USING SEMI-SUPERVISED CLASSIFIER TO FORECAST EXTREME CPU UTILIZATION

Nitin Khosla[1] and Dharmendra Sharma[2]

[1]Assistant Director- Performance Engineering, ICTCAPM, Dept. of Home Affairs, Canberra, Australia
[2]Professor – Computer Science, University of Canberra, Australia

ABSTRACT

A semi-supervised classifier is used in this paper is to investigate a model for forecasting unpredictable load on the IT systems and to predict extreme CPU utilization in a complex enterprise environment with large number of applications running concurrently. This proposed model forecasts the likelihood of a scenario where extreme load of web traffic impacts the IT systems and this model predicts the CPU utilization under extreme stress conditions. The enterprise IT environment consists of a large number of applications running in a real time system. Load features are extracted while analysing an envelope of the patterns of work-load traffic which are hidden in the transactional data of these applications. This method simulates and generates synthetic workload demand patterns, run use-case high priority scenarios in a test environment and use our model to predict the excessive CPU utilization under peak load conditions for validation. Expectation Maximization classifier with forced-learning, attempts to extract and analyse the parameters that can maximize the chances of the model after subsiding the unknown labels. As a result of this model, likelihood of an excessive CPU utilization can be predicted in short duration as compared to few days in a complex enterprise environment. Workload demand prediction and profiling has enormous potential in optimizing usages of IT resources with minimal risk.

KEYWORDS

Semi-Supervised Learning, Performance Engineering, Load And Stress Testing, Machine Learning.

1. INTRODUCTION

The current cloud based environment is very dynamic in which the web traffic or number of hits to some applications increases exponentially in a short span of time (burst in traffic) and it drastically slows down the enterprise application system. At many occasions the IT system crashes because it cannot sustain the excessive load under stress conditions. In enterprise applications environment in big departments, some crucial applications providing services to the public, e.g. benefits payments, custom clearances at airports, etc., halt suddenly. It is observed that at many occasions the systems crash randomly due to unpredictable load because of excessive web traffic for short period. This results in loss of efficiency and productivity of service providers. Many at times, the system crashes without any alerts and practically it is not feasible to take any remedial actions e.g. load balancing, etc. The high transaction-rate (excessive number of hits / second) at some moment of time for a very short duration can drag the IT applications or the computer systems to be very sluggish because the system becomes irresponsive and unable to process large number of transactions simultaneously at different servers.

As we know, our reliance on cloud internet computing is increasing every day and it has become unavoidable. Therefore, it has become very important for big enterprises to keep the key applications running 24/7 to an acceptable efficiency during the whole year. Marinating the key applications running at a high efficiency level in the big enterprise system and developing new functionality at the same time is a constant challenge between functionality and resource management [8]. T many instances it is observed that the system has arrived to a situation when practically very negligible memory is available for the critical applications to run in an enterprise set-up and this situation can lead to a system crash. The scenario becomes even more complex when transactions are generated from wide area distributed networks where the network traffic, latency and bandwidth are key factors impacting the performance and behaviour of IT applications.

The main aim of this research paper is to develop and implement a practical approach to forecast unpredictable burst in traffic by using semi-supervised neural nets classifier. To perform this we have analysed the work-load patterns hidden in of the key transactions over a year and observed CPU utilization under stress conditions (high volume of web traffic) using data mining techniques.

2. RESEARCH PROBLEM

In this research, we have studied the load profiles of the last year to identify patterns at different time periods. These patterns were analysed and used to develop test scenarios in the test environment. We also analysed the big transactional data and extracted load patterns from the raw transactions with the help of implementing profile points [2]. This enabled us to identify issues related to load estimation in testing environment as well the real world (production) environments. We then developed a performance predictive model to forecast the CPU performance in enterprise IT infrastructure under extreme stress conditions.

