Middle-East Journal of Scientific Research 24 (11): 3616-3619, 2016
ISSN 1990-9233
© IDOSI Publications, 2016
DOI: 10.5829/idosi.mejsr.2016.3616.3619

Context Aware Load Balancing in Grid Computing Using Prediction with Time Series Data

[1]R. Rajeswari and [2]N. Kasthuri

[1]Assistant Professor, Department of Computer Science
and Engineering, M.P.N.M.J Engineering College, Erode, India
[2]Professor, Department of Electronics and Communication Engineering,
Kongu Engineering College, Erode, India

Abstract: The basic nature of increasing complexity of grid systems presents a huge challenge for grid tasks such as grid schedules. Load prediction is significant for increasing the overall performance so as to improve load balancing. Contemporary software designed to manage distributed applications should focus on the problem of forecasting the load of computing systems. In the past, predictions of load in grid were performed only with static history. In this approach, a novel load prediction model namely Context Aware Load Balancing using Machine Learning (CALB-ML) is presented to predict the load of grid computing systems from time-series data. With the help of this approach, effective decisions will be made based on the current context. Experimental result shows that proposed model outperforms well when compared to the existing approaches.

Key words: Context-Aware Computing • Grid Computing • Load Balancing • Prediction • Time-Series Data

INTRODUCTION

Grid computing is the collection of computing resources from multiple sources used to reach a common goal. The name “Grid” can be denoted as a distributed system with workloads that involve huge number of files.

Grid environments are dynamic in nature. Hence grid scheduling algorithms always need of assistance to predict the load for making decisions on how to utilize the resources efficiently and effectively. Analyzing the performance of the way in which Grid components respond to forecast the future load requirements is the significant research in Grid Computing.

The primary contribution of this manuscript is in three folds:

- Brief review on Prediction of Load in Grid is presented.
- Challenges in Context Aware Load Prediction are discussed.
- Way of predicting the Load in Grid Environment is visualized.

The rest of this paper is structured as follows. Section two discusses a detailed review on prediction of load in Grid. In section three, the main focus is given to present the proposed model. Section four presents experimental evaluation. Section five elevates the conclusion and possible trends for future research directions.

Related Works: The potential to accurately forecast the future resources and their capabilities is more important for many applications and scheduling algorithms which determine how effectively use time-shared resources in a dynamic grid environment.

Zhang *et al*. [1] presented a method to predict one-step ahead CPU load in a grid. This strategy uses polynomial fitting method to forecast the future CPU load based on the behavior of several steps in the past and history of similar patterns.

Goyeneche *et al*. [2] studied how recent data mining and statistical approaches define jobs in grid based production environments and presented a methodology which defines a template with the help of two

Corresponding Author: R. Rajeswari, Assistant Professor, Department of Computer Science and Engineering, M.P.N.M.J Engineering College, Erode, India.

characteristics using various priorities and weights with respect to the level of templates' prediction accuracy for future purpose.

Wu *et al*. [3] proposed a hybrid model to predict the status of n-step-ahead load with the help of interval values. This model also integrates autoregressive (AR) model with the estimations of confidence interval to predict the future load of a grid system. They introduced two filtering strategies to eliminate noise of the data and improve prediction accuracy.

Demand of computational resources in a grid increases rapidly. Due to this issue, complexity of grid increases and software level conflict occurs. To prevent these problems, Nayak & Gupta [4] provided the CPU load prediction model which is deployed over clustered grid environments. They found loads and load parameters which are not only based on historical patterns but also on other factors such as network, number of processes, memory, background processes and supporting applications.

Pop *et al* [5] presented an approach for the optimization of meta-scheduling in Grid environment using prediction methods. Various methods were analyzed in the paper for the state prediction of resources to be used in meta-scheduling. Distribution of resources such as CPU or free memory was considered in order to improve the availability of system. Notable improvements were obtained for scheduling optimization, with an immediate effect on load balancing and resource utilization.

Proposed Model: CALB-ML: This novel model uses Context Aware Load Balancing using Machine Learning (CALB-ML) approach. The proposed approach is illustrated in the following Figure 1. Time series data is given as input to data transformation / translation engine which facilitate the data mining algorithm to analyze the patterns using machine learning algorithms. The main task of this engine also includes the removal of noise and replacement of missing values in the dataset. Based on the patterns generated by the machine learning algorithms, load balancing task is carried out in grid through better predictions of future load.

In this approach, varieties of time-series based machine learning algorithms such as Perceptron, Decision Stump, Hoeffding Tree [6], Stochastic Gradient Descent (SGD) and Naïve Bayes [7] are investigated. Experiments are conducted for choosing the best approach to perform effective decision making.

