

Int-FLBCC: Model for Load Balancing in Cloud Computing using Fuzzy Logic Type-2 and Admissible Orders

Int-FLBCC: Modelo para Balanceamento de Carga em Nuvens Computacionais empregando Lógica Fuzzy Tipo-2 e Ordens Admissíveis

Guilherme B. Schneider^{1*}, Bruno M. P. Moura¹, Adenauer C. Yamin¹, Renata H. S. Reiser¹

Abstract: Dynamic consolidation of virtual machines (VMs) is an effective way to improve resource utilization and power efficiency in cloud computing, directly affecting Quality of Service aspects. This paper presents Int-FLBCC, a new proposal with exploring a Type-2 Fuzzy Logic approach to address the uncertainties and inaccuracies in determining resource usage, aiming at energy savings with minimal performance degradation. Validation results in a simulated cloud computing environment showed improvements in energy efficiency of 8.83% with IQR_XY and 22.43% with MAD_XY. For fulfillment of Service Level Agreements (SLA), the best values achieved were 9.06% with MAD_XY and 25% of THR_Lex1.

Keywords: Fuzzy logic — Uncertainty — Cloud computing — Resource management — Data centers — Random access memory — Quality of service

Resumo: A consolidação dinâmica de máquinas virtuais (*Virtual Machine - VM*) é uma maneira eficaz de melhorar a utilização de recursos e a eficiência energética em data centers da nuvem, afetando diretamente os aspectos de Qualidade de Serviço (*Quality of Service - QoS*). Este artigo apresenta uma nova proposta explorando a Lógica Fuzzy Tipo-2 para tratar as incertezas e imprecisões na determinação da utilização dos recursos na computação em nuvem, visando economia de energia sem degradação de desempenho. Para validação da proposta executamos testes em um ambiente simulado de computação em nuvem com a ferramenta CloudSim. A validação dos resultados através de simulação mostraram melhorias na eficiência energética de 8,83% com IQR_XY e 22,43% com MAD_XY. Para o cumprimento dos Acordos de Nível de Serviço (*Service Level Agreement - SLA*), os melhores valores alcançados foram 9,06% com MAD_XY e 25% de THR_Lex1.

Palavras-Chave: Lógica Fuzzy — Incerteza — Computação em Nuvem — Gerenciamento de recursos — Data center — Memória de acesso aleatório — Qualidade de serviço

¹Centro de Desenvolvimento Tecnológico (CDTEC), Universidade Federal de Pelotas (UFPel), Pelotas, Rio Grande do Sul, Brasil

*Corresponding author: gbschneider@inf.ufpel.edu.br

DOI: <http://dx.doi.org/10.22456/2175-2745.98362> • Received: 20/11/2019 • Accepted: 17/05/2020

CC BY-NC-ND 4.0 - This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

1. Introdução

A Computação em Nuvem (CN) é um paradigma computacional constituído das categorias de Infraestrutura (IaaS), Plataforma (PaaS) e Software (SaaS) como Serviço usando um modelo de pagamento por uso, cujo valor pago é proporcional ao uso. Na CN *data centers - (DC)* são usados para hospedagem das aplicações e conhecidos por demandar grande quantidade de eletricidade, elevando o custo operacional para os provedores e contribuindo com a emissão de CO₂ no meio ambiente [1]. A principal vantagem deste paradigma é que ambos clientes e provedores podem ajustar rápida e continuamente a alocação de recursos com base nas necessidades atuais, característica conhecida como elasticidade [2].

De acordo com o relatório do Conselho de Defesa dos Recursos Naturais (NRDC)¹ dos Estados Unidos (EUA), em 2014, somente os seus DC consumiram uma estimativa de 70 bilhões de quilowatts/hora (kWh), representando cerca de 1.8% do seu consumo total de eletricidade [3].

O consumo energético deverá continuar aumentando no futuro próximo, em torno de 4% de 2014 a 2020, a mesma taxa dos últimos cinco anos. Com base nas estimativas de tendências atuais, espera-se que somente os data centers dos EUA consumam aproximadamente 73 bilhões de kWh em 2020. Empresas como Google², Microsoft³ e Amazon⁴ estão

¹<https://www.nrdc.org/>

²<https://environment.google/>

³<https://www.microsoft.com/en-us/environment/energy/>

⁴<https://aws.amazon.com/about-aws/sustainable-energy/>

trabalhando para atingir esse objetivo usando fontes de energia renováveis para alimentar suas infraestruturas computacionais e fazer investimentos direto na geração de energia verde no local.

A busca por eficiência energética sem perda de desempenho fez com que diversos conceitos fossem propostos, com destaque na CN, consistindo em uma distribuição dos serviços de computação (alocação de servidores, armazenamento, banco de dados) em que não há a necessidade do usuário fazer grande investimento em equipamentos, e ainda, com pagamento associado ao tempo de uso [4].

Aliado a isso, surge o desafio em prover técnicas de Balanceamento de Carga (*Load Balancing - LB*) em ambientes de nuvens computacionais buscando a eficiência energética ao mesmo tempo mantendo bons Níveis de Acordo de Serviço (*Service Level Agreement - SLA*) e Qualidade de Serviço (*Quality of Service - QoS*). O problema de minimização do consumo energético sobre restrições de QoS é analiticamente complexo como um todo, e faz parte da consolidação dinâmica de VM, que é um problema NP-Difícil [5].

Os principais desafios da CN são: (i) provisionamento automático de serviço; (ii) migração de VM; (iii) consolidação de servidores, (iv) gerenciamento de energia, e ainda, (v) segurança de dados [6].

Este trabalho contempla a migração e consolidação de VM com uma abordagem para utilização eficiente dos servidores físicos do ambiente da CN e buscando, simultaneamente manter um padrão satisfatório de QoS e SLA.

Para o LB entre *data centers* de nuvens computacionais, frequentemente, dentre os fatores que implicam em incerteza, tem-se o Poder Computacional (PC), o Custo de Comunicação (CC), e o uso de Memória de Acesso Aleatório (*Random Access Memory - RAM*). Por exemplo, o CC entre equipamentos que compartilham a Internet como meio de interoperabilidade é dependente do uso médio da infraestrutura de comunicação no momento que ocorre o LB entre as máquinas dos *data centers*. A intratabilidade do uso de RAM, conseqüentemente, leva a violação do SLA caso o LB não vier a considerá-la e, em consequência acarreta a obtenção de níveis insatisfatórios de QoS.

A utilização de ambientes distribuídos de CN para execução de aplicações que necessitem de grandes computações vem crescendo a cada dia, estimulando para que os desafios da área busquem o uso eficiente dos recursos e, dentre os quais destacam-se as seguintes questões de pesquisa:

- (i) os LB consideram as incertezas associadas a fatores como Poder Computacional (PC), Custo de Comunicação (CC), e uso de RAM?
- (ii) os balanceadores estão munidos de técnicas consistentes para o tratamento das incertezas associadas aos ambientes de CN?
- (iii) os usuários possuem o preciso conhecimento sobre a demanda computacional, e de comunicação para suas aplicações?

Considerando este contexto e levando em conta alguns trabalhos como os projetos [1] e [7], que utilizam Lógica Fuzzy (*Fuzzy Logic - FL*) para modelagem de incerteza no LB na CN, este trabalho discute *Int-FLBCC (Interval Fuzzy Load Balancing for Cloud Computing)*. O desenvolvimento desta proposta estende a abordagem proposta em [8], contribuindo para a consolidação dinâmica de VM empregando a Lógica Fuzzy Tipo-2 (*Type-2 Fuzzy Logic - T2FL*). Particularmente, interpreta-se nesta modelagem a incerteza e a imprecisão inerentes às variáveis da CN, neste caso PC, CC e RAM, buscando ainda o equilíbrio entre a eficiência energética e SLA.

O artigo está estruturado em nove seções. A seção 1 trata dos fundamentos contextuais do trabalho. A seção 2, tem-se a fundamentação da CN. Na seção 3 apresenta-se os conceitos básicos da T2FL relevantes para modelagem e análise do LB na CN. A seção 4 destaca os trabalhos relacionados. Na seção 5 é exposta uma visão geral dos processos operacionais envolvidos. A seção 6, apresenta a modelagem do componente *Int-FLBCC*, incluindo detalhamento da base de dados, fuzzificação, base de regras, inferência e defuzzificação. Na seção 7 são definidas as métricas avaliadas. A seção 8, trata assuntos referente ao estudo de caso aplicado a *Int-FLBCC*, com uma descrição dos testes realizados. E, por fim, na seção 9 constam as considerações finais.

2. CN: Principais Conceitos

O advento das tecnologias de processamento, armazenamento e comunicações, aliadas na busca da eficiência energética, motivou a intensificação das pesquisas em torno de alternativas para o gerenciamento de recursos no paradigma da CN. Neste contexto, os recursos são disponibilizados sobre demanda de acordo com o tempo de usabilidade.

Apesar do atual destaque dado a pesquisa em CN, seu conceito não é novo e tão pouco suas tecnologias [9]. Em 1960, John McCarthy acreditava que um dia corporações conseguiriam vender recursos computacionais como “*commodity*” [10].

O surgimento da CN causou um grande impacto nas indústrias de Tecnologia da Informação (*Information Technology - IT*), dentre elas Google, Amazon e Microsoft que constantemente batalham para terem um ambiente de CN mais potente, confiável e eficiente energeticamente.

A CN é um modelo que permite o acesso conveniente, ubíquo e sob-demanda de um agrupamento de recursos computacionais, como por exemplo, rede, servidores, armazenamento, aplicações e serviços, os quais podem ser rapidamente alocados e liberados com esforço mínimo de gerenciamento ou interação com o provedor de serviços [2].

Este modelo CN é composto de cinco características essenciais, detalhadas a seguir:

- **Autoatendimento Sob Demanda:** um consumidor pode provisionar unilateralmente recursos de computação,

como tempo do servidor e armazenamento de rede, conforme necessário, sem exigir interação com cada provedor de serviços;

- **Ampla acesso à rede:** os recursos estão disponíveis na Internet e são acessados através de mecanismos padrões que possibilitam o uso em plataforma heterogêneas integrando telefones celulares, tablets, notebooks e estações de trabalho, dentre outros;
- **Agrupamento de recursos:** os recursos computacionais do provedor são agrupados para atender a vários consumidores usando um modelo de multilocação, com diferentes recursos físicos e virtuais atribuídos dinamicamente de acordo com a demanda do consumidor;
- **Elasticidade Rápida:** os recursos podem ser provisionados e liberados elasticamente, em alguns casos automaticamente, de acordo com a demanda. Para o consumidor, os recursos disponíveis para provisionamento geralmente parecem ilimitados e podem ser apropriados em qualquer quantidade, e a qualquer momento;
- **Serviços mensurados:** permitindo que recursos possam ser monitorados, controlados, e relatados. Proporcionando transparência tanto para o fornecedor quanto para consumidor do serviço utilizado.

