UNIVERSIDADE ESTADUAL DO CEARÁ

CENTRO DE CIÊNCIAS E TECNOLOGIA

MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO

FILIPE FERNANDES DOS SANTOS BRASIL DE MATOS

VBalance: Uma Política de Seleção de Máquinas Virtuais para Balanceamento de Cargas em Cloud Computing

Fortaleza - Ceará

2014

FILIPE FERNANDES DOS SANTOS BRASIL DE MATOS

VBalance: Uma Política de Seleção de Máquinas Virtuais para Balanceamento de Cargas em Cloud Computing

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação Acadêmica em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito para obtenção do título de Mestre em Ciência da Computação.

Orientação: Prof. Ph.D. Joaquim Celestino Júnior
Coorientação: Prof. Dr. André Ribeiro Cardoso

Fortaleza - Ceará

2014

Filipe Fernandes dos Santos Brasil de Matos.

VBalance: Uma Política de Seleção de Máquinas Virtuais para Balanceamento de Cargas em Cloud Computing / Filipe Fernandes dos Santos Brasil de Matos. – Fortaleza - Ceará, 2014.

65 p.;il.

Dissertação - Universidade Estadual do Ceará, Centro de Ciências e Tecnologia, Mestrado Acadêmico em Ciência da Computação, Fortaleza - Ceará, 2014.

Orientação: Prof. Ph.D. Joaquim Celestino Júnior.

1. Computação em nuvens 2. Política 3. Seleção 4. Regressão linear 5.
6.
. I. Título.

FILIPE FERNANDES DOS SANTOS BRASIL DE MATOS

VBalance: Uma Política de Seleção de Máquinas Virtuais para Balanceamento de Cargas em Cloud Computing

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação Acadêmica em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito para obtenção do título de Mestre em Ciência da Computação.

Aprovado em: 20/06/2014

BANCA EXAMINADORA

Prof. Ph.D. Joaquim Celestino Júnior
(Orientador)
Universidade Estadual do Ceará - UECE

Prof. Dr. André Ribeiro Cardoso (Co-Orientador)
Universidade Estadual do Ceará - UECE

Prof. Dr. Paulo Henrique Mendes Maia
Universidade Estadual do Ceará - UECE

Prof. Dr. Cidcley Teixeira de Souza
Instituto Federal de Educação, Ciência e Tecnologia do Ceará - IFCE

AGRADECIMENTOS

Aos meus amigos e familiares pelo apoio incondicional, principalmente aos meus pais por acreditarem e investirem sempre no meu potencial.

À Sabrina Kécia Falcão Pereira pela compreensão durante esses últimos meses e por todos os momentos felizes ao qual passamos juntos nesses últimos anos.

Aos meus amigos de UECE pela companhia nos momentos mais difíceis (e mais divertidos) dessa longa jornada e pelos diversos "Unidos venceremos" no decorrer do curso. Em especial, aos colegas Silas Santiago, Sávio Moreira, João Eduardo e David Lira.

Em especial, aos meus amigos Silvio Roberto, Bruno Lopes e Filipe Maciel pela paciência que tiveram e a valorosa ajuda para que esse trabalho fosse concluído.

Aos meus amigos da Acens por me proporcionarem tamanha experiência de vida, acreditarem em minhas capacidades e contribuírem substancialmente para o que sou hoje.

A todos os professores que, de alguma forma, contribuiram para o meu engrandecimento acadêmico e profissional.

Em especial, aos professores Joaquim Celestino e André Cardoso por todas as oportunidades proporcionadas e pela paciência, dedicação e orientação durante a elaboração desse trabalho.

”Para ter sucesso, seu desejo de sucesso deve ser maior que o medo de falhar”
Bill Cosby

RESUMO

Computação em Nuvens (*Cloud Computing*) é um novo paradigma computacional onde o processamento e armazenamento de dados são ofertados como serviços para os clientes que os acessam através da Internet. Tais serviços são representados sob a forma de Máquinas Virtuais (*Virtual Machines* ou VMs) hospedando-se e compartilhando os recursos disponibilizados por servidores reais, encontrados nos grandes *data centers*. A alta procura por soluções na Nuvem culminou em maiores gastos dos provedores de serviço com energia elétrica. Hoje, tal dispêndio financeiro representa uma das maiores despesas na manutenção de um *data center*. Dessa maneira, surgiram diversas soluções que objetivam o máximo atendimento de requisitos e demandas solicitados pelos clientes, minimizando os gastos energéticos. Este trabalho apresenta o projeto de uma política de seleção de Máquinas Virtuais para balanceamento de carga entre servidores de um *data center* ponderando as questões de consumo de energia durante sua escolha. A política VBalance, utilizando-se do modelo de Regressão Linear Múltipla, selecionará a VM que mais contribui com a energia do servidor em análise para migração. Uma análise comparativa da solução proposta com outros mecanismos existentes, visando avaliar o desempenho, será apresentada ao final do trabalho.

Palavras-Chave: Computação em nuvens. Política. Seleção. Regressão linear.

.

ABSTRACT

Cloud computing is a new computing paradigm which the data processing and data storage are offered to customers like services accessed over the Internet. These services are represented by Virtual Machines (VMs) hosted by real servers located in large data centers. The Virtual Machines share the physical resources offered by servers. The growing demand for solutions related to Cloud resulted in greater investments with electricity for service providers. Today, this financial costs represents one of the greatest expenditures to maintain data centers. Therefore, solutions have been proposed to minimize power expenditures, while maintains a maximum care with requirements demanded by customers. This work presents a selection policy project used to balance workload among servers in a data center. The policy proposed will consider power issues during its choice. The VBalance policy will select to migration the VM that most contibutes with the power consumption of the server using the Multiple Linear Regression model. Furthermore, a comparative analysis is made of the proposed solution with other existing models, to evaluate the performance.

Keywords: Cloud computing. Policy. Selection. Linear regression.

.

SUMÁRIO

1 INTRODUÇÃO

1.1 Contextualização

A crescente busca por soluções em Sistemas Distribuídos, em especial, o modelo computacional conhecido como Computação em Nuvens (*Cloud Computing*), modificou profundamente a interação entre usuários e computadores. Ao contrário do paradigma tradicional, onde usuários comuns utilizam computadores pessoais (PCs) para realizar suas atividades e grandes empresas precisam construir seu próprio parque de servidores para processamento e armazenamento de dados, o modelo em nuvem oferece a possibilidade de criação e utilização desses computadores de maneira remota através da Internet (ARMBRUST et al., 2009). Todas essas ações são oferecidas na forma de serviço.

Essa nova interação entre usuários e provedores de serviços é regulamentada por um documento chamado de Acordo de Nível de Serviço (*Service Level Agreement* ou SLA) (C.GUO et al., 2010) onde ficam registradas regras e valores do serviço prestado, bem como as penalizações em caso de alguma infração à norma, tanto por parte do provedor, como do consumidor do serviço. Desta forma, o SLA deve ser seguido de maneira precisa por ambas as partes envolvidas para evitar prejuízos financeiros.

