JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

# Determining the Duration of Cloud Workflow Task Execution using a Two-Step Machine Learning Approach

**Kalyani Sanjay Vispute1**
**Dr. D.S. Waghole2**
ME Student, Assistant Professor

**ABSTRACT: -**

Scheduling and resource provisioning are two tactics that require the ability to predict how workflow processes will respond to changes in input data. However, it is difficult to generate such approximations in the cloud. This research proposes a novel two-stage machine learning approach for predicting the execution time of cloud workflow tasks for various input data. To provide high accuracy forecasts, our method relies on two stages of prediction and parameters providing runtime information. Our method outperforms current prediction methods, as demonstrated by empirical results for four real-world workflow applications and a variety of commercial cloud providers. Comparatively, prior methods produced estimation errors of greater than 20% (and often greater than 50%) in more than 75% of the workflow tasks that were looked at. In our experiments, our technique yields 1.6% estimate errors in the best-case scenario and 12.2% estimation errors in the worst situation. We also show that the models predicted by our method for a certain cloud can be reliably and readily transferred to other clouds with a few numbers of executions.

**Keywords—:** Performance Prediction, Workflow Execution, Task Distribution

## I. Introduction

For scientific applications, pay-per-use, changeable resource quantity, and quick resource delivery are advantages of the cloud computing paradigm. Nowadays, a lot of scientists develop cloud-based applications using scientific methodologies. A significant amount of workflow applications might consist of workflow tasks, which can involve complex simulations, legacy programs, data analysis, computational techniques, and even smaller subprocesses. The fact that these elements depend on data and control flow connects them. In technical terms, a workflow can be represented as a directed graph, which is typically a directed acyclic graph, with tasks represented at the edges and dependencies on other activities or data indicated at the centers. Processing scientific workflow applications on cloud infrastructures can be expensive and time-consuming at times. Scientific processes require effective optimization of resources, costs, and runtime. Numerous strategies can be used to accomplish these goals, the most important being resource-provisioning, which establishes the kind and quantity of resources required, and scheduling, which specifies the resource where each workflow activity is to be finished. Most people on the earth are aware of frauds since credit card scams in particular have been in the news for the past few years. Due to the fact that there will be more legitimate transactions than illegal ones, the credit card dataset is highly biased. Despite some on-card transactions becoming safer as a result of banks switching to EMV cards—smart cards that store data on integrated circuits rather than magnetic stripes—the rates of card-not-present fraud have not decreased. As chip card security has increased, criminals' attention has shifted to CNP transactions, according to a 2017 [10] US Payments Forum report. In this study, we provide a novel approach for forecasting workflow task execution times using different input data. We represent these execution durations as functions that are dependent on cloud properties and workflow inputs.

These models are based on previous cloud workflow runs and are built using regression techniques. The characteristics of the kind of virtual machine (VM) used to carry out the operation are explained by cloud features. Additionally, we gather runtime data for various cloud configurations because virtual machines (VMs) may be launched on various physical servers, leading to variations in their execution durations. Our method estimates the time needed for a task to be completed on a certain virtual machine (VM) using two stages of predictions. It takes into account the VM type1, the cloud provider, and the input data for the workflow task before executing the workflow job. Our method first sets the runtime parameters for that execution. If the job has already been completed, these parameters can be accessible as historical data. If not, a regression approach based on the VM type and workflow input data will be used to estimate the runtime parameters. In the second step, the input from the first stage's results, the workflow input data, and the virtual machine information are utilized to anticipate the task's execution time using a final regression method. Regression is one of the machine learning (ML) approaches that are examined in this study. More specifically, we choose a range of machine learning techniques that have been researched for performance prediction in pertinent research. Regression trees, neural networks, bagging with regression trees, and linear regression are all included in this set. We also look at the application of random forest [3], an additional regression method whose accuracy has outperformed all others in a number of domains. As far as we are aware, workflow task predictions have not been applied to random forest. The results of our experiments demonstrate that our two-step approach works better than the most advanced prediction methods, which use only one stage to estimate the time needed to complete the task. Furthermore, our system produces prediction errors between 1.6% and 12.2% when combined with random forest, whereas the majority of the jobs across a range of workflow applications that have been evaluated by existing methods produce mistakes greater than 20%.