2.1.Performance Issues

Computer applications are developed and based upon business specifications and these specifications are primarily depend upon the user requirements [1]. We have noticed in our department that there are some critical limitations in determining performance of applications, in the current practices such as –

- Not Reliable: Predicting system behaviour is not reliable e.g. response time, performance, specially under a short burst traffic (hits) situation
- Not Robust: Lack of a robust approach due to the volatile and unpredictable web traffic
- Using Risk Based Approach: Performance testing (load and stress testing) is mainly done using the key or crucial transactions which are considered as a high-risk to the departments. Testing each and every scenario is in an enterprise applications environment is extremely time consuming and costly [2]. So, load tests are generally performed on -
    - Key transactions which have critical impact on systems or services
    - Critical functions which could impact people or important services

3. FEATURE EXTRACTION AND DATA ANALYSIS IN LARGE ENTERPRISE ENVIRONMENT

Big public departments or corporates comprises of different types of system architectures (latest and legacy) e.g. mobile applications, cloud computing, etc. [1]. To cater most of the real work scenarios, we have performed our experiments in a test environment which simulates a large and complex environment having more than 350 servers and large number of which were distributed across multiple sites (countries). This test environment (called as "pre-production" environment) is a subset of the whole enterprise set-up with all applications integrated as per the specifications and represent the most recent releases but with limited data set. This test environment also represents a system simulating all the applications working in more than 52 overseas posts across the world.

Raw transnational data was captured from data logs / files which were created at periodic intervals. To perform the data collection, profile points are configured in the IT system architecture at different layers which collects the data continuously at pre-defined time intervals [1]. Different type of transactional data was captured e.g. % CPU utilization, transaction response times, bandwidth, memory usages, etc. and used for analysis and debugging purposes.

Load and stress experiments for validation were done in the IT test environment (called as pre-production environment) which represents the production environment. This test environment emulates the real-world type of scenarios or behavior. Profile points are used to monitor the transactions, responses times and other key parameters during the full path both ways (server to client and client to server). The analysis of data, recognition of patterns are used and extremely important for optimization [1]. This also helps to continuously improve the models to predict reaching critical load while meeting the dynamic needs and variability of the dynamic load patterns [12].

4. IDENTIFYING WORKLOAD PATTERNS

We have developed a trace-based approach to identify patterns of the CPU utilization of servers over a period. For this we have captured transactions for over a year and collected relevant data for the last one year. The profile points were configured at different threads and nodes of the applications path flows and capture the data at regular pre-defined intervals. Then we have studied the work load patterns of different transactions and their respective behaviour.

Workload patterns under stress conditions were quite typical and different from normal behaviour of a CPU [1]. Signs of high CPU utilization can be predicted while simulating the virtual traffic in test environment. The test environment also executes large number of applications simultaneously as like real word scenario.

Assumption: it is assumed that CPU utilization follows a cyclic behaviour for some types of transactions. These patterns can be represented by a time series consisting of a pattern and/or a cyclical component [1],[2].

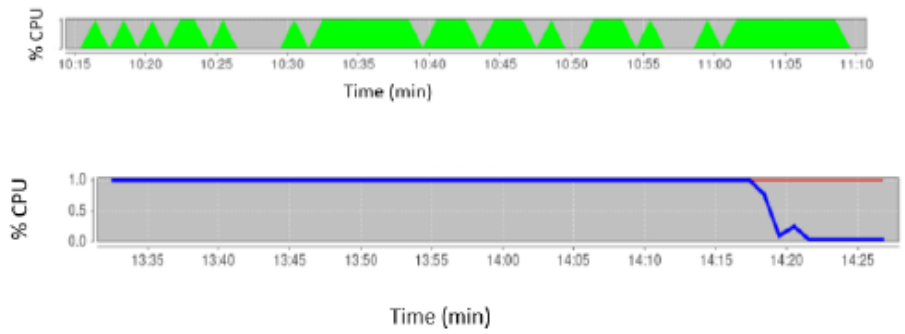

Figure 1. Different Load pattern of CPU Utilization, approx. 3000 – 4000 hits per minute.

