

## Intelligent reliability management in hyper-convergence cloud infrastructure using fuzzy inference system

Nadia Tabassum<sup>1,\*</sup>, Muhammad Saleem Khan<sup>2</sup>, Sagheer Abbas<sup>2</sup>, Tahir Alyas<sup>3</sup>, Atifa Athar<sup>4</sup> and Muhammad Adnan Khan<sup>2</sup>

<sup>1</sup> Department of Computer Science, NCBA&E, Virtual University of Pakistan, Lahore,

<sup>2</sup> Pakistan Department of Computer Science, NCBA&E, Lahore, Pakistan

<sup>3</sup> Department of Computer Science, Lahore Garrison University, Lahore, Pakistan

<sup>4</sup> Department of Computer Science, CUI, Lahore, Pakistan

### Abstract

Hyper-convergence is a new innovation in data center technology, it changes the way clouds manage and maintain enterprise IT infrastructure. Hyper-convergence is more efficient and basically agile technology environment. Cloud computing is a latest technology due to provision of latest cloud services over the internet. The cloud service providers cannot promise accurate reliability of their services i.e. problem in provisioning of software or hardware failure etc. Reliability of cloud computing services depends on the ability of fault tolerance during the execution of services. There are so many factors can cause faults, such as network failure, browser crash, request time out or hacker attacks. When users are facing these types of faults, they usually resubmit their requests. However, if there is any key element involved in faults or errors, additional action may be needed to deal with system logs. If there is anomaly behavior occurred in faulted virtual machine, these VMs may need extra attention from cloud system protection and security point of view. In this paper, provision of reliability management in hyper-convergence cloud infrastructure is proposed and self-healing techniques in software as a service on the basis of failure in cloud services. Intelligent cloud service reliability framework will increase the reliability during execution of cloud service.

**Keywords:** Hyper-convergence, Cloud Computing, Datacenter Software as a service, reliability management

Received on 20 May 2019, accepted on 19 June 2019, published on 10 July 2019

Copyright © 2019 Nadia Tabassum *et al.*, licensed to EAI. This is an open access article distributed under the terms of the Creative Commons Attribution licence (<http://creativecommons.org/licenses/by/3.0/>), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi: 10.4108/eai.13-7-2018.159408

### 1. Introduction

In current era, we are living in the universe where data is being produced day by day on demand, on command by the organization, institutions and many other firms. The amount of huge data that we consumed and produced by different smart devices. Devices like, smart phone, computers and sensors. Most of the tools and individuals are at high rate and produce huge amount of data. It is difficult to manage huge amount of data[1]. New technologies introduced to overcome these issues. To overcome these issues and making our work cost effective, we use cloud computing services. Cloud computing is cheap and pay-per-use, this

attitude of cloud computing putting resources over cloud infrastructure. Cloud computing service uses internet to provide service to consumer and use data center to host applications. Cloud services are available for consumer as pay-per-use, quality and services over the internet[2].

#### 1.1 Essential characteristics of Cloud Computing

Some of the most important characteristics of cloud computing is following: that the concept of a ranking system came into being. This ranking system receives the requests from different users, which may differ w.r.t their requirements. Then, this system will look for some services

\*Corresponding author. Email: [Nadiatabassum@vu.edu.pk](mailto:Nadiatabassum@vu.edu.pk)

for users and assign a possible rank according to the Quality of Service (QoS) [3].

### 1.1.1. On-Demand Service:

On-demand service is a model or technique in which we just provide the cloud users the facility that they can get their services from service providers anytime at any place. The users can get this facility to fulfil its work or any application so the users can have this facility to avail the services on demand. The service providers provide these resources to their users at any time[4].

### 1.1.2. Broad Network Access:

By its name we can get the idea of it the cloud system is just used in broad network area so that everyone can get access their services. Most of the companies can use this facility to remain updated to their clients or other organizations by using the cloud services and anyone can use these services but it is depend on the network used like whether it is private or public it means if you use the private cloud than all the information will be under the members of the private members those are sign in to the cloud services but if it is public network than anyone can get the access of about any information and anyone can get the advantage of the services provided to the cloud users[5].

### 1.1.3. Resource Pooling:

Resource pooling is a technique in which the consumer can acquire and release the resource when it will be required on demand. The PaaS users can get the resource from the resource pool on demand so that the user can make use of this resource and then give it back to the resource pool. It also reduces all the complexities that cloud has to face by using resource pooling techniques[6].

### 1.1.4. Rapid Elasticity:

Rapid Elasticity refers to enhancement of cloud services without effecting the cloud users and buyers all the services in a very reliable and flexible manner. Cloud just provides their services to their users very easily and their users can easily get the services and get advantage of these services on demand. The cloud users can get extra storage space to have more resources from cloud provider in this way they can use more services[7].

### 1.1.5 Measured Service:

Measured service means paying cloud service cost as per their usage. It also known as metered service. In measured services all the problems and faults are being controlled and monitored. Measured service is a term that IT experts apply to distribute computing

## 1.2 Cloud Service Models

The most commonly used service models through which the cloud system provides different services to the users consumers are following:

### 1.2.1. Infrastructure-as-a-Service:

Infrastructure is just like a support or foundation which is used to provide these types of services to your customers and cloud users like by using infrastructure u can give resources to the equipment's that are used in different works like virtualization, storage area, networking etc. You can easily give these resources or facilities to your customers and cloud buyers and users to enhance your services provided to the customers. You can increase your storage area and the network speed provided to the customers so that they can easily get use the benefits of the best networking speed available[8].

