

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



Trabalho de Conclusão de Curso

**Um Módulo de Gerenciamento e Sensoriamento de Recursos Computacionais
de um Ambiente em Nuvem com Lógica Fuzzy**

Ariel Furtado Azevedo

Pelotas, 2015

Ariel Furtado Azevedo

**Um Módulo de Gerenciamento e Sensoriamento de Recursos Computacionais
de um Ambiente em Nuvem com Lógica Fuzzy**

Trabalho de Conclusão de Curso apresentado
ao Centro de Desenvolvimento Tecnológico
da Universidade Federal de Pelotas, como re-
quisito parcial à obtenção do título de Bacha-
rel em Ciência da Computação

Orientador: Maurício Lima Pilla
Colaborador: Vilnei Marins Neves

Pelotas, 2015

**Dedico esse TCC principalmente a mim mesmo!!! Sem mim ele não teria saído,
com certeza! Também dedico a minha avó, e minha tia que me ajudaram ao
longo de todo o caminho.
E dedico a todos os meus amigos, os quais alguns cito nos agradecimentos,
outros não, mas a todos vocês dedico esse trabalho.**

AGRADECIMENTOS

Bom, nunca fui muito bom com palavras, mas tentarei ao máximo mencionar e agradecer a todos envolvidos e que me apoiaram ao longo desse caminho. Por ordem dos fatos então, primeiro gostaria de agradecer aos Vogons, por não terem destruído a terra para criar uma estrada do hiperespaço, se isso tivesse ocorrido, provavelmente eu não estaria aqui hoje. Depois disso, gostaria de agradecer ao meu tio Ademir que sempre me incentivou a entrar em uma universidade e me ajudou muito para que isso ocorresse, agradecer também a todos meus amigos de Canoas, Erick, Digo, Flávinho (nem tanto), Gregory e Dandy, também os pais do Erick, tio Élio e Tia "Cenira", e do Digo, Tia Rita e Tio Flávio, os quais alguns até me deram moradia por um tempo e me suportaram e apoiaram, como fazem ainda hoje. Depois disso agradecer a minha avó, Diva, por ter me acolhido de volta a cidade para então embarcar nessa jornada que é a primeira graduação, e meus amigos da cidade que apesar de tudo sempre pagaram bebida pra mim, sem eles não seria a mesma coisa.

Agradecimentos especiais então a três amigos que são mais irmãos que qualquer outra coisa, Pablo, Bruno, Henrique que sempre estiveram comigo pra tudo, e sempre me apoiaram até nas piores coisas que já me aconteceram, também agradecimentos especiais a todos os meus colegas que continuam no curso hoje, que são poucos, tipo 3 ou 4 (risos), João, Balotinha, Reginho Freddy e Mauricio, tá, 5, que apesar de todo esse tempo e de quase não frequentar mais as mesmas disciplinas continuam de boa e junto pra sempre dar uma força, ao Vilnei (Já pagou bebida pra mim, merece o espaço dele aqui) que sempre me ajudou, inclusive esse trabalho foi feito por causa dele, usando como base a própria pesquisa dele. Depois também gostaria de agradecer ao meu orientador, que a maioria pode até não achar, mas é um dos professores mais legais (não é puxar saco e vocês sabem) do curso, que aceitou a mim como orientando sabendo o quão preguiçoso e folgado eu sou. (risos de novo) PS: espero que saiba na verdade. Bom, e pra finalizar e não me incomodar, agradeço a minha namorada que não me deixa sair pra ficar escrevendo isso. (Se não cito ela, acho que ela me bate). Bom, acho que não esqueci de ninguém, e se esqueci é porque não merecia estar aqui mesmo.

É isso!!!

Obrigado a todos!

So long and thanks for all the fish. – TODOS GOLFINHOS DA TERRA

RESUMO

AZEVEDO, Ariel Furtado. **Um Módulo de Gerenciamento e Sensoreamento de Recursos Computacionais de um Ambiente em Nuvem com Lógica Fuzzy**. 2015. 33 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2015.

Esse projeto de graduação tem como objetivo criar e analisar uma técnica de sensoreamento de recursos computacionais utilizados por um servidor no qual está hospedado um ambiente em nuvem, com a ajuda de lógica *fuzzy* para a tomada de decisões.

Assim com base na lógica *fuzzy* será construído o módulo de tomada de decisões que analisará os recursos computacionais de um ambiente com carga de trabalho e gerará resultados que serão analisados posteriormente

Palavras-chave: Computação em Nuvem, Módulo Fuzzy, Logica Fuzzy, Openstack, Gerenciamento.

ABSTRACT

AZEVEDO, Ariel Furtado. **A Computational Resource Management and Sensing Module of a Cloud Environment with Fuzzy Logic**. 2015. 33 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2015.

This graduation project aims to create and analyze a sensing technique of computational resources used by a server that is hosted a cloud environment with fuzzy logic help in decision making .

So based on fuzzy logic will be built the decision-making module that will analyze the computational resources of a workload with environment and generate results that will be analyzed

Keywords: Cloud Computing, Fuzzy Logic, Management, Resources.

LISTA DE FIGURAS

Figura 1	Exemplo de uma Nuvem	13
Figura 2	Exemplo de máquina física e virtual	17
Figura 3	Modelos de Implantação	18
Figura 4	Serviços oferecidos pela Computação em Nuvem e alguns exemplos	20
Figura 5	Atores e seu Papéis na Computação em Nuvem	20
Figura 6	Arquitetura do <i>Openstack</i> (OPENSTACK, 2015)	22
Figura 7	Sistema Fuzzy	28

LISTA DE TABELAS

Tabela 1	Nome da Tabela	31
----------	--------------------------	----

LISTA DE ABREVIATURAS E SIGLAS

TI	Tecnologia da Informação
SLA	<i>Services Level Agreements</i>
QoS	<i>Quality of Service</i>
VMM	<i>Virtual Machine Monitor</i>
IaaS	Infraestrutura como Serviço
PaaS	Plataforma como Serviço
SaaS	<i>Software</i> como Serviço
SGBD	Sistema Gerenciador de Banco de Dados
API	<i>Application Programming Interface</i>
LF	Lógica <i>Fuzzy</i>
IP	<i>Internet Protocol</i>

SUMÁRIO

1	INTRODUÇÃO	11
2	COMPUTAÇÃO EM NUVEM	12
2.1	Fundamentos	12
2.2	Características	14
2.2.1	Características Essenciais	15
2.2.2	Virtualização	16
2.2.3	Modelos de Implantação	17
2.2.4	Serviços	19
2.3	Desafios	20
2.3.1	Segurança	21
2.3.2	Gerenciamento de Dados	21
2.3.3	Autonomia	21
2.3.4	Gerenciamento de Recursos	22
2.4	Openstack	22
2.4.1	Introduzindo o <i>OpenStack</i>	23
2.4.2	Como o <i>OpenStack</i> é usado em um ambiente em nuvem?	23
2.4.3	Componentes do <i>OpenStack</i>	23
2.5	Considerações Sobre o Capítulo	24
3	LOGICA FUZZY	25
3.1	Fundamentos	25
3.1.1	Características	26
3.1.2	Sistema <i>Fuzzy</i>	27
3.2	Realizações Futuras	28
3.3	Considerações Sobre o Capítulo	29
4	TRABALHOS RELACIONADOS	30
4.1	Infraestrutura como Serviço na Computação em Nuvem	30
4.2	<i>Provising, Resource Allocation, and DVFS in Green Clouds</i>	30
4.3	<i>Consolidation of VMs to Improve Energy Efficiency in Cloud Computing Environments</i>	30
4.3.1	Uma subseção	30
	REFERÊNCIAS	32

1 INTRODUÇÃO

A computação em Nuvem tem tido um crescimento notável nos últimos anos. Cada vez mais esse paradigma tem se estabelecido em nossas vidas facilitando nosso dia-a-dia, tanto para profissionais da área, como para pessoas totalmente desligadas. Dificilmente uma pessoa que use um computador não utilize mesmo que indiretamente algum serviço da nuvem, e com a popularização da *internet* e dos computadores, bem como avanço tecnológico é bem raro uma pessoa que não utilize um computador, ou até um telefone móvel que tenha acesso à rede. Isso faz com que a computação em nuvem seja cada vez mais tendência na tecnologia e garante cada vez mais demanda para esse serviço.