Uma nuvem pode comportar a execução de inúmeros serviços de forma simultânea. Essa funcionalidade é oriunda da utilização de uma tecnologia denominada virtualização (ZHANG; CHENG; BOUTABA, 2010). A virtualização consiste em criar instâncias virtuais de componentes físicos do *hardware* de um servidor e distribuí-los entre máquinas virtuais (*Virtual Machines* ou VMs) que serão utilizadas pelos consumidores do serviço. Esses componentes virtualizados de *hardware* estão livres para serem configurados de acordo com as exigências especificadas pelo SLA, desde que obedeçam o limite físico do servidor. A virtualização também trata de gerenciar a utilização dos recursos físicos, permitindo o compartilhamento de *hardware* entre máquinas virtuais.

As vantagens desse novo modelo em relação ao tradicional são inúmeras, mas algumas merecem um destaque especial: em primeiro lugar, a economia financeira gerada pela não aquisição e manutenção de máquinas para a montagem de um *data center* privado, visto que esses procedimentos deverão ser realizados pelos provedores de serviço. Em segundo lugar, graças a técnicas de virtualização, a possibilidade de implementar um mecanismo de alocação de recursos (e de tarifação) sob demanda, o que, mais uma vez, representa economia para seus usuários; por fim, mas não menos importante, a exploração máxima dos recursos computacionais disponíveis nos grandes *data centers* que, antes desse modelo, eram subutilizados (NAYAK; YASSIR, 2012).

As vantagens desse novo conceito resultaram em um aumento significativo na busca pelos serviços ofertados por esse novo paradigma. A crescente demanda, por sua vez, motivou o surgimento de novos provedores de serviço, assim como a expansão daqueles que já estavam consolidados no mercado. Essa ampliação trouxe consigo uma série de problemas e desafios, dentre os quais se destaca o consumo de energia elétrica em um *data center*. Para se ter uma idéia, segundo Beloglazov e Buyya (2012), o consumo energético de um único *data center* pode ser comparado ao consumo de 25000 domicílios. Tal informação constata a importância de soluções de gerenciamento do *data center* voltadas para a economia de energia elétrica, porém sem causar grandes impactos no desempenho das máquinas virtuais em execução.

Estudos como (FAN; WEBER; BARROSO, 2007) e (BARROSO; HöLZLE, 2007) apontam elevados gastos energéticos observados em grandes *data centers* indo muito mais além do que a quantidade de servidores constituintes ou a presença, ou não, de tecnologias voltadas para reduzir o consumo de energia das máquinas. Os dois trabalhos citados mostram que servidores, enquanto ativos, tendem a ser subutilizados (operam com, no máximo, 50% de sua capacidade total) e consomem, sem executar qualquer atividade computacional, em média, 70% de sua energia total. Logo, se fazem necessárias políticas inteligentes de alocação e seleção de máquinas virtuais para migrações entre servidores de forma a maximizar a consolidação das VMs (alocá-las em um menor número de servidores), mas sem esquecer questões importantes de desempenho.

Neste contexto, surge a necessidade de um mecanismo de seleção de máquinas virtuais para balanceamento de cargas de trabalho entre os servidores constituintes de um *data center* que deverá, acima de tudo, priorizar a economia de energia geral por ele consumida.

1.2 Motivação

O constante aumento da demanda do mercado pelo paradigma de Computação em Nuvens motiva o surgimento de novos provedores desse tipo de serviço, assim como uma expansão estrutural das empresas que já oferecem soluções baseadas nesse paradgima. Esse crescimento dos recursos de *hardware* culmina com um aumento significativo no consumo de energia observado nos *data centers*, tornado-os uma das entidades, no meio computacional que mais consomem eletricidade. E, como consequência, um estudo sobre metodologias que tenham o objetivo de reduzir esse consumo se faz necessário.

A importância das políticas de seleção para um balanceamento eficiente de cargas de trabalho entre os servidores de um *data center* se faz presente. Contudo, a maior parte dessas políticas não pondera o impacto no consumo de energia durante o processo de escolha de uma máquina virtual para migração. E, mesmo aquelas que a consideram, realizam apenas uma análise do consumo corrente de recursos das VMs envolvidas.

1.3 Objetivos

Os objetivos geral e específicos deste trabalho estão descritos abaixo.

1.3.1 Objetivo geral

Esse trabalho tem como objetivo geral definir uma política de seleção de máquinas virtuais que priorize minimizar o consumo energético geral dos *data centers*, mas que também leve em conta, durante o seu processo de escolha, o comportamento das VMs envolvidas, visando manter os seus

respectivos desempenhos computacionais mesmo que elas estejam sob migração. A política aqui proposta consistirá em coletar informações de consumo de CPU das máquinas virtuais e de Energia das máquinas físicas e, através da técnica de Regressão Linear Múltipla, escolher a VM com maior influência energética para migração.

1.3.2 Objetivos específicos

- Criar e implementar uma nova política de seleção para o simulador *CloudSim*;
- Definir um cenário próximo a um *data center* real para ser utilizado na validação da proposta;
- Definir métricas que retratem o desempenho energético e de violações de SLA das técnicas envolvidas na simulação;
- Realizar uma comparação entre a política de seleção proposta e as já existentes;
- Garantir que a política de seleção proposta reduza o consumo energético geral do *data center* e mantenha o nível de obediência a SLA observado nas demais políticas;

1.4 Metodologia

Inicialmente, deverá ser feito um estudo sobre os principais conceitos e características de assuntos relacionados à problemática deste trabalho, em especial, Computação em Nuvens, Computação Verde e Virtualização.

Em seguida será realizado um levantamento sobre a problemática de alocação e migração de máquinas virtuais e os seus respectivos impactos no consumo de energia de um *data center*.

Feito este levantamento inicial, serão identificados as melhores soluções propostas dentro do grupo de trabalhos relacionados, objetivando futuras comparações com a proposta VBalance. Também será realizada uma análise crítica do comportamento de cada uma delas, buscando salientar os seus pontos positivos e negativos e ponderar sobre a incorporação de algumas dessas soluções na proposta VBalance.

De posse de todas essas informações e análises o algoritmo VBalance poderá, enfim, ser criado. Inicialmente, deverá ser construída uma formulação matemática da proposta para, em seguida, escrever o algoritmo em questão. Também serão realizados testes preliminares visando calibrar algumas possíveis constantes que virão a ser utilizadas pelo VBalance.

Por fim, a solução proposta nesse trabalho e aquelas escolhidas como referência serão implementadas no simulador *CloudSim* e testadas. Ao final dos testes, com base nos dados coletados, serão realizadas análises comparativas entre todas as propostas envolvidas e serão identificadas as principais vantagens e desvantagens das propostas, em especial, do VBalance.

