Modeling Fuzzy Scheduling in Infrastructure as a Service Cloud

Anita Prakash Patil
Research Scholar
JJTU University, Rajasthan

Haresh Chaudhari
Associate Professor
Department of Mathematics
Pratap College Amalner

ABSTRACT
This work addresses the results of lease scheduling in the Infrastructure as a Service (IaaS) Cloud in the uncertain conditions, and use of Fuzzy Logic for the same. The use of Fuzzy Control Logic (FCL) for the scheduling policies defined by the IEC standard is converted and final defuzzifed values have been used for lease scheduling in IaaS Cloud. The JFuzzyLogic library, have been used for solving this problem in the focus.

General Terms
Cloud Computing, Scheduling.

Keywords
Fuzzy logic, Center of Gravity, Scheduler, Hypervisor, JFuzzyLogic.

1. INTRODUCTION
The high capacity computing units like Super Computers, Clusters are need to be virtualized and necessarily served to the Consumers with their specific requirements, because perhaps resources underutilized. Different hypervisors are designed by the open source community and utilized by commercial Cloud players like AWS Amazon, and Open Source Software like OpenNebula (ONE) and Nimbus. Researchers tried the Mathematical modeling of the virtualized, hypervisor based infrastructure service is done. The partial allocation, full allocation of resources, changing scenarios in the virtualized environment due to consumers should be shown mathematically so that the definiteness can be introduced.

The "uncertainty" in the consumer's requirement needs to be modeled, because it affect the allocation of resources. The modeling of the deployment methodology on virtualized infrastructure and same scenario happened in the different flavors of hypervisors, and integration with sophisticated environment is highly required.

For certain parallel applications i.e. long running task, the set of machines to be launched (in a cluster) are represented in the form of "leases". These future requirements could be registered by the users (consumer) "in-advance". However, in dynamic load conditions, it perhaps limits the decision power of the scheduler, in IaaS Cloud environment. Uncertainties due to consumer's requirements create the complex situation in the lease processing, and it does not fit into the precise categories of the conventional set theory. The terms like "almost", "Partially" are modeled using the fuzzy design system with our conventional development. Some kind of work is done by the Desktop Cloud Developers [9], using VirtualBox. Further the scheduler can be made powerful using the Fuzzy Logic technique.

It is difficult to decide in case of complex situations like, if the resources are available but owner has not permitted to use the resource. In addition, if the application execution by the consumer is seems to be very urgent, then scheduler need to balance "Satisfaction" level of the both sides. Hence fuzzy logic helps to decide launching possibility. Form the matrix of values the rule based knowledge base is produced. The values are generated using the triangular, trapezoidal functions available with fuzzy logic supporting tools.

The unpredictable uncertainties due to the owners behavior and consumers requirements, creates complex situations. The mathematical modeling using fuzzy logic is done in this work. The results of the work can be easily plugged up with the existing system of the Cloud infrastructure.

2. BACKGROUND
Alaa et. al. [1] discussed allocation of idle generic resources to maximize the Cloud computing QoS at low cost. The middleware represents a bridge from OpenStack Cloud computing platform to generic resources donated by the on-premises desktop machines. The service elasticity and scalability through using both of the dedicated and non-dedicated infrastructures is achieved.

CloudSim Tool is used by few researchers, including Bala et. al. [4] used CloudAnalyst, which is a CloudSim based tool for modeling and analysis. The performance analysis of load balancing policies in combination with different broker for large-scaled applications using different infrastructural environments has been done. The large scale Cloud computing environments and experimental results reveal that large scaled software systems. This can minimize their costs and improve service quality.

Amin et. al, [3] considers CloudSim simulator for fuzzy logic based scheduling of virtual machine (VM) between the data centers. The effectiveness of algorithm and its comparison like scheduling with the first fit (FCFS) and round robin had been performed. Choudhury et. al, [6] hybrid Cloud model for the user requirements like request of VM on the data centers as per the need of users, consisting of resources (mips, VM image size, network bandwidth, number of CPU's) containing Cloudlets which in turn are Cloud-based application services. The priority mechanisms namely low, medium and high priorities are considered. The Java based CloudSim simulation tool has been used for the implementation purpose. These considerable significant factors useful for this kind of work.

