Implementation of Fuzzy Inference System with Best-Worst Method for Cost Efficiency on Amazon Web Services

Annisya Dira Prastiwi [1*], Anggyi Trisnawan Putra [1]

[1] Department of Computer Science, Faculty of Mathematics and Natural Sciences, Semarang State University, Semarang, Indonesia
*Corresponding author: annisyadp7@students.unnes.ac.id

ARTICLE INFO

Article history
Received 13 October 2022
Revised 21 October 2022
Accepted 9 November 2022

Keywords
Multi Criteria Decision Making
Cloud Computing
Fuzzy Inference System
Best-Worst Method

ABSTRACT

This study aims to reduce the cost of using computing services on AWS. Cost reduction is needed because there is a possibility that the total cost of using cloud services exceeds the estimated budget. One type of EC2 that offers a large discount is the Spot Instance. The downside of this type of EC2 is that AWS reserves the right to stop it at any time. The proposed solution is an automation system to select and run EC2 Spot Instance types based on price, discount, amount of memory, and vCPU usage from previous instances. The automation system is built with the implementation of fuzzy inference system and Best-Worst Method (BWM). All input data is obtained using the Boto3 SDK. System deployment is done in Lambda functions. This Lambda function is automatically executed whenever a Spot Instance is terminated by AWS. The EventBridge service will catch the event and then trigger the Lambda to run. System testing was run for four days with event simulation using the Send Events feature. From these tests it is known that the automation system can select the appropriate instance and generate a total cost of $3.85 (USD). After calculating the total cost with regular EC2 estimation (On Demand), the cost is reduced by 71.28%. This number proved to be 4.28% greater than previous similar studies.

1 Introduction

Virtualization is a process of utilizing hardware more efficiently. The concept of virtualization is the basis of cloud computing technology (Rashid & Chaturvedi, 2019). One of the leading cloud service providers in the world is Amazon Web Services (AWS). AWS services that are dedicated to performing compute workloads are called Elastic Compute Cloud (EC2). Instances on EC2 are virtual servers for running applications that can be rented and configured by users. AWS has 3 rental options on EC2 services namely On Demand Instance, Spot Instance, and Reserved Instance (Choudhary, 2021). On Demand Instances are regular instances that are paid per second of use with no long-term commitment. Spot Instances are surplus/backup/remaining instances of an AWS data center that are leased to users. As for Reserved Instances, users must commit to using the instance for one to three years to get a discount.

One of the goals of cloud computing services is to reduce costs. Although the cloud has a pay-as-you-go tendency or pay according to the resources used, the total cost of use may exceed the estimated budget (Alalawi & Al-Omary, 2020). AWS has several cost-saving mechanisms, one of which is Spot Instances which can provide discounts of up to 90% but there is a limitation that Amazon reserves the right to preempt Spot Instances at any time (López García et al., 2019).

Surplus VM tends to be a computing solution for some people because the cost is much cheaper than VM On Demand. In the research by Haugerud et al, (2020), an automation system has been developed using Google Cloud Platform (GCP) services that can shift workloads from Preemptible VMs to another. There are various methods and algorithms that can be used to build similar

systems, such as the Greedy Algorithm (Haugerud et al., 2020), Neural Network (Domanal & Reddy, 2018), Modified Particle Swarm Optimization (Jana et al., 2019), Ant Colony Optimization (Tawfeek et al., 2015), Fuzzy Dominance Sort Based HEFT (Zhou et al., 2019), and Fuzzy Logic Controller (Oikonomou et al., 2020) which are capable of selecting cloud resources in automation systems.

Fuzzy logic is a multi-criteria decision-making system based on expert knowledge (Abbaspour-Gilandeh & Abbaspour-Gilandeh, 2019). This method is often used to deal with the concept of partial truth, whose truth value is between 1 (true) and 0 (false). With fuzzy logic, output with precise and detailed values can be generated from crisp input values that are not clear and inaccurate (fuzzy). The crisp input is carried out by the process of fuzzification, application of rules, and defuzzification. Fuzzy logic requires variables as a consideration to produce output but there is no level of importance assigned to these variables. To give importance to each variable, the Best-Worst Method (BWM) can be used.