## II. Related Work

In order to guarantee optimal use of the resources that are available while minimizing energy consumption and fulfilling service level agreements, the study [1] emphasizes the significance of resource allocation in cloud computing environments. This paper presents many approaches to resource allocation, including as load balancing, virtual machine migration, and priority-based allocation. The writers also cover the difficulties in allocating resources in cloud computing, including the necessity to strike a balance between energy usage and performance, the variability of resources, and the dynamic nature of workloads. The resource allocation algorithms in use today are frequently rigid and unable to change to accommodate shifting user demands and workloads. In order to efficiently manage resources, they suggest a priority-based dynamic resource allocation strategy that takes into account both the jobs' priorities and the resources at hand. Three stages comprise the approach: resource allocation, resource monitoring and optimization, and priority-based job scheduling.[1]

Afzal S and Kavitha G's study [2] offers a thorough analysis of load balancing strategies used in cloud computing. In cloud computing, load balancing techniques such as global load balancing, DNS-based load balancing, network load balancing, application load balancing, virtual machine load balancing, and container load balancing are all included in the author's proposed hierarchical taxonomical taxonomy. The essay detailing the benefits and drawbacks of each. While DNS-based load balancing divides traffic across several servers using a DNS server, global load balancing routes user requests to the closest data center, ensuring reduced latency and an enhanced user experience. While application load balancing distributes traffic according to certain application needs such user location or protocol, network load balancing makes use of specialized hardware devices like load balancers or switches. The distribution of workload among several virtual machines or containers is the basis of both virtual machine load balancing and container load balancing. The authors shed light on the importance of load balancing in cloud computing, pointing out that it increases performance, decreases downtime, and optimizes resource usage. They also go over the difficulties with fault tolerance, security, and scalability that come with load balancing in cloud computing.[2]

A mathematical model is proposed in the publication [3] with the goal of optimizing the workload allocation over several servers in a cloud context. The authors then go over their suggested mathematical model, which optimizes the load balancing process by taking into account the variables of processing time, communication time, and migration time. The suggested model minimizes the makes pan, which is the amount of time needed to finish all of the jobs in the cloud, by using an optimization approach based on the particle swarm optimization (PSO) technique. The authors give a thorough examination of their suggested strategy and contrast it with tried-and-true load balancing methods like least-connection and round-robin algorithms. The findings demonstrate that the suggested strategy performs better in terms of pan and resource utilization than the current methods. The author emphasizes the importance of load balancing in cloud computing and the necessity for more study to improve the procedure. The suggested method can be applied to improve the functionality of cloud-based services and apps, and it represents a significant advancement in the field of cloud computing.[3]

A dynamic load balancing technique that distributes the workload among virtual machines in cloud computing is presented in the study [4]. A novel approach is put forth by authors Kumar M. and Sharma S. with the intention of increasing resource efficiency, decreasing reaction times, and improving cloud-based application performance as a whole. The authors next go over their suggested dynamic load balancing algorithm, which divides resources among virtual machines based on thresholds. The suggested algorithm determines a threshold value for every virtual machine by taking into account variables like CPU, RAM, and network bandwidth. The program dynamically distributes the resources to other virtual machines with lower utilization rates if the utilization beyond the threshold value. In addition to contrasting, it with other load balancing strategies including least-connection methods, weighted round-robin, and round-robin, the authors offer a thorough analysis of their suggested algorithm. The outcomes demonstrate that the suggested algorithm performs better than current methods in terms of resource usage and response time.[4]

Senthamarai N, the author, suggests a novel method for predicting workload and enhancing cloud resource consumption that makes use of machine learning techniques. The suggested migration prediction method, which employs machine learning algorithms to evaluate past data and forecast future workload patterns, is then presented by the author. The suggested method makes use of a variety of machine learning methods, including support vector machines, decision trees, and neural networks, to evaluate past data and forecast workload trends in the future. In order to anticipate the workload, the method takes into account variables like CPU and memory use as well as network bandwidth. The suggested method is contrasted with other workload balancing strategies currently in use, including least connection methods, weighted round robin, and round robin. The outcomes demonstrate that, in terms of response time and resource usage, the suggested strategy performs better than current methods.[5].

