ATLANTIS PRESS
*Atlantis Studies in Uncertainty Modelling, volume 1*
11th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT 2019)

# Type-2 Fuzzy Logic Approach for Overloaded Hosts in Consolidation of Virtual Machines in Cloud Computing

**Bruno M. P. Moura**†, **Guilherme B. Schneider**†,
**Adenauer C. Yamin**†, **Mauricio L. Pilla**† and **Renata H. S. Reiser**†
†Federal University of Pelotas (UFPEL)
Centre for Technological Development (CDTEC),
Laboratory of Ubiquitous and Parallel Systems (LUPS)
Rua Gomes Carneiro, 1 – 96010-610 – Pelotas – Brazil
Email: {bmpdmoura, gbschneider, adenauer, pilla, reiser}@inf.ufpel.edu.br

## Abstract

Dynamic consolidation of virtual machines (VMs) is an effective way to improve resource utilization and power efficiency in cloud computing. Determining when it is best to relocate VMs from an overloaded host is one aspect of dynamic VM consolidation that directly influences the resource utilization and Quality of Service (QoS) offered by the system. This paper presents a new proposal with a Type-2 Fuzzy Logic approach to address the uncertainties and inaccuracies in determining resource usage, aiming at energy savings with minimal performance degradation. Results in a simulated cloud computing environment show improvements in energy efficiency of 15.64% in the Median Absolute Deviation (MAD), 15.43% in the Inter Quartile Range (IQR), and 9.19% in the Local Regression (LR).

**Keywords:** Type-2 Fuzzy Logic, Cloud Computing, Green Computing.

## 1 Introduction

The computational infrastructures commonly used to run applications in Computational Clouds use features such as data centers that are known to consume large amounts of energy, leading to high operating costs for cloud providers negatively and contributing to the environment [21].

According to the USA Natural Resources Defense Council's (NRDC)[1] report in 2014, USA data centers consumed an estimated 70 billion kilowatts/hour (KWh), representing about 1.8% of total USA's consumption. Based on current trend estimates, it is expected that USA's data centers will consume approximately 73 billion kWh by 2020 [19].

[1]https://www.nrdc.org/

Therefore, improve energy efficiency while maintaining good Service Level Agreement (SLA) and Quality of Service (QoS) levels is an important challenge to be addressed through load balancing in clouds. The problem of minimizing energy consumption over QoS constraints is analytically complex as a whole and is part of the dynamic consolidation of VM, which is an NP-Hard problem [9, 3].

For the load balancing among data centers of clouds, the factors impacting uncertainty include Computational Power (CP), Communication Cost (CC), and the use of Random Access Memory (RAM).

The following research questions stand out: (i) do load balancers consider the uncertainties associated with factors such as CP, CC, and RAM? (ii) are the load balancers equipped with consistent techniques for the treatment of the uncertainties related to cloud computing environments? and (iii) do the users have the necessary knowledge about the computational and communication demands of their applications?

This proposal extends the approaches in [16, 15], modeling the uncertain variables generating CP, CC, and RAM in Load Balancing (LB) considering Type-2 Fuzzy Logic (T2FL) which can preserve the balance between energy efficiency and manage resources while ensuring QoS through SLAs for such cloud-based services.

The paper is structured as follows. Related works are presented in section 2. Section 3 introduces basic concepts of Type-2 Fuzzy Logic. In section 4, details of the *Int-FLBCC* component and its conception are discussed, including database, fuzzification, rule base, inference, and defuzzification. Section 5 describes the experimental evaluation. Finally, section 6 presents conclusions and future work.

## 2 Related Works

Six related works using FL are briefly presented in this section, discussing distinct methods to deal with uncertainties in LB and illustrating the need for preventive handling of uncertainties in the host selection step in VM allocation in a cloud computing environment.

(1) Toosi and Buyya [21] propose the efficient use of renewable energy sources through the Fuzzy Logic approach to treat uncertainties based on the intermittent and unpredictable power availability due to factors such as climatic variations and energy price at the data center's hosting site. The central goal of the work is to provide cloud providers with multiple geographically distributed data centers in a region, the treatment of temporal variations and the price of network energy on the spot by routing the load to a data center to reduce costs and increase the use of renewable energy. To reach this objective, a load-balancing algorithm based on fuzzy logic has been proposed.