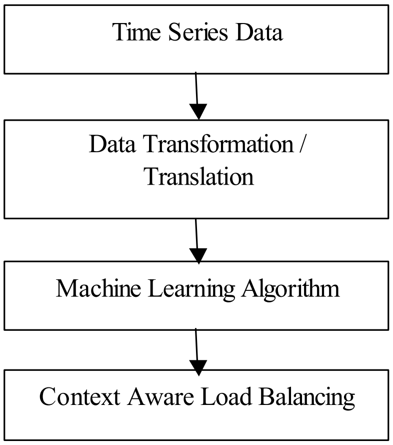
Time Series Data
Data Transformation / Translation
Machine Learning Algorithm
Context Aware Load Balancing

Fig. 1: Flow of CABL-ML Model

Perceptron: It is a supervised learning algorithm of binary classifiers. It consist functions that can make a decision whether an input, represented by a vector of numbers, belongs to some specific classes or not. This algorithm also allows for online learning.

Decision Stump: It is a machine learning model also called as 1-rules, consisting of one-level decision tree. It is a decision tree with one root tree which is immediately connected to the leaves. It generates a prediction based on the value of a single input feature. It is offten used as components such as bagging and boosting.

Hoeffding Tree: It is an incremental, decision tree induction algorithm that is capapble of learning from big data / data streams, assuming that the distribution generating examples does not change over time. Hoeffding trees utilize the fact that a small sample can often be enough to choose an optimal splitting attribute.

SGD: This approach implements stochastic gradient descent for learning various linear models (binary class SVM, binary class logistic regression, squared loss, Huber loss and epsilon-insensitive loss linear regression). The main advantage of SGD is that it replaces all missing values and transforms nominal attributes into binary values. It also normalizes all attributes so that the coefficients in the output are based on the normalized data [6].

Naïve Bayes: It is one kind of probabilistic classifier based on Bayes theorem. Naïve Bayes classifiers are highly scalable, requiring a number of parameters linear in number of variables (features / Predictors) in a learning

Table 1: Core Parameters

S. No	Parameters	Description	Possible Values
1	Latency	Represents any kind of delay that happens in data communication over a network	{low, average, high}
2	Bandwidth	The amount of data that can be transmitted in a fixed amount of time.	{low, average, high}
3	Efficient use of Grid resources	Denotes whether Grid resources are efficiently used	{yes, no}
4	Maximum load	Denotes whether the system supports maximal load	{yes, no}
5	Proper load balancing	Denotes whether proper load balancing is available	{yes, no}
6	MOS	Mean Opinion Score given by the user	{poor, average, good}

problem. Naive Bayes classifiers can be trained very efficiently and effectively in a supervised learning setup.

The core parameters used in this approach are Latency, Bandwith, Efficient use of Grid resources, Maximum Load, Proper Load Balancing and Mean Opinion Score (MOS) which collectively provide better predictions of future load. They are detailed in Table 1. Mean Opinion Score [8] is an average value of scores across subjects on a predefined scale that the subjects are assigned to the opinion of the performance of a system.

Experimental Evaluation: Large Hadron Collider (LCG) dataset had been utilized in this research provided by the e-Science Group of High Energy Physics (HEP) at Imperial College, London. This dataset was further improved by synthetic generation of the data. This data set contains 10,000 instances. Load prediction models concerning time-series data were applied for the purpose of making effective decisions. The proposed model was experimented using various predictive methods. This experimentation was conducted in Massive Online Analyzer (MOA) which is an integrated module of Waikato Environment for Knowledge Analysis (WEKA) Toolkit. WEKA is a Java based workbench that contains a collection of visualization tools and algorithms for data analysis and predictive modeling together with graphical user interfaces for easy access to these functions. MOA is open-source framework software that allows building and running experiments of machine learning or data mining on evolving data streams. It includes a set of learners and stream generators and contains several collections of machine learning algorithms including the above mentioned algorithms used in our research. The performance of proposed system was evaluated using various measures such as accuracy, kappa and kappa-temp which are detailed in equations (1), (2) and (3) respectively.

1) Accuracy is defined as

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

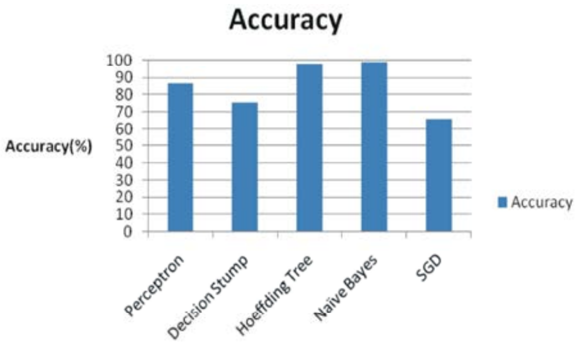

Fig. 2: Comparison of Machine Learning Algorithms in terms of Accuracy

Where, TP, TN, FP and FN are the rate of values calculated in four categories namely True Positive, True Negative, False Positive and False Negative.