2.1 CN: Modelos de Serviços

A CN provê serviços sob demanda por meio da Internet [11]. Para prover esses serviços a arquitetura básica de CN é dividida em três categorias, as quais são apresentadas na Figura 1 e detalhadas logo a seguir.



Figure 1. Serviços oferecidos em CN [12]

- **Software as a Service (SaaS)**
Modelo em que o *software* é oferecido por um provedor de serviços, disponível sob demanda, geralmente por meio de um navegador *Web*;
- **Platform as a Service (PaaS)**
Modelo intermediário em que o *software* é normalmente desenvolvido através de *Application Programming Interface (API)* fazendo com que o foco se concentre so-

mente no desenvolvimento do *software*, salvo algumas configurações necessárias no ambiente;

- **Infrastructure as a Service (IaaS)**

Modelo em que o *hardware* é abstraído do usuário através de VM, fazendo com que o usuário não precise se preocupar com aspectos relacionados a parte física, como servidores e redes;

2.2 CN: Desafios Propostos

Apesar da CN estar amplamente presente na indústria e no desenvolvimento tecnológico, diversas propostas de pesquisa vem sendo consideradas. Nesta seção são apresentados principais desafios da área abordados na literatura [13].

- **Provisionamento automatizado de serviços:** consiste em alocar e realocar VM de acordo com a necessidade do cliente. As realocações devem ser feitas respeitando os Objetivo de Nível de Serviço (*Service Level Objective - SLO*) do ambiente e buscando minimizar o custo operacional. Contudo, não é fácil determinar como mapear SLOs para cumprir requisitos de QoS, como por exemplo, definir a utilização da CPU e memória, pois essas oscilam a cada segundo. Justifica-se a necessidade de constantes atualizações e otimizações no sistema *online* para lidar com esses provisionamentos;
- **Migração de Máquina Virtual:** a virtualização trouxe diversas vantagens para a CN, como a possibilidade da criação de VM com o intuito de balancear a carga por todo o *data center*, e ainda, possibilitando robustez e rápida resposta de provisionamento. Os principais benefícios da migração de VM são otimizar uso dos recursos, no entanto, isso não é uma tarefa simples. Atualmente, a detecção de picos de carga de trabalho e o início de uma migração não têm a agilidade necessária para responder às mudanças súbitas e dinâmicas de carga de trabalho;
- **Consolidação de servidores:** a consolidação de servidores é uma maneira eficaz de maximizar a utilização de recursos enquanto minimizado o consumo energético do ambiente de CN. A técnica de migrar VM em tempo real é constantemente utilizada para consolidar VM que estão alocadas em múltiplos servidores, frequentemente sub-utilizados. A migração para um único servidor, viabiliza que os outros servidores possam ser desligados. Contudo, o problema de consolidação de servidores em um *data center* é constantemente avaliado como uma variante do Problema do Empacotamento (*Bin-Packing*), apresentado com um problema computacional NP-Difícil [14];
- **Gerenciamento de Energia:** melhorar a eficiência energética é outra questão importante na CN, empresas de IT estão sobre constante evolução, visando novas tecnologias que permitem diminuir o consumo energético,

não só pela questão financeira, mas sobretudo para cumprir as regulamentações governamentais e os padrões ambientais.

2.3 CN: Caracterização dos Simuladores

Uma desvantagem da CN está na modelagem de ambientes de testes. Os usuários finais, esperam uma experiência satisfatória com os serviços da CN, por outro lado, as empresas provedoras não desejam ter gastos desnecessários em energia ou compra demasiada de equipamentos.

Com o intuito de solucionar esse problema foram desenvolvidos simuladores para auxiliar na implementação e testes de ambientes de CN controlados. Há muitos benefícios na utilização de simuladores, como exposto em [15], e dentre eles destacam-se três parâmetros relevantes:

- (i) **Custo Mínimo:** considera-se o desenvolvimento via *software*, gerando menor custo, quando comparado com *hardware*;
- (ii) **Repetível e Controlável:** pode-se testar diversificando os testes por várias vezes até obter-se a saída desejável;
- (iii) **Ambientes Heterogêneos:** pode-se planejar avaliações para cenários diversificados sob diferentes cargas de trabalho e medição de custos.

Dentre os simuladores citados em [15], os critérios de seleção para esta pesquisa foram motivadas pelo seguintes aspectos: (i) ter amplo emprego científico, (ii) disponibilizar referencial teórico, e principalmente abranger o ponto principal, (iii) apresentar um modelo energético consolidado. Três simuladores e suas principais características são ressaltados a seguir:

- **GreenCloud** [16] um simulador projetado para cálculo do consumo energético de qualquer componente específico do *data center* como *link*, *switch*, *gateway*, dentre outros. Porém, requerendo muita memória e tempo de simulação, sendo então recomendado somente para pequenos *data centers*;
- **CloudAnalyst** [17] simulador projetado para avaliar o desempenho e o custo de grandes ambientes de CN distribuídos em diferentes locais, e manipulando grandes cargas de trabalho dos usuários, possui uma *interface* gráfica que facilita a visualização dos dados através de gráficos;
- **CloudSim** [18] simulador de grande popularidade e que possui uma comunidade ativa, possibilitando a modelagem e criação *data centers*, com número ilimitado de VM, e suportando o importante recurso do modelo referido como *pay-as-you-go* da CN.

Com o estudo dos simuladores atuais, em especial os três citados anteriormente, o escolhido para implementação foi

o CloudSim [18] devido ao provimento de um modelo consolidado para simulação do consumo energético de *data centers*, baixa utilização de memória, possibilitando a reprodução simulada de grandes infraestruturas de CN, e por conter nativamente diversas políticas de alocação e escalonamento de VM.

3. T2FL: Fundamentos Relevantes

Loff Zadeh introduziu a T2FL em 1975 como extensão da FL [19]. Seu surgimento está relacionado com a insuficiência da teoria dos Conjuntos Fuzzy (*Fuzzy Sets - FS*) tradicionais na modelagem das incertezas inerentes à definição das funções de pertinência dos antecedentes e consequentes em um Sistema de Inferência Fuzzy (*Fuzzy Inference System - FIS*) [20]. Esses conjuntos podem ser usados em situações onde existe incerteza sobre o grau, formas, e/ou parâmetros das funções de pertinência [21], fornecendo uma estratégia potencial no tratamento das incertezas em modelos baseados em múltiplos critérios obtidos através de distintos especialistas, ou ainda, extraído de simuladores.

A T2FL fundamenta-se na teoria dos Conjuntos Fuzzy do Tipo-2 (*Type-2 Fuzzy Sets - T2FS*), modelando o tratamento de incertezas e permitindo especificar um intervalo $\mu_{\tilde{A}} = [\mu_{\tilde{A}}, \bar{\mu}_{\tilde{A}}]$ como grau de pertinência de tal elemento x em um FS \tilde{A} [22]. Desta forma, a T2FS estende os conceitos da LF, modelando a imprecisão como habilidade adicional considerando a não-especificidade como outro aspecto importante da incerteza. E neste contexto, refletindo essa falta de especificação por toda a extensão do grau de pertinência, que é determinado por um FS – no caso, um subintervalo do intervalo unitário $[0, 1]$.

Este trabalho considera a Lógica Fuzzy Tipo-2 Intervalar (*Interval Type-2 Fuzzy Set Logic - IT2FL*). Particularmente, a semântica considerada provê a interpretação do grau de pertinência do elemento x do universo \mathcal{X} em um conjunto fuzzy A , como um valor numérico no intervalo de pertinência $\mu_A(x)$. Nesta interpretação, não é possível precisar qual é exatamente esse valor e portanto, apenas fornecer limites (superior e inferior) correspondendo aos extremos do seu intervalo de pertinência.

O grau intervalar de pertinência de um elemento em um T2FS possibilita a modelagem das opiniões de múltiplos especialistas quando da atribuição dos graus de pertinências, auxiliando na tomada de decisão baseada em múltiplos critérios. Neste contexto, uma das métricas que avalia a imprecisão nas diferentes atribuições de especialistas está identificada como o diâmetro do correspondente grau intervalar de pertinência.

Definição 1 [21] *Seja $\mathcal{X} \neq \emptyset$ um universo. Um T2FS \tilde{A} é caracterizado por uma Função de Pertinência do Tipo-2 (Membership Function Type-2 - T2MF) onde $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$, $x \in \mathcal{X}$ e $u \in J_x \subseteq [0, 1]$. Para cada T2FS \tilde{A} tem-se:*

$$\tilde{A} = \{((x, u), \mu_{\tilde{A}}(x, u)) \mid \forall x \in \mathcal{X}, \forall u \in J_x \subseteq [0, 1]\}. \quad (1)$$

Um T2FS atribui a um elemento no universo \mathcal{X} um mapea-

mento $A(x) : [0, 1] \rightarrow [0, 1]$, e pode ser definido como

$$\{(x, A(x, t)) : x \in \mathcal{X}, t \in [0, 1]\}$$

quando $A(x, \cdot) : [0, 1] \rightarrow [0, 1]$ é dado como $A(x, t) = A(x)(t)$, para cada $x \in \mathcal{X}$, $t \in [0, 1]$. Em particular nos FS $A(x)$ é um número real em $[0, 1]$, para cada $x \in \mathcal{X}$.

Definição 2 [22] Se $\mu_A(x) = 1$ então A é um conjunto fuzzy intervalar

$$A(x) = \{(u, 1) : u \in J_x \subseteq [0, 1]\}, \forall x \in \mathcal{X}.$$

Observa-se que os conjuntos fuzzy valorados por intervalos [23] é um caso particular de T2FS. Seja A um T2FS $A(x) = [\underline{A}(x), \overline{A}(x)]$, $\forall x \in \mathcal{X}$. Além disso, sejam $A, B \in T2FS$. Para todo $x \in \mathcal{X}$, tem-se os operadores:

- (i) Complemento: $A_C(x) = [1 - \overline{A}(x), 1 - \underline{A}(x)]$;
- (ii) União: $A(x) \cup B(x) = [\max(\underline{A}(x), \underline{B}(x)), \max(\overline{A}(x), \overline{B}(x))]$;
- (iii) Intersecção: $\mu_{A \cap B}(x) = [\min(\underline{A}(x), \underline{B}(x)), \min(\overline{A}(x), \overline{B}(x))]$.