1.5 Organização do trabalho

O restante deste trabalho está organizado da seguinte forma. O capítulo 2 faz um levantamento teórico sobre os principais conceitos ligados à proposta. Os trabalhos relacionados serão vistos no capítulo 3. O capítulo 4, por sua vez, apresenta uma descrição detalhada da problemática e do algoritmo proposto incluindo uma formularização matemática de ambos os tópicos. O capítulo 5 contém os dados referentes a simulação com o *CloudSim*, bem como definição de métricas para análise e a avaliação de desempenho da proposta. Por fim, o capítulo 6 mostra as conclusões obtidas com os resultados dos testes realizados e também os possíveis trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a fundamentação teórica necessária para o desenvolvimento e o entendimento da política de seleção proposta neste trabalho. Dentre os assuntos abordados estão os principais conceitos de Computação em Nuvens, Computação Verde, Virtualização e a técnica de Regressão utilizada pela política VBalance para definir qual máquina virtual será migrada.

2.1 Computação em nuvens

A tecnologia de Computação em Nuvens (*Cloud Computing*) está cada vez mais presente tanto no meio empresarial, como no meio acadêmico (BAGULEY, 2013). Constantemente, novas empresas aderem à esta tecnologia em busca de, principalmente, praticidade e economia na criação e na manutenção dos seus ambientes computacionais. A comunidade científica, em resposta a esta crescente demanda, pesquisa soluções para os inúmeros problemas que ainda existem em Computação em Nuvens (MORENO-VOZMEDIANO; MONTERO; LLORENTE, 2013).

Embora o conceito de Computação em Nuvens seja relativamente novo, a idéia por trás do termo é bem antiga (HASSAN, 2011). Durante as décadas de 1960 e 1970, os *mainframes*, embora caros, dominavam o mercado, pois facilitavam e tornavam mais rápidas as atividades diárias em instituições públicas e privadas. Um *mainframe*, basicamente, era formado por um grande componente responsável pelo processamento e armazenamento de dados e por inúmeros terminais burros. Esses terminais realizavam somente a interação do usuário com o sistema, ou seja, eles serviam apenas como entrada e saída de informações, sem qualquer tipo de computação.

Motivado por esse ambiente dominante na epóca, John McCarthy sugeriu que a computação fosse tratada como um serviço, semelhante ao que ocorre, por exemplo, com a distribuição de energia elétrica e de água, onde o valor a ser pago é proporcional à quantidade de recursos utilizados. Segundo ele, *"a computação poderia, algum dia, ser organizada como uma utilidade pública do mesmo modo que o telefone"*. Alguns pesquisadores vêem esta como a primeira definição sobre Computação em Nuvens (ZHANG, 2012).

O pensamento definido por John McCarthy descrito no parágrafo anterior se tornou um dos principais alicerces para a formulação conceitual da tecnologia de Computação em Nuvens hoje conhecida. A idéia de organizar a computação como um serviço se tornou a característica chave deste novo paradigma e o seu diferencial mais destacável. De forma bastante resumida, os clientes do modelo em Nuvem podem executar, via navegador de internet, seus aplicativos sem a necessidade de instalar qualquer componente de *software* ou *hardware* em sua máquina local, pois o real processamento ocorre dentro da Nuvem.

A Computação em Nuvens se assemelha bastante com a arquitetura do *mainframe* descrita anteriormente: processamento concentrado em um único local e terminais burros para permitir uma interação entre usuários do sistema e centro de computação. Já as principais diferenças entre as duas tecnologias estão na escala do ambiente computacional e no modelo de tarifação.

O centro de processamento e armazenamento de dados em Computação em Nuvens é denominado *data center* e é composto por milhares de máquinas conhecidas como servidores. Estes

servidores são responsáveis por, localmente, hospedar e executar as aplicações do usuário, recebendo parâmetros de entrada e retornando resultados para o computador do cliente que, por sua vez, serve apenas como um meio de entrada e saída de informações (terminal burro). Tal estrutura é parecida com aquela adotada pelos *mainframes*.

Ao contrário dos *mainframes*, onde a interação entre usuários e computadores é local, em Computação em Nuvens este relacionamento cresceu a um nível mundial. Melhor explicando, utilizando-se dos avanços tecnológicos obtidos em infraestrutura de rede, foi possível conectar usuários e parques de computação, *data centers*, através da internet. Assim, não era mais necessário uma proximidade física entre usuário e computador, os dois poderíam estar geograficamente separados e, mesmo assim, interagirem.

A Computação em Nuvens adota um esquema de pagamento personalizado, onde cada um dos seus usuários pagará, apenas, pelos recursos destinados a ele durante o período em que a computação de seus programas ocorrem. Ou seja, será cobrado do cliente uma quantia proporcional ao nível de recursos do *data center* que ele utilizou. Este modelo é conhecido como *pay-per-use*.

Qualquer forma de computação neste novo paradigma é tratado como um serviço. Dessa forma, existem aqueles responsáveis por fornecer e aqueles que utilizam esses serviços da Nuvem, conhecidos, respectivamente, como Provedores e Consumidores. Provedores de serviço, comumente, são grandes empresas, donas de imensos *data centers* que oferecem um ou mais serviços a serem executados em seus servidores. O bom exemplo deste grupo é a Google com seu conjunto de ferramentas de escritório Google Apps (GOOGLE, 2013a).

Os Consumidores de serviço, por sua vez, são, em sua grande maioria, pessoas comuns que usam os serviços prestados pelos Provedores. Contudo, nada impede que outras empresas, da área de Tecnologia da Informação, ou não, se comportem como Consumidores de serviço. A Figura 1 mostra exemplos de empresas que já oferecem/utilizam serviços na Nuvem. Esta relação entre Provedores e Consumidores de serviços exige um documento escrito onde as regras do acordo fechado entre as partes seja registrada e oficializada perante os orgão responsáveis. Esse documento é conhecido como Acordo de Nível de Serviço (*Service Level Agreement* ou SLA). O SLA também define os direitos e deveres de cada um dos envolvidos no acordo, bem como a penalidade aplicada em caso de alguma violação ao acordo.

Embora a alocação de serviços em servidores dedicados, de um servidor por cliente, seja possível em Computação em Nuvens em *data centers*, o uso deste tipo de hospedagem é desaconselhado por três motivos principais: a) o alto custo com a manutenção do sistema quando o número de clientes aumenta, mais servidores têm de ser comprados e instalados; b) a constante subutilização dos servidores onde, nem sempre, os clientes usam todo o poder computacional que um servidor é capaz de oferecer; c) a impossibilidade de oferecer elasticidade de recursos aos clientes, pois não é possível mensurar o uso de cada Consumidor e, muito menos, tarifar em cima do nível de utilização. Motivado por tais limitações, foi adotado um modelo de Nuvem baseado na técnica de Virtualização.

A Virtualização, basicamente, consiste em criar instâncias virtuais de componentes físicos reais de tal forma que, estes elementos, quando agrupados, criam uma versão lógica da máquina real onde ela foi montada. Esta versão lógica é conhecida como máquina virtual (*Virtual Machine* ou VM). Cada VM pode ter uma configuração própria, desde que sejam obedecidas as limitações de *hardware* da máquina física e, graças a camada de Virtualização, é possível duas ou mais máquinas virtuais serem alocadas em um mesmo servidor, bem como a alocação dinâmica de seus recursos. A técnica de Virtualização será melhor explicada na seção posterior.

