Adaptive Prediction Models for Data Center Resources Utilization Estimation

Lameya Afroze

Summer term 2020

This paper[4] presents a new technique which automatically selects the most promising machine learning method to estimate the data center resources utilization.

1 Introduction

Cloud infrastructure services provide virtualization technology where clients can access their storage and servers directly like a traditional data center. Accurate estimation of resource utilization is important for minimizing the operational cost and maximizing the performance of data center by job scheduling, load balancing, allocate virtual machines efficiently. Clients share computing resources in multi-tenant cloud which makes it difficult to estimate future resource utilization for data center. The authors has presented an adaptive multi-methods approach that adapt the most appropriate method and give prediction based on present scenario.

2 Proposed System Methodology

Data center uses traditional machine learning techniques for resource utilization as has low dimension[4]. Recently, some works has also been done using deep learning but is avoided here as it performs well with large amount of data with high frequencies. The proposed system has been described here which is showed in Figure 1.

2.1 Workload prediction

Resource utilization logs are divided into fixed size sliding windows and contains specific interval of time period. Each of these window data is used for predicting workload with minimum error. The authors used four machine learning models for getting desired results (each of these models has different set of property). These models are Linear Regression, Support vector regression, Gradient boosting and Gaussian Process Regression (in paper mentioned as Kriging).

2.2 Adaptive Model Selector (AMS)

Adaptive Model Selector (AMS) works as a trained decision maker to decide which prediction model should be used in multi-model methodologies among several machine learning models. AMS predicts the best regression model based on some input features in every time stamp. After that, this regression model is used to generate workload. For adaptive model selector five classification models are used which are- K-Nearest Neighbors, Naïve Bayes, Multilayer Perception, Random Decision Forest (RDF) and Gradient Boosting. Each of this model has their unique characteristics which helps to predict better result (for example, KNN gives prediction by providing more priority of nearest neighbor, Naïve Bayes classifier gives the prediction based on conditional probability, RDF selects the most voted class and so on).

3 Proposed model evaluation

3.1 Datasets

Three different datasets are used for evaluate proposed multi model. First dataset is Alibaba Data Set[1] where authors focused only CPU time series for doing the experiment. The second one is Bitbrains Data set[2] where 20 VM were selected randomly and used CPU utilization which are more than 30 %. The third data set is Google cluster traces[3] where 500 VMs were randomly selected with average 21.89% CPU utilization. For Alibaba dataset the authors used the performance traces of 1313 machines for 12 hours to train the multi model and test that model using Bitbrains dataset and Google cluster traces.

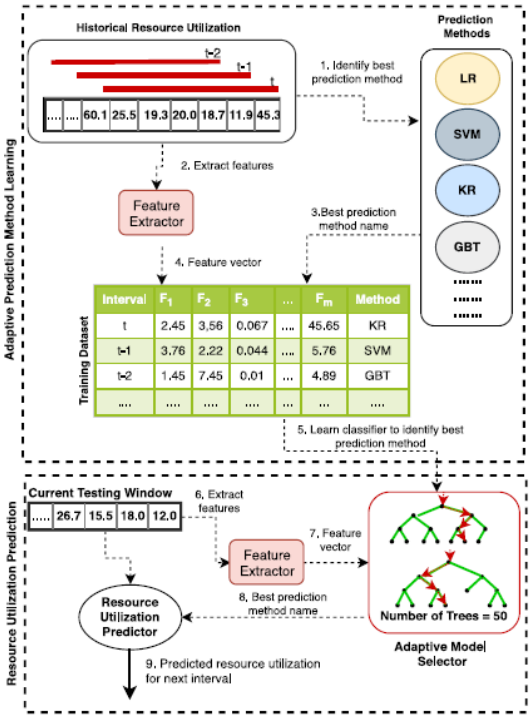

Figure 1: Purposed system overview to learn adaptive model selector and using it to estimate the data center resource utilization.Source :[4], Figure 2.

3.2 Feature extraction & AMS evaluation

The authors used both manual and automatic extraction process to get the features from time series data. The proposed system first filtered the data using TSFRESH and then use a open source library that use three different methods to filter data[4]. After the feature extraction process, 1006 features were selected that consists standard deviation, skewness, auto-co relation at different lags etc.

All classifiers are trained with 80% of Alibaba dataset and remaining 20% were used for validate the best classifier model[4]. The classifier was evaluated through training time, prediction time, size, accuracy, precision, recall, F-measure, true positive rate (TPR), false positive rate (FPR), true negative rate (TNR) and false negative rate (FNR).