Mas mesmo a nuvem precisa estar alocada fisicamente em algum lugar, e muito provavelmente um *datacenter* responsável pela disponibilização dos serviços em nuvem. Tendo isso em mente, pensa-se então uma forma de como otimizar a utilização dos recursos físicos disponíveis por esse *datacenter* na gerência desse ambiente fazendo o sensoreamento do consumo de recursos.

Esse documento está organizado da seguinte forma: No Capítulo 1 fica a Introdução do trabalho, onde é feita uma breve abordagem do problema e mostra a estrutura do documento, no Capítulo 2 fala-se sobre a Computação em Nuvem e seus conceitos e fundamentos, bem como características e desafios. No Capítulo 3 trata-se sobre a Lógica *Fuzzy* em si, e nos seus fundamentos, também é falado sobre um sistema *fuzzy* simples e suas abordagens.

2 COMPUTAÇÃO EM NUVEM

A *Cloud Computing* ou Computação em Nuvem é uma área da computação que vem crescendo diariamente e tomando um grande espaço nas nossas vidas, de forma que pode-se dizer que estamos constantemente em contato de alguma forma com a "nuvem". Esse conceito um tanto quanto recente, nos proporciona facilidades de vários modos diferentes como plataformas e *softwares* sendo disponibilizados como serviços, bem como infraestrutura, reduzindo custos tanto em espaço como recursos computacionais. Muitas empresas já utilizam esse paradigma para distribuir seus *softwares*, e acredita-se que ocorrerá uma migração constante para esse ambiente. Esse capítulo abordará sobre os principais conceitos e fundamentos da *Cloud Computing*, bem como suas características e áreas de aplicação.

2.1 Fundamentos

A premissa central da Computação em Nuvem é fornecer recursos computacionais complexos na forma de serviços que podem ser acessados através de uma rede, através de um modelo de negócios onde provedores de serviços em nuvem podem cobrar dos consumidores para acessar seus serviços (BUYAYA et al., 2009).

Porém o conceito de entregar recursos computacionais como um serviço não é necessariamente uma ideia nova para a área de TI. Em 1961, John McCarthy (PARKHILL, 1966), sugeriu que a computação pudesse algum dia ser organizada como uma utilidade pública. Essa ideia era também compartilhada por um dos principais cientistas da ARPANET, Leonard Kleinrock, que idealizava em 1969 que isso poderia acontecer em um futuro onde houvesse o crescimento das redes de computadores e sua evolução.

Mell (MELL; GRANCE, 2011) dizia que a Computação em Nuvem poderia ser entendida como um modelo que visa permitir o acesso de forma conveniente, ubíqua e sob demanda, a um conjunto de recursos computacionais configuráveis, tais como: redes, servidores, armazenamento, aplicativos e serviços, que poderiam ser rapidamente provisionados e liberados com o mínimo de esforço de gestão ou de interação

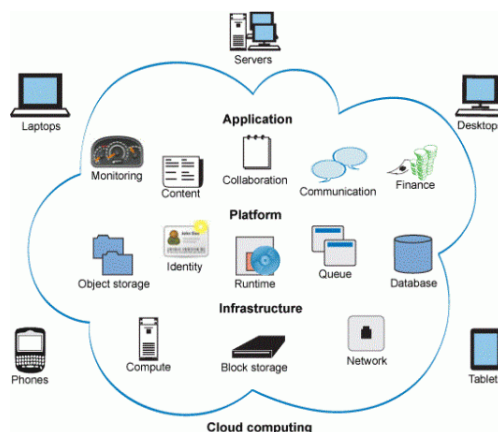


Figura 1: Exemplo de uma Nuvem

com o provedor de serviços. A Figura 1 mostra um exemplo geral de Nuvem de forma que é possível notar brevemente como os aspectos citados se organizam para compor esse tipo de ambiente. Ainda nessa figura podemos visualizar a abstração de uma nuvem caracterizada pelos principais tipos de serviços que são fornecidos pelo ambiente. (i) Aplicação, (ii) Plataforma, (iii) infraestrutura e alguns exemplos de recursos fornecidos por cada camada de abstração.

Do ponto de vista do uso de recursos de *hardware*, (ARMBRUST et al., 2010; FOX; GRIFFITH et al., 2009) destacam três pontos centrais que colaboraram com a adoção da Computação em Nuvem:

1. A ilusão de recursos de computação infinitos e disponíveis sob demanda, eliminando assim a necessidade dos usuários realizarem planejamentos de longo prazo quanto ao provisionamento de recursos.
2. A eliminação de um compromisso antecipado quanto ao uso de recursos e consequente alocação dos mesmos por parte dos usuário, permitindo assim que as empresas possam começar com um investimento menor e aumentar os recursos de *hardware* apenas quando há um aumento nas suas necessidades.
3. A capacidade de pagar pelo uso de recursos de computação com base em períodos de curto prazo (horas/dias), utilizando-os conforme necessário e liberá-los quando os mesmos não forem mais úteis.

A redução do custo operacional é um dos aspectos centrais para adoção de ambientes em nuvem. Essa redução é notada principalmente nos custos relacionados a energia elétrica, largura de banda de rede, na operação e administração de serviços ofertados por grandes *datacenters* (ARMBRUST et al., 2010; FOX; GRIFFITH et al., 2009). Como os recursos podem ser rapidamente alocados e desalocados conforme a demanda, não é necessário dimensioná-los para atender a maior carga de trabalho

a longo prazo, fornecendo assim a noção de elasticidade característica desse paradigma. Da mesma forma, essa flexibilidade pode garantir uma alta escalabilidade, pois os provedores de infraestrutura podem assim disponibilizar uma grande quantidade de recursos a partir do seu conjunto de servidores e torná-los acessíveis, possibilitando expandir os serviços para grandes escalas, mesmo com o rápido aumento na demanda. No entanto para que se atinja esse objetivo e amplie ainda mais a capacidade de redução de custos, é necessário um gerenciamento eficiente dos recursos que fazem parte da infraestrutura da nuvem (BUYA et al., 2009). Pensando em um forma de economia, a Computação em Nuvem tem então causado um grande impacto na área de TI, pois visando na otimização e redução de custos e riscos respectivamente, passou-se a pensar em um ambiente onde as empresas sejam "*datacenter-less*", ou seja, não tenham *datacenters* físicos, e sim utilizem de empresas que forneçam esses produtos e serviços envolvendo tecnologia da informação por meio de nuvem, sem que as primeiras tenham que lidar com os custos associados a administração e operação dos *datacenters* (ARMBRUST et al., 2010), e com isso focando somente na prestação de seu serviço ou desenvolvimento de seu produto.