Figura 1 – Empresas que utilizam soluções em Nuvem

Segundo (TAURION, 2009), as nuvens podem ser classificadas de duas maneiras distintas: quanto ao seu grau de compartilhamento e quanto ao seu foco de atuação. Vale ressaltar que essas duas classificações não são excludentes entre si, ou seja, toda nuvem sempre tem um grau de compartilhamento e um foco de atuação.

Em relação ao grau de compartilhamento, as nuvens podem ser subdividas em nuvens públicas, privadas e híbridas. A Figura 2 representa, graficamente, o relacionamento entre elas:

- **Nuvens Públicas:** As nuvens públicas, mais conhecidas e populares, são aquelas desenvolvidas para serem utilizadas por terceiros. Embora recebam esse título, nem sempre, as nuvens públicas são de natureza gratuita. O pagamento de taxas por utilização depende do modelo de negócios da empresa provedora do serviço (TAURION, 2009). Devido a diversidade de acesso de clientes e a heterogeneidade de hardware e software que a compõem, questões de segurança como confiabilidade (ou seja, serviço prestado ao cliente sem problemas), integridade (ou seja, os dados na nuvem são acessados ou modificados por usuários maliciosos) e portabilidade (ou seja, possibilidade de migração de serviços entre nuvens) são muito importantes;

- **Nuvens Privadas:** Já as nuvens privadas ou empresariais são montadas por empresas para utilização particular, ou seja, para armazenar suas próprias informações e gerenciar a execução de suas aplicações. Em virtude do alto investimento empregado na implantação dos equipamentos, além da quantidade reduzida de tipos de clientes e recursos computacionais, as nuvens privadas tendem a ser mais homogêneas e apresentam mecanismos de segurança mais poderosos;

- **Nuvens Híbridas:** Por fim, as nuvens híbridas unem as principais características dos tipos das nuvens supracitados. Normalmente, elas são compostas por uma parte pública e uma privada, sendo esta última protegida da outra por *firewall*. Muitas empresas que adotam a abordagem

híbrida tendem a alocar informações e aplicativos de alta confidencialidade na parte privada da nuvem. Já os não pertinentes, são alocados no lado público onde estão abertamente disponíveis a todos. Segundo (ZAGE; FRANKLIN; VINCENT, 2013), as Nuvens Híbridas são o futuro da Computação em Nuvens.

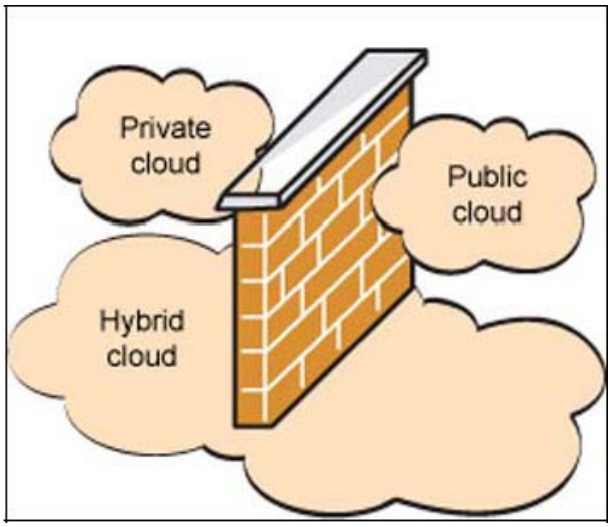


Figura 2 – Tipos de Nuvens (DUSTIN; SCOTT, 2009)

Em relação ao foco de atuação, as Nuvens podem ser classificadas como:

- **Voltadas para usuários finais:** Compostas por dezenas de milhares de usuários e servidores e caracterizadas por não possuir um modelo matemático de acesso a Nuvem bem definido. Dessa maneira, se faz necessário mecanismos robustos de balacenamento de carga e, principalmente, de escalonamento, visando um gerenciamento eficiente da execução em paralelo dessa quantidade variável de máquinas virtuais.
- **Voltadas para empresas:** São, normalmente, compostas por um número reduzido de servidores e apresentam melhores configurações quando comparadas as Nuvens voltadas para usuários finais. Essas nuvens se caracterizam por trabalhar com um número reduzido de clientes , normalmente, usuários de uma única empresa e operar aplicativos de alta importância e informações confidenciais. Assim, a segurança durante a execução de programas e manipulação de dados recebem atenção especial.

Os serviços providos em Computação em Nuvem podem ser classificados em diferentes níveis, onde cada um deles especifica um tipo de recurso a ser comercializado. Em todos eles, a tarifação é feita sob a utilização dos recursos por cliente. Segundo (BUYYA; BELOGLAZOV; ABAWAJY, 2010) e (ZHANG; CHENG; BOUTABA, 2010) os três principais níveis são (ver Figura 3):

- ***Infrastructure as a Service* (IAAS):** Neste nível, a infraestrutura de hardware é oferecida como um serviço. O usuário requisita equipamentos, sob a forma de máquinas virtuais, com as configurações de processador, memórias e interface de rede pretendidas por ele. Os principais exemplos de ferramentas que prestam serviços desse nível são a Amazon EC2 (AMAZON, 2013) da própria Amazon e o IBM Cloud da IBM (IBM, 2013);
- ***Plataform as a Service* (PAAS):** Está relacionada a serviços de desenvolvimento e gerenciamento de aplicativos para a Nuvem. Aqui são disponibilizadas ferramentas importantes para construção de programas: APIs, editores de texto, compiladores, ferramentas de controle de versão e etc. O principal exemplo desse nível é o *Google App Engine* (GOOGLE, 2011);
- ***Software as a Service* (SAAS):** Referente as aplicações desenvolvidas para serem processadas na Nuvem. Neste nível, os serviços são executados na Nuvem do Provedor e acessados, via internet, pelos seus consumidores. Os principais exemplos dessa camada são o Google Docs (GOOGLE, 2013b), Salesforce.com (SALESFORCE, 2013) e o Microsoft Office 2010 (MICROSOFT, 2013).

Segundo (VERDI et al., 2010), as principais características vinculadas à Computação em Nuvem são:

- **Serviços sob demanda:** De maneira automatizada, os recursos físicos dos servidores tais como ciclos de CPU, largura de banda, memória e etc. são alocados e desalocados proporcionalmente as necessidades dos clientes;
- **Elasticidade:** O gerenciamento dos recursos utilizados, seja feito de uma maneira rápida e invisível para os usuários dos recursos;
- **Amplo acesso aos recursos:** Graças ao uso da internet como meio de acesso aos serviços da nuvem, é possível requisitar tais recursos em qualquer lugar do mundo, bastando haver uma conexão com a internet;

