Energy Efficient Utilization of Computational Resources by Predicting CPU Idle Time

Ramesh T, R M Suresh

Abstract--- Consumption of energy in cloud data centers are growing at an alarming rate owing to regularizing of using cloud servers for data and resource sharing applications. The consumption of power usage in CPUs is not directly proportional to its utilization as the power feed to the cloud server is not regulated as per its CPU utilization levels. Hence, increasing the operational and maintenance cost. Though this problem can be curved if we could predict the idle CPS utilization time of variable lengths and thereby regulate its operational mode into various low-power or sleep mode in order to automatically control the power utilization. But such a system would require to synchronously train on the live CPU data as the problem of waking up the CPU at unoptimized level would result into latency issues and thereby reversing the outcome causing more power wastage then actually destined to save so. Therefore, development of such a online learning framework is a significant challenge for achieving energy savings at data centers. In this study, we present aonline learning based algorithm for CPU idle state prediction by training it on real time over the live streaming data of Cloud servers and its state of CPUs. Our results show that about 50-60% of power savings can be achieved over variety of other existing techniques for prediction CPU idle times.

Keywords—Cloud computing, CPU ideal time, workflows, management, online learning.

I. INTRODUCTION

Nowadays the trend in technology is towardscloud servers and remote sharing of computational resources. This has lead to more and more power usage. Thereby to cut the cost of operation it is required to build a system which can predict the downtime or idle time of CPUS utilization on order to switch it adaptively based on CPU usage. This is an inconvenient issue which can't be used for this application. Therefore, we need to come up with an online learning based solution i.e a learning algorithm which can be trained on real time live streaming data of CPU utilization. One step in this online learning based method is kernel based online learning along with ubiquitously computational load balancing method[1,3-5]. The working of the algorithm is based on closed loops. The outcome of the algorithm is confusion matrix where many new instances are created. The support set consists of the consolidation of the confusion matrices. The problem with support set is many of the confusion matrixes cannot be stored in it. Moreover it also leads to loss in memory. The online algorithm consists of set of learned data and is called as hypothesis. But the computation of online jobs require lots of memory and the support set is not able to provide such huge memory and the work pertaining to decision making is halted. Hence the weighted sum of different learned data are classified and they are placed in the support set. The real arise in allowing the system to make all the updates on its own based on the changes in the weighted sum. The hypothesis which is online are then prioritized. The drawback in the prioritization of online hypothesis is the storage requirements grew in size and there will not be enough memory to store all these data and this still continues to be the major concern in online learning algorithms.

Many algorithms involve data analytics and agents. Finding a solution based on data analytics and agents are a research concern. Budget is a new term coined to work towards its solution. This budget acts as a delimiter where instances of support set can be added or deleted [2, 3, 6]. SILK and NORMA can also be used for this purpose [5,7]. The relative mistake bound approach is used in [8]. Stochastic algorithm for the same purpose was also used in [9,10]. The Gaussian approach [12] uses the support sets and deletes the wrong support set. New parameters are used in [13] to get rid of scare resources in budgets.

The problematic features that restricts the usage of online learning for the designated application are as follows:

The limits of bandwidth and
The Limits of latency
Relative Memory Loss Bound
Memory Explosions

The substantial part of the degraded performance is the slowdown posed by all in combination.

Sparsity of the data sets there occurs latency to access RAM or cache for many cycles and ultimately causing substantially slowdown the computation & learning. Large data instances are required to be loaded for a complete online hypothesis to function.

II. METHODOLOGY

So far researchers have used the online learning architectures based on either linear model or that of Navie Bayes. In the following section we discussed the various other effective approach to facilitate online learning over a large scale and manage an effective trade-offs for the local training of the online hypothesis. Since, the online learning can't be simply on the single machine bit over the network where the network is ubiquitously connected to dynamically share the hardware resources over the multiple machines. Thus, the solution of the problem of memory explosions lies

Revised Manuscript Received on April 12, 2019.

Ramesh T, Sathyabama University Chennai, Tamilnadu ,India. (E-mail: tumaati@gmail.comline)

Dr R M Suresh, Sri Lakshmi Ammal Engineering College ,Chennai Tamilnadu ,India. (E-mail: rmsuresh@hotmail.com)

in the strategy to model the job processing environment based on parallel learning algorithms to manage delays and resource management.

Here the proposed online learning algorithm for CPU idle time prediction is shown below.

