Event Log Analysis: A Systematic Review

Shweta Thakur
Shri Shankaracharya Group of Institutions
Dept. of Computer Science and Engineering
Bhilai, Chhattisgarh, India
Shwetathakur181291@gmail.com

Prof. Sampada Vishwas Massey
Shri Shankaracharya Group of Institutions
Dept. of Computer Science and Engineering
Bhilai, Chhattisgarh, India
sampada.satav@gmail.com

Abstract

Log is main source of any system operations status, system actions, web user behaviors etc. There are various sources of generation of logs. These logs are in around petta bytes to zetta bytes in size. There are many powerful tool to process the data but they are not sufficient as they are limited by size of file. Not only processing, the storage of these big amount of data are also very crucial. There may be some performance as well as security issues in handling these much amount of data. This paper reviews some of the technique used for log analysis. This paper includes step by step introduction to Hadoop, MapReduce Programming Model and its working process.

Keywords— *Hadoop, MapReduce, Log File, Click Stream Data..*

I. INTRODUCTION

Log files which are generated by the company such as Amazon, Google, and Facebook are increasing as a record rate. In a day about petabytes and terra bytes of data are being generated by a data center. The most challenging task of these stored data is to Store and analyze this huge amount of data. The problem not only related to its volume but the major problem is lies in its structure. The log is collected from various sources and has a very complex structure. Processing the huge amount of data and retrieving the relevant information out of this is a tedious task.

Now the questions raise, Is Traditional database system are capable of processing this huge amount of logs generated if we provide huge volume to database system? Answer is No!

Why No! The answer to these questions comes from another trend in disk drives: seek time is improving more slowly than transfer rate. Seeking is the process of moving the disk's head to a particular place on the disk to read or write data. It characterizes the latency of a disk operation, whereas the transfer rate corresponds to a disk's bandwidth. If the data access pattern is dominated by seeks, it will take longer to read or write large portions of the dataset than streaming through it, which operates at the transfer rate. On the other hand, for updating a small proportion of records in a database, a traditional B-Tree (the data structure used in relational databases, which is limited by the rate it can perform, seeks) works well. For updating the majority of a database, a B-Tree is less efficient than MapReduce, which uses Sort/Merge to rebuild the database. [1]

The differences between the two systems are shown in Table 1.

TABLE- 1: RDBMS Compared to MapReduce

	Traditional RDBMS	MapReduce
Data Size	Gigabytes	Petabytes
Access	Interactive and Batch	Batch
Updated	Read and Write many times	Write once, read many times
Structure	Static Schema	Dynamic Schema
Integrity	High	Low
Scaling	Nonlinear	linear

A. Apache Hadoop

The Apache Hadoop is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. [2]

Hadoop enables application to work in a distributed environment. There may be thousands of distributed component working together to accomplish a single task. Generally the huge log files are distributed over various clusters known as HDFS cluster (Hadoop distributed file system). Hadoop breaks up the records into the number of block and these blocks are distributed over various clusters. And they are processed in each system in a parallel fashion. Performance of the hadoop system is gained by operating files in parallel environment.

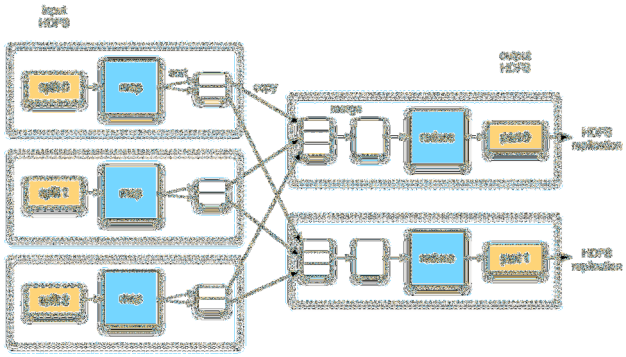

Fig. 1. Shows the working of MapReduce Phases

Hadoop MapReduce is a software framework for easily writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner [3].

II. HADOOP MAPREDUCE IMPLEMENTATION

A. *Input Files*

This is the data file which contains information which is needed to be processed. It is stored in HDFS (hadoop Distributed file system).

B. *Input Format*

It describe about the input-specification for MapReduce job. It breaks the input file into logical splits chunk, then each split chunks assigned to individual mapper.

Several input format are specified with hadoop framework.

i) TextInputFormat
ii) KeyValueInputFormat
iii) SequenceFileInputFormat

TextInputFormat is default input format. This treats each line of the file as a separate record. KeyValueInputFormat also treats each line of the file as a separate record but it breaks the records of each line into key and value pairs. SequenceFileInputFormat reads special binary files which are specific to hadoop.