A genetic algorithm-based method for allocating virtual machines (VMs) in cloud systems is presented in the work [6]. According to the authors, existing allocation policies frequently overlook interference, which can result in performance degradation and inefficient use of resources. The writers go over a number of strategies that have been put out in the literature to handle interference in virtual machine allocation, including load balancing, dynamic allocation, and optimization-based strategies. The study suggests a genetic algorithm-based strategy known as IAGA, which modifies the fitness function the genetic algorithm uses to include interference metrics.[6]

In a cloud computing setting, the authors Jena UK, P.K. the B, and M.R. Kabat offer a load balancing technique. In cloud computing environments, the authors contend that load balancing is essential for achieving optimal resource usage and enhancing system performance. The authors draw attention to the shortcomings of current methods, including: B. Cannot handle dynamic workloads and necessitates previous understanding of the workload. This work suggests a hybrid meta-heuristic His algorithm that combines particle swarm and ant colony optimization to get over these drawbacks. The suggested approach seeks to effectively and dynamically distribute VM load. The suggested hybrid metaheuristic algorithm presents a viable way to deal with the difficulties brought on by fluctuating workloads. [7]

## III. Proposed System

The background information required to describe our process is provided in this section. In order to anticipate the execution duration of a workflow activity, we first give some background knowledge on machine learning. We then discuss the basic concepts underlying cloud computing platforms. In conclusion, we outline the four scientific processes that we employed in our research and provide a detailed explanation of one workflow application. Generally speaking, machine learning (ML) methods determine the connection between a collection of input data and an output. Usually, this link is discovered by looking at a collection of data whose input and output values are known. In the area of machine learning, this set of data is frequently referred to as a training set, or simply training data.

### Cloud-Based Software

Cloud computing is a technology that allows users to access and use computing resources (such as servers, storage, databases, networking, software, analytics, and intelligence) over the internet (the cloud) as opposed to relying on local servers or personal computers to handle their applications and data.

The following are some of the main features of cloud computing:

**On-Demand Self-Service:** Users can provision and manage computer resources as needed without the help of a service provider's human staff.

**Broad Network Access:** Cloud services are available over the network and may be accessed using standard protocols, which encourages the usage of a variety of client devices (e.g., computers, cellphones, tablets).

**Resource pooling:** This technique enables resources to be pooled to serve numerous clients by dynamically assigning and reassigning different physical and virtual resources based on demand. Customers are typically not in charge of or aware of the exact placement of the resources.

Rapid Elasticity: Resources can be released fast to reduce in size as well as provisioned swiftly and elastically to scale up and down. This makes it possible to adjust to varying workloads.

Measured Service: Cloud systems can automatically control and optimize resource usage by leveraging a metering feature at some abstraction level. Monitoring, controlling, and reporting on resource utilization leads to transparency for both the consumer and the provider.

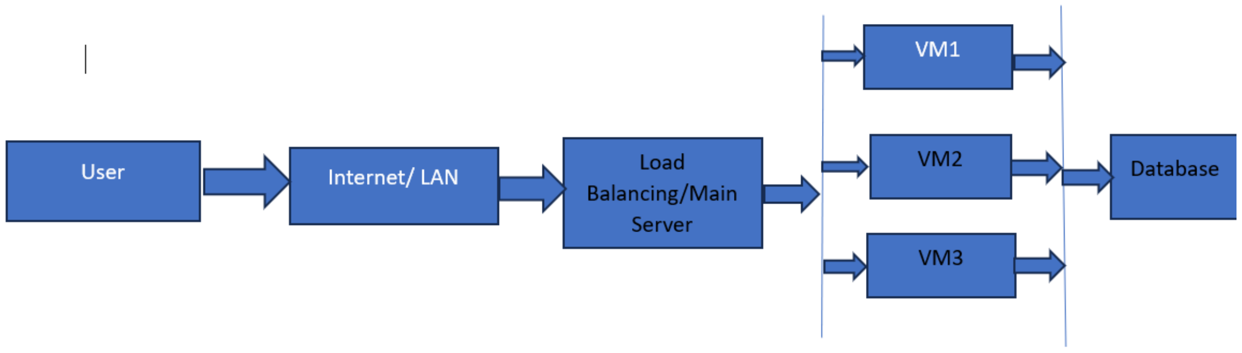

Figure 1 Proposed System

Deep learning, a branch of artificial intelligence, is similar to the human brain in that both are made up of many layers of neurons. The network as it is presently consisting of an input layer, an output layer, and one or more hidden layers. As the network attempts to learn from the data that is fed into it, predictions are produced.

The operation of a computer system with many moving parts that are prone to modification and adaptation over time is depicted in the accompanying diagram. These parts—referred to as virtual computers—are a part of the cloud, which is a larger system.

Log: The program log keeps track of how much time the CPU and input/output (IO) of a computer are utilized. This data is saved for public viewing in a shared folder.

Dump: When virtual machines are transferred from one computer to another, a copy of their memory is stored in an accessible folder.

Conf: The system's ability to function depends on the files in this folder.

Private: Only the virtual machine has access to the files in this folder, including its own hard drive. The ANN is the most fundamental kind of neural network.

