ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 3, March 2017

Resource Requiremens In Cloud

[1] Sudha Pelluri,[2] Ramachandram Sirandas
[1][2] Dept. of Computer Science & Engineering Univ. College of Engg. (A), Osmania University Hyderabad

Abstract— Resource providers on Cloud offer heterogeneous resources such as compute units and storage in Virtual Machine instances (VM). Cloud providers expect users to request for resources. In this process, overestimation of resources by cloud users lead to unused resources. The cumulative unutilized resources for each job of the user, amount to unnecessary expenditure for users and wasted resources for providers. Large scale data centers that provide reliable high performance computational and storage services for Cloud providers, face problems of increased energy consumption and CO2 emission as a consequence of huge resource wastage. Therefore, for environmental and financial reasons it is imperative to reduce unnecessary resource reservation. This can be done by using resource prediction technique that ensures resource allocation only as much as is necessary for the customer. Currently, there are no suitable prediction techniques for Cloud resource usage because of absence of pattern, trend and seasonality in users' resource usage [2].The proposed prediction approach applies Enhanced Instance Based Learning and is tested using Google Cluster Trace Data [1] . The contribution of this work to existing body of work on cloud resource management by effective resource prediction approaches are discussed.

***Keywords*— Resource Prediction, Enhanced Instance Based Learning, Cloud Usage Data.**

I. INTRODUCTION

Cloud computing is being widely adopted by many organizations because of cost effective services offered to meet users' requirements.

This work contributes to managing IaaS by proposing a new approach for resource prediction. Cloud service management is a complex task consisting of among various other activities, resource provisioning. Currently users are allocated resources based on their requests. But, resources requested by users are found to be overestimated than their actual requirement [3]. Underestimation of resources can cause resource shortage and consequent revenue loss due to penalties for SLA (Service Level Agreement) violation. Overestimation can lead to idle resources and increased costs. This cumulative wasted resources, results in unnecessary expenditure for customer (he has to pay for each resource unit he reserves for certain period of time) and wasted resource for provider (he could have provisioned resources to other users) for each request. It is necessary to utilize these resources properly in order to decrease cost to each user.

Provisioning of resources is a challenging issue being faced by the service providers in Cloud because, the requests come from numerous users, requirements dynamically change and there is no specific pattern, trend or seasonality in the resource usage of users on Cloud. The Google cloud usage data published as trace, has been studied and the usage of resources and resource requested in shown in figure 1.1.

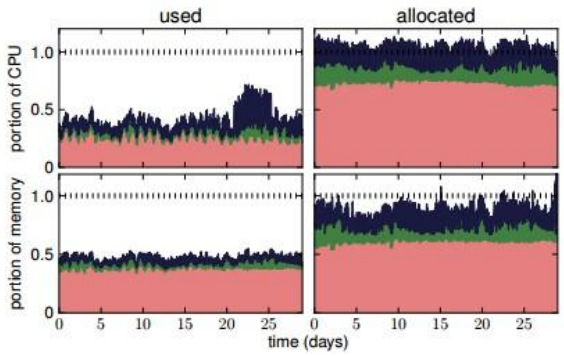

Figure 1.1: Used Resources Vs Allocated Resources of Google Cluster Data. (From [2])

This shows that, resource requirements of the users need to be met as per their use rather than what users request for. The real challenge is to be able to correctly predict such quantity of resources they will actually use.
As another important insight, currently resources are provisioned by the cloud providers (Google compute Engine, Amazon AWS, Rightscale etc.,) only based on users' requests. Provider tries to optimally use resources available with them while being able to meet the SLAs (Service Level Agreements) of various users. Currently, mechanisms that try to continuously provision requested resources are said to be reactive. Such mechanisms use threshold based rules to detect and react on resource shortages, e.g., time, performance, sum total of minimum available resources. With this resource

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 3, March 2017

allocation approach, to overcome even small resource shortages, it takes time of order of few minutes. This is very costly for applications which need frequent scaling. The mean instantiation latency in Amazon EC2 [3] is around two minutes [4]. In experiments conducted as specified in [5], it takes additional two minutes for a Cassandra server [6] to reach its maximum throughput.