### 1.2.2 Platform-as-a-Service:

Platform as a service is a worldview for conveying working frameworks and it is used to provide the cloud users all the facilities through internet which is a worldwide network where we can access and download anything but the difference is quite simple that these facilities are provided by the cloud service providers and you just need to pay for what you used on cloud. PaaS gives a stage apparatuses to test, create, and have applications in a similar domain[9].

### 1.2.3 Software as-a-Service:

Software as a service is used in which the user does not need to download any application on its own computer because now the SaaS is available through which we can access any application not just only on our own computer but at any place. It is a model in which the other person through internet gets access the other persons and they can access these all application on internet. It also decreases the expenses of installation, provisioning cost etc[10].

## 1.3 Data reliability in cloud computing

Reliability means working the virtual machines, if the exceptions and malfunctioning occur. The system functioning is error free and in good conditions for service delivery. Service consumers consider reliability as proper functioning, security, and ease of use. Service providers also consider reliability in service creation, deployment, integration and separation.[11]

The reliability in cloud computing environment also includes providing proper functioning in different stages in service lifecycle. Service integration and separation allow service providers to offer both full set of functionality and part of functionality to service consumers according to service level agreements. The reliability covers various aspects of cloud computing. The base line of the reliability is to provide functioning services.[12]

### 1.3.1 Types of Failure in cloud computing

In conventional software engineering reliability, four main approaches to design a reliable software system. These four approaches are fault prevention, fault removal, fault tolerant, and fault forecasting. However, in cloud computing environment, cloud applications only accept fault-prevention techniques and fault removal techniques to develop fault-free software as service[13].

The large-scale cloud services involve large number of virtual machines and middle ware layers. Failures of these components affect reliability of cloud applications directly. The idea of most cloud service providers deploys their services in large data centers. All of services are running in virtual machines that reside in physical machines. There are usually multiple virtual machines running in one physical machine. When a virtual machine is initialized, the administrator or virtual machine monitoring system gets resources from a resource pool to build requested virtual machine. Reliability is cloud computing under the different conditions like network resources, latency and cloud monitoring can result in low performance. Such conditions must be observing by autonomous system to avoid the delivery of cloud services failures[14].

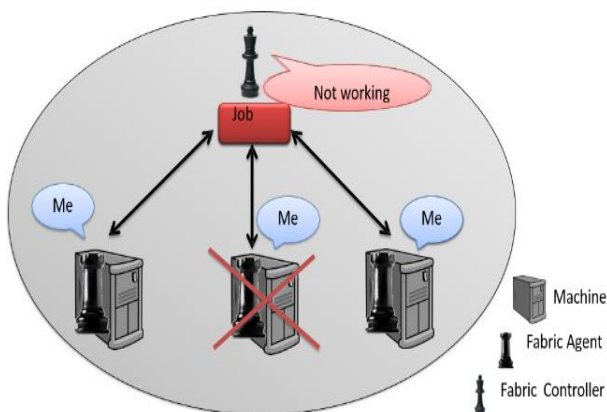


Figure 1. VM Checker

In order to enhance reliability, we need to identify faults. system event logs record most of system events that include system fault related events. We can trace system faults through system event logs. There are always system critical events happened before system enter fault states as shown in

figure 1. Therefore, if a system could predict system critical events, it can predict system faults before they really happen. Researchers dig into this problem from different aspects. Following study presents several techniques for system fault monitoring. Through machine learning techniques, we can find some patterns that always appear when system faults occur. Statistical data is used in mining and detecting fault patterns in service event logs are normalized according to domain information in Memory Module.

## 2. Related Work

Cloud computing has allowed the efficient management, deployment and configuration of clusters where the aforementioned frameworks can be deployed by taking advantage of both the elastic nature of the cloud, where reserved resources are used for as long as they are needed (i.e., pay-as-you-go), and the deployment/management ease of use. To exploit these two properties and fulfil enterprise needs for minimizing infrastructure maintenance and operation costs, companies follow a similar approach for either public or on premise cloud offerings: they utilize a service that launches and manages big data clusters in order to execute the requested workloads whereas a single storage back-end is used to host data[15].

Ian andrusaik proposed framework in his paper titled “A Reliability-Aware Framework for Service-Based Software Development” allows the idea of hyper-convergence is to simplify operation and management of data centers by converging the computing, storage and networking components into a single, software-driven appliance. It’s defined as an IT infrastructure framework in which storage, virtualized computing and networking are tightly integrated within a data center.

A prototype implementation has been developed as a proof of concept of the design which has been evaluated and showed that the system is successful at providing availability when failure occurs at a cost to overall performance[16].

Different autonomic monitoring system is proposed by the researchers. Monitoring framework is proposed by [17] for the web service-enabled applications. In paper he discussed the hardware and software resources in the form of web service-enabled. The elasticity of both hardware and software resources are monitored. In autonomous system, a self-optimized monitoring algorithm is proposed which updating dynamic information and self-adaptive events.