K. Dinesh et. al, [11] focuses on job scheduling using Berger model and Neural Network for task resource matching based on different parameter s like bandwidth, memory, Completion time and Resources Utilization. The classified user tasks are passed to the neural network consists of input layer, hidden layer and output layer. With the help of hidden layer, the jobs are matched with the resources by adjusting weight. The performance of the system has been improved by means of efficient use of bandwidth, reducing a completion time which in turn improves resources utilization, using CloudSim Tool.

Arunkumar et. al. [2] proposed the algorithm based dependency and deadlock avoidance (De-De algorithm), based on Banker's algorithm for workflow-based applications. Many applications need for high processing and storage capacity without any deadlocks between those instances. The algorithm makes distinction between the active and passive resources. The high concurrency is achieved for the performance improvement.

Bharati et. al., [5] discussed the Cloud security aspects with integrity verification using web server in multi-cloud environment. The use of third party is introduced, which provides verification and confidentiality. Hashemi Sajjad [13] discussed about the data position, data separation, and flexibility of servers. The geographical place, the separate data in the Cloud, and the customer notification during the flexibility constraints shows some security concerns in the Clouds.

Devare M, Mehdi SheikhaliShai & Grandinetti et. al. [9][10] have done efforts towards mathematical modeling the scheduling schemes. Furthermore, the model of improving accessibility of Cloud and evaluating user's satisfaction using if-then concept of fuzzy linguistic Inference system. The complexity of lease processing and managing data and flexibility to access data is achieved using fuzzy system.

Florence et. al., [12] discussed the load balancing algorithm providing fairness to all the jobs in Cloud. The different parameters like RAM size, bandwidth and image size are used to determining a balance threshold value of each VM for scheduling the jobs.

Hirai et. al. [14] elaborated that the performance of the system can be measured for getting better response. The slow worker node can be get be allocated with the task like backups. The task is split up into the more than two sub tasks. The scheduling is considered as two scheduling tasks. i.e. normal processing model and backup task processing model. In each model the system is having the infinite buffer, Poisson arrival has been discussed.

3. METHODOLOGY

The Integrated Development Enviornment NeatBeans 6.8, and recent version of JFuzzyLogic is used. The work has base of Desktop Clouds [9] developed using VirtualBox 3.1.6 , which internally supported the use of Java. Following actions have been taken step by step for the use of the Fuzzy rule building. The aggregation, activation and accumulation methods are incorporated in the JFuzzyLogic, for processing the antecedents and consequents from the If-Then rules. The Defuzzification technique and uses the center of gravity, center of gravity singleton, center of area, rightmost max, leftmost max, and mean max. The necessary steps to be done during this work are as mentioned below.

1. Find the current condition of the available resources.
2. Create the fuzzified values and pass it to the fuzzifier
3. Define and pass the input of the linguistic terms and values to Fuzzy System.
4. Decision making using the knowledge base of rule blocks.
5. The defuzzified values, the digital values should be passed to the lease scheduling

The JFuzzyLogic [7][8] library used for the fuzzification, defuzzification and rule blocks are written using simple English constructs as described here. The International Electrotechnical Commission (IEC 61131-7) has been used. A Fuzzy Logic Controller (FLC) has been used as shown below.

```
FUNCTION_BLOCK result_occuppier
VAR_INPUT
        OWNER_CONDITIONS : REAL;
        OCCUPIED_CAPACITY : REAL;
END_VAR

VAR_OUTPUT
        ALLOCATION_CHANCE : REAL;
END_VAR

FUZZIFY OWNER_CONDITIONS
{'low', 'medium' , 'high'}
        TERM low := (0, 1) (4, 0) ;
        TERM medium := (1, 0) (4,1) (6,1) (9,0);
        TERM high := (6, 0) (9, 1);
END_FUZZIFY

FUZZIFY OCCUPIED_CAPACITY
{ 'low', 'medium_low', 'medium_high','high' }
        TERM low := (0, 1) (1, 1) (3,0) ;
        TERM medium_low := (7,0) (9,1);
      TERM medium_high := (7,0) (9,1);
      TERM high := (7,0) (9,1);
END_FUZZIFY

DEFUZZIFY ALLOCATION_CHANCE

'ALLOCATION_CHANCE' : {'very_low', 'low',
'medium_low','medium_high','high', 'very_high' }
        TERM very_low := (0,0) (5,1) (10,0);
        TERM low := (10,0) (15,1) (20,0);
        TERM medium_low := (20,0) (25,1) (30,0);
         TERM medium_high := (20,0) (25,1) (30,0);
         TERM high := (20,0) (25,1) (30,0);
        TERM very_high := (20,0) (25,1) (30,0);

        METHOD : COA;
        DEFAULT := 0;
END_DEFUZZIFY

RULEBLOCK No1
        AND : MIN;
        ACT : MIN;
        ACCU : MAX;

        RULE 1 : IF OWNER_CONDITIONS IS low and
OCCUPIED_CAPACITY IS low THEN
ALLOCATION_CHANCE IS low;
        RULE 2 : IF OWNER_CONDITIONS IS high
AND OCCUPIED_CAPACITY IS low THEN
ALLOCATION_CHANCE IS very_high;
END_RULEBLOCK
```