In order to achieve their goals, the fuzzy inference mechanism considers: (i) the use of renewable energy sources $U_i$, (ii) the amount of conventional network energy consumption $B_i$ and (iii) the average price of local electricity of the data center calculated within a time window $F_i$, generating as output the adequacy of the data center to receive the workload. Tests were based on real-world traits obtained from the National Renewable Energy Laboratory (NREL), the US Energy Information Administration (EIA) and through Google's cluster usage information. Compared to other benchmark algorithms, this method was able to reduce costs without prior knowledge of the future price of electricity, renewable energy availability, and workloads.

(2) Portaluri et al. [17] implement different allocation algorithms based on a Fuzzy Logic approach for single or multi-objective, two variants of a multi-objective heuristic, and an analytic approach. The primary objective of the work is the focus on the routing of packages aiming at energy efficiency. VM requests are defined based on the computational resource variables of the system which are: CPU, RAM, Disk, and network bandwidth. Tests for performance evaluation of scheduling were performed providing results about the number of VMs allocated for each policy. The results showed that the multi-objective approach could assign the most significant amount of VMs, on average, having the narrowest distribution when Worst Fit (WF) is adopted.

(3) Singh and Chana [20] propose a framework called Energy-aware Autonomic Resource Management TecHnique (EARTH) presented for scheduling resources based on Fuzzy Logic to obtain energy efficiency. The fuzzy inference system considered the following variables: Workload Waiting Time (WWT), Workload Execution Time (WET) and Resource Energy Consumption (REC), thus generating the Workload Processing Priority (WPP) output. The evaluation of the proposed EARTH framework was based on a simulation using CloudSim and a real computational cloud environment from Thapar University, India (Thapar Cloud). The experimental results show that the proposed framework performed better regarding resource utilization and energy consumption, along with other QoS parameters.

(4) Salimian et al. [18] developed an algorithm based on adaptive fuzzy limits designed to detect overloaded and under-loaded hosts. Their objectives included reducing energy consumption, SLA violations, and the number of migrations. The metrics are energy consumption, SLA violation, number of migrations, and workload data which define the value of resource use. The proposed algorithm dynamically generates rules and updates membership functions to adapt to changes in the workload. It was validated with a real workload from PlanetLab. The simulation results demonstrated that the proposed algorithm significantly outperforms the other competitive algorithms.

(5) Arianyan et al. [1] not only proposed with Dynamic Voltage and Frequency Scaling (DVFS) to eliminate the inconsistencies between consolidation techniques and DVFS but also introduced new DVFS algorithms for the four sub-problems of the online consolidation problem, as well as a new DVFS controller. The four consolidation sub-problems are (i) determination of overloaded Physical Machines (PMs), (ii) determination of under-loaded PMs, (iii) selection of VMs that should be migrated from overloaded PMs, and (iv) placement of migrant VMs in PMs [2]. This work considers the crucial criteria being CPU, RAM, and network bandwidth in all proposed algorithms. They also introduced a new multi-functional allocation algorithm enabling resource managers to apply the importance of different criteria in the resource management solution using fuzzy weights. The results of experiments obtained from an extensive evaluation of the policies proposed in the CloudSim tool showed that the proposal exceeded the management of resources of existing solutions due to the simultaneous optimization of essential criteria in the decision-making process. In this research, it was concluded that the combination of the proposed policies for the resource management process in cloud data centers obtained a notable reduction in the metrics of energy consumption, SLA violation and number of migrations compared to state of the art.

(6) Haratian et al. [8] address the issue of how to

reduce the number of SLA violations based on the optimization of resources allocated to users, applying an autonomous control cycle and a fuzzy knowledge management system. A framework called Adaptive and Fuzzy Resource Management (AFRM) is proposed in which the last values of resources of each VM are gathered through the sensors of the environment and sent to a fuzzy controller. The variables used for the fuzzy inference system are utility and scale, to propose the dynamic coefficient based on the workload of the computational cloud environment. AFRM decides how to allocate resources at each iteration of a self-adaptive control cycle. Every membership functions and fuzzy rules are dynamically updated based on workload changes to meet QoS requirements. Two sets of experiments were conducted, evaluating the AFRM compared to rule-based approaches and static controls regarding Resource Allocation Efficiency (RAE), utility, some SLA violations and cost by applying HIGH, MEDIUM, MEDIUM-HIGH, and LOW workloads. The tests considered the use of JFuzzyLogic as an alternative for analysis and modeling of the fuzzy inference system and the CloudSim for simulation. The results showed that the AFRM overcomes fuzzy approaches based on rules and static rules in several aspects.

