

A FUZZY MATHEMATICAL MODEL FOR PERFORMANCE TESTING IN CLOUD COMPUTING USING USER DEFINED PARAMETERS

A.Vanitha Katherine⁽¹⁾ and K.Alagarsamy⁽²⁾

¹Department of Master of Computer Applications, PSNA College of Engineering and Technology, Dindigul.
Email: avanitha@yahoo.com

²Computer Centre, Madurai Kamaraj University, Madurai.
Email: alagarsamymku@gmail.com

ABSTRACT

Software product development life cycle has software testing as an integral part. Conventional testing requires dedicated infrastructure and resources that are expensive and only used sporadically. In the growing complexity of business applications it is harder to build in-house testing facilities and also to maintain that mimic real-time environments. By nature, cloud computing provides resources which are unlimited in nature along with flexibility, scalability and availability of distributed testing environment, thus it has opened up new opportunities for software testing. It leads to cost-effective solutions by reducing the execution time of large application testing. As a part of infrastructure resource, cloud testing can attain its efficiency by taking care of the parameters like network traffic, Disk Storage and RAM speed. In this paper we propose a new fuzzy mathematical model to attain better scope for the above parameters.

KEYWORDS:

Cloud, disk storage, Fuzzy, Network Traffic, RAM.

1. INTRODUCTION

In Software product development software testing has become an integral part. To improve performance, consistency and other important factors testing is important in life cycle of software development. Software bugs could be dangerous and expensive, thus testing is important. Resources and dedicated infrastructure is expensive in testing thus scrutinizing the application's speed, reliability, performance, functionality and security[1] were used sporadically. Testing opportunity's new vistas is opened by cloud computing. A combination of pay peruse, lower cost and upfront capital expenditure elimination[2] is offered by cloud testing than that of conventional test environment. At the same time, new set of challenges are introduced by cloud testing especially in public cloud which includes lack of standards and data security [8]. A traditional test environment restricts Time-to-market[3] since it can be delay prone and time consuming. Single application can take weeks or months which leads to delay release. Forward thinking companies due to on-demand provisioning starts the process[18], since cloud can provide instantaneously time-to-market demands. The test environment is scaled up in minutes so that the testers utilize the virtualized infrastructure to run existing applications and virtual machines with no or minimal rewriting. The performance of testing in cloud [10] depends on the efficiency of computing in the cloud which includes factors like network traffic, Disk storage and RAM. When the network traffic is low the performance of testing is high, similarly when the disk

storage and RAM is high the performance increases [11]. Thus the goal is to use a mathematical fuzzy model to increase the performance of testing in cloud by handling the user defined parameters network traffic, disk storage and RAM in the cloud server.

Rest of the paper is organized as follows Section 2 presents an algorithm in fuzzy model. Section 3 depicts the application of the model and Section 4 provides a structural view of the results generated. The major contribution of this paper is to produce a fuzzy mathematical model for improving performance of cloud testing.

2. FUZZY MATHEMATICAL MODEL

Fuzzy logic depends upon the context that cannot be defined precisely and is a form of knowledge representation suitable for notions. It is a Problem-solving control system methodology. We move to fuzzy due to the following reasons. Conceptually fuzzy logic is easy to understand[22], Fuzzy reasoning mathematical concepts are simple. Fuzzy logic does not need any far-reaching approach, it in an intuitive approach. It need not start any thing from scratch and flexible with any system and easily layer on more functionality. Even on careful inspection most things are imprecise when looked closely but fuzzy logic tolerates on imprecise data[23]. Rather than tacking it onto the end fuzzy reasoning builds this understanding into the process. Fuzzy system can be created to match any set of input-output data, nonlinear functions of arbitrary complexity can be modeled in fuzzy logic. Fuzzy logic toolbox software[24] has a technique like Adaptive Neuro-Fuzzy Inference Systems(ANFIS), which is an adaptive technique which ease the above process. Neural networks is in direct contrast with fuzzy logic which generates opaque data taking training data as input with impenetrable models, fuzzy logic rely on the experience of people who already understand the system thus built on top of the experienced experts. A fuzzy system does not replace conventional control methods in turn blend with the conventional control techniques. It simplifies their implementation and augments them in many cases. Fuzzy logic is the basis for human communication thus based on natural language[23]. Many other statements about fuzzy logic are underpinned by this observation, because it is easy to use and built on the structures of qualitative description used in everyday language. A fuzzy logic system (FLS) is defined as the nonlinear mapping of an input data set to a scalar output data[4]. A FLS as in figure 2.1 consists of four main parts: fuzzifier, rules, inference engine, and defuzzifier.