Outros aspectos também têm contribuído para a adoção da Computação em Nuvem e sua popularização enquanto paradigma computacional. É possível destacar pelo menos dois aspectos principais: **(i)** aumento na capacidade de processamento e armazenamento de *hardware*, bem como sua rápida diminuição do custo, incentivando sua adoção massiva na construção de *datacenters*; **(ii)** a adoção generalizada de serviços de computação e aplicações da *Web*, provocando crescimento exponencial da quantidade de dados disponíveis, especialmente os relacionados com o conteúdo da *Internet*, envolvendo redes sociais, comércio eletrônico, computação móvel (ARMBRUST et al., 2010; FOX; GRIFFITH et al., 2009; FOSTER et al., 2008).

2.2 Características

De uma forma geral, podemos imaginar a Computação em Nuvem como uma forma de sistema paralelo e distribuído, que consiste em um conjunto de computadores interligados e virtualizados, que, dinamicamente, são provisionados e apresentados como um ou mais recursos de computação unificada. O acesso e utilização desses recursos são regidos através do *Services Level Agreements* (SLA), ou Acordos de Nível de Serviço, estabelecidos através de negociação entre o prestador de serviço e os consumidores, e que determinam o nível de qualidade aceitável para um serviço disponibilizado pela Computação em Nuvem (BUYA et al., 2009).

2.2.1 Características Essenciais

No entanto a Computação em Nuvem possui algumas características essenciais que a definem e fazem um modelo distinto de outros paradigmas. De acordo com Mell (MELL; GRANCE, 2011), existem cinco características essenciais que distinguem a Computação em Nuvem.

2.2.1.1 Self-Service sob demanda

O usuário pode adquirir unilateralmente recurso computacional, como tempo de processamento no servidor ou armazenamento na rede na medida em que necessite e sem precisar de interação humana com os provedores de cada serviço. O *hardware* e o *software* dentro de uma nuvem podem ser automaticamente reconfigurados, orquestrados e estas modificações são apresentadas de forma transparente para os usuários, que possuem perfis diferentes e assim podem personalizar os seus ambientes computacionais, por exemplo, instalação de *software* e configuração de rede para a definição de determinados privilégios.

2.2.1.2 Amplo Acesso

Recursos são disponibilizados por meio da rede e acessados através de mecanismos padronizados que possibilitam o uso por plataformas *thin* ou *thin client*, tais como celulares e *laptops*. A interface de acesso à nuvem não obriga os usuários a mudar suas condições e ambientes de trabalho, como por exemplo, linguagens de programação e sistema operacional. Já os *softwares* clientes instalados localmente para o acesso à nuvem são leves, como um navegador de *Internet*.

2.2.1.3 Pooling de recursos

Os recursos computacionais do provedor são organizados em um pool para servir múltiplos usuários usando um modelo *multitenant*, com diferentes recursos físicos e virtuais, dinamicamente atribuídos e ajustados de acordo com a demanda dos usuários. Estes usuários não precisam ter conhecimento da localização física dos recursos computacionais, podendo somente especificar a localização em um nível mais alto de abstração, tais como país, estado ou *datacenter*.

2.2.1.4 Elasticidade Rápida

Recursos podem ser adquiridos de forma rápida e elástica, em alguns casos automaticamente, caso haja a necessidade de escalar com o aumento da demanda, e liberados, na retração dessa demanda. Para os usuários, os recursos disponíveis para uso parecem ser ilimitados e podem ser adquiridos em qualquer quantidade e a qualquer momento. A virtualização ajuda na característica de elasticidade rápida na computação nuvem, criando várias instâncias de recursos requisitados utilizando um

único recurso real (ABOULNAGA A. SALEM K., 2009). Além disso, a virtualização é uma maneira de abstrair características físicas de uma plataforma computacional dos usuários, exibindo outro *hardware* virtual e emulando um ou mais ambientes que podem ser independentes ou não.

2.2.1.5 Modelos de Serviços

Sistemas em nuvem automaticamente controlam e otimizam o uso de recursos por meio de uma capacidade de medição. A automação é realizada em algum nível de abstração apropriado para o tipo de serviço, tais como armazenamento, processamento, largura de banda e contas de usuário ativas. O uso de recursos pode ser monitorado e controlado, possibilitando transparência para o provedor e o usuário do serviço utilizado. Para garantir o QoS (*Quality of Service*) ou Qualidade de Serviço, pode-se utilizar a abordagem baseada em níveis de acordo de serviço SLA. Ele fornece informações sobre os níveis de disponibilidade, funcionalidade, desempenho ou outros atributos do serviço como o faturamento e até mesmo penalidades em caso de violação destes níveis.

2.2.2 Virtualização

Outra característica importante para o paradigma de Computação em Nuvem é o uso da virtualização, pois ela oferece a abstração necessária para que os recursos computacionais físicos (processamento, armazenamento, recursos de rede) possam ser unificados como um conjunto de recursos mais abstratos, permitindo que outros serviços possam ser construídos sobre estes (FOSTER et al., 2008). Através dela é possível compartilhar a mesma máquina física com múltiplos sistemas operacionais e/ou aplicações de usuários finais, mantendo garantias ao desempenho (ADDIS et al., 2010) e isolamento entre as mesmas. Tal propriedade garante certo nível de encapsulamento e flexibilidade que permite que tais máquinas possam ser manipuladas e tenham suas aplicações migradas entre servidores com maior facilidade do que se isto ocorresse usando-se somente máquinas físicas, garantindo assim melhor segurança, gerenciamento e isolamento. Na Figura 2 vemos a diferença entre uma máquina física e o funcionamento de um ambiente virtualizado, onde a camada inferior representa a camada física, a camada intermediária representa o sistema operacional que é executado no servidor físico e o *software* de virtualização ou *Virtual Machine Monitor* VMM, responsável por gerenciar o acesso ao *hardware* subjacente e gerenciar e dar suporte a próxima camada, que representa as máquinas virtuais de fato, onde cada máquina virtual possui um sistema operacional independente e executa suas próprias aplicações.

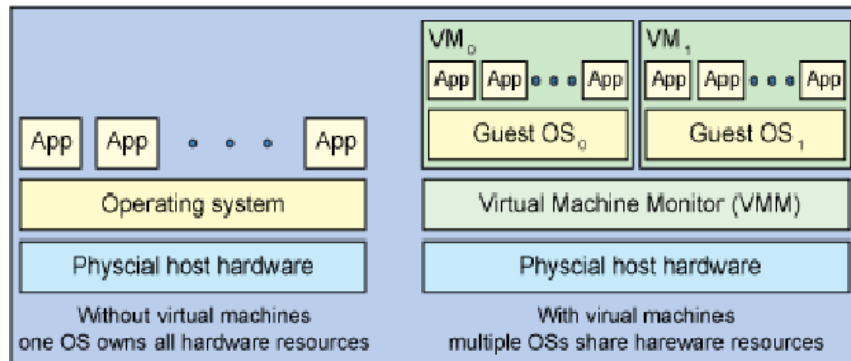


Figura 2: Exemplo de máquina física e virtual

2.2.3 Modelos de Implantação

Tratando-se do acesso e disponibilidade de ambientes de computação em nuvem, tem-se diferentes tipos de modelos de implantação. A restrição ou abertura de acesso depende do processo de negócios, do tipo de informação e do nível de visão desejado. Pode-se perceber que certas empresas não desejam que todos os usuários possam acessar e utilizar determinados recursos no seu ambiente de computação em nuvem. Assim sendo, pensa-se então em criar ambientes mais restritos, onde somente usuários devidamente autorizados possam utilizar os serviços providos. Os modelos de implantação da Computação em Nuvem podem ser: Público, Privado, Híbrido e Comunitário (NIST, 2009). A Figura 3 mostra os respectivos modelos de implantação de nuvens.