Thu reactive elastic resource provisioning

scheme like in Amazon EC2 [3], RightScale [7] etc., needs some improvement. Predictive resource provisioning can save the setup time that would be required, can make the required resources available, without the user having to make exact quantum of resource request to the provider. The price that providers have to spend comes down due to power saved by resource saving. Such prediction is useful to meet the objective of GREEN COMPUTING. However, prediction of host load in cloud is challenging because it fluctuates drastically at small timescales. Data Centers are used to house large compute and storage resources to be made available to users by cloud providers. Cloud computing applications run, using multiple computers connected by a network. As mentioned in [5] only power costs individually, are lower than infrastructure costs, and less than the servers themselves. Power is only 23% of the total, but power distribution and cooling make up 82% of the costs of infrastructure. Cost of building is 12-15%. Hence, overall power consumption costs are considerable. PUE is ratio of non-computing overhead energy (like cooling and power distribution) to the amount of energy used to power actual machines. Google claims that its data centers have current overhead of just 12%, making their PUE 1.12. Even with best practices, most companies are not able to reduce some inherent overhead power wastage. Hence, alternatives that help in reducing power to servers need to be identified.

Energy consumed by a machine is proportional to its CPU requests. The more the CPU requests, the larger the frequency scaling factor, which results in cubic increment in power consumption. Hence, goal of this work is to predict and measure the use of resources (compute units and memory units) as close to the actual demand as possible without having to wait for resource requests from users.

Challenges

Resource provisioning and Energy estimation gets complicated in the Cloud scenario as compared to the Web based and Grid based systems because:

1. Resources are requested by the user in real time. The resources are requested when the application starts and this information is not available beforehand. Users expect Instantaneous Resource availability. The resources are to be made available to the users as the application execution proceeds. This is difficult to implement by the Cloud Provider because provisioning of remote resources takes time as explained in [3] ,[4],[5] and [6] . Problem is alleviated because of erratic requests by various users as shown by the Cloud usage trace - Google Trace data as described in [9].

2. To make resources available as per changing requests of users, to enable dynamic scalability of resources, reactive approaches are easier to implement (as the provider gets information from users regarding the required resource scale-up) but take unacceptable time to provision resources (as explained in previous section). As new users without historical information are more common, predictive and proactive provisioning is difficult for cloud.

3. Unlike scientific applications that run on Grids and HPC platforms, cloud tasks are shorter and more interactive. Most common examples are word search, image or mail search. Hence, they tend to be drastic and short-term load fluctuations in clouds compared to grids. Cloud workload shows huge variability with respect to time as seen in the poor auto correlation functions(ACF). Hence, the trace of Cloud usage shows that it does not lend itself to existing prediction approaches like Time series, Queuing models Bayesian model, SVM, Neural networks etc., because of absence of patterns, trends and seasonality. Corresponding correlogram is shown here in fig.1.2. The Google Cloud workload shows no specific pattern in users resource usage. The various tasks and their resource consumption do not follow an exact pattern as discussed in [2]. Basic difference is non availability of trend, seasonality as shown in the figure 1.2 and figure 1.3.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 3, March 2017

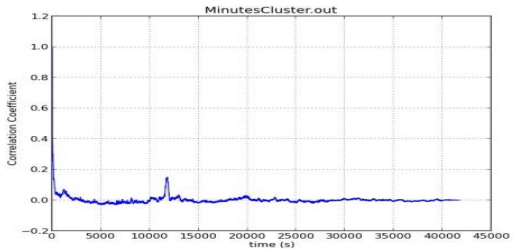

Figure 1.2: Correlogram of Google Cluster Data(from[21])

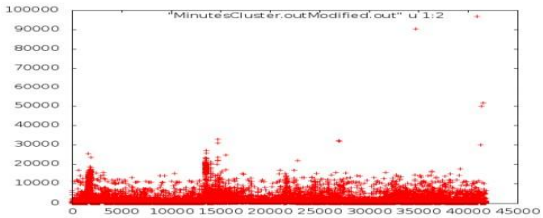

Figure 1.3: Trace of Cloud workload- Request arrival Pattern (X-axis: Time in seconds, Y-axis : CPU Usage) (from [21])