C. *Input Split*

Input split is group of records which is also called as a chunk which are processed by the single mapper. Each mapper processes single split and split are divided into records. The mapper processed each record as a key-value pair in turn. By processing these input splits in parallel with many distributed system we can improve the performance over the file on a single system.

D. *Record Reader*

RR breaks the data into key-value pairs for input to the Mapper.

E. *Mapper*

The Mapper maps input key-value pairs to a set of intermediate key-value pairs. It transforms input records into intermediate key-value pairs. Using Map() method one can write their own implementation of map task to generate intermediate key-value pair. Map() collects four parameters

i. key
ii. Value
iii. Object of OutputCollector which will forward the key-value pair to the reduce phase.
iv. Object or Reporter which gives the information of the current task.

F. *Input Split*

After completion of first map task, the record reader keep on getting data from the file and giving to map() for generation of intermediate key-value pair. In parallel to this, exchanging of intermediate output from one map task to another where they are required by the reducer. Shuffling is the processes of transferring of data from mapper to reducer. Partition class finds which partition a given key value pair will go.

G. Input Split

Hadoop framework calls Reduce() methods for each unique key in the sorted order. The reduce work is to iterate through the key and produce zero or more outputs. Thus the produced output is stored in HDFS in various parts as collected by various reducers.

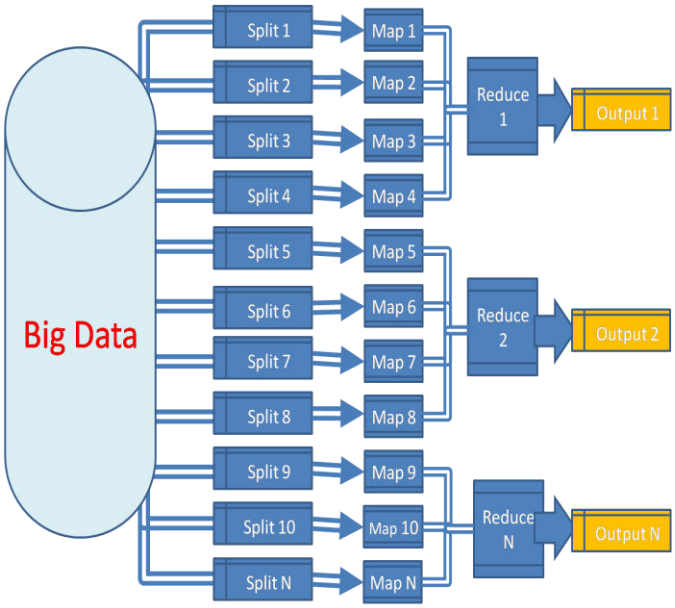

Fig.2. shows MapReduce job work flow

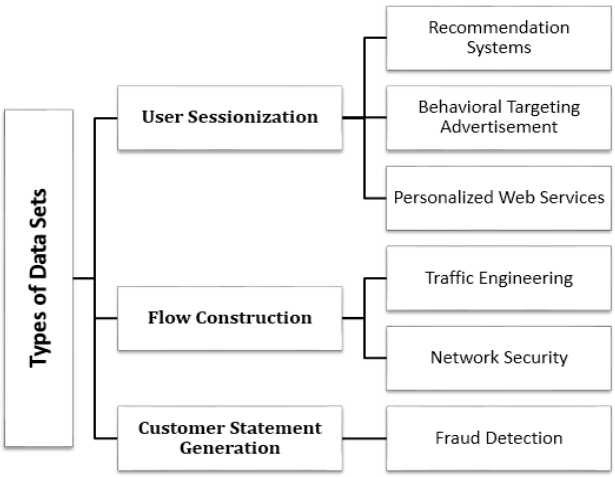

Fig. 3. Shows the different datatpes and its application

H. Event Log File Types

There are several event files types some of them are described below.

a. **HTTP Event File**: Http event log file is the collection of event of click stream data of user clicked events.

b. **FPT Event File**: It contains the FTP access information of each and every users on the server

c. **Network Event File:** It contains the several attributes of the network related parameters such as: Packet loss, packet transmitted, acknowledgement etc.

III. LITERATURE SURVEY

One of the most common datasets exploited by many corporations to conduct business intelligence analysis is event log files.

Xiaokui Shu [4] present a lightweight Distributed and parallel security log analysis framework that allows organizations to analyze a massive number of system, network, and transaction logs efficiently and scalably. Madhury Mohandas [5], builds a failure monitoring system from the scratch, by parsing and analyzing the Hadoop log files generated in the cluster. The monitoring system gives all relevant details related to the application, and points out the specific reason for failure, that is, whether an application failure or a network failure (these are the most common failures in the cluster).