END_FUNCTION_BLOCK

Here, the rule block shows here having only two rules, for the simplicity of the code. The separate rule block shown in consequent section. We are interested to create the knowledge base so that, the necessary decisions should be taken properly by the lease scheduler in the uncertain conditions. As OWNER_CONDITIONS having three terms "low" , "medium" and "high". The OCCUPIED_CAPACITY having four terms "low", "medium_low", "medium_high" and "high". The ALLOCATION_CHANCE having six terms "very_low", "low", "medium_low", "medium_high", "high" and "very_high". Following rules have been formed. The fuzzy logic file is mentioned in separate file "Cloud.fcl".

RULE 3 : IF OWNER_CONDITIONS IS low and OCCUPIED_CAPACITY IS medium_low THEN ALLOCATION_CHANCE IS medium_low;
RULE 4 : IF OWNER_CONDITIONS IS low and OCCUPIED_CAPACITY IS medium_high THEN ALLOCATION_CHANCE IS low;
RULE 5 : IF OWNER_CONDITIONS IS low and OCCUPIED_CAPACITY IS high THEN ALLOCATION_CHANCE IS very_high;
RULE 6 : IF OWNER_CONDITIONS IS medium and OCCUPIED_CAPACITY IS low THEN ALLOCATION_CHANCE IS high;
RULE 7 : IF OWNER_CONDITIONS IS medium and OCCUPIED_CAPACITY IS medium_low THEN ALLOCATION_CHANCE IS medium_high;
RULE 8 : IF OWNER_CONDITIONS IS medium and OCCUPIED_CAPACITY IS medium_high THEN ALLOCATION_CHANCE IS medium_low;
RULE 9 : IF OWNER_CONDITIONS IS medium and OCCUPIED_CAPACITY IS high THEN ALLOCATION_CHANCE IS low;
RULE 10 : IF OWNER_CONDITIONS IS high and OCCUPIED_CAPACITY IS medium_low THEN ALLOCATION_CHANCE IS high;
RULE 11 : IF OWNER_CONDITIONS IS high and OCCUPIED_CAPACITY IS medium_high THEN ALLOCATION_CHANCE IS medium_high;
RULE 12 : IF OWNER_CONDITIONS IS high and OCCUPIED_CAPACITY IS high THEN ALLOCATION_CHANCE IS low;

The Java Swing components have been used for the further automation to process the "Cloud.fcl" file. Following is the program code to execute automation part, and generate the graphical output. The graphs generated are shown as it is from the JFuzzyLogic.

```
private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
      String fileName = "Cloud.fcl";
      FIS fis = FIS.load(fileName,true);

      if( fis == null ) {
     System.err.println("Can't load file: '" + fileName + "'");
         return;
      }
         FunctionBlock funblk =fis.getFunctionBlock(null);

      JFuzzyChart.get().chart(funblk);
      funblk.setVariable("OWNER_CONDITIONS", 3);
      funblk.setVariable("OCCUPIED_CAPACITY",0);
      funblk.evaluate();

Variable hit =
funblk.getVariable("ALLOCATION_CHANCE");
      JFuzzyChart.get().chart(hit, hit.getDefuzzifier(), true);
      System.out.println("Result Value="+
hit.getLatestDefuzzifiedValue());
      JFuzzyChart.get().chart(funblk);
      double
dd=funblk.getVariable("ALLOCATION_CHANCE").
getValue();
     System.out.println("Result Occupy"+dd);
     System.out.println(fis);
    }
```

