavioral Sciences 129 pp. 518-526, 2014.

[6] L. K. Joshila Grace, V. Maheswari and D. Nagamalai, Analysis of web logs and web user in web mining, IJNSA, Vol. 3 No. 1, pp. 99-110, 2011.

[7] T. Pamutha, S. Chimphlee, C. Kimpan and P. Sanguansat, Data preprocessing on web server log _les for mining users access patterns, IJRRWC, Vol. 2, No. 2, pp. 92-98, 2012.

[8] A. Guerbas, O. Addam, O. Zaarour, M. Nagi, A. Elhajj, M.Ridley and R. Alhajj, E_ective web log mining and online navigational pattern prediction, In Proc. Elsevier, Knowledge Based Systems 49, pp. 50-62, 2013.

[9] P. Patel and M. Parmar, Improve heuristics for user session identi_cation through web server log in web usage mining, IJCSIT, Vol. 5 No. 3, pp. 3562-3565, 2014.

[10] Savitha K and Vijaya MS, Mining of web server logs in a distributed cluster using big data technologies, IJACSA, Vol. 5 No. 3, pp. 137-142, 2014.