This paper provides an approach with Type-2 Fuzzy Logic to address uncertainties and dynamic behavior in the evaluation of the level of use of the host for load balancing VM allocation in cloud environments. The proposal was validated through simulations in the CloudSim 3.0.3 framework.

Table 1 shows that CloudSim and Matlab are the most used tools, the number of input variables is usually 3, and the number of outputs is usually 1. The most used fuzzification method is Triangular Membership Functions, the inference is a tie between Takagi-Sugeno and Mandani, deffuzification is made with diverse methods, and the most used connective is AND.

## 3 Foundations of Fuzzy Logic

Lotf Zadeh introduced T2FL in 1975 [24] as an extension of the traditional FL modeling the inherent uncertainties related to the antecedent and consequent membership functions, enabling the manipulation of imprecise terms throughout its fuzzy inference system [14].

Type-2 fuzzy sets emerged from the observation that, in many applications, when no objective procedure is available to select the crisp membership degree $\mu_A(x)$ of an element $x \in \chi$ in a fuzzy set $A$, meaning that it is not a single real value [11]. Such sets can be used in situations where there exists uncertainty about the degrees, forms or parameters of the membership functions [10], providing potential strategy on the treatment of uncertainties in information models based on multiple-criteria obtained from distinct specialists and/or extracted from simulators.

In this proposal, Interval-valued Fuzzy Logic (IvFL) is considered based on Interval-valued Fuzzy Set (IvFS) theory, suggesting to alleviate that problem by allowing to specify only an interval $X = [\underline{\mu_A(x)}, \overline{\mu_A(x)}]$ as the membership degree of such element $x$ in an fuzzy set $A$ [7]. Thus, by complementing FS theory, IvFS theory can model vagueness with an additional ability to consider imprecision (non-specificity) as two important aspects of uncertainty, reflecting this uncertainty by the length of the interval membership degree.

According to [12], let $\mathbb{U}$ be the set of all real intervals in the unitary interval $U = [0, 1]$ and the partial order: *Product order*: $X \leq Y$ iff $\underline{X} \leq \underline{Y}$ and $\overline{X} \leq \overline{Y}$.

By [7], a function $\mathbb{T}(\mathbb{S}) : \mathbb{U}^2 \to \mathbb{U}$ qualifying fuzzy union, is an *interval-valued t-norm* (*t-conorm*) if it is commutative, associative, monotonic w.r.t. the product order and has $\mathbf{1} = [1, 1]$ ($\mathbf{0} = [0, 0]$) as the neutral element.

An interval function $\mathbb{N} : \mathbb{U} \to \mathbb{U}$ is an *interval-valued fuzzy negation* if, for all $X, Y \in \mathbb{U}$, it holds that: $(i)$ $\mathbb{N}1$: $\mathbb{N}(\mathbf{0}) = \mathbf{1}$ and $\mathbb{N}(\mathbf{1}) = \mathbf{0}$; $(ii)$ $\mathbb{N}2$: If $X \geq Y$ then $\mathbb{N}(X) \leq \mathbb{N}(Y)$; and $(iii)$ $\mathbb{N}3$: If $X \subseteq Y$ then $\mathbb{N}(X) \subseteq \mathbb{N}(Y)$.

A system based on IvFL can estimate input and output functions by using heuristic and interval techniques. See Figure 1, graphically presenting the architecture of the inference system based on T2FL. In the following, its main blocks are briefly described:

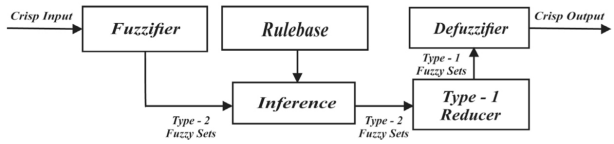



Figure 1: Type-2 Fuzzy Inference System Architecture

1 **Fuzzification Interface**: The fuzzification process based on IvFL is performed according to the nature and definition of such type-2 set, associating an input value with an interval function and not simply with a single value of $U$. In other words, it is inserted into the mechanism of inference the uncertainty regarding the input membership function. Thus, for each IvFS $A$, the transformation of an input vector $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \chi^n$ when $n \in \mathbb{N}$, to a pair of vectors in $\mathbb{U}^n$ is given by as follows:

$$\left(\overline{\mu_A(x_1)}, \overline{\mu_A(x_2)}, \ldots, \overline{\mu_A(x_n)}\right), \left(\underline{\mu_A(x_1)}, \underline{\mu_A(x_2)}, \ldots, \underline{\mu_A(x_n)}\right).$$

