

UNIVERSIDADE FEDERAL DE PELOTAS
Centro de Desenvolvimento Tecnológico
Curso de Bacharelado em Ciência da Computação



Trabalho de Conclusão de Curso

**f-Cloud: Sistema Fuzzy para Avaliação da Utilização de Servidores em um
Ambiente Simulado de Computação em Nuvem**

Guilherme Bayer Schneider

Pelotas, 2019

Guilherme Bayer Schneider

**f-Cloud: Sistema Fuzzy para Avaliação da Utilização de Servidores em um
Ambiente Simulado de Computação em Nuvem**

Trabalho de Conclusão de Curso apresentado
ao Centro de Desenvolvimento Tecnológico
da Universidade Federal de Pelotas, como
requisito parcial à obtenção do título de
Bacharel em Ciência da Computação

Orientadora: Profa. Dr. Renata Hax Sander Reiser
Coorientador: Prof. Dr. Adenauer Corrêa Yamin
Colaborador: Ms. Bruno Moura Paz de Moura

Pelotas, 2019

**Insira AQUI a ficha catalográfica
(solicitada na página da biblioteca)**

**Insira AQUI a folha de aprovação
(será entregue após a banca)**

Dedico este trabalho principalmente minha família que me apoiou incondicionalmente em todo o desenvolvimento e também aos meus colegas que me auxiliaram em toda minha trajetória.

AGRADECIMENTOS

A todos os professores, em especial a Renata, pela orientação e apoio durante o curso e no desenvolvimento deste trabalho.

A todos os amigos do curso, com quais pude compartilhar momentos e experiências da graduação.

A minha família por me acompanharem nesse processo e estarem sempre à disposição quando eu precisei.

RESUMO

SCHNEIDER, Guilherme Bayer. **f-ClouD: Sistema Fuzzy para Avaliação da Utilização de Servidores em um Ambiente Simulado de Computação em Nuvem**. 2019. 62 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2019.

A consolidação dinâmica de máquinas virtuais (MVs) é uma maneira eficaz de melhorar a utilização de recursos e a eficiência energética na Computação em Nuvem (CN). Determinar quando é melhor realocar MVs de um host sobrecarregado é um aspecto dinâmico da consolidação de MVs que influencia diretamente na utilização de recursos e na qualidade de serviço oferecida pelo provedor. Este trabalho apresenta uma nova proposta com uma abordagem de Lógica Fuzzy Tipo-2 para tratar as incertezas e imprecisões quanto da determinação do uso de recursos no ambiente de CN, visando economia energética com degradação mínima de desempenho. Resultados são obtidos em um ambiente simulado, e mostram melhorias na eficiência energética alcançando valores de 15,64% no emprego com o algoritmo *Median Absolute Deviation* (MAD), 15,43% no caso do *Inter Quartile Range* (IQR) e 9,19% com *Local Regression* (LR).

Palavras-Chave: lógica fuzzy tipo-2; computação em nuvem; computação ecológica; alocação de máquinas virtuais

ABSTRACT

SCHNEIDER, Guilherme Bayer. **f-Cloud: Fuzzy System for Evaluating Server Usage in a Simulated Cloud Computing Environment**. 2019. 62 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2019.

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

Keywords: type-2 fuzzy logic; cloud computing; green computing; virtual machine allocation

LISTA DE FIGURAS

1	Serviços oferecidos em CN (BATISTA, 2016)	18
2	Arquitetura de Camadas do CloudSim. Adaptado de (CALHEIROS et al., 2011)	21
3	Taxonomia do gerenciamento de recursos. Adaptado de (BELO-GLAZOV, 2013)	22
4	Número fuzzy triangular	29
5	Número fuzzy trapezoidal	29
6	Número fuzzy senoidal	30
7	Altura, Kernel e suporte de um conjunto fuzzy	30
8	Conjuntos Fuzzy Tipo 2 Intervalar.	32
9	Exemplos de conjuntos fuzzy do tipo 2 intervalares.	33
10	Função de pertinência secundária intervalar em $x=4$	33
11	Conjunto fuzzy do tipo 2.	34
12	Sistema Baseado em Regras Fuzzy 2.	36
13	Modelo adotado para as Funções de Pertinência dos Conjuntos Fuzzy.	39
14	Visão geral das funcionalidades do <i>fCloud</i>	40
15	Visão geral das entradas e saídas do <i>fCloud</i>	41
16	PC na escala padrão	42
17	RAM na escala padrão	43
18	CC na escala padrão	43
19	U na escala padrão	44
20	Processo de Fuzzificação do <i>fCloud</i>	45
21	Processo de Inferência do <i>fCloud</i>	47
22	Fluxograma dos processos operacionais do <i>fCloud</i>	48
23	Características da Arquitetura da Nuvem Simulada.	51
24	Consumo energético	53
25	Violação de SLA por política	54
26	Tempo médio de SLA infringido por host	54
27	VMs migradas	55
28	Degradação do SLA devido a migração de MVs	56
29	Consumo Energético	57
30	Média de Violação de SLA	57

LISTA DE TABELAS

1	Exemplos básicos de t-normas	27
2	Exemplos de t-normas intervalares	35
3	Exemplos básicos de t-conormas intervalares	35
4	Funções de Pertinência PC	42
5	Funções de Pertinência RAM	43
6	Funções de Pertinência CC	44
7	Funções de Pertinência U	45
8	Base de Regras fCloud	46
9	Características dos hosts do ambiente da nuvem	51
10	Configuração das Cloudlets	52

LISTA DE ABREVIATURAS E SIGLAS

LC	Lógica Clássica
LF	Lógica Fuzzy
SF	Sistema Fuzzy
CF	Conjunto Fuzzy
TI	Tecnologia da Informação
LF1	Lógica Fuzzy Tipo-1
LF2	Lógica Fuzzy Tipo-2
CN	Computação em Nuvem
MF	Máquina Física
MV	Máquina Virtual
QoS	<i>Qualidade of Service</i>
SLA	<i>Service Level Agreements</i>
SLO	<i>Service Level Object</i>
TCF	Teoria dos Conjuntos Fuzzy
IVFL	Lógica fuzzy valorada intervalarmente
T1FS	<i>Type-1 Fuzzy System</i>
T2FS	<i>Type-2 Fuzzy System</i>
SBRF1	Sistema Baseado em Regras Fuzzy Tipo 1
SBRF2	Sistema Baseado em Regras Fuzzy Tipo 2
BR	Base de Regras

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Objetivos	14
1.2	Estrutura do trabalho	14
2	COMPUTAÇÃO EM NUVEM	16
2.1	Definição	16
2.2	Características	17
2.3	Modelos de serviço da Nuvem	17
2.4	Desafios	18
2.4.1	Provisionamento automatizado de serviços	18
2.4.2	Migração de Máquina Virtual	19
2.4.3	Consolidação de servidores	19
2.4.4	Gerenciamento de Energia	19
2.5	Simuladores	19
2.5.1	CloudSim	21
2.6	Classificação do Gerenciamento de Recursos para Eficiência Energética	22
3	LÓGICA FUZZY	24
3.1	Lógica Clássica x Lógica Fuzzy	24
3.2	Fundamentos da Lógica Fuzzy	25
3.2.1	Conectivos da Lógica Fuzzy	25
3.2.2	Números Fuzzy	28
3.2.3	Implicações Fuzzy	30
3.3	Fundamentos da Lógica Fuzzy Valorada Intervalarmente	31
3.3.1	Conjuntos Fuzzy Tipo 2	31
3.3.2	Conectivos da Lógica Fuzzy Valorada Intervalarmente	33
3.3.3	Sistema Baseado em Regras Fuzzy Tipo 2	36
3.3.4	Ordens Admissíveis	37
4	F-CLOUD: VISÃO GERAL E MODELAGEM	38
4.1	Tratamento de Incertezas na CN via Lógica Fuzzy Intervalar	38
4.1.1	Modelagem do Sistema Fuzzy para Tratamento das Incertezas	38
4.1.2	Base de Dados e Definição das Funções de Pertinência	39
4.1.3	Fuzzificação	45
4.1.4	Base de regras	45
4.1.5	Inferência	47
4.1.6	Defuzzificação	47

4.2	Visão Geral dos Procedimentos Operacionais do fCloud	48
5	F-CLOUD: ANÁLISE DE DESEMPENHO	50
5.1	Configuração do ambiente	50
5.2	Resultados	52
5.2.1	Estudo de caso 1	53
5.2.2	Estudo de caso 2	56
6	CONCLUSÃO	58
	REFERÊNCIAS	59

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. Na CN *data centers* 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 (TOOSI; BUYYA, 2015). 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 (MELL; GRANCE et al., 2011).

De acordo com o relatório do Conselho de Defesa dos Recursos Naturais (NRDC)¹ dos Estados Unidos (EUA), em 2014, somente os seus *data centers* consumiram uma estimativa de 70 bilhões de quilowatts/hora (kWh), representando cerca de 1,8% do seu consumo total de eletricidade (SHEHABI et al., 2016).

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, etc) em que não há a necessidade do usuário fazer grande investimento em equipamentos, e ainda, com pagamento associado ao tempo de uso (NATHANI; CHAUDHARY; SOMANI, 2012).

O problema de minimização do consumo energético na CN considerando restrições referentes a qualidade de serviços (*Quality of Service* - QoS) é analiticamente complexo, e demanda investimentos na consolidação dinâmica de Máquinas Virtuais (MVs), que é um problema NP-Difícil (HILTUNEN et al., 2010; BELOGLAZOV; BUYYA, 2013).

Neste contexto, os principais desafios da CN são: (i) provisionamento automático de serviço; (ii) migração de MVs; (iii) consolidação de servidores, (iv) gerenciamento de energia, e ainda, (v) segurança de dados. Este trabalho contempla a migração e consolidação de MVs com uma abordagem para utilização eficiente dos servidores fí-

¹ <https://www.nrdc.org/>

sicos do ambiente da CN e buscando, simultaneamente manter um padrão satisfatório de *Qualidade de serviço* (QoS) e *Service Level Agreements* (SLA).

Na concepção da abordagem proposta, foi utilizado o formalismo matemático e a fundamentação da Lógica Fuzzy (LF). A LF, lógica nebulosa, ou ainda, lógica difusa, é uma lógica multivalorada capaz de absorver informações vagas, normalmente descritas em uma linguagem natural e convertê-las para um formato numérico de fácil manipulação computacional. Esta facilidade se mostra oportuna para o uso na etapa de consolidação de MVs, procurando com isso modelar os modos imprecisos do raciocínio na tomada de decisões a partir de informações obtidas através da infraestrutura computacional.

1.1 Objetivos

O objetivo deste trabalho é o desenvolvimento do *fCloud*: Um módulo para tomada de decisões empregando abordagem da Lógica Fuzzy Tipo-2 capaz de tratar a incerteza do especialista em determinar a utilização de uma máquina física no ambiente de CN, considerando desde sua modelagem até a prototipação e validação.

O trabalho considera como estratégia a integração da CN com a LF, objetivando filtrar as principais métricas disponíveis no ambiente e, assim propor a modelagem e consolidar a implementação do componente *f-Cloud*, incluindo a posterior prototipação, e considerando a análise de desempenho dos resultados gerados.

Considerando este objetivo geral, os objetivos específicos elencados são:

1. Estudo das fundamentações da CN;
2. Estudo das fundamentações da LF;
3. Estudo do Int-fGrid (MOURA et al., 2018) para fundamentação do f-Cloud;
4. Desenvolvimento e prototipação do componente *f-Cloud* estendendo o Int-fGrid (MOURA et al., 2018);
5. Integração do módulo *f-Cloud* com simulador de CN;
6. Validação do componente *f-Cloud*, comparando com algoritmos clássicos utilizados na CN.

1.2 Estrutura do trabalho

Este trabalho está estruturado da seguinte forma: no Capítulo 1, está a introdução e os objetivos. No Capítulo 2, tem-se a fundamentação da CN. No Capítulo 3 apresenta a fundamentação da LF. No Capítulo 4, são apresentados os principais aspectos para a concepção, modelagem e prototipação dos sistemas fuzzy propostos

para auxiliar na tomada de decisão. Logo a seguir, no Capítulo 5, são apresentados os resultados obtidos. E, por fim, no Capítulo 6 consta as principais conclusões.

2 COMPUTAÇÃO EM NUVEM

Com o rápido desenvolvimento das tecnologias de processamento e armazenamento juntamente com a busca por maximizar o consumo energético intensificam-se a pesquisa por novas alternativas frente aos paradigmas antigos de computação. Neste contexto, surge o modelo computacional denominado CN, no qual recursos são disponibilizados de acordo com o tempo de usabilidade. Sendo assim, tem-se a possibilidade de haver uma cobrança mais flexível, como por exemplo, um determinado servidor ser utilizado por um determinado número de horas sendo cobrado só por aquele tempo e não por uma mensalidade.

Apesar do atual destaque dado a pesquisa em CN, seu conceito não é novo e tão pouco suas tecnologias (HAMDAQA; TAHVILDARI, 2012). Em 1960, John McCarthy acreditava que um dia corporações conseguiriam vender recursos computacionais como “commodity” (HOLICK et al., 2011).

O surgimento da CN causou um grande impacto nas indústrias de tecnologia de informação, sendo elas Google, Amazon e Microsoft que constantemente batalham para terem um ambiente de CN mais potente, confiável e eficiente energeticamente.

O objetivo deste capítulo é apresentar um estudo sobre a CN, suas características, definições, desafios e simuladores.

2.1 Definição

A NIST (National Institute of Standards and Technology) em “The NIST Definition of Cloud Computing” (MELL; GRANCE et al., 2011) define a CN nos termos abaixo:

"A Computação em Nuvem 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 que podem ser rapidamente alocados e liberados com esforço mínimo de gerenciamento ou interação com o provedor de serviços."

Este modelo de CN é composto de cinco características essenciais, três modelos de serviço e quatro modelos de desenvolvimento.

2.2 Características

Em (MELL; GRANCE et al., 2011) os autores descrevem a CN como composta de cinco características essenciais, que são reapresentadas nessa seção.

- **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.
- **Amplio 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).
- **Agrupamento de recursos:** Os recursos de computação do provedor são agrupados para atender a vários consumidores usando um modelo de multilocação, com diferentes recursos físicos e virtuais atribuídos e reatribuídos 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:** O uso de recursos que podem ser monitorados, controlados e relatados, proporciona transparência tanto para o fornecedor quanto para consumidor do serviço utilizado.

2.3 Modelos de serviço da Nuvem

A CN em provê serviços sob demanda por meio da Internet (CHIEU et al., 2009). Para prover esses serviços a CN é dividida em três categorias as quais são apresentadas na Figura 1 e detalhadas logo a seguir.

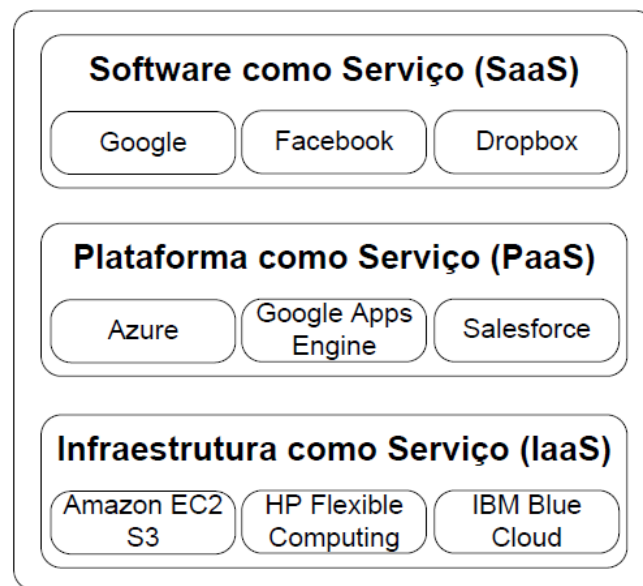


Figura 1 – Serviços oferecidos em CN (BATISTA, 2016)

- **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 API's fazendo com que o foco se concentre somente 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 MVs, fazendo com que o usuário não precise se preocupar com aspectos relacionados a parte física, como servidores e redes.