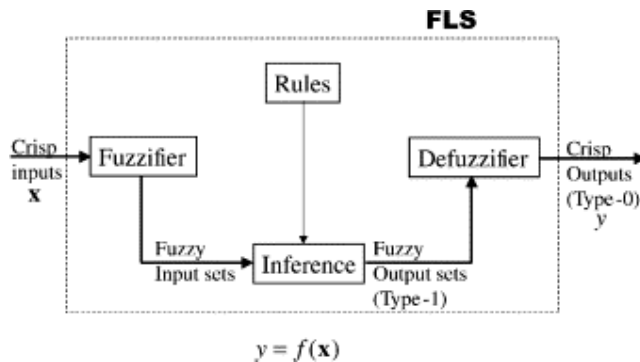


Figure 2.1 - A Fuzzy Logic System

To capture expert knowledge mamdani fuzzy additive rule model is used. It allows us to describe the expertise in more human-like and intuitive manner. However, Mamdani-type fuzzy inference entails a substantial computational burden[5]. Figure 2.2 is a model pattern for mamdani.

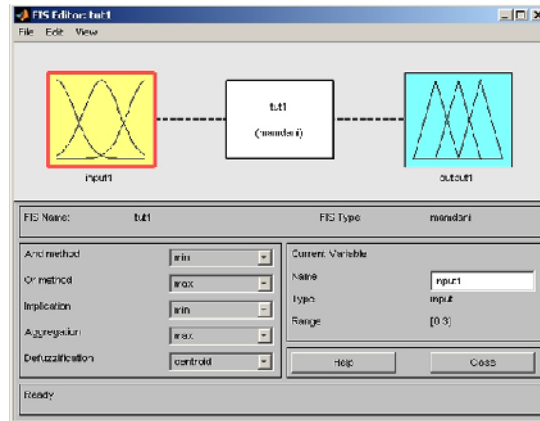


Figure 2.2 – Mamdani pattern

Fuzzy logic algorithm

1. Define the linguistic variables and terms (initialization)
2. Construct the membership functions (initialization)
3. Construct the rule base (initialization)
4. Using the membership functions convert crisp input data to fuzzy values.(fuzzification)
5. Using rule base evaluate the rules.(inference)
6. Combine the results of each rule (inference)
7. The output data should be converted to non-fuzzy values (defuzzification)

3. FUZZY MODEL FOR CLOUD TESTING

One of the major advantages of cloud testing towards conventional testing is that the performance can be improved due to the availability of resources in the cloud. Performance in cloud can be improved by taking care of parameters network traffic, disk storage [9] and RAM. The above is taken as data for fuzzy model, When network traffic is low then the performance of the system will be high[12], when the disk storage and RAM is also high the performance will improve.

3.1 – PERFORMANCE BASED ON NETWORK TRAFFIC

Performance issues in cloud computing mainly depends on the overall traffic patterns and peaks in the system[25]. Network traffic plays major role in performance of cloud computing. When network traffic in high the performance is low and when the network traffic is low the performance is high. Table 3.1 show the performance range of network traffic.

Table 3.1 Performance Range of Network traffic

Performance	Network Traffic
High	<200MB
Average	200MB
Low	>200MB

Linguistic Variables

A set of linguistic terms that are decomposed are called linguistic variables. They are the input or output variables of the system whose values are not numerical values but words or sentences from a natural language [6].

INPUT : Network Traffic NT, range[0-100] MB
 OUTPUT: Performance P, range [0-1]

Membership Functions

A membership function is used to quantify a linguistic term[24], which are used in the fuzzification and defuzzification steps of a FLS, to map the non-fuzzy input values to fuzzy linguistic terms and vice versa. With a condition and a conclusion a fuzzy rule is a simple IF-THEN rule. Figure 3.1 shows that if the When network traffic in high the performance is low and when the network traffic is low the performance is high. Membership function is constructed to control the output variable based on rule.