The processed input values can be used to bring VMs up and down (using power down and power up). The VirtualBox is the choice to do the operations on the VM

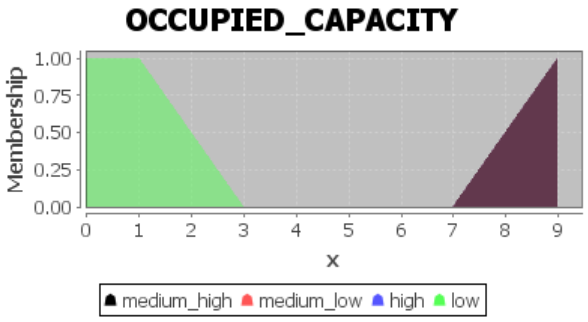

Fig.1: The current Resource Occupied Capacity is considered using four different terms.

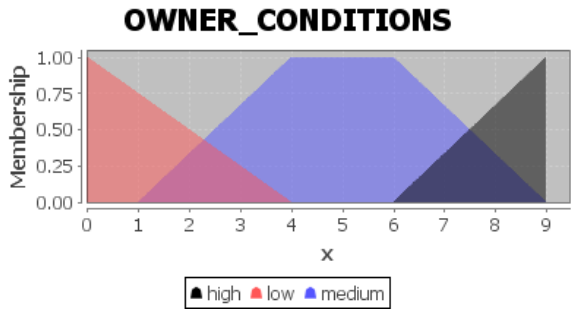

Fig. 2: The Owner Permissions for Resource Allocation expressed using three different terms

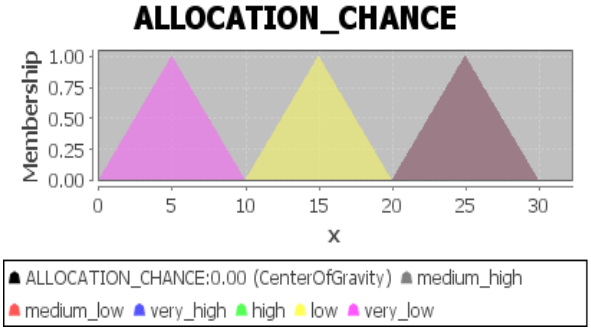

Fig. 3: Logical Resource in terms of VMs Allocation Chance using COG

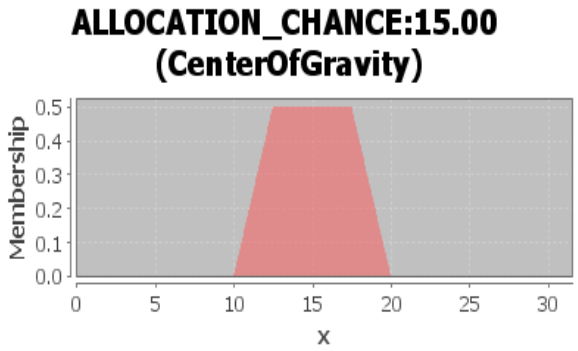

Fig. 4: Use of Center of Gravity for calculating the specific inputs conditions.

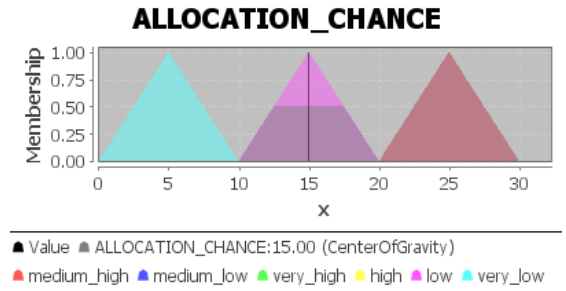

Fig. 5: Owner Conditions value as three and Resource Occupied Capacity as zero i.e. empty.

The Figure 1, Figure 2, figure 3, Figure 4 and Figure 5 are shown as it is generated from the JfuzzyLogic. Hence not shown with separate graph generation tool.

