International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 11 | Nov 2020 www.irjet.net p-ISSN: 2395-0072

Server Log Prediction using Apache Spark

Kameshwari Soundararajan[1], Rahul R[2], Pradip Kumar R[2]

[1,2]*Student, Department of Computer Science Engineering, Sri Ramakrishna Engineering College, Coimbatore, India.*

Abstract - *One of the most popular and effective ways of predicting server log error is log analytics. Almost every Organization (small, big) have multiple systems and infrastructure running whole day, to effectively keep their business running and it is important for the organizations to know if their infrastructure is performing to its maximum potential. Typically, server logs are a very common data source in enterprises and often contain a gold mine of actionable insights and information. Log data is generated from various sources in an enterprise, such as the web, client and compute servers, applications, user-generated content, flat files, etc. These involve analyzing system and application logs and maybe even apply predictive analytics on log data. The amount of log data engendered depends on the type of organizational infrastructure and applications running on it. Powered by big data, better and distributed computing, big data processing and open-source analytics frameworks like Spark, we can perform scalable log analytics on potentially millions and billions of log messages daily Spark allows you to dump and store your logs in files on disk cheaply, while still providing rich APIs to perform data analysis at scale.*

1. INTRODUCTION

The proposed system comprises of log analytics for predicting errors in the log data of the server. While there are a lot excellent open-source frameworks and tools out there for log analytics including elastic search, the system is proposed to showcase how Spark can be leveraged for analysing logs at scale. Apache Spark is an intense and best open-source framework for performing wrangling, analysing and modelling on structured and unstructured data. Spark allows us to store logs in files on disk cheaply, while still providing rich APIs to perform data analysis at scale.

For analysing, two datasets are taken which contain two months' worth of all HTTP requests to the NASA Kennedy Space Center WWW server in Florida. Data wrangling, Data parsing and extraction with regular expression are carried out as the process of analysing the log data. Firstly, the NASA log data set is loaded and viewed. This allows one to see the schema of the log data which actually looks like text data and this data gets inspected. Spark Data Frames are used to convert a data frame into an RDD. Data wrangling is used to clean and parse our log dataset to really extract structured attributes with meaningful information from each log message. Data parsing is done to parse our semi-structured log data into individual columns. Some special functions are used to perform the parsing. These functions match a column with a regular expression of one or more groups and allow one to extract the matched groups. For each field, one wants to extract a regular expression is used.

1.2 Related work

The existing system uses two approaches for log clustering. The supervised learning method requires users first to manually label a set of log categories and use classifiers such as Naive Bayes to perform text categorization. The log analytics is done with the help of elastic search. The other approach uses unsupervised learning for clustering, such as hierarchical partitioning process proposed in, and multi-pass data summarization method. Once the structure of the cluster is obtained from the logs, string matching is done to extract the common patterns from multiple logs within the cluster.

1.3 Work flow

The design flow of the project commences with the collection and compilation of datasets from two types of machinery, say, a normal and an abnormal machine. The datasets are verified for its noisy data and missing values. The datasets are then pre-processed. Classification of vibrations has been made when it has to be verified for noisy data and non-continuous human made vibration. The datasets are analysed corresponding to the threshold values. The regression technique has been carried out on multiple attributes of the dataset. The prediction of health status of a machine has been made.

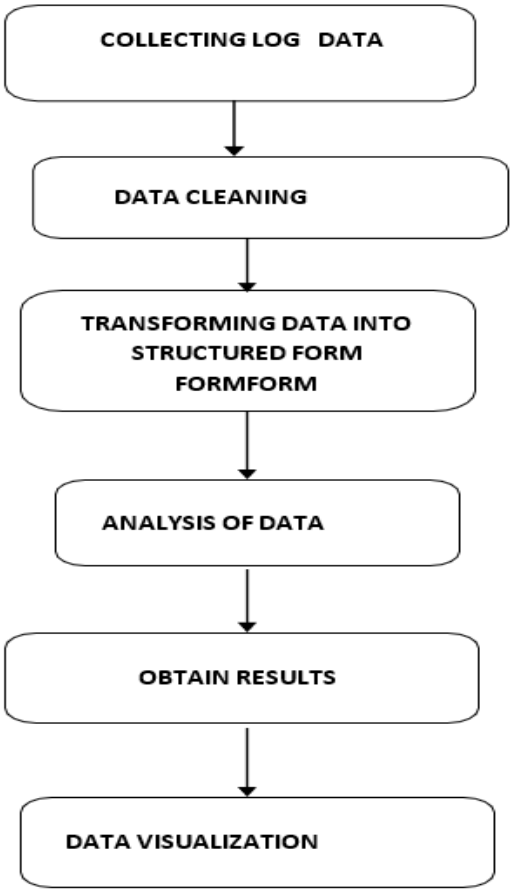

Fig 1 Flow Diagram of server log prediction.