The ANN simply consists of several neural layers to be utilized for prognosis; it has no unique structure.

1. In the first step, input units—data with weights attached—are passed to the hidden layer. It could have multiple hidden layers.

2. Every hidden layer is composed of neurons. Every input is connected to every neuron.

3. After the inputs are transferred, the hidden layer does all of the calculations. There are two stages to the computation of hidden layers. First, the weights allocated to the inputs are multiplied. Each variable has a gradient or coefficient that is matched to its weight. It illustrates the strength of a certain stimulus. After weights are assigned, a bias variable is added.

Bias guarantees that the model is both viable and matches the data.

$$z1 = w1 \times In1 + w2 \times In2 + w3 \times In3 + w4 \times In4 + w5 \times In5 + b\ldots\ldots\ldots (2)$$

Weights are assigned to inputs In1, In2, In3, In4, and In5, and the bias is denoted by b. In the next phase, the activation function is applied to linear equation Z1. The activation function is a nonlinear modification that information goes through before it is transferred to the next layer of neurons. For the model to be nonlinear, the activation function is essential.

4. Every hidden layer follows the entire process described in the third phase. After going through each hidden layer, the system advances to the top layer, which is called the output layer and produces the final output. The process described above is called forwarding or propagation.

5. After obtaining forecasts from the output layer, the error—that is, the difference between the actual and expected output—is determined.

## Conclusion

The paper recommends adopting the SSUR technique to maximize virtual machine allocation in cloud data centers while accounting for service response time, energy consumption, and resource utilization based on user needs. The results of the experiment show that by better balancing user demands and resource utilization in the data center, the recommended strategy performs better than traditional methods. Additionally, by reducing energy consumption and carbon emissions in cloud data centers, the SSUR approach can improve sustainability and efficiency. This study concludes with a possible strategy for meeting customer expectations while improving the efficiency and sustainability of cloud data centers.

## References

[1] J. Qin and T. Fahringer, Scientific Workflow: Programming, Optimization, and Synthesis with ASKALON and AWDL. Springer, 2012.

[2] J. Durillo and R. Prodan, “Multi-objective workflow scheduling in

amazon ec2,” Cluster Computing, vol. 17, no. 2, pp. 169–189, 2014.

[3] L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001. [Online]. Available: http://dx.doi.org/10.1023/ A%3A1010933404324

[4] K. L. Spafford and J. S. Vetter, “Aspen: A domain specific language for performance modeling,” in Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, ser. SC ’12. Los Alamitos, CA, USA: IEEE Computer Society Press, 2012, pp. 84:1–84:11. [Online]. Available: http://dl.acm.org/citation.cfm?id=2388996.2389110

[5] S. Seneviratne and D. C. Levy, “Task profiling model for load profile prediction,” Future Generation Computer Systems, vol. 27, no. 3, pp. 245–255, 2011. [Online]. Available: http://www. sciencedirect.com/science/article/pii/S0167739X10001743

[6] B.-D. Lee and J. M. Schopf, “Run-time prediction of parallel applications on shared environments,” in Cluster Computing, 2003. Proceedings. 2003 IEEE International Conference on, Dec 2003, pp. 487–491.

[7] N. Kapadia, J. Fortes, and C. Brodley, “Predictive applicationperformance modeling in a computational grid environment,” in High Performance Distributed Computing, 1999. Proceedings. The Eighth International Symposium on, 1999, pp. 47–54.

[8] H. Li, D. Groep, and L. Wolters, “An evaluation of learning and heuristic techniques for application run time predictions,” in Proceedings of 11 th Annual Conference of the Advance School for Computing and Imaging (ASCI), 2005.

[9] T. Miu and P. Missier, “Predicting the execution time of workflow activities based on their input features,” in Proceedings of the 2012 SC Companion: High Performance Computing, Networking Storage and Analysis, ser. SCC ’12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 64–72. [Online]. Available: http://dx.doi.org/10.1109/SC.Companion.2012.21

[10] R. F. da Silva, G. Juve, M. Rynge, E. Deelman, and M. Livny, "Online task resource consumption prediction for scientific workflows," Parallel Processing Letters, vol. 25, no. 03, p. 1541003, 2015. [Online]. Available: http://www.worldscientific.com/doi/ abs/10.1142/S0129626415410030

[11] A. Matsunaga and J. A. B. Fortes, "On the use of machine learning to predict the time and resources consumed by applications," in Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, ser. CCGRID '10. Washington,

DC, USA: IEEE Computer Society, 2010, pp. 495–504. [Online]. Available: http://dx.doi.org/10.1109/CCGRID.2010.98

[12] W. Smith, I. Foster, and V. Taylor, "Predicting application run times with historical information," J. Parallel Distrib. Comput., vol. 64, no. 9, pp. 1007–1016, Sep. 2004. [Online]. Available: http://dx.doi.org/10.1016/j.jpdc.2004.06.008

[13] P. Dinda, "Online prediction of the running time of tasks," in High Performance Distributed Computing, 2001. Proceedings. 10th IEEE International Symposium on, 2001, pp. 383–394.