2.4 Desafios

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 (ZHANG; CHENG; BOUTABA, 2010).

2.4.1 Provisionamento automatizado de serviços

Consiste em alocar e realocar MV de acordo com a necessidade do cliente. As realocações devem ser feitas respeitando os *SLO (Service Level Objective)* do ambi-

ente 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.

2.4.2 Migração de Máquina Virtual

A virtualização trouxe diversas vantagens para a CN, como a possibilidade da criação de MVs com o intuito de balancear a carga por todo o *data center*, e ainda, possibilitando robustez e rápida resposta de provisionamento em *data centers*.

Os principais benefícios da migração de MVs 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.

2.4.3 Consolidação de servidores

A consolidação de servidores é uma maneira eficaz de maximizar a utilização de recursos enquanto minimizamos o consumo energético do ambiente de CN. A técnica de migrar MV em tempo real é constantemente utilizada para consolidar MVs 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 (FERDAUS et al., 2014) e amplamente conhecido.

2.4.4 Gerenciamento de Energia

Melhorar a eficiência energética é outra questão importante na CN. E empresas de TI estão sob constante pressão, 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.

2.5 Simuladores

Uma das desvantagens da CN está na sua modelagem de ambientes de testes. O usuário final, que espera uma experiência satisfatória do serviço, e no outro lado, as empresas, que 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 de ambientes de CN controlados. Há muitos benefícios em utilizarmos simuladores como bem descrito em (AHMED; SABYASACHI, 2014), com destaque nos apresentados na sequência:

- (i) Custo Mínimo: considere o desenvolvimento via software, gerando muito menor custo quando comparando com hardware.
- (ii) Repetível e controlável: podemos testar diversificando os testes por várias vezes até obtermos a saída desejável.
- (iii) Ambientes heterogêneos: fornece avaliação para cenário diversificados sob diferentes cargas de trabalho e medição de custos.

Dentre os simuladores citados em (AHMED; SABYASACHI, 2014), nossas escolhas 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. Os três simuladores e suas principais características são ressaltadas a baixo:

- GreenCloud, Proposto em (KLIASOVICH; BOUVRY; KHAN, 2012) e apresentando um simulador projetado para cálculo do consumo de energia em qualquer componente específico do *data center* como *link*, *switch*, *gateway*, etc. Porém, requerendo muita memória e tempo de simulação sendo, então recomendado somente para pequenos *data centers*;
- CloudAnalyst, Proposto em (WICKREMASINGHE; CALHEIROS; BUYYA, 2010) e apresentado um simulador projetado para avaliar a performance e o custo de grandes ambientes de CN distribuídos em diferentes locais e manipulando enorme carga de trabalho do usuário, tem uma interface gráfica que facilita a visualização dos dados através de gráficos
- CloudSim, Proposto em (CALHEIROS et al., 2011) e apresentado como simulador com a maior popularidade no contexto de uma comunidade ativa, possibilitando a modelagem e criação de grandes *data centers*, número ilimitado de MVs 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 acima o simulador escolhido para implementação foi o CloudSim (CALHEIROS et al., 2011) devido ao seu suporte a simulação energética, baixa utilização de memória possibilitando a simulação de grandes *data centers* e por já conter nativamente diversas políticas de alocação e escalonamento de MVs.

2.5.1 CloudSim

A Figura 2 mostra o projeto de várias camadas da estrutura do software CloudSim e seus componentes de arquitetura. O CloudSim suporta várias funcionalidades, como enfileiramento e processamento de eventos, criação de entidades do sistema da Nuvem, como ,serviços, host, *data center*, *broker*, MVs, incluindo ainda comunicação entre componentes e gerenciamento do relógio de simulação.

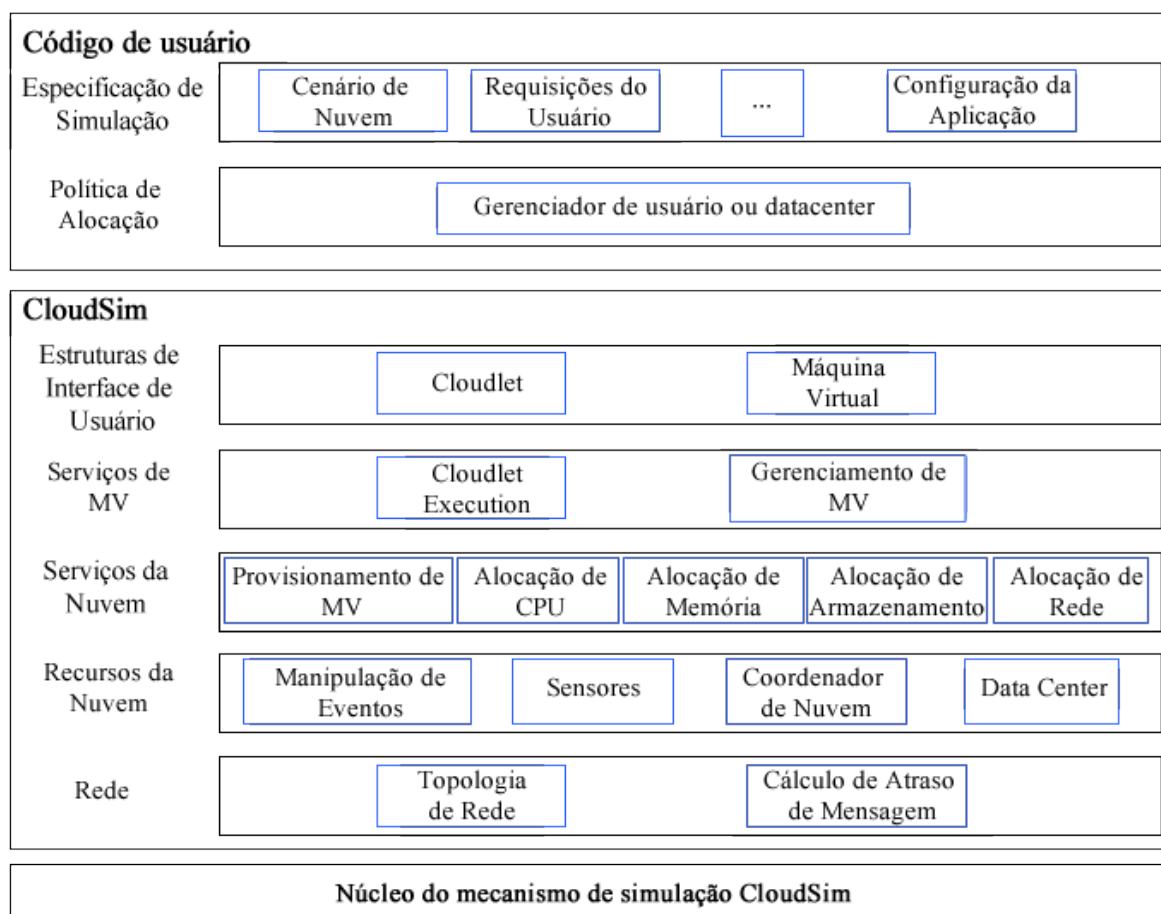


Figura 2 – Arquitetura de Camadas do CloudSim. Adaptado de (CALHEIROS et al., 2011)

A camada de simulação do CloudSim fornece suporte para modelagem e simulação de ambientes de *data center* virtualizados baseados em nuvem, incluindo interfaces de gerenciamento dedicadas para MVs, memória, armazenamento e largura de banda. Os problemas fundamentais, como provisionamento de hosts para MVs, gerenciamento de execução de aplicativos e monitoramento do estado dinâmico do sistema, também são tratados por essa camada.

Além destes recursos, esta camada também expõe as funcionalidades para o desenvolvimento de aplicativos, executando perfis de carga de trabalho complexos e estudos de desempenho de aplicativos. A camada mais superior na pilha do CloudSim

é o Código de usuário, disponibilizando entidades básicas para hosts como (número de máquinas, suas especificações e outras), suporte a aplicativos (número de tarefas e seus requisitos), MVs, número de usuários e seus tipos de aplicativo e políticas de escalonamento.

O CloudSim foi escolhido para emprego neste trabalho considerando sua ampla aceitação e uso pela comunidade científica. Além disto, fornece em sua estrutura as funcionalidades para o emprego na avaliação energética, da mesma forma proporciona facilidade no controle do ambiente simulado.

2.6 Classificação do Gerenciamento de Recursos para Eficiência Energética

Em (BELOGLAZOV, 2013), define-se uma taxonomia das técnicas para gerenciamento de ambiente de CN que busca a eficiência energética. Essa taxonomia é reapresentada na Figura 3. A taxonomia está dividida em duas principais classes na qual fazem o tratamento do consumo energético de duas formas distintas. (i) modo estático, e (ii) dinâmico.

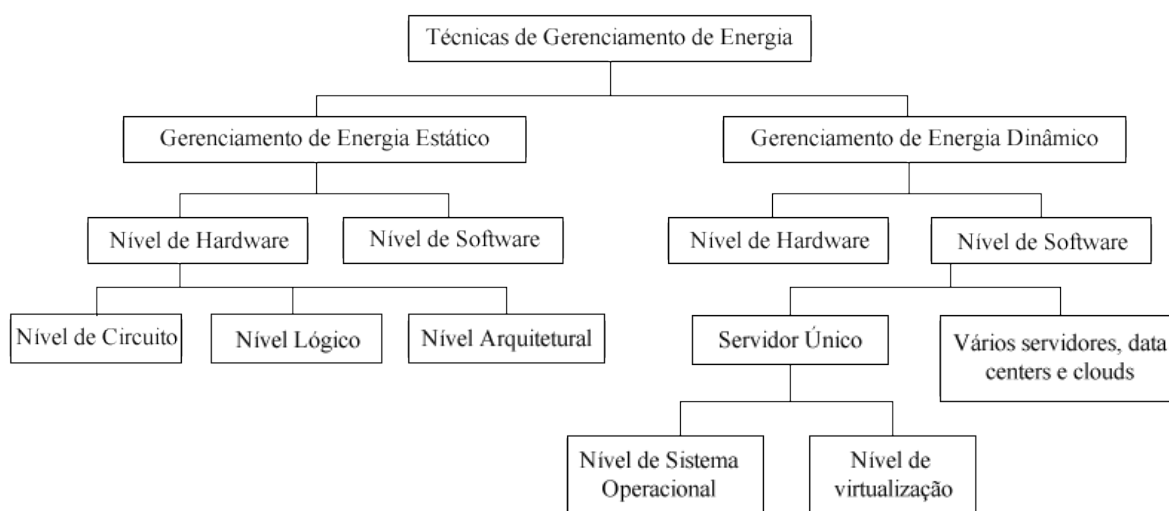


Figura 3 – Taxonomia do gerenciamento de recursos. Adaptado de (BELOGLAZOV, 2013)

As técnicas estáticas contêm todas aquelas otimizações que são implementadas a nível de hardware, sendo estas desenvolvidas no projeto dos circuitos, da parte lógica, arquitetônica ou sistêmica.

Aprimoramentos a nível de circuito tem como objetivo a redução da alteração do estado (ligado/desligado) de transistores e portas lógicas, essas otimizações são obtidas através do desenvolvimento de projetos complexos de desenho de portas, e incluindo

o redimensionamento dos transistores.

Otimizações a nível lógico tem como objetivo diminuir a redução da alteração do estado (ligado/desligado) dos circuitos sequenciais. Otimizações a nível arquitetônico incluem análise do desenho do sistema para posterior aplicação de técnicas de otimização energética. Porém, para essas otimizações a nível de hardware surtirem efeito é essencial considerar a implementação de programas específicos para serem executados, pois mesmo com otimizações a nível de hardware, um programa que não satisfaz as propriedades desse hardware pode trazer degradação de desempenho, e redução da eficiência energética.

Como apresentado na Seção 2.2, a CN tem como objetivo abstrair a manipulação física dos servidores aos usuários e dar liberdade sobre como utilizarão o ambiente, criando heterogeneidade do ambiente, tanto nas cargas de trabalho, quanto no tipo de operação (CPU, RAM, I/O).

Para obter eficiência energética, é necessário o desenvolvimento de técnicas que se adaptam às necessidades e aos comportamentos diferentes do ambiente, são desenvolvidas técnicas que tratam o ambiente de modo dinâmico, adaptando-se de acordo com o comportamento do ambiente, estas técnicas, são ainda divididas nas seguintes sub-categorias: (i) dinâmicas em nível de hardware, e (ii) software.

Técnicas dinâmicas a nível de hardware normalmente podem ser classificadas em técnicas de escalonamento dinâmico de desempenho, variação de tensão e escala de frequência, ou ainda, desligamento parcial ou completo de componentes em determinados períodos. Já técnicas a nível de software possibilitam ao administrador do ambiente optar por diferentes políticas de acordo com a necessidade.

A camada de software é dividida em outras duas, que são as camadas de Sistema Operacional (SO) e Virtualização. Técnicas dinâmicas que operam no nível de Sistema Operacional tem como intuito fazer com que a camada de software auxilie na obtenção de determinados objetivos, podendo eles serem: (i) adaptar a aplicação ao ambiente, (ii) otimizar a utilização dos recursos do sistema, (iii) aplicar técnicas de economia de energia, e (iv) manipular as cargas de trabalho.

A outra camada é a de virtualização, que acrescenta uma camada de abstração entre o sistema operacional e o hardware, possibilitando a criação de MVs com o intuito de otimizar a utilização dos recursos.

Finalmente, tem-se a camada de múltiplos servidores e *data centers* foco da pesquisa neste trabalho. Técnicas nesse nível tem por objetivo usufruir da heterogeneidade presente nesses ambientes, estendendo as práticas de um único servidor para diversos servidores. Normalmente, tem-se um gerenciador global e algumas operações sendo realizadas internamente nas máquinas para evitar sobrecarga do gerenciador global.

3 LÓGICA FUZZY

A LF está fundamentada na Teoria dos Conjuntos Fuzzy (TCF) introduzida por Zadeh (ZADEH, 1965, 2008), estendendo os conceitos da Lógica Clássica. Diferentemente da rigidez Booleana na avaliação de argumentos, a LF promove a flexibilidade da relação de pertinência, ao considerar graus intermediários de pertinência no intervalo unitário $[0,1]$, fornecendo técnicas poderosas de modelagem para o tratamento de informações vagas ou imprecisas.

Esta seção apresenta a fundamentação teórica a partir de conceitos da LF, fazendo correlação entre LF e lógica clássica. Reporta-se também uma extensão dos conjuntos fuzzy, os conjuntos fuzzy valorados intervalarmente.

3.1 Lógica Clássica x Lógica Fuzzy

Na Lógica Clássica só é possível assumir dois valores para a pertinência de um elemento a um conjunto: verdadeiro (1) e falso (0). Entretanto, na abordagem fuzzy pode-se assumir valores num intervalo de 0 a 1, flexibilizando a modelagem e, também, a aplicação dos valores podem ser expressos em termos linguísticos, aproximando-os da linguagem natural. Alguns exemplos são as expressões: alto, baixo, médio, muito alto, muito baixo, pouco alto, pouco baixo, etc. Esse tipo de modelagem, em processos mais complexos, permite resultados mais acurados, melhorando o desempenho quando da tomada de decisão (*Multiple criteria decision making* - MCDM).

Na TCF, todo conjunto está identificado por sua função de pertinência. Assim cada elemento x de um conjunto universo $\chi \neq \emptyset$ está associado a um conjunto fuzzy A definido pela função de pertinência $\mu_A : \chi \rightarrow [0, 1]$. Neste contexto, um conjunto fuzzy tipo-1 (T1FS) A é dado pela expressão

$$A = \{(x, \mu_A(x)) : x \in \chi\}. \quad (1)$$

Em outras palavras, um conjunto universo que representa todos os alunos de uma sala, e um subconjunto desse universo, que seria os que são bons em matemática.