Fault prevention and fault tolerance intend to give the capacity to convey a service that can be trusted, while fault expulsion and fault forecasting mean to achieve trust in that capacity by defending that the utilitarian and the steadfastness and security details are sufficient and that the framework is probably going to meet them. It is significant that repair and fault tolerance are connected ideas; the

refinement between fault tolerance and support in this paper is that upkeep includes the cooperation of an external agent[18].

In a cloud storage framework, many components like storage, services, hardware can result in data failure. Data failures additionally prompt cloud benefit failures. The fundamental causes of cloud data failures hardware, system, software and power failures. Data reliability incorporates augmenting solidness and accessibility of data. Durability mitigates perpetual failures and accessibility mitigates transient failures.

For automation in reliability monitoring, an agent-based approach is helpful where diver diverse provision of software services is required. This approach will support in automated system in every unconditional situation where software behaviour possible to specify. In autonomous situation, agent can evolve, learn, cooperation with entities and negotiation can perform. Expanding system required agent behaviour role while rapid change occur[19].

When the indexing procedure is going on in cloud services, the key factor is that the requirement of the user should be satisfied. Such kind of framework is desired that will fulfil these requirements. By looking on to the above figure, it is known that indexing manager will receive the information and after that, process it according to the ranking parameters like performance, usability, and cost. Indexing Manager will consider it for the best service as desired by user necessities. Indexing Administrator will also be answerable for other activities as well, i.e. taking characteristics for ranking, the track record of characteristic value, and ranking result[20].

### 2.1 Transformation of hyper-converged

The journey of transformation is started from the conventional system. In traditional system, all modules need different skill to manage. All entities configured and tested separately as shown in figure 2. In converged era, the hardware defined infrastructure was introduced along with monitoring software and backup. In hyper-converged, all server components on single unit and integrated through software defined environment. All the components are readily available and ready to use [21].

Data centre is a facility that contains several computers that are connected together for the purpose of storing and transmitting data. The facility is designed to be used by several peoples and is equipped with hardware, software, peripherals, power conditioning, backup, and communication and security systems. Different architectures of data centres involves Traditional infrastructure (TI), Converged infrastructure (CI) and Hyper Converged infrastructure (HCI)[22].

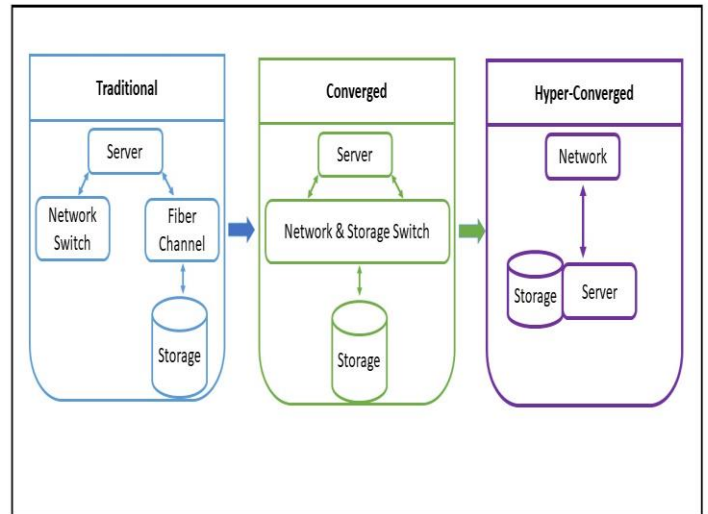


Figure 2. Evolution of hyper-convergence

Cloud ecosystem a type of hype-convergence consists of different cloud providers, reseller and adopters that work together to enable cloud services. The cloud ecosystem is interconnected with cloud resources, different cloud services and applications for cloud consumers & businesses. The cloud ecosystem can be categorized according to their service model i.e. Infrastructure as service (IaaS), software as a service (SaaS) and platform as a service (PaaS). [23]

### 3. Proposed Methodology

In this proposed model, the first it will collect the service history, service weight, QoS parameters, execution time. After collecting the service detail, it will analyzed the virtual machines status and calculate the utility, if the existing pattern are available in Memory Module then no need for new plan. In provision of reliable services in cloud computing, system will determine the planning strategy in decide phase. At the plan phase, the system will generate the plan according to the situation and finally execute the plan.

The proposed framework of Intelligent Cloud Service Reliability management as shown in figure 3 consists of five major modules.

- Service Monitoring Agent
- Self-Healing Layer
- Service usability Layer
- Learning Module
- Memory Module

#### 3.1 Service Monitoring Agent

Service Monitoring Agent is divided into three type of monitoring checks

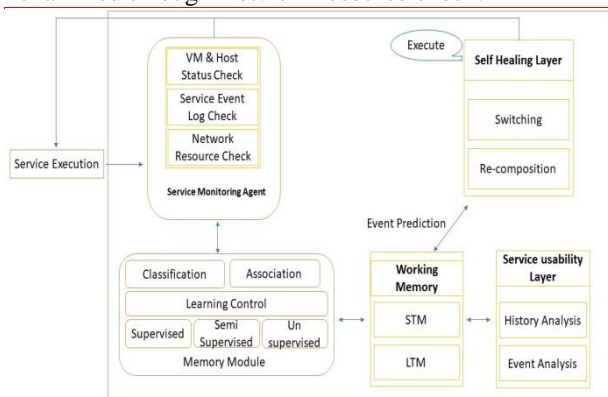


1. Virtual machine & Host Checking
2. Service Event Log Check
3. Network Resource Check

The Service Monitoring Agent is responsible for monitoring provision of cloud services over the network.