[14] P. Dinda and D. O'Hallaron, "An evaluation of linear models for host load prediction," in High Performance Distributed Computing, 1999. Proceedings. The Eighth International Symposium on, 1999, pp. 87–96.

[15] A. M. Chirkin and S. V. Kovalchuk, "Towards better workflow execution time estimation," IERI Procedia, vol. 10, pp. 216 – 223, 2014. [Online]. Available: http://www.sciencedirect.com/ science/article/pii/S2212667814001282

[16] I. Pietri, G. Juve, E. Deelman, and R. Sakellariou, "A performance model to estimate execution time of scientific workflows on the cloud," in Proceedings of the 9th Workshop on Workflows in Support of Large-Scale Science, ser. WORKS '14. Piscataway,NJ, USA: IEEE Press, 2014, pp. 11–19. [Online]. Available: http://dx.doi.org/10.1109/WORKS.2014.12

[17] D. A. Monge, M. Holec, F. Zelezn ˇ y, and C. Garino, "Ensemble ´ learning of runtime prediction models for gene-expression analysis workflows," Cluster Computing, vol. 18, no. 4, pp.

1317–1329, 2015. [Online]. Available: http://dx.doi.org/10.1007/ s10586-015-0481-5

[18] S. Lee, J. S. Meredith, and J. S. Vetter, "Compass: A framework for automated performance modeling and prediction," in Proceedings of the 29th ACM on International Conference on Supercomputing, ser. ICS '15. New York, NY, USA: ACM, 2015, pp. 405–414. [Online].

Available: http://doi.acm.org/10.1145/2751205.2751220

[19] N. R. Tallent and A. Hoisie, "Palm: Easing the burden of analytical performance modeling," in Proceedings of the 28th ACM International Conference on Supercomputing, ser. ICS '14. New

York, NY, USA: ACM, 2014, pp. 221–230. [Online]. Available: http://doi.acm.org/10.1145/2597652.2597683

[20] A. Bhattacharyya and T. Hoefler, “Pemogen: Automatic adaptive performance modeling during program runtime,” in Proceedings of the 23rd International Conference on Parallel Architectures and Compilation, ser. PACT ’14. New York, NY, USA: ACM, 2014, pp. 393–404. [Online]. Available: http://doi.acm.org/10.1145/ 2628071.2628100

[21] A. Li, X. Zong, S. Kandula, X. Yang, and M. Zhang, “Cloudprophet: Towards application performance prediction in cloud,” SIGCOMM Comput. Commun. Rev., vol. 41, no. 4, pp. 426–427, Aug. 2011. [Online]. Available: http://doi.acm.org/10. 1145/2043164.2018502

[22] I. H. Witten, E. Frank, and M. A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.

[23] S. Salzberg, “C4.5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993,” Machine Learning, vol. 16, no. 3, pp. 235–240, 1994. [Online]. Available:

http://dx.doi.org/10.1007/BF00993309

[24] J. W. Shavlik, R. J. Mooney, and G. G. Towell, “Symbolic and neural learning algorithms: An experimental comparison,” Machine learning, vol. 6, no. 2, pp. 111–143, 1991.

[25] T. M. Mitchell, Machine Learning, 1st ed. New York, NY, USA:

McGraw-Hill, Inc., 1997.

[26] H. Trevor, T. Robert, and F. Jerome, The Elements of Statistical

Learning, 2nd ed. Springer, 2009.

[27] L. Breiman, “Bagging predictors,” Machine Learning, vol. 24,

no. 2, pp. 123–140, 1996. [Online]. Available: http://dx.doi.org/

10.1007/BF00058655

[28] T. G. Dietterich, “Ensemble methods in machine learning,” in Proceedings of the First International Workshop on Multiple Classifier Systems, ser. MCS ’00. London, UK, UK: Springer-Verlag, 2000, pp. 1–15. [Online]. Available: http://dl.acm.org/citation.cfm?id= 648054.743935