BWM is a multi-criteria decision-making method (MCDM) proposed by Dr. Jafar Rezaei in 2015. The BWM can be applied to problems where flexibility is expected (non-linear) or not (linear). Comparison of BWM against other multi-criteria decision-making methods such as the Analytic Hierarchy Process (AHP) shows that BWM requires fewer comparisons, produces more consistent final weights, can be combined with other multi-criteria decision-making methods, and is easier to use because it only contains integer data required in the comparison matrix (Rezaei, 2015).

2 The Proposed Method/Algorithm

2.1 Spot Instance Selection Automation System

This research focuses on how to maximize the use of surplus resources on Amazon Web Services and at the same time reduce the cost of using AWS computing services. The proposed solution is an automation system to select and run EC2 Spot Instance types based on price, discount, amount of memory, and vCPU usage from the previous running instance. The Spot Instance Selection Automation System is built with the implementation of fuzzy inference system for the first selection stage and BWM for the second and is deployed in Lambda function. With the help of the EventBridge service, Lambda function can be triggered automatically whenever AWS sends the termination signal. The system uses input data from both Boto3 method calls and Amazon CloudWatch metrics.

3 Method

3.1 Fuzzy Inference System

Fuzzy inference system with the Sugeno method is used to build the first selection stage in the Spot Instance selection system. This method allows for the members of the set of consequent variables in the form of constants or equations. This first stage serves to select EC2 Spot instances based on vCPU usage in the last five minutes. There are two variables that affect the number of vCPUs to be selected, these variables are divided into two, namely premise variables consisting of Current vCPU and vCPU Usage. Meanwhile, the consequent variable is vCPU Verdict. table 1 shows detailed sets on each fuzzy variable.

Table 1. Fuzzy sets on each variable.

Variable	Set	Description
Current vCPU	{2, 4, 8}	The number of vCPUs available are 2, 4, and 8.
vCPU Usage	{low, fair, high}	low = less than or equal to 50%; fair = between 25% and 75%; high = more than or equal to 50%.

vCPU Verdict	{lower, constant, higher}	lower = half of Current vCPU;
		constant = number of Current vCPU;
		higher = twice Current vCPU.

The Sugeno method is defined by the IF-THEN rule in (1)

$$\text{Rule} \quad R_i : IF\ x_1\ is\ G_1^i,\ x_2\ is\ G_2^i,\ \ldots,\ x_m\ is\ G_m^i \tag{1}$$

$$THEN\ y\ is\ f^i(x_1,\ x_2,\ \ldots, x_m),\ i = 1,\ 2,\ \ldots,\ n$$

With R denotes the i-th rule. Hi and Gji, where j = 1, 2, …, m are fuzzy sets. n is the sequence number of the fuzzy rule. The function fi (.) is usually a linear function as shown in (2) (Sugeno's rule of order 1), or it can be a constant (Sugeno's rule of order 0) (Nguyen et al., 2019).

$$f^i\left(x_1,\ x_2,\ \ldots, x_m\right) = b^i + a_1^i x_1 + a_2^i x_2 + \ldots + a_m^i x_m \tag{2}$$

The Minimum (Min) Function is used for each rule in the inference stage. And use the Maximum Function (Max) in the composition between the rules. The defuzzification process is carried out after the inference process is formed (Selvaraj et al., 2020).

Fuzzy rules are created based on changes in the number of vCPUs. There are two premise variables each with three fuzzy set members, therefore the total number of possible fuzzy rules is 32 = 9 as shown in table 2.

Table 2. Fuzzy rules.

Rule	Current vCPU	vCPU Usage	vCPU Verdict
R1	2	Low	Constant
R2	2	Fair	Constant
R3	2	high	Higher
R4	4	Low	Lower
R5	4	Fair	Constant
R6	4	high	Higher
R7	8	Low	Lower
R8	8	Fair	Constant
R9	8	high	Constant

The result of defuzzification (z value) is shown in table 3 using the input value example of the variable Current vCPU and vCPU Usage.