When cloud services are provisioning over the network through user interface, the service are being execute will monitor through our proposed intelligent cloud service reliability framework. In Service Monitoring Agent, service monitoring as well as service analysis will be monitored. In service examination layer, there are three sub modules to monitor the virtualized cloud service through different check points.

In service examination layer, the cloud service will be examined through Network resource check.



**Figure 3.** Proposed Intelligent reliability management Framework

In Virtual machine checking, the health status of all virtual machine will be monitored through VM checker as shown in figure. 3. Fabric controller monitor all the virtual machine through fabric agent. If any virtual machine is not working then the fabric agent will report to fabric controller for alternate virtual machine to provide error free service delivery over the network. Similarly, the fabric controller will check the status of host machine.

Service Event Log checking will responsible for the auditing of cloud service in the form of log checking and log will be analyzed with the history checking in service usability layer. Network resource check include the physically resources during provision of cloud services.

### 3.2 Service usability Layer

This layer is responsible for the cloud service history analysis. This layer is divided into two sub layers.

1. History analysis
2. Event analysis

History analysis layer will keep record the failure types and stored in Memory Module for future failure predictor.

Event analysis will be responsible for the event failure status and send status for event prediction to recover such type of failures.

### 3.3 Self-Healing Layer

This layer will responsible for recovery of failure components in cloud service provision. This layer will keep track of all the available virtual machines and recovery of failure by keeping the track of such failures and learn the previous event failures. Self-healing layer is divided into two sub modules

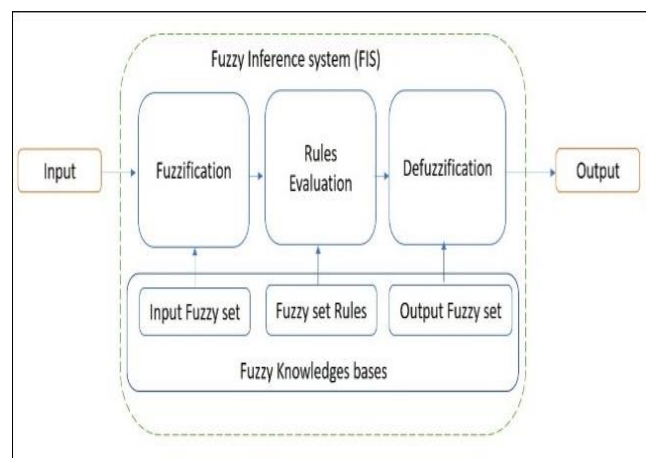
1. Switching
2. Re-composition

In provision of reliable cloud service provision, the switching will be performed if, the recovery of virtual machine is not working after re-composition. The fabric controller will activate the new cloud service after checking the service examination layer parameters.

While service healing process the recovery of service is subject to system security analysis. The required service is for the same host or system. During healing process, the system will check the service examination in term of virtual machine, Host machine, designation machine, current service status, network resource.

## 4. Implementation and Evaluation

In this section Mamdani Fuzzy Inference System is used to simulate the proposed reliability management in cloud computing. The description of fuzzy inference system (FIS) is explained in Figure 4.



**Figure 4.** Proposed Fuzzy Inference System for IRM For the reliability management in cloud computing, Service response time (SRT), Virtual machine utilization (VMU), Event Log and network resource check will perform

important role in Service Monitoring Agent. To implement the intelligent reliability management in software as a service, MATLAB FIS tool is used as shown in Figure 5. The selected parameters for intelligent reliability management are following:

- Service response time (SRT)
- Virtual machine utilization (VMU)
- Event log
- Network resource Utilization (NRU)

Reliability Management depends on four input variables SRT, VMU, EventLog and NRU. The parameter of SRT explains if the response time of cloud execution is low then the reliability in cloud is high. Three membership variables for SRT is used low, medium and high which explain the degree of reliability. We can say the reliability is high if response time of SRT is low, medium and high. The second input variable is Virtual machine utilization (VMU) having three membership functions low, medium and high. The third input variable is EventLog having three membership functions No, Yes and Critical. If the value of EventLog is no then no uncertainty is seen during cloud service execution. If the value of EventLog is Yes then cloud service is not functioning during cloud service execution.

### 4.1. Inference Engine

Characterizes administrators and defuzzifier utilized as a part of the surmising procedure.