4. The machines that are made available to users by various providers like Google Cloud Platform- Google Compute Engine-available at cloud.google.com/compute/docs are quite variable. For e.g. Standard machine types, High memory machine types, High CPU machine types. Requests for these machines compute units, the virtual CPUs are actually implemented on Intel Sandy Bridge, Intel IVY Bridge, Intel Haswell machines. Google Compute Engine unit is a unit of CPU capacity that is used to describe the compute capability of machine types. Google has chosen 2.75 GCEUs to represent the minimum computational capacity of one virtual CPU (a hardware hyper-thread) on Sandy Bridge, Ivy Bridge, or Haswell platforms. These are different in their energy consumption - idle, full load and average. Therefore it becomes difficult to exactly estimate the power consumed per user request for CPU.

II. WORKLOAD DESCRIPTION

Workload that is representative of real Cloud workload has been identified and characterized. A real trace of Cloud usage as presented in Google Cluster-usage traces published in Nov. 2011 is used as workload for analyzing the efficiency of our approach. From the various tables two tables of interest are Task Events table and Task resource Usage Table. These tables are merged on jobid. CPU Request and CPU Usage units are core count, and for Memory units are bytes which are normalized. The normalization is a scaling relative to the largest capacity of the resource on any machine in the trace. Hence it is an absolute number.

Other traces used in similar works are - Real VM trace log of IBM Smart Cloud Enterprise (SCE) product to conduct the experiments. This trace is actually available only within IBM and described in [34]. Some selected Universities were given access to Yahoo trace logs. To gain insight on MapReduce workloads 10 months of trace data from the M45 supercomputing cluster, a production Hadoop environment of Yahoo was used. Other traces from Grids like AuverGrid, Nordu Grid and Web access logs like worldCup1998 trace, are not exact representatives of real cloud usage. Hence those traces are not used in experimentation and testing.

III. METHODOLOGY

Initially the two files - Task events table and Task resource usage table are merged on jobid. "IBM SPSS statistics" is loaded with these tables. The proposed algorithms, Distance Weighted Averaging and Locally Weighted Regression are tested with the merged file using SPSS Statistics.

Enhanced Instance Based Learning

In implementation of Enhanced instance based learning, we select from Google Cluster trace data, user id, CPU requested, CPU used, memory requested, memory used are selected for various cases. Proposed Enhanced Instance Based Learning (EIBL) approach involves following steps -

Resource requirement prediction by :

a) Distance Weighted Averaging
b) Locally weighted regression

IV. EVALUATION AND ANALYSIS

After having followed the various steps in EIBL, performance of the approach is measured using parameters mentioned below :

1. Resource savings obtained
2. Prediction accuracy
3. Data set size used

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 3, March 2017

Resource Savings

By using proposed EIBL approach there is a huge reduction in wastage of resources as compared to ad- hoc method of resource reservation by the user. For users who have multiple instances (many jobs for which resources were requested) in each cluster, resource requirement prediction by both DWA and LWR are performed.

CPU savings : CPU savings are shown with line graphs as below. Here the sum of savings obtained per job (savings obtained across various tasks of each job are summed) are shown. This is a small sample of the large number of Jobs, each having variable tasks per job.

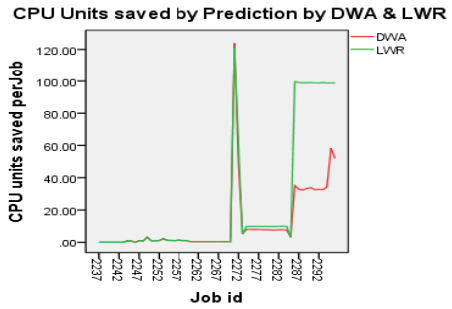

Figure 4.1:Compute units by using DWA and LWR

Memory savings : Memory savings are shown with line graphs as below. Here the sum of savings obtained per job (savings obtained across various tasks of each job are summed) are shown. One observation that can be made is, quantum of memory saved is not as huge as quantum of CPU saved by the prediction approaches. This is a small sample of the large number of Jobs, each having variable tasks per