Denotando, $A(x) = X$, $B(x) = Y$, $\forall x \in \mathcal{X}$, U como o conjunto de todos os sub-intervalos reais no intervalo unitário $[0, 1]$ e \cup como o conjunto dos valores fuzzy intervalares, a ordem " \leq " parcial em \cup é a *Ordem Produto* [24] dada como:

$$X \leq Y \Leftrightarrow \underline{X} \leq \underline{Y} \wedge \overline{X} \leq \overline{Y}.$$

As funções que qualificam as intersecções e uniões fuzzy são modeladas neste trabalho por normas e conormas triangulares, respectivamente. Segundo [22] e considerando os intervalos em \cup , tem-se que:

- Uma norma triangular (t-norma) intervalar é uma função $\mathbb{T} : \cup^2 \rightarrow \cup$ que satisfaz as propriedades de comutatividade, associatividade, monotonicidade e tem o $\mathbb{1} \in U$ como elemento neutro. Alguns exemplos de t-normas mais utilizadas são: Intersecção Padrão, Produto Algébrico, Intersecção Drástica, Lukasiewicz e Nilpotente Mínimo.
- Uma conorma triangular (t-conorma) intervalar é uma função binária $\mathbb{S} : \cup^2 \rightarrow \cup$ que satisfaz as propriedades de comutatividade, associatividade, monotonicidade e tem o $\mathbb{0} \in U$ como elemento neutro. Alguns exemplos de t-conormas são: União Padrão, Soma Probabilística, União Drástica, Lukasiewicz e Nilpotente Máximo.

Um sistema baseado em T2FL pode estimar funções de entrada e saída, por meio do uso de heurísticas e técnicas intervalares. Na Figura 2 é apresentada a arquitetura do sistema de inferência baseado em T2FL. A seguir os principais blocos que constituem um Sistema de Inferência Fuzzy Tipo-2 (*Type-2 Fuzzy Inference System - T2FIS*) são brevemente descritos:

- (1) **Interface de Fuzzificação:** O processo de fuzzificação baseado em T2FS é realizado de acordo com a natureza e definição de um FST2, associando um valor de entrada

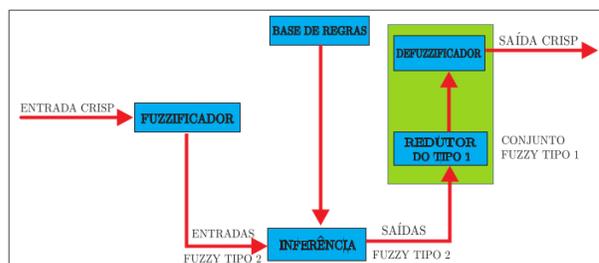


Figure 2. Arquitetura do Sistema de Inferência Fuzzy Tipo-2

com uma função intervalar e não simplesmente com um único valor em U . Em outras palavras, é inserido no mecanismo de inferência a incerteza relacionada as funções de pertinência de entrada.

Assim, para cada entrada $A(x)$ um vetor de entrada $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathcal{X}^n$ quando $n \in \mathbb{N}^*$ está relacionado a um par de vetores em \cup^n obtidos da seguinte forma:

$$(\overline{A}(x_1), \overline{A}(x_2), \dots, \overline{A}(x_n)), (\underline{A}(x_1), \underline{A}(x_2), \dots, \underline{A}(x_n))).$$

- (2) **Base de Regras (Rule Base - RB):** constituída por regras que classificam as Variáveis Linguísticas (VL) de acordo com os T2FS valorados por intervalos;
- (3) **Unidade de Decisão Lógica:** realiza as operações de inferência entre os dados de entrada e as condições impostas na RB, gerando a ação a ser realizada no T2FIS;
- (4) **Defuzzificação:** nesta fase, são consideradas duas principais etapas, que são elas:
 - (i) **Redutor de Tipo:** responsável por reduzir T2FS em FS, ao buscar o melhor FS que representa o T2FS deve satisfazer a seguinte premissa: quando toda a incerteza desaparecer, o resultado do T2FIS reduz-se a um FIS [25].
 - (ii) **Defuzzificação:** O T2FIS usa a média dos pontos limites \underline{y} e \overline{y} da saída $B(x)$:

$$y = \frac{\underline{y} + \overline{y}}{2} = \frac{\underline{B}(x) + \overline{B}(x)}{2}, \forall x \in \mathcal{X}, \quad (2)$$

onde os valores \underline{y} e \overline{y} são calculados via método iterativo de Karnik e Mendel (algoritmo KM), ou obtido através do uso de um método convencional, como o centroide, no valor final da inferência.

Nem sempre uma lista de intervalos gerada com a saída do T2FIS é comparável pelos métodos convencionais [26]. Neste estudo aplicam-se as abordagens das ordens admissíveis exemplificadas em [27, 26], contornando as situações onde dois intervalos possam ser incomparáveis, como por exemplo, o produto " \leq ", uma relação de ordem parcial, e permite que dois intervalos possam ser incomparáveis ($X \not\leq Y$ and $Y \not\leq X$). As duas classes admissíveis consideradas neste trabalhos são relacionadas brevemente a seguir:

1. A Ordem de Xu e Yager [28]: $\forall X, Y \in \mathcal{U}$,

$$[\underline{X}, \bar{X}] \leq_{XY} [\underline{Y}, \bar{Y}] \Leftrightarrow \begin{cases} \underline{X} + \bar{X} < \underline{Y} + \bar{Y}; \text{ or} \\ \underline{X} + \bar{X} = \underline{Y} + \bar{Y} \text{ and } \bar{X} - \underline{X} \leq \bar{Y} - \underline{Y}. \end{cases}$$

2. As ordens Lexicográficas \leq_{Lex1} relacionada a primeira variável, e \leq_{Lex2} a segunda variável, que são respectivamente definidas da seguinte forma:

$$[\underline{X}, \bar{X}] \leq_{Lex1} [\underline{Y}, \bar{Y}] \Leftrightarrow \begin{cases} \underline{X} < \underline{Y}; \text{ or} \\ \underline{X} = \underline{Y} \text{ and } \bar{X} \leq \bar{Y}; \end{cases}$$

$$[\underline{X}, \bar{X}] \leq_{Lex2} [\underline{Y}, \bar{Y}] \Leftrightarrow \begin{cases} \bar{X} < \bar{Y}; \text{ or} \\ \bar{X} = \bar{Y} \text{ and } \underline{X} \leq \underline{Y}. \end{cases}$$

4. Trabalhos Relacionados

Oito trabalhos relacionados que utilizam FL são brevemente relatados nesta seção, discutindo métodos distintos para lidar com as incertezas no LB, e ilustrando a necessidade do tratamento preventivo das incertezas nas etapas do gerenciamento de recursos nos ambientes de CN.

Em Masoumzadeh et al. [29] é constituída uma abordagem baseada na consolidação dinâmica de VM distribuídas, uma estratégia eficaz para melhorar a eficiência energética em ambientes de nuvem. Neste trabalho a estratégia é decomposta em quatro tarefas de tomada de decisão: (i) detecção de sobrecarga do host, (ii) seleção de VM, (iii) detecção de subcarga do *host* e (iv) alocação de VM. O objetivo é consolidar as VM dinamicamente de forma a otimizar o *trade-off* entre desempenho e eficiência energética. Esse trabalho concentra-se na tarefa (iv) seleção de VMs, onde é proposta uma técnica *Fuzzy Q-Learning* (FQL) [35] para tomada de decisões ao selecionar VM para migração.

A abordagem considera as seguintes variáveis em sua modelagem: (i) Consumo Energético (CE); (ii) Desempenho da Nuvem (DN), o DN da nuvem é definido como uma função linear que avalia o SLA entregue em qualquer VM implantada em uma IaaS, as métricas para medir as violações de SLA em um *data center* foram: violação de SLA devido a superutilização (SLAVO) e Violação do SLA devido a Migrações (SLAVM); (iii) Utilização de CPU do *host* (CU); e o (iv) Número de VM (NumVM) que reside no *host*. Para avaliação da abordagem o kit de ferramentas CloudSim é usado considerando carga de trabalho do mundo real do PlanetLab. Os resultados experimentais mostraram que, ao utilizar o FQL, foi produzido aceitável nível de *trade-off* entre desempenho energético e violação do SLA nos *data centers* da nuvem computacional, em comparação com outros algoritmos.

Em Toosi and Buyya [1] a proposta é a utilização eficiente de fontes de energia renováveis empregando a abordagem da FL para o tratamento das incertezas com base na intermitência e imprevisibilidade dos fatores como: (i) variações climáticas, e (ii) preço pago pela energia no local de hospedagem do

DC. Com isso o objetivo central do trabalho é fornecer aos provedores de nuvem com vários DC distribuídos geograficamente em uma região o tratamento das variações temporais e no preço da energia na rede no local, roteando a carga para um DC, a fim de reduzir custos e aumentar a utilização de energia renovável. Para atingir esse objetivo foi proposto um algoritmo de LB baseado em lógica fuzzy que atua sem conhecimento futuro. Para atingir o objetivo o mecanismo de inferência fuzzy projetado considerou: (i) a utilização de fontes de energia renovável U_i , (ii) quantidade do consumo de energia da rede convencional B_i e (iii) preço médio da eletricidade no local do DC, calculado dentro de uma janela de tempo F_i , obtendo como saída a adequação do DC para recebimento da carga de trabalho. Os testes realizados foram com base em traços de carga de trabalho do mundo real obtidos no *National Renewable Energy Laboratory* (NREL), *Energy Information Administration* (EIA) nos EUA e através de informações de uso do *cluster* do Google. Comparado a outros algoritmos de *benchmark*, o método foi capaz de reduzir custo sem conhecimento prévio do preço futuro da eletricidade, da disponibilidade de energia renovável e das cargas de trabalho.

Em Portaluri et al. [30] os autores realizam a implementação de diferentes algoritmos de alocação baseados em otimização com abordagem da FL para único ou multi-objetivo, duas variantes de uma heurística multi-objetiva e uma abordagem analítica. O objetivo central do trabalho é o foco no encaminhamento de pacotes visando a eficiência energética, além de agregar ao trabalho o objetivo da pesquisa anterior em Adami et al. [36]. As solicitações de VM são definidas com base nas variáveis de recursos computacionais do sistema que são elas: (i) CPU, (ii) RAM, (iii) Disco rígido e (iv) largura de banda da rede. Testes para avaliação de desempenho dos escalonadores foram realizadas em termos de números de VM alocadas para cada política. Os resultados mostraram que a abordagem multi-objetiva é capaz de alocar o maior número de VM, em média, tendo o distribuição mais estreita quando a Pior Ajuste (*Worst Fit* - WF) é adotada.