3.3 Resource estimation & window size sensitivity

The proposed model estimates the resource utilization for next time interval based on the available data of present time. The authors measured the deviation of regression model by using the following equation of Root-Mean Square Error (RMSE) and Mean Absolute Error (MAE). Here n is number of performed estimations at any interval t where a_t represents true CPU utilization and p_t represents estimated CPU utilization[4].

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(a_t - p_t)^2}{n}}$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|a_t - p_t|$$

Window size indicates the amount of data that the receiver can receive at any point of time. The authors observed the effect of using several window size (window sizes of 20, 40, 60, 80 and 90 minutes) to train AMS model with less error for getting estimated resource utilization .

4 Experimental results

4.1 AMS evaluation

AMS selects the best model considering these matrices - true positive rate (TPR), false positive rate

(FPR), true negative rate (TNR),false negative rate (FNR), precision, recall, f-measure, and accuracy. Among these five classification model RDF shows the best performance for measuring the CPU resources and KNN is the second best classifier according to the features showed in Figure 2. After

Classifier	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	Accuracy
KNN	0.62	0.11	0.88	0.37	0.65	0.65	0.65	0.65
MLP	0.64	0.11	0.88	0.35	0.66	0.67	0.66	0.67
NB	0.33	0.22	0.77	0.66	0.38	0.31	0.29	0.31
RDF	**0.65**	**0.10**	**0.89**	**0.34**	**0.68**	**0.68**	**0.68**	**0.68**
GBT	0.48	0.16	0.83	0.51	0.55	0.53	0.51	0.53

Figure 2: AMS evaluation results using different classifiers for Alibaba dataset[4]. Source: TABLE 2

that these classifiers are used to measure time and space efficiency using the features shown in Figure 3. It depicted that KNN doesnot produced good performance for time and space efficiency. On the other hand Naïve Bayes takes less time to train and test the AMS model but it has poor efficiency for other features showing in Figure 3. So, RDF is chosen as AMS model as it takes less time and outperforms rest of the evaluated model for other evaluation features.

Classifier	Training Time (sec)	Prediction Time (sec)	Prediction Time per Request (ms)	Size (KB)
KNN	3.23	593.61	17.017	255283.2
Multi-layer Perceptron	728.13	0.34	0.010	180.7
Naive Bayes (Guassian)	0.59	0.13	0.004	7.5
RDF	**57.43**	**0.51**	**0.015**	**201523.2**
GBT	186.45	0.28	0.008	140.9

Figure 3: Time and space efficiency of AMS using different classifiers for Alibaba dataset[4]. Source: TABLE 3

4.2 Dataset resource estimation & window size sensitivity

Table 1 shows the experimental results of CPU utilization for three datasets using RMSE and AME. For each datasets the proposed method generates minimum RMSE and AME and shows better performance than other existing methods. Figure 4 shows the actual CPU utilization rate for Alibaba datasets using different methods along with the proposed AMS model for four different machines with different characteristics (heavy workload, low workloads, high variance and low variance). The estimated resource utilization is close to the actual resource utilization for all these four machines having different workloads as the proposed model dynamically selects different estimators. The authors used 60 minutes prediction time to train these machines and they found that the estimated results is close to actual utilization. Through the experiment it has been found that optimal size of the window is 60 minutes and proposed model generates less estimated error with this window size. If the window size has been increased more or decreased less then error will also increased. For this reason, 60 minutes window size is used to conduct the whole experiment.

Table 1: RMSE and MAE for resource estimation using proposed system for three dataset[4].

Method	Alibaba Dataset		Bitbrains Dataset		Google Dataset	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
GBT	4.57	3.43	9.74	2.85	2.31	1.24
LR	5.12	3.87	15.01	6.03	2.40	1.32
SVM	5.63	4.23	19.94	7.19	2.35	1.28
Kriging	5.26	3.99	15.80	6.05	2.28	1.24
Liu[6]	5.34	3.94	19.80	7.09	2.26	1.24
Proposed	3.32	2.29	9.13	2.57	2.22	1.14

5 Discussion

- **Identify best prediction model:** RDF is selected as AMS because it can predicts resource utilization appropriately with highest accuracy. RDF works as an ensemble containing individual decision trees. Each decision tree predicts the class and the most voted class is considered as classifier. It prevents the over fitting of individual decision tree to produce more accurate results. Using this classifier, the proposed model increased the accuracy from 6% to 27% which is far better than other existing methodologies[4].
- **Comparison of proposed model with existing works:** Several techniques has been proposed in recent years to predict the resource of data center more efficiently.Liu et al. [6] proposed an adaptive selective model that change