Table 1: Comparison of related works considering used tools, number of variables, Fuzzification methods, Inference, Defuzzification, and Connectives used in each application.

| Work | Tools | In/Out | Fuzzification | Inference | Defuzzification | Connectives |
|---|---|---|---|---|---|---|
| (1) | ♠ | 3/1 | Tri MF | Ma | Centroid | AND |
| (2) | ♢ ♣ | 3/1 | Tri/Tra MF | TS | CoG | AND |
| (3) | ♠ | 3/1 | Gau MF | Ma | Maximum | AND |
| (4) | ♠ △ | 2/1 | Tri MF | TS | CoG | AND |
| (5) | ♠ | 8/1 | Tri MF | TOPSIS | TOPSIS | TOPSIS |
| (6) | ♠ △ | 2/1 | Tri MF | TS | CoG | AND |
| *Int-FLBCC* | ♡ ♢ ♠ | 3/1 | Tra MF | Ma | CoA | AND |

♠ CloudSim ♢ Matlab ♣ Simulink △ JFuzzyLogic ♡ Juzzy Gaussian MF (Gau MF) Triangular MF (Tri MF) Trapezoidal MF (Tra MF) Takagi-Sugeno (TS) Mandani (Ma) Center of Gravity (CoG) Center of Area (CoA) Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS)

2 **Rule Base (RB)**: Composed of rules classifying linguistic variables (LVs) according to the IvFSs;

3 **Logic Decision Unity**: Executing inference operations between the input data and the rules defined in the RB to obtain performance by the system action;

4 **Defuzzification**: Considering two main stages of IvFSs:

*(i)* **Type Reducer** has the function of transforming an IvFSs into fuzzy sets, that is, it tries the best fuzzy set that represents the type-2 fuzzy set, and that must satisfy the following premise: When all uncertainties disappear, the result of System Based on Fuzzy Rules 2 (SBFR2) is reduced to a System Based on Fuzzy Rules 1 (SBFR1) [23];

*(ii)* **Defuzzification**: Providing an output given as the average of limits points $y_L$ and $y_R$:

$$y(x) = \frac{y_L + y_R}{2}, \forall x \in \chi, \tag{1}$$

when values $y_L$ and $y_R$ can be calculated using the iterative method of Karnik and Mendel (KM algorithm). Therefore, defuzzification step can still be obtained through the use of a conventional method such as the centroid, resulting in the final value of an inference system performance.

## 4 Modeling Fuzzy System

The *Int-FLBCC* is responsible for verifying host level of use on load balancing for computational clouds. The *Int-FLBCC* system considers a Rule Base acting on three steps: Fuzzification, Inference, and Defuzzification, returning as output the utilization level of each host. The modeling of the type-2 fuzzy system was performed using the Interval Type-2 Fuzzy Logic System Toolbox (IT2FLT) module [5, 6], and Juzzy [22].

### 4.1 Data Base - Membership Functions

Through the study of variables with a specialist, each one of LVs was associated with four distinct FSs, using the trapezoidal graphical representation to corresponding membership functions.

A reading of the settings of the simulated cloud computing environment is performed to measure CP, CC, and RAM. These values are then applied to a standard scale adopted, considering the interval $[0; 10]$, for CP Eq.(2), CC Eq.(3) and RAM Eq.(4) to obtain their degrees of membership.

$$CP = (h_i(MM)/MaxCP * 10) \tag{2}$$

$$CC = ((10 * h_i(UoB))/MinCC) \tag{3}$$

$$RAM = (h_i(UoR)/MaxRam) * 10 \tag{4}$$

- where $h_i$ represents the analyzed $host(i)$ of the cloud environment;
- $MM$ Maximum Million Instructions per Second (MIPS) available on host $i$ among all Processing Element (PEs);
- $UoB$ Utilization of host bandwidth $i$;
- $UoR$ Memory usage of host $i$;
- $MaxCP$, the highest MIPS total value of the best host in the cloud environment;
- $MinCC$, the lowest communication cost of the cloud environment, and
- $MaxRAM$, the value of the RAM capacity of the best host.

