Cost-Effective Resource Provisioning of Cloud Computing via Supervised Machine Learning

Daniel Andresen[1], Mohammed Tanash[1], Caughlin Bohn[2], and William Hsu[1]

[1] Kansas State University, Manhattan KS 66506, USA,
{dan,tanash,bhsu}@ksu.edu,
[2] University of Nebraska-Lincoln, Lincoln NE, 68588
cbohn4@unl.edu

Abstract. In this paper, we investigate the impact on cost effectiveness of employing Machine Learning techniques for predicting job resources (memory and time) in terms of resources and cost provisioning for running HPC workloads on the cloud. We evaluated our AMPRO-HPCC tool by comparing the run time and cost of 4.46 million jobs covered 2018-2021 years derived from Kansas State University HPC cluster (BEOCAT) logs. We found that our Machine Learning tool reduced the average cost of running jobs on the cloud by up to 39% and decrease the average running time by up to 39%.

Keywords: Supervised Machine Learning, HPC, Performance, Scheduling, Cloud Computing

1 Introduction

Cloud computing service providers such as Amazon Web Services (AWS) [1], Microsoft Azure [2], Google Cloud [3], and IBM Spectrum Computing [4] have been caught the attention in the field of High Performance Computing (HPC) and scientific computing community in the last decade due to availability and competition [5]. At the same time, cloud computing infrastructures are becoming more well known and popular because of the various number of services and different quality of service (QoS) they offer [6] [7].

On the other hand, running many of jobs on the cloud can become quite costly, especially when users require significant resources for their submitted jobs. Moreover, cloud users frequently request many more resources than their submitted jobs actually need to avoid potential jobs being terminated due to an insufficient amount of resources requested. Therefore, "one of the most challenging problems with real-time workflows in cloud computing is to get a cost-effective way to complete the workflow within the deadline" [8]. Thus, HPC users would benefit from getting feedback about determining resource allocations (memory and time) for their submitted jobs on the cloud, such as getting information about whether the amount of resources (memory and time) required of a particular submitted job were not enough, or too much of a particular run.

Such feedback requires more information history about actual usage of the resources of previous runs to assess and give feedback regarding required resources for any new runs. This data can be found in the sacct data provided by the Slurm resource manager.

Cloud computing has become more readily available and users are getting the advantage of the powerful resources and receiving results of running their extensive computations and simulations that require lots of resources in a short period of time. Cloud computing is highly capable because of the powerful hardware provided by cloud service providers. Two of the critical challenges of running jobs on the cloud are: cost-effectiveness and meeting critical deadlines.

Executing each particular job using cloud provider services is similar to submitting jobs on a local cluster. After a HPC user submits a job with a specific resource requirement, the cloud provider will then assign and reserve the needed computing resources from the resources pool. The job will then be assigned to the best-suited resources in order to be executed. The cost of running a particular job on the cloud depends on the amount of resources the user asked for that job. This means the more resources requested, the more cost for executing the job. On the other hand, the more resources the job requires, the more probability the job will be waiting on the queue for allocating the required resources and vice versa [9].

One of the most important factors of using HPC in the cloud is the cost. The cost to use the HPC cloud depends on many factors such as types of hardware offered, the amount of resources needed (cores, memory, time, etc.), and the capacity of storage needed. While most of HPC-intensive users are looking for cost-effective HPC cloud resources, it is still hard and challenging to decide how many resources are needed for a particular submitted job on the cloud. Hence, HPC cloud users usually consume much more resources than needed for their submitted jobs. This process is not cost efficient and can consume significant funding from their budgets and grants.

The basic ideas of our work are: **i)** Helping HPC users estimate and reduce the amount of money and resources needed on the cloud while maintaining the minimum resources needed for their submitted jobs on the cloud. Hence, reduce the budget. **ii)** measure the average saving budget using our machine learning model published in [10], and [11], versus not using our model for most popular HPC cloud services providers such as (AWS, Azure, Google Cloud, and IBM Spectrum Computing).

To achieve our goals, we calculate the amount of average resources and costs of usage of running millions of jobs from both HPC resources of the University of Colorado Boulder RMACC-Summit and from the Kansas State University Beocat, using actual, requested, and predicted usage generated from our ML tool AMPRO-HPCC [12]. We finally provide the comparison of resource usage and costs for both HPC resources using four well known cloud services providers Amazon Web Services (AWS), Microsoft Azure, Google Cloud, IBM Spectrum Computing, and on on-premises machine.

