Carleton University

Frequency-Based Log File Analyser

COMP 4905 - Honours Project Proposal
Thursday, May 17th 2018

Daniel Fitzhenry - 100949529 - daniel.fitzhenry@carleton.ca

Submitted to: Prof. Anil Somayaji COMP 4905
School of Computer Science Carleton University

Table of Contents

Background

In order to effectively manage a web server, it is necessary to get feedback about the activity and performance of the server as well as any problems that may be occuring. For this, most modern web servers perform system logging: automatically create and maintain a history of actions performed known as a log, recording events that were executed in order of occurrence. This is not limited to web servers, as virtually all software applications and systems produce log files, usually application specific. Essentially for the internet and web servers, activities of users are traced in different types of log files hence, prove to be extremely useful in understanding user behavior, improving server performance, improving cache replacement policy, intrusion detection, etc. Log analysis is therefore a much-needed task by administrators to extract the valuable information and with the management of real-time data, can use the logs for making decisions.

The relevance of log data can differ from one person to another. It may be possible that the particular log data can be beneficial for one user but irrelevant for another. Therefore, the usage of log data can be lost inside the large cluster (or grand scheme of things). To make matters worse, traffic of computer networks is notorious for being fundamentally hard to understand. “Enterprises add numerous database, file service, network management, and proprietary protocols as part of custom applications. Even the smallest home networks today connect to thousands of remote hosts in order to access email, instant messaging, voice-over-IP, peer-to-peer file sharing, streaming media, and social media” [2]. Traditional methods to analyze log files and determine valid feedback are quickly becoming infeasible with the ever increasing advances in network protocols and traffic complexity. Information overload leads to the contradiction between vast information and limited need; the presentation of log data directly affects their ability to correlate with the other data. This forces professionals to seek quicker, more accurate ways of discerning relevant data within an overabundance of information, as ignoring data can consequently leave a significant gap of relevant information unutilized.

There exist many techniques because of which data is filtered and instead of getting access to every data available, we get relevant results. Recommender Systems are a type of web intelligence technique that can provide two types of classification: Personalized and Non-Personalized recommendations. Personalized recommendation takes into consideration previous history for rating and predicting items whereas Non-personalized recommendation systems recommend what is popular and relevant to all. One of if not the most important techniques in the Recommender System is information

filtering however both categories are associated with varying degrees of overhead (preprocessing) and/or expensive computational requirements.

Proposed Work

Web log files record the whole process of interaction between the website and visitors, containing valuable behavior characteristics and visitor access patterns that can benefit performance and security. However, with the constantly evolving network protocols and traffic complexity, there exists a tradeoff between precision and efficiency regarding methods of classifying web traffic. Research done on standard network monitoring tools demonstrates that traditional regular expression-encoded signature-based or header-content based analysers are not sufficient in their ability to learn about network traffic. The key disadvantage lies in the fact that those techniques do not identify and characterize structural patterns in a general way, such that activity matching those patterns can be efficiently identified.

The main benefits of any clustering algorithm is the ability to blindly (without prior knowledge) aggregate data into semantically meaningful clusters. Using a technique similar to that used by A. Hijazi in Approximate Divisive HIerarchical Clustering (ADHIC) for network packet analysis, we propose structurally representative n-grams can be found efficiently using hierarchical clustering of log activity. This can form a “fingerprint” of network protocols that may be used to identify traffic anomalies in a fashion similar to that of hand-crafted regular expression signatures.

Code and Evaluation Plan

An apache server log analyser based on n-gram frequency statistics rather than regular expression signatures will be explored and constructed throughout the duration of the project. This application will be an open-sourced web-browser based log management and analysis tool built using Node.js (javascript). It will serve as a lexical analyser (binary-tree constructor) to be used for research purposes and further develop understanding of server log files and the associated network activity which they imply. Javascript was selected as the base language for its relation to core technologies of the World Wide Web, and the recently introduced Tensorflow.js library for training and deploying machine learning models.

Timeline

May 6th - May 19th:

1. Meet with Mr. Anil Somayaji to discuss topic
2. Decide on feasible implementation
3. Start coding basic functionality (parse log files)
4. Finish and **submit project proposal** (17th)

May 20th - June 2nd:

1. Learn more about apache log format
2. Meet with Mr. Anil Somayaji to discuss implementation details
3. Learn Tensorflow.js
4. Construct example ML models

June 3rd - June 16th:

1. Start the user interface
2. Research n-gram analysis
3. Implement prototype n-gram analyser
4. Report bottlenecks and functionality

June 17th - June 30th:

1. Prepare and **submit mid-term report** (28th)
2. Discuss further functionality with Mr. Anil Somayaji
3. Fix (or continue) coding aspect
4. Tweak

July 1st - July 14th:

1. Start (if not already begun) final draft
2. Meet with Mr. Anil Somayaji for new data to test

July 15th - July 28th:

1. Try to extend functionality
2. Continue on final draft

July 29th - August 11th:

1. Polish code and application
2. Finish and submit **draft of final report** (2nd)

August 12th - August 16th:

1. Finalize code
2. **Submit final report** (16th)

References

1. R. Vaarandi, "A data clustering algorithm for mining patterns from event logs," *Proceedings of the 3rd IEEE Workshop on IP Operations & Management (IPOM 2003) (IEEE Cat. No.03EX764)*, 2003, pp. 119-126. doi: 10.1109/IPOM.2003.1251233, URL: http://ieeexplore.ieee.org.proxy.library.carleton.ca/stamp/stamp.jsp?tp=&arnumber=1251233&isnumber=28012
2. A. Hijazi, “Network Traffic Characterization Using (p, n)-grams Packet Representation”, *The Faculty of Graduate Studies and Research in partial fulfilment of the requirements for the degree of DOCTOR OF PHILOSOPHY in Computer Science,* Carleton University 2011. URL: http://people.scs.carleton.ca/~soma/pubs/students/abdulrahman-hijazi-phd.pdf
3. J. Pelemans, K. Demuynck, H. Van hamme and P. Wambacq, "Improving n-gram probability estimates by compound-head clustering," *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, South Brisbane, QLD, 2015, pp. 5221-5225. doi: 10.1109/ICASSP.2015.7178967URL: http://ieeexplore.ieee.org.proxy.library.carleton.ca/stamp/stamp.jsp?tp=&arnumber=7178967&isnumber=7177909
4. D. Jacobs, S. Sarasvady and P. Pichappan, "Transaction clustering of web log data files using genetic algorithm," *2007 2nd International Conference on Digital Information Management*, Lyon, France, 2007, pp. 665-669. doi: 10.1109/ICDIM.2007.4444300 URL: http://ieeexplore.ieee.org.proxy.library.carleton.ca/stamp/stamp.jsp?tp=&arnumber=4444300&isnumber=4444274
5. Xie, Yunjuan & Phoha, Vir. (2001). Web user clustering from access log using belief function. 202-208. 10.1145/500737.500768. URL: https://www.researchgate.net/publication/220916862_Web_user_clustering_from_access_log_using_belief_function