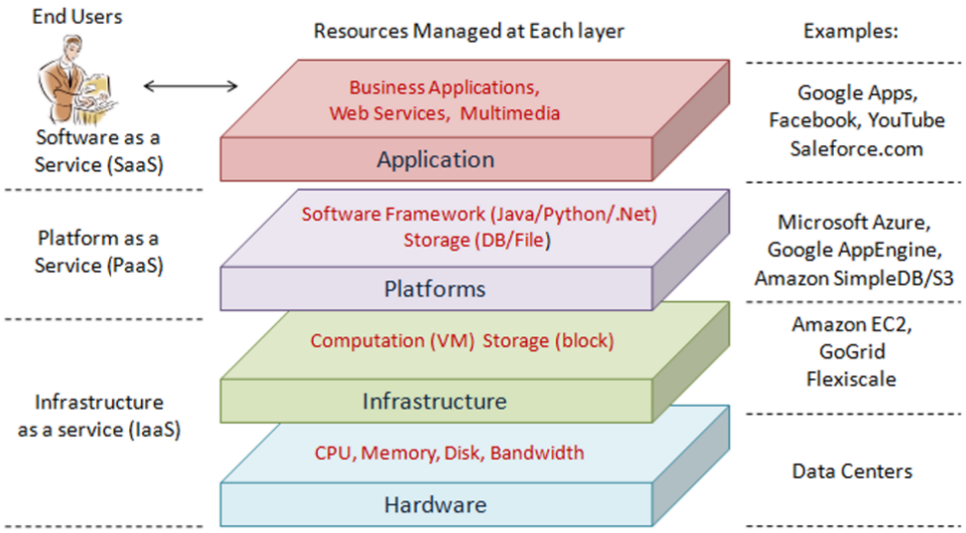


Figura 3 – Modelo organizacional dos serviços *Cloud* (ZHANG; CHENG; BOUTABA, 2010)

- ***Resource Pooling*:** A camada de virtualização inclusa sobre o hardware do *data center* passa, ao cliente, a falsa sensação de recursos ilimitados e sempre disponíveis;
- **Medição dos serviços:** Faz-se importante, que os recursos consumidos pelos usuários sejam, constantemente, monitorados de forma a mensurar, corretamente, o período de utilização e a quantidade usufruída para posterior tarifação;
- **Auto-organização:** A Nuvem deve garantir um certo nível de confiabilidade nos serviços prestados. Para isso é importante que ela coordene a execução de todos os serviços alocados e atenda, de maneira eficaz, as variações de demanda de cada um dos clientes.

2.2 Virtualização

A técnica de Virtualização é importante para o funcionamento da tecnologia de Computação em Nuvens, pois segundo sua utilização se torna possível implementar funcionalidades chaves tais como, por exemplo, a elasticidade de recursos, a independência em relação ao *hardware* e o isolamento de máquinas virtuais, diferenciando e destacando Computação em Nuvens das demais soluções existentes.

O trabalho de (NANDA; CHIUEH, 2005) apresenta uma análise sobre as principais ferramentas relacionadas a Virtualização, assim como os conceitos mais importantes desta tecnologia. Em virtude da abrangência deste estudo, a definição proposta por ele será adotada como um referencial para a concepção desta dissertação: *"Virtualização é uma tecnologia que combina ou divide recursos computacionais para executar um ou mais sistemas operacionais usando metodologias como particionamento ou agregação de hardware e software, simulação parcial ou completa de máquina virtual, emulação e muitas outras."*

As primeiras tentativas de criação de um ambiente virtualizado remotam das décadas de 1960 e 1970 (TANENBAUM; OTEEN, 2007). Naquela época, as aplicações eram desenvolvidas para sistemas computacionais específicos e, muitas delas, eram antigas e exigiam *hardware* já obsoletos

para serem executadas. Dessa maneira, era importante a construção de uma solução capaz de garantir a portabilidade dos programas entre os sistemas existentes e de simular o comportamento dos recursos físicos não existentes quando necessário.

Visando atingir os objetivos supracitados, no início da década de 1970, a IBM desenvolveu a primeira Máquina Virtual conhecida (TAKEMURA; CRAWFORD, 2010): *Virtual Machine Facility/370* ou *VM/370*. O *IBM System/370* particionava o hardware e criava instâncias lógicas de cada componente real distribuídas entre as suas VMs (VM/370). Tais ações, somado ao monitoramento constante das VMs, possibilitavam a execução simultânea de múltiplos sistemas operacionais sob o mesmo *hardware*.

O gerenciamento das VMs ressaltado no parágrafo anterior é realizado por uma entidade especial do sistema denominada Monitor de Máquinas Virtuais (MMV) ou *hypervisor*. A funcionalidade básica do *hypervisor* é isolar as VMs entre si a fim de uma não interfir no processamento das demais, bem como criar a abstração de recuros oferecida para as máquinas virtuais.

Segundo (SOARES, 2010) as principais aplicações da Virtualização são:

- **Consolidação de servidores:** A técnica de Virtualização oferece a execução paralela de múltiplas VMs, prestando diferentes serviços e consumindo variados tipos de recursos, sobre o mesmo servidor. Assim, é possível usufruir mais do potencial computacional de um servidor, bem como reduzir o seu nível de subutilização;
- **Consolidação de aplicações:** A independência entre software e hardware possibilita que todos os tipos de aplicações possam ser executados sobre um equipamento virtualizado. Mesmo no caso onde recursos obsoletos sejam requisitados, a Virtualização é capaz de simular o *hardware* necessário para o processamento dos programas;
- ***Sandboxing*:** O isolamento entre VMs garante a criação e a manutenção de inúmeros ambientes para execução de aplicações. Assim, é possível garantir uma separação lógica entre esses ambientes e assegurar as atividades de uma VM, mesmo maliciosas, não influenciando no funcionamento das demais;
- **Suporte a múltiplos ambientes de execução:** A Virtualização, conforme observado anteriormente, é capaz de hospedar várias VMs que, em muitos casos, podem ter configurações distintas entre si. Dessa maneira, o sistema está sempre apto a executar, paralelamente, diferentes programas com variados requisitos de *hardware* e *software*.

O conceito de Virtualização pode ser aplicado diferentemente. Embora, em uma visão geral, estas implementações busquem os mesmos objetivos, simular recursos físicos e garantir a portabilidade dos programas entre sistemas, estas técnicas podem diferir uma da outra em seus modos de implementação. Os principais tipos, segundo (MATTHEWS et al., 2008), são:

- **Emulação:** A Virtualização por emulação deverá simular a arquitetura de hardware necessária para o funcionamento dos sistemas encapsulados pelas máquinas virtuais. Este tipo de Virtualização permite simular diversas arquiteturas distintas, incluindo aquelas completamente diferentes da real, sobre uma única máquina física. São exemplos deste tipo de Virtualização, os emuladores de videogame;
- **Virtualização Completa:** Possui consideráveis semelhanças conceituais à Virtualização por emulação, porém a Virtualização Completa não simula arquiteturas diferentes da máquina nativa, ou seja, ela está apta a receber, apenas, VMs trabalhando nos moldes da máquina física. Aceita

programas convencionais, ou seja, sem modificações em sua estrutura, em contrapartida, não permite acesso direto ao hardware, tornando-a mais lenta. A Virtualização Completa é representada por ferramentas como VirtualBox (ORACLE, 2013) e VMware (VMWARE, 2013).