The CLOUDSIM simulator is used for the execution purpose where the learning of data is done . The data which is missed are later recovered through out proposed algorithm. Our algorithm is based on checkpoints where the system learns itself and identifies the different route. The system works in a different way from the existing systems. Most of the existing work deletes the instances so as to maintain the memory that can hold the support system. In our proposed approach, a hybrid technique is used which creates and stores the support system dynamically. Here the algorithm builds itself identifying the support set [10].

Algorithm: Sparse Online Learning Algorithm (SOLA)

Input:

s_i,s_j- instances of the support pairs,

S_k - support set

H_k - online hypothesis at time t

p_k- pair equilibrium sequence of CPU idle times.

Output:

H_o- online hypothesis

Step 1: For k = 1, 2, 3 …

Step 2: Get Instance S_k

Step 3: Calculate paired sequence

$$p_k = \frac{[s_i * s_j * q]}{[s_i * s_j]}$$

$$S_k = \sum_i p_k\,(t - t_i)$$

Step 4: calculate idle time bound:

While $l_i \neq l_n$

{

$$R[l_i] = \frac{1}{p_k}\sum_{l=1}^{t}(S_k - S_{k-1}) \int_t l_i\,dt$$

$$M_t = \sum_k l\,(k - k_i)$$

$$where, l_i = \frac{(p_k\,\Delta k)^l}{l!}\,\exp(-p_k\,.\Delta k)$$

Step 5: Hypothesis updation:

if $R[k_i] \leq R[S_t]$

{

$$\sum_{l=l}^{\infty}\sum_{l_i}^{m}\frac{(p_k)^{k-k_i}}{(M_t - l_i)!}.\exp(-p_k\,.\Delta t).\frac{(p_k)^{k_i}}{(k_i)!}$$

}

else

{

Print "Failed to Update"

$$l_i = Null\ //Assign$$

$$l_i = l_{i+1}$$

}

Step 6: Addition of support set instances

if $p_t t_i \leq k_i$

{

$$\sum_{k=l}^{\infty}\frac{(p_k l_i + p_{k+1} l_{i+1})^k}{k!}\exp(-pk\quad .\Delta k)$$

}

else

{

$$S_k = S_{k-1}$$

}

} //end while

Step 7: End process

III. RESULTS & DISCUSSION

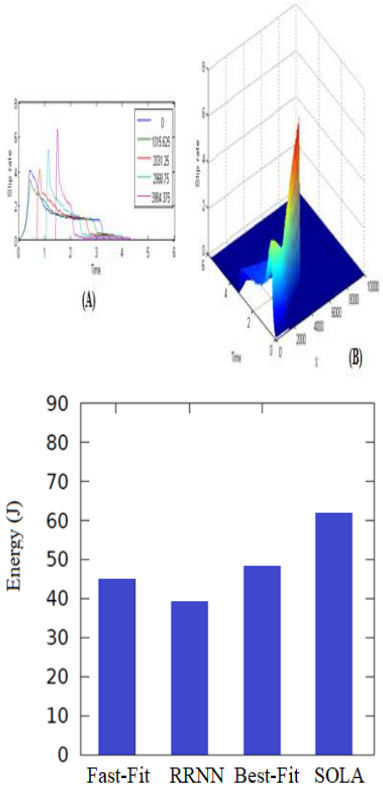

Figure 1: (A) Idle time prediction slip rate for the network flow during the idle time prediction. (B) 3D plot of the idle time prediction slip rate in combination with the time flow and the transmitted data (x). (C) Performance compraission of the energy reduction in simulated cloud server setting. The proposed algorithm works better than the existing ones in that it performs well under less memory constraints. When new instance pairs are added , our algorithm works fine by expanding the memory needed to suit the learned data that can be accommodated in the support set. The hypothesis that undergoes the training process can be found by the pair of instances that was formulated during the previous training process of the hypothesis.

The performance comparison chart of the proposed algorithm is given in figure 1. In figure 1(A), the updating of hypothesis based on the idle time prediction is highlighted. The figure 1(B) depicts the results during the saturated time limit. The main advantage of our proposed system is that the learning algorithm adapts itself to new data and forming new support set.