In the cloud environment, the level of use of the hosts is uncertain and imprecise due to several factors, such as the fluctuation of computational power, bandwidth

and available memory at the moment of the execution of the applications submitted by the users. Consider an online algorithm applying the Type-2 Fuzzy Logic approach that traverses the available hosts in the cloud architecture by obtaining the level of use at each iteration. The algorithm returns the evaluated host utilization level based on a fuzzy type-2 inference system (Algorithm 1).

```
input  : Host h_i
output: Level of use of the host i
for i ← 0 to getHostList().size() do
    if maxCPHost <
     getHostList().get(i).getTotalMips() then
        maxCPHost ←
         getHostList().get(i).getTotalMips();
    end
    if minCCHost <
     getHostList().get(i).getBw() then
        minCCHost ←
         getHostList().get(i).getBw();
    end
    if maxRamHost <
     getHostList().get(i).getRam() then
        maxRamHost ←
         getHostList().get(i).getRam();
    end
end
EvaluationFuzzy f ← EvaluationFuzzy();
Utilization ← f.getLevelOfUse(CP, CC, RAM);
return Utilization;
```

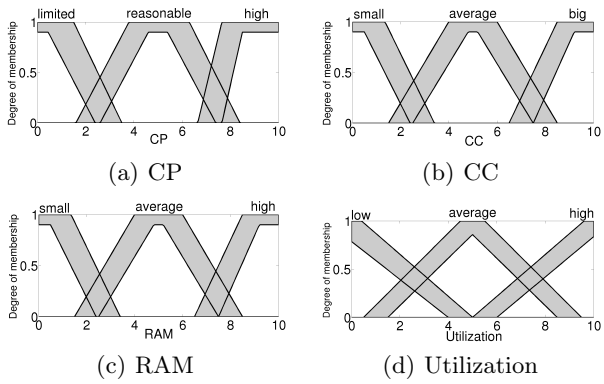
**Algorithm 1:** Host use level assessment

The Linguistic Terms (LTs) defining the FSs of this variable CP are stated as follows: "Limited" (CPL), "Reasonable" (CPR) and "High" (CPH - best case). Being $CP = a$ and $a \in [0; 10]$. The Membership Functions are shown in Figure 2(a).

Communication is measured on each host that constitutes cloud computing. The LTs to the FSs defined for this variable are: "Small" (SCC - best case), "Average" (ACC) and "Big" (BCC). Being $CC = b$ and $b \in [0; 10]$. These membership functions are presented in Figure 2(b).

In the design of the FSs for RAM, the following LTs were created: " Limited " (RL), " Reasonable " (RR) and " High " (RH - best case). Being $RAM = c$ and $c \in [0; 10]$. These membership functions are presented in Figure 2(c).

The output (Utilization) of host is also adapted to a standard scale, and the LTs for FSs used are: "Low" (LU - best case), "Average" (AU) and "High" (HP). Being $U = d$ and $d \in [0; 10]$. These Membership Functions are presented in Figure 2(d).




(a) CP (b) CC (c) RAM (d) Utilization

Figure 2: CP, CC, RAM, and Utilization in the default scale

### 4.2 Fuzzification

At this stage, the input values (already set for an observed scale in the section 4.1) is mapped to the fuzzy domain as shown in Figure 3.

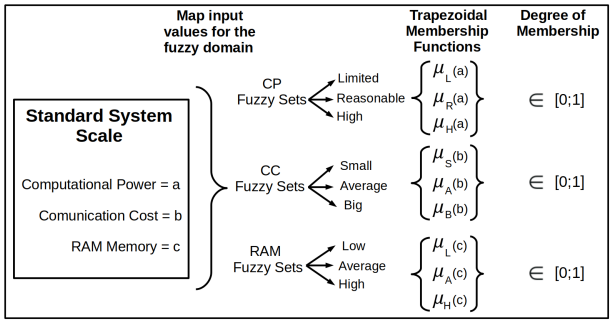



Figure 3: Fuzzification Process

### 4.3 Rule Base

The Rule Base (RB) of *Int-FLBCC*, is developed to be easily understandable and editable since there is no difficulty in adding new rules whether other input variables are desired to be manipulated. RB considers the performance of type 2 fuzzy system as subjected to the rules, describing the consistently control strategy [13]. It considers three factors for its construction:

- LV's name the FS's, turning the modeling closer to the real world system;
- The type "AND" connections are taken into account to create the relationship among the input variables;
- The type of implications *generalized modus ponens* (affirmative): "if X is A, then Y is B".

### 4.4 Inference

In the Inference process, the operations between the FSs occur, a combination of the antecedents of the rules and implications using the generalized modus ponens operator.