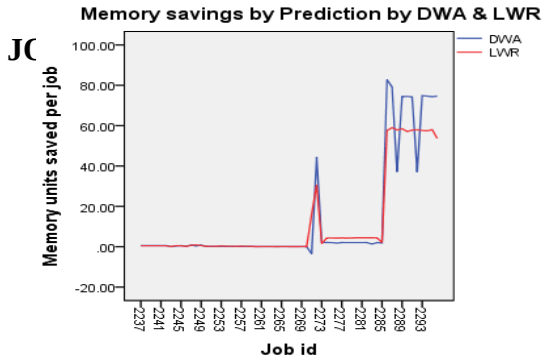

Figure 4.2: Memory units saved by each job by using DWA and LWR

Analysis of Resource savings obtained

The results of resource savings obtained are available in comprehensive comparative tables

UID	CPU Savings				Memory Savings			
	% reduction in estimation Difference		Quantum of CPU Savings		% reduction in estimation Difference		Quantum of Memory Savings	
	DWA	LWR	DWA	LWR	DWA	LWR	DWA	LWR
148	93.5224	89.345	4.1060	4.1088	39.5163	89.8876	0.24872	0.5645
169	81.8071	34.2984	12.2526	24.8200	90.7856	76.5354	12.2389	10.3178
412	96.7212	96.6687	382.4585	382.6592	89.523	98.8988	39.0773	43.1701

Table 4.1. Resource Savings obtained (Per user)

Three users (with id – 148, 169 and 412) who have significant presence(more number of jobs) in Data set are selected .Their Cumulative savings across various jobs, as percentage of savings (for CPU and Memory) are compared - Ad-hoc estimation Vs prediction by DWA and LWR are shown. The Quantum of savings(CPU and Memory) are also shown. CPU Request and CPU Usage units are core count, and for Memory units are bytes which are normalized. The normalization is a scaling relative to the largest capacity of the resource on any machine in the trace. Hence it is an absolute number.

Accuracy

Accuracy of Prediction is measured by :

a) By Measuring Residuals.
b) By using paired sample ' t ' tests

Proof of Accuracy using Residuals:

In order to evaluate the accuracy of prediction, the predicted values are compared against the real values of the resource actually used by the user. Based on readings obtained from the experiments, we can check how near is our prediction to resource used (Zero residual is accurate).
Residual = Resource units predicted by proposed approach - Actual Resource units used by the user

Accuracy of CPU prediction **:** The line graphs show the residuals for CPU usage by initial users estimate by ad-hoc method, values predicted by using DWA and values predicted by using LWR as seen in fig.4.3.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 3, March 2017

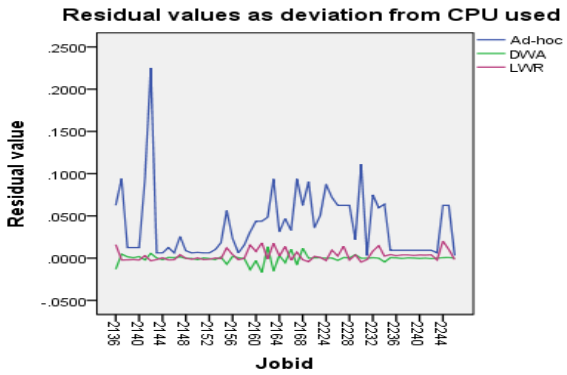

Figure 4.3: Residuals for CPU usage of by Ad-hoc estimate, DWA and LWR

Accuracy of memory prediction : Residuals because of using DWA or LWR can be seen in fig. 4.4. This shows that predicted values of resource are much nearer to the actual resource usage values.