- *Privado*: No modelo de implantação privado, a infra-estrutura de nuvem é utilizada exclusivamente para uma organização, sendo esta nuvem local ou remota e administrada pela própria empresa ou por terceiros. Neste modelo de implantação são empregadas políticas de acesso aos serviços. As técnicas utilizadas para prover tais características podem ser em nível de gerenciamento de redes, configurações dos provedores de serviços e a utilização de tecnologias de autenticação e autorização. Um exemplo deste modelo seria o cenário de uma universidade e seus departamentos. A universidade pode estar interessada em disponibilizar serviços para seus departamentos e outros órgãos desta instituição não devem ter acesso a esses serviços.
- *Público*: No modelo de implantação público, a infra-estrutura de nuvens é disponibilizada para o público em geral, sendo acessado por qualquer usuário que conheça a localização do serviço. Neste modelo de implantação não podem ser aplicadas restrições de acesso quanto ao gerenciamento de redes, e menos ainda, aplicar técnicas de autenticação e autorização.
- *Comunitário*: No modelo de implantação comunitário ocorre o compartilhamento

de uma nuvem por diversas empresas, sendo a nuvem suportada por uma comunidade específica que partilhou seus interesses, tais como a missão, os requisitos de segurança, política e considerações sobre flexibilidade. Este tipo modelo de implantação pode existir localmente ou remotamente e pode ser administrado por alguma empresa da comunidade ou por terceiros.

- *Híbrido*: No modelo de implantação híbrido, existe uma composição de duas ou mais nuvens, que podem ser privadas, comunitárias ou públicas e que permanecem como entidades únicas e ligadas por uma tecnologia padronizada ou proprietária que permite a portabilidade de dados e aplicações permitindo por exemplo um balanceamento de carga entre nuvens distintas (WIND, 2011).

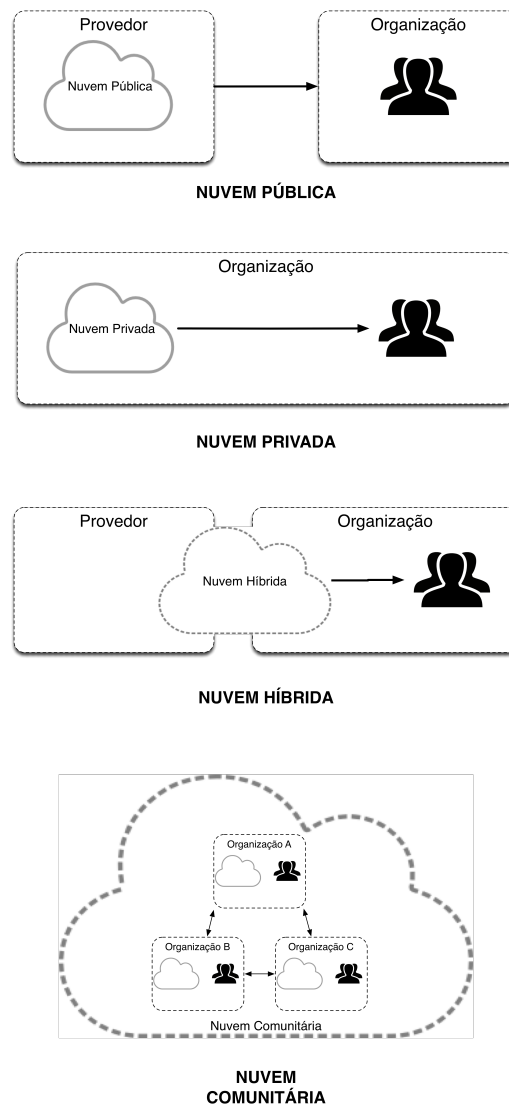


Figura 3: Modelos de Implantação

2.2.4 Serviços

Outra das características do paradigma de Computação em Nuvem, é os serviços disponibilizados e seus tipos. Assim a classificação quanto ao tipo de serviço ofertado é feita de acordo com o nível de abstração do serviço fornecido. Como é mostrado na Figura 4, existem três camadas de abstração nos serviços que podem ser disponibilizados pelo ambiente na nuvem com diferentes tipos de serviços por camada, e as camadas e exemplos são as seguintes:

- **Infraestrutura como Serviço (IaaS):** nesse nível de serviço, o provedor oferece uma infraestrutura virtualizada de servidores, armazenamento e rede, que pode ser fornecida sob demanda e cobrada conforme seu uso. Nesse tipo de serviço, o cliente pode criar uma máquina virtual e instalar sobre a mesma, uma pilha completa de software, incluindo nisso, o sistema operacional. O acesso ao servidor ou servidores, e aos serviços fornecidos por esse, acontece normalmente através da rede. Esse formato de serviço garante uma maior flexibilidade, no entanto, exige do cliente maior conhecimento para que esse possa usá-lo. Um exemplo para este tipo de serviço é o fornecido pela Amazon através do *Amazon's Elastic Compute Cloud (EC2)* (AMAZON, 2013).
- **Plataforma como Serviço (PaaS):** oferece um ambiente de alto nível integrado para construir, testar e implementar aplicativos personalizados. A vantagem imediata desse tipo de serviço é a facilidade de uso e a possibilidade dos clientes em manter o foco na criação e manutenção de seus próprios serviços. A manutenção do *hardware* (virtual) e software é responsabilidade do provedor de nuvem. Uma restrição nesse nível, deve se ao fato de que os serviços dos clientes precisam ser compatíveis com a nuvem. Como exemplo, pode-se citar o *Google Apps* (GOOGLE, 2013).
- **Software como Serviço (SaaS):** fornece *Software* de propósito especial que é acessível remotamente pelos consumidores através da Internet com um modelo de cobrança baseada em uso. A principal característica desse tipo de serviço é o fato dos fornecedores ficarem responsáveis por gerenciar toda a plataforma virtual e as aplicações sobre esta. Como toda a manutenção é feita pelo fornecedor, os clientes só precisam se concentrar na avaliação de serviços de *software* e usá-los. Nesse nível de serviço não é possível negociar atualizações do *software*. Serviços como Dropbox (DROPBOX, 2012), Facebook (FACEBOOK, 2012), Gmail (GMAIL, 2012), Salesforce (SALESFORCE, 2012) são exemplos desse nível de serviço.