Todos os elementos do conjunto universo teriam um grau de pertinência, referente

a esse subconjunto, que significaria o quanto cada elemento é bom em matemática, em sendo então valorado no um intervalo de 0 a 1. Por exemplo, se um elemento tivesse um grau de pertinência 0, ele não seria bom em matemática e se tivesse 1 ele seria muito bom em matemática, e se tivesse um 0,8 ele seria somente bom em matemática. Essa forma de modelar é diferente do modelo clássico, onde só poderia ser classificados em bons em matemática (1) e ruins em matemática(0).

3.2 Fundamentos da Lógica Fuzzy

A Lógica Fuzzy está baseada na Teoria dos Conjuntos Fuzzy, incorporando o conceito de pertinência, onde os elementos tem um grau de pertinência a um conjunto fuzzy.

Formalmente um conjunto fuzzy A é caracterizado por uma função de pertinência $\mu_A : \chi \rightarrow [0, 1]$, a qual associa a cada elemento de um universo $\chi \neq \emptyset$, um número real no intervalo unitário $[0, 1]$:

$$A = \{(x, \mu_A(x)) \mid x \in \chi \text{ e } \mu_A(x) \in [0, 1]\}, \quad (2)$$

e, para $x \in \mathbf{U}$, $\mu_A(x)$ representa o grau de pertinência de x em A .

3.2.1 Conectivos da Lógica Fuzzy

Definição 1 A função $N : [0, 1] \rightarrow [0, 1]$ é uma negação fuzzy se verifica as seguintes propriedades:

$$N_1 : N(0) = 1 \text{ e } N(1) = 0;$$

$$N_2 : \text{Se } x \geq y \text{ então } N(x) \leq N(y), \forall x, y \in [0, 1].$$

Uma negação fuzzy que também satisfaz a propriedade involutiva

$$N_3 : N(N(x)) = x, \forall x \in \mathbf{U}.$$

é chamada de negação fuzzy forte.

Apresentam-se a seguir as principais definições de negações fuzzy:

1. **Negação Complementar ou Negação Padrão**, negação forte dada por

$$N_C(x) = 1 - x. \quad (3)$$

2. **Negações de Gödel**, as quais são negações fuzzy, mas não são fortes:

$$N_{D1}(x) = \begin{cases} 0, & \text{se } x > 0 \\ 1, & \text{se } x = 0; \end{cases} \quad N_{D2}(x) = \begin{cases} 1, & \text{se } x < 1 \\ 0, & \text{se } x = 1; \end{cases} \quad (4)$$

Uma negação fuzzy é utilizada geralmente para representar o operador de complemento entre conjuntos fuzzy:

$$A_C = \{(x, \mu_{A_C}) : x \in \mathbf{U} \text{ e } \mu_{A_C}(x) = N(\mu_A(x))\}. \quad (5)$$

Definição 2 *Uma norma triangular, conhecida como t-norma, é uma operação binária $T : [0, 1]^2 \rightarrow [0, 1]$ tal que, $\forall x, y, z \in [0, 1]$, as seguintes propriedades são verificadas:*

T_1 *Comutatividade:* $T(x, y) = T(y, x)$;

T_2 *Associatividade:* $T(x(T(y, z))) = T(T(x, y), z)$;

T_3 *Elemento Neutro:* $T(x, 1) = x$;

T_4 *Monotonicidade:* $T(x, y) \leq T(x, z)$ se $y \leq z$.

Uma t-norma é utilizada geralmente para representar o operador de conjunção (e) ou a intersecção entre conjuntos fuzzy:

$$A \cap B = \{(x, \mu_{A \cap B}) : x \in \mathbf{U} \text{ e } \mu_{A \cap B}(x) = T(\mu_A(x), \mu_B(x))\}. \quad (6)$$

Tabela 1 – Exemplos básicos de t-normas

Designação	Representação
T_M : Mínimo	$T_M = \min(x, y)$
T_P : Produto	$T_P = xy$
T_L : Lukasiewicz	$T_L = \max(x + y - 1, 0)$
T_D : Drástica do produto	$T_D = (x, y) = \begin{cases} 0, & \text{se } x, y \in [0, 1[; \\ \min(x, y), & \text{c.c.} \end{cases}$
T_{nM} : Nilpotente mínimo	$T_{nM}(x, y) = \begin{cases} 0, & \text{se } x + y \leq 1; \\ \min(x, y), & \text{c.c.} \end{cases}$

A Tabela 1 apresenta exemplos de t-normas.

Definição 3 *Seja o intervalo unitário real $[0, 1] \subseteq \mathbf{R}$. Uma conorma triangular, indicada por t-conorma, é uma operação binária $S : [0, 1]^2 \rightarrow [0, 1]$ que satisfaz as seguintes propriedades:*

- S_1 *Comutatividade:* $S(x, y) = S(y, x)$;
- S_2 *Associatividade:* $S(x(S(y, z))) = S(S(x, y), z)$;
- S_3 *Elemento Neutro:* $S(x, 0) = x$;
- S_4 *Monotonicidade:* $S(x, y) \leq S(x, z)$ se $y \leq z$.

Uma t-conorma é utilizada geralmente para representar o operador de disjunção (ou) ou a união entre conjuntos fuzzy:

$$A \cup B = \{(x, \mu_{A \cup B}) : x \in \mathbf{U} \text{ e } \mu_{A \cup B}(x) = S(\mu_A(x), \mu_B(x))\}. \quad (7)$$

Este trabalho também considera o estudo do operador OWA (YAGER, 2004a), provendo um meio de agregar valores associados à satisfação de múltiplos critérios, e, ainda, unificando ambos os comportamentos de elementos em conjunto fuzzy, o conjuntivo e o disjuntivo. Operadores OWA são funções de agregação comutativos, idempotentes e apresentam um comportamento compensatório. (YAGER, 2004b)

A agregação obtida pela aplicação de um operador OWA está entre os valores máximo e mínimo. Esse operador generaliza o mínimo e o máximo, podendo ser visto como uma função parametrizada variando no intervalo unitário $[0, 1]$.

Definição 4 *Seja $\sigma : N_n \rightarrow N_n$ uma permutação consistindo na ordenação não decrescente, ou seja, $0 \leq x_{\sigma(1)} \leq x_{\sigma(2)} \leq \dots \leq x_{\sigma(n)} \leq 1$, $0 \leq w_i \leq 1$ e $\sum_{i=1}^n w_i = 1$. Um operador OWA: $[0, 1]^n \rightarrow [0, 1]$ é definido pela expressão:*

$$OWA(x_1, x_2, \dots, x_n) = \sum_{j=1}^n w_j x_{\sigma(j)} \quad (8)$$

A partir de um operador OWA, tem-se uma família parametrizada de operadores de agregação incluindo o máximo, o mínimo, a mediana e a média aritmética, os quais são gerados por escolhas de parâmetros.

3.2.2 Números Fuzzy

Assim como no caso clássico, também temos o objetivo de fazer algebras a partir da estrutura lógica das T2FS. A diferença é que se pretende calcular quantidades imprecisas. Por exemplo, é opinião unânime dizer que o dobro de uma quantidade ‘em torno de 15’ resulta em outra ‘em torno de 30’. Para isto, construíram-se objetos matemáticos que generalizam os números reais. Tais objetos serão chamados de números fuzzy (GEORGE J; BO, 2008).

As principais definições e propriedades de números são apresentadas logo a seguir.

Definição 5 Um conjunto fuzzy F é chamado número fuzzy quando o conjunto universo, onde F está definido, é o conjunto dos números reais \mathbb{R} e a função de pertinência $\mu_F : \mathbb{R} \rightarrow [0, 1]$, é tal que:

1. $\mu_F(x)$ atinge o 1, isto é, $\sup_{x \in \mathbb{R}} \mu_F(x) = 1$.
2. $[F]^\alpha$ é um intervalo fechado, $\forall \alpha \in (0, 1]$.
3. O suporte de F é limitado.

Os números fuzzy mais comuns são triangulares, trapezoidais e senoidais.

Definição 6 Um número fuzzy F é dito triangular se sua função de pertinência é da forma apresentada na Eq. 9.

$$\mu_F(x) = \begin{cases} 0, & \text{se } x \leq a, \\ \frac{x-a}{m-a}, & \text{se } a < x \leq m, \\ \frac{x-b}{m-b}, & \text{se } m < x \leq b, \\ 0, & \text{caso contrário.} \end{cases} \quad (9)$$

O gráfico da função de pertinência de um número fuzzy triangular tem a forma de um triângulo, tendo como base o intervalo $[a, b]$ e, como único vértice fora desta base, o ponto $(m, 1)$. Deste modo, os números reais a , m e b definem o número fuzzy triangular F , graficamente representado na Figura 4.

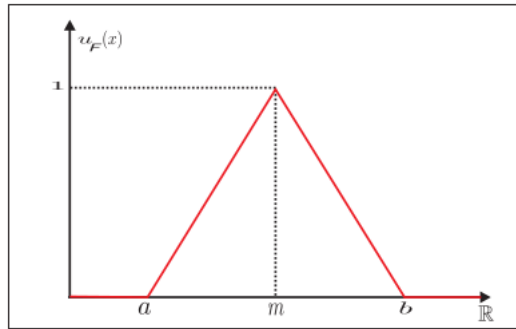


Figura 4 – Número fuzzy triangular

Definição 7 Um número fuzzy F é dito trapezoidal se sua função de pertinência tem a forma de um trapézio, com expressão algébrica dada na Eq.10 e graficamente apresentada na Figura 5.

$$\mu_F(x) = \begin{cases} \frac{x-a}{b-a}, & \text{se } a \leq x < b, \\ 1, & \text{se } b \leq x \leq c, \\ \frac{d-x}{d-c}, & \text{se } c < x \leq d, \\ 0, & \text{caso contrário.} \end{cases} \quad (10)$$

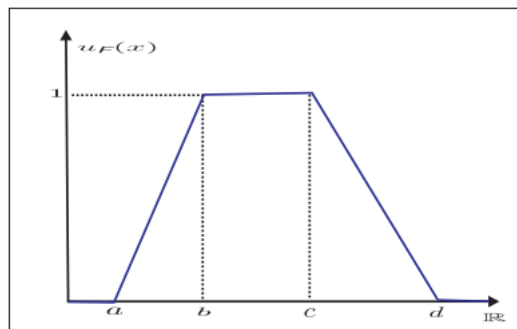


Figura 5 – Número fuzzy trapezoidal

Definição 8 Um número fuzzy tem forma senoidal se a função de pertinência for suave e simétrica em relação ao número real. A seguinte expressão analítica da função de pertinência da Eq.11 verifica as propriedades da Definição 8 para n , a e $\delta \in \mathbb{R}$ dados. Veja a correspondente representação gráfica na Figura 6.

$$\mu_F(x) = \begin{cases} \exp\left(-\left(\frac{x-n}{a}\right)^2\right), & \text{se } n - \delta \leq x < n + \delta, \\ 0, & \text{caso contrário.} \end{cases} \quad (11)$$

Definição 9 Seja F um conjunto fuzzy. O Kernel de F é o conjunto de todos os elementos $x \in U$ cujo grau de pertinência é 1, isto é, $Ker(F) = \{x \in U : \mu_F(x) = 1\}$.

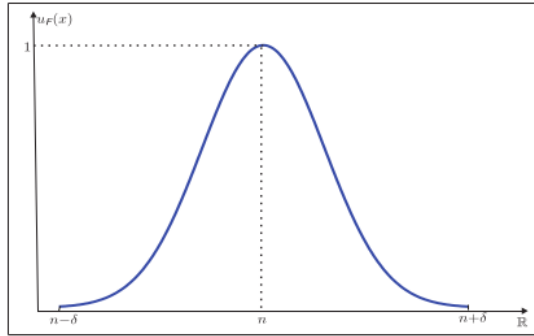


Figura 6 – Número fuzzy senoidal

Definição 10 Um elemento $x \in U$, no qual $\mu_F(x) = 0.5$ é denominado, ponto crossover.

Definição 11 Um conjunto F cujo suporte é um único ponto em U , com $\mu_F(x) = 1$, é denominado, singleton fuzzy.

Definição 12 A altura de um conjunto fuzzy F é o valor máximo da pertinência de x em U , isto é, $hgt(F) = \sup_{x \in U} \mu_F(x)$.

A Figura 7 apresenta a identificação da altura, kernel e suporte de um conjunto fuzzy.

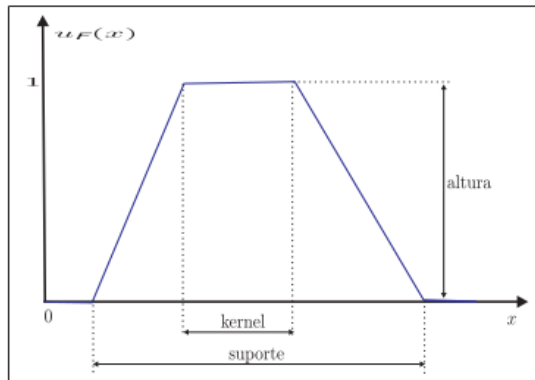


Figura 7 – Altura, Kernel e suporte de um conjunto fuzzy

3.2.3 Implicações Fuzzy

A literatura apresenta uma diversidade de definições e aplicações no assunto de implicações fuzzy e suas duais (BACZYŃSKI, 2004; BACZYŃSKI; JAYARAM, 2008; BALASUBRAMANIAM, 2007). Sendo a implicação fuzzy uma generalização da abordagem clássica, o único consenso em suas definições é que a implicação fuzzy deve ter o mesmo comportamento da implicação clássica quando os valores de entrada da implicação forem uma combinação dos extremos do intervalo $[0, 1]$. As implicações estão contidas no conjunto das subimplicações fuzzy, as quais são operadores que não verificam totalmente o comportamento da implicação clássica.

Um operador binário $I : U^2 \rightarrow U$ é uma **implicação fuzzy** se I satisfaz as condições de contorno, (I2 e I3) :

I0: $I(1, 1) = I(0, 1) = I(0, 0) = 1$.

I1: $I(1, 0) = 0$.

I2: Se $x \leq z$ então $I(x, y) \geq I(z, y)$ (antitonicidade no primeiro argumento);

I3: Se $y \leq z$ então $I(x, y) \leq I(x, z)$ (isotonicidade no segundo argumento);

3.3 Fundamentos da Lógica Fuzzy Valorada Intervalarmente

Nesta sessão apresentam-se os conceitos básicos da lógica fuzzy valorada intervalarmente (IVFL), na qual são atribuídos não um número mas um outro conjunto do tipo intervalo como grau de pertinência associado a um elemento do universo.

Um conjunto fuzzy valorado intervalarmente (IVFS) é um caso particular de (T2FS).

3.3.1 Conjuntos Fuzzy Tipo 2

A teoria dos conjuntos fuzzy tipo 2 (T2FS) foi introduzida por (ZADEH, 1975) como uma extensão do conjunto fuzzy tradicional. T2FS são conjuntos fuzzy, com grau de pertinência em um T1FS.

A T2FS, modela a incerteza na definição do grau de pertinência. No caso especial dos conjuntos fuzzy da IVFL, a incerteza entre dois valores fuzzy 0.3 e 0.4 resulta em atribuir $[0.3, 0.4]$ como grau de pertinência. As inferências baseadas em T2FL permitem que se aplique o Sistema Baseado em Regras Fuzzy Tipo 2.

Um elemento do tipo T2FS é caracterizado por uma função de pertinência em T2FS, onde seu grau de pertinência passa a ser um conjunto de pares ordenados constituídos por T1FSs.

Definição 13 Um T2FS \tilde{A} , é caracterizado por uma função de pertinência $\mu_{\tilde{A}}(x, u)$, onde $x \in X$ e $u \in J_x \subseteq [0, 1]$, ou seja:

$$\tilde{A} = \{((x, u), \mu_{\tilde{A}}(x, u)) : \forall x \in X, \forall u \in J_x \subseteq [0, 1]\}, \quad (12)$$

onde, $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$.

Definição 14 T2FS se reduzem a IVFS quando $\mu_{\tilde{A}}(x, u) = 1$.