Table 3. Fuzzy Inference System Calculation Results.

Current vCPU input	vCPU Usage input	Z value
2	0	2,0
	25	2,0
	50	2,0
	75	2,66

	100	2,66
	0	3,33
	25	3,33
4	50	4,0
	75	5,33
	100	5,33
	0	6,66
	25	6,66
8	50	8,0
	75	8,0
	100	8,0

It seen from table 3 that the z value is not a number that can be directly processed for the Spot Instance selection stage because the number of vCPUs available is only 2, 4, and 8. The adjustment process is carried out by applying a subtraction operation with the input data of the number of vCPUs (Current vCPU) on the output value. This operation produces a value that can be used as a reference in the selection of Spot Instances whether or not to reduce or increase the number of vCPUs.

3.2 Best-Worst Method (BWM)

In this study, BWM is used as a method for selecting Spot Instances on AWS based on the amount of memory. The selection using BWM is the second stage of selection after the selection with a fuzzy inference system which if the results are two. The BWM model implemented is a linear model. Using this model, one global optimal solution is obtained.

There are five steps in the BWM method (Kheybari et al., 2019). It starts by defining the set of criteria. Furthermore, from these criteria the best and worst criteria are determined. Followed by determining the level of importance between the best criteria and all criteria (Best-to-Others). The fourth step is to determine the importance of all criteria with the worst criteria (Others-to-Worst). Finally, calculate the optimal weight with the min-max model.

Three criteria are used: total memory, lowest price, and highest discount. The Total Memory criterion is used because the purpose of implementing BWM is to select a Spot Instance based on memory. While the other two criterions are taken based on the objective of this study: price efficiency.

In its implementation, BWM requires data from experts. Experts in this study were described as people who passed the cloud certification exam at least professional level in the past two years. This data is used to determine the best and worst criteria as well as the Best-to-Others (BO) and Others-to-Worst (OW) scores. From these data, it can be determined that the best criteria is total memory, the worst is highest discount, the Best-to-Others vector is BO = (1, 6, 5), and the Others-to-Worst vector is OW = (8, 6, 1). The optimal weight value for the total memory criterion is 0.74, the lowest price criterion is 0.18, and the highest discount criterion is 0.08. This calculation also calculates a reliability level of 0.32 (a value close to 0 indicates that it is more reliable, while a value close to 1 less reliable).

3.3 Integration of AWS Services

AWS service integration begins by manually setting up the required AWS resources in the ap-southeast-3 (Jakarta) region via console such as EC2 Spot Instance, CloudWatch, EventBridge, and Lambda. The Jakarta region was chosen because the region is the closest to the research location. In EC2 Spot Instance, the Stress-ng tool will be installed. This tool serves to dynamically manipulate the workload on the vCPU instance. The AWS architecture is depicted in figure 1.

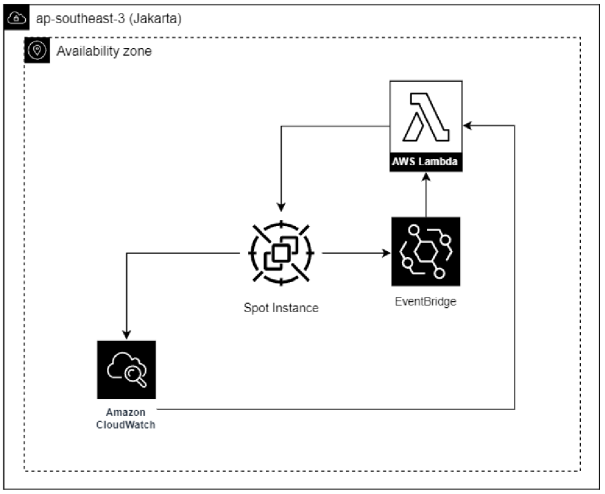

Figure 1. AWS Architecture.