2. COLLECTION OF DATA SET

Server log data set is a common data, coming from many sources such as the web, client and compute servers, applications, user-generated content, flat files. They can be used for monitoring servers, improving business and much more. The amount of log data is usually massive, depending on the type of the organizational infrastructure and the applications running on it.

```
+-----------------------------------------------------------------------------------------------------------------------+
|value                                                                                                                  |
+-----------------------------------------------------------------------------------------------------------------------+
|199.72.81.55 - - [01/Jul/1995:00:00:01 -0400] "GET /history/apollo/ HTTP/1.0" 200 6245                                 |
|unicomp6.unicomp.net - - [01/Jul/1995:00:00:06 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3985                      |
|199.120.110.21 - - [01/Jul/1995:00:00:09 -0400] "GET /shuttle/missions/sts-73/mission-sts-73.html HTTP/1.0" 200 4085   |
|burger.letters.com - - [01/Jul/1995:00:00:11 -0400] "GET /shuttle/countdown/liftoff.html HTTP/1.0" 304 0               |
|199.120.110.21 - - [01/Jul/1995:00:00:11 -0400] "GET /shuttle/missions/sts-73/sts-73-patch-small.gif HTTP/1.0" 200 4179|
|burger.letters.com - - [01/Jul/1995:00:00:12 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 304 0                   |
|burger.letters.com - - [01/Jul/1995:00:00:12 -0400] "GET /shuttle/countdown/video/livevideo.gif HTTP/1.0" 200 0       |
|205.212.115.106 - - [01/Jul/1995:00:00:12 -0400] "GET /shuttle/countdown/countdown.html HTTP/1.0" 200 3985            |
|d104.aa.net - - [01/Jul/1995:00:00:13 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3985                               |
|129.94.144.152 - - [01/Jul/1995:00:00:13 -0400] "GET / HTTP/1.0" 200 7074                                              |
+-----------------------------------------------------------------------------------------------------------------------+
only showing top 10 rows
```

Fig 2 Server log sample Dataset

3. DATA WRANGLING

Data wrangling is the process of cleaning, structuring and making raw data into the desired format for better decision making in less time. This involves transforming and mapping data from one "raw" data form into another form with the intent of making data more appropriate and valuable for a variety of purposes such as analytics.

Data parsing is the process of breaking the data blocks into smaller chunks by following a set of rules so that it can be more easily interpreted and managed. Here, the unstructured log data is parsed into individual columns by using built-in regexp extract() function. This function matches a column with a regular expression with one or more groups and allows us to extract one of the matched groups. By using this regexp extract() function, the hostnames, timestamps, HTTP request methods, HTTP response content size, URI's and protocols are extracted. The extracted expressions are put up together to build data frame with all the log attributes neatly extracted in their separate columns.

```
['199.72.81.55',
 'unicomp6.unicomp.net',
 '199.120.110.21',
 'burger.letters.com',
 ...,
 ...,
 'unicomp6.unicomp.net',
 'd104.aa.net',
 'd104.aa.net']
```

FIG 3.1 Extracting host names

```
['01/Jul/1995:00:00:01 -0400',
 '01/Jul/1995:00:00:06 -0400',
 '01/Jul/1995:00:00:09 -0400',
  ...,
  ...,
 '01/Jul/1995:00:00:14 -0400',
 '01/Jul/1995:00:00:15 -0400',
 '01/Jul/1995:00:00:15 -0400']
```

FIG 3.2 Extracting timestamps

```
[('GET', '/history/apollo/', 'HTTP/1.0'),
 ('GET', '/shuttle/countdown/', 'HTTP/1.0'),
  ...,
  ...,
 ('GET', '/shuttle/countdown/count.gif', 'HTTP/1.0'),
 ('GET', '/images/NASA-logosmall.gif', 'HTTP/1.0')]
```

```
['200', '200', '200', '304', ..., '200', '200']
```

FIG 3.3 Extracting http request method, URI's and protocol

```
['6245', '3985', '4085', '0', ..., '1204', '40310', '786']
```