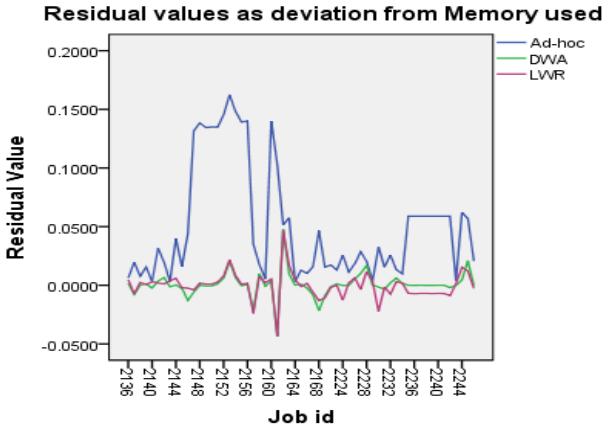

Figure 4.4: Residuals for Memory usage by Ad-hoc estimate, DWA and LWR

Analysis of Accuracy of prediction.

The comprehensive data in table 5.2 shows Mean Absolute Percentage Error(MAPE) and percentage reduction in error by Ad-hoc method of requests by user, resource requirement prediction by DWA and by LWR.

Masure of Accuracy- Paired sample ' t ' test: The paired sample 't' test show that mean difference between the predicted values of CPU using DWA , using LWR, from actual CPU usage values is not statistically different from 0,(as the p value observed > 0.05). Hence the results are very near to actual resource usage values when DWA is used for CPU usag e prediction.

Table 4.2. Resource requirement estimation accuracy

UID	Residual in CPU Estimation					Residual in Memory Estimation				
	MAPE			% reduction in Error		MAPE			% reduction in Error	
	Ad-hoc	DWA	LWR	DWA	LWR	Ad-hoc	DWA	LWR	DWA	LWR
148	2357.85	291.81	255.35	87.63	89.17	69.52	47.77	56.21	37.71	21.18
169	2341.81	4.694	307.21	99.91	94.25	173.66	57.34	104.87	100.71	49.39
412	2976.85	63.614	173.44	97.83	94.17	209.99	77.76	61.72	62.97	70.61

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 1	CPUrate_mean - DWA	.000919	.006860	.001194	-.00151	.003351	.769	32	.447
Pair 2	CPUrate_mean - LWR	.00285949	.00668205	.00116320	.00049014	.00522885	2.458	32	.020

Table 4.3. : Result of paired 't' test showing the accuracy of DWA for CPU for user uid 148

The paired Sample "t" test for memory requirement prediction shows accuracy of DWA and LWR.

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 1	assignedmemoryusage_mean - MemDWA	-.00037	.013945	.002428	-.00532	.004574	-.153	32	.880
Pair 2	assignedmemoryusage_mean - MemLWR	.001591	.013620	.002371	-.00324	.006421	.671	32	.507

Table 4.4: Result of paired 't' test showing accuracy of DWA and LWR for user uid 148

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 3, March 2017

Data size used

UID	Residual in CPU Estimation					Residual in Memory Estimation				
	MAPE			% reduction in Error		MAPE			% reduction in Error	
	Ad-hoc	DWA	LWR	DWA	LWR	Ad-hoc	DWA	LWR	DWA	LWR
148	2357.85	291.81	255.35	87.63	89.17	69.52	47.77	56.21	37.71	21.18
169	2341.81	4.694	307.21	99.91	94.25	173.66	57.34	104.87	100.71	49.39
412	2976.85	63.614	173.44	97.83	94.17	209.99	77.76	61.72	62.97	70.61

Actually there are 5536 jobs in the bucket considered. But for one user request, resource prediction requires data from one cluster. Based on request size, Cluster size can be 76 jobs, 12 jobs or 14 jobs on considered data. Of the 30 features that characterize a job, only 2 features (CPU Request and CPU usage mean) or (CPU Request and Memory Request) are required in DWA and 3 features (CPU Request and Memory Request) or (CPU Request, Memory Request and CPU usage mean) in LWR. The five Clusters into which the data file was divided based on k means clustering contain information regarding various users. Cluster 1 and Cluster 3 had only 3 users each, Cluster 4 had 34 users, cluster 2 had 4611 users and Cluster 5 had 352 users. Hence, only Cluster 2 and Cluster 5 information is used in experiments. Of the various users, only 3 users had significant presence in these two clusters, cluster 2 and Cluster 5, User id- 148, user id.- 169 and User id - 412 show significant number of cases. So, most of the results being discussed here are for these 3 users. It is not possible to measure accuracy of prediction using our approach with work of any other users as no other work gives prediction on cloud usage data as on date.