*(i)* Fuzzy Operation Application: in this stage, there is the application of fuzzy operators and the input consists of three values, resulting from fuzzification. The "AND" fuzzy operator forms the rules. The app uses the method MIN (minimum) on the three returned values of fuzzification;

*(ii)* Implication Fuzzy Method Application: this step performs a combination of the value obtained in the fuzzy operator applied and the values of FS output rule, using the method MIN (minimum) on these combinations;

*(iii)* Aggregation Fuzzy Method Application: In this stage, there are results composition of the fuzzy output of each rule by using the method MAX (maximum), thus creating a single fuzzy region to be analyzed by the next Fuzzy process module.

### 4.5 Defuzzification

With the research progress, the region transformation happens to be the result of the inference in a discrete value (which is the utilization). The defuzzification method used was the center of the area.

This method calculates the centroid (x) of the area consisted of the output of the fuzzy inference system (connection of all contributions rules stated in sections 4.3 and 4.4). The following formula calculates the centroid:

$$u = \frac{\sum_{i=1}^{N} u_i \mu_{OUT}(u_i)}{\sum_{i=1}^{N} \mu_{OUT}(u_i)} \tag{5}$$

## 5 Evaluation

The configuration of the test environment and the evaluation of the fuzzy module to detect the load level of hosts in the cloud employed was developed for CloudSim [4], a toolkit for modeling and simulating cloud computing services. An open-source Java library called Juzzy [22] is used to implement the fuzzy inference system.

For all experiments, real-world workloads provided from the PlanetLab infrastructure are used. The Infrastructure as a Service (IaaS) cloud environment represented for the tests considered a large-scale data center that comprised 800 heterogeneous physical hosts, contemplating two types of configurations as described in Table 2. The results of the runs, as well as the *Int-FLBCC*, are available in GitHub[2] in an extended version of CloudSim 3.0.3.

Table 2: Hosts of Cloud

| **Vendor** | **Model** | **CPU Name** | **CPU Characteristics** | **Memory** |
|---|---|---|---|---|
| Hewlett Packard Enterprise | ProLiant DL325 Gen10 | AMD EPYC 7551P 2.0 GHz | 32-Core, 2.0 GHz, 64MB L3 Cache | 128 GB |
| Dell Inc. | PowerEdge R840 | Intel Xeon Platinum 8180 2.50 GHz | 28 core, 2.50 GHz, 38.5 MB L3 Cache | 384 GB |

The frequency of server CPUs is mapped into MIPS classifications. Half of the hosts are the ProLiant DL325 Gen 10 with 4721 MIPS for each core, and the other half consists of the PowerEdge R840 server with 4520 MIPS for each core. Each server is modeled to have 5 GB/s of network bandwidth. Characteristics of VM types are similar to Amazon EC2 instance types, including Medium High CPU Instance (4000 MIPS, 32 GB); Extra Large Instance (3000 MIPS, 8 GB); Small Instance (2000 MIPS, 8 GB); and Micro Instance (2000 MIPS, 16 GB). The usage measurement interval (schedule interval) is 5 minutes. The characteristics of each workload are shown in Table 3. CPU load workload data for more than 1000 VMs of servers located in over 500 locations worldwide were used.

Table 3: Characteristics of the workload data

| **Workload** | **VMs** | **Mean** (%) | **St. dev.** (%) |
|---|---|---|---|
| 20110303 | 1052 | 12.31 | 17.09 |
| 20110306 | 898 | 11.44 | 16.83 |
| 20110309 | 1061 | 10.7 | 15.57 |
| 20110322 | 1516 | 9.26 | 12.78 |
| 20110325 | 1078 | 10.56 | 14.14 |
| 20110403 | 1463 | 12.39 | 16.55 |
| 20110409 | 1358 | 11.12 | 15.09 |
| 20110411 | 1233 | 11.56 | 15.07 |
| 20110412 | 1054 | 11.54 | 15.15 |

Nine sets of workload data collected on different days were applied, which are allocated to each VM. For all the experiments the allocation algorithms Inter Quartile Range (IQR), Inter Quartile Range Fuzzy (IQRF), Local Regression (LR), Local Regression Fuzzy (LRF), Local Regression Robust (LRR), Local Regression Robust Fuzzy (LRRF), Median Absolute Deviation (MAD), Median Absolute Deviation Fuzzy (MADF), Static Threshold (THR) and Static Threshold Fuzzy (THRF) were used. In each execution, the following selection policies of VMs have changed: Maximum Correlation (MC), Random Selection (RS), Minimum Migration Time (MMT), Maximum Correlation (MC).