- **Paravirtualização:** No conceito de Paravirtualização, o *Hypervisor* não necessita ofertar uma abstração completa dos recursos físicos abaixo dela, pois, os programas executados dentro de cada máquina virtual sofrem algumas alterações para serem processados neste tipo de ambiente. Essa possibilidade de interação direta com alguns componentes físicos faz desta abordagem a mais rápida dentre as três aqui citadas. Xen (FOUNDATION, 2013) é o principal exemplo de ambiente que utiliza Paravirtualização.

- **Virtualização de Sistema Operacional:** Consiste em substituir o Monitor de Máquinas Virtuais por um Sistema Operacional (SO) especial de tempo compartilhado e com capacidade de isolar recursos físicos entre as máquinas virtuais hospedadas por ele. Cada VM continua com suas próprias necessidades de *hardware* e *software*, porém, são gerenciadas pelo mesmo SO. Tal característica representa economia de recursos, pois todas as VMs compartilham o mesmo binário e o mesmo conjunto de instruções do Sistema Operacional, embora possam realizar atividades diferentes. Linux VServer (GELINAS, 2003) é um dos exemplos desse tipo de Virtualização.

2.2.1 *Live migration*

Conforme visto anteriormente, a técnica de Virtualização permite alocação e desalocação automática de *hardware* em resposta ao aumento ou redução na demanda de recursos por parte das VMs. Tal comportamento pode culminar em estados indesejados de sobrecarga ou de subutilização dos servidores. Por exemplo, um equipamento com a alta carga de trabalho está mais suceptível a falhas, enquanto uma máquina com baixa carga de trabalho pode representar gastos desnecessários com energia (BELOGLAZOV; BUYYA, 2012). Logo, faz-se necessário um mecanismo capaz movimentar VMs em execução para um balanceamento de cargas entre os equipamentos físicos em tempo real. Este mecanismo é denominado migração de máquinas virtuais.

O procedimento de migração de VMs consiste em transferir máquinas virtuais completas, ou seja, processos, estados de *software* e *hardware*, conexões de rede, dentre outros, de um determinado equipamento físico fonte para uma máquina destino específica. É desejável um processo imperceptível ao usuário, ou seja, a máquina virtual deve ser migrada sem alterações significativas em sua performace. Nesse caso, medidas como tempo total de migração e de *downtime* são indispensáveis para avaliar o desempenho das técnicas utilizadas.

O tempo total de migração consiste no período gasto para realizar uma migração completa de uma máquina virtual, ou seja, consiste na diferença entre o tempo decidido para migrar uma VM na máquina fonte e o tempo para reativá-la na máquina destino. Quanto menor o valor desta métrica, maior será a eficácia do procedimento de migração.

Já o período de *downtime* é a fatia de tempo onde a máquina virtual deve ser interrompida para uma migração ser completada. Não importa qual o modelo de migração usada, a VM sempre será suspensa para que as últimas atualizações em seu processamento sejam repassadas a máquina destino. Durante este período a VM não responderá às requisições do usuário, assim, para uma migração transparente, esta métrica deverá ser reduzida ao máximo.

Segundo (TAKEMURA; CRAWFORD, 2010), a migração pode ser segmentada em dois tipos principais: a *cold migration* ou a *live migration*. Na *cold migration* a máquina virtual em execução é suspensa no equipamento fonte, salva e enviada para a máquina física alvo onde, ao ser completada a transmissão, é reiniciada. Fica fácil perceber os elevados períodos de *downtime* da *cold migration*, tornado-a uma técnica limitada no ponto de migração invisível ao usuário.

A migração é considerada *live migration* quando uma parte de seu processamento ocorre com a máquina virtual ainda em plena execução. Esta migração pode ocorrer de maneira proativa (Pré-Cópia) ou sob demanda (Pós-Cópia)(SHRIBMAN; HUDZIA, 2012). As Figuras 4 e 5 representam, graficamente, o funcionamento destes dois tipos de *live migration*:

- **Pré-Cópia:** Os três primeiros passos estão diretamente relacionados com a parte proativa do processo. O primeiro deles consiste em escolher o melhor dispositivo alvo para receber a VM a ser migrada. O próximo passo é alocar todos os recursos solicitados pela VM no equipamento alvo escolhido no passo anterior. A parte proativa do processo tem fim com o envio parcial de informações referentes a VM, ou seja, periodicamente, são enviadas alterações ocorridas, especialmente na memória, desde o último turno de envio. O quarto passo consite em parar a VM em migração e enviar as últimas atualizações sofridas por ela. Ao seu final, existirá duas máquinas virtuais iguais em servidores diferentes. Os dois últimos passos estão associados, respectivamente, com a desalocação dos recursos utilizados pela VM migrada no dispositivo fonte e com a retomada das atividades da máquina virtual em seu novo equipamento físico.

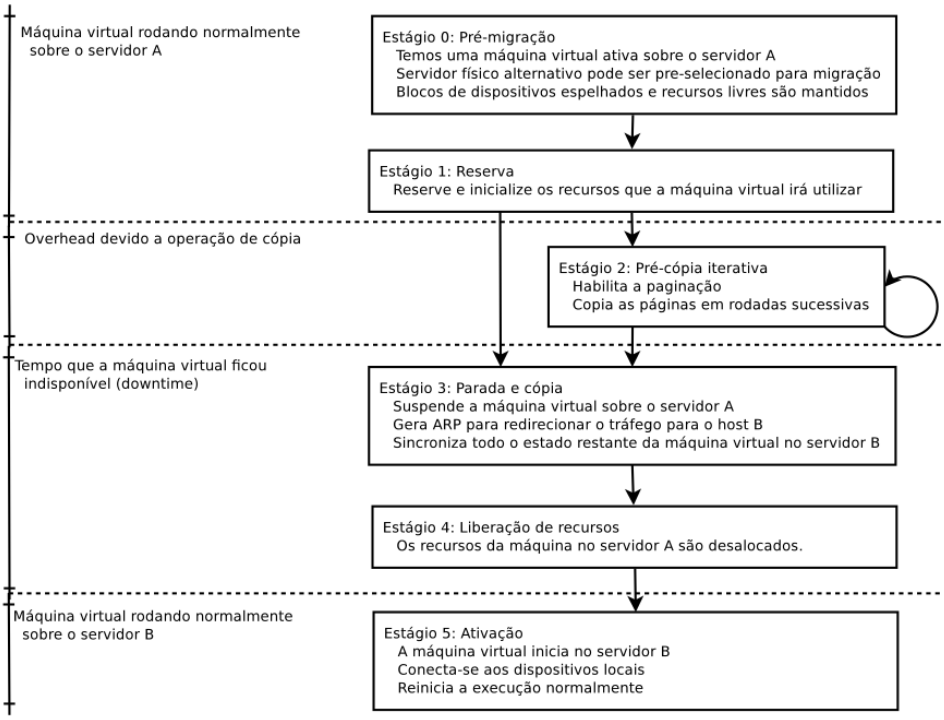


Figura 4 – Funcionamento da *live migration* pré-cópia (AMARANTE, 2013)

- **Pós-Cópia:** Esta abordagem consiste em requisitar sob demanda os dados necessários para o processamento. O primeiro passo consiste em parar a VM e migrar apenas uma parte fundamental dos dados para a continuidade do processamento na máquina destino. Ao fim desse passo, a VM é reativada em seu novo servidor. Caso alguma requisição por um dado ainda não migrado seja feita,

a VM é temporariamente parada e uma interrupção é gerada no *hypervisor* da máquina destino que a solicita, junto ao MMV do servidor fonte, o dado faltante. Com a chegada da informação pendente, a VM é novamente reativada. Este procedimento se repete sempre que algum dado solicitado não for encontrado.