V. RELATED WORK

Useful work that enables us to understand the basic difference between Cloud, Grid and Cluster computing is a early work by Ian foster, discussed in[10]. Dynamic resource allocation strategy at the VM level is implemented as **horizontal scalability or vertical scalability** in [11]. Vertical Resizing adjusts logical partition of resources (e.g, CPU, Memory, Bandwidth, I/o etc.) in a VM. Without rebooting by using Dynamic Logical Partition Resizing (DLPAR) it is possible to attach and detach resources from logical partitions. Amazon AWS EC2 and some third party cloud management services like Rightscale [7], Azure-Watch [12], Scalr [13] etc., offer schedule based and rule based auto-scaling mechanisms. Rule based mechanisms work on user defined triggers by specifying instance scaling limits and corresponding actions. RightScale [7] and AzureWatch [12] use some middleware metrics like database connections, web server requests, name resolution queries and queue sizes.

Basic approaches of workload prediction for auto-scaling is introduced in [14]. The ARMA method is suggested as prediction technique. The most common resource prediction approach is the Time series approach as explained in [15]. A comprehensive text on time series [16] introduces the concepts of time series in detail. An excellent text which gives basic information is [17]. Clear explanation of various types of time series analysis methods is available. The linear models for prediction of host loads as originally suggested by Box Jenkins by using AR, MA, ARMA, ARIMA, ARFIMA are discussed in [18]. Application of ARIMA model is discussed in [19]. In [20] the authors have proposed a resource prediction approach by using Double Exponential Smoothing which considers current state and history of resource used. Cloud workload is modeled as a G/G/N queuing model in [21]. Authors have described the detailed approaches to Bayesian forecasting techniques with case studies in [22]. The use of Support vector machines and Artificial Neural Networks for application performance modeling is described in [23]. Artificial Neural networks are efficient when the fitness function Chosen is efficient. A modified Genetic Algorithm is discussed in [24]. In [25], three different approaches for prediction, Artificial Neural Networks, ARIMA time series and Regression are used for prediction of spring flow. An integrated approach for predictive elastic scaling which involves multiple approaches combined is discussed in [26]. A fast Fourier transform for repeating patterns and Discrete Markov chain with finite number of states for applications without repeating patterns are integrated. A combination approach for prediction system is presented in [27]. Multiple types of VM demands based on request history, with specific ensembles for each type are

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 3, March 2017

used.The use of input features in prediction is described in [28]. The Google Trace Ver 2.0 , which is a 29 day trace on 12k-machine cell in May 2011 has been used in proposed work. The paper by Mishra et al.,[29] captures the heterogeneity and dynamicity of data in Google trace. In [30], authors compare the two workloads-GridMix3 and Yahoo production cluster by using k means clustering approach. By using time series the data of real cloud usage trace from IBM hosted cloud is studied for both CPU and Memory usage in [31]. A useful paper that helps in understanding the workload (for jobs, tasks) and host load (at machine level) in a Google Data Center in comparison to the Grid system is provided in [32] by Sheng Di.

VI. CONCLUSION

Saving resources on cloud by the resource providers and paying for only those resources that will actually be used, rather than paying for resources reserved by the user is the motivation for this work. This is huge business advantage if this work can be adopted by cloud providers. Though the problem is interesting, there are no easy approaches possible in resource prediction possible for Cloud environment.