Já a Figura 5 mostra um pouco melhor como essas camadas de abstração e seus respectivos serviços são fornecidos e quem faz uso de seus benefícios. O Provedor

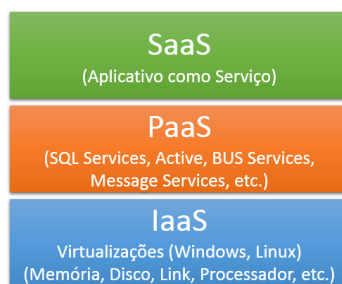


Figura 4: Serviços oferecidos pela Computação em Nuvem e alguns exemplos

é responsável por disponibilizar, gerenciar e monitorar toda a estrutura para a solução em nuvem, deixando os desenvolvedores e usuários finais sem esses tipos de responsabilidades. Para isso, o provedor fornece serviços nas três camadas da nuvem. Já os desenvolvedores utilizam os recursos fornecidos e provêem serviços para os usuários finais. Esta organização em papéis ajuda a definir os atores e seus diferentes interesses. Os atores podem assumir vários papéis ao mesmo tempo de acordo com os interesses, sendo que apenas o provedor fornece suporte a todas as camadas.

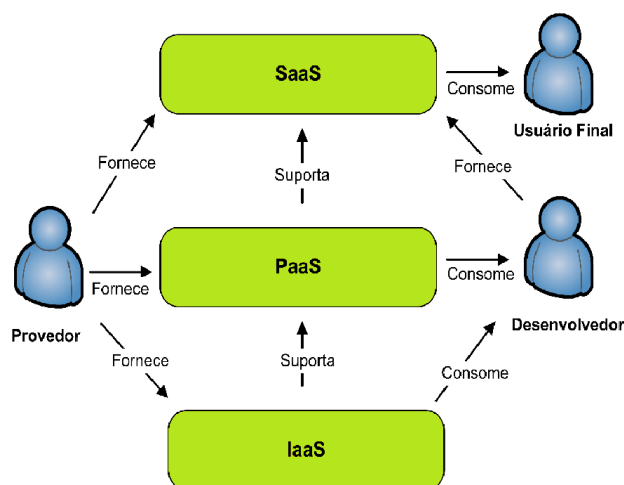


Figura 5: Atores e seu Papéis na Computação em Nuvem

2.3 Desafios

Mesmo hoje, com o avanço da tecnologia bem como o avanço da Computação em Nuvem, ambientes assim ainda apresentam uma série de desafios a serem superados, agora será abordado sobre alguns desses desafios.

2.3.1 Segurança

Por ser um modelo que utiliza da *internet* para disponibilizar seus serviços, segurança torna-se então um aspecto de fundamental importância e deve-se ter formas de impedir o acesso não autorizado a informações e garantir que os dados sensíveis permaneçam privados, pois estes serão processados fora da empresa. Questões de segurança devem ser consideradas para prover a autenticidade, confiabilidade e integridade. No que diz respeito à confiabilidade e responsabilidade, o provedor deve fornecer recursos confiáveis, especialmente se a computação a ser realizada é crítica e existindo uma clara delimitação de responsabilidade.

2.3.2 Gerenciamento de Dados

O gerenciamento de dados é considerado um ponto crítico no contexto de computação em nuvem. Os SGBDs (Sistema gerenciador de banco de dados) relacionais não possuem escalabilidade quando milhares de sítios são considerados (WEI Z.; CHI, 2009). Assim, aspectos de armazenamento de dados, processamento de consultas e controle transacional têm sido flexibilizados por algumas abordagens para garantir a escalabilidade, mas ainda não existem soluções que combinem estes aspectos de forma a melhorar o desempenho sem comprometer a consistência dos dados. Um aspecto importante é o *trade-off* entre funcionalidades e custos operacionais enfrentados pelos provedores de serviços. Os serviços em nuvem para dados oferecem APIs *Application Programming Interface* ou Interface de Programação de Aplicações mais restritas que os SGBDs relacionais, com uma linguagem minimalista de consulta e garantia de consistência limitada. Porém isso exige esforço de programação dos desenvolvedores, mas permite aos provedores construir serviços mais previsíveis e oferecerem SLA. De acordo com Armbrust (ARMBRUST et al., 2010), a criação de um sistema de armazenamento que combine com diversos aspectos de computação em nuvem, de forma a aumentar a escalabilidade, disponibilidade e consistência dos dados é um problema de pesquisa em aberto.

2.3.3 Autonomia

Na computação em nuvem, vários modelos evoluíram rapidamente para aproveitar as tecnologias de *software*, plataformas de programação, armazenamento de dados e infraestrutura de *hardware* como serviços. Enquanto estes modelos se referem ao núcleo dos serviços de computação em nuvem, suas inter-relações têm sido ambíguas e a viabilidade de sua interoperabilidade é questionável. Além disso, cada serviço da nuvem tem interfaces e protocolos diferentes. Como os diversos serviços estão dispersos no ambiente e possuem características distintas, desenvolver técnicas eficazes e eficientes para descrever, descobrir e compor estes serviços torna-se um desafio interessante.

2.3.4 Gerenciamento de Recursos

Pelo fato de a computação em nuvem prover recursos computacionais de forma transparente para seus usuários, o provedor por ter que se preocupar com a parte física da nuvem, deve então ter como gerenciar os recursos utilizados pelo ambiente, como por exemplo energia, processamento, armazenamento, etc... Com os problemas ecológicos e ambientais atualmente, a parte de gerencia de energia tem se tornado cada vez mais um desafio para os provedores desse serviço, e por isso surgem campos de pesquisa para as *Green Clouds*, ou Nuvens Verdes, termo que refêrencia a computação em nuvem de uma forma limpa e ecologicamente consciente, onde se pensa de uma forma de otimizar todo e qualquer consumo de recurso computacional dos *datacenters*.

2.4 Openstack

O *openstack* é um conjunto de tecnologias capaz de controlar múltiplas infraestruturas de computação, armazenamento e recursos de rede, nesta sessão será explicado um pouco mais detalhado como funciona essa ferramenta.

A Figura 6 mostra como funciona a arquitetura do *openstack*, a qual podemos ver que há um *hardware* padrão, onde é de fato executado o ambiente, este por sua vez tem um ambiente de serviços compartilhados, que são Computação, Rede, e Armazenamento, gerenciados todos por um painel de ferramentas, (*DashBoard*), e em seguida fica a camada de aplicação, que pode estar qualquer um desses tipos de serviço.

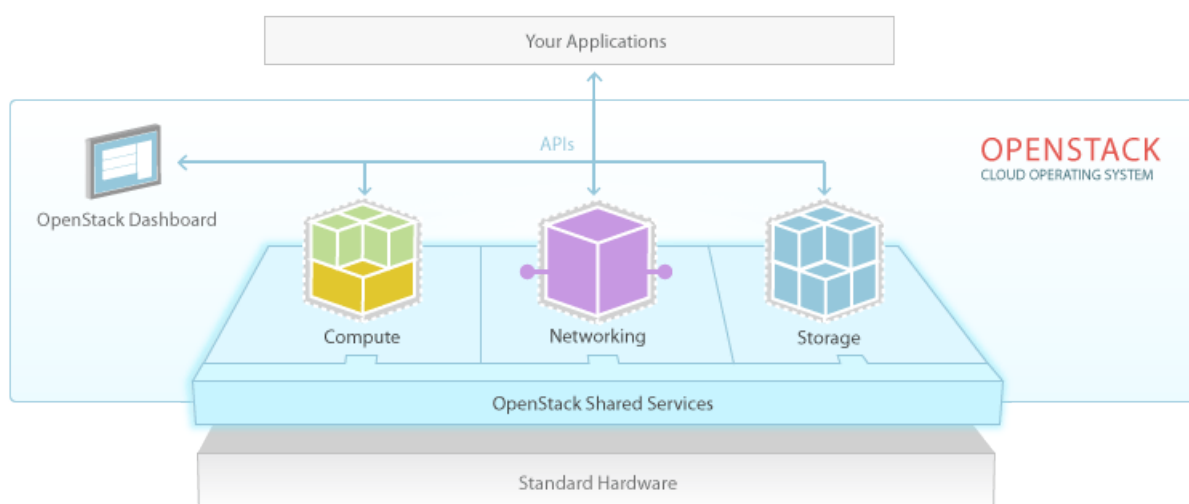


Figura 6: Arquitetura do *Openstack* (OPENSTACK, 2015)