Um meio de representar T2FS é através da forma geométrica de sua função de pertinência. As Figuras 8(a), e 8(b) mostram IVFS e suas respectivas áreas coloridas em azul correspondem a região denominada "Foot print of Uncertainty" (FOU).

O grau secundário dos T2FS é sempre igual a 1. Portanto, a terceira dimensão acaba não mostrando nenhuma informação adicional e um T2FS pode sempre ser re-

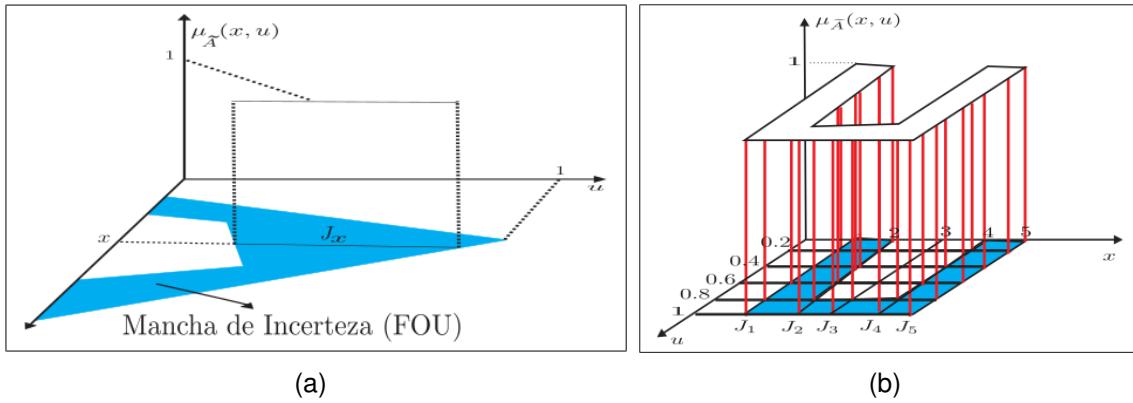


Figura 8 – Conjuntos Fuzzy Tipo 2 Intervalar.

apresentado apenas por sua região de FOU. As Figuras 9(a), 9(b), 9(c) e 9(d), apresentam várias formas de T2FS: intervalar, triangular, trapezoidal, gaussiano e “singlenton”, respectivamente.

Definição 15 O corte vertical de $\mu_{\tilde{A}}(x, u)$ é definido como sendo o plano bidimensional para plano $x = x'$, cujos eixos são u e $\mu_{\tilde{A}}(x', u)$.

Definição 16 A função de pertinência secundária é o corte vertical de $\mu_{\tilde{A}}(x, u)$ em determinado valor de $x = x'$. Como mostra, a Figura 10.

Definição 17 A pertinência primária J_x de x , é definida como o domínio da função de pertinência secundária para o valor de x , com $J_x = [\underline{J}_x, \overline{J}_x] \subseteq [0, 1], \forall x \in X$.

Definição 18 A “mancha” de incerteza (FOU) é definida como a união de todas as pertinências primárias, isto é,

$$FOU(\tilde{A}) = \bigcup_{x \in X} (x, j_x) \quad (13)$$

a FOU é um conjunto fuzzy do tipo 2 delimitada por uma função de pertinência do tipo 1 superior e inferior (RIZOL; MESQUITA; SAOTOME, 2011).

Definição 19 A função de pertinência superior é representada na forma $\overline{\mu}_{\tilde{A}}(x), \forall (x) \in X$, e dada pela expressão abaixo:

$$\overline{\mu}_{\tilde{A}}(x) = \bigcup_{x \in X} (x, \overline{J}_x), \quad (14)$$

logo, a função de pertinência inferior é representada na forma $\underline{\mu}_{\tilde{A}}(x), \forall (x) \in X$, dados por:

$$\underline{\mu}_{\tilde{A}}(x) = \bigcup_{x \in X} (x, \underline{J}_x) \quad (15)$$

onde \bigcup representa o operador de união.

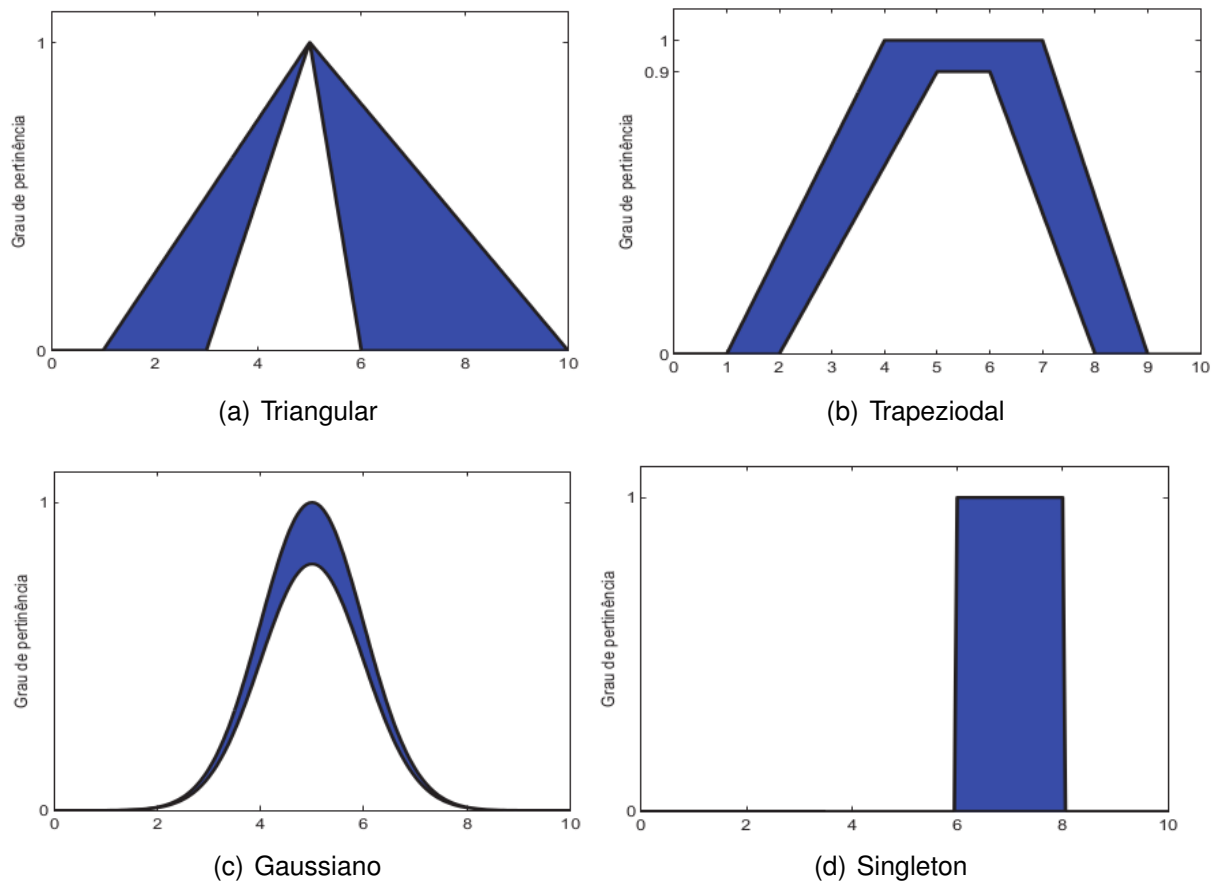


Figura 9 – Exemplos de conjuntos fuzzy do tipo 2 intervalares.

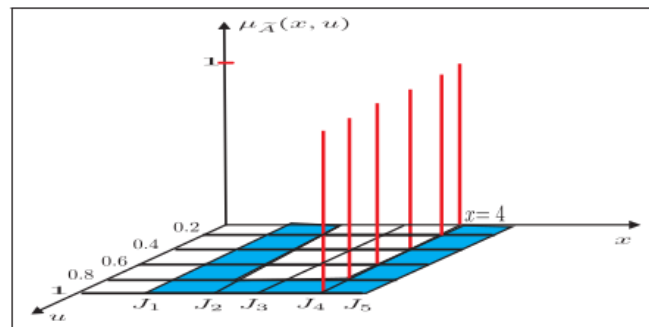


Figura 10 – Função de pertinência secundária intervalar em $x=4$.

A Figura 11 exemplifica uma FOU e funções de pertinência superior e inferior.

3.3.2 Conectivos da Lógica Fuzzy Valorada Intervalarmente

Seja $\mathbb{U} = \{[a, b] \mid 0 \leq a \leq b \leq 1\}$ o conjunto dos intervalos de reais entre 0 e 1, ou seja, dos valores fuzzy intervalares. Considerando-se que $X \in \mathbb{U}$ então, a partir das funções de projeção, $X = [\underline{X}, \bar{X}]$ representa o grau de pertinência de um elemento num conjunto fuzzy valorado intervalarmente.

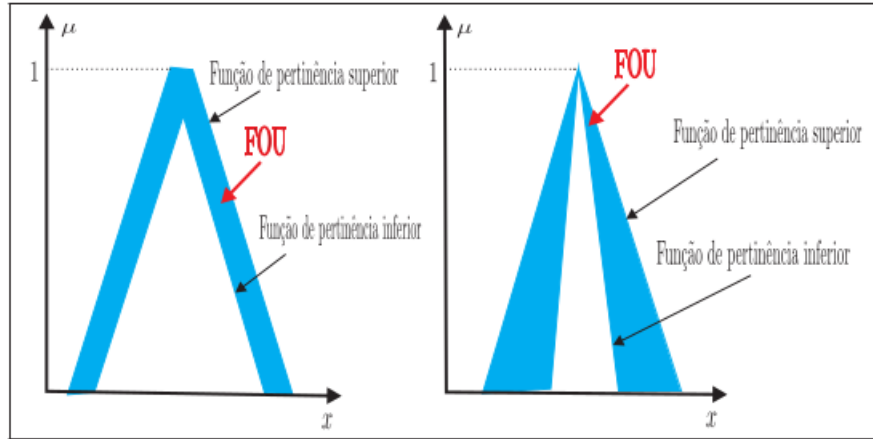


Figura 11 – Conjunto fuzzy do tipo 2.

Definição 20 Uma função intervalar $\mathbb{N} : \mathbb{U} \rightarrow \mathbb{U}$ é uma **negação fuzzy valorada intervalarmente** se, para todo $X, Y \in \mathbb{U}$, as seguintes condições são satisfeitas:

$$\mathbb{N}1: \mathbb{N}([0, 0]) = [1, 1] \text{ e } \mathbb{N}([1, 1]) = [0, 0];$$

$$\mathbb{N}2a: \text{Se } X \geq Y \text{ então } \mathbb{N}(X) \leq \mathbb{N}(Y);$$

$$\mathbb{N}2b: \text{Se } X \subseteq Y \text{ então } \mathbb{N}(X) \supseteq \mathbb{N}(Y).$$

Se \mathbb{N} também satisfaz a propriedade involutiva,

$$\mathbb{N}3: \mathbb{N}(\mathbb{N}(X)) = X.$$

\mathbb{N} é uma **negação fuzzy valorada intervalarmente forte**.

Seja $N : U \rightarrow U$ uma negação fuzzy. Então \mathbb{N} é negação fuzzy intervalar representável se \mathbb{N} é dada pela expressão:

$$\mathbb{N}(X) = [N(\overline{X}), N(\underline{X})]. \quad (16)$$

Pela Eq.(17), a extensão intervalar da negação padrão é dada por:

$$\mathbb{N}_S(X) = [1 - \overline{X}, 1 - \underline{X}]. \quad (17)$$

Definição 21 Uma norma triangular valorada intervalarmente, conhecida como *t-norma intervalar*, é uma operação binária $\mathbb{T} : [0, 1]^2 \rightarrow [0, 1]$, que satisfaz as seguintes propriedades:

$$\mathbb{T}_1 \text{ Comutatividade: } \mathbb{T}(X, Y) = \mathbb{T}(Y, X)$$

$$\mathbb{T}_2 \text{ Associatividade: } \mathbb{T}(X(\mathbb{T}(Y, Z))) = \mathbb{T}(\mathbb{T}(X, Y), Z)$$

$$\mathbb{T}_3 \text{ Elemento Neutro: } \mathbb{T}(X, [1, 1]) = X$$

\mathbb{T}_4 *Monotonicidade em relação às ordens:*

A) *Produto:* $\mathbb{T}(X, Y) \leq \mathbb{T}(Z, W)$ se $X \leq Z$ e $Y \leq W$

B) *Inclusão:* $\mathbb{T}(X, Y) \subseteq \mathbb{T}(Z, W)$ se $X \subseteq Z$ e $Y \subseteq W$

Se $T : [0, 1]^2 \rightarrow [0, 1]$ é uma t-norma, então $\mathbb{T} : [0, 1]^2 \rightarrow [0, 1]$ definida por

$$\mathbb{T}(X, Y) = [T(\underline{X}, \underline{Y}), T(\overline{X}, \overline{Y})] \quad (18)$$

é uma t-norma intervalar, denominada de t-norma intervalar representável por T .

Definição 22 *Uma conorma triangular intervalar, conhecida como t-conorma intervalar, é uma operação binária $\mathbb{S} : [0, 1]^2 \rightarrow [0, 1]$, que satisfaz as seguintes propriedades:*

\mathbb{S}_1 *Comutatividade:* $\mathbb{S}(X, Y) = \mathbb{S}(Y, X)$

\mathbb{S}_2 *Associatividade:* $\mathbb{S}(X((Y, Z))) = \mathbb{S}(X, Y), Z)$

\mathbb{S}_3 *Elemento Neutro:* $\mathbb{S}(X, [0, 0]) = X$

\mathbb{S}_4 *Monotonicidade em relação à ordem:*

A) *Produto:* $\mathbb{S}(X, Y) \leq \mathbb{S}(Z, W)$ se $X \leq Z$ e $Y \leq W$

B) *Inclusão:* $\mathbb{S}(X, Y) \subseteq \mathbb{S}(Z, W)$ se $X \subseteq Z$ e $Y \subseteq W$

Se $S : [0, 1]^2 \rightarrow [0, 1]$ é uma t-conorma, então $\mathbb{S} : [0, 1]^2 \rightarrow [0, 1]$ definida por

$$\mathbb{S}(X, Y) = [S(\underline{X}, \underline{Y}), S(\overline{X}, \overline{Y})] \quad (19)$$

é uma t-conorma intervalar, denominada de t-conorma intervalar representável por T .

Tabela 2 – Exemplos de t-normas intervalares

Designação	Representação
\mathbb{T}_M : t-norma do Mínimo	$\mathbb{T}_M(X, Y) = \min(X, Y)$
\mathbb{T}_P : t-norma Produto	$\mathbb{T}_P(X, Y) = X \cdot Y$
\mathbb{T}_L : t-norma de Lukasiewicz	$\mathbb{T}_L(X, Y) = \max(X + Y - [1, 1], [0, 0])$

Tabela 3 – Exemplos básicos de t-conormas intervalares

Designação	Representação
\mathbb{S}_M : t-conorma do Máximo	$\mathbb{S}_M(X, Y) = \max(X, Y)$
\mathbb{S}_P : t-conorma Soma Produto	$\mathbb{S}_P(X, Y) = X + Y - X \cdot Y$
\mathbb{S}_L : t-conorma de Lukasiewicz	$\mathbb{S}_L(X, Y) = \min(X + Y, [1, 1])$

Outros conectivos da Lógica Fuzzy Valorada Intervalarmente podem ser também definidos de forma análoga.

3.3.3 Sistema Baseado em Regras Fuzzy Tipo 2

O Sistema Baseado em Regras Fuzzy Tipo 2 (SBRF2) é utilizado em aplicações onde existe incerteza na determinação exata do grau de pertinência e em aplicações onde não existe alta confiança no modelo (RIZOL; MESQUITA; SAOTOME, 2011). O diagrama de blocos do SBRF2 é composto por cinco componentes: (i) fuzzificador, (ii) inferência, (iii) base de regras, (iv) redutor de tipo e (v) defuzzificador. Este sistema é composto por, no mínimo, um conjunto fuzzy do tipo 2 presente em um dos antecedentes ou consequentes que compõem uma das regras que formam o sistema. A descrição de cada bloco do SBRF2 é apresentada na Figura 12.