Figure 1 shows the load patterns which follow a cyclic sequence. In the second graph, % CPU suddenly drops from high usages (95%) to about 12% average. When it was about 95%, the transaction response times were very high and the system was showing sluggishness during the peak spikes. Data mining is done to capture two peak intervals where we observed a pattern, reaching a higher CPU utilization for a longer duration of time, this clearly shows abnormal behaviour of CPU utilization. We have also collected some data and did analytics on other parameters e.g. hard disk usages, database hits, etc. and these can also provide insights from these patterns for predictive modelling. In this paper, it is out of scope and will be investigated as an extension to current work.

5. TRAINING WITH SEMI SUPERVISED LEARNING (SSL)

We have used an expected-maximization semi-supervised classifier to train our model. In this approach we used labelled data with some amount of unlabelled data. This is used in conjunction with a small amount of data can produce considerable improvement in learning accuracy over unsupervised learning [5]. There are some advantages, in context to this research work, such as –

a) A scalable probabilistic approach
b) It can generate a model which simulates analogies of patterns based upon on different profile data sets in a complex enterprise applications environment
c) Can achieve optimisation in terms of time and accuracy by predicting results

Considering some assumptions for the semi-supervised learning to work e.g. if two distinct points d1, d2 are close enough, then there might be respective outputs b1, b2. If we do not consider these assumptions, it would be hard to develop a practical model for a known number of training data sets to predict a set of infinitely possible test-cases which are mainly unseen [16]. We also have used other parameters in using labelled data points such as - effort, time, tools and resources. Based upon the nature and potential implementation of this research, semi-supervised learning with forced-training [3][7] may provide some useful outcomes as it is based upon –

a) Learning (training) of data set with both labelled and unlabelled data
b) Results are obtained in less time
c) Assumptions of forced-training can reduce the training time

6. EXPERIMENTS FOR VALIDATION

In our validation process, we simulated the load pattern showing burst in traffic in complex enterprise test environment as we observed after collecting the data. This represented the real-world scenario patterns representing different transactions. The test environment has limited data (a sub-set of full data) which proportionally represents large data sets associated with the integrated applications. More than 129 live applications fully functional as the real applications environment. The process included –

i) Data collection using profile points and analysis. Data logs were created and extracted for a very short intervals of 5 minutes
ii) Data extraction and analysis of the of workload demand patterns over a long period of time – during last one year
iii) Generate synthetic workloads patterns
iv) Execute stress tests in test environment with large number of virtual users using system-applications as in real world scenario
v) Validate the results by extracting data from different profile points of the application threads and nodes on completion of the tests
vi) Training the model using semi-supervised learning approach (deep learning paradigm) [7],[11].
vi) Forecast the likelihood of the traffic burst (excessive CPU usages) using the trained model [4],[6].

6.1. Experimental Set-Up

We designed and configured the following experiment set-up to perform our experiments in the test environment.

I) *Virtual User Generator:* used to simulate key end-user business processes and transactions
Ii) *Controller:* to manage, control and monitors the execution of load tests
Iii) *Load Generators:* to generate virtual load and simulate work-load patterns while large number of virtual users generating web-traffic and exhibiting load patterns simulating web-traffic bursts

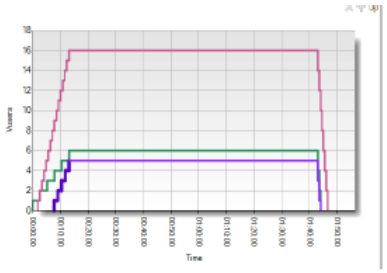

Figure 2. A Typical Load Profile with Virtual Users

Figure 2 shows user work-load profile (stress conditions) with different ramp up (slopes) times. This set up is used for validation of our results.

7. FORECASTING TRENDS

To study and analyse a trend in the load patterns we have worked out the aggregate demand difference of each occurrence of the pattern from the original workload [15]. We used a modified ETS (exponential smoothing) algorithm with ETS point predicts are equal to the medians of the predict distributions.