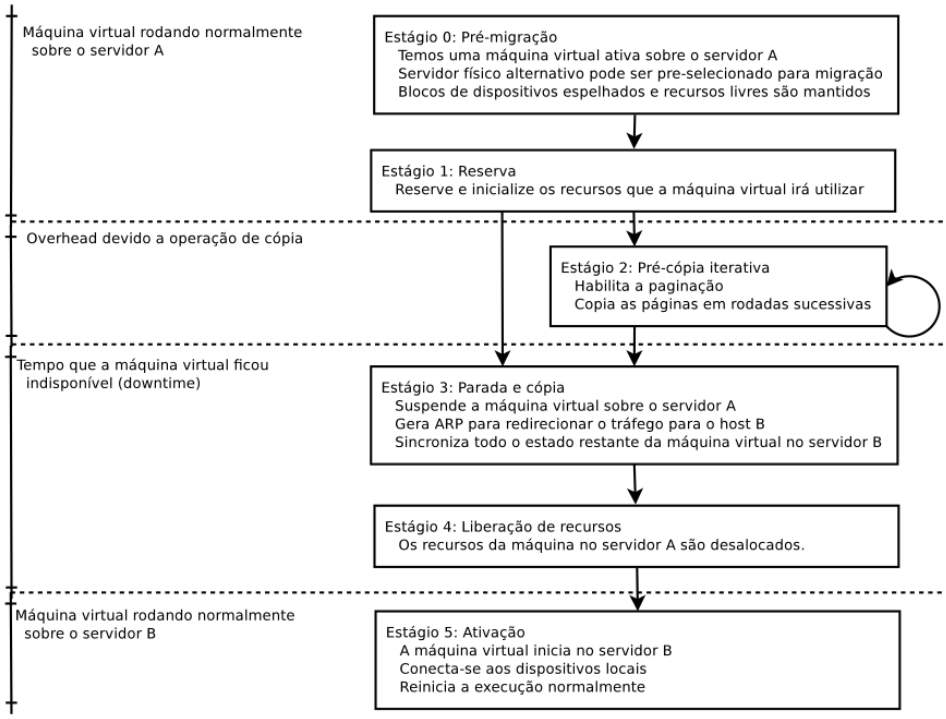


Figura 5 – Funcionamento da *live migration* pós-cópia (AMARANTE, 2013)

2.3 Computação verde

Nos dias atuais, tem-se observado com frequência a preocupação da população mundial em conservar o meio ambiente. Não só no ramo da Tecnologia da Informação como em todas as áreas do conhecimento, é visível o interesse de todos por produtos e ações ecologicamente corretas. Recentemente, foi criado o termo Computação Verde (*Green Computing*) para designar estudos e medidas, dentro da Ciência da Computação, voltados para questões ambientais. Estas práticas vão desde a reciclagem de componentes antigos ou falhos, até pesquisas voltadas para desenvolver equipamentos que consomem menos energia, sem perder desempenho.

2.3.1 Consumo de energia em um *data center*

Basicamente, um *data center* é composto por um grande número de servidores e equipamentos de rede responsáveis por, respectivamente, processar requisições e possibilitar a comunicação do *data center* com seus usuários. Contudo, como qualquer outro equipamento, todos estes computado-

res geram calor que, se não tratado, pode levar a falhas de funcionamento tais como reinicializações inesperadas, travamentos do software dentre outros, ou, até mesmo, a redução na vida útil destas máquinas (ADRIEL, 2009). Para evitar tais problemas, são instalados sistemas de resfriamento responsáveis por dissipar o calor produzido.

Para se ter uma idéia, estima-se um *data center* com, aproximadamente, 9000 metros quadrados com 1000 *racks* de computação padrão e um consumo energético médio de 10kW por *rack* precisa de um sistema de resfriamento orçamentado entre 2 a 5 milhões de dólares e, para mantê-lo em funcionamento é necessário gastar uma quantia aproximada entre 4 e 8 milhões de dólares (PATEL et al., 2003). Isso, sem falar nos gastos para manter os equipamentos computacionais do *data center* em atividade. Segundo (BELADY, 2007), nos últimos oito anos, percebeu-se um incremento de dezesseis vezes na relação performace e energia, ou seja, aumentou-se o desempenho computacional para cada Watt gasto. Ainda segundo o mesmo material, a previsão para 2014 é de, apenas, 25% dos custos para manter um *data center* serão relacionados a equipamentos de TI. Os outros 75% estarão ligados ao restante da infraestrutura, principalmente, o sistema de refrigeração.

Este aumento na performace pode levar a uma subutilização dos recursos computacionais. Estudos como (BARROSO; HöLZLE, 2007) comprovam que a grande maioria dos servidores operam muito abaixo de sua capacidade total, entre 10% a 50% de seu real desempenho. E trabalhos como (FAN; WEBER; BARROSO, 2007) afirmam que servidores, mesmo quando ociosos, chegam a consumir, aproximadamente, 70% de sua energia de pico, ou seja, com apenas uma pequena carga de seu trabalho gasta, no máximo, 30% menos energia do caso onde ele opera com toda sua capacidade. Foram feitos muitos esforços para amenizar tais problemas como, por exemplo, a Virtualização para atacar a subutilização de recursos físicos e a técnica de *Dynamic Voltage and Frequency Scaling* (DVFS) (SUEUR; HEISER, 2010), para aumentar a eficiência energética dos servidores.

A Virtualização, conforme visto anteriormente, permite cria instâncias virtuais de um equipamento que, compartilhando os recursos físicos desta máquina, são capazes de atender mais de um usuário, ao invés de apenas um no modelo tradicional. Fica fácil observar: quanto maior for a quantidade de máquinas virtuais alocadas em um único servidor, menor será a probabilidade de, em algum momento, algum de seus recursos físicos esteja ocioso. Por isso, muitas técnicas desenvolvidas buscam a consolidação de Máquinas Virtuais, objetivando alocar o maior número de VMs na menor quantidade de servidores possível (BELOGLAZOV; BUYYA, 2012) e (LAGO; MADEIRA; BITTENCOURT, 2011).

Contudo a consolidação de VMs deve ser feita de uma maneira cuidadosa e inteligente, pois podem levar a outros tipos de problemas. Por exemplo, um maior número de máquinas virtuais compartilhando os mesmos recursos criam condições de corrida culminando na perda de desempenho de algumas das VMs envolvidas. Além disso, a proximidade de servidores sobrecarregados pode acarretar na formação de zonas de calor intensas dentro do *data center* exigindo um gasto energético igual ou superior ao economizado com a consolidação de máquinas virtuais.

