

Alocação de Máquinas Virtuais no CloudSim e Openstack Symphony utilizando Lógica Fuzzy

Guilherme B. Schneider¹; Maurício L. Pilla¹; Vitor A. Ataídes¹;
Renata H. S. Reiser¹

¹Universidade Federal de Pelotas – {gbschneider, pilla, vataides, reiser} @inf.ufpel.edu.br

1. INTRODUÇÃO

A Computação em Nuvem (CN) possibilita acessar recursos computacionais (servidores, redes, serviços e aplicações) de maneira prática e sob demanda, e que podem ser liberados para o usuário sem qualquer envolvimento gerencial [1]. E ainda, viabilizam que desenvolvedores possam implantar suas aplicações em qualquer lugar do mundo, efetuando cobranças específicas dependendo das necessidades de qualidade de serviços de cada cliente. Essas características advém da criação de máquinas virtuais (VMs) em acordância com as necessidades da aplicação, alocandas em máquinas físicas (MFs). Embora tanto as VMs quanto as MFs apresentem configurações heterogêneas, nesta etapa inicial deste trabalho, restringem-se os estudos às configurações homogêneas. Assim um dos principais desafios de pesquisa é a alocação de VMs, e para tal alocadores são propostos com o intuito de otimizar o uso das MFs, considerando a maior quantidade de recursos na menor quantidade de MFs, garantindo desempenho e visando maior eficiência energética.

A principal motivação deste trabalho é prover uma recomendação considerando maior eficiência energética e otimização do escalonamento de VMs. Nesta meta, utiliza-se a bordagem fuzzy na tomada de decisão, buscando utilizar o menor número de MFs sem perder o desempenho. Para o ambiente simulado, optou-se em utilizar o framework CloudSim, com suporte a políticas definidas pelo usuário para alocação de recursos de MFs para VMs e ainda, em avaliar o estudo de vários projetos que estendem suas funcionalidades como RealCloudSim, CloudAnalyst, CloudSimEx, entre outros [3].

CloudSim [2] é um framework para modelar e simular a infraestrutura e os serviços de um ambiente de Computação em Nuvem. Seu principal objetivo é prover um ambiente completo de simulação que possibilita a modelagem e simulação de grandes ambientes de Computação em Nuvem, permitindo que provedores testem seus serviços, repetidamente, e sob um ambiente controlado, tudo isso com custo zero.

Já para o ambiente real optou-se pelo Symphony por ser uma ferramenta recém desenvolvida e de simples e eficiente manipulação. O Symphony é o módulo de escalonamento de VMs do OpenStack Orchestra [4]. Seu objetivo é definir em qual nó de computação uma nova VM será alocada e para isso o Symphony utiliza uma política de escalonamento de VMs. O Symphony foi desenvolvido com o intuito de simplificar a adição de novas políticas de alocação de VMs. Uma política consiste em uma função que tem como parâmetro uma lista de *logs*, onde cada *log* contém uma lista de nós de computação e a data do *log*. Cada nó de computação contém uma lista de VMs, seu nome, *IP*, consumo de CPU e consumo de memória. Cada VM contém seu nome, consumo de CPU, consumo de memória. O retorno deve ser o nome do nó de computação escolhido para receber a correspondente VM.

Para a tomada de decisão utilizou-se a Lógica Fuzzy, um Fuzzy Logic System (FLS) inclui um processo de fuzzificação, uma tabela de regras, mecanismo de inferência e etapa de defuzzificação. Muitas vezes, o conhecimento usado para construir as regras em um FLS é incerto. Três maneiras pelas quais tal incerteza de regra pode ocorrer são: 1) as palavras que são usadas em antecedentes e consequentes de regras podem significar coisas diferentes para pessoas diferentes; 2) consequentes obtidos por meio de votação de um grupo de especialistas serão muitas vezes diferentes para a mesma regra, porque os especialistas não estarão necessariamente de acordo; e 3) dados de treinamento ruidosos. [5]. Essas características presentes no FLS fazem dessa abordagem um excelente modelo para avaliar as incertezas extraídas das infraestruturas empregadas devido à variação no consumo de CPU e memória. Outro problema que o FLS ajuda a resolver é a classificação do uso dos recursos, enquanto o FLS fornece um valor fuzzificado do consumo de energia e não apenas um valor simples.

O artigo descreve as metodologias utilizadas na Seção 2. A Seção 3 mostra resultados e discussões do trabalho. Na na sequência, tem-se a conclusão.