Na proposta de Singh and Chana [31], é apresentado um *framework* denominado *Energy-aware Autonomic Resource management Technique* (EARTH) para escalonamento de recursos de nuvens computacionais baseado em FL com o objetivo da obtenção da eficiência energética. O sistema de inferência fuzzy considerou as seguintes variáveis: (i) *Workload waiting Time* (WWT), (ii) *Workload Execution Time* (WET) e (iii) *Resource Energy Consumption* (REC), gerando como saída *Workload Processing Priority* (WPP). A avaliação do *framework* EARTH proposto foi baseada em simulação empregando o CloudSim e um ambiente real de nuvem computacional da Universidade de Thapar, na Índia (Thapar Cloud). Os resultados experimentais mostram que o *framework* proposto tem melhor desempenho em termos de utilização de recursos e consumo de energia, juntamente com outros parâmetros de QoS.

Em Salimian et al. [32], foi concebido um algoritmo

Table 1. Comparação dos trabalhos relacionados considerando as ferramentas utilizadas, o número de variáveis de entrada (Ent) e Saída (Sai), os métodos de fuzzificação (Fuz), inferência (Inf) e defuzzificação (Def), e ainda, os conectivos (Con) utilizados em cada uma das aplicações.

Referência	Ferramentas	Ent/Sai	Fuz	Inf	Def	Con
[29]	♠	4/1	MF Gau	TS	FQL	AND
[1]	♠	3/1	MF Tri	Ma	Centroid	AND
[30]	◇ ♣	3/1	MF Tri/Tra	TS	CoG	AND
[31]	♠	3/1	MF Gau	Ma	Maximum	AND
[32]	♠ △	2/1	MF Tri	TS	CoG	AND
[33]	♠	8/1	MF Tri	TOPSIS	TOPSIS	TOPSIS
[7]	♠ △	2/1	MF Tri	TS	CoG	AND
[34]	♠ △	2/1	MF Tri/Tra	Ma	CoG	AND
Int-FLBCC	♡ ◇ ♠	3/1	Tra MF	Ma	CoA	AND

♠ CloudSim ◇ Matlab ♣ Simulink △ jFuzzyLogic ♡ Juzzy (MF) Função de Pertinência (MF Gau) Gaussiana (MF Tri) Triangular (MF Tra) Trapezoidal (TS) Takagi-Sugeno (Ma) Mamdani (CoG) Center of Gravity (CoA) Center of Area (TOPSIS) Technique for Order Performance by Similarity to Ideal (FQL) Fuzzy Q-Learning

fuzzy adaptativo baseado em limites para detectar *hosts* sobre-carregados e subcarregados visando a redução do consumo energético, da violação do SLA, e do número de migrações, com objetivo de melhorar a utilização dos recursos aplicando a tomada de decisão fuzzy adaptativa. As métricas consideradas no sistema de inferência fuzzy são: (i) consumo de energia, (ii) violação de SLA, (iii) número de migrações e (v) dados de carga de trabalho que definem o valor do uso de recursos. O algoritmo proposto gera regras dinamicamente e atualiza as funções de pertinência para se adaptar às mudanças na carga de trabalho. Ele é validado com uma carga de trabalho real do PlanetLab. Os resultados da simulação demonstraram que o algoritmo proposto supera significativamente os outros algoritmos competitivos.

Em Arianyan et al. [33] os autores não apenas propõem um novo procedimento de consolidação com algoritmo de Tensão Dinâmica e Escalonamento de Frequência (*Dynamic Voltage and Frequency Scaling - DVFS*) para eliminar as inconsistências entre as técnicas de consolidação e DVFS, mas também propõe novos algoritmos de DVFS para quatro subproblemas do problema de consolidação online, bem como um novo regulador DVFS. Os quatro subproblemas de consolidação são: (1) determinação de Máquinas Físicas (*Physical Machine - PM*) sobrecarregadas, (2) determinação de PM sub-carregados, (3) Seleção de VM que devem ser migradas de PM sobrecarregados e (4) colocação de VM migrantes em PM [37].

Além disso, este artigo considera os critérios importantes, incluindo (i) CPU, (i) RAM e (iii) largura de banda da rede em todos os algoritmos propostos. E ainda, este artigo propõe um novo algoritmo multifuncional de alocação que possibilita aos administradores de recursos aplicar a importância de diferentes critérios na solução de gerenciamento de recursos usando operador de agregação fuzzy ponderados. Os resultados de experimentos obtidos a partir de uma extensa

avaliação das políticas propostas na ferramenta CloudSim mostrou que a proposta superou o gerenciamento de recursos de soluções existentes devido à otimização simultânea de critérios importantes no processo de tomada de decisão. Nesta pesquisa foi concluído que a combinação das políticas propostas para o processo de gerenciamento de recursos em *data centers* da nuvem obteve notável redução nas métrica de consumo energético, violação de SLA e número de migrações em comparação com o estado da arte.

Em Haratian et al. [7], a questão abordada na pesquisa inclui como reduzir o número de violações de SLA com base na otimização de recursos alocados para usuários que aplicam um ciclo de controle autônomo e um sistema de gerenciamento de conhecimento fuzzy. Neste artigo, um *framework* denominado *Adaptive and Fuzzy Resource Management (AFRM)* é proposto, na qual os últimos valores de uso dos recursos de cada VM são reunidos através dos sensores do ambiente e enviados para um controlador fuzzy. As variáveis usadas para o sistema de inferência fuzzy são: (i) utilidade e (ii) escala; propondo também (iii) coeficiente dinâmico com base na carga de trabalho do ambiente da CN.

Na sequência, o AFRM analisa as informações recebidas para tomar decisões sobre como realocar os recursos em cada iteração de um ciclo de controle auto-adaptativo. Todas as funções de pertinência e regras fuzzy são atualizadas dinamicamente com base nas alterações de carga de trabalho para satisfazer os requisitos de QoS. Dois conjuntos de experimentos foram conduzidos para avaliar o AFRM em comparação com abordagens baseadas em regras estáticas em termos de Eficiência na Alocação de Recursos, violações de SLA e custo computacional, aplicando cargas de trabalho ALTA, MÉDIA, MÉDIA-ALTA e BAIXA. Os resultados revelam que o AFRM supera as abordagens fuzzy baseadas em regras estáticas em vários aspectos. Os testes consideraram o uso do jFuzzyLogic enquanto alternativa para análise e modelagem do sistema de

inferência fuzzy e o CloudSim para simulação.

No trabalho de [34], é apresentada uma proposta para ajuste dinâmico do valor limite de subutilização e sob utilização do host físico visando minimizar o número de migrações em diferentes ambientes de carga de trabalho. A abordagem proposta, denominada *Dynamic Threshold-based Fuzzy Approach* (DTFA) é uma abordagem baseada em limite fuzzy usada para ajustar os valores limites das máquinas físicas no ambiente da nuvem. A proposta permite reduzir o número de migrações causadas pela sobrecarga, e cumprir os SLA aplicando um sistema de inferência fuzzy, que recebe como entrada a capacidade restante dos recursos computacionais, e temperatura do *host* físico. Esses valores são calculados a partir da utilização atual previamente prevista dos recursos computacionais. O valor previsto é obtido através da utilização do método de média móvel integrada auto regressiva (*Autoregressive Integrated Moving Average* - ARIMA).

O sistema de inferência fuzzy utilizado por DTFA é com base em Mamdani, modelando os conjuntos fuzzy através de funções de pertinência trapezoidais e triangulares. Na fase de defuzzificação é empregado o método do centro da área. Três conjuntos de experimentos com diferentes cargas de trabalho foram realizados para validar a abordagem proposta. Os resultados obtidos demonstraram que o DTFA superou soluções existentes em média de 22,52%, 45,63% e 56,68% no que trata consumo energético, número de migração de VM e violações de SLA, respectivamente.

A proposta de *Int-FLBCC* detalhada nas próximas seções deste artigo fornece uma abordagem com T2FL e ordens admissíveis para tratar as incertezas e o comportamento dinâmico na seleção de máquinas físicas para alocação de VM no LB em ambientes de CN, diferentemente dos trabalhos relacionados. A realização dos testes foi através de simulações com o kit ferramentas CloudSim 3.0.3.

Analisando a Tabela 1 destaca-se que na maioria das pesquisas é utilizado o framework CloudSim [18] como ferramenta para simulação do ambiente computacional, e o projeto de jFuzzyLogic [38] para o sistema de inferência fuzzy, podendo este, ser agregado ao CloudSim com maior facilidade, pois ambos são desenvolvidos com a mesma tecnologia. O Matlab, Simulink e FuzzyLite [39] também apresentam-se como alternativas para modelagem, implementação e testes dos sistemas de inferência fuzzy. Na maior parte dos projetos são tratadas duas e três variáveis de entrada. O método de fuzzificação é aplicado através das MF dos tipos Trapezoidais, Triangulares e Gaussianas. O processo de inferência utilizado é com base em Mamdani [40] e Takagi e Sugeno [41], considerando uma base de regras com conectivos lógicos do tipo “AND” aplicando normas triangulares. E por fim, a etapa de defuzzificação aplicando o centro da área.

5. Int-FLBCC: Visão Geral dos Processos

A visão geral dos processos envolvidos na operação do *Int-FLBCC* é descrita pelo fluxograma apresentado na Figura 3. O mesmo recebe como entrada o estado atual dos recursos da

nuvem computacional, e a carga de trabalho a ser atribuída ao ambiente da nuvem.

Na sequência os recursos computacionais são identificados, (i) o estado atual do ambiente da CN; e (ii) as máquinas que estão disponíveis, sendo as informações de ambos recursos armazenadas em uma lista. A seguir são coletadas as informações de CC, RAM e PC de cada máquina disponível, adaptando-as para uma Escala Padrão, onde então estarão prontas para ingressarem no módulo T2FL, contemplando as três etapas de um sistema de inferência fuzzy, que são elas: (i) Fuzzificação, (ii) Inferência e (iii) Defuzzificação, desta forma obtendo o intervalo correspondente ao nível de utilização de cada máquina disponível, armazenando estes em uma Lista de Níveis de Utilização, e após verificado se ainda há alguma máquina disponível não analisada. Enquanto houver, analisa os dados e armazena na lista de Níveis de Utilização. Assim é obtida a lista contendo os intervalos de nível de uso de cada *host* do ambiente da CN.

A seguir, a Lista de Níveis de Utilização entra na etapa de ordenação, onde são aplicados os métodos das Ordens Admissíveis: (i) Lex1; (ii) Lex2 e (iii) Xu and Yager. Neste caso, têm-se uma solução para o problema de incompatibilidade dos intervalos no procedimento, pois nem sempre uma lista de intervalos pode ser ordenada por métodos convencionais (ordenação usual dos intervalos reais).

Esta lista é enviada ao Tomador de Decisões do Escalonador que, por sua vez, irá mapear quais os hosts devem ter VM realocadas, e quais podem receber VM, isto ocorre de acordo com os níveis de utilização e o método de ordenação selecionado. Onde máquinas com valores de utilização muito baixo podem ter suas VMs realocadas, e serem desligadas, visando um menor consumo energético.