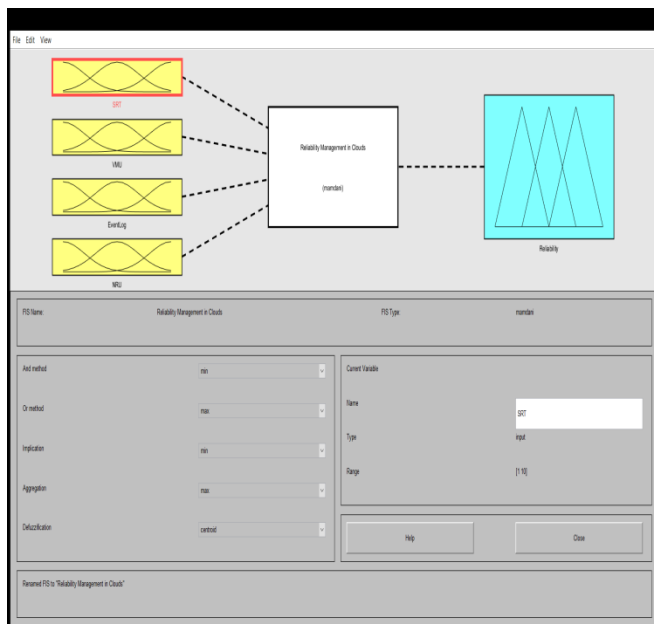


Figure 5. Proposed FIS for IRM

### 4.2. Membership Functions

Graphical & mathematically representation of the above mentioned I/O MF of AFIS Input variables are shown in table 7. Detail of each input variable is explained in Table. 2-6.

### 4.3. Rule Base

It is a set of “If-Then” rule set that characterizes the derivation demonstrate. The control structure resembles: "If cloud parameter Then what is reliability of cloud service” as shown in figure 6. In Table 1 showed the possible outcome of the four inputs. The detail are explained in table 7.

Table 1. Proposed Lookup table for IRM

IF SRT	AND VMU	AND EventLog	AND NRU	THEN Reliability
Low	High	No	High	Reliable
Low	Medium	No	Medium	Reliable
Low	High	No	Low	Reliable
Low	Low	No	Medium	Reliable
Medium	Medium	Yes	Medium	Re-composition
Medium	Medium	Yes	Low	Re-composition
Medium	Low	Yes	High	Re-composition
Medium	Medium	Yes	Medium	Re-composition
Medium	Medium	Yes	Low	Re-composition
High	High	Critical	High	Switching
High	Medium	Critical	High	Switching
High	High	Critical	High	Switching
High	High	Critical	Medium	Switching
High	High	Critical	High	Switching

Table 2. SRT

Member Functions	Ranges	Regions
Low	1-1-5	1
Medium	1-5-10	1-2
High	5-10-10	2

Table 3. VMU

Member Functions	Ranges	Regions
Low	1-1-5	1
Medium	1-5-10	1-2
High	5-10-10	2

Table 4. EventLog

Member Functions	Ranges	Regions
No	1-1-5	1
Yes	1-5-10	1-2
Critical	5-10-10	2

Table 5. NRU

Member Functions	Ranges	Regions
Low	1-1-5	1
Medium	1-5-10	1-2
High	5-10-10	2

Table 6. Reliability

Member Functions	Ranges	Regions
Reliable	1-1-5	1
Re-composition	1-5-10	1-2
Switching	5-10-10	2

De fuzzifier play a very important role in any fuzzy inference system. In this article centroid DE fuzzifier is used. Eq 1 shown the mathematical representation of DE fuzzifier.

$$\text{Defuzzifier} = \frac{\sum_{i=0}^m c_i \min\{\mu_{srt_1}(s), \mu_{vmu_1}(v), \mu_{eventlog_1}(e), \mu_{nru}(n)\}}{\sum_{i=0}^m c_i \{\mu_{srt_1}(s), \mu_{vmu_1}(v), \mu_{eventlog_1}(e), \mu_{nru}(n)\}} \text{Eq-1}$$

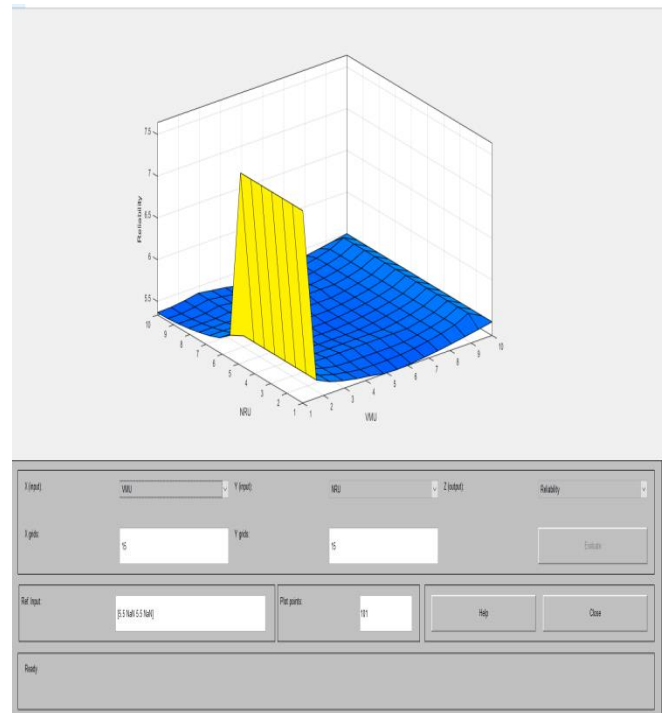


Figure 7. Rule surface for Reliability Management

In Figure No. 7 showing the rule surface for the proposed intelligent reliability management. it observed that if the value of the NRU is 7, VMU is 5.32, and reliability is 6.7. the rule viewer shows that the reliability is at re-composition state, where virtual machines need to re-compile again.

### 5. Simulation and Results

MATLAB 2017b is used for simulation purpose. Figure 8-10 shown the performance of the proposed Intelligent Reliability Management system in terms of low, medium & high reliability.