In this work, we investigate the impact of using our ML tool AMPRO-HPCC [12] in running jobs in the cloud by comparing the cost of running all jobs with and without using our machine learning tool on the most popular cloud computing resources.

2 Related Work

Using local HPC resources could be inadequate for application executions in many cases, such as big jobs that request more resources than the available ones on the local cluster. Moreover, big jobs need to wait a long time in a queue [13]. Thus, the ideal solution would be running these resource intensive jobs to the HPC cloud, which is known as HPCaaS (HPC as a Service) [14, 15]. Our work focused on all good matches jobs for cloud usage. Another study focuses on evaluating the performance on running applications on the cloud [13,16–18].

Jiyuan Shi et al. introduced an elastic resource provisioning and task scheduling mechanism to perform scientific workflows in the cloud. Their goal is to complete as many high-priority workflows as possible under budget and deadline constraints. Their techniques consist of three phases: workflow pre-processing, elastic resource provisioning, and task scheduling [19]. Lei Wu et al. proposed a cost optimization algorithm that emphasizes on resource provisioning in order to meet the deadlines of real-time workflow [8].

In the area of cost provisioning for real-time workflow in cloud computing environment, recent studies show that resource provisioning is much capable and successful than task scheduling [20]. Verma and Kaushal introduced a heuristic that benefits trade-off between deadline and budget under given constraints. Their proposed constrained heuristic is based on Heterogeneous Earliest Finish Time (HEFT) to schedule workflow tasks over the available cloud resources [21]. Wei Zheng proposed a variety of algorithms to help minimize the monetary cost of running big jobs on the cloud with deadline constraints to a satisfactory level. Their proposed work uses separate CPU frequency for each task to reduce the overall user cost [22].

Our work focuses on studying the effects of using machine learning techniques provided in our work [10,23], and conducting a detailed comparison of resource usage and costs of running jobs on the cloud. We mainly study the benefits of using our ML techniques in terms of saving resources usage and costs on running jobs on the cloud providers such as AWS, Azure, Google Cloud, and IBM Spectrum Computing.

3 Methodology

In this work, we focus on resource provisioning in cloud computing using our proposed ML tool AMPRO-HPCC provided in GitHub [23], which uses our ML Mixed Account Regression Model (MARM), explained in detail in [11], that helps and recommends the HPC users for predicting the amount of resources

needed (memory and time) for their submitted jobs. To be able to perform and implement our work, the workflow process includes the following four stages:

3.1 Stage 1: Collecting the HPC log data (sacct data)

One data set (sacct data) was collected from the Slurm workload manager database of HPC resources of the Kansas State University (Beocat) [24]. The data side has **10.9 million** instances and covers the years 2018-2019.

The collected data include the required features for the time requested set for each job (Timelimit), actual time usage for running each job (CPUTimeRAW), Minimum required memory (ReqMem), Maximum resident set the size of all tasks in each job (MaxRSS), State of the job (State), accounts information, name of Quality of Service (QoS), etc. We need that information in order to calculate the cost and build our MARM ML model to predict the amount of resources for each newly submitted job.

3.2 Stage 2: Data Cleaning and Filtration

At this stage, we prepare the sacct data by removing all certain jobs associated with missing values (NaN) associated with features MaxRSS or CPUTimeRAW. We replaced missing values of Timelimit, Partition, and QoS with default values. We consider all completed jobs only, therefore, jobs with incomplete State ('Cancelled', 'Failed', 'Deadline', etc.) were removed. At the end of this stage, we ended up having **4.46 million** jobs left in Beocat cluster.

3.3 Stage 3: Resources Prediction

At this stage, we calculate the predicted amount of resources (memory and time) using our ML tool AMPRO-HPCC that uses our MARM methodology. So, we can use the predicted values to calculate the accuracy of our ML model, the amount of resources provisioning, and the cost-effective resource provisioning in the cloud. Our machine learning tool achieves high predictive accuracy R^2 (**0.72** using LightGBM for predicting the memory and **0.74** using Random Forest for predicting the time). Note that, we did not predict the required number of CPUs because the system always guaranteed the requested number of CPUs provided by the user.