No caso das máquinas com valores de utilização muito alto, poderá ocorrer a realocação de suas VM na tentativa de evitar perda de desempenho do ambiente. Desta forma, as máquinas disponíveis, que estiverem entre esses dois casos, serão as primeiras a receberem VM.

6. Int-FLBCC: Modelagem do Sistema Fuzzy Tipo-2

O *Int-FLBCC* é responsável por verificar o Nível de Utilização (Ut) do host relacionado ao balanceamento de carga na CN. O sistema de *Int-FLBCC* considera uma base de regras atuando em três etapas: Fuzzificação, Inferência e Defuzzificação retornando como saída o nível de utilização de cada *host* do ambiente de nuvem computacional.

O diagrama de blocos apresentado na Figura 4 consiste em três entradas PC, CC e RAM, com cada uma delas considerando três VLs no T2FS de *Int-FLBCC*, em sua totalidade a modelagem utiliza funções de pertinência trapezoidais. A base de regras consiste em vinte e sete regras usando a abordagem de inferência de Mamdani. A saída Ut, também considerava a modelagem através funções de pertinência trapezoidais para os FST2, e com três VLs, mais detalhes da arquitetura

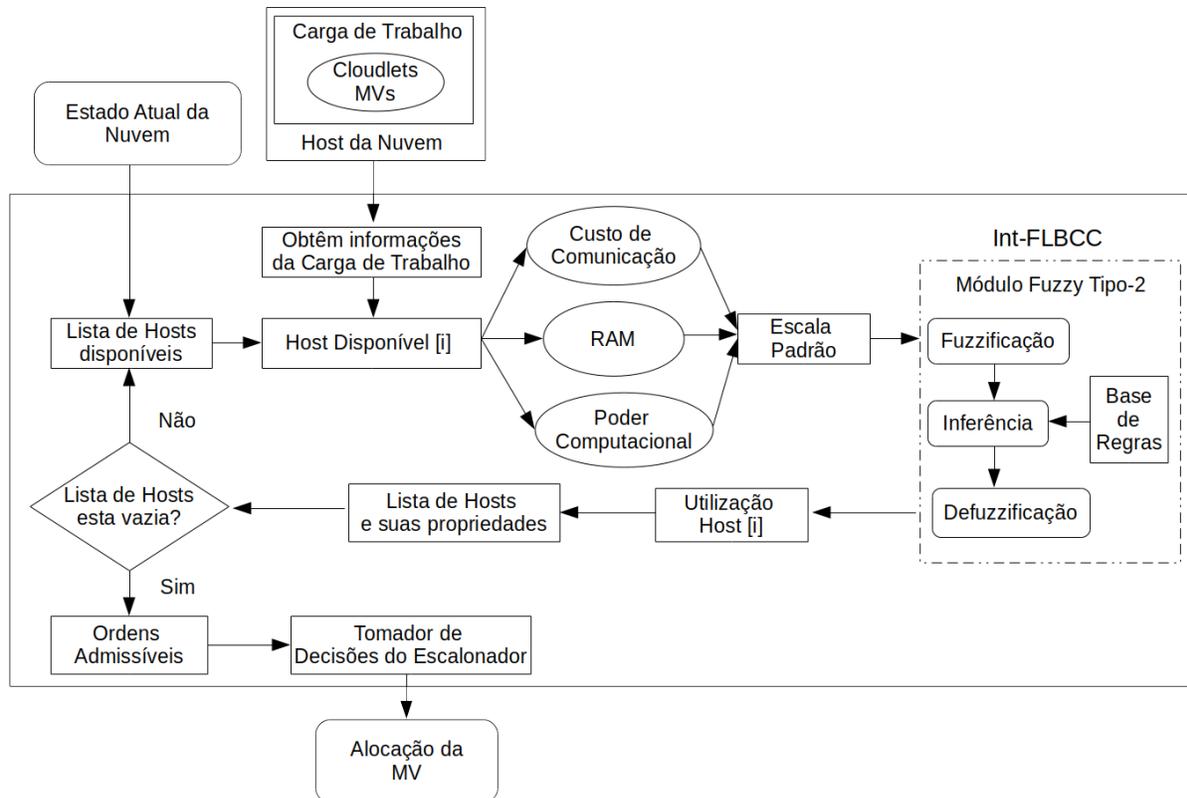


Figure 3. Fluxograma dos processos de *Int-FLBCC*

são apresentados a seguir.

6.1 Definição das Funções de Pertinência

Através do estudo das variáveis junto a um especialista foram transformadas as VLs relativas a cada uma das variáveis de incerteza em FST2, usando a forma trapezoidal na representação gráfica das suas funções de pertinência.

A leitura das configurações aplicadas no ambiente de computação em nuvem simulado é realizada na mensuração dos três atributos PC, CC e RAM. Estes valores são aplicados a uma escala padrão adotada considerando o intervalo $[0; 10]$, para PC Eq. (3), CC Eq. (4) e RAM Eq. (5) na obtenção dos correspondentes graus de pertinência.

$$PC = (h_i(MM)/MaxPC * 10) \quad (3)$$

$$CC = ((10 * h_i(UtoB))/MinCC) \quad (4)$$

$$RAM = (h_i(UtoR)/MaxRam) * 10 \quad (5)$$

Considerando a descrição dos parâmetros a seguir:

- h_i representando o $host(i)$ do ambiente da nuvem;
- MM as Máximo de MIPS disponível no $host i$ considerando todos os *Processing Elements (PE)*;
- $UtoB$ representando a utilização de largura de banda do $host i$;
- $UtoR$ representando a utilização de RAM do $host i$;

- $MaxPC$, valor total em MIPS do melhor $host$ do ambiente da nuvem;
- $MinCC$ o menor custo de comunicação no ambiente da nuvem;
- $MaxRAM$ o valor de RAM do melhor $host$.

As Equações (3), (4) e (5) estão associadas as T2FS das Figuras 5(a), 5(b), 5(c) e 5(d) modelando as respectivas variáveis PC, CC, RAM e Ut. A seguir são descritos os Termos Linguísticos utilizados na definição dos FST2.

- FST2 da variável PC são os seguintes: “Limitado” (PCL), “Razoável” (PCR) e “Elevado” e considerando (PCE - como melhor caso). Sendo $PC = a$ e $a \in [0; 10]$, obtemos as FPT2 apresentadas em modo gráfico na Figura 5(a).
- FST2 da variável CC são: “Pequeno” (CCP - melhor caso), “Médio” (CCM) e “Grande” (CCG). Sendo $CC = b$ e $b \in [0; 10]$, têm-se as FPT2 da Figura 5(b).
- FST2 da variável RAM são: “Baixo” (RAMB - melhor caso), “Médio” (RAMM) e “Grande” (RAMG). Sendo que para $RAM = c$ e $c \in [0; 10]$, têm-se as FPT2 da Figura 5(c).
- FST2 da variável Ut usados nesse caso são: “Baixa” (UB), “Média” (UM) e “Alta” (UA). Sendo $U = d$ e $d \in [0; 10]$.

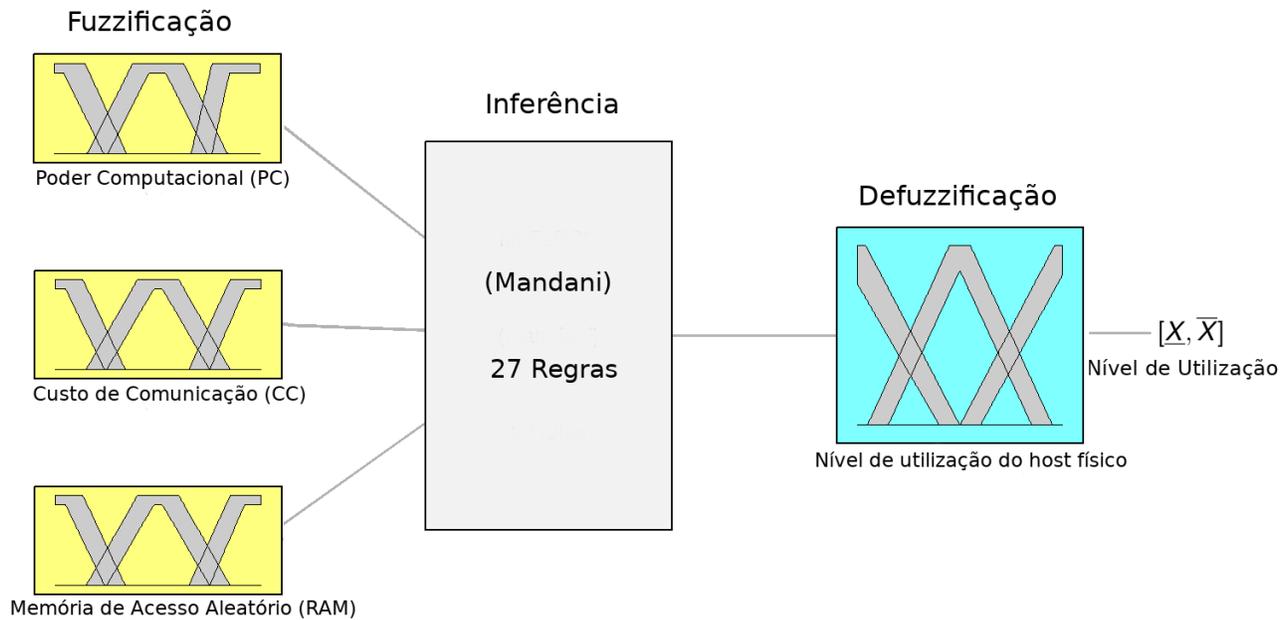


Figure 4. Diagrama de blocos do *Int-FLBCC*

[0; 10]. O nível de utilização U_t das máquinas também é adaptado para escala padrão conforme mostra a Figura 5(d).

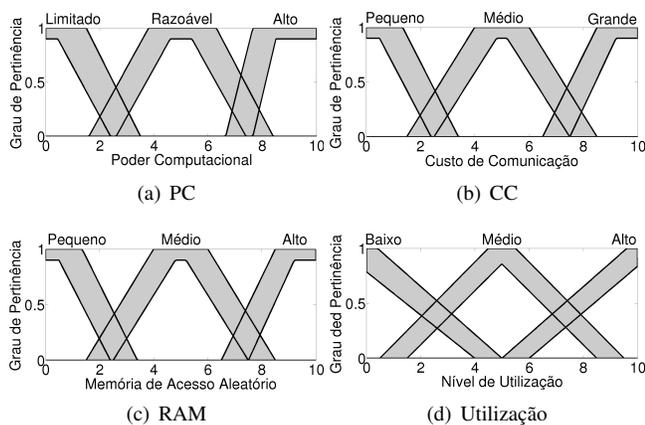


Figure 5. PC, CC, RAM, e U_t na escala padrão

No ambiente da CN o nível de uso dos *hosts* é incerto e impreciso devido a vários fatores, como por exemplo, a flutuação do poder computacional, largura de banda e memória disponível no momento da execução das aplicações submetidas pelos usuários. Considera-se um algoritmo online aplicando abordagem da T2FS que percorre os *hosts* disponíveis na arquitetura da CN obtendo o nível de uso a cada iteração. O algoritmo retorna o nível de utilização do host avaliado com base em um T2FIS (Algoritmo 1).

