A Machine Learning Approach to Improve the Performance of Web Proxy Cache Replacement

S. Nimishan and S. Shriparen

Department of Computer Science, Faculty of Science, University of Jaffna

INTRODUCTION

Web caching is a well-known strategy for improving the performance of Web proxy servers. Proxy servers play a key roles between users and Websites in reducing the response time and saving network bandwidth. As shown in Fig 1. the proxy cache is found in the proxy server, which is located between the client machines and origin server.

Due to cache space limitations, a Web proxy cache replacement policy is required to manage and manipulate the Web proxy cache contents efficiently and effectively.

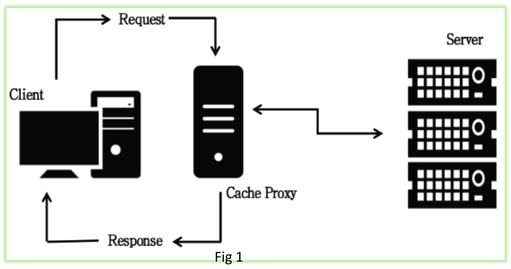


Fig 1

EXPERIMENTAL SETUP

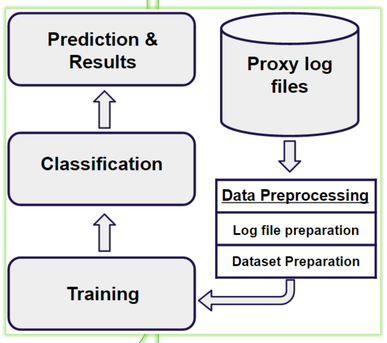
The target attribute is the column used for classification. Then the datasets divided into training and testing datasets. First from training dataset the classification model is built using a machine learner tool. Then by the classification model, target attribute is predicted for the testing data which we expected the target attribute for identify the re-requested web objects, from the same tool. WSO2 ML (version 1.0.0) is used for the purpose of Data mining tasks. for each training dataset, SVM models were trained from the training data by validating 30% of the training data and select the algorithm with target attribute and the training data fraction as 0.7 for SVM.

Regarding Decision Tree training, the settings and default values of parameters as determined in WSO2 ML were changed according to our requirement.

This study considers that each Web object belongs to the positive class if the object is re-requested again. Otherwise, the Web object belongs to the negative class.

METHODOLOGY

Proxy cache miss occurs in a web client communication with the proxy server, if the requested object is not in the proxy cache or not fresh. The proxy cache manager needs an approach to know which objects to keep or which objects to remove that is the unwanted Web objects in order to release enough space for the new coming object. This approach is to use machine learning classifiers to evict unwanted web objects from the early stage before caching.



In the classification of log files, unwanted attributes in the log are removed and a new attribute as target attribute is added. That will have a value of '1' if the domain name re-requested again within the 30 minutes Sliding Window Length (SWL), otherwise it will be assigned a value of '0'. Then the trace set is ready to be applied under the Machine Learner approaches using the WSO2 ML and therefore divide the dataset to take 70% of data as training data and 30% as test data.

Data pre-processing requires two steps: log file preparation and training dataset preparation. In the log file preparation, irrelevant or invalid requests are removed from log files such as un-cacheable web requests.

PERFORMANCE EVALUATION

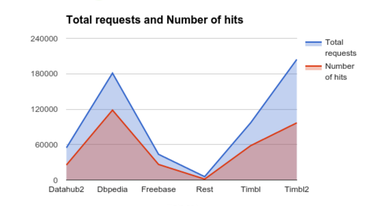
We can improve the accuracies by removing the unwanted attributes such as size of the web object requested from the dataset. Because there are few fields that are not used for classification.