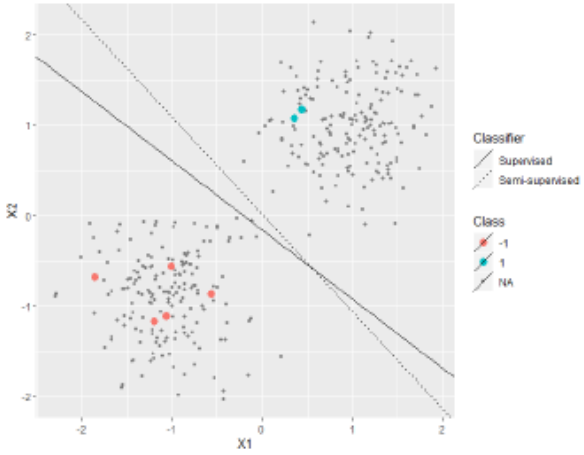

Figure 3 (a) Semi-supervised v/s Supervised learning (1year data set)

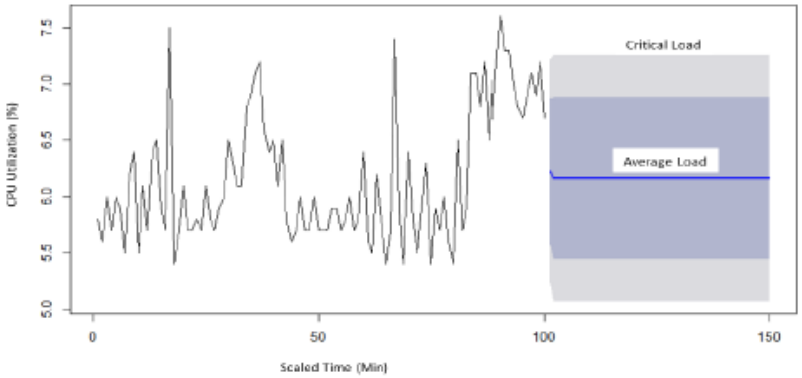

Figure 3(b) CPU Utilization under burst of traffic load conditions

Figure 3(a) shows the results of a modified semi-supervised neural network model, which is used to predict burst of traffic [9][10]. This model is now a part of our monitoring process in continuous evaluation of the demand patterns, as shown in Figure 3(b). This model predicts the burst of traffic behaviour and sets alarm for the system architects to take remedial actions e.g. re-allocation of IT resources to avoid a system crash or failure.

8. CONCLUSION

We have developed and implemented a novel practical approach to predict burst in traffic behaviour in a complex and highly integrated environment (test or pre-production) where more than 130 IT applications were live and thousands virtual users generate user-load under stress conditions. Our integrated enterprise environment had a distributed system with more than 300

servers serving more than 450 clients simultaneously. With a semi-supervised neural net classifier, the proposed approach predicts and identifies the burst in traffic in a complex enterprise IT infrastructure.

Data analytics enabled the system architects and system capacity planners to distribute the work-load appropriately. The proposed practical approach helped the IT architects to mitigate the risk of an unexpected failure of the IT systems, due to burst of traffic patterns, within a very short duration of time (3 to 4 hours) compared to 1 - 2 weeks as in the current practice. Validation of our results were done in an integrated test environment where alerts are activated as soon as the collective CPU utilization of the server's crosses 70% threshold critical limit. Experiments performed in test environment validated that our approach to predict potential burst of traffic worked effectively. In addition, we have found that this approach has benefited our department in efficient management of IT resources and helped to plan IT capacity for future demand predictions. This resulted in saving cost due to the optimum resource allocation in our IT enterprise IT environment.

As further work, we are working on investigating the impact of different parameters e.g. hard - disk failures, network latency [14], different types of transactions [13] and trying to develop a hierarchical semi-supervised learning model to extract patterns and to design an accelerated semi-supervised learning for predictive modelling.

REFERENCES

[1] Daniel Gmach, Jerry Rolia, Ludmila Cherkasova, Alfons Kemper, (2007) "Workload Analysis And Demand Prediction Of Enterprise Data Center Applications", IEEE 10th International Symposium On Workload Characterization, Boston, USA.