Entrada : Host h_i

Saída : Nível de Utilização do Host i

```

for  $i \leftarrow 0$  to  $getHostList().size()$  do
  if  $maxPCHost <$ 
     $getHostList().get(i).getTotalMips()$  then
     $maxPCHost \leftarrow$ 
       $getHostList().get(i).getTotalMips();$ 
  end
  if  $minCCHost <$   $getHostList().get(i).getBw()$ 
  then
     $minCCHost \leftarrow$ 
       $getHostList().get(i).getBw();$ 
  end
  if  $maxRamHost <$ 
     $getHostList().get(i).getRam()$  then
     $maxRamHost \leftarrow$ 
       $getHostList().get(i).getRam();$ 
  end
   $AvaliacaoFuzzy f \leftarrow AvaliacaoFuzzy();$ 
   $Utilizacao \leftarrow$ 
     $f.getLevelOfUse(PC,CC,RAM);$ 
  return  $Utilizacao;$ 
end
  
```

Algorithm 1: Avaliação do nível de utilização do host

6.2 Fuzzificação

Nessa etapa, ocorre o mapeamento dos valores de entrada (já ajustados para escala observada na seção 6.1) para o domínio fuzzy, como destaca a Figura 6.

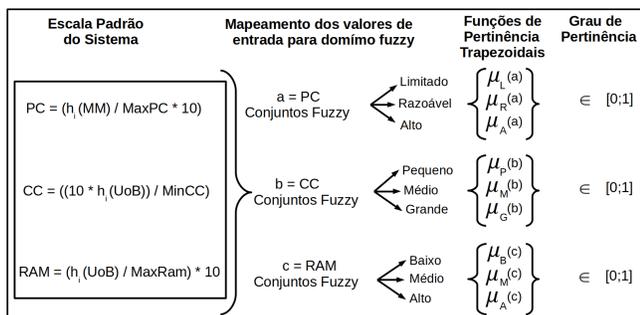


Figure 6. Processo de Fuzzificação

6.3 Base de Regras

A Base de Regras (*Rule Base - RB*) do *Int-FLBCC*, que é apresentada na Tabela 2, foi desenvolvida com o intuito de ser facilmente compreensível e editável, considerando três fatores: (i) as VL nomeiam os FS, tornando a modelagem do sistema mais próxima do mundo real; (ii) são utilizadas conexões lógicas do tipo “AND” para criar a relação entre as variáveis de entrada; (iii) as implicações são do tipo *modus ponens* (modo afirmativo):

Se “ $x_1 \in A_1$ ” E “ $x_2 \in A_2$ ” E “ $x_3 \in A_3$ ” então “ $y \in B$ ”

Table 2. RB do *Int-FLBCC*

Regra	PC	CC	RAM	Utilização
1	limitado	pequeno	baixo	alto
2	limitado	pequeno	médio	médio
3	limitado	pequeno	alto	médio
4	limitado	médio	baixo	alto
5	limitado	médio	médio	médio
6	limitado	médio	alto	médio
7	limitado	alto	baixo	alto
8	limitado	alto	médio	alto
9	limitado	alto	alto	alto
10	razoável	baixo	baixo	médio
11	razoável	baixo	médio	médio
12	razoável	baixo	alto	médio
13	razoável	médio	baixo	alto
14	razoável	médio	médio	médio
15	razoável	médio	alto	baixo
16	razoável	alto	baixo	alto
17	razoável	alto	médio	médio
18	razoável	alto	alto	médio
19	alto	baixo	baixo	médio
20	alto	baixo	médio	baixo
21	alto	baixo	alto	baixo
22	alto	médio	baixo	médio
23	alto	médio	médio	médio
24	alto	médio	alto	médio
25	alto	alto	baixo	alto
26	alto	alto	médio	alto
27	alto	alto	alto	médio

6.4 Inferência

No processo de Inferência, ocorrem as operações entre os FS, combinação dos antecedentes das regras e implicações via o operador *modus ponens generalizado*. Conforme a Figura 7, esta fase ocorre em três etapas que são detalhadas na sequência.

- (i) Aplicação da Operação Fuzzy: Como as regras são formadas pelo operador fuzzy “AND”, a aplicação utiliza o método MIN (mínimo) sobre os valores retornados da fuzzificação;
- (ii) Aplicação do Método de Implicação Fuzzy: realizada pela combinação entre o valor obtido na aplicação do operador fuzzy e os valores do FS da saída de cada regra, utilizando o método MIN (mínimo) sobre estas combinações;
- (iii) Aplicação do Método de Agregação Fuzzy: considera a composição dos resultados fuzzy da saída de cada regra, utilizando o método MAX (máximo), assim criando uma única região fuzzy para ser analisada pelo próximo processo do módulo fuzzy.

6.5 Defuzzificação

A defuzzificação é a transformação da região resultado da inferência, a técnica utilizada foi o Centro da Área. Esse método calcula o centroide (x) da área composta pela saída T2FIS (união de todas as contribuições de regras demonstradas nas seções 6.3 e 6.4), definido pela seguinte fórmula:

$$u = \frac{\sum_{i=1}^N u_i \mu_{OUT}(u_i)}{\sum_{i=1}^N \mu_{OUT}(u_i)} \tag{6}$$

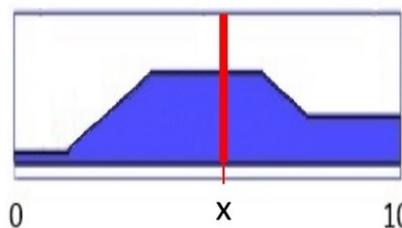


Figure 8. Defuzzificação Método Centro da Área

7. Int-FLBCC: Métricas Avaliadas

Esta seção apresenta a definição das métricas avaliadas nesta pesquisa, usadas nas etapas da consolidação dinâmica de VM eficiente em energia e desempenho, que são aplicadas sob restrições de QoS.

7.1 Modelo de Consumo Energético

O consumo de energia dos *data centers* da CN é determinado principalmente pela CPU, memória, armazenamento em disco, fontes de alimentação e sistemas de refrigeração [42].

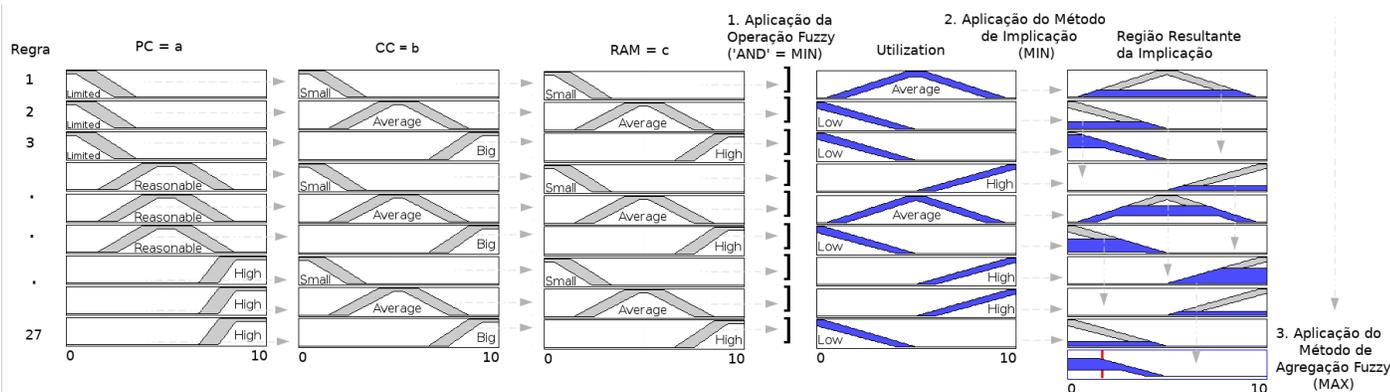


Figure 7. Processo de inferência do Int-FLBCC

As pesquisas [43, 3] demonstraram que o consumo de energia dos servidores pode ser descrito com precisão por uma relação linear entre o consumo de energia e a utilização da CPU, mesmo quando o DVFS é aplicado. O motivo está no número limitado de estados que podem ser configurados para a frequência e tensão de uma CPU e o fato de que a escala de voltagem e desempenho não é aplicada a outros componentes do sistema, como interfaces de rede e memória.

No entanto, devido à proliferação de CPUs com vários núcleos e virtualização, os servidores modernos normalmente são equipados com grandes quantidades de memória, o que acarretou o aumento do consumo de energia por um servidor [42]. Esse fato, combinado com a dificuldade de modelar o consumo de energia das modernas CPUs multinúcleo, torna a construção de modelos analíticos precisos um complexo problema de pesquisa. Portanto, em vez de usar um modelo analítico de consumo de energia por um servidor, este trabalho utiliza dados reais sobre o consumo de energia fornecidos pelos resultados do benchmark SPECpower⁵.

Foram selecionadas duas configurações de servidores descritos na Seção 8, Figura 9. As características de configuração e consumo de energia dos servidores selecionados são mostradas na Tabela 3. A razão pela qual os servidores com mais núcleos foram escolhidos, se justifica pela importância de simular um grande número de servidores para avaliar o efeito da consolidação de VM.

7.2 Métricas de Violação de SLA

O atendimento aos requisitos de QoS é altamente importante para ambientes de CN. Os requisitos de QoS geralmente são formalizados via forma de SLA e determinados em termos de características como taxa de transferência mínima ou tempo máximo de resposta entregue pelo sistema implantado. Como essas características podem variar para diferentes aplicativos, é necessário definir uma métrica independente da carga de trabalho que possa ser usada para avaliar a QoS entregue para qualquer VM implementada na IaaS.