2.4.1 Introduzindo o *OpenStack*

O *OpenStack* permite a seus usuários desenvolver máquinas virtuais e outras instâncias que operam diferentes tarefas para gerenciar um ambiente em nuvem. Ele faz com que escalonamento horizontal se torne mais fácil de ser feito, o que significa que tarefas que se beneficiam de serem executadas de forma concorrente possam servir de forma mais fácil mais ou menos usuários na nuvem somente subindo mais instâncias. (e.g : uma aplicação móvel que precisa se comunicar com um servidor remoto pode ter como dividir o trabalho de comunicação com cada usuário de através de muitas instâncias diferentes, todos se comunicando entre si, mas escalonando rapidamente e facilmente conforme a aplicação ganha mais usuários.)

2.4.2 Como o *OpenStack* é usado em um ambiente em nuvem?

A nuvem serve basicamente para prover serviços computacionais para usuários finais em um ambiente remoto, onde o *software* atual executa como um serviço em servidores confiáveis e escalonáveis ao invés de executar no computador de cada usuário. Basicamente *OpenStack* provê IaaS, o que significa que ele provê infraestrutura como serviço, o que significa que ele faz ser fácil para o usuário adicionar rapidamente uma nova instância da nuvem sobre qual os outros componentes da nuvem podem rodar. Tipicamente a infraestrutura executa uma plataforma sobre qual o desenvolvedor pode criar aplicações, as quais são entregadas para o usuário final.

2.4.3 Componentes do *OpenStack*

O *OpenStack* é feito de diferentes partes móveis. Por causa de sua natureza "livre", qualquer um pode adicionar componentes a ele para ajudar a satisfazer suas necessidades. Mas a comunidade do *OpenStack* tem colaborativamente identificado componentes chave que são parte de um núcleo da ferramenta, as quais são distribuídas como uma parte de qualquer sistema *OpenStack* e oficialmente mantido por sua comunidade. Esses componentes são:

- *Openstack Compute*: o Sistema Operacional em Nuvem *Openstack* permite às empresas e aos prestadores de serviço oferecer recursos computacionais sob demanda graças ao provisionamento e gerenciamento de grandes redes de máquinas virtuais. Estes recursos são acessíveis através de APIs para desenvolvedores de aplicativos em nuvem e por meio de interfaces *web* para administradores e usuários.
- *Openstack Storage*: Além da tecnologia tradicional de armazenamento de classe empresarial, muitas organizações têm agora uma variedade de necessidades de armazenamento com diferentes requisitos de desempenho e preço. *OpenStack* tem base para *Object Storage* e *Block Storage*, com muitas opções de

implantação para cada, dependendo do caso de uso. O primeiro é ideal para o armazenamento de baixo custo, *scale-out*. Ele fornece uma plataforma de armazenamento acessível, totalmente distribuída, que pode ser integrado diretamente em aplicativos ou utilizados para *backup*, arquivamento e retenção de dados. Já *Block Storage* permite que os dispositivos de bloco sejam expostos e conectados para calcular instâncias de armazenamento expandido, melhor desempenho e integração com plataformas de armazenamento corporativo, tais como *NetApp*, *Nexenta* e *SolidFire*.

- *Openstack Networking*: As redes de *datacenters* atuais contém mais dispositivos conectados do que mesmo servidores, equipamento de redes, sistemas de armazenamento e aplicações de segurança – muitos dos quais são comumente divididos em máquinas e redes virtuais. O número de endereços IPs, configurações de rotas e até mesmo regras de segurança podem crescer até a casa dos milhões facilmente. Técnicas tradicionais de gerenciamento de redes falham rapidamente ao tentar prover uma verdadeira escalabilidade a um sistema desse porte. Ao mesmo tempo, os clientes esperam maior facilidade de acesso e mais velocidade e garantia de qualidade. *Openstack Networking* é um sistema escalonável e plugável, orientado a API, para gerenciamento de redes e endereços IPs. Como outros aspectos de um sistema operacional em nuvem ele pode ser usado para aumentar o valor de *datacenters* ativos.

2.5 Considerações Sobre o Capítulo

Neste capítulo foi abordado os principais fundamentos e conceitos sobre o paradigma de Computação em Nuvem, fazendo uma introdução sobre o tema e mostrando aspectos relacionados com a computação em nuvem. Além disso alguns dos desafios foram discutidos, e um deles, Gerenciamento de Recursos, Subseção 2.3.4 é o tema abordado nessa pesquisa. Ainda também é feita uma introdução ao *OpenStack*, que será a ferramenta utilizada para desenvolvimento do ambiente em Nuvem e realização dos testes. No capítulo a seguir, veremos alguns fundamentos e conceitos de Lógica *Fuzzy*, e como essa técnica pode ser útil e relevante na recente pesquisa, bem como alguns trabalhos relacionados onde a lógica *fuzzy* é utilizada para técnicas semelhantes as que serão utilizadas ao longo desse projeto.

3 LOGICA *FUZZY*

Nesse capítulo serão abordados os principais fundamentos sobre Lógica *Fuzzy*, LF. Após mostrar alguns conceitos sobre o assunto, procura-se fazer uma relação com o tema da pesquisa, mostrando assim que essa técnica pode dar um suporte melhor na tomada de decisões para um gerenciamento e uma otimização melhores dos recursos utilizados pela unidade física da Nuvem.

3.1 Fundamentos

A lógica binária é a base de todo e qualquer cálculo computacional, sendo essas operações básicas que constituem todo o poderio dos computadores da atualidade. Qualquer operação, por mais complexa que pareça é traduzida internamente pelo processador para estas operações. Esta lógica trabalha simplesmente com verdadeiro e falso, porém nem tudo na computação se resolve de forma ótima, ou mesmo satisfatória simplesmente com verdadeiro ou falso. Assim sendo então, pensou-se em uma forma de trabalhar com a lógica, não de forma binária, mas de forma com que exista níveis de certeza para cada sentença, passando assim a ter um grau de pertinência, e assim nasceu a Lógica *Fuzzy*.

O Conceito de Conjunto *Fuzzy* foi introduzido em 1965, por Lofti A. Zadeh (ZADEH, 1965), professor de Ciências da Computação da Universidade da Califórnia. A ele, é atribuído o reconhecimento como grande colaborador do controle moderno. Na época, Zadeh observou que os recursos tecnológicos disponíveis eram incapazes de automatizar as atividades relacionadas a problemas de natureza industrial, biológica ou química, que compreendessem situações ambíguas, não passíveis de processamento pela lógica binária. Ainda assim, somente em 1974 conseguiu-se usar conjuntos *Fuzzy* para controlar uma máquina a vapor com tipos distintos de controladores. (MAMDANI; ASSILIAN, 1975).