Network Traffic NT: Low, Average, High
 Performance : Low, Average, High

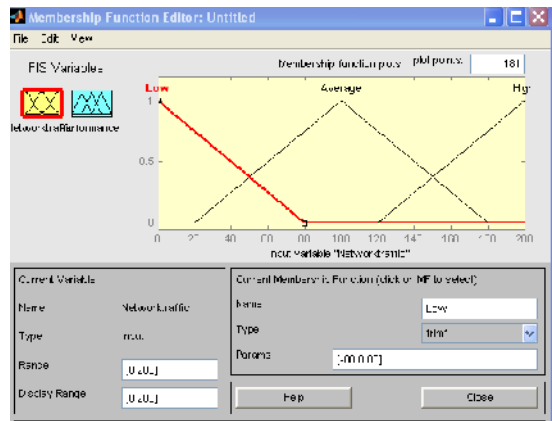


Figure 3.1 Membership function editor for network traffic

Fuzzy Set Operations

Fuzzy set operations perform evaluation of the fuzzy rules and combine the results of the individual rules. The non-fuzzy sets and fuzzy set operations are different. The final result is obtained by combining the result of each rule after evaluation. This process is called inference. The results are combined in different ways for individual rules. Table 3.2 shows the fuzzy rules for network traffic and figure 3.2 shows the corresponding fuzzy rule editor.

Table 3.2 – Fuzzy rules for network traffic

1. IF (Network traffic is LOW) then (performance is HIGH)
2. IF (Network traffic is AVERAGE) then (performance is AVERAGE)
3. IF (Network traffic is HIGH) then (performance is LOW)

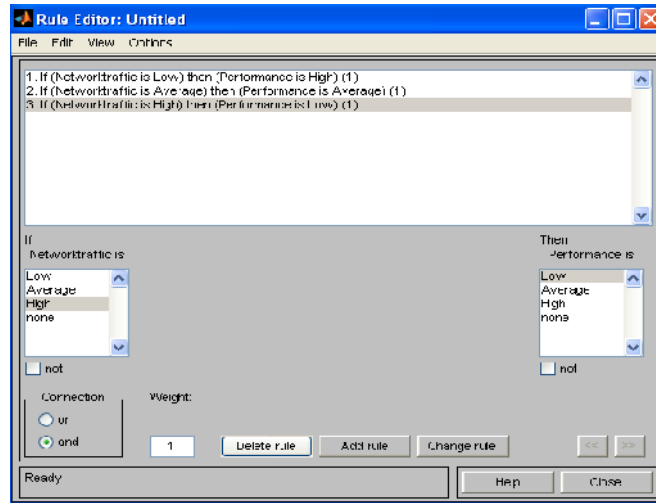


Figure 3.2 – Fuzzy Rule Editor

Here the results are ranged between [0-1] as shown in table 3.3

Table 3.3 Result Ranges

Result	Ranges
Low	0.0 - 0.5
Average	0.5
High	0.5 – 1.0

Based on the fuzzy rules in table 3.2 combined with the table 3.3 the final result surface viewer figure 3.3 is got which depicts that when network traffic(x-axis) is low (i.e) 0 the performance(y-axis) is high (i.e) between 8 and 9.

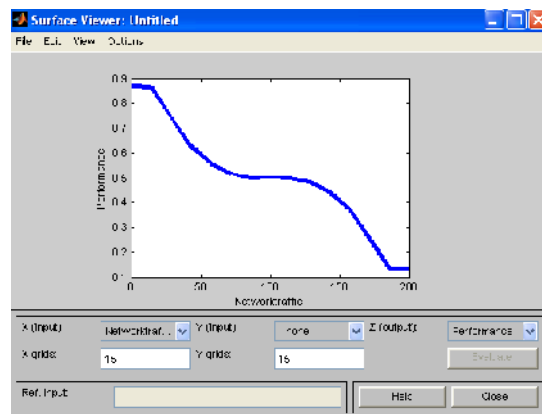


Figure 3.3 – Surface viewer of the results

Defuzzification

Overall result is a fuzzy value which is got after inference step. Final crisp output is got after the defuzzification of the result. This is the purpose of defuzzier component of a FLS. According to the membership function of the output variable defuzzification is performed. Figure 3.4 shows the rule viewer of the results obtained for three rules as in table 3.2. Here the left most triangles show low value, middle triangle for average value and right most triangles for high value. The yellow part is the input; network traffic which ranges from 0 to 200 and blue is for output, the performance ranges from 0 to 1. The red line shows the final result.