3.4 Stage 4: Calculate HPC resources needed

At this stage, we extract the amount of resources required (memory and time) for each job in Beocat resources from the cleaned data and calculate the average usage of memory and time for all jobs for each HPC resource as the following:

- Calculate the amount of resources (memory and time) required for each job using the amount of requested resources provided by the user (ReqMem,

and Timelimit). Hence, we can calculate the total and average amount of resources using the requested usage of all completed jobs for each HPC resource.
- Calculate the amount of resources (memory and time) required for each job using the amount of actual usage of resources provided from the Slurm workload manager (MaxRSS, and CPUTimeRAW). Hence, we can calculate the total and average amount of resources used by all of the completed jobs for each HPC resource.
- Calculate the amount of resources (memory and time) required for each job using the amount of predicted usage of resources provided from our ML tool AMPRO-HPCC that uses our MARM methodology. Hence, we can calculate the total and average amount of resources using the predicted usage of all completed jobs for each HPC resource.

3.5 Calculate the Cost of Running Jobs on the cloud

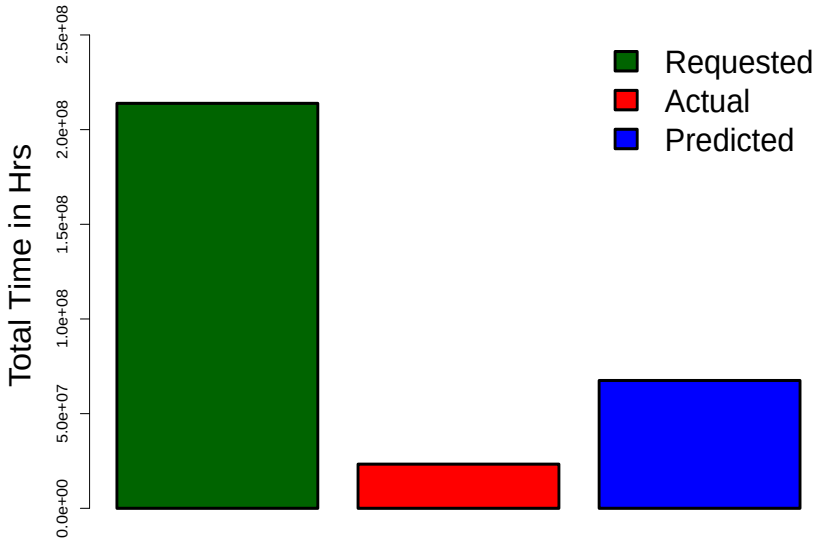
Our study will provide analytical comparison and calculation for the cost of running all successfully completed jobs for Beocat resources using multiple cloud service providers (Amazon Web Services, Google Cloud, Microsoft Azure, Digital Ocean, IBM Cloud) and the local HPC resources of Holland Computing Center. We have used the Seat Pricing model to estimate the cost of a computing job. A seat will be used as a portion of a compute node, either with 1 CPU Core and 4 GB of RAM or 1 CPU Core and 8 GB of RAM as the unit. For example, an HPC job requesting 4 CPU cores and 20GB of RAM would require a minimum of 4 "seats". The RAM requirement could fit in three 8 GB seats or five 4 GB seats. Whichever of the two combinations is the lowest will be the price used as the estimated cost of the job being analyzed. We compared the cost pricing of using the most popular cloud services provides using Google Cloud Platform [25], Microsoft Azure [26], Digital Ocean [27], IBM Cloud [28] and Amazon Web Services (AWS) [29]. While we chose Holland Computing Center at the University of Nebraska-Lincoln as an example to calculate the cost for using the local resource [30].

4 Results and Discussion

4.1 Beocat

Figures 1 (a) and (b) show the total aggregate execution time and memory requested, used and predicted by our MARM algorithm on 4.5 million job logs obtained from Beocat. The requested time and memory is significantly higher than the actual and time and memory used. Our predicted time and memory are both closer to the actual configurations. **Figures 2 (a) to (f)** show the logarithm of cost distribution across six cloud platforms using requested, actual and predicted configurations of time and memory. These statistics were obtained on the 4.5 million job logs from Beocat. The cost of running services