Spot Instances can be terminated by AWS at any time for various reasons. To overcome this, the researcher uses the EventBridge service to capture the termination signal and then trigger Lambda to run the Spot Instance selection system. Amazon CloudWatch is used to monitor vCPU usage of Spot Instances. This data will later be used as input data for the Spot Instance selection automation system. And since the experiment will only run for four days, it is not effective to only rely on termination signals from AWS which the researchers themselves do not know when the signal will be sent. So, to prevent the possibility of a termination signal not being sent by AWS during the experiment, the researchers conducted a system test using the Send Events feature in EventBridge.

4 Results and Discussion

Researchers created and sent a termination event simulation six times at different times. The total usage cost of Spot Instance can be seen on the AWS cost management service console page using the cost explorer feature. Through this feature, researchers can find out the total usage cost of Spot Instance by setting the "resource" filter and adding six instance ids from the instances selected. table 4 shows a breakdown of the cost of using Spot Instances.

Table 4. The total usage cost of Spot Instance.

No	Day	Cost
1	1st	$0,084000
2	2nd	$0,084000
3	3rd	$1,895819
4	4th	$0,223436
5	5th	$0,070000
Total		$3,85659

The next step after getting the total cost of using Spot Instance is to calculate the range of total cost of using EC2 On Demand with adjusted uptime and instance type. For this stage the researcher uses the AWS Pricing Calculator. The estimates provided by the service are monthly estimates. Therefore, it is necessary to divide the estimation by 30 so that the resulting figure is a daily estimate. table 5 shows the results of the estimated cost of using EC2 On Demand.

Table 5. The estimation usage cost of EC2 On Demand.

No	Instance Type	Monthly Estimation	Daily Estimation
1	t3.small	$15	$0,5
2	t3.xlarge	$80,58	$2,686
3	t3.2xlarge	$148,38	$4,946
4	t3.xlarge	$129,16	$4,3053
5	t3.small	$18,91	$0,6303
6	t3.small	$10,94	$0,3646
Total		$402,97	$13,4316

The results of the calculation show that the discounted price obtained from the use of EC2 Spot Instances is 71.28% of the cost of using EC2 On Demand. This study also proves that the application of a fuzzy inference system with the BWM method can automate the launch of new Spot Instances according to vCPU usage when the old Spot Instances experience termination from AWS so that instances are always available.

5 Conclusion

The topic of this research is the cost efficiency of using Amazon Web Services (AWS) computing services. There are two methods used, namely fuzzy inference system and Best-Worst Method (BWM). The fuzzy inference system is implemented using the Sugeno method. The first step the researcher did was to implement a fuzzy inference system using the Sugeno method for the selection of stage one in the AWS Spot Instance Selection automation system. The implementation includes determining variables, fuzzy sets, membership functions, and fuzzy rules, as well as applying the Sugeno method in fuzzy inference systems and also the defuzzification process that produces z values. Next, the researcher applies the BWM for the second stage selection on instances that have two vCPUs. BWM uses data from experts to determine which criteria are the best and worst and determine the values of the Best-to-Others (BO) and Others-to-Worst (OW) vectors. From the resulting optimal weight, it can produce an output in the form of a Spot Instance decision with calculations using data from the performance matrix. The third stage is the integration of several AWS services including EventBridge, Lambda, CloudWatch, and EC2.

During the 4 four days the system was running, the simulation signal was sent six times and the system succeeded in requesting a new Spot Instance with specifications that matched the input data. Based on the researcher's evaluation, it was found that the reduced cost of AWS computing services was 71.28% of the EC2 On Demand service.