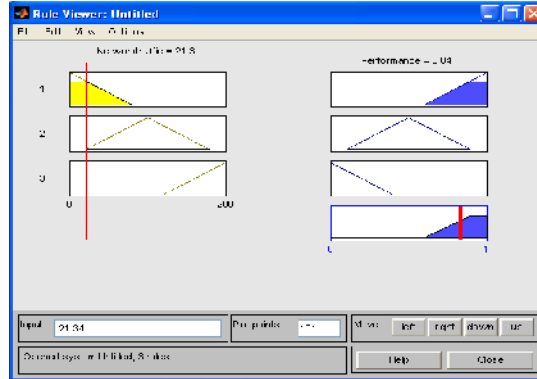


Figure 3.4 – Rule viewer for network traffic

3.2 PERFORMANCE BASED ON NETWORK TRAFFIC AND DISK STORAGE

New rule with two inputs as in figure 3.5 can be framed to improve the performance. Along with network traffic, disk storage is also considered. When disk storage space is high then performance of system is also high. Combining this with network traffic based on AND rule new rules can be framed. Though we might frame nine rules we can consider a few for result satisfactory[26]. Table 3.4 show the performance range of disk storage.

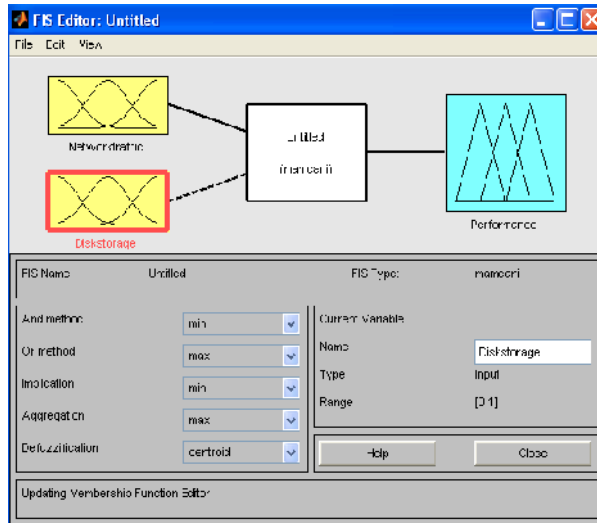


Figure 3.5 Pattern with two inputs network traffic and disk storage.

Table 3.4 Performance Range of Disk storage in cloud

Performance	Disk storage
Low	<1TB
Average	10TB
High	>10TB

The input and output variable of the system now are

INPUT : Network Traffic NT, range[0-200]MB AND Disk storage, range[0-10]TB

OUTPUT: Performance P, range [0-1]

The fuzzy rules with AND logic are shown in table 3.5

Table 3.5 Fuzzy rules for AND logic for two inputs

1. IF (Network traffic is LOW) AND (Disk Storage is HIGH) then (performance is HIGH)
2. IF (Network traffic is AVERAGE) AND (Disk Storage is AVERAGE) then (performance is AVERAGE)
3. IF (Network traffic is HIGH) AND (Disk Storage is LOW) then (performance is LOW)

The membership function, rule viewer and surface viewer for these two inputs are shown in figure 3.6, 3.7 and 3.8 respectively.