The hit ratio (HR) is the most widely used metrics for evaluating the performance of Web caching. HR is defined as follows:

$$\text{Hit Ratio} = \frac{\text{Number of Hits}}{\text{Cacheable requests}} * 100$$

HIT RATIO FOR SVM MODEL

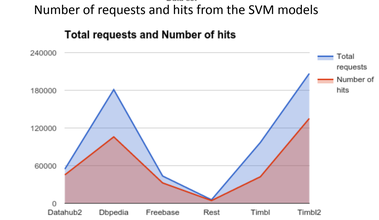
Data set	Total requests	Number of hits	Hit ratio (%)
Datahub2	54557	25470	46.68
Dbpedia	181114	118418	65.38
Freebase	43507	26359	60.58
Rest	5685	1519	26.71
Timbl	97039	58243	60.02
Timbl2	204288	96822	47.39



Number of requests and hits from the SVM models

HIT RATIO FOR DECISION TREE MODEL

Data set	Total requests	Number of hits	Hit ratio (%)
Datahub2	54557	45357	83.13
Dbpedia	181114	105883	58.46
Freebase	43507	32527	74.76
Rest	5685	4428	77.88
Timbl	97039	42390	43.68
Timbl2	206708	135149	66.15



Number of requests and hits for the Decision tree mode

DATA COLLECTION AND PRE-PROCESSING

The data have been downloaded from Billion Triples Challenge 2012 Dataset as raw log entry files which contains 4,751,262 entries. In the dataset and there are no HITs or TCP HITs in the http result code column. Each line in the proxy logs file represents access proxy log entry, which contains the following ten fields: *Timestamp, Elapsed Time, Client IP address, Log tag and HTTP code, Size, Request Method, URL, User Identification, Hierarchy Data and Hostname,* and *Content type*.

```
1422310703.703 532 192.168.1.250 TCP_MISS/200 2562 GET http://www.bbc.com/news/10284448/ticker.sjson? - HIER_DIRECT/212.58.246.90 text/javascript
1422310705.090 151 10.10.20.38 TCP_MISS/200 956 HEAD http://www.google.lk/? - HIER_DIRECT/74.125.130.94 text/html
1422310737.950 149 10.10.20.38 TCP_MISS/200 956 HEAD http://www.google.lk/? - HIER_DIRECT/74.125.130.94 text/html
1422311429.391 600 10.10.20.38 TCP_MISS/200 451 GET http://logs.newstatsclientcloud.com/monetization.gif? - HIER_DIRECT/69.16.175.42 image/gif
```

Dataset	Requests	Cacheable Requests
Datahub	398547	181850
DBpedia	1382090	537038
Freebase	333956	145010
Rest	71972	18942
Timbl 1	889591	323451
Timbl 2	1675106	680952

In the pre-processing step unsuccessful HTTP requests are eliminated and have included the successful HTTP requests that had a states code of 200. Also out of the totally collected 4,751,262 entries only 39.73% were the cacheable requests.

CONCLUSION

Here the machine learning approaches are presented to improve web proxy caching. The two machine learning approaches namely SVM and decision tree, to predict and store popular in a proxy cache.

The results revealed that improvements in cache hit ratios by using machine learning approaches in which the raw data does not contain any hits.

The average hit ratios achieved by SVM and Decision tree are 51.12% and 67.34% respectively. Therefore we can ensure that the web proxy caching works more efficiently by using this approach.

REFERENCES

[1] Jain, Behera, Mandal, Mohapatra, Computational Intelligence in Data Mining, Vol.2, International Conference on CIDM, 2014

[2] S. Romano and H. ElAarag, A neural network proxy cache replacement strategy and its implementation in the Squid proxy server, Neural Computing & Applications, Vol l. 20, No. 1, pp. 59-78, 2011.

[3] W. Ali, S.M. Shamsuddin, and A.S. Ismail, A Survey of Web Caching and Prefetching, Int. J. Advance. Soft Comput. Appl., Vol. 3, No. 1, pp. 18, 2011

[4] W. Ali S. Sulaiman, and N. Ahmad Performance Improvement of Least Recently Used Policy in Web Proxy Cache Replacement Using Supervised Machine Learning Int. J. Advance. Soft Comput. Appl., Vol. 6, No.1, 2014

[5] Squid: Optimizing Web Delivery Available: http://www.squid-cache.org

[6] Performance Analysis of Web Caching Through Cache Replacement Based on User Behavior, Available online at: www.ijarcsse.com

[7] Billion Triples Challenge 2012 Dataset. Available: https://km.aifb.kit.edu/projects/btc-2012/000-CONTENTS