6 References

Abbaspour-Gilandeh, M., & Abbaspour-Gilandeh, Y. (2019). Modelling soil compaction of agricultural soils using fuzzy logic approach and adaptive neuro-fuzzy inference system (ANFIS) approaches. *Modeling Earth Systems and Environment*, *5*(1), 13–20. https://doi.org/10.1007/s40808-018-0514-1

Alalawi, A., & Al-Omary, A. (2020). Cloud Computing Resources: Survey of Advantage, Disadvantages and Pricing. *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy, ICDABI 2020*. https://doi.org/10.1109/ICDABI51230.2020.9325645

Choudhary, A. (2021). A walkthrough of Amazon Elastic Compute Cloud (Amazon EC2): A Review. *International Journal for Research in Applied Science and Engineering Technology*, *9*(11), 93–97. https://doi.org/10.22214/ijraset.2021.38764

Domanal, S. G., & Reddy, G. R. M. (2018). An efficient cost optimized scheduling for spot

instances in heterogeneous cloud environment. *Future Generation Computer Systems*, *84*, 11–21. https://doi.org/10.1016/j.future.2018.02.003

Haugerud, H., Krüger Svensson, J., & Yazidi, A. (2020). Autonomous Provisioning of Preemptive Instances in Google Cloud for Maximum Performance Per Dollar. *Proceedings of 2020 5th International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications, CloudTech 2020*. https://doi.org/10.1109/CloudTech49835.2020.9365879

Jana, B., Chakraborty, M., & Mandal, T. (2019). A task scheduling technique based on particle swarm optimization algorithm in cloud environment. In *Advances in Intelligent Systems and Computing* (Vol. 742). Springer Singapore. https://doi.org/10.1007/978-981-13-0589-4_49

Kheybari, S., Kazemi, M., & Rezaei, J. (2019). Bioethanol facility location selection using best-worst method. *Applied Energy*, *242*(November 2018), 612–623. https://doi.org/10.1016/j.apenergy.2019.03.054

López García, Á., Fernández del Castillo, E., & Campos Plasencia, I. (2019). An efficient cloud scheduler design supporting preemptible instances. *Future Generation Computer Systems*, *95*, 68–78. https://doi.org/10.1016/j.future.2018.12.057

Nguyen, A. T., Taniguchi, T., Eciolaza, L., Campos, V., Palhares, R., & Sugeno, M. (2019). Fuzzy control systems: Past, present and future. *IEEE Computational Intelligence Magazine*, *14*(1), 56–68. https://doi.org/10.1109/MCI.2018.2881644

Oikonomou, P., Kolomvatsos, K., Tziritas, N., Theodoropoulos, G., Loukopoulos, T., & Stamoulis, G. (2020). A Fuzzy Logic Controller for Tasks Scheduling Using Unreliable Cloud Resources. *2020 IEEE 19th International Symposium on Network Computing and Applications, NCA 2020*. https://doi.org/10.1109/NCA51143.2020.9306707

Rashid, A., & Chaturvedi, A. (2019). Virtualization and its Role in Cloud Computing Environment. *International Journal of Computer Sciences and Engineering*, *7*(4), 1131–1136. https://doi.org/10.26438/ijcse/v7i4.11311136

Rezaei, J. (2015). Best-worst multi-criteria decision-making method. *Omega (United Kingdom)*, *53*, 49–57. https://doi.org/10.1016/j.omega.2014.11.009

Selvaraj, A., Saravanan, S., & Jennifer, J. J. (2020). Mamdani fuzzy based decision support system for prediction of groundwater quality: an application of soft computing in water resources. *Environmental Science and Pollution Research*, *27*(20), 25535–25552. https://doi.org/10.1007/s11356-020-08803-3

Tawfeek, M., El-Sisi, A., Keshk, A., & Torkey, F. (2015). Cloud task scheduling based on ant colony optimization. *International Arab Journal of Information Technology*, *12*(2), 129–137. https://doi.org/10.1109/ICCES.2013.6707172

Zhou, X., Zhang, G., Sun, J., Zhou, J., Wei, T., & Hu, S. (2019). Minimizing cost and makespan for workflow scheduling in cloud using fuzzy dominance sort based HEFT. *Future Generation Computer Systems*, *93*, 278–289. https://doi.org/10.1016/j.future.2018.10.046