Este trabalho define que os requisitos de SLA são satisfeitos quando 100% do desempenho solicitado pelos aplica-

tivos dentro de uma VM é fornecido a qualquer momento, limitado apenas pelos parâmetros da VM. Duas métricas para medir o nível de violações do SLA em um ambiente IaaS são utilizadas: (1) a fração de tempo durante a qual os *hosts* ativos experimentaram a utilização da CPU de 100%, Fração de tempo de sobrecarga (FTS); e (2) a degradação geral do desempenho por VM devido a migrações, caracterizando a degradação do desempenho devido a migrações (DDM) Eq. 7. O raciocínio por trás da métrica FTS é a observação de que, se um *host* está com 100% de utilização, o desempenho dos aplicativos a ele associados fica limitado pela capacidade do *host*; portanto, as VM não estão sendo fornecidas com o nível de desempenho necessário.

$$FTS = \frac{1}{N} \sum_{i=1}^N \frac{T_{S_i}}{T_{a_i}} \quad DDM = \frac{1}{M} \sum_{j=1}^M \frac{C_{d_j}}{C_{r_j}} \quad (7)$$

onde N é o número de *hosts*; T_{S_i} é o tempo total durante o qual o *host* i ficou com utilização de 100%, levando a uma violação do SLA; T_{a_i} é o total de *host* i ativo, e que estão executando VM; M é o número de VM; C_{d_j} é a estimativa da degradação do desempenho da VM j causada por migrações; C_{r_j} é a capacidade total da CPU solicitada pela VM j durante sua vida útil.

Neste trabalho C_{r_j} é estimado em 10% da utilização da CPU no MIPS durante todas as migrações da VM j .

As métricas FTS e DDM caracterizam de forma independente o nível de violações de SLA no sistema, portanto aplica-se uma métrica combinada que abrange a degradação do desempenho devido aos fatores: (1) sobrecarga do *host* e (2) às migrações de VM, denominada violação de SLA (SLAV). A métrica é calculada conforme mostrado na Eq. (8).

$$SLAV = FTS \cdot DDM \quad (8)$$

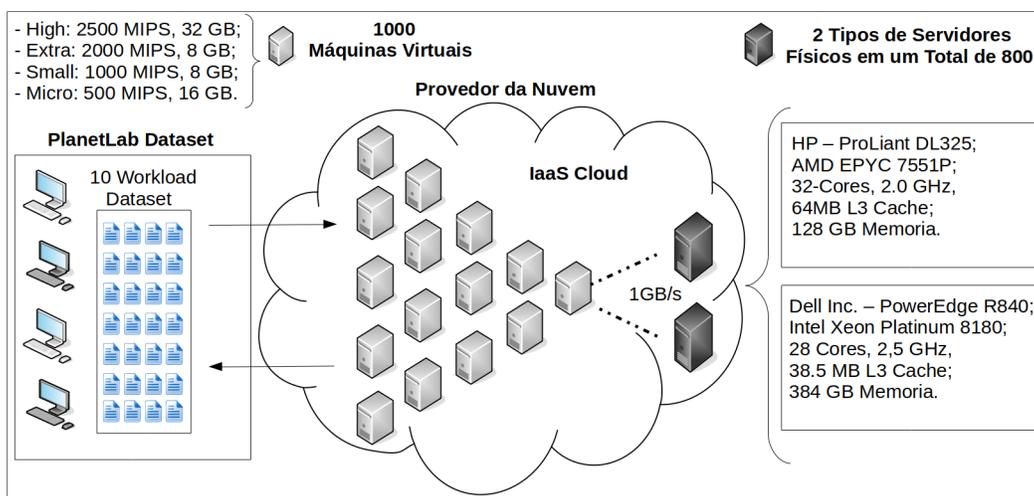
8. Int-FLBCC: Resultados

A configuração do ambiente de CN para os testes empregou o CloudSim [18], constituído em um conjunto de ferramentas

⁵(http://www.spec.org/power_ssj2008/)

Table 3. Consumo energético dos servidores selecionados em diferentes níveis de carga em Watts

Servidor	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
ProLiant DL325 Gen10	61.7	92.3	105	116	127	136	145	153	161	170	181
PowerEdge R840	88.3	137	148	160	172	184	199	218	246	269	295

**Figure 9.** Características da Arquitetura da Nuvem Simulada

para modelagem e simulação de infraestruturas e serviços da CN. Uma biblioteca Java de código aberto chamada Juzzy [44] foi usada para implementar o F2TIS.

Para todas as experimentações, cargas de trabalho do mundo real fornecidas como parte do projeto CoMon [45], uma infraestrutura de monitoramento para a PlanetLab⁶ foram empregadas.

O ambiente de nuvem IaaS representado para os testes considerou *data centers* de grande escala que compreendeu 800 *hosts* físicos heterogêneos, contemplando dois tipos de configurações como descritas na Figura 9.

Os resultados das execuções, bem como o o projeto de *Int-FLBCC* estão disponíveis no GitHub⁷ em uma versão estendida de CloudSim 3.0.3.

A frequência das CPU do servidor é descrita em MIPS (*Millions of Instructions Per Second*). Metade dos hosts são ProLiant DL325 Gen 10 com 4721 MIPS para cada núcleo, e a outra metade consiste no servidor PowerEdge R840 com 4520 MIPS para cada núcleo. Cada servidor é modelado para ter 1 GB/s de largura de banda de rede.

As características dos tipos de VM são tipos de instâncias: (1) Instância de CPU de Média Alta (2500 MIPS, 32 GB); (2) Instância Extra Grande (2000 MIPS, 8 GB); (3) Instância Pequena (1000 MIPS, 8 GB); e (4) Micro Instância (500 MIPS, 16 GB).

O intervalo de medição de uso (intervalo de escalonamento) é de 5 minutos, baseado em [37]. Esse intervalo tem como objetivo obter a oscilação do nível de utilização das máquinas durante a execução.

⁶(<https://www.planet-lab.org>)

⁷(<https://github.com/brunomourapaz/CloudSim>)

As características de cada carga de trabalho são apresentadas na Tabela 4. Foram utilizados dados de carga de trabalho da CPU para mais de 1000 VM de servidores localizados em mais de 500 locais em todo o mundo.

Table 4. Características das cargas de trabalho

Workload	VM	Média (%)	Desvio Padrão (%)
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
20110420	1033	10.43	15.21

Dez conjuntos de dados de cargas de trabalho coletados em dias diferentes foram aplicados, que são alocados para cada VM. Nas execuções dos experimentos foram utilizados os algoritmos de alocação descritos a seguir, e agregando as abordagens das Ordens Admissíveis discutidas na Seção 3.

- **Inter Quartile Range (IQR)**: algoritmo baseado na medida de dispersão estatística;
- **Local Robust Regression (LRR)**: método de regressão local que agrega ao modelo o tratamento para dados discrepantes (*outliers*) através de um método iterativo dos mínimos quadrados.

- **Median Absolute Deviation (MAD):** limite de utilização adaptativo, um algoritmo heurístico para ajuste automático do limite de utilização com base na análise estatística de dados históricos coletados durante a vida útil das máquinas virtuais.
- **Static Threshold (THR):** limite de utilização de CPU, distinguindo os estados de não sobrecarga, e sobrecarga do host físico, através da comparação da utilização atual do host com um limite fixo definido, caso o limite for excedido o algoritmo detecta sobrecarga do host.

Em cada execução, a política de seleção de VM utilizada foi a *Random Selection (RS)*.

Os resultados são apresentados de forma gráfica através de boxplot, constituído das seguintes informações estatísticas: (i) o mínimo, (ii) primeiro quartil (Q1), (iii) a mediana, (iv) terceiro quartil (Q3) e (v) o máximo. As hastes inferiores e superiores se estendem, respectivamente, do quartil inferior até o menor valor não inferior ao limite inferior e, do quartil superior até o maior valor não superior ao limite superior.

A avaliação do consumo de energia é apresentada através da Figura 10. Pode-se observar que a abordagem de seleção de *host* proposta alcançou ganhos consideráveis em economia de energia: (i) política IQR_XY em relação ao IQR alcançou 8.83% de ganho, no caso de (2) MAD_Lex2 em relação para MAD alcançou 8.72%, em (3) THR_XY comparado a THR 22.43%.

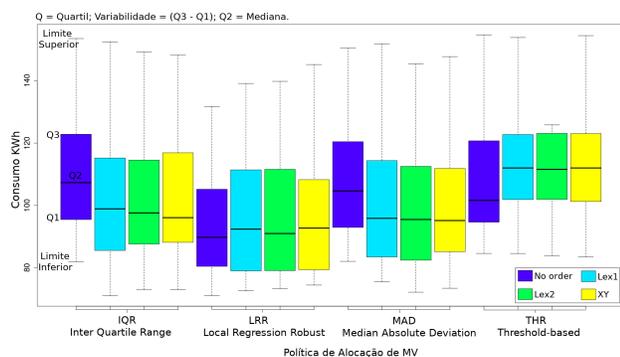


Figure 10. Consumo Energético

O boxplot da Figura 11 apresenta resultados de SLA. A abordagem proposta alcançou resultados de (1) 6.11% com IQR_XY comparado a IQR, para (2) LRR_Lex2 4.02% confrontado com LRR, (3) THR_Lex1 25% em oposição a THR, e finalmente, (3) MAD_XY analisado com MAD obteve 9.06%, correspondendo à melhor média de violação de SLA.

9. Considerações Finais

Neste trabalho, apresentamos o *Int-FLBCC*, uma nova abordagem para a seleção de *hosts* na CN com uso eficiente de energia empregando a T2FL e Ordens Admissíveis, como parte da consolidação dinâmica de VM.

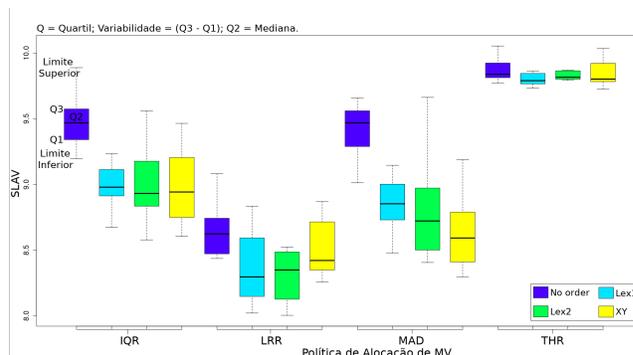


Figure 11. Média de Violação de SLA

O resultado das simulações foi comparado com quatro algoritmos de alocação de VM bem conhecidos (IQR, LRR, MAD e THR).

Nossa avaliação usando traços reais de carga de trabalho mostrou que o método proposto reduziu o consumo de energia enquanto manteve QoS e SLA, essa é a principal vantagem do *Int-FLBCC*. Na maioria dos casos, a abordagem proposta usando T2FL obteve bons resultados sobre a eficiência energética, alcançando ganhos de 8.83%, 8.72% e 22.43% com IQR, MAD e LRR, perdeu 4.19% com o uso da ordem de Xu e Yager. Considerando o SLA, os ganhos foram 9.06% de MAD com a ordem de Xu e Yager, e 25% com THR aplicando a ordenação de Lex1.