a



b

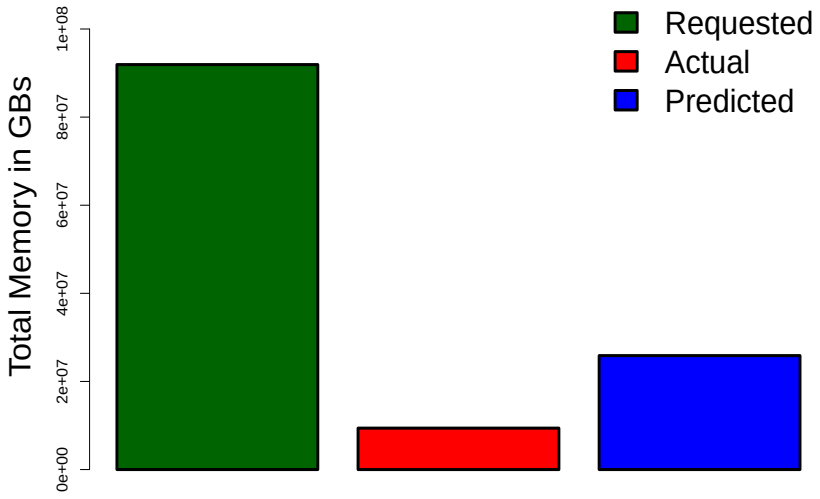


Fig. 1. (a) Total aggregate execution time requested (green), used (red), and predicted (blue) in hours and (b) Total aggregate memory requested (green), used (red), and predicted (blue) in gigabytes on Beocat HPC system across 4.5 million jobs.

with requested configuration is significantly higher than actual cost in all cases. Our MARM predicted configurations incur smaller costs that are close to the actual costs. **Figures 3** shows the mean cost distribution across various cloud platforms when using requested, actual, and predicted time and memory configurations. The cost distributions are comparable for Beocat with the Holland Computing Center costing the least amount of money and other services costing. Similarly, the cost distribution changes in magnitude, being the highest for requested configuration followed by predicted configuration, the lowest being the actual configuration. **Figure 4 and Table 1** show more cost related statistics across cloud platforms, where the cost of MARM predicted configuration is smaller than the requested configuration.

Table 1. Mean, median, 75th-quantile, and 95th-quantile cost incurred across various cloud platforms when using requested, actual, and predicted configurations of time and memory

	Requested			
	Mean	Median	75th Quantile	95th Quantile
AWS	5.31$	1.21$	1.85$	10.58$
GCP	4.91$	1.14$	1.71$	9.97$
Microsoft Azure	7.55$	1.71$	2.64$	14.96$
Digital Ocean	6.20$	1.43$	2.14$	12.50$
IBM Cloud	7.27$	1.64$	2.54$	14.39$
HCC	0.68$	0.17$	0.25$	1.47$
	Actual			
	Mean	Median	75th Quantile	95th Quantile
AWS	1.08$	0.04$	0.12$	0.77$
GCP	0.99$	0.04$	0.10$	0.70$
Microsoft Azure	1.55$	0.06$	0.17$	1.10$
Digital Ocean	1.26$	0.04$	0.13$	0.89$
IBM Cloud	1.49$	0.05$	0.16$	1.06$
HCC	0.13$	0.00$	0.01$	0.09$
	Predicted			
	Mean	Median	75th Quantile	95th Quantile
AWS	2.08$	0.25$	0.50$	2.04$
GCP	1.90$	0.24$	0.46$	1.89$
Microsoft Azure	2.97$	0.36$	0.71$	2.92$
Digital Ocean	2.42$	0.30$	0.58$	2.38$
IBM Cloud	2.86$	0.34$	0.69$	2.81$
HCC	0.25$	0.03$	0.06$	0.25$