2) Kappa Statistic is defined as

$$K = \frac{P_0 - P_c}{1 - P_c} \quad (2)$$

where, P_0 and P_c are classifier's prequential accuracy and probability that a chance classifier makes a correct prediction respectively

3) Kappa Temp represents Kappa statistic in temporal dependence which is defined as

$$K = \frac{P - P_{per}}{1 - P_{per}} \quad (3)$$

Where, P and P_{per} are prior probability for any observation and accuracy of the persistent classifier respectively [7, 8].

Performance of the above mentioned machine learning algorithms Perceptron, Decision Stump, Hoeffding Tree, SGD and Naïve Bayes are compared which is shown in Figure 2 to 4.

From the results, it is observed that Naïve Bayes approach outperforms well in terms of accuracy, kappa and kappa-temp.

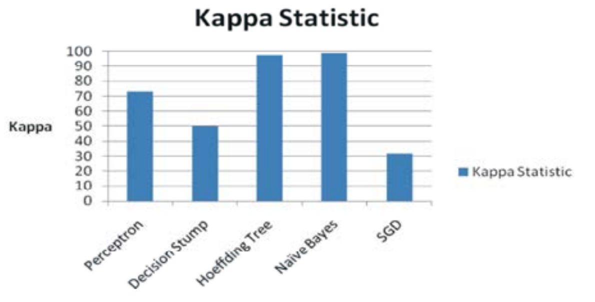

Fig. 3: Comparison of Machine Learning Algorithms in terms of Kappa Statistic

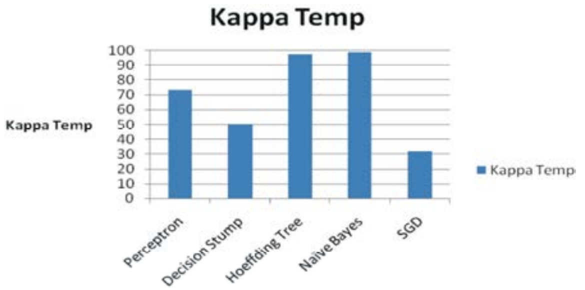

Fig. 4: Comparison of Machine Learning Algorithms in terms of Kappa temp

CONCLUSION

This paper has presented several prediction methods applied in Grid based environments. This research work has used data streams to predict the load based on the context. Grid computing has been evolved as a significant research area due to the increasing number of scientific applications. This work has also presented the new model for prediction system. Various prediction algorithms have been implemented and analyzed as a part of the research work. The Naïve Bayes predictions have been analyzed in comparison with other simple prediction algorithms. Predicting the future status of the grid resources composing a distributed system is inevitable in a highly dynamic system.

REFERENCES

1. Zhang Yuanyuan, Wei Sun and Yasushi Inoguchi, 2006. CPU Load Predictions on the Computational Grid, Sixth IEEE International Symposium on Cluster Computing and the Grid Workshops.
2. Goyeneche Ariel, Gabor Terstyanszky, Thierry Delaitre and Stephen Winter, 2007. Improving Grid Computing Performance Prediction using Weighted Templates, Conference Proceedings of the UK e-Science 2007 All Hands Meeting, Nottingham, UK.
3. Wu Yongwei, Yulai Yuan, Guangwen Yang and Weimin Zheng, 2007. Load Prediction using Hybrid Model for Computational Grid, 8th Grid Computing Conference and IEEE.
4. Nayak Rahul and Prof. Rashmi Gupta, 2013. CPU Load Predictions on the Computational Grid using Distance Based Algorithm, International Journal of Advances in Computer Science and Technology, 2(7).
5. Florin Pop, Alexandru Costan, Ciprian Dobre and Valentin Cristea, 2009. Prediction based Meta-Scheduling for Grid Environments, CSCS-17 Conference.
6. Hulten Geoff, Laurie Spencer and Pedro Domingos, 2001. Mining Time-changing Data Streams, ACM International Conference on Knowledge Discovery and Data Mining, pp: 97-106.
7. Johm George, H. and Pat Langley, 1995. Estimating Continuous Distributions in Bayesian Classifiers, Eleventh Conference on Uncertainty in Artificial Intelligence, San Mateo, pp: 338-345.
8. https://en.wikipedia.org/wiki/Mean_opinion_score