Como trabalhos futuros pretendemos agregar ao modelo outras variáveis com objetivo de melhorar os níveis de SLA e eficiência energética, e ainda, aplicar técnicas dinâmicas na concepção do T2FS.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), PqG-FAPERGS (310106/2016-8), Pq-CNPq (309160/2019-7) e FAPERGS/CNPq 12/2014 - PRONEX (16/2551-0000488-9) GREEN-CLOUD: Computação Sustentável em Cloud.

Contribuições dos autores

Todos os autores contribuíram de forma significativa para o desenvolvimento do trabalho, nas discussões sobre a modelagem, bem como na análise e desenvolvimento das simulações. A parte relacionada a introdução, especificações de trabalhos relacionados e desenvolvimento das simulações, teve maior contribuição dos autores Bruno Moura Paz de Moura e Guilherme Bayer Schneider. Os autores Adenauer Correa Yamin e Renata Hax Sander Reiser contribuíram de forma expressiva na escrita, organização, correção do texto e orientação.

References

- [1] TOOSI, A. N.; BUYYA, R. A fuzzy logic-based controller for cost and energy efficient load balancing in geodistributed data centers. In: *2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC)*. [S.l.: s.n.], 2015. p. 186–194.
- [2] MELL, P.; GRANCE, T. et al. The nist definition of cloud computing. Computer Security Division, Information Technology Laboratory, National . . . , 2011.
- [3] SHEHABI, A. et al. United states data center energy usage report. 2016.
- [4] NATHANI, A.; CHAUDHARY, S.; SOMANI, G. Policy based resource allocation in iaas cloud. *Future Generation Computer Systems*, Elsevier, v. 28, n. 1, p. 94–103, 2012.
- [5] BELOGLAZOV, A.; BUYYA, R. 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*, v. 24, n. 7, p. 1366–1379, July 2013.
- [6] ZHANG, Q.; CHENG, L.; BOUTABA, R. Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, v. 1, n. 1, p. 7–18, May 2010. Disponível em: <https://doi.org/10.1007/s13174-010-0007-6>.
- [7] HARATIAN, P. et al. An adaptive and fuzzy resource management approach in cloud computing. *IEEE Transactions on Cloud Computing*, p. 1–1, 2018.
- [8] MOURA, B. M. P. et al. Int-fGrid: BoT Tasks Scheduling Exploring Fuzzy Type-2 in Computational Grids. In: *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. [S.l.: s.n.], 2018. p. 1–8.
- [9] HAMDQA, M.; TAHVILDARI, L. Cloud computing uncovered: a research landscape. In: *Advances in Computers*. [S.l.]: Elsevier, 2012. v. 86, p. 41–85.
- [10] HOLICK, M. F. et al. Evaluation, treatment, and prevention of vitamin d deficiency: an endocrine society clinical practice guideline. *The Journal of Clinical Endocrinology & Metabolism*, Oxford University Press, v. 96, n. 7, p. 1911–1930, 2011.
- [11] CHIEU, T. C. et al. Dynamic scaling of web applications in a virtualized cloud computing environment. In: *IEEE. 2009 IEEE International Conference on e-Business Engineering*. [S.l.], 2009. p. 281–286.
- [12] BATISTA, B. G. *Modelos de negócio para ambientes de computação em nuvem que consideram atributos de qos relacionados a desempenho e a segurança*. Tese (Doutorado) — Universidade de São Paulo, 2016.
- [13] ZHANG, Q.; CHENG, L.; BOUTABA, R. Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications*, Springer, v. 1, n. 1, p. 7–18, 2010.
- [14] FERDAUS, M. H. et al. Virtual machine consolidation in cloud data centers using aco metaheuristic. In: *SPRINGER. European conference on parallel processing*. [S.l.], 2014. p. 306–317.
- [15] AHMED, A.; SABYASACHI, A. S. Cloud computing simulators: A detailed survey and future direction. In: *IEEE. Advance Computing Conference (IACC), 2014 IEEE International*. [S.l.], 2014. p. 866–872.
- [16] KLIAZOVICH, D.; BOUVRY, P.; KHAN, S. U. Greencloud: a packet-level simulator of energy-aware cloud computing data centers. *The Journal of Supercomputing*, Springer, v. 62, n. 3, p. 1263–1283, 2012.
- [17] WICKREMASINGHE, B.; CALHEIROS, R. N.; BUYYA, R. Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications. In: *IEEE. Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*. [S.l.], 2010. p. 446–452.
- [18] CALHEIROS, R. N. et al. Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and experience*, Wiley Online Library, v. 41, n. 1, p. 23–50, 2011.
- [19] ZADEH, L. The concept of a linguistic variable and its application to approximate reasoning—i. *Information Sciences*, v. 8, n. 3, p. 199 – 249, 1975.
- [20] MENDEL, J. M. Fuzzy sets for words: a new beginning. In: *Fuzzy Systems, 2003. FUZZ '03. The 12th IEEE International Conference on*. [S.l.: s.n.], 2003. v. 1, p. 37–42.
- [21] KARNIK, N. N.; MENDEL, J. M. Introduction to type-2 fuzzy logic systems. In: *1998 IEEE International Conference on Fuzzy Systems Proceedings. IEEE World Congress on Computational Intelligence*. [S.l.: s.n.], 1998. v. 2, p. 915–920 vol.2.
- [22] MENDEL, J. M.; JOHN, R. I.; LIU, F. Interval type-2 fuzzy logic systems made simple. *IEEE Trans. Fuzzy Systems*, v. 14, n. 6, p. 808–821, 2006.
- [23] GEHRKE, M.; WALKER, C.; WALKER, E. Some comments on interval valued fuzzy sets. *International Journal of Intelligent Systems*, v. 11, n. 10, p. 751–759, 1996.
- [24] KLEMENT, E.; MESIAR, R.; PAP, E. Triangular norms. position paper I: basic analytical and algebraic properties. *Fuzzy Sets and Systems*, v. 143, n. 1, p. 5–26, 2004.
- [25] WU, D.; NIE, M. Comparison and practical implementation of type-reduction algorithms for type-2 fuzzy sets and systems. In: *FUZZ-IEEE*. IEEE, 2011. p. 2131–2138. Disponível em: <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=5976945>.
- [26] ZAPATA, H. et al. Interval-valued implications and interval-valued strong equality index with admissible orders. *International Journal of Approximate Reasoning*, Elsevier, v. 88, p. 91–109, 2017.

- [27] BUSTINCE, H. et al. Generation of linear orders for intervals by means of aggregation functions. *Fuzzy Sets and Systems*, Elsevier, v. 220, p. 69–77, 2013.
- [28] XU, Z.; YAGER, R. R. Some geometric aggregation operators based on intuitionistic fuzzy sets. *International journal of general systems*, Taylor & Francis, v. 35, n. 4, p. 417–433, 2006.
- [29] MASOUMZADEH, S. S.; HLAVACS, H. Integrating VM selection criteria in distributed dynamic VM consolidation using Fuzzy Q-Learning. In: *Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013)*. [S.l.: s.n.], 2013. p. 332–338.
- [30] PORTALURI, G. et al. Power consumption-aware virtual machine allocation in cloud data center. In: *2016 IEEE Globecom Workshops (GC Wkshps)*. [S.l.: s.n.], 2016. p. 1–6.
- [31] SINGH, S.; CHANA, I. Earth: Energy-aware autonomic resource scheduling in cloud computing. *Journal of Intelligent & Fuzzy Systems*, IOS Press, v. 30, n. 3, p. 1581–1600, 2016.
- [32] SALIMIAN, L.; ESFAHANI, F. S.; NADIMI-SHAHRAKI, M.-H. An adaptive fuzzy threshold-based approach for energy and performance efficient consolidation of virtual machines. *Computing*, v. 98, n. 6, p. 641–660, Jun 2016. Disponível em: <https://doi.org/10.1007/s00607-015-0474-5>.
- [33] ARIANYAN, E.; TAHERI, H.; KHOSHDEL, V. 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*, v. 78, p. 43 – 61, 2017. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1084804516300169>.
- [34] ALSADIE, D. et al. DTFA: A Dynamic Threshold-Based Fuzzy Approach for Power-Efficient VM Consolidation. In: *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*. [S.l.: s.n.], 2018. p. 1–9.
- [35] GLORENNEC, P. Y.; JOUFFE, L. Fuzzy Q-learning. In: *Proceedings of 6th International Fuzzy Systems Conference*. [S.l.: s.n.], 1997. v. 2, p. 659–662 vol.2.
- [36] ADAMI, D. et al. A fuzzy logic approach for resources allocation in cloud data center. In: *2015 IEEE Globecom Workshops (GC Wkshps)*. [S.l.: s.n.], 2015. p. 1–6.
- [37] BELOGLAZOV, A.; BUYYA, R. 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.*, John Wiley and Sons Ltd., Chichester, UK, v. 24, n. 13, p. 1397–1420, set. 2012. Disponível em: <http://dx.doi.org/10.1002/cpe.1867>.
- [38] CINGOLANI, P.; ALCALÁ-FDEZ, J. jFuzzyLogic: a robust and flexible Fuzzy-Logic inference system language implementation. In: *2012 IEEE International Conference on Fuzzy Systems*. [S.l.: s.n.], 2012. p. 1–8.
- [39] RADA-VILELA, J. *The FuzzyLite Libraries for Fuzzy Logic Control*. 2018. Disponível em: <https://fuzzylite.com/>.
- [40] MAMDANI, E. H. Application of fuzzy logic to approximate reasoning using linguistic synthesis. In: *Proceedings of the Sixth International Symposium on Multiple-valued Logic*. Los Alamitos, CA, USA: IEEE Computer Society Press, 1976. (MVL '76), p. 196–202. Disponível em: <http://dl.acm.org/citation.cfm?id=800111.803609>.
- [41] TAKAGI, T.; SUGENO, M. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15, n. 1, p. 116–132, Jan 1985.
- [42] MINAS, L.; ELLISON, B. *Energy Efficiency for Information Technology: How to Reduce Power Consumption in Servers and Data Centers*. [S.l.]: Intel Press, 2009.
- [43] FAN, X.; WEBER, W.-D.; BARROSO, L. A. Power provisioning for a warehouse-sized computer. *SIGARCH Comput. Archit. News*, ACM, New York, NY, USA, v. 35, n. 2, p. 13–23, jun. 2007. Disponível em: <http://doi.acm.org/10.1145/1273440.1250665>.
- [44] WAGNER, C. Juzzy - a java based toolkit for type-2 fuzzy logic. In: *2013 IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems (T2FUZZ)*. [S.l.: s.n.], 2013. p. 45–52.
- [45] PARK, K.; PAI, V. S. Comon: A mostly-scalable monitoring system for planetlab. *SIGOPS Oper. Syst. Rev.*, ACM, New York, NY, USA, v. 40, n. 1, p. 65–74, jan. 2006. Disponível em: <http://doi.acm.org/10.1145/1113361.1113374>.