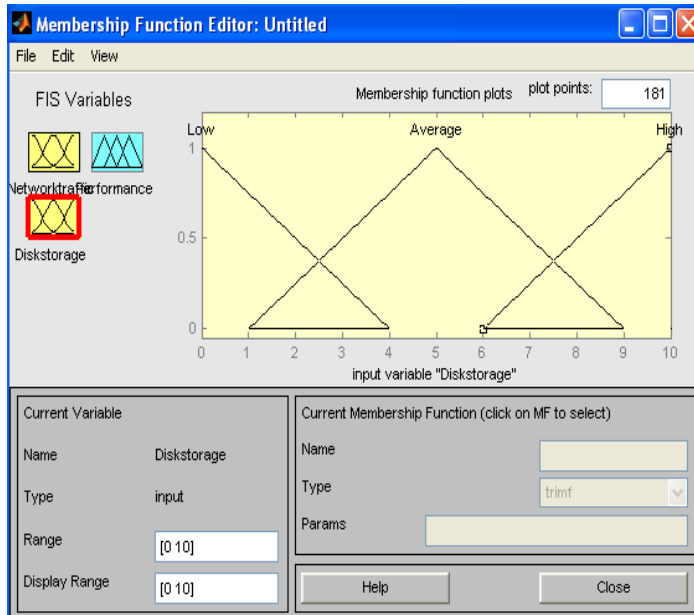


Figure 3.6 – Membership function editor for disk storage

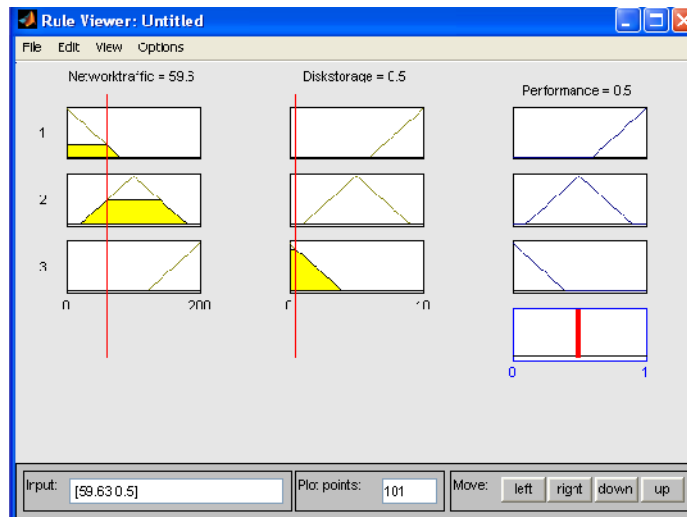


Figure 3.7 – AND logic Rule viewer for network traffic and disk storage

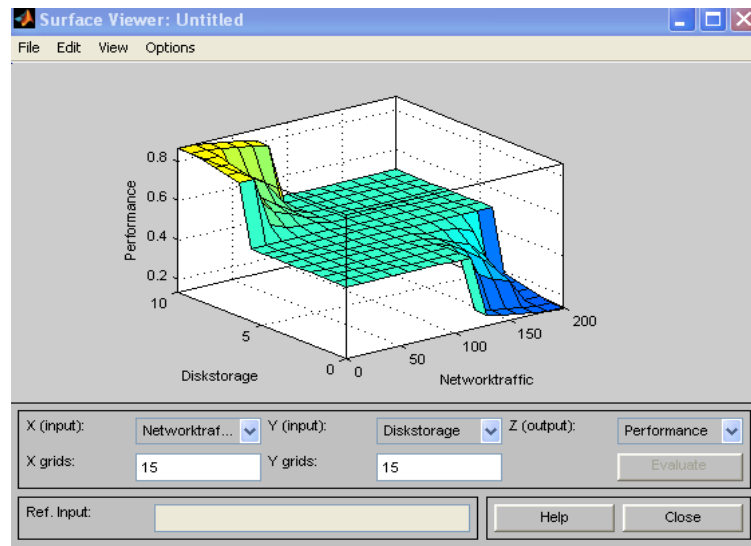


Figure 3.8 – AND logic Surface viewer for network traffic and disk storage

3.2 Performance Based On Network Traffic, Disk Storage And Ram

To look into a better performance one more input RAM is considered along with network traffic and disk storage. Figure 3.9 depicts the pattern. When RAM speed is high then performance of system is also high. Combining this with network traffic, disk storage based on AND rule new rules can be framed. Table 3.6 show the performance range of disk storage.