Após essa conquista, surgiu então um novo campo para a Computação, a Lógica *Fuzzy*. Seus sistemas foram amplamente utilizados para muitas aplicações, como no controle de operações de um forno de cimento, controladores de plantas nucleares,

refinarias, processos biológicos e químicos, controladores de temperatura, máquina diesel, tratamento de água e sistema de operação automática de trens.

3.1.1 Características

A lógica *fuzzy* é a lógica baseada na teoria dos conjuntos *fuzzy*. Ela difere dos sistemas lógicos tradicionais em suas características e seus detalhes. Nesta lógica, o raciocínio exato corresponde a um caso limite do raciocínio aproximado, sendo interpretado como um processo de composição de relações nebulosas. Na lógica *fuzzy*, o valor verdade de uma proposição pode ser um subconjunto *fuzzy* de qualquer conjunto parcialmente ordenado, ao contrário dos sistemas lógicos binários, onde o valor verdade só pode assumir dois valores: verdadeiro (1) ou falso (0). Nos sistemas lógicos multi-valores, o valor verdade de uma proposição pode ser ou um elemento de um conjunto finito, num intervalo, ou uma álgebra booleana. Na lógica nebulosa, os valores verdade são expressos linguisticamente, (e.g. : verdade, muito verdade, não verdade, falso, muito falso, ...), onde cada termo linguístico é interpretado como um subconjunto *fuzzy* do intervalo unitário. Outra diferença é que nos sistemas lógicos binários, os predicados são exatos (e.g. : par, maior que), ao passo que na lógica *fuzzy* os predicados são nebulosos (e.g. : alto, baixo, ...). Nos sistemas lógicos clássicos, o modificador mais utilizado é a negação enquanto que na lógica *fuzzy* uma variedade de modificadores de predicados são possíveis (e.g. : muito, mais ou menos, ...). Estes modificadores são essenciais na geração de termos linguísticos (e.g. : muito alto, mais ou menos perto, etc). Nos sistemas lógicos clássicos existem somente os quantificadores existenciais (\exists) e universais (\forall). A lógica *fuzzy* admite, em adição, uma ampla variedade de quantificadores (e.g. : pouco, vários, usualmente, frequentemente, em torno de cinco, etc). A probabilidade, no contexto da lógica clássica, é um valor numérico ou um intervalo. Na lógica nebulosa existe a opção adicional de se empregar probabilidades linguísticas (e.g. : provável, altamente provável, improvável, etc), interpretados como números *fuzzy* e manipuladas pela aritmética *fuzzy* (KAUFMANN; GUPTA, 1985). Também, em contraste com a lógica modal clássica, o conceito de possibilidade é interpretado utilizando-se subconjuntos *fuzzy* no universo dos reais (ZADEH, 1988). Nas teorias de controle clássica e moderna, o primeiro passo para implementar o controle de um processo, é derivar o modelo matemático que descreve o processo. O procedimento, requer que se conheça detalhadamente o processo a ser controlado, o que nem sempre é factível se o processo é muito complicado. As teorias de controle existentes se aplicam a uma grande variedade de sistemas, onde o processo é bem definido. Várias técnicas, tais como para controle linear multivariável, estimação de estado a partir de medidas ruidosas, controle ótimo, sistemas lineares estocásticos, além de certas classes de problemas não-lineares determinísticos (HOLTZMAN, 1970), foram desenvolvidas e aplicadas com sucesso em um grande

número de problemas bem postulados. Entretanto, todas estas técnicas não são capazes de resolver problemas reais cuja modelagem matemática é impraticável. Por exemplo, em diversas situações um volume considerável de informações essenciais só é conhecido a priori de forma qualitativa. Do mesmo modo, critérios de desempenho só estão disponíveis em termos linguísticos. Este panorama leva a imprecisões e falta de exatidão que inviabilizam a maioria das teorias utilizadas até agora.

A modelagem e o controle *fuzzy* (LEE, 1990) são técnicas para se manusear informações qualitativas de uma maneira rigorosa. Tais técnicas consideram o modo como a falta de exatidão e a incerteza são descritas e, fazendo isso, tornam-se suficientemente poderosas para manipular de maneira conveniente o conhecimento. A sua utilização em sistemas de controle de processos em tempo real, em computadores ou micro-controladores, é das mais convenientes, dado que, geralmente, não envolvem nenhum problema computacional sério. A teoria de modelagem e controle *fuzzy* trata do relacionamento entre entradas e saídas, agregando vários parâmetros de processo e de controle. Isso permite a consideração de processos complexos, de modo que os sistemas de controle resultantes proporcionam um resultado mais acurado, além de um desempenho estável e robusto. A grande simplicidade de implementação de sistemas de controle *fuzzy* pode reduzir a complexidade de um projeto a um ponto em que problemas anteriormente intratáveis passam agora a ser solúveis.

3.1.2 Sistema Fuzzy

”Conforme a complexidade de um sistema aumenta, a nossa habilidade de fazer declarações precisas e significativas sobre o seu comportamento diminui, até alcançar um limite além do qual precisão e relevância tornam-se características mutuamente exclusivas.”

Conforme estudado por ZADEH (1988), a transcrição a cima, é definida como o Princípio da Incompatibilidade, e mostra a relevância em utilizar a Lógica *Fuzzy* para auxiliar na resolução de problemas que tradicionalmente são difíceis de resolver. A ideia básica de um sistema desses é a existência do que é chamado de Conjuntos *Fuzzy*. Esses conjuntos são funções que mapeiam um valor escalar em um número limitado entre 0 e 1, que indica o grau de pertinência desse valor ao conjunto.

Um sistema assim pode estimar funções de entrada e saída, por meio do uso de técnicas heurísticas, onde um especialista humano, entrevistado para ajudar a formular o conjunto de regras, pode articular associações de entrada/saída linguísticas. Assim pode-se produzir estimativas de um sistema complexo sem recorrer a modelos matemáticos. Nesse escopo, a metodologia *fuzzy* é um método de estimação de entrada e saída que considera modelos matemáticos (SHAW, 1999).

Um Sistema de Inferência considera os seguintes blocos principais, conforme mostrado na Figura 7 (SANTANNA, 2015).

- **Base de Regras**, contendo um conjunto de regras/proposições onde as variáveis antecedentes/consequentes são variáveis linguísticas e os possíveis valores de uma variável linguística são representados por um conjunto *fuzzy*.
- **Base de dados**, que define as funções de pertinência do conjunto nas regras *fuzzy*.
- **Unidade de Decisão Lógica**, realizando operações de inferência, para obter, a partir da avaliação dos níveis de compatibilidade das entradas, com as condições impostas pela base de regras, uma ação a ser realizada pelo sistema.
- **Interface de Fuzzificação**, utilizando as funções de pertinência pré-estabelecidas para mapear cada variável de entrada do sistema em graus de pertinência de algum conjunto *fuzzy* que representa a variável em questão
- **Interface de Defuzzificação**, que transforma os resultados *fuzzy* da inferência em valores de saída calculados com base na inferência obtida no módulo Unidade de Decisão Lógica, com as funções de pertinência das variáveis linguísticas da parte consequente das regras para obter uma saída não *fuzzy*. Nesta etapa as regiões resultantes são convertidas em valores de saída do sistema.