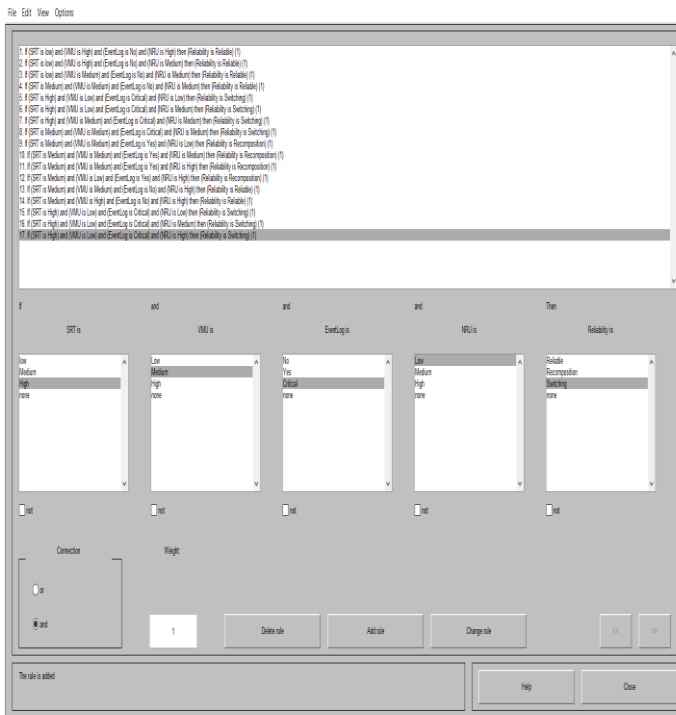


Figure 6. Rule base for Proposed IRM

### 4.4. DE fuzzifier

If SRT value is 5, VMU value is near about 5, event log value is 5, NRU value is 5.5 then the reliability will be in Low as shown in figure 8.

If SRT value is 1.22, VMU value is near about 10, event log value is 3.7, NRU value is 10 then the reliability will be medium as shown in figure 9.



Figure 8. Lookup diagram for Low Reliability

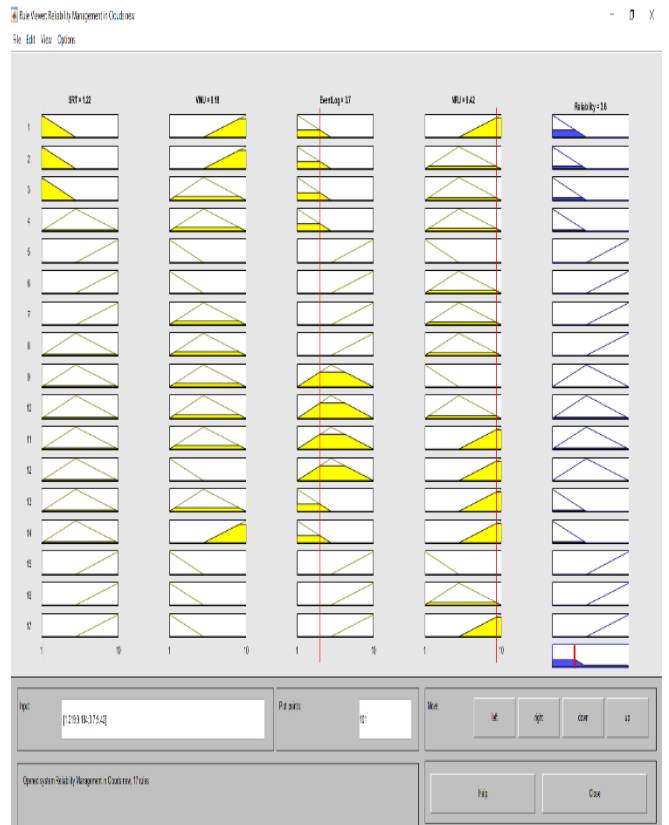
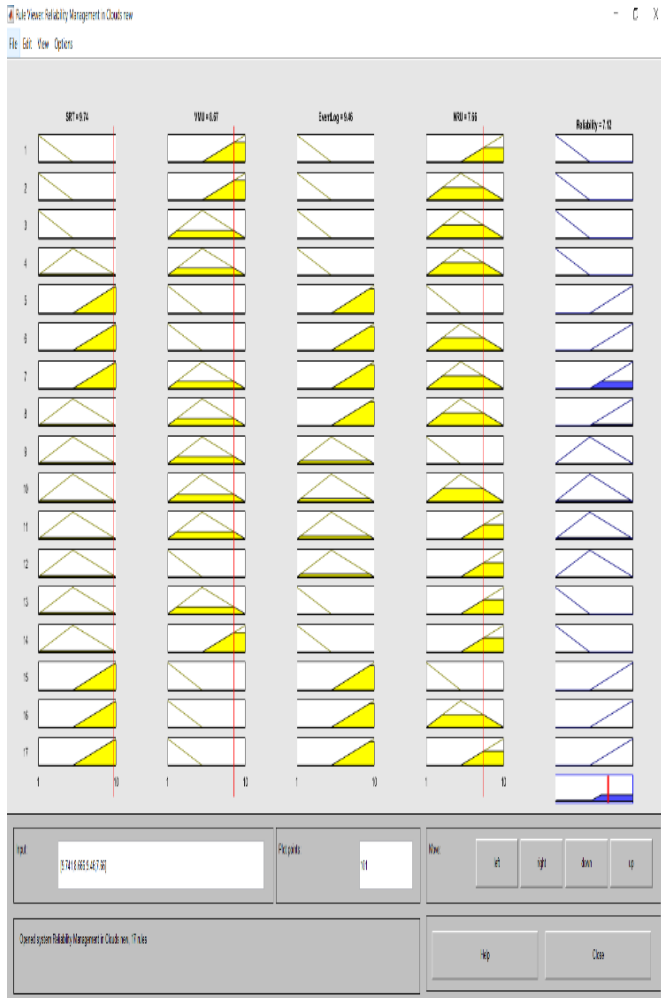