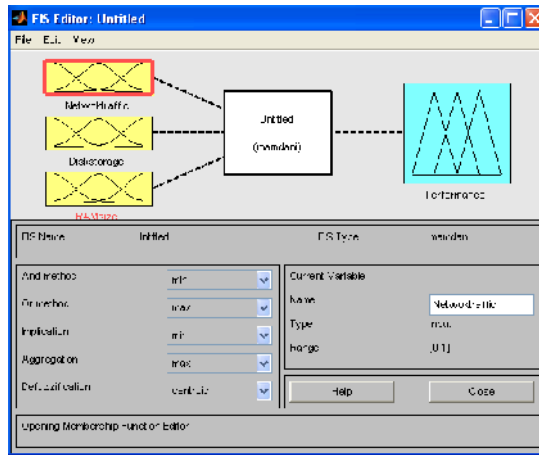


Figure 3.9 – Pattern with three inputs network traffic, disk storage and RAM

Table 3.6 Performance Range of RAM speed in cloud

Performance	RAM speed
Low	<400GB
Average	400GB
High	>400GB

The input and output variable of the system now are

INPUT : Network Traffic NT, range[0-200]MB AND Disk storage, range[0-10]TB
 AND RAM speed, range[0-40]GB
 OUTPUT: Performance P, range [0-1]

The fuzzy rules with AND logic are shown in table 3.7

Table 3.5 Fuzzy rules for AND logic for two inputs

1. IF (Network traffic is LOW) AND (Disk Storage is HIGH) AND (RAM is HIGH) then (performance is HIGH)
2. IF (Network traffic is AVERAGE) AND (Disk Storage is AVERAGE) AND (RAM is AVERAGE) then (performance is AVERAGE)
3. IF (Network traffic is HIGH) AND (Disk Storage is LOW) AND (RAM is LOW) then (performance is LOW)

The rule viewer and surface viewer for these three inputs are shown in figure 3.10 and 3.11 respectively.

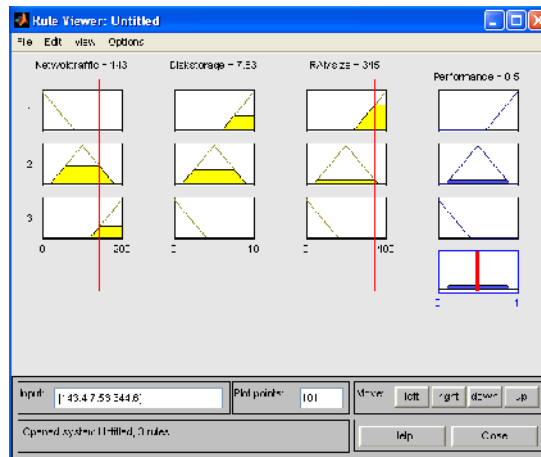


Figure 3.10 – AND logic Rule Viewer for network traffic, disk storage and RAM

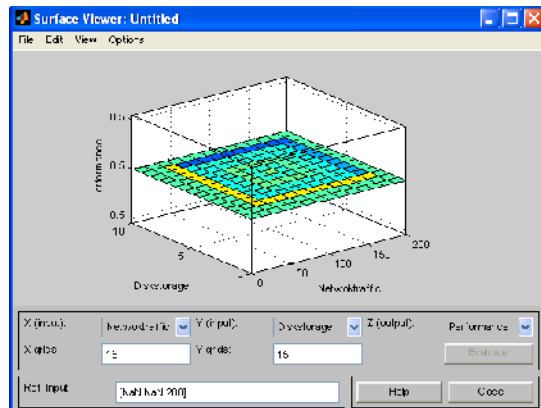


Figure 3.11 – AND logic Surface Viewer for network traffic, disk storage and RAM

4. RESULTS AND DISCUSSION

Thus from the previous session we conclude that though the disk storage and RAM plays an important role in the increase of performance, network traffic contributes more in that. Even though the disk storage and RAM are more since network traffic is high the overall performance is only average (0.5) by figure 3.10.

In addition to AND logic OR logic is also used to test the performance. By surface rule viewer figure 4.1 and surface viewer 4.2 it is proved that the performance is low (0.3) in OR rules than that of and RULES.