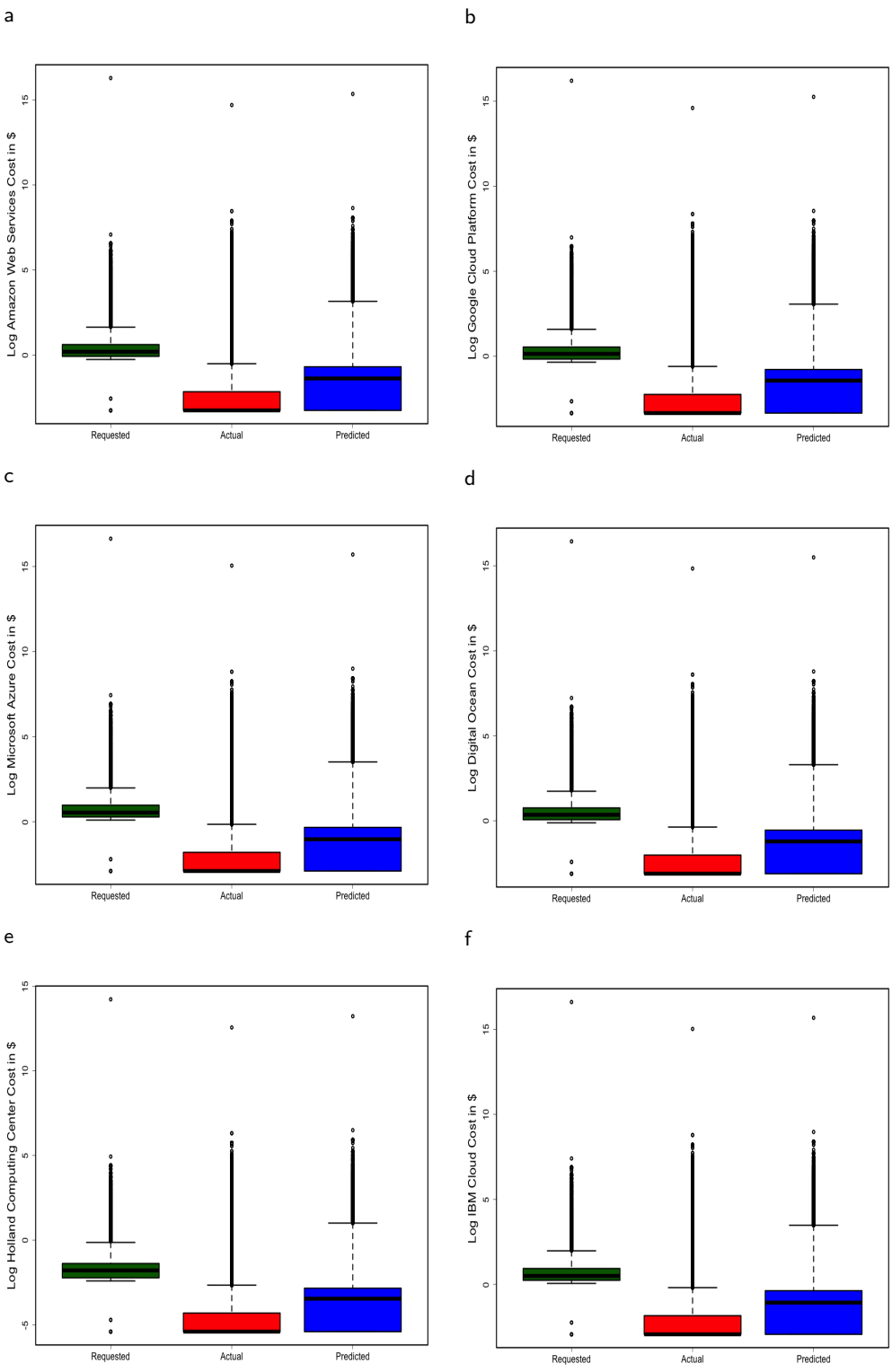


Fig. 2. Logarithm of cost distribution incurred in dollars for running 4.5 million jobs with requested (green), used (red), and predicted (blue) configurations for execution time and memory on (a) Amazon Web Services, (b) Google Cloud Platform, (c) Microsoft Azure, (d) Digital Ocean, (e) Holland Computing Center, and (f) IBM Cloud.

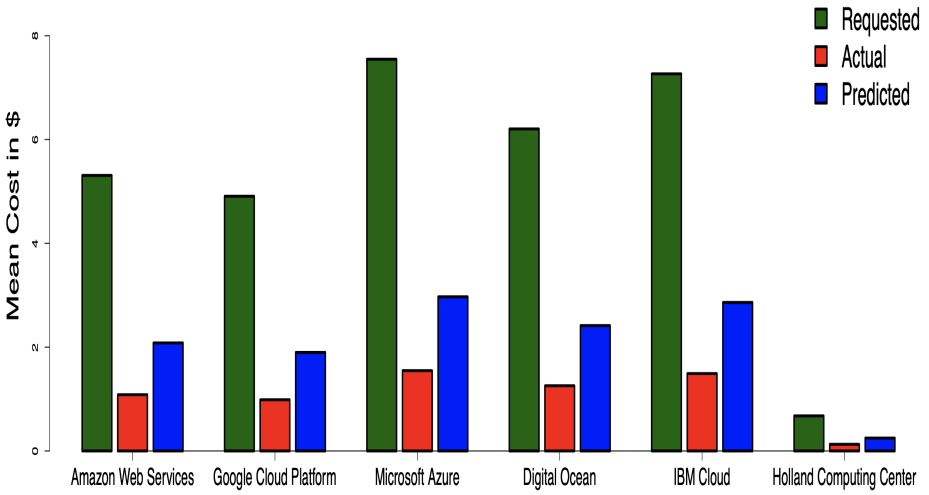


Fig. 3. Mean cost incurred in dollars for running 4.5 million jobs across various cloud platform with requested (green), used (red), and predicted (blue) configurations.