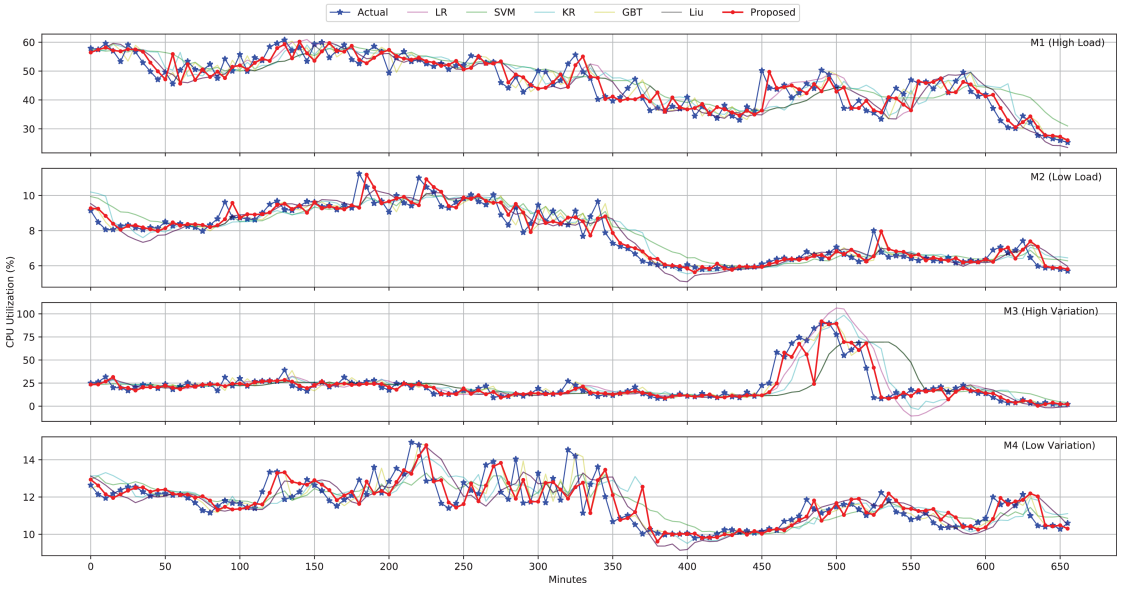

Figure 4: Actual vs proposed method CPU prediction for Alibaba data set for four selected machines. M1 = Heavy workload, M2 = Low workload, M3 = High variation, M4 = Low variation. The window size used to train the prediction model is 60 minutes. Source:[4], Figure 10.

the prediction model depending on the change of workload. If the workload is slow it selects LR otherwise it selects SVM to estimate CPU resource.Rahmanian et al.[5] proposed an ensemble approach that uses Learning Automata (LA) to weight each predictor.The propose model works with time series features which the authors claimed that has not done before using this feature in their knowledge.

6 Conclusion

This novel technique put more emphasis on window size to select the best prediction model adaptively for resource utilization. The authors have a plan to find out adaptive window size in future and predict estimate resource utilization for t+n interval of time.

References

[1] *Alibaba Cluster Log*. Available url = https://github.com/alibaba/clusterdata. Accessed: 16-May-2020.

[2] *Bitbrains Cluster Log*. Available url = http://gwa.ewi.tudelft.nl/datasets/gwa-t-12-bitbrains. Accessed: 16-May-2020.

[3] *Google Cluster Log*. Available url = https://github.com/google/cluster-data. Accessed: 16-May-2020.

[4] S. Baig, W. Iqbal, J. L. Berral, A. Erradi, and D. Carrera. "Adaptive Prediction Models for Data Center Resources Utilization Estimation". In: *IEEE Transactions on Network and Service Management* 16.4 (2019), pp. 1681–1693.

[5] A. A. Rahmanian, M. Ghobaei-Arani, and S. Tofighy. "A Learning Automata-Based Ensemble Resource Usage Prediction Algorithm for Cloud Computing Environment". In: *Future Gener. Comput. Syst.* 79.P1 (Feb. 2018), pp. 54–71. URL: https://doi.org/10.1016/j.future.2017.09.049.

[6] C. Liu, C. Liu, Y. Shang, S. Chen, B. Cheng, and J. Chen. "An adaptive prediction approach based on workload pattern discrimination in the cloud". In: *Journal of Network and Computer Applications* 80 (2017), pp. 35–44.