FIG 3.4 Extracting status code

host	timestamp	method	endpoint	protocol	status	content_size
199.72.81.55	01/Jul/1995:00:00...	GET	/history/apollo/	HTTP/1.0	200	6245
unicomp6.unicomp.net	01/Jul/1995:00:00...	GET	/shuttle/countdown/	HTTP/1.0	200	3985
199.120.110.21	01/Jul/1995:00:00...	GET	/shuttle/missions...	HTTP/1.0	200	4085
burger.letters.com	01/Jul/1995:00:00...	GET	/shuttle/countdow...	HTTP/1.0	304	0
199.120.110.21	01/Jul/1995:00:00...	GET	/shuttle/missions...	HTTP/1.0	200	4179
burger.letters.com	01/Jul/1995:00:00...	GET	/images/NASA-logo...	HTTP/1.0	304	0
burger.letters.com	01/Jul/1995:00:00...	GET	/shuttle/countdow...	HTTP/1.0	200	0
205.212.115.106	01/Jul/1995:00:00...	GET	/shuttle/countdow...	HTTP/1.0	200	3985
d104.aa.net	01/Jul/1995:00:00...	GET	/shuttle/countdown/	HTTP/1.0	200	3985
129.94.144.152	01/Jul/1995:00:00...	GET	/	HTTP/1.0	200	7074

FIG 3.5 Extracting http response content size

4. ANALYSIS OF DATA

Now the data frame that contains the parsed and cleaned log file is analysed. The total number of times each status code has arrived, number of unique daily host, average number of daily requests per host, number of 404 response codes are analysed and determined.

	status	count
0	200	3100524
2	304	266773
1	302	73070
5	404	20899
4	403	225
6	500	65
7	501	41
3	400	15

FIG 4.1 HTTP Status Code Analysis

	day	total_reqs	total_hosts	avg_reqs
0	1	98710	7609	12.972795
1	2	60265	4858	12.405311
2	3	130972	10238	12.792733
3	4	130009	9411	13.814579
4	5	126468	9640	13.119087
...	...	...	...	...
26	27	94503	6846	13.804119
27	28	82617	6090	13.566010
28	29	67988	4825	14.090777
29	30	80641	5265	15.316429
30	31	90125	5913	15.241840

FIG 4.2 Average Number of Daily Requests per Host

```
Total 404 responses: 20899
```

FIG 4.3 Counting 404 Response Codes

```
+----------------------------+-----+
|host                        |count|
+----------------------------+-----+
|hoohoo.ncsa.uiuc.edu        |251  |
|piweba3y.prodigy.com        |157  |
|jbiagioni.npt.nuwc.navy.mil|132  |
|piweba1y.prodigy.com        |114  |
|                            |112  |
|www-d4.proxy.aol.com        |91   |
|piweba4y.prodigy.com        |86   |
|scooter.pa-x.dec.com        |69   |
|www-d1.proxy.aol.com        |64   |
|phaelon.ksc.nasa.gov        |64   |
|dialip-217.den.mmc.com      |62   |
|www-b4.proxy.aol.com        |62   |
|www-b3.proxy.aol.com        |61   |
|www-a2.proxy.aol.com        |60   |
|www-d2.proxy.aol.com        |59   |
|piweba2y.prodigy.com        |59   |
|alyssa.prodigy.com          |56   |
|monarch.eng.buffalo.edu     |56   |
|www-b2.proxy.aol.com        |53   |
|www-c4.proxy.aol.com        |53   |
+----------------------------+-----+
```

FIG 4.4 List of 404 Response Code Hosts

5. VISUALISATION OF ANALYSED DATA

Data visualization refers to a technique that is used to communicate data or inform by encoding it as visual objects in form of graphics. It is easy to predict when the analyzed data is in any visual format.

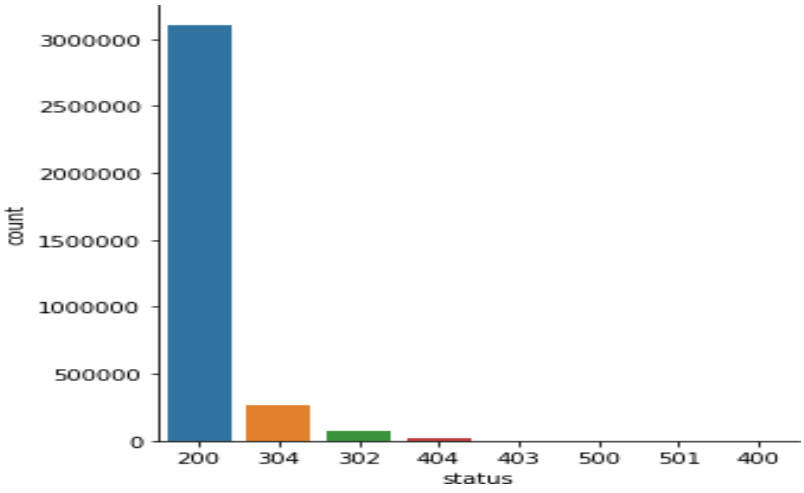

FIG 5.1 HTTP status code occurrences

Several status codes are almost not visible due to the huge skew in the data. So log transform is done to improve the visibility.