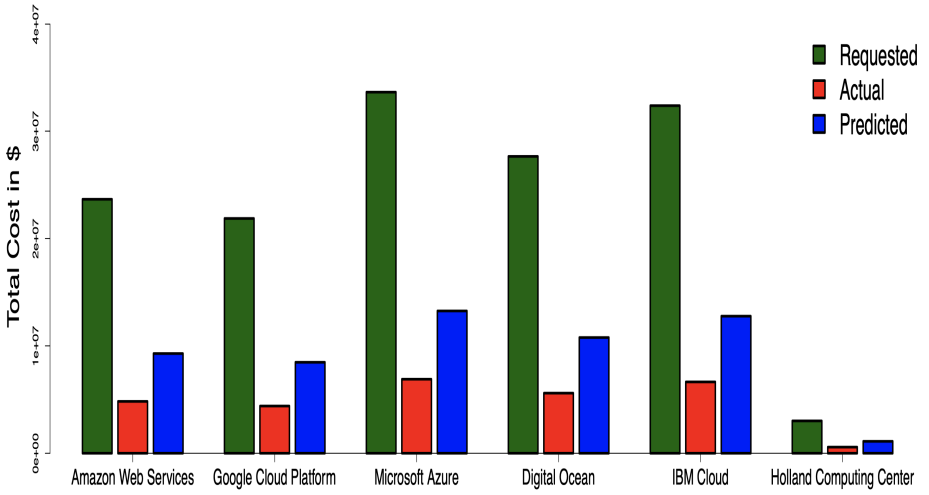


Fig. 4. Total aggregate cost incurred in dollars for running 4.5 million jobs across various cloud platform with requested (green), used (red), and predicted (blue) configurations.

5 CONCLUSIONS

High Performance Computing and cloud computing in particular have recently become more and more prominent. The performance and availability of these resources have allowed researchers to accelerate their research, using a large quantity of resources in a short period of time to meet their deadlines. Cloud computing services such as AWS, Microsoft Azure, Google cloud, and IBM Spectrum Computing have helped further increase availability at a considerable financial cost to projects and researchers. This study further verified the machine learning models, AMPRO-HPCC and MARM, to help use both financial and computational resources more efficiently by better predicting the resources required for HPC jobs, reducing the overall cost of running HPC jobs on the cloud by 39%. This potentially provides a great benefit to researchers by saving their budget on resources and allowing researchers to more efficiently reach their deadlines quicker. This study also demonstrated the financial and computational cost of the overestimation of time and memory resources by HPC users on a project or researcher's budget and the impact it can have on HPC systems.

References

1. Amazon, E.: Amazon web services. Available in: http://aws. amazon. com/es/ec2/(November 2012) p. 39 (2015)
2. Chappell, D., et al.: Introducing the azure services platform. White paper, Oct 1364(11) (2008)
3. Geewax, J.J.J.: Google Cloud Platform in Action. Simon and Schuster (2018)
4. Bernasconi, A., Beretta, C., Bernocchi, W., Richelli, G., et al.: IBM Spectrum Virtualize and IBM Spectrum Scale in an Enhanced Stretched Cluster Implementation. IBM Redbooks (2015)
5. Carlyle, A.G., Harrell, S.L., Smith, P.M.: Cost-effective hpc: The community or the cloud? In: 2010 IEEE Second International Conference on Cloud Computing Technology and Science. pp. 169–176. IEEE (2010)
6. Hu, Y., Wong, J., Iszlai, G., Litoiu, M.: Resource provisioning for cloud computing. In: Proceedings of the 2009 Conference of the Center for Advanced Studies on Collaborative Research. pp. 101–111 (2009)
7. Gong, W., Qi, L., Xu, Y.: Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment. Wireless Communications and Mobile Computing 2018 (2018)
8. Wu, L., Ding, R., Jia, Z., Li, X.: Cost-effective resource provisioning for real-time workflow in cloud. Complexity 2020 (2020)
9. Rodriguez, M.A., Buyya, R.: Deadline based resource provisioningand scheduling algorithm for scientific workflows on clouds. IEEE transactions on cloud computing 2(2), 222–235 (2014)
10. Tanash, M., Yang, H., Andresen, D., Hsu, W.: Ensemble prediction of job resources to improve system performance for slurm-based hpc systems. In: Practice and Experience in Advanced Research Computing, pp. 1–8 (2021)
11. Tanash, M., Dunn, B., Andresen, D., Hsu, W., Yang, H., Okanlawon, A.: Improving hpc system performance by predicting job resources via supervised machine learning. In: Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (learning), pp. 1–8 (2019)