Amrit Pal [6], presents experimental work done on Hadoop by applying a number of files as input to the system and then analyzing the performance of the Hadoop system. SayaJee

Narkhede [7], applied Hadoop MapReduce programming model for analyzing web log files so that we could get hit count of specific web application. This system uses Hadoop file system to store log file and results are evaluated using Map and Reduce function. Experimental results show hit count for each field in log file. Also due to MapReduce runtime parallelization response time is reduced.

Hemant Hingave [8], propose a log analyzer with the combination of Hadoop and Map-Reduce paradigm. The joint of Hadoop and MapReduce programming tools makes it possible to provide batch analysis in minimum response time and in memory computing capacity in order to process log in a high available, efficient and stable way.

Milind Bhandare [9], proposes a design of generic log analyzer using hadoop map-reduce framework. This generic log analyzer can analyze different kinds of log files such as- Email logs, Web logs; Firewall logs Server logs, Call data logs.

TABLE II. Observations on Literature Review

S.No	Reference No.	Technique Used / Mode of Operation	Data Source	Approach to Analyze Log File	Strength	Limitations
1	[4]	Hadoop / MapReduce, Distributed Mode	Security Log Data, Pure HTTP Request Data	Three Level Hash map	Implemented with the minimal set of component	Complex Log files are not taken into consideration
2	[5]	Hadoop / MapReduce Standalone Mode	DataNode log, TaskTracker log, NameNode log of Hadoop Framework	Uses Hadoop Vaidya which is a performance diagnostic tool for map/reduce jobs	Able to detect Network and Application failure effectively	Does not provide any visualization for the log files generated
3	[9]	Hadoop/ MapReduce Standalone Mode	Email logs, Web logs, Firewall logs Server logs	Uses Traditional MapReduce Algorithm	Effectively analyzes the log files	Runs in standalone mode
4	[6]	Hadoop/ MapReduce Standalone Mode	Lyrics of the English songs	Implements Word count application to show that the bytes written do not increase in the same proportion as compared to the amount of files increase.	Effectively analyzes the behavior of the MapReduce task with a number of files	Did not take into account the number of nodes which has great effect on analysis.
5	[7]	Hadoop/ MapReduce, Distributed Mode	HTTP log file	Uses Traditional Map Reduce (word count algorithm) to find Hit Count	Useful for HTTP log files	Small dataset is used
6	[8]	Hadoop/ MapReduce, Standalone Mode	NASA Web Log File	Uses Traditional MapReduce (word count) to find Page View Activity	Effectively compared the time taken by Hadoop Framework and RDBMS	Runs in standalone mode

IV. CONCLUSION

One of the most common data sets exploited by many companies to conduct business intelligence analysis is web log files. Often the records in the log files are temporally ordered and need to be grouped by certain key with order preserved to effectively analyze the files. Log Analysis for any website becomes more important when it comes to the marketing and improving the productivity of any company. This paper introduces various log processing approaches which helps to analyze the log files. Further, analysis of log processing techniques given in Table 2 shows that there are limitations in existing approaches which necessitates further research in the area of log processing using MapReduce Model.

REFERENCES

[1] Tom White: “Hadoop: The Definitive Guide (3rd Ed.)”, O’Reilly Media

[2] https://hadoop.apache.org/

[3] https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html

[4] Xiaokui Shu, John Smiy, Danfeng (Daphne) Yao, and Heshan Lin, "Massive Distributed and Parallel Log Analysis For Organizational Security", Globecom 2013 Workshop - First International Workshop on Security and Privacy in Big Data pages 194-199

[5] Madhury Mohandas, Dhanya P M, "An Approach for Log Analysis Based Failure Monitoring in Hadoop Cluster" 2013 International Conference on Green Computing, Communication and Conservation of Energy (ICGCE) pages 861-867

[6] Amrit Pal, Kunal Jain,Pinki Agrawal,Sanjay Agrawal, "A Performance Analysis of MapReduce Task with Large Number of Files Dataset in Big Data Using Hadoop", 2014 Fourth International Conference on Communication Systems and Network Technologies pages 587-591

[7] SayaJee Narkhede, Trupti Baraskar, Debajyoti Mukhopadhyay, "Analyzing Web Application Log Files to Find Hit Count Through the Utilization of Hadoop MapReduce in Cloud Computing", Environment 2014 IEEE

[8] Hemant Hingave, Prof. Rasika Ingle, "An approach for MapReduce based Log analysis using Hadoop", IEEE Sponsored 2nd International Conference on Electronics and Communication System(ICECS ‘2015) pages 1264-1268

[9] Milind Bhandare, Prof. Kuntal Barua, Vikas Nagare, Dynaneshwar Ekhande, Rahul Pawar, "Generic Log Analyzer Using Hadoop Mapreduce Framework", International Journal of Emerging Technology and Advanced Engineering pages 603-607