The real challenge is that we want to predict what amount of resource user will use in the future. There is no direct relationship between the users requests for resources and actual usage values. Also, there is no pattern of users resource usage data. This makes use of various existing approaches time series, Queuing models, Neural networks etc., infeasible for the cloud environment. The other major challenge is that Laboratory setup to generate synthetic workload cannot replicate the real data of cloud usage. The real cloud usage is much different from any synthetic workload that can be generated. This problem was overcome by using cloud usage trace data published by Google. Proposed Cloud Resource usage prediction method has been evaluated using Google Cluster Data. Using the proposed Enhanced Instance Based Learning (EIBL), the saving of resource units per Job is quite significant as shown in Table 4.1. This also translates to significant saving of energy units per user jobs. This enables better and efficient utilization of the resources by the service provider. This proposed approach when scaled and worked in real cloud system, will be extremely beneficial to both - the users and service providers. Savings in terms of CPU units and memory units.. As shown, the sum total of saving obtained for multiple instances of a single user amounts to huge savings. This when translated to cost that a user will save for all resource units, it is great economics. For the provider, it is an excellent method to allocate resources to users only based on prediction and hence not block resources with multiple users at any time. Overall, this is a substantial contribution to existing knowledge on much needed effective prediction techniques for Cloud.

Future Work

Capacity planning is very useful consequence of this work. Machine consolidation based on efficient resource usage possible. Efficient Scaling is now possible because resource requirements are predicted beforehand. **Therefore, applications that require instant and variable resources can easily adopt this EIBL approach to predictive scaling and efficient energy saving**. Further, adoption of this approach into tools that can display to user information on resource availability, reservation and billing information will be extremely useful. **As an extension, the energy saved by use of this work, hence, green cover saved can be explicitly computed.**

REFERENCES

[1]. Reiss, Charles, John Wilkes, and Joseph L. Hellerstein. "Google cluster-usage traces: format+ schema." *Google Inc., White Paper* (2011): 1-14.

[2]. C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M.Kozuch, " Heterogeneity and dynamicity of clouds at scale: Google trace analysis," in *Proceedings of the Third ACM Symposium on Cloud Computing.* ACM, 2012, p. 7.

[3]. A. E. C. Cloud, "Web page at http://aws. amazon. com/ec2," Date of last access: 20th Jan, 2016.

[4]. R. Bryant, A. Tumanov, O. Irzak, Scannell, K. Joshi, M. Hiltunen, A. Lagar-Cavilla, and E. De Lara, "Kaleidoscope: cloud microelasticity via vm state coloring," in *Proceedings of the sixth conference on Computer systems.* ACM, 2011, pp. 273–286.

[5]. H. Nguyen, Z. Shen, X. Gu, S. Subbiah, and J. Wilkes, "Agile: Elastic distributed resource scaling for infrastructure as a service," in *Proc. of the USENIX International Conference on Automated Computing* (ICACAZ´13). San Jose, CA, 2013.

[6]. A. Cassandra, "Apache cassandra," 2013.

[7]. T. Clark, "Rightscale," 2010.

[8]. J. Hamilton, "Cooperative expendable micro-slice servers (cems): low cost, low power servers for internet-scale services," in *Conference on Innovative Data Systems Research CIDRA¨´09)(January 2009)*, 2009.

[9]. A. Ali-Eldin, M. Kihl, J. Tordsson, and E. Elmroth, "Efficient provisioning of bursty scientific workloads on the cloud using adaptive elasticity control," in *Proceedings of the 3rd workshop on Scientific Cloud Computing Date*.ACM, 2012, pp. 31–40.

[10]. I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," in *Grid Computing Envi- ronments Workshop, 2008. GCE'08*.IEEE, 2008, pp. 1–10.

[11]. W. Wang, H. Chen, and X. Chen, "An availability aware virtual machine placement approach for dynamic scaling of cloud applications," in *Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing 9th International Conference on*. IEEE, 2012, pp. 509–516.

[12]. T. Redkar and T. Guidici, *Windows Azure Platform*. Springer, 2009.

[13].A. Singhai, S. Lim, and S. R. Radia, "The scalr framework for internet services," in *Proceedings of the 28th Fault-Tolerant Com- puting Symposium (FTCS-28), page (to appear)*, 1998.

[14]. Z. Shen, S. Subbiah, X. Gu, and J. Wilkes, "Cloudscale: elastic resource scaling for multi-tenant cloud systems," in Proceedings of the 2nd ACM Symposium on Cloud Computing.ACM, 2011, p. 5.