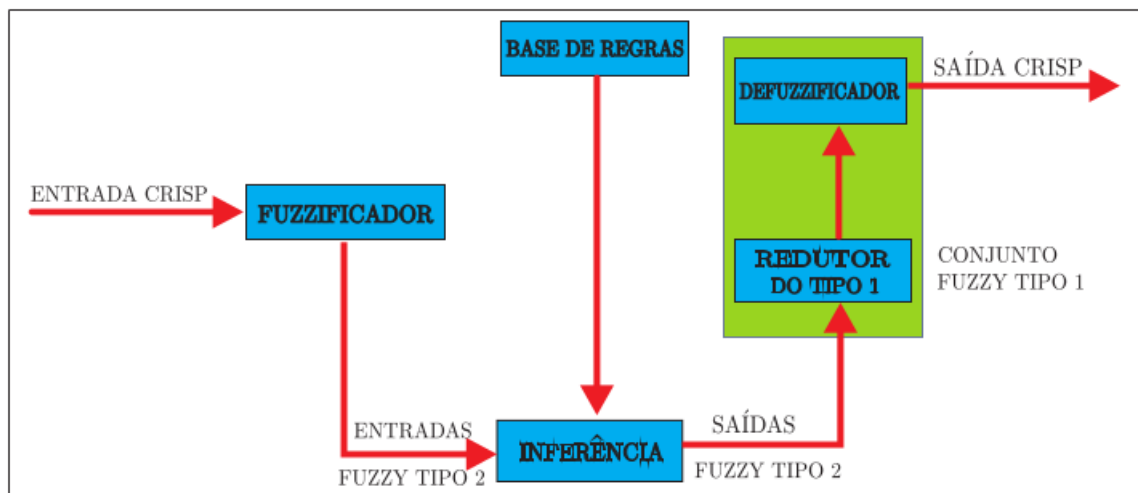


Figura 12 – Sistema Baseado em Regras Fuzzy 2.

- 1 **Fuzzificador:** o bloco fuzzificador transforma o vetor de entrada $x' = (x'_1, x'_2, \dots, x'_I), i = 1, 2, \dots, I$, do SBRF2 em conjuntos fuzzy do tipo 2;
- 2 **Base de Regras:** a base de regras do SBRF2 permanece a mesma forma do tipo 1. A diferença entre SBRF1 e SBRF2 está na natureza das funções de pertinência;
 - Regra 1:** Se $((x_1 \text{ é } \tilde{X}_1^1) \text{ e } (x_2 \text{ é } \tilde{X}_2^1))$ então $y \text{ é } Y^1$;
 - Regra 2:** Se $((x_1 \text{ é } \tilde{X}_1^2) \text{ e } (x_2 \text{ é } \tilde{X}_2^2))$ então $y \text{ é } Y^2$;
- 3 **Inferência:** o bloco de inferência realiza o cálculo do SBRF2 com base nas regras fuzzy;
- 4 **Redutor do Tipo 1:** o bloco redutor do tipo 1 tem função de transformar T2FS em T1FS, ou seja, procura o melhor T1FS que representa o T2FS e que deve satisfazer a seguinte premissa: quando toda a incerteza desaparecer, o resultado do SBRF2 é reduzido em um SBRF1 (MENDEL, 2001);

5 Defuzzificador: a saída defuzzificada do SBRF2 é dada pela média dos pontos limites \underline{Y} e \overline{Y} , ou seja;

$$Y_x = \frac{\underline{Y} + \overline{Y}}{2}, \forall x \in \chi, \quad (20)$$

Os valores \underline{Y} e \overline{Y} , podem ser calculados utilizando o método iterativo de Karnik e Mendel (algoritmo KM) (RIZOL; MESQUITA; SAOTOME, 2011). Da mesma forma, o passo de defuzzificação pode ainda ser obtido através da utilização de um método convencional, tal como o centróide para obter o valor final da inferência.

3.3.4 Ordens Admissíveis

Além da abordagem de SBRF2 também considera-se o uso de uma ordem total para intervalos por meio das ordens admissíveis explicadas em (BUSTINCE et al., 2013; ZAPATA et al., 2017), pois, por vezes, nem sempre uma lista de intervalos gerada com a saída do SBRF2 pode ser comparável através dos métodos convencionais. A seguir, as ordens admissíveis consideradas neste trabalho são relatadas:

Definição 23 *Ordem de Xu e Yager (XU; YAGER, 2006): $\forall (X, Y) \in \mathbb{U}$,*

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

Definição 24 *As Ordens Lexicográficas, \leq_{Lex1} relacionada à primeira variável, e \leq_{Lex2} correspondente a segunda variável, que são respectivamente definidas da seguinte forma:*

$$\begin{aligned} [\underline{X}, \overline{X}] \leq_{Lex1} [\underline{Y}, \overline{Y}] &\Leftrightarrow \begin{cases} \underline{X} < \underline{Y}; \text{ ou} \\ \underline{X} = \underline{Y} \text{ e } \overline{X} \leq \overline{Y}; \end{cases} \\ [\underline{X}, \overline{X}] \leq_{Lex2} [\underline{Y}, \overline{Y}] &\Leftrightarrow \begin{cases} \overline{X} < \overline{Y}; \text{ ou} \\ \overline{X} = \overline{Y} \text{ e } \underline{X} \leq \underline{Y}. \end{cases} \end{aligned}$$

4 F-CLOUD: VISÃO GERAL E MODELAGEM

Este capítulo apresenta a concepção do sistema fuzzy para o fCloud bem como a proposta metodológica para sua aplicação em um simulador de CN.

4.1 Tratamento de Incertezas na CN via Lógica Fuzzy Intervalar

Esta seção apresenta a concepção, modelagem e prototipação do sistema fuzzy do fCloud, um módulo para auxiliar na tomada de decisão sobre as migrações de MVs em um ambiente de CN baseado em LF tipo-2, e caracterizado principalmente pela capacidade em lidar com as incertezas presentes nas variáveis do ambiente.

4.1.1 Modelagem do Sistema Fuzzy para Tratamento das Incertezas

O fCloud foi desenvolvido para lidar com incertezas referentes a utilização de CPU, memória e rede nos computadores físicos presentes num ambiente de CN. Esta seção enfatiza os ganhos que o escalonador obteve com o auxílio do fCloud em situações as quais incertezas são presentes devido a oscilação nos valores das variáveis.

O fCloud considera que a disponibilidade dos valores de CPU, memória e rede são dadas por números fuzzy. O componente devolve ao escalonador uma lista de valores referentes ao nível de utilização de cada uma das máquinas físicas do ambiente.

Considera-se que os números fuzzy utilizam funções trapezoidais, para reproduzir situações em que o erro na medida extraída da grade pode ser tanto positiva como negativa. Assim, o poder computacional de um computador no ambiente é representado, utilizando a notação de números fuzzy, que representa a incerteza em torno do nível de utilização de processamento naquele momento.

De modo similar, para o valor de utilização da memória RAM de um computador no ambiente é representado, utilizando da notação de números fuzzy, que representa a incerteza em torno do nível de utilização da memória naquele momento.

Também, a largura de banda disponível entre os computadores utiliza a notação de números fuzzy, que representa a incerteza em torno do nível de utilização da largura de banda naquele momento.

4.1.2 Base de Dados e Definição das Funções de Pertinência

Antes de começar a trabalhar com as informações de entrada, fez-se necessário estudá-las juntamente com um especialista da área com a finalidade de determinar o comportamento do sistema a fim de obter valores de saída satisfatórios, de acordo com a necessidade do ambiente.

Nesta etapa, ocorre o processo de transformação de sentenças linguísticas em CFs. Também são determinados quantos e quais CF estão associados a cada variável do sistema. Ocorrendo desta forma, o processo de representação gráfico e análise funcional do comportamento das variáveis fuzzy, tanto de entrada como de saída.

Conforme a Figura 13, neste projeto a modelagem dos CFs das variáveis são formadas por funções de pertinência trapezoidais, devido ao fato de representarem adequadamente o comportamento das variáveis dentro do T2FS.

Outra questão que também teve de ser estudada com o auxílio de especialistas, foi a definição dos pontos de intersecção entre os CFs de uma mesma variável. A intersecção destes conjuntos é essencial no desenvolvimento do sistema utilizando LF, com determinados valores de entrada mostrem pesos distintos em mais de um CF. Assim o sistema poderá aplicar a técnica de defuzzificação utilizada adequadamente para obtenção das saídas desejadas.

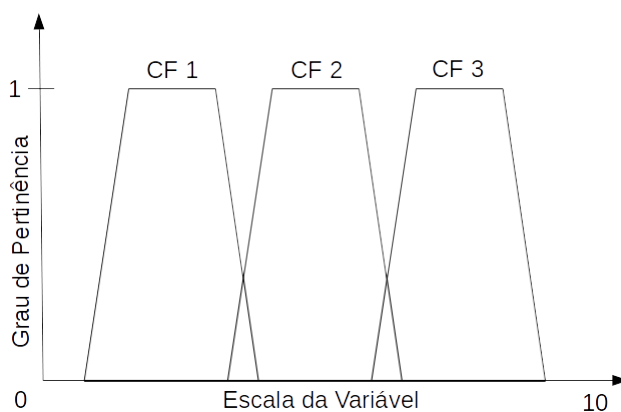


Figura 13 – Modelo adotado para as Funções de Pertinência dos Conjuntos Fuzzy.

O tratamento de incerteza e precisão utilizando a LF, foi possível dar maior robustez às informações obtidas, tornando a saída do escalonador mais confiável contribuindo para uma maior eficiência energética ao mesmo tempo mantendo desempenho.

Os processos executados pelo *fCloud*, expostos no fluxograma da Figura 14, representam a forma como ocorre a seleção dos recursos, na seguinte sequência: (i) na primeira etapa é realizada a coleta de informações sobre os recursos, (ii) depois estas informações são tratadas com uso da LF, que por fim (iii) é retornada uma lista com os valores referentes ao nível de utilização de cada máquina física do ambiente.

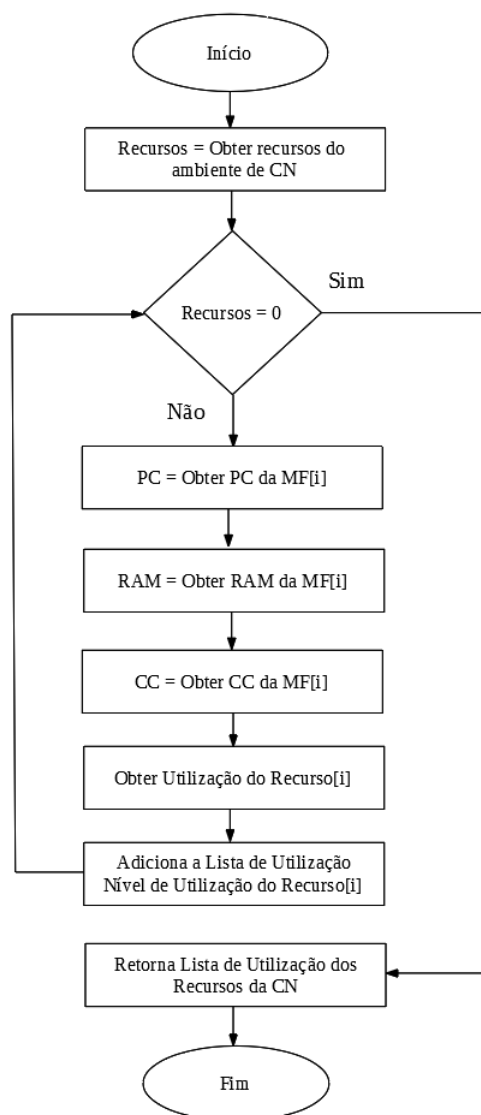


Figura 14 – Visão geral das funcionalidades do *fCloud*.

Posteriormente, cabe ao escalonador realizar o mapeamento das MFs, determinando quais precisam que suas VMs sejam migradas e quais podem receber novas VMs, de acordo com os valores de utilização apresentados pelo *fCloud*.

Como os ambientes de CN possuem uma série de recursos a serem mensurados, este projeto focou em três recursos: Poder Computacional (PC), Memória RAM (RAM) e Custo Computacional (CC). Tais recursos são considerados relevantes nesse contexto e estão destacados na Figura 15.

O *fCloud* é responsável pela análise das MFs de um ambiente heterogêneo de CN e, para isso, faz o uso de um sistema fuzzy com uma base de regras atuando em três

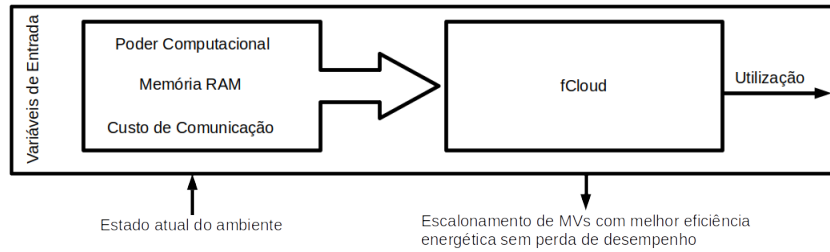


Figura 15 – Visão geral das entradas e saídas do *fCloud*.

etapas: (i) Fuzzificação, (ii) Inferência e (iii) Defuzzificação e abrangendo um módulo especial de Redutor de Tipos. Este sistema retorna como saída o nível de utilização de cada máquina.

O estudo das variáveis teve como suporte as experiências do especialista em CN, as variáveis linguísticas relativas a cada uma das variáveis de incerteza foram transformadas em CFs, usando a representação gráfica trapezoidal para suas funções de pertinência. Na modelagem do sistema concedeu-se o emprego da ferramenta Juzzy (WAGNER, 2013).

Para mensurar o PC, RAM e o CC é realizada a leitura das configurações do ambiente de CN simulado no CloudSim. Esses valores são então ajustados a uma escala padrão, considerando o intervalo $[0; 10]$, e aplicando as equações:

$$PC = (h_i(MM)/MaxPC * 10) \quad (21)$$

$$CC = ((10 * h_i(UoB))/MinCC) \quad (22)$$

$$RAM = (h_i(UoR)/MaxRam) * 10 \quad (23)$$

De acordo com os seguintes parâmetros:

- 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);
- UoB representando a utilização de largura de banda do host i ;
- UoR o uso de memória RAM no 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 (21), (22) e (23) estão associadas as funções de pertinência das Figuras 16, 17, 18, complementando as variáveis PC, CC e RAM, respectivas. Os Termos

Linguísticos (TLs) definidos para os CFs da variável PC são: “Limitado” (PCL), “Razoável” (PCR) e “Elevado” (PCE - melhor caso). Sendo que para $PC = a$ e $a \in [0; 10]$, têm-se as Funções de Pertinência da Tabela 4, as quais são graficamente descritas na Figura 16.