Figure 9. Lookup diagram for Medium Reliability





**Figure 10.** Lookup diagram for High Reliability

If SRT value is 10, VMU value is near about 9, event log value is 9, NRU value is 8 then the reliability will be high as shown in figure 10.

Table 7. Mathematical & Graphical MF of AFIS Input/output variables

Sr NO	i/p	Membership Function (MF)	Graphical Representation of MF
1	S ( $\mu_{Srt}(s)$ )	$\mu_{Srt,Low}(s) = \begin{cases} 1 & s \in [1, 5] \\ 0 & e \end{cases}$ $\mu_{Srt,Medium}(s) = \begin{cases} 0 & s \in [1, 5] \\ 1 & s \in [5, 10] \end{cases}$ $\mu_{Srt,High}(s) = \begin{cases} 0 & s \in [5, 10] \\ 1 & e \end{cases}$	
2	V ( $\mu_{vmu}(v)$ )	$\mu_{vmu,Low}(v) = \begin{cases} 1 & v \in [1, 5] \\ 0 & e \end{cases}$ $\mu_{vmu,Medium}(v) = \begin{cases} 0 & v \in [1, 5] \\ 1 & v \in [5, 10] \end{cases}$ $\mu_{vmu,High}(v) = \begin{cases} 0 & v \in [5, 10] \\ 1 & e \end{cases}$	
3	EI ( $\mu_{EI}(e)$ )	$\mu_{EI,No}(e) = \begin{cases} 1 & e \in [1, 5] \\ 0 & e \end{cases}$ $\mu_{EI,Yes}(e) = \begin{cases} 0 & e \in [1, 5] \\ 1 & e \in [5, 10] \end{cases}$ $\mu_{EI,Critical}(e) = \begin{cases} 0 & e \in [5, 10] \\ 1 & e \end{cases}$	
4	N ( $\mu_{nru}(n)$ )	$\mu_{nru,Low}(n) = \begin{cases} 1 & n \in [1, 5] \\ 0 & e \end{cases}$ $\mu_{nru,Medium}(n) = \begin{cases} 0 & n \in [1, 5] \\ 1 & n \in [5, 10] \end{cases}$ $\mu_{nru,High}(n) = \begin{cases} 0 & n \in [5, 10] \\ 1 & e \end{cases}$	

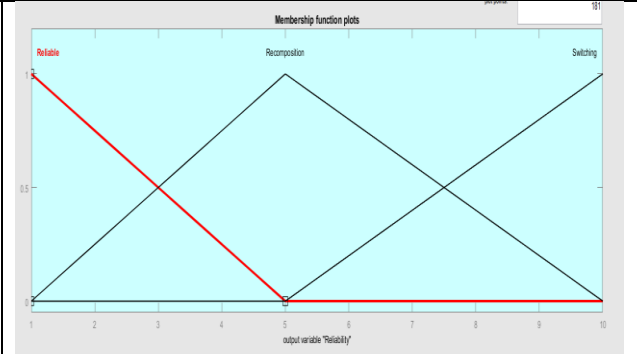
5	R ( $\mu_{rel}(r)$ )	$\mu_{rel, Reliable}(r) = \begin{cases} 1 & r \in [1, 5] \\ e & \end{cases}$ $\mu_{rel, recomposition}(r) = \begin{cases} 1 & r \in [1, 5] \\ e & r \in [5, 10] \end{cases}$ $\mu_{rel, Switching}(r) = \begin{cases} 1 & r \in [5, 10] \\ e & \end{cases}$	
---	-------------------------	---	--

Table 8. Mathematical & Graphical MF of AFIS Input/output variables

Rule No.	Inputs				Values Outputs	Singleton Values
	SRT	VMU	EventLog	NRU	Reliability	
S1	L	H	No	H	R	Low=0.2
S2	L	M	No	M	R	Low=0.2
S3	L	H	No	H	R	Low=0.2
S4	L	H	No	L	R	High=0.4
S5	L	H	No	M	R	Low =0.2
S6	M	M	Yes	M	RC	Low =0.4
S7	M	M	Yes	H	RC	Medium=0.6
S8	M	L	Yes	L	RC	Medium=0.6
S9	M	H	Yes	H	RC	Low =0.2
S10	H	H	Critical	H	S	Medium =0.8
S11	H	H	Critical	M	S	Best=0.8
S11	H	H	Critical	L	S	Best=0.8
S11	H	M	Critical	H	S	Best=0.8

## 6. Conclusion

In cloud computing, service providers always want to provide reliable services to customers or service consumers. However, there are obstacles between service providers and consumers. Customers need customized services with various configurations, these customizations and configurations are error free and seamless functionality during service execution. In software as a service, monitoring of virtual resources are hard to control the stability of their service, especially from end to end service provision.