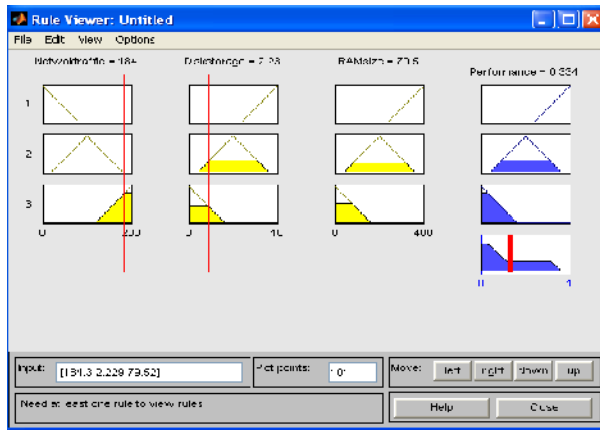


Figure 4.1 – Rule viewer for OR rule

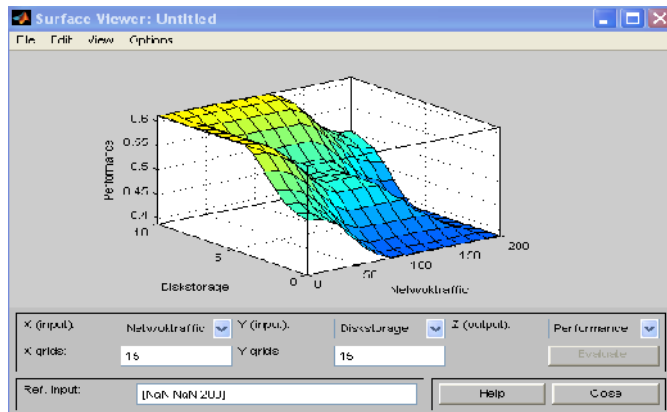


Figure 4.2 – Surface viewer for OR rule

Table 4.1 shows that when the network traffic is high even when disk storage and RAM are high/medium the final performance is low/medium. When the network traffic is low in spite of disk storage and RAM low/medium the performance is high/medium. Thus network traffic plays major role in the performance of cloud testing. Figure 4.3 depicts the rule viewer of in which performance is high (0.868) when network traffic ,disk storage and RAM are high.

Table 4.1 – Performance based on all rules

Network Traffic	Disk Storage	RAM	Performance of cloud
Low	High	High	high(0.868)
Low	Low	Low	Medium(0.504)
Low	Low	High	Medium(0.541)
Low	High	Low	Medium(0.519)
Low	Medium	High	Medium(0.612)
Low	Medium	Medium	Medium(0.612)
high	High	Low	Medium(0.5)
high	Medium	Low	Low(0.391)
high	Low	Low	Low(0.134)
high	Medium	Medium	Low(0.393)

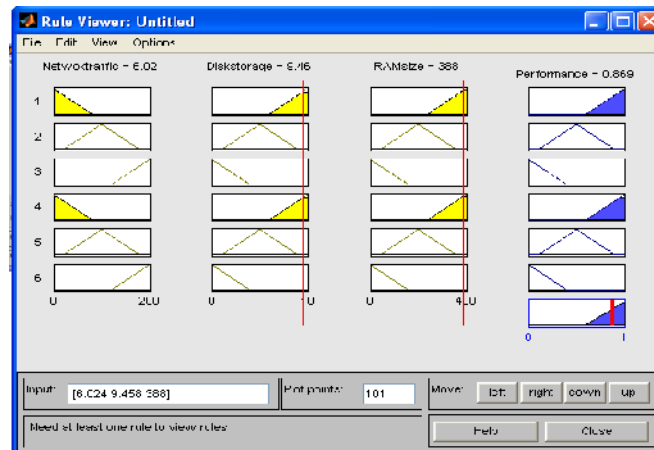


Figure 4.3 – Rule viewer

5. CONCLUSION

New fuzzy mathematical model which provides unlimited resources with scalability[13], flexibility and availability of distributed testing environment is created for cloud computing which opens new opportunities for software testing. Software testing performance can be increased in cloud by increasing the working of cloud parameters namely network traffic, disk storage and RAM. New fuzzy mathematical model is created using these parameters. The model is analysed using AND logic fuzzy rules and OR logic logic fuzzy rules. Mamdani fuzzy additive rule model is used for the same. We found by results and discussion that AND logic rules provide better than that of OR logic rules. Thus inspite of challenges faced in cloud for software testing the performance can be increased by improving the performance of the parameters network traffic, disk storage and RAM in cloud[14]. As a future work XOR logic can analysed..