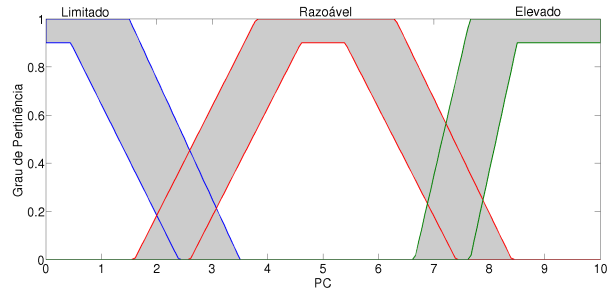


Figura 16 – PC na escala padrão

PC	$\mu_A(x)$	$\mu_A(x)$
PCL	$\begin{cases} 1, & \text{se } 0 \leq x < 1.5 \\ \frac{-x+3.5}{2}, & \text{se } 1.5 \leq x < 3.5 \\ 0, & \text{caso contrário.} \end{cases}$	$\begin{cases} 0.9, & \text{se } 0 \leq x < 0.45 \\ \frac{-0.9x+2.4}{2.4}, & \text{se } 0.45 \leq x < 2.4 \\ 0, & \text{caso contrário.} \end{cases}$
PCR	$\begin{cases} \frac{x-1.6}{2.2}, & \text{se } 1.6 \leq x < 3.8, \\ 1, & \text{se } 3.8 \leq x < 6.3, \\ \frac{-x+8.4}{2.1}, & \text{se } 6.3 \leq x < 8.4, \\ 0, & \text{caso contrário.} \end{cases}$	$\begin{cases} 0.45x - 1.17, & \text{se } 2.6 \leq x < 4.6, \\ 0.9, & \text{se } 4.6 \leq x < 5.4, \\ -0.45x + 3.33, & \text{se } 5.4 \leq x < 7.4, \\ 0, & \text{caso contrário.} \end{cases}$
PCE	$\begin{cases} x - 6.65, & \text{se } 6.65 \leq x < 7.65, \\ 1, & \text{se } 7.65 \leq x < 10, \\ 0, & \text{caso contrário.} \end{cases}$	$\begin{cases} \frac{18x}{17} - 8.1, & \text{se } 7.65 \leq x < 8.5, \\ 0.9, & \text{se } 8.5 \leq x < 10, \\ 0, & \text{caso contrário.} \end{cases}$

Tabela 4 – Funções de Pertinência PC

A variável RAM reporta-se, a porcentagem de utilização de memória RAM de uma máquina em determinado momento. Os TLs para os CFs definidos para essa variável são: “Baixo” (RAMB), “Médio” (RAMM) e “Grande” (RAMG - melhor caso). Sendo que para $RAM = b$ e $b \in [0; 10]$, têm-se as Funções de Pertinência da Tabela 5 que, graficamente estão descritas na Figura 17.

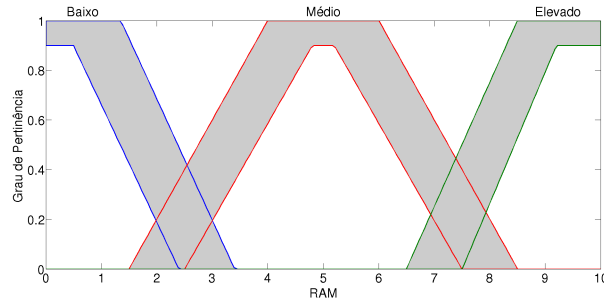


Figura 17 – RAM na escala padrão

RAM	$\mu_A(x)$	$\mu_A(x)$
RAMB	$\begin{cases} 1, \text{ se } 0 \leq x < 1.5 \\ \frac{-x+3.5}{2}, \text{ se } 1.5 \leq x < 3.5 \\ 0, \text{ caso contrário.} \end{cases}$	$\begin{cases} 0.9, \text{ se } 0 \leq x < 0.45 \\ \frac{-0.9x+2.4}{2.4}, \text{ se } 0.45 \leq x < 2.4 \\ 0, \text{ caso contrário.} \end{cases}$
RAMM	$\begin{cases} \frac{x-1.6}{2.2}, \text{ se } 1.6 \leq x < 3.8, \\ 1, \text{ se } 3.8 \leq x < 6.3, \\ \frac{-x+8.4}{2.1}, \text{ se } 6.3 \leq x < 8.4, \\ 0, \text{ caso contrário.} \end{cases}$	$\begin{cases} 0.45x - 1, 17, \text{ se } 2.6 \leq x < 4.6, \\ 0.9, \text{ se } 4.6 \leq x < 5.4, \\ -0.45x + 3.33, \text{ se } 5.4 \leq x < 7.4, \\ 0, \text{ caso contrário.} \end{cases}$
RAMG	$\begin{cases} x - 6.65, \text{ se } 6.65 \leq x < 7.65, \\ 1, \text{ se } 7.65 \leq x < 10, \\ 0, \text{ caso contrário.} \end{cases}$	$\begin{cases} \frac{18x}{17} - 8.1, \text{ se } 7.65 \leq x < 8.5, \\ 0.9, \text{ se } 8.5 \leq x < 10, \\ 0, \text{ caso contrário.} \end{cases}$

Tabela 5 – Funções de Pertinência RAM

A variável CC reporta-se a porcentagem de utilização de banda da máquina sobre o máximo que essa suporta. Os TLs para os CFs definidos para essa variável são: “Pequeno” (CCP - melhor caso), “Médio” (CCM) e “Grande” (CCG). Sendo que para $CC = c$ e $c \in [0; 10]$, têm-se as Funções de Pertinência da Tabela 6 que, estão descritas graficamente na Figura 18.

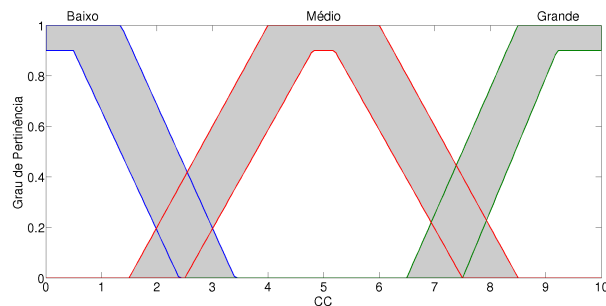


Figura 18 – CC na escala padrão

CC	$\overline{\mu_B \tilde{x}}$	$\overline{\mu_B \tilde{x}}$
CCP	$\begin{cases} 1, & \text{se } 0 \leq x < 1.35, \\ \frac{-x+3.4}{2.05}, & \text{se } 1.35 \leq x < 3.4, \\ 0, & \text{caso contrário.} \end{cases}$	$\begin{cases} 0.9, & \text{se } 0 \leq x < 0.5, \\ \frac{-0.9x+2.16}{1.9}, & \text{se } 0.5 \leq x < 2.4, \\ 0, & \text{caso contrário.} \end{cases}$
CCM	$\begin{cases} \frac{x-1.5}{2.5}, & \text{se } 1.5 \leq x < 2.5, \\ 1, & \text{se } 4 \leq x < 6, \\ \frac{-x+8.5}{2.5}, & \text{if } 6 \leq x < 8.5, \\ 0, & \text{caso contrário.} \end{cases}$	$\begin{cases} \frac{0.9x-2.25}{2.3}, & \text{se } 2.5 \leq x < 4.8, \\ 0.9, & \text{se } 4.8 \leq x < 5.2, \\ \frac{-0.9x+6.75}{2.3}, & \text{se } 5.2 \leq x < 7.5, \\ 0, & \text{caso contrário.} \end{cases}$
CCG	$\begin{cases} \frac{x-6.5}{2}, & \text{se } 6.5 \leq x < 8.5, \\ 1, & \text{se } 8.5 \leq x < 10, \\ 0, & \text{caso contrário.} \end{cases}$	$\begin{cases} \frac{18x-13.5}{3.4}, & \text{se } 7.5 \leq x < 9.2, \\ 0.9, & \text{se } 9.2 \leq x < 10, \\ 0, & \text{caso contrário.} \end{cases}$

Tabela 6 – Funções de Pertinência CC

A saída U , que define o nível de utilização das máquinas em determinado momento, também é adaptada para uma escala padrão, conforme mostra a representação gráfica das correspondentes funções de pertinência na Figura 19. Os TLs para os CFs usados nesse caso são: “Baixa” (UB), “Média” (UM) e “Alta” (UA). Sendo $U = d$ e $d \in [0; 10]$, têm-se as Funções de Pertinência da Tabela 7.

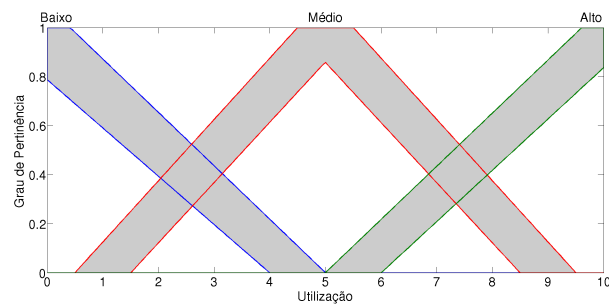


Figura 19 – U na escala padrão

U	$\overline{\mu_C \bar{x}}$	$\mu_C \bar{x}$
UB	$\begin{cases} \frac{x}{1.16}, & \text{se } 0 \leq x < 1.16, \\ 1, & \text{se } 1.16 \leq x < 2.16, \\ \frac{-x+3.5}{1.34}, & 2.16 \leq x < 3.5, \\ 0, & \text{caso contrário.} \end{cases}$	$\begin{cases} \frac{0.9x-0.45}{1.333}, & \text{se } 0.5 \leq x < 1.66, \\ \frac{-0.9x+2.52}{1.3}, & \text{se } 1.66 \leq x < 2.8, \\ 0, & \text{caso contrário.} \end{cases}$
UM	$\begin{cases} \frac{-0.9x-3.447}{1.333}, & \text{se } 3.15 \leq x < 4.5, \\ 1, & \text{se } 4.5 \leq x < 5.5, \\ \frac{-x+6.83}{1.327}, & \text{se } 5.5 \leq x < 6.83, \\ 0, & \text{caso contrário.} \end{cases}$	$\begin{cases} \frac{0.9x-3.447}{1.33}, & \text{se } 3.83 \leq x < 4.99, \\ \frac{-0.9x+5.544}{1.12}, & \text{se } 4.99 \leq x < 6.16, \\ 0, & \text{caso contrário.} \end{cases}$
UA	$\begin{cases} \frac{x-6.5}{1.33}, & \text{se } 6.5 \leq x < 7.83, \\ 1, & \text{se } 7.83 \leq x < 8.83, \\ \frac{-x+10}{1.17}, & \text{se } 8.83 \leq x < 10, \\ 0, & \text{caso contrário.} \end{cases}$	$\begin{cases} \frac{0.9x-6.444}{1.34}, & \text{se } 7.16 \leq x < 8.33, \\ \frac{-0.9x+8.55}{1.34}, & \text{se } 8.33 \leq x < 9.5, \\ 0, & \text{caso contrário.} \end{cases}$

Tabela 7 – Funções de Pertinência U

4.1.3 Fuzzificação

Nessa etapa, ocorre o mapeamento dos valores de entrada (já ajustados para escala observada na Seção 4.1.1) para o domínio fuzzy, como pode ser observado na Figura 20.

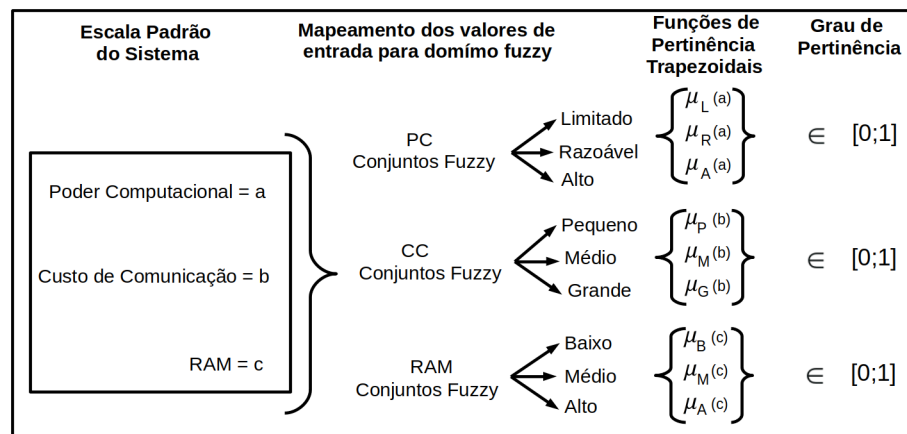


Figura 20 – Processo de Fuzzificação do fCloud

4.1.4 Base de regras

A Base de Regras (BR) do *fCloud* é desenvolvida com o intuito de ser facilmente compreensível e editável, visto que não há dificuldade em adicionar novas regras, caso seja desejado adicionar variáveis de entrada. A BR, observada na Tabela 8, leva em conta que o bom desempenho do sistema fuzzy está condicionado às regras que descrevem a estratégia de controle de forma consistente (KLIR, 2005). Leva-se em conta três fatores para sua construção:

- Modelando o sistema mais perto do **mundo real**;

- São utilizadas conexões lógicas do tipo “AND” para criar a relação entre as variáveis de entrada;
- As implicações são do tipo *modus ponens* (modo afirmativo):

Se ((x é A) e (y é B) e (z é C)) então t é D.

Tabela 8 – Base de Regras fCloud

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

4.1.5 Inferência

No processo de Inferência, ocorrem as operações entre os CFs, com combinação dos antecedentes das regras e aplicação de implicações utilizando o operador *modus ponens generalizado*. Este processo ocorre em três etapas detalhadas a seguir.

- (i) **Aplicação da Operação Fuzzy:** nesta etapa, ocorre a aplicação dos operadores fuzzy sendo que a entrada consta de três valores resultantes da fuzzificação. Como as regras são formadas pelo operador fuzzy “AND”, a aplicação utiliza o método MIN (mínimo) sobre os dois valores retornados da fuzzificação;
- (ii) **Aplicação do Método de Implicação Fuzzy:** nesta etapa, é realizada uma combinação entre o valor obtido na aplicação do operador fuzzy e os valores do CF de saída da regra, utilizando o método IMP (implicação) sobre estas combinações;
- (iii) **Aplicação do Método de Agregação Fuzzy:** nesta etapa, ocorre a composição dos resultados fuzzy de saída de cada regra, utilizando o método MAX (máximo), assim criando uma única região fuzzy para ser analisada pelo próximo processo do módulo fuzzy.

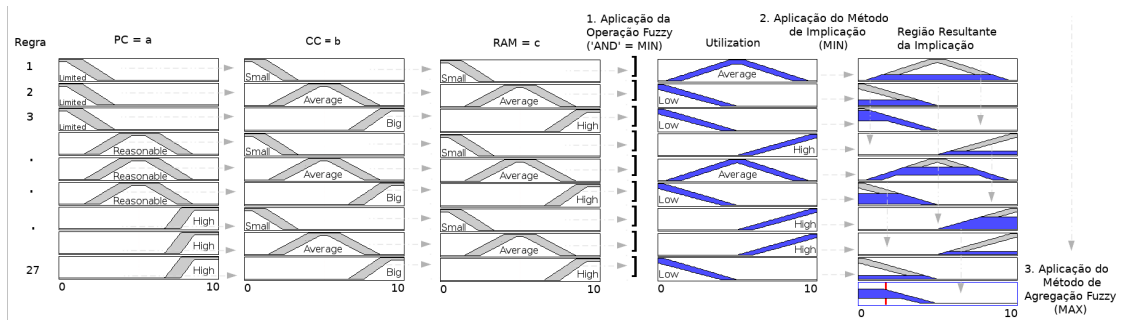


Figura 21 – Processo de Inferência do fCloud

4.1.6 Defuzzificação

Nessa etapa, ocorre a transformação da região resultado da Inferência em um valor discreto que representa a Utilização (U). Essa transformação foi feita no *fCloud* através do emprego do método centro da área. Esse método calcula o centroide (x) da área composta pela saída fuzzy do sistema de Inferência, o qual considera a união de todas as contribuições de regras discutidas nas seções 4.1.4 e 4.1.5. A expressão analítico da centroide é dada pela equação

$$u = \frac{\sum_{i=1}^N u_i \mu_{OUT}(u_i)}{\sum_{i=1}^N \mu_{OUT}(u_i)} \quad (24)$$

4.2 Visão Geral dos Procedimentos Operacionais do fCloud

Nesta seção, a visão geral dos procedimentos relacionados à operação do fCloud é discutida, apresentando de de forma integrada todas as etapas descritas na Seção 4.