4. CONCLUSION

The use of the rule based knowledge wrote using the FCL and mathematical solutions using various functions like COG, Sigmoid useful for the decision making in the uncertain conditions. The use and testing of the several options in the fuzzy system is possible in the said direction of the work. The preliminary results have been presented here. The choice of the different functions in the proposed work of the Cloud Lease Processing is possible. The linguistic terms and their fuzzification are achieved here. The simple rule based knowledge base is possible to built up (as shown the sample code in this work) and the de-fuzzified values can be fed-up to the Cloud Scheduler.

ACKNOWLEDGEMENTS:

We are thankful to Dr. Manoj Devare, Associate Professor at PK Technical Campus, for valuable guidance during the use of Fuzzy logic in the combination of the open source Java execution environment.

5. REFERENCES

[1] Alaa Eldeen S Ahmed, Abdulwahab K Alsammak and Essam Algizawy (2013) "A New Approach to Manage and Utilize Cloud Computing Underused Resources", International Journal of Computer Applications 76(11):29-36, Published by Foundation of Computer Science, New York, USA.

[2] Arunkumar B and T Ravichandran. (2013) "De-De Dodging Algorithm for Scheduling Multiple Workflows in Hybrid Cloud", International Journal of Computer Applications 79(15):5-9, Published by Foundation of Computer Science, New York, USA.

[3] Amin Mehranzadeh, Seyyed Mohsen Hashemi, (2013) "A Novel-Scheduling Algorithm for Cloud computing based on fuzzy logic", International Journal of Applied Information Systems (IJAIS), Volume 5– No.7.

[4] Bala Minu and Devanand (2013) "Performance Evaluation of Large Scaled Applications using Different Load Balancing Tactics in Cloud Computing" , International Journal of Computer Applications 76(14):17-22, Published by Foundation of Computer Science, New York, USA.

[5] Bharati Vaibhav and M R Patil (2013) "Advanced Cooperative Provable Data Possession based Data Integrity Verification for Multi-Cloud Storage" , International Journal of Computer Applications 81(13):25-28, Published by Foundation of Computer Science, New York, USA.

[6] Choudhury Kaustav, Diptam Dutta and Kasturi Sasmal (2013) "Resource Management in a Hybrid Cloud Infrastructure", International Journal of Computer Applications 79(12):41-45, Published by Foundation of Computer Science, New York, USA.

[7] Cingolani, Pablo, and Jesús Alcalá-Fdez. "jFuzzyLogic: a Java Library to Design Fuzzy Logic Controllers According to the Standard for Fuzzy Control Programming" , International Journal of Computational Intelligence Systems, (2013), pp. 61-75.

[8] Cingolani, Pablo, and Jesus Alcala-Fdez. "jFuzzyLogic: a robust and flexible Fuzzy-Logic inference system language implementation." Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on. IEEE, 2012.

[9] Devare M., Grandinetti L., M. Sheikhalishahi (2010) "Prototype Implementation of Desktop Clouds", Published by IOS press, Netherland, presented in Cetraro International workshop on HPC, Grids and Clouds. [21-25th June 2010]

[10] Devare M., Anita Devare, Harish Chaudhari, Handling Uncertain leases in Infrastructure as a Service Clouds, International Journal of Computer Applications.

[11] Dinesh K., G. Poornima, K. Kiruthika (2012) "Efficient Resources Allocation for Different Jobs in Cloud", International Journal of Computer Applications, Volume 56– No.10, October 2012.

[12] Florence Paulin A and V Shanthi (2013) "Intelligent Dynamic Load Balancing Approach for Computational Cloud", IJCA Proceedings on National Conference on Recent Trends in Computer Applications NCRTCA: 15-18, Published by Foundation of Computer Science, New York, USA.

[13] Hashemi Sajjad, (2013), "Data Storage Security Challenges in Cloud computing", International Journal of Security, Privacy and Trust Management (IJSPTM) Vol. 2, No. 4. PP.1-10.

[14] Hirai Tsuguhito et. al, (2010) "Performance analysis of large-scale parallel-distributed processing with backup tasks for cloud computing", Journal Of Industrial And Management Optimization, Vol. 10, No. 1.