2. METODOLOGIA

Para a execução dos testes considerou-se uma nuvem homogênea que contém 9 nós, onde um nó é o *controller* e os outros 8 nós são nós de computação. A configuração é apresentada abaixo:

- **Processador:** Intel Core i5 2310 2.9GHz (80000 MIPS, ou milhões de instruções por segundo)
- **Memória:** 16324MB, DDR3 1333 MHz
- **Disco:** ATA MB1000CBZQE, 1TB, 7200RPM
- **Sistema Operacional:** Ubuntu 14.04.3 LTS
- **Kernel:** Linux 3.19.0-37-generic

As VMs a serem alocadas são de características homogêneas, onde seus requisitos tem como objetivo restringir que cada MF contenha no máximo 5 VMs, de acordo com configuração apresentada a baixo:

- **Processador:** 15000 MIPS
- **Memória:** 3000MB
- **Disco:** 150GB

Os testes foram realizados considerando duas políticas diferentes:

- **Round Robin:** política consistindo em tentar manter uma distribuição de VMs homogênea. Neste caso como as máquinas iniciavam com nenhuma VM alocada, primeiramente ocorre alocação de uma VM em cada nó e na sequência o processo continua visando todos os nós com o mesmo número de VMs.

- **Compacta:** é uma política que consiste em manter o número máximo de nós de computação desocupados. Para isso, inicialmente, tenta-se manter o máximo de VMs num mesmo nó de computação e somente quando este nó não suportar mais VMs, alocam-se as novas VMs em outro nó.

Os testes consistiram em 50 execuções com cada algoritmo, divididos em: execuções com 8 VMs, 16 VMs, 24 VMs, 32 VMs e 40 VMs.

3. RESULTADOS E DISCUSSÃO

Tanto no CloudSim quanto no Symphony os resultados para todas as execuções com as políticas de alocação Round-Robin e Compacta foram os mesmos. Nas Figuras 1 e 2 são apresentados os valores obtidos na alocação de

8 VMs. O eixo horizontal representa o tempo necessário para a inserção de uma VM, e o eixo vertical representa cada nó de computação. A mudança de cores representa alteração no número de VMs de um nó de computação. Esses valores mantiveram constância nos demais testes em ambas políticas.

Na figura 1 são apresentados os resultados obtidos com o algoritmo Round Robin, observando-se que o intervalo entre os tempos 1 e 8, caracterizando o período onde o poder computacional não está mais homogêneo, o algoritmo procura retomar essa homogeneidade alocando cada VM subsequente no próximo nó ainda sem nenhuma VM.

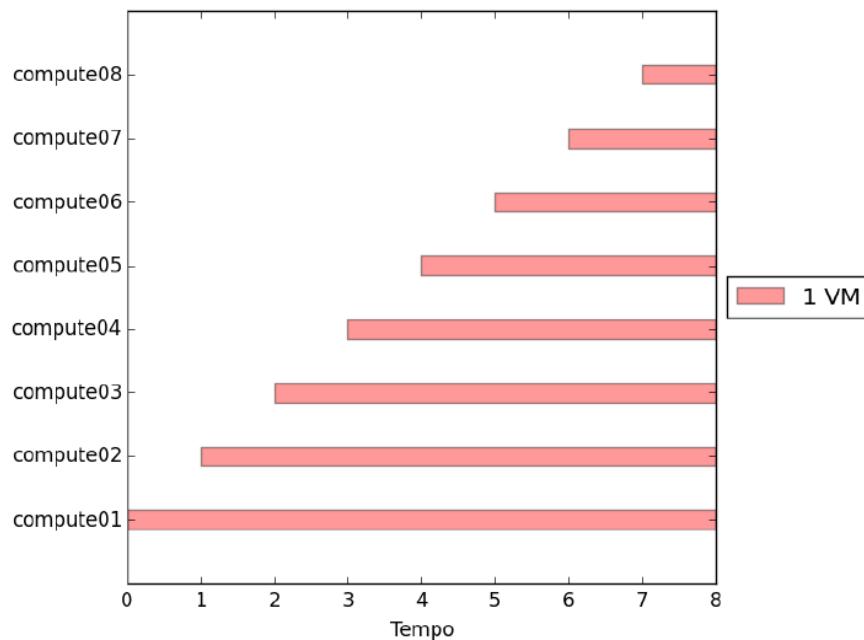


Figura 1. Inserção de 8 VMs com a política RoundRobin

Na figura 2, tem-se os resultados obtidos pelo algoritmo Compacta, ressaltando que até o tempo 5 somente o nó compute01 possui VMs. Já, no tempo 6, o algoritmo procura o próximo nó a ser preenchido com VMs devido a limitação proposta nas configurações de VMs, cada nó suporta no máximo 5 VMs.