12. Tanash, M., Andresen, D., Hsu, W.: Ampro-hpcc: A machine-learning tool for predicting resources on slurm hpc clusters. In: The Fifteenth International Conference on Advanced Engineering Computing and Applications in Sciences ADVCOMP. pp. 20–27 (2021)
13. Marathe, A., Harris, R., Lowenthal, D.K., De Supinski, B.R., Rountree, B., Schulz, M., Yuan, X.: A comparative study of high-performance computing on the cloud. In: Proceedings of the 22nd international symposium on High-performance parallel and distributed computing. pp. 239–250 (2013)
14. Vecchiola, C., Pandey, S., Buyya, R.: High-performance cloud computing: A view of scientific applications. In: 2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks. pp. 4–16. IEEE (2009)
15. AbdelBaky, M., Parashar, M., Kim, H., Jordan, K.E., Sachdeva, V., Sexton, J., Jamjoom, H., Shae, Z.Y., Pencheva, G., Tavakoli, R., et al.: Enabling high-performance computing as a service. Computer 45(10), 72–80 (2012)
16. Sadooghi, I., Martin, J.H., Li, T., Brandstatter, K., Maheshwari, K., de Lacerda Ruivo, T.P.P., Garzoglio, G., Timm, S., Zhao, Y., Raicu, I.: Understanding the performance and potential of cloud computing for scientific applications. IEEE Transactions on Cloud Computing 5(2), 358–371 (2015)
17. Xavier, M.G., Neves, M.V., Rossi, F.D., Ferreto, T.C., Lange, T., De Rose, C.A.: Performance evaluation of container-based virtualization for high performance computing environments. In: 2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing. pp. 233–240. IEEE (2013)
18. Jackson, K.R., Ramakrishnan, L., Muriki, K., Canon, S., Cholia, S., Shalf, J., Wasserman, H.J., Wright, N.J.: Performance analysis of high performance computing applications on the amazon web services cloud. In: 2010 IEEE second international conference on cloud computing technology and science. pp. 159–168. IEEE (2010)
19. Shi, J., Luo, J., Dong, F., Zhang, J.: A budget and deadline aware scientific workflow resource provisioning and scheduling mechanism for cloud. In: Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD). pp. 672–677. IEEE (2014)
20. Abrishami, S., Naghibzadeh, M., Epema, D.H.: Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds. Future Generation Computer Systems 29(1), 158–169 (2013)
21. Verma, A., Kaushal, S.: Cost-time efficient scheduling plan for executing workflows in the cloud. Journal of Grid Computing 13(4), 495–506 (2015)
22. Zheng, W., Qin, Y., Bugingo, E., Zhang, D., Chen, J.: Cost optimization for deadline-aware scheduling of big-data processing jobs on clouds. Future Generation Computer Systems 82, 244–255 (2018)
23. tanash1983: Tanash1983/ampro-hpcc: A machine-learning-tool for predicting job resources on hpc clusters, https://github.com/tanash1983/AMPRO-HPCC
24. Beocat. https://support.beocat.ksu.edu/BeocatDocs/index.php/Main_Page, (Accessed on 03/013/2022)
25. https://cloud.google.com/products/calculator
26. Pricing microsoft azure, https://azure.microsoft.com/en-us/pricing/details/virtual-machines/linux/, (Accessed on 02/012/2022)
27. https://www.digitalocean.com/pricing, (Accessed on 02/012/2022)
28. https://cloud.ibm.com/vpc-ext/provision/vs, (Accessed on 02/011/2022)
29. https://calculator.aws/#/createCalculator/EC2, (Accessed on 02/012/2022)
30. Priority access pricing, https://hcc.unl.edu/priority-access-pricing, (Accessed on 02/011/2022)