[2] `https://github.com/brunomourapaz/CloudSim`

The results are presented graphically through boxplot, consisting of the following statistical information: (i) the minimum, (ii) the first quartile (Q1), (iii) the median, (iv) the third quartile (Q3) and (v) maximum. The lower and upper stems extend respectively from the lower quartile to the smallest value not lower than the lower limit, and from the upper quartile to the largest amount not exceeding the upper limit.

The energy consumption assessment is presented through the boxplot of Figure 4. It should be noted that for all test cases the proposed overload detection approach achieved considerable gains in energy savings, for the IQRF policy with respect to IQR it reached 15.43% gain, in the case of the LRF in relation to LR it achieved 9.39%, in MADF compared to MAD 15.64% and finally in THRF for THR it was 0.7%.

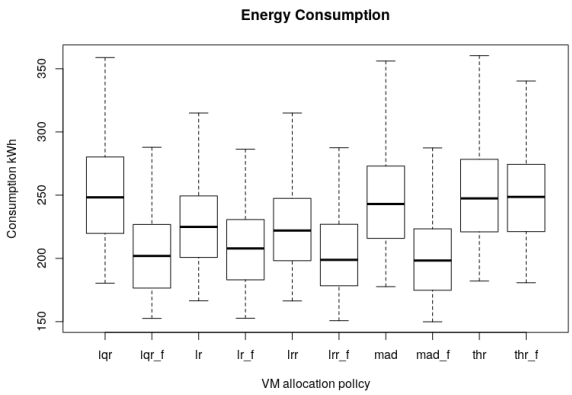



Figure 4: Energy consumption

In the boxplot of Figure 5, the result is shown about the average SLA violation, and the proposed approach achieved better results reaching 9.7% by comparing THRF to THR, considering the SLA violation minimums, and in the case of maximums reached gains of up to 4.5%.

## 6 Conclusion

In this work, we present a new approach to host overload detection in energy-efficient cloud computing using Type-2 Fuzzy Logic as part of the consolidation of VMs. We simulated it and compared the results with five well-known VM allocation algorithms, IQR, LR, LRR, MAD and THR applying the VM selection policies MC, MMT, MU, RS. Most proposals deal with uncertainties through mechanisms that react to lessen the negative impact of uncertainties. The extra load and the intrusion generated in the cloud environment by reactive approaches justify the implementation of mechanisms that deal with the uncertainties directly

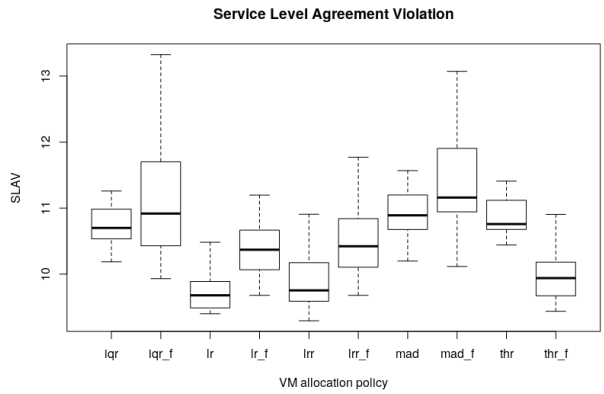



Figure 5: Average SLA violation

in the load balancers; this is the main advantage of Int-FLBCC. In most cases, the proposed approach using Type-2 Fuzzy Logic obtained good results about energy efficiency, reaching gains of 15.64, 15.43 and 9.19% to MAD, IRQ, and LR respectively.

In future work, we intend to approach effective techniques to calibrate the rules base of the fuzzy inference system to improve SLA levels and energy efficiency, as well as apply trade-off between performance and energy efficiency. Compare to other methods found in the literature and to implement in the real cloud computing environment.

**Acknowledgement**

This work was supported by CAPES/Brasil, Brazilian Funding Agency CAPES, MCTI/CNPQ Universal (448766/ 2014-0) and PQ(310106/ 2016-8) and PqG/FAPERGS 02/2017(17/2551-0001207-0).

## References

[1] E. Arianyan, H. Taheri, V. Khoshdel, Novel fuzzy multi objective dvfs-aware consolidation heuristics for energy and sla efficient resource management in cloud data centers, Journal of Network and Computer Applications 78 (2017) 43 – 61.

[2] A. Beloglazov, R. Buyya, Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers, Concurr. Comput. : Pract. Exper. 24 (2012) 1397–1420.