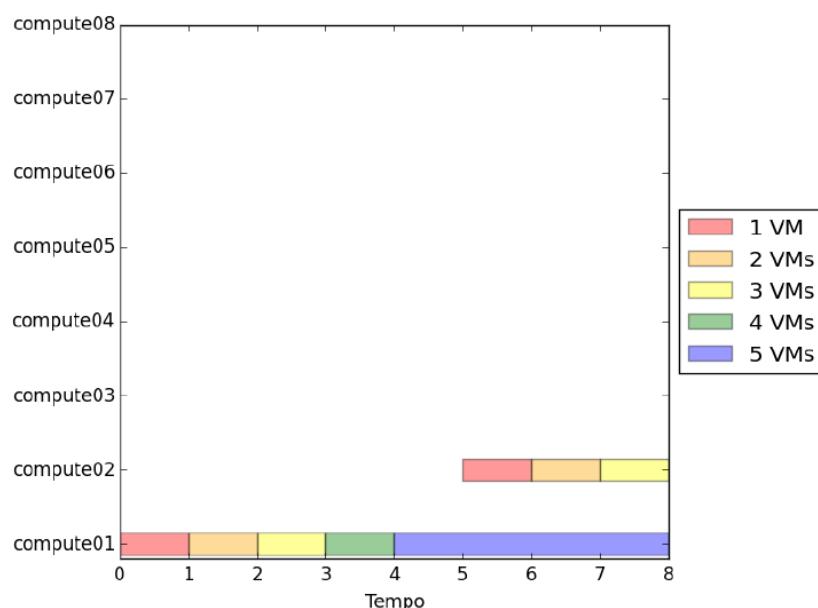


Figura 2. Inserção de 8 VMs com a política Compacta

Para a elaboração da política de escalonamento a ser desenvolvida irá ser utilizada a Lógica Fuzzy, onde as entradas serão o consumo atual da CPU e da memória e a saída será a utilização atual daquela MF para que o sistema escolha uma máquina da maneira mais eficiente possível sem desperdiçar energia.

4. CONCLUSÕES

Neste trabalho foi introduzido o conceito de CN juntamente com duas ferramentas de aplicação deste conceito. Para os testes foram selecionados dois algoritmos clássicos esperando que ambas ferramentas obtivessem o mesmo resultado, validando a característica de similaridade. Os resultados obtidos nos testes, tanto no CloudSim quanto no OpenStack Simphony, foram iguais e comprovando a similaridade das aplicações, tanto no ambiente simulado quanto no ambiente real. Assim possibilitando que sejam feitas simulações de ambientes de CN que corresponderão no ambiente real.

Em trabalhos futuros, pretende-se implementar novas políticas de escalonamento utilizando Lógica Fuzzy, considerando as incertezas nas variáveis da CN.

Para implementar essas novas políticas pretende se basear no trabalho [6] onde é proposto um algoritmo de escalonamento de VMs de acordo com a condição climática dos datacenters em determinado momento devido a produção de energia ser renovável e também no trabalho [7] onde o escalonamento das tarefas é dado de acordo com a disponibilidade dos recursos (largura de banda, memória e espaço em disco) devido a ambos trabalhos utilizarem Lógica Fuzzy como tomada de decisão.

5. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Mell, P., Grance, T., et al. (2011). **The nist definition of cloud computing.** Page 7.
- [2] Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., and Buyya, R. (2011). Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. **Software: Practice and experience**, 41(1):23–50.
- [3] CLOUDS. Cloudsim: A framework for modeling and simulation of cloud computing infrastructures and services. Disponível em <<http://www.cloudbus.org/cloudsim/>>. 2013.
- [4] Vítor A. Ataides, Laercio L. Pilla, M. L. P. (2017). Openstack orchestra, um escalonador de máquinas virtuais é balanceador de carga para nuvens openstack.
- [5] C. Alvarez and J. NaCorbal and M. Valero. Fuzzy memoization for floating-point multimedia applications. *IEEE Transactions on Computers*(2005) 922-927
- [6] Toosi, A. N., & Buyya, R. (2015, December). A fuzzy logic-based controller for cost and energy efficient load balancing in geo-distributed data centers. In *Utility and Cloud Computing (UCC), 2015 IEEE/ACM 8th International Conference on* (pp. 186-194). IEEE.
- [7] Nine, M. S. Z., Azad, M. A. K., Abdullah, S., & Rahman, R. M. (2013, July). Fuzzy logic based dynamic load balancing in virtualized data centers. In *Fuzzy Systems (FUZZ), 2013 IEEE International Conference on* (pp. 1-7). IEEE.