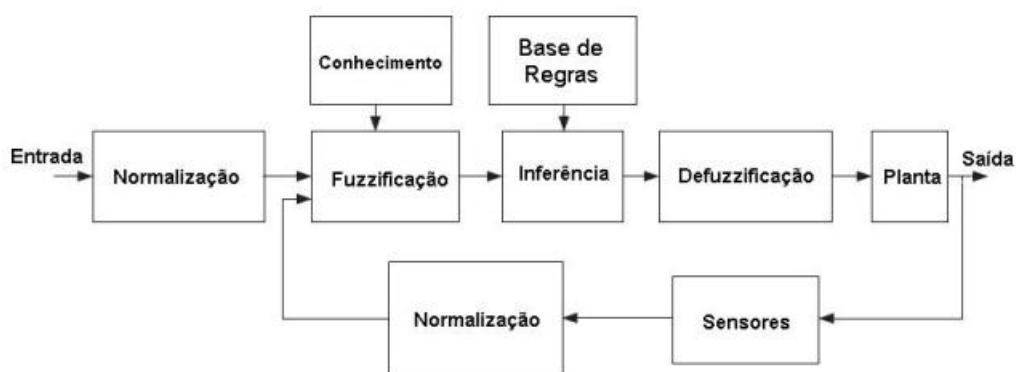


Figura 7: Sistema Fuzzy

3.2 Realizações Futuras

Após ter abordado então sobre as características e técnicas da lógica *fuzzy*, fica então a seguir definir o sistema *fuzzy* que realizará o sensoramento, tendo como entrada a quantidade de processador utilizado, a utilização da memória RAM, a utilização do disco rígido e o consumo de energia da estação relacionado com os anteriores.

Através dessa entrada, busca-se definir um sistema que apoiará na tomada de decisão no gerenciamento de instâncias físicas do ambiente. Com essas entradas,

podemos então realizar as etapas vistas na sessão anterior, e realizar um sistema *fuzzy* utilizando de fato como suporte e recebendo saídas, as quais serão analisadas e apresentadas, possibilitando um aprofundamento para possíveis trabalhos futuros.

3.3 Considerações Sobre o Capítulo

Nesse capítulo foi abordado sobre os conceitos de Lógica *Fuzzy* e suas características, bem como foi falado sobre um sistema *fuzzy* em geral e como essa técnica pode ser útil na realização desse trabalho.

Demonstrou-se também como pretende-se utilizar essa técnica na realização desse trabalho para uma pesquisa a seguir, e quais as entradas definidas para o sistema. No próximo capítulo será abordado sobre trabalhos relacionados, e quais as dificuldades desse trabalho.

4 TRABALHOS RELACIONADOS

Existem muitos trabalhos e pesquisas com foco em Computação em Nuvem, devido a sua importância e relevância hoje em dia na área de TI. Essa sessão falará sobre alguns trabalhos relacionados e sua relevância para o andamento dessa pesquisa.

4.1 Infraestrutura como Serviço na Computação em Nuvem

vilnei et al (?)

4.2 Provising, Resource Allocation, and DVFS in Green Clouds

4.3 Consolidation of VMs to Improve Energy Efficiency in Cloud Computing Environments

4.3.1 Uma subseção

[illegible]

REFERÊNCIAS

ABOULNAGA A. SALEM K., S. A. A. M. U. K. P. K. S. Deploying Database Appliances in the Cloud. **IEEE Data Eng. Bull.**, [S.l.], v.32, 2009.

ADDIS, B.; ARDAGNA, D.; PANICUCCI, B.; ZHANG, L. Autonomic Management of Cloud Service Centers with Availability Guarantees. In: CLOUD COMPUTING (CLOUD), 2010 IEEE 3RD INTERNATIONAL CONFERENCE ON, 2010. **Anais...** [S.l.: s.n.], 2010. p.220 –227.

AMAZON. **Amazon Elastic Compute Cloud (Amazon EC2)**. <http://aws.amazon.com/pt/ec2/>.

ARMBRUST, M.; FOX, A.; GRIFFITH, R.; JOSEPH, A. D.; KATZ, R.; KONWINSKI, A.; LEE, G.; PATTERSON, D.; RABKIN, A.; STOICA, I. et al. A view of cloud computing. **Communications of the ACM**, [S.l.], v.53, n.4, p.50–58, 2010.

BUYYA, R.; YEO, C. S.; VENUGOPAL, S.; BROBERG, J.; BRANDIC, I. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. **Future Gener. Comput. Syst.**, Amsterdam, The Netherlands, The Netherlands, v.25, n.6, p.599–616, June 2009.

DROPBOX. **Página Web**. <https://www.dropbox.com/>.

FACEBOOK. **Página Web**. <http://www.facebook.com/>.

FOSTER, I.; ZHAO, Y.; RAICU, I.; LU, S. Cloud Computing and Grid Computing 360-Degree Compared. In: GRID COMPUTING ENVIRONMENTS WORKSHOP, 2008. GCE '08, 2008. **Anais...** [S.l.: s.n.], 2008. p.1 –10.

FOX, A.; GRIFFITH, R. et al. Above the clouds: A Berkeley view of cloud computing. **Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Tech. Rep. UCB/EECS**, [S.l.], v.28, 2009.

GMAIL. **Página Web**. <http://mail.google.com/>.

GOOGLE. **Google App Engine**. <https://developers.google.com/appengine/>.

HOLTZMAN, J. M. **Nonlinear system theory: a functional analysis approach**. [S.l.]: Prentice Hall, 1970.

KAUFMANN, A.; GUPTA, M. **Introduction to Fuzzy Arithmetic: Theory and Applications**. [S.l.]: Van Nostrand Reinhold, New York, 1985.

LEE, C. C. Fuzzy logic in control systems: fuzzy logic controller. II. **Systems, Man and Cybernetics, IEEE Transactions on**, [S.l.], v.20, n.2, p.419–435, 1990.

MAMDANI, E. H.; ASSILIAN, S. An experiment in linguistic synthesis with a fuzzy logic controller. **International journal of man-machine studies**, [S.l.], v.7, n.1, p.1–13, 1975.

MELL, P.; GRANCE, T. The NIST definition of cloud computing (draft). **NIST special publication**, [S.l.], v.800, p.145, 2011.

NIST. **National Institute of Standards and Technology Draft Definition of Cloud Computing**.

OPENSTACK. **Página do projeto**.
url <http://www.openstack.org>.

PARKHILL, D. F. **The challenge of the computer utility**. [S.l.]: Addison-Wesley Publishing Company Reading, 1966. v.2.

SALESFORCE. **Página Web**. <http://www.salesforce.com>.

SANTANNA, S. R. Um Módulo para Tomada de Decisões em Regime de Incerteza Direcionado ao Escalonamento de Recursos em Grades Computacionais. **Computer**, [S.l.], p.93, 2015.

SHAW, I. S. **Controle e modelagem fuzzy**. [S.l.]: Edgard Blucher, 1999.

WEI Z., P. G.; CHI, C.-H. Scalable transactions for web applications in the cloud. In **Euro-Par**, [S.l.], p.442–53, 2009.

WIND, S. Open source cloud computing management platforms: Introduction, comparison, and recommendations for implementation. In: OPEN SYSTEMS (ICOS), 2011 IEEE CONFERENCE ON, 2011. **Anais...** [S.l.: s.n.], 2011. p.175 –179.

ZADEH, L. A. Fuzzy logic. **Computer**, [S.l.], n.4, p.83–93, 1988.

ZADEH, L. Fuzzy Sets. **Information and Control**, [S.l.], v.8, p.338–353, 1965.