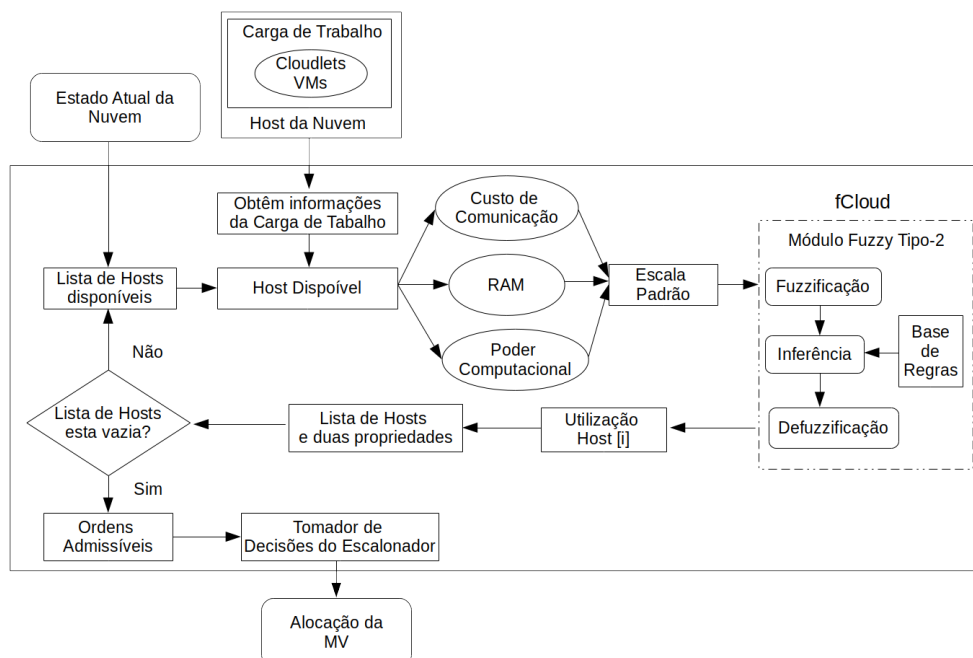


Figura 22 – Fluxograma dos processos operacionais do fCloud.

A visão geral dos processos envolvidos na operação do *fCloud* é descrita pelo fluxograma apresentado na Figura 22. 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 duas variáveis são identificadas (i) o estado atual do ambiente da CN; (ii) as máquinas que estão disponíveis, sendo ambas armazenadas em uma lista. A seguir são coletadas as informações de CC, RAM e PC de cada máquina disponível, adaptando-os para uma Escala Padrão, onde estarão prontas para ingressarem no Módulo Fuzzy Tipo-2. O módulo fuzzy tem a composição de três etapas: (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 é verificado se ainda há alguma máquina disponível não analisada. Enquanto houver, analisa os dados e armazena na lista de Níveis de Utilização. Assim obtêm-se 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 descritas na Seção 3.3.4. Neste caso, têm-se uma solução para o problema de incompatibilidade dos intervalos no procedimento. 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 que devem ter MVs realocadas, e quais podem receber MVs, 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 MVs 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 MVs 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 MVs.

5 F-CLOUD: ANÁLISE DE DESEMPENHO

Este capítulo apresenta as avaliações realizadas durante o andamento do projeto. Os níveis de utilização dos recursos obtidos através do emprego do módulo fCloud com SBRF2 foram aplicados na tomada de decisão quando da realização do escalonamento de MVs em um ambiente simulado de CN.

5.1 Configuração do ambiente

As avaliações empregaram, nesta pesquisa, simulações realizadas com uso das funcionalidades do framework CloudSim (CALHEIROS et al., 2011), um kit de ferramentas para modelagem e simulação de serviços de CN. Na escolha do ambiente, considerou-se compatibilidade de código, suporte a T2FL e abordagem *open-source* das bibliotecas

Na implementação do sistema de inferência fuzzy do f-Cloud foi utilizada a biblioteca Java de código aberto Juzzy (WAGNER, 2013), devido a seu potencial na manipulação de SBRF2, e ainda, por ser concebida em Java, mesma linguagem que o CloudSim foi desenvolvido. O uso de tecnologias compatíveis viabilizou a integração entre o ambiente simulado de CN, e o sistema de inferência fuzzy.

Para todas as simulações, considerou-se também as cargas de trabalho do mundo real fornecidas como parte do projeto CoMon (PARK; PAI, 2006), uma infraestrutura de monitoramento para o PlanetLab (CHUN et al., 2003). O ambiente de nuvem IaaS, aplicado para os testes, considerou um *data center* de grande escala com 800 hosts físicos heterogêneos, contemplando dois tipos de configurações, como descritas na Tabela 9.

Fabricante	Modelo	Nome do CPU	Características de CPU	RAM (GB)
Hewlett Packard Enterprise	ProLiant DL325 Gen10	AMD EPYC 7551P 2.0 GHz	32-Core, 2.0 GHz, 64MB L3 Cache	128
Dell Inc.	PowerEdge R840 (Intel Xeon Platinum 8180 2.50 GHz)	Intel Xeon Platinum 8180 2.50 GHz	28 core, 2.50 GHz, 38.5 MB L3 Cache	384

Tabela 9 – Características dos hosts do ambiente da nuvem

Na Figura 23, são apresentadas as características consideradas na arquitetura de nuvem computacional simulada.

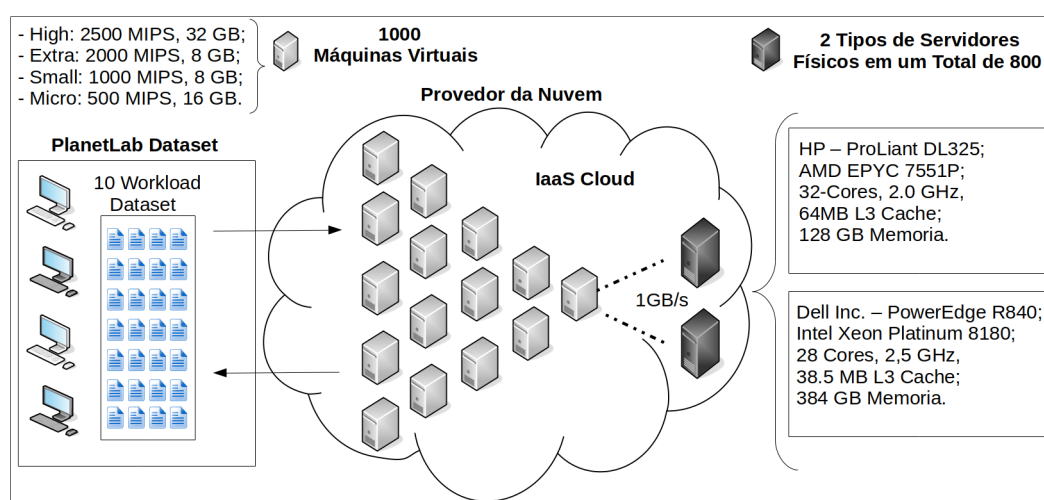


Figura 23 – Características da Arquitetura da Nuvem Simulada.

A frequência das CPUs do servidor é descrita em MIPS. 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 5 GB/s de largura de banda de rede. As características dos tipos de MV são semelhantes aos tipos de instância do Amazon EC2 (AMAZON, 2015), incluindo Instância de CPU Média Alta (4000 MIPS, 32 GB); Instância Extra Grande (3000 MIPS, 8 GB); Instância Pequena (2000 MIPS, 8 GB); e Micro Instância (2000 MIPS, 16 GB). O intervalo de medição de uso (intervalo de escalonamento) é de 5 minutos. As características de cada carga de trabalho são mostradas na Tabela 10.

Carga de Trabalho	VMs	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

Tabela 10 – Configuração das Cloudlets

Com esses dez conjuntos de dados de carga de trabalho, aloca-se cada uma das MVs de acordo com suas necessidades. Para a etapa de seleção de algoritmos foi considerada a pesquisa de (BELOGLAZOV; BUYYA, 2012), devido a gama de testes já executados que possibilitaram uma comparação com outros trabalhos. Em todos os experimentos foram utilizados os algoritmos de alocação, que são eles: (i) Intervalo Interquartílico (*Inter Quartile Range (IQR)*), (ii) Regressão Robusta Local (*Local Robust Regression (LRR)*), (iii) Desvio Absoluto Mediano (*Median Absolute Deviation (MAD)*) e (iv) Limite estático (*Static Threshold (THR)*). Em cada execução a política de seleção de VMs usada foi a Seleção aleatória (*Random Selection (RS)*).

5.2 Resultados

Nesta seção apresentam-se os resultados obtido na andamento do trabalho.

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.

5.2.1 Estudo de caso 1

A avaliação energética é apresentada na Figura 24. Quanto menor o consumo mais positivo é o resultado.

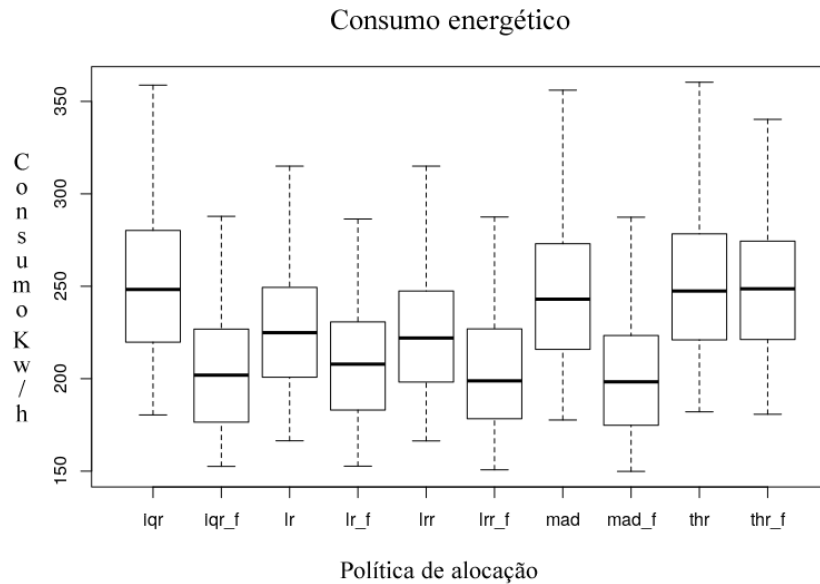


Figura 24 – Consumo energético

É possível observar a economia de energia em relação aos outros algoritmos: para a política IQR_fuzzy em comparação com a IQR clássica o ganho foi de 15.43%, no caso da política LR_fuzzy em comparação a LR clássica o ganho foi de 9.39%, na MAD_fuzzy comparada com a MAD clássica o ganho foi de 15.64% e por fim na THR_fuzzy comparada com a THR clássica o ganho foi de 0.7%.

A percentagem do número de VMs que infringiram o SLA é apresentado na Figura 25, quanto menor o valor melhor é o resultado.

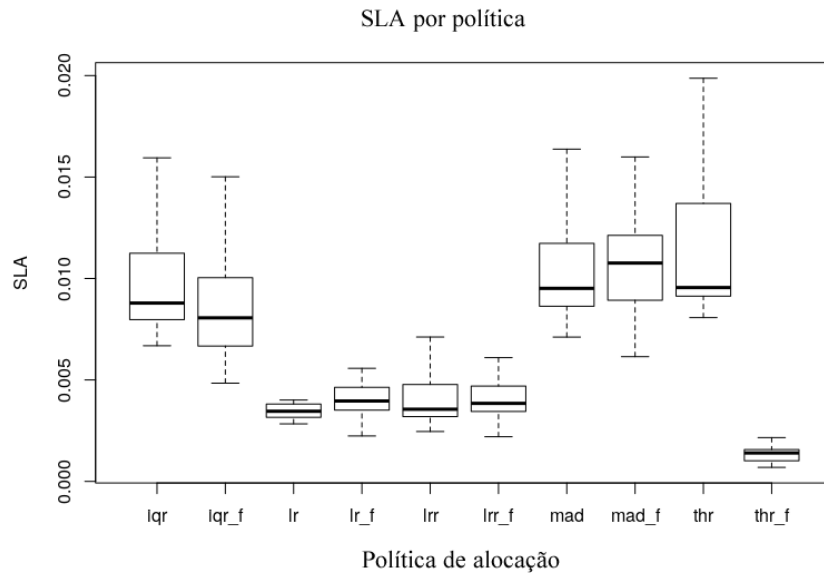


Figura 25 – Violação de SLA por política

No caso da política IQR_fuzzy a média ficou 15.68% mais baixa que o IRQ clássico, no LR_fuzzy ficou 2.5% maior que o LR clássico, no LRR_fuzzy ficou 4.76% menor comparado com o LRR clássico, no MAD_fuzzy ficou 0.93% maior comparado com o MAD clássico e por fim no THR_fuzzy ficou 88.8% menor que o THR clássico.

O tempo médio que o SLA permaneceu sendo infringido é apresentado na Figura 26, quanto menor o valor melhor é o resultado.

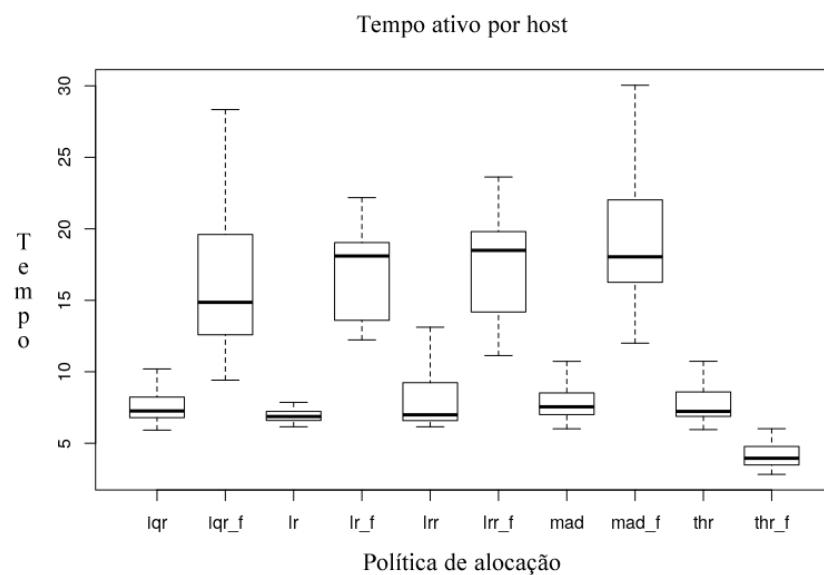


Figura 26 – Tempo médio de SLA infringido por host

No caso da política IQR_fuzzy a média obtida foi 50.77% maior que a IQR clássica, na LR_fuzzy a média obtida foi 54.71% maior que a LR clássica, na LRR_fuzzy a média obtida foi 53.71% maior comparado com a LRR clássica, na MAD_fuzzy a média obtida foi 58.62% maior comparado com a MAD clássica e, por fim, na THR_fuzzy a média obtida foi 47.58% menor que o THR clássico.

A quantidade de MVs migradas é apresentada na Figura 27, quanto menor o valor melhor é o resultado.

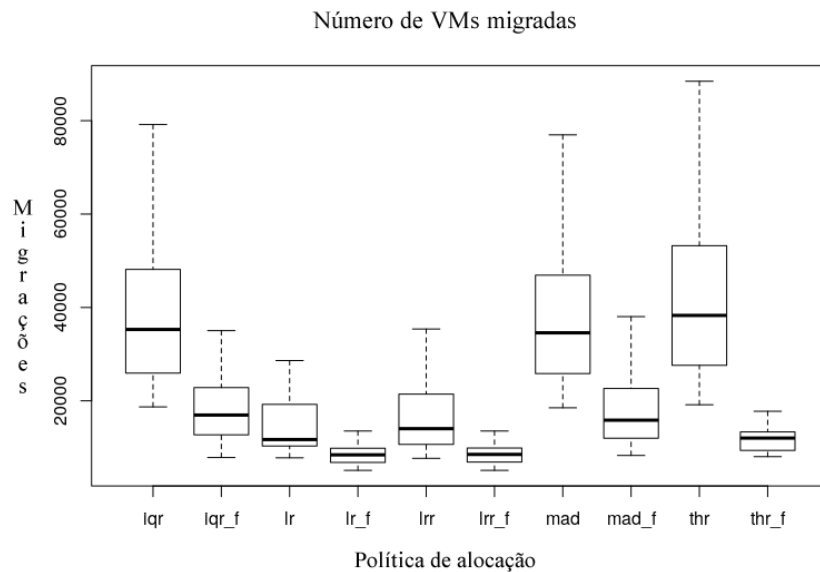


Figura 27 – VMs migradas

Foram obtidos os valores de 49.36%, 41.67%, 51.56% , 71.95% , são respectivamente para a diferença da média de IQRF, LRF, LRRF, MADF e THRF obtendo menor número de MVs migradas comparado com suas respectivas abordagens que empregam a lógica clássica.