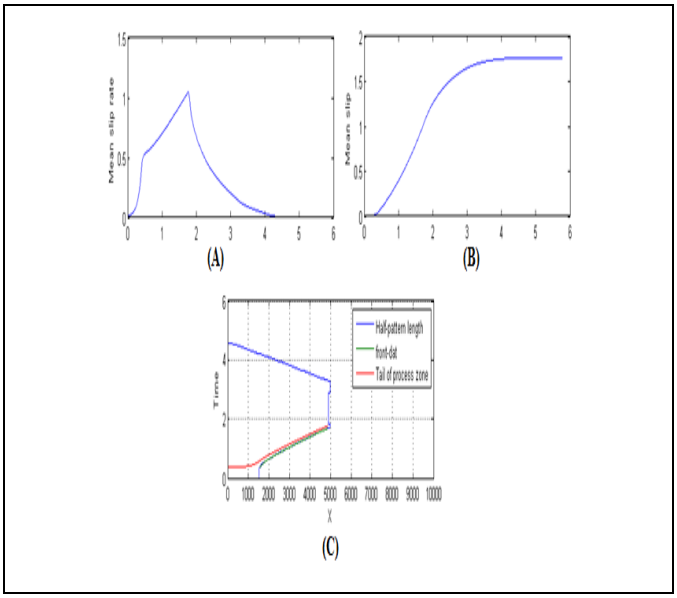

Figure 1: (A) The mean slip rate v/s time in seconds for the first phase in flow of idle time prediction, (B) Mean slip rate v/s time in seconds for the end phase in flow idle time prediction achieving saturation at converge level, (C) plot of the time v/s flow and the transmitted data (x); giving rise to curvelet transformation for the half pattern length of sequences used in the proposed algorithm.

IV. CONCLUSIONS

The average online error of the proposed SOLA algorithm is a function directly dependent on the size of support sets for different data sets. This algorithm solves the challenges with online learning by computationally proving convergence rates and error bounds in the size of the support sets with descent in the number of the mistakes made which gradually saturates at nominal level. Here, the energy reduction achieved is comparatively better than other widely used method. With our method we have peaked 50-60% of energy savings in simulated setting. This allows the applications of to rely over online based processing methods for anomaly detection for online and time-varying problems. Our works solve the online to batch conversion to a bounded batch solution, which is scalable in quadratic size of the support sets. The result put forth in the study provides the essential theoretical insights in the scenario of drifting targets and dynamic hypothesis.

REFERENCES

1. Y. Freund and R. E. Schapire. “Large margin classification using the Perceptron algorithm,” Machine Learning, pp. 277–296, 1999.
2. K. Crammer and Y. Singer,“Ultraconservative online algorithms for multiclass problems” Journal of Machine Learning Research, vol. 3, pp. 951–991, 2003.
3. K. Crammer, J. Kandola, and Y. Singer,“Online classification on a budget. In Advances in Neural Information Processing Systems 16, 2003.
4. K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. Journal of Machine Learning Research, 7:551–585, 2006.
5. J. Kivinen, A. Smola, and R. Williamson, “Online learning with kernels,” IEEE Trans. on Signal Processing, vol. 52, no. 8, pp. 2165–2176, 2004.
6. J. Weston, A. Bordes, and L. Bottou. Online (and offline) on an even tighter budget. In Robert G. Cowell and Zoubin Ghahramani, editors, Proc. of AISTATS 2005, pages 413–420, 2005.
7. L. Cheng, S. V. N. Vishwanathan, D. Schuurmans, S. Wang, and T. Caelli. Implicit online learning with kernels. In Advances in Neural Information Processing Systems 19, pp. 249–256, 2007.
8. O. Dekel, S. Shalev-Shwartz, and Y. Singer, “The Forgetron: A kernel-based perceptron on a budget. SIAM,” Journal on Computing, vol. 37, no. 5, pp. 1342–1372, 2007.
9. N. Cesa-Bianchi, A. Conconi, and C. Gentile,“On the generalization ability of on-line learning algorithms,” IEEE Trans. on Information Theory, vol. 50, no. 9, pp. 2050–2057, 2004.
10. N. Cesa-Bianchi, A. Conconi, and C. Gentile,“A second-order perceptron algorithm. SIAM Journal on Computing,”vol. 34, no. 3, pp. 640–668, 2005.
11. N. Cesa-Bianchi, A. Conconi, and C. Gentile. Tracking the best hyperplane with a simple budgetPerceptron. In Proc. of the 19th Conference on Learning Theory, pp. 483–498, 2006.
12. L. Csat´o andM. Opper,“Sparse on-line gaussian processes,” Neural Computation, vol. 14, pp. 641–668, 2002.
13. J. Langford, L. Li, and T. Zhang, “Sparse online learning via truncated gradient”. In Advances in Neural Information Processing Systems, vol. 21, pp. 905–912, 2008.