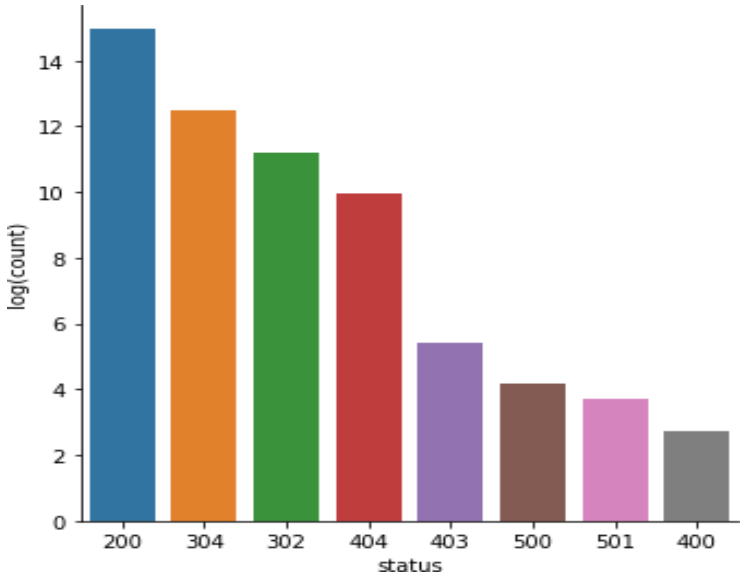

FIG 5.2 HTTP status code occurrences-log transformed

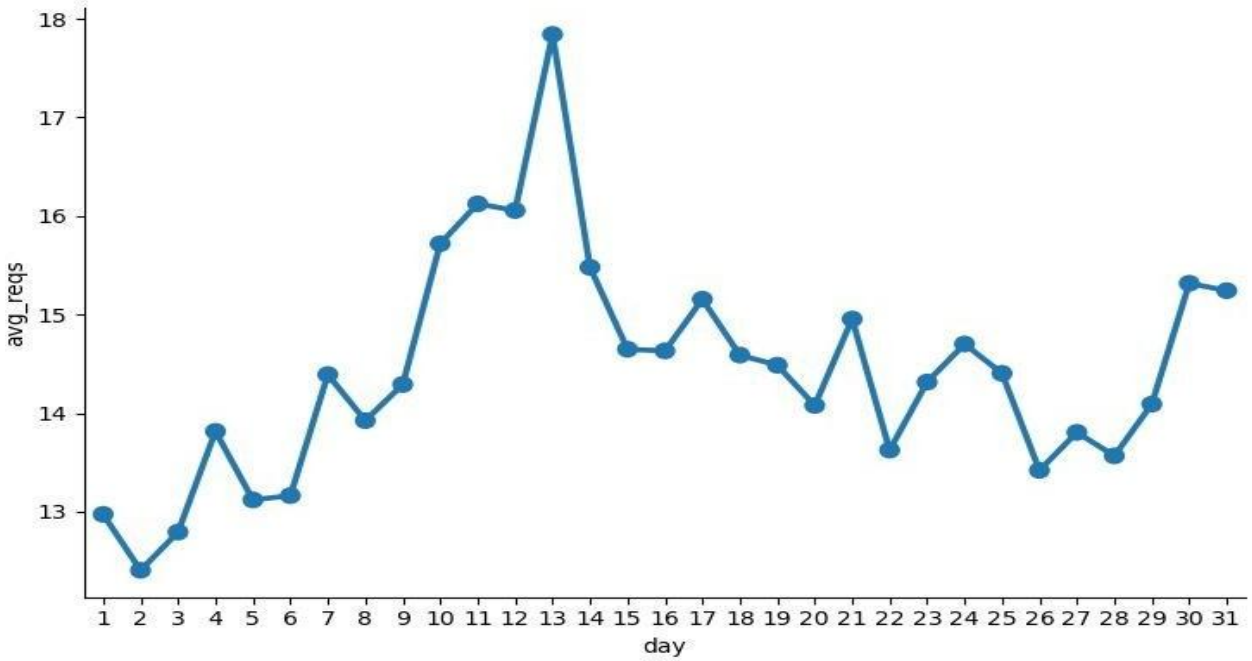

FIG 5.3 Average daily requests per host

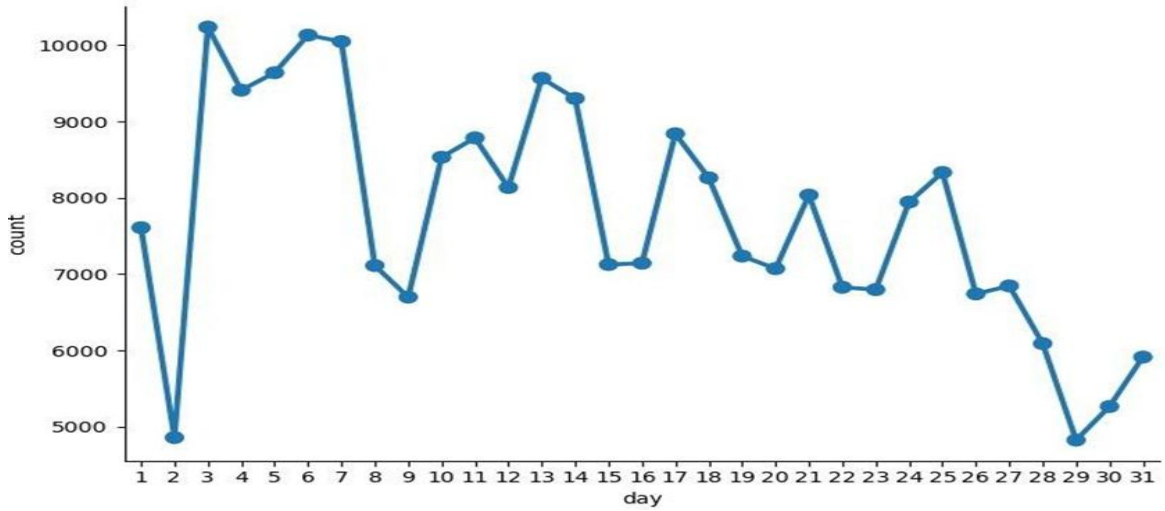

FIG 5.4 Unique hosts per day

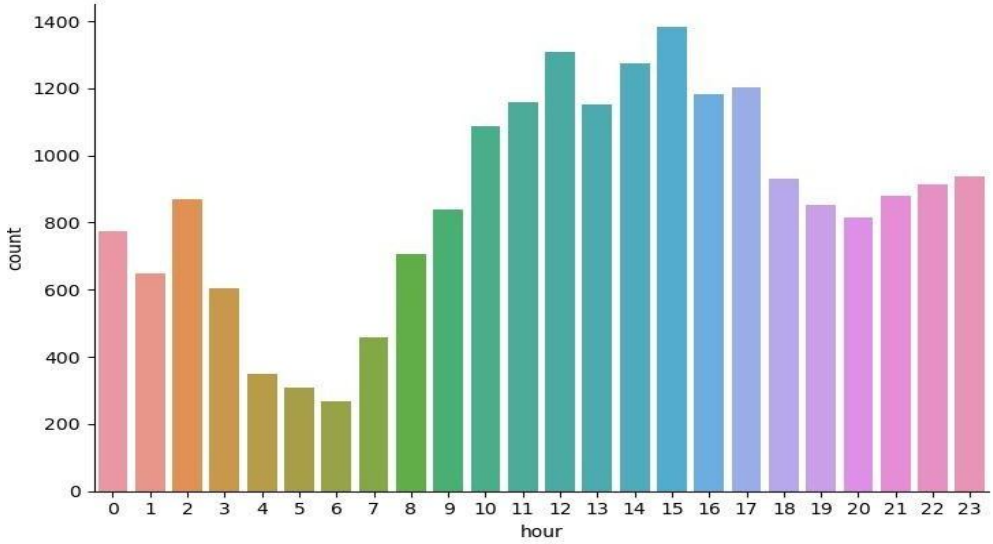

FIG 5.5.1 Total 404 error per hour

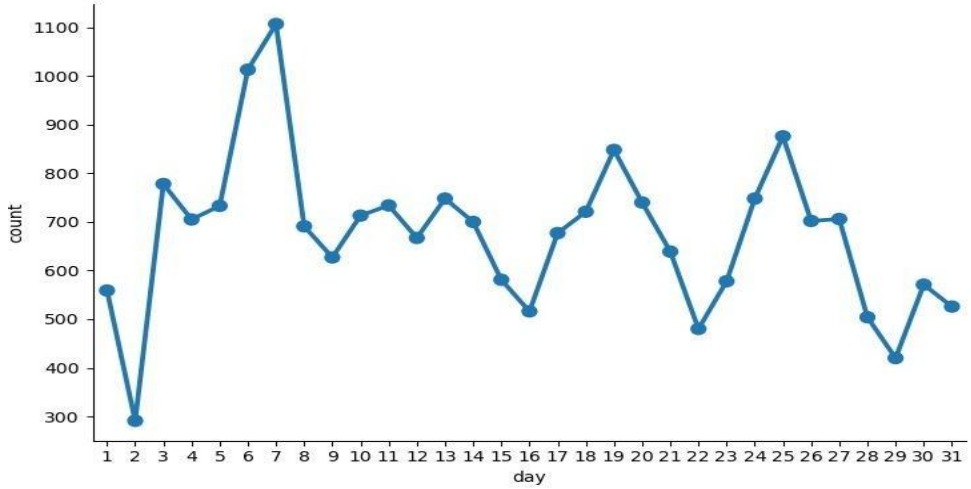

FIG 5.5.2 Total 404 error per hour

6. CONCLUSION

It is common perception today that log files, and in particular error logs, are a fruitful source of information both for analysis after failure and for proactive fault handling which frequently builds on the anticipation of upcoming failures. However, in order to get the machine, access the information contained in logs, the data should be put into shape and valuable pieces of information must to be picked from the vast amount of data. The information obtained from the proposed technique show how many log errors has occurred, number of unique hosts, average number of unique host and more. This information can be used for improving business, customer intelligence and much more. The major challenge is that the logging process itself requires additional