A Degradação do SLA devido a migração de MVs é apresentada na Figura 28, quanto menor o valor melhor é o resultado.

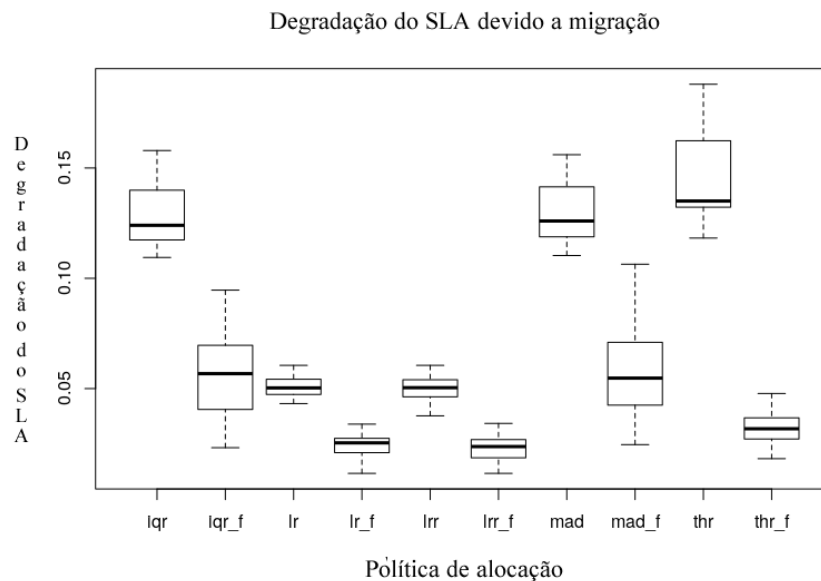


Figura 28 – Degradação do SLA devido a migração de MVs

Foram obtidos os valores de 54.97%, 52.26%, 54.45%, 55.18% e 77.70%, são respectivamente para a diferença da média de IQRf, LRF, LRRf, MADf e THRF obtendo menor degradação comparado com suas respectivas abordagens que empregam a lógica clássica.

5.2.2 Estudo de caso 2

Nesta seção, são apresentados os resultados obtidos com o emprego dos métodos das Ordens Admissíveis para ordenação da lista de intervalos obtida como saída do SBRF2, com intuito da detecção mais aproximada do nível de utilização dos hosts no ambiente de nuvem simulado.

A avaliação do consumo de energia é apresentada através do boxplot da Figura 29.

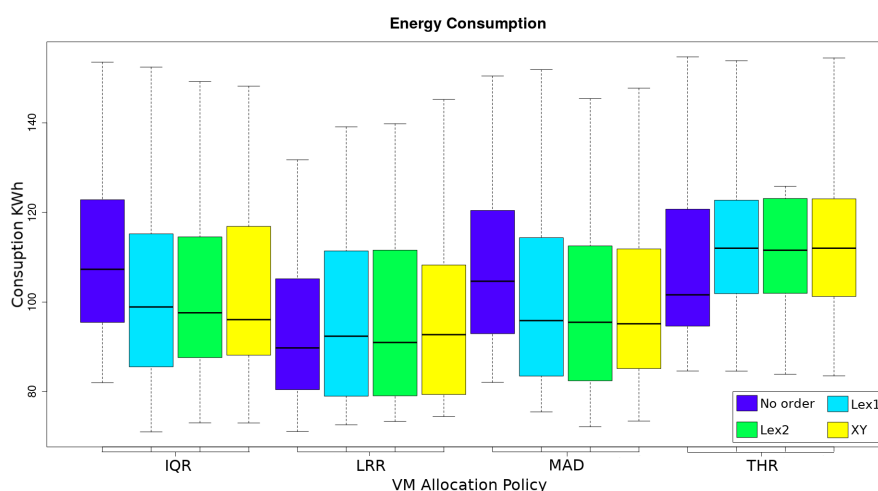


Figura 29 – Consumo Energético

Pode-se observar que a abordagem de seleção de host proposta com emprego das Ordens Admissíveis alcançou ganhos consideráveis em economia de energia, com a política IQR_XY em relação ao IQR alcançou 8,83% de ganho, no caso do MAD_Lex2 em relação para MAD alcançou 8,72%, em THR_XY comparado a THR 22,43%.

O boxplot da Figura 30 apresenta resultados de SLA.

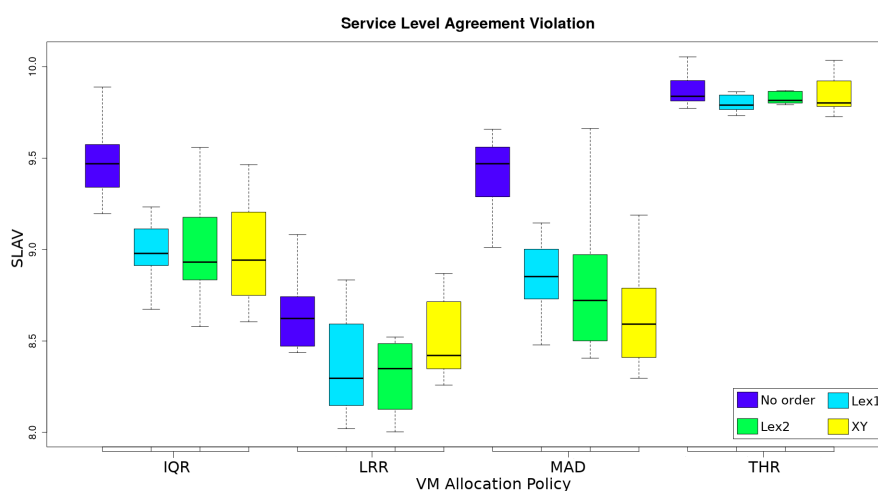


Figura 30 – Média de Violação de SLA

A abordagem proposta alcançou resultados de 6.11% com IQR_XY comparado a IQR, para LRR_Lex2 4.02% em relação a LRR, THR_Lex1 25% em oposição a THR e finalmente, MAD_XY analisado a MAD obteve 9.06%, correspondendo as melhores médias de violação de SLA.

6 CONCLUSÃO

Neste trabalho, apresentamos o desenvolvimento do módulo fCloud: Sistema Fuzzy que seja capaz de tratar a incerteza do especialista em determinar a utilização de uma máquina física em um ambiente de CN usando Lógica Fuzzy Tipo-2 e Ordens Admissíveis como parte da consolidação das VMs.

Foram simulados e comparados com o f-Cloud cinco algoritmos conhecidos: Intervalo Interquartilico (*Inter Quartile Range (IQR)*), Regressão Robusta Local (*Local Robust Regression (LRR)*), Desvio Absoluto Mediano (*Median Absolute Deviation (MAD)*) e Limite estático (*Static Threshold (THR)*).

Em cada execução, a política de seleção de VMs usada foi a Seleção aleatória (*Random Selection (RS)*).

Todos os algoritmos escolhidos tratam as incertezas presentes no ambiente usando uma abordagem clássica. No f-Cloud, a abordagem mostra o potencial da lógica fuzzy, conquistando ganhos consideráveis em especial na eficiência energética, atingindo ganhos de 15.64, 15.43 e 9.19% em relação a MAD, IRQ e LR respectivamente.

Também houve um ganho expressivo no caso do tempo médio que o SLA permaneceu sendo infringido, no caso do algoritmo THR o ganho foi de 47.58%.

No caso da abordagem que explorava uso de ordens admissíveis, na maioria dos casos obteve-se bons resultados em relação a eficiência energética, alcançando ganhos de 8.83%, 22.43% em relação aos algoritmos clássicos IRQ e MAD, usando ordens de Xu e Yager. Também houve um ganho expressivo considerando o SLA, com ganhos de 9.06% no caso do MAD com ordem de Xu e Yager e de 25% na THR utilizando ordem Lex1.

Os trabalhos futuros buscam otimizações no f-Cloud no sentido de novas políticas de alocação de MVs. No uso de fundamentação baseada em LF, considera-se o uso de múltiplos especialistas com suporte na análise de medidas de consenso para conciliação de opiniões.

REFERÊNCIAS

AHMED, A.; SABYASACHI, A. S. Cloud computing simulators: A detailed survey and future direction. In: ADVANCE COMPUTING CONFERENCE (IACC), 2014 IEEE INTERNATIONAL, 2014. **Anais...** [S.l.: s.n.], 2014. p.866–872.

AMAZON, E. Amazon web services. **Available in:** [http://aws. amazon. com/es/ec2/](http://aws.amazon.com/es/ec2/)(November 2012), [S.l.], 2015.

BACZYŃSKI, M. Residual implications revisited. Notes on the Smets–Magrez theorem. **Fuzzy Sets and Systems**, [S.l.], v.145, n.2, p.267–277, 2004.

BACZYŃSKI, M.; JAYARAM, B. **An Introduction to Fuzzy Implications**. [S.l.]: Springer, 2008.

BALASUBRAMANIAM, J. Yager's new class of implications J_f and some classical tautologies. **Information Sciences**, [S.l.], v.177, n.3, p.930–946, 2007.

BATISTA, B. G. **Modelos de negócio para ambientes de computação em nuvem que consideram atributos de qos relacionados a desempenho e a segurança**. 2016. Tese (Doutorado em Ciência da Computação) — Universidade de São Paulo.

BELOGLAZOV, A. **Energy-efficient management of virtual machines in data centers for cloud computing**. 2013. Tese (Doutorado em Ciência da Computação) —

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. **Concurrency and Computation: Practice and Experience**, [S.l.], v.24, n.13, p.1397–1420, 2012.

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**, [S.l.], v.24, n.7, p.1366–1379, July 2013.

BUSTINCE, H.; FERNÁNDEZ, J.; KOLESÁROVÁ, A.; MESIAR, R. Generation of linear orders for intervals by means of aggregation functions. **Fuzzy Sets and Systems**, [S.l.], v.220, p.69–77, 2013.

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**, [S.l.], v.41, n.1, p.23–50, 2011.

CHIEU, T. C.; MOHINDRA, A.; KARVE, A. A.; SEGAL, A. Dynamic scaling of web applications in a virtualized cloud computing environment. In: IEEE INTERNATIONAL CONFERENCE ON E-BUSINESS ENGINEERING, 2009., 2009. **Anais...** [S.l.: s.n.], 2009. p.281–286.

CHUN, B. et al. Planetlab: an overlay testbed for broad-coverage services. **ACM SIGCOMM Computer Communication Review**, [S.l.], v.33, n.3, p.3–12, 2003.

FERDAUS, M. H.; MURSHED, M.; CALHEIROS, R. N.; BUYYA, R. Virtual machine consolidation in cloud data centers using ACO metaheuristic. In: EUROPEAN CONFERENCE ON PARALLEL PROCESSING, 2014. **Anais...** [S.l.: s.n.], 2014. p.306–317.

GEORGE J, K.; BO, Y. Fuzzy sets and fuzzy logic, theory and applications. -, [S.l.], 2008.

HAMDAQA, M.; TAHVILDARI, L. Cloud computing uncovered: a research landscape. In: **Advances in Computers**. [S.l.]: Elsevier, 2012. v.86, p.41–85.

HILTUNEN, M. A. et al. Mistral: Dynamically Managing Power, Performance, and Adaptation Cost in Cloud Infrastructures. In: IEEE 30TH INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS(ICDCS), 2010., 2010. **Anais...** [S.l.: s.n.], 2010. v.00, p.62–73.

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**, [S.l.], v.96, n.7, p.1911–1930, 2011.

KLIAZOVICH, D.; BOUVRY, P.; KHAN, S. U. GreenCloud: a packet-level simulator of energy-aware cloud computing data centers. **The Journal of Supercomputing**, [S.l.], v.62, n.3, p.1263–1283, 2012.

KLIR, G. J. **Uncertainty and Information: Foundations of Generalized Information Theory**. [S.l.]: Wiley-Interscience Malden, USA, 2005.

MELL, P.; GRANCE, T. et al. The NIST definition of cloud computing. , [S.l.], 2011.

MENDEL, J. M. **Uncertain rule-based fuzzy logic systems**: introduction and new directions. [S.l.]: Prentice Hall PTR Upper Saddle River, 2001.

MOURA, B. M. et al. Int-fGrid: BoT Tasks Scheduling Exploring Fuzzy Type-2 in Computational Grids. In: IEEE INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS (FUZZ-IEEE), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p.1–8.

NATHANI, A.; CHAUDHARY, S.; SOMANI, G. Policy based resource allocation in IaaS cloud. **Future Generation Computer Systems**, [S.l.], v.28, n.1, p.94–103, 2012.

PARK, K.; PAI, V. S. CoMon: A Mostly-scalable Monitoring System for PlanetLab. **SI-GOPS Oper. Syst. Rev.**, New York, NY, USA, v.40, n.1, p.65–74, Jan. 2006.

RIZOL, P.; MESQUITA, L.; SAOTOME, O. Lógica Fuzzy Tipo-2. **Revista Sodebras**, [S.l.], v.6, p.27–46, 2011.

SHEHABI, A. et al. United states data center energy usage report. , [S.l.], 2016.

TOOSI, A. N.; BUYYA, R. A Fuzzy Logic-Based Controller for Cost and Energy Efficient Load Balancing in Geo-distributed Data Centers. In: IEEE/ACM 8TH INTERNATIONAL CONFERENCE ON UTILITY AND CLOUD COMPUTING (UCC), 2015., 2015. **Anais...** [S.l.: s.n.], 2015. p.186–194.

WAGNER, C. Juzzy - A Java based toolkit for Type-2 Fuzzy Logic. In: IEEE SYMPOSIUM ON ADVANCES IN TYPE-2 FUZZY LOGIC SYSTEMS (T2FUZZ), 2013., 2013. **Anais...** [S.l.: s.n.], 2013. p.45–52.

WICKREMASINGHE, B.; CALHEIROS, R. N.; BUYYA, R. Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications. In: ADVANCED INFORMATION NETWORKING AND APPLICATIONS (AINA), 2010 24TH IEEE INTERNATIONAL CONFERENCE ON, 2010. **Anais...** [S.l.: s.n.], 2010. p.446–452.

XU, Z.; YAGER, R. R. Some geometric aggregation operators based on intuitionistic fuzzy sets. **International journal of general systems**, [S.l.], v.35, n.4, p.417–433, 2006.

YAGER, R. R. Generalized OWA aggregation operators. **Fuzzy Optimization and Decision Making**, [S.l.], v.3, n.1, p.93–107, 2004.

YAGER, R. R. OWA aggregation over a continuous interval argument with applications to decision making. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, [S.l.], v.34, n.5, p.1952–1963, 2004.

ZADEH, L. A. Fuzzy Sets. **Information and Control**, [S.l.], n.8, p.338–358, 1965.

ZADEH, L. A. The concept of a linguistic variable and its application to approximate reasoning—I. **Information sciences**, [S.l.], v.8, n.3, p.199–249, 1975.

ZADEH, L. A. Is there a need for fuzzy logic? **Information sciences**, [S.l.], v.178, n.13, p.2751–2779, 2008.

ZAPATA, H. et al. Interval-valued implications and interval-valued strong equality index with admissible orders. **International Journal of Approximate Reasoning**, [S.l.], v.88, p.91–109, 2017.

ZHANG, Q.; CHENG, L.; BOUTABA, R. Cloud computing: state-of-the-art and research challenges. **Journal of internet services and applications**, [S.l.], v.1, n.1, p.7–18, 2010.