REFERENCES

- [1] A Vanitha Katherine, K Alagarsamy, "Software Testing in Cloud Platform: A Survey", IJCA, Volume 46– No.6, May 2012
- [2] <http://www.cognizant.com/insights/perspectives/taking-testing-to-the-cloud>
- [3] Mrs.A.Vanitha Katherine, Dr. K. Alagarsamy, "Conventional software testing Vs cloud testing", International Journal Of Scientific & Engineering Research, Volume 3, Issue 3, March-2012 1 ISSN 2229-5518
- [4] J. Mendel. Fuzzy logic systems for engineering: a tutorial. Proceedings of the IEEE, 83(3):345{377, Mar 1995.
- [5] Aleksandar Raki , Introduction 3 Fuzzy Inference
- [6] Fuzzy control programming. Technical report, International ElectrotechnicalCommision, 1997.
- [7] J. Mendel. Fuzzy logic systems for engineering: a tutorial. Proceedings of the IEEE, 83(3):345{377, Mar 1995.
- [8] Jerry Gao, Xiaoying Bai, and Wei-Tek Tsai, Cloud Testing- Issues, Challenges, Needs and Practice , SEIJ, Vol. 1, No. 1, Sept 2011.
- [9] www.cloudrive.com.au-InstallationManual.pdf
- [10] Vinod Venkataraman, Ankit Shah, Yin Zhang, "Network-based Measurements on Cloud Computing Services",<http://www.cs.utexas.edu/~vinodv/files/cc-measure.pdf>.
- [11] Zach Hill, Marty Humphrey, "A Quantitative Analysis of High Performance Computing with Amazon's EC2 Infrastructure: The Death of the Local Cluster?", the 10th IEEE/ ACM International Conference on Grid Computing (Grid 2009). Oct 13-15 2009.

- [12] Dominique A. Heger, “A Performance Centric Introduction to Cloud Computing”,<http://www.dhtusa.com/media/CldPerf.pdf>.
- [13] Raj Jain, “The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling”, Wiley, 1991, pp 203-220, pp 381-389.
- [14] “Cloud computing – Wikipedia, the free encyclopedia”,http://en.wikipedia.org/wiki/Cloud_computing.
- [15] “httpperf – Wikipedia, the free encyclopedia”,<http://en.wikipedia.org/wiki/Httpperf>.
- [16] “Planetlab testbed”,<http://www.planet-lab.org>.
- [17] Kaiqi Xiong, Harry Perro, “Service Performance and Analysis in Cloud Computing”<http://www4.ncsu.edu/~hp/Kaiqi10.pdf>.
- [18] Daniel A. Menasce, Paul Ngo, “Understanding Cloud Computing: Experimentation and Capacity Planning”, Proc. 2009 Computer Measurement Group Conf, Dec.7-11, 2009.
- [19] Novák, V., I. Perfilieva, and J. Mo ko , Mathematical Principles of Fuzzy Logic, Kluwer Academic, Dodrecht, 1999.
- [20] Pedrycz, W., Fuzzy Control and Fuzzy Systems, John Wiley & Sons, 1993.
- [21] Peter, B., N. Stephan, and W. Roman, A Brief Course in Fuzzy Logic and Fuzzy Control, Department of Knowledge-based Mathematical Systems, 1996 available at <http://www.esru.strath.ac.uk/Reference/concepts/fuzzy/fuzzy.htm>.
- [22] Rajashekar, S., and G.A. Vijayalksmi, Neural Network, Fuzzy Logic and Genetic Algorithms, Prentice Hall, 2003.
- [23] Srivastava, P., S. Sagggar, A.P. Singh, and G. Raghurama, Optimization of software testing effort using fuzzy logic, International Journal of Computer Sciences and Engineering Systems, vol.3, no.3, pp.179–184, 2009.
- [24] Zadeh, L.A., Fuzzy sets, Information and Control, vol.8, pp.338–353, 1965.
- [25] <http://cloud.libero.it/it/wp-content/uploads/2013/05/Joyent-Performance-and-Scale-in-Cloud-Computing-White-Paper.pdf>
- [26] https://dSPACE.ist.utl.pt/bitstream/2295/198492/1/CDI_PL_Fuzzy_Toolbox.pdf