## References

[1] G. G. Castañé, H. Xiong, D. Dong, and J. P. Morrison, "An ontology for heterogeneous resources management interoperability and HPC in the cloud," *Futur. Gener. Comput. Syst.*, vol. 88, no. 2018, pp. 373–384, 2018.

[2] J. B. Hong, A. Nhlabatsi, D. S. Kim, A. Hussein, N. Fetais, and K. M. Khan, "Systematic Identification of Threats in the Cloud: A Survey," *Comput. Networks*, vol. 150, pp. 46–69, 2018.

[3] B. L. Muhammad-Bello and M. Arisugi, "A transparent approach to performance analysis and comparison of infrastructure as a service providers," *Comput. Electr. Eng.*, vol. 69, pp. 317–333, 2018.

[4] S. J. E. Taylor *et al.*, "The CloudSME simulation platform and its applications: A generic multi-cloud platform for developing and executing commercial cloud-based simulations," *Futur. Gener. Comput. Syst.*, vol. 88, pp. 524–539, 2018.

[5] O. Rolik, E. Zharikov, and S. Telenyk, "Microcloud-based architecture of management system for IoT infrastructures," *2016 3rd Int. Sci. Conf. Probl. Infocommunications Sci. Technol. PIC S T 2016 - Proc.*, pp. 149–151, 2017.

[6] B. C. Keegan and E. S. G. S. Analyst, "Hyperconverged Transformation : Getting the Software - defined Data Center Right," no. February, 2016.

- [7] N. Zacheilas, N. Chalvantzis, I. Konstantinou, V. Kalogeraki, and N. Koziris, "ORiON: Online ResORce Negotiator for Multiple Big Data Analytics Frameworks," *2018 IEEE Int. Conf. Auton. Comput.*, pp. 11–20, 2018.
- [8] C. Zhang, J. Bi, Y. Zhou, and J. Wu, "HyperVDP: High-Performance Virtualization of the Programmable Data Plane," *IEEE J. Sel. Areas Commun.*, vol. PP, no. c, pp. 1–1, 2019.
- [9] I. Principles, "Putting It All Together :Continued," *Director*, pp. 126–130, 2014.
- [10] B. M. Leone and E. S. G. S. Analyst, "Performance Evaluation Criteria for Hyperconverged Infrastructures," no. November, 2016.
- [11] J. Chen *et al.*, "WaaS: Wisdom as a service," *IEEE Intell. Syst.*, vol. 29, no. 6, pp. 40–47, 2014.
- [12] C. O. M. M. I. S. S. I. O. N. E. D. By, "All-Flash Datacenter — Hyperconvergence and."
- [13] C. Melo *et al.*, "Availability models for hyper-converged cloud computing infrastructures," *12th Annu. IEEE Int. Syst. Conf. SysCon 2018 - Proc.*, pp. 1–7, 2018.
- [14] S. Park, B. Cha, and J. Kim, "Preparing and Inter-Connecting Hyper-Converged SmartX Boxes for IoT-Cloud Testbed," *Proc. - Int. Conf. Adv. Inf. Netw. Appl. AINA*, vol. 2015-April, pp. 695–697, 2015.
- [15] X. Li, Y. Liu, R. Kang, and L. Xiao, "Service reliability modeling and evaluation of active-active cloud data center based on the IT infrastructure," *Microelectron. Reliab.*, vol. 75, pp. 271–282, 2017.
- [16] I. Andrusiak and Q. H. Mahmoud, "A Reliability-Aware Framework for Service-Based Software Development," 2017.
- [17] T. Alyas and M. S. Khan, "Intelligent reliability management in software based cloud ecosystem using AGI," vol. 17, no. 12, pp. 134–139, 2017.
- [18] A. Avižienis, J. C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Trans. Dependable Secur. Comput.*, vol. 1, no. 1, pp. 11–33, 2004.
- [19] S. M. Babu, A. J. Lakshmi, and B. T. Rao, "A study on cloud based Internet of Things: CloudIoT," *2015 Glob. Conf. Commun. Technol.*, no. Gcct, pp. 60–65, 2015.
- [20] L. Wang, Y. Ma, J. Yan, V. Chang, and A. Y. Zomaya, "pipsCloud: High performance cloud computing for remote sensing big data management and processing," *Futur. Gener. Comput. Syst.*, vol. 78, pp. 353–368, 2018.
- [21] S. A. Miller, O. Behalf, and C. America, "CASE STUDY : HYPERCONVERGENCE VS CLOUD."
- [22] A. N. Toosi and R. Buyya, "A Fuzzy Logic-Based Controller for Cost and Energy Efficient Load Balancing in Geo-distributed Data Centers," *Proc. - 2015 IEEE/ACM 8th Int. Conf. Util. Cloud Comput. UCC 2015*, pp. 186–194, 2015.
- [23] R. E. Spiridonov, V. D. Cvetkov, and O. M. Yurchik, "Data Mining for Social Networks Open Data Analysis," pp. 395–396, 2017.