management. Controlling the verbose of logging is necessary, especially when potential adversarial behavior, to manage the overhead and facilitate the analysis process. The logging mechanism must not be a network to propagate malicious activity.

REFERENCES

[1] Marcello Cinque, Domenico Cotroneo, and Antonio Pecchia. 2013. Event logs for the analysis of software failures: A rule-based approach. IEEE Transactions on Software Engineering (TSE) (2013), 806–821.

[2] Liang Tang, Tao Li, and Chang-Shing Perng. 2011. LogSig: Generating system events from raw textual logs. In Proc. Conference on Information and Knowledge Management (CIKM). 785– 794.

[3] Ting-Fang Yen, Alina Oprea, Kaan Onarlioglu, Todd Leetham, William Robertson, Ari Juels, and Engin Kirda. 2013. Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks. In Proc. International Conference on Dependable Systems and Networks (ACSAC). 199–208.

[4] Ding Yuan, Haohui Mai, Weiwei Xiong, Lin Tan, Yuanyuan Zhou, and Shankar Pasupathy. 2010. SherLog: error diagnosis by connecting clues from run-time logs. In ACM SIGARCH computer architecture news. ACM, 143–154.

[5] Karthik Nagaraj, Charles Killian, and Jennifer Neville. 2012. Structured comparative analysis of systems logs to diagnose performance problems. In Proc. USENIX Symposium on Networked Systems Design and Implementation (NSDI). 26–26.

[6] Ingram, Albert L. "Using web server logs in evaluating instructional web sites." Journal of educational technology systems 28.2 (1999): 137-157.

[7] Abd Wahab, Mohd Helmy, et al. "Data pre-processing on web server logs for generalized association rules mining algorithm." World Academy of Science, Engineering and Technology 48 (2008): 190-197.

[8] Seibel, John C., Yu Feng, and Robert L. Foster. "Text mining system for web-based business intelligence applied to web site server logs." U.S. Patent No. 7,330,850. 12 Feb. 2008.

[9] Rowbottom, Nicholas, Amir Allam, and Andrew Lymer. "An exploration of the potential for studying the usage of investor relations information through the analysis of Web server logs." International Journal of Accounting Information Systems 6.1 (2005): 31-53.

[10] Dikaiakos, Marios, Athena Stassopoulou, and Loizos Papageorgiou. "Characterizing crawler behavior from web server access logs." International Conference on Electronic Commerce and Web Technologies. Springer, Berlin, Heidelberg, 2003.