[2] Jia Li, Andrew W. Moore, (2008) "Forecasting Web Page Views: Methods And Observations", Journal Of Machine Learning Research.

[3] Adams, R. P. And Ghahramani, Z. (2009) "Archipelago: Nonparametric Bayesian Semi-Supervised Learning", In Proceedings Of The International Conference On Machine Learning (ICML).

[4] H. Zhao, N. Ansari, (2012) "Wavelet Transform Based Network Traffic Prediction: A Fast Online Approach", Journal Of Computing And Information Technology, 20(1).

[5] Yuzong Liu, Katrin Krichhoff, (2013), "Graph Based Semi-Supervised Learning For Phone And Segment Classification", France.

[6] Danilo J Rezende, Shakir Mohamed, Daan Wierstra, (2014) "Stochastic Backpropagation And Approximate Inference In Deep Generative Models", Proceedings Of The 31st International Conference On Machine Learning, Beijing, China.

[7] Diederik P. Kingma, Danilo J Rezende, Shakir Mohamad, Max Welling, (2014) "Semi-Supervised Learning With Deep Generative Models", Proceedings Of Neural Information Processing Systems (NIPS), Cornell University, USA.

[8] Pitelis, N., Russell, C., And Agapito, L. (2014) "Semi-Supervised Learning Using An Unsupervised Atlas". In Proceedings Of The European Conference On Machine Learning (ECML), Volume LNCS 8725, Pages 565 –580.

[9] Kingma Diederik, Rezende Danilo, Mohamed Shakir, Welling M, (2014) "Semi-Supervised Learning With Deep Generative Models", Proceedings Of Neural Information Processing Systems (NIPS).

[10] L. Nie, D. Jiang, S. Yu, H. Song, (2017) "Network Traffic Prediction Based On Deep Belief Network In Wireless Mesh Backbone Networks", IEEE Wireless Communication And Networking Conference, USA.

[11] Chao Yu, Dongxu Wang, Tianpei Yang, Et., (2018) "Adaptive Shaping Reinforcement Learning Agents Vis Human Reward", PRICAI Proceedings Part-1, Springer.

[12] Xishun Wang, Minjie Zhang, Fenghui Ren, (2018) "Deep RSD: A Deep Regression Method For Sequential Data", PRICAI Proceedings Part-1, Springer.

[13] Avital Oliver, Augustus Odena, Colin Raffel, Ekin D Cubuk, Et. (2018) "Realistic Evaluation Of Semi-Supervised Learning Algorithms", 6th International Conference On Learning Representations, ICLR, Vancouver, BC, Canada.

[14] Kenndy John, Satran Michael, (2018) "Preventing Memory Leaks In Windows Applications", Microsoft Windows Documents.

[15] M.F. Iqbal, M.Z. Zahid, D. Habib, K. John, (2019) "Efficient Prediction Of Network Traffic For Real Time Applications", Journal Of Computer Networks And Communications.

[16] Verma. V, Lamb. A, Kannala. J, Bengio. Y, Paz DL, (2019) "Interpolation Consistency Training For Semi Supervised Learning", Proceedings Of 28th International Joint Conference On Artificial Intelligence IJCAI Macao, China.

AUTHORS

Nitin Khosla Mr Khosla has worked about 15 years as Asst. Professor at MNIT in the Department of Electronics and Communication Engineering before moving to Australia. He acquired Master of Philosophy (Artificial Intelligence) from Australia, Master of Engineering (Computer Technology) from AIT Bangkok and Bachelor of Engineering (Electronics) from MNIT. His expertise is in Artificial Intelligence (neural nets), Software Quality Assurance and IT Performance Engineering. Also, he is a Certified Quality Test Engineer, Certified Project Manager and a Quality Lead Assessor. During last 14 years, he worked in private and public services in New Zealand and Australia as a Senior Consultant in Software Quality. Currently he is Asst. Director in Australian Federal Government in Performance and Capacity Management and leading multiple IT projects.