[15]. C. You and K. Chandra, "Time series models for internet data traffic," in *Local Computer Networks, 1999. LCN'99. Conference on*. IEEE, 1999, pp. 164–171.

[16]. P. J. Brockwell, *Introduction to time series and forecasting*. Taylor & Francis, 2002, vol. 1.

[17]. C. Chatfield, *The analysis of time series: an introduction*.CRC press, 2013.

[18]. P. A. Dinda and D. R. O'Hallaron, "Host load prediction using linear models," *Cluster Computing*, vol. 3, no. 4, pp. 265–280, 2000.

[19]. W. Fang, Z. Lu, J. Wu, and Z. Cao, "Rpps: A novel resource prediction and provisioning scheme in cloud data center," in *Services Computing (SCC), 2012 IEEE Ninth International Conference on*. IEEE, 2012, pp. 609–616.

[20]. J. Huang, C. Li, and J. Yu, "Resource prediction based on double exponential smoothing in cloud computing," in *Consumer Electronics, Communications and Networks (CECNet)*, 2012 2nd International Conf. on. IEEE, 2012, pp. 2056– 2060.

[21].A. Ali-Eldin, J. Tordsson, E. Elmroth, and M. Kihl, "Workload classification for efficient auto-scaling of cloud resources," Technical Report, 2005.[Online]. Available: http://www8. cs. umue/research/uminf/reports/2013/013/part1.pdf, Tech. Rep., 2013.

[22]. J. Geweke and C. Whiteman, "Bayesian forecasting," *Handbook of economic forecasting*, vol. 1, pp. 3–80, 2006.

[23].S. Kundu, R. Rangaswami, A. Gulati, M. Zhao, and K. Dutta, "Modeling virtualized applications using machine learning tech- niques," in *ACM SIGPLAN Notices*, vol. 47, no.7. ACM, 2012, pp. 3–14.

[24]. L. Aranildo Rodrigues, P. S. de Mattos Neto, and T. Ferreira, "A prime step in the time series forecasting with hybrid methods: The fitness function choice," in *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*. IEEE, 2009, pp. 2703– 2710.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 3, March 2017

[25].K. Mohammadi, H. Eslami, S. D. Dardashti, *et al.*, "Comparison of regression, arima and ann models for reservoir inflow forecasting using snowmelt equivalent (a case study of karaj)," *J. Agric. Sci. Technol*, vol. 7, pp. 17–30, 2005.

[26].Z. Gong, X. Gu, and J. Wilkes, "Press: Predictive elastic resource scaling for cloud systems," in *Network and Service Management (CNSM), 2010 International Conference on*. IEEE, 2010, pp. 9–16.

[27].Y. Jiang, C.s. Perng, T. Li, and R. Chang, "Asap: A self-adaptive prediction system for instant cloud resource demand provisioning," in *Data Mining (ICDM), 2011 IEEE 11th International Confer- ence on*. IEEE, 2011, pp. 1104– 1109.

[28].T. Miu and P. Missier," *Predicting the Execution Time of Workflow Blocks Based on Their Input Features*." Computing Science, New-castle University, 2013.

[29]. A. K. Mishra, J. L. Hellerstein, W. Cirne, and C. R. Das, "Towards characterizing cloud backend workloads: insights from google compute clusters," *ACM SIGMETRICS Performance Evaluation Review*, vol. 37, no. 4, pp. 34–41, 2010.

[30]. S. Aggarwal, S. Phadke, and M. Bhandarkar, "Characterization of hadoop jobs using unsupervised learning," in *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*. IEEE, 2010, pp. 748–753.

[31]. J. Tan, P. Dube, X. Meng, and L. Zhang, "Exploiting resource usage patterns for better utilization prediction," in *Distributed Computing Systems Workshops (ICDCSW), 2011 31st International Conf. on*.IEEE, 2011, pp. 14–19.

[32].S. Di, D. Kondo, and W. Cirne, "Characterization and comparison of cloud versus grid workloads , in *Cluster Computing (CLUSTER), 2012 IEEE International Conf. on*. IEEE, 2012,pp.230–238