[3] A. Beloglazov, R. Buyya, Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints, IEEE Transactions on Parallel and Distributed Systems 24 (7) (2013) 1366–1379.

[4] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, R. Buyya, Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, Software: Practice and experience 41 (1) (2011) 23–50.

[5] J. R. Castro, O. Castillo, L. G. Martínez, Interval type-2 fuzzy logic toolbox, Engineering Letters 15 (1) (2007) 89–98.

[6] J. R. Castro, O. Castillo, P. Melin, A. Rodríguez-Díaz, Building Fuzzy Inference Systems with a New Interval Type-2 Fuzzy Logic Toolbox, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 104–114.

[7] M. Gehrke, C. Walker, E. Walker, Some comments on interval valued fuzzy sets, International Journal of Intelligent Systems 11 (10) (1996) 751–759.

[8] P. Haratian, F. Safi-Esfahani, L. Salimian, A. Nabiollahi, An adaptive and fuzzy resource management approach in cloud computing, IEEE Transactions on Cloud Computing (2018) 1–1.

[9] M. A. Hiltunen, R. D. Schlichting, G. Jung, C. Pu, K. R. Joshi, Mistral: Dynamically managing power, performance, and adaptation cost in cloud infrastructures, in: 2010 IEEE 30th International Conference on Distributed Computing Systems(ICDCS), Vol. 00, 2010, pp. 62–73.

[10] N. N. Karnik, J. M. Mendel, Introduction to type-2 fuzzy logic systems, in: 1998 IEEE International Conference on Fuzzy Systems Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98CH36228), Vol. 2, 1998, pp. 915–920 vol.2.

[11] N. N. Karnik, J. M. Mendel, Q. Liang, Type-2 fuzzy logic systems, IEEE Transactions on Fuzzy Systems 7 (6) (1999) 643–658.

[12] E. Klement, R. Mesiar, E. Pap, Triangular norms. position paper I: basic analytical and algebraic properties, Fuzzy Sets and Systems 143 (1) (2004) 5–26.

[13] G. J. Klir, Uncertainty and Information: Foundations of Generalized Information Theory, Wiley-Interscience, 2005.

[14] J. M. Mendel, Fuzzy sets for words: a new beginning, in: Fuzzy Systems, 2003. FUZZ '03. The 12th IEEE International Conference on, Vol. 1, 2003, pp. 37–42.

[15] B. M. P. Moura, G. B. Schneider, A. C. Yamin, M. L. Pilla, R. H. S. Reiser, Int-fgrid: Bot tasks scheduling exploring fuzzy type-2 in computational grids, in: 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2018, pp. 1–8.

[16] B. M. P. Moura, G. B. Schneider, A. C. Yamin, R. H. S. Reiser, M. L. Pilla, Int-fGrid: A type-2 fuzzy approach for scheduling tasks of computational grids, in: 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2017, pp. 1416–1423.

[17] G. Portaluri, D. Adami, A. Gabbrielli, S. Giordano, M. Pagano, Power consumption-aware virtual machine allocation in cloud data center, in: 2016 IEEE Globecom Workshops (GC Wkshps), 2016, pp. 1–6.

[18] L. Salimian, F. Safi Esfahani, M.-H. Nadimi-Shahraki, An adaptive fuzzy threshold-based approach for energy and performance efficient consolidation of virtual machines, Computing 98 (2016) 641–660.

[19] A. Shehabi, S. Smith, D. Sartor, R. Brown, M. Herrlin, J. Koomey, E. Masanet, N. Horner, I. Azevedo, W. Lintner, United states data center energy usage report.

[20] S. Singh, I. Chana, Earth: Energy-aware autonomic resource scheduling in cloud computing, Journal of Intelligent & Fuzzy Systems 30 (3) (2016) 1581–1600.

[21] A. N. Toosi, R. Buyya, A fuzzy logic-based controller for cost and energy efficient load balancing in geo-distributed data centers, in: 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC), 2015, pp. 186–194.

[22] C. Wagner, Juzzy - a java based toolkit for type-2 fuzzy logic, in: 2013 IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems (T2FUZZ), 2013, pp. 45–52.

[23] D. Wu, M. Nie, Comparison and practical implementation of type-reduction algorithms for type-2 fuzzy sets and systems, in: FUZZ-IEEE, IEEE, 2011, pp. 2131–2138.

[24] L. Zadeh", The concept of a linguistic variable and its application to approximate reasoning—i, Information Sciences 8 (3) (1975) 199 – 249.