A técnica *Dynamic Voltage and Frequency Scaling* defende a idéia onde a CPU é capaz de regular sua voltagem e frequência de *clock* de forma proporcional a sua carga de trabalho, ou seja, quanto maior a quantidade de instruções para processamento, maior será a frequência de *clock*. Em casos onde a CPU apresenta-se ociosa ou com pouca carga de processamento, ela é capaz de reduzir sua voltagem, gastando assim uma menor quantidade de energia. Outra idéia também é colocar os servidores ociosos para hibernar, pois trata-se de um estado econômico de energia e possui um baixo tempo de reativação.

2.3.2 *Green cloud computing*

O aumento na demanda por serviços em Computação em Nuvens motivou o surgimento de novas empresas provedoras de serviços, bem como a expansão daquelas que já faziam parte deste mercado. A exigência constante dos consumidores por serviços de melhor qua-lidade afastou a atenção do mercado e da academia para o problema referente ao alto consumo energético dos *data centers*. Segundo (BUYYA; BELOGLAZOV; ABAWAJY, 2010), um *data center* chega a consumir a mesma quantidade de energia necessária para alimentar 25000 casas. Essas e outras estatísticas levaram governos e entidades não governamentais a pressionar os provedores de serviço a reduzir o consumo de energia dos seus *data centers*. Dentro deste contexto, surgiu o termo *Green Cloud Computing*.

A *Green Cloud Computing* é apenas um dos ramos da Computação Verde e, por isso, ambas compartilham os mesmos objetivos. Contudo, o termo em questão está mais focado em desenvolver mecanismos que reduzam o consumo energético dos *data centers*. Segundo (MURUGESAN, 2008), a produção de eletricidade contribui, significativamente, para mudanças climáticas no planeta, pois algumas de suas fontes geradoras (como carvão e óleo) não são limpas, ou seja, além de produzirem energia elétrica, emitem gases tóxicos e poluentes acumulando-se na atmosfera e contribuindo com o Efeito Estufa. Sem falar da energia nuclear que, se não manipulada corretamente, pode trazer problemas sérios a saúde, como, por exemplo, mutação ou câncer nas pessoas residentes próximas às suas usinas.

Os estudos realizados pela *Green Cloud Computing* são bastante diversificados. Vão desde pesquisas direcionadas a compreender a influência do posicionamento de servidores na formação de áreas de calor em um *data center* (SULLIVAN, 2002), até o desenvolvimento de técnicas para consolidar Máquinas Virtuais. Ressaltar a adoção de soluções como essas, além de auxiliar a preservar o meio ambiente, representa economia para os provedores de serviço e, talvez, até para os consumidores.

2.3.3 Elementos de uma arquitetura *green cloud computing*

O principal desafio da *Green Cloud Computing* é possibilitar a economia de energia em um *data center*, mas sem perder o desempenho da prestação dos serviços. Até então, desenvolveu-se muitas técnicas e componentes sofisticados visando, unicamente, a melhoria de performace das máquinas virtuais, enquanto as questões energéticas foram deixadas de lado. Dessa forma, faz-se necessário uma arquitetura da Nuvem replanejada para permitir a implementação de novos mecanismos voltados para solucionar os problemas pendentes. (BUYYA; BELOGLAZOV; ABAWAJY, 2010) sugere uma arquitetura verde para a Nuvem (Figura 6):

- **Consumidores:** São compostos por todos aqueles que utilizam os serviços de Computação em Nuvens. (BUYYA; BELOGLAZOV; ABAWAJY, 2010) faz uma diferenciação entre consumidores e usuários de serviços na Nuvem, o primeiro refere-se a entidades que desejam usar a Nuvem para oferecerem seus aplicativos como serviços, enquanto o segundo está relacionado aqueles que vão acessar a Nuvem para utilizar tais serviços;
- **Alocador Verde de Recursos:** Esta entidade opera como uma mediadora entre a infraetrutura da Nuvem e seus consumidores. Além de visar uma prestação de serviço satisfatória, com baixas

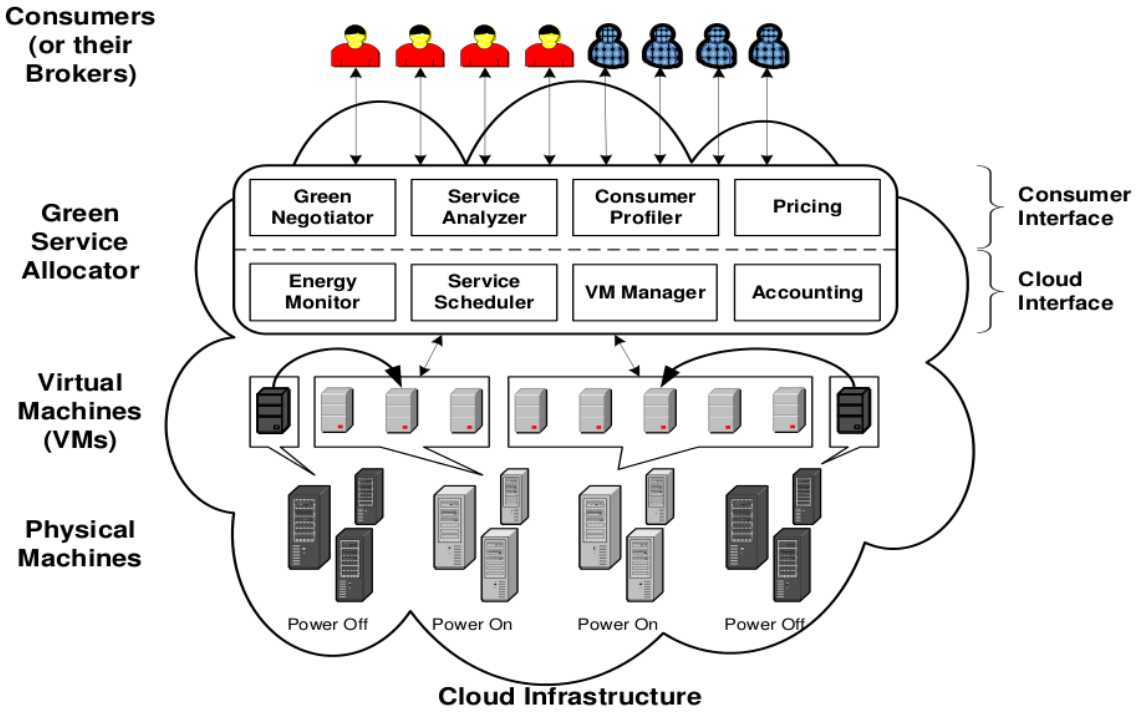


Figura 6 – Arquitetura *Green Cloud Computing* (BUYYA; BELOGLAZOV; ABAWAJY, 2010)

violações de SLA, por se tratar de um escalonador verde, ele também irá ponderar sobre questões relacionadas, principalmente, ao consumo de energia visando sua economia. Ela é composta pelos seguintes componentes:

- *Negociador Verde:* Responsável por definir o SLA entre o consumidor e o provedor do serviço em questão. Para isso ser possível, este componente precisará conhecer as necessidades do consumidor, bem como o que o provedor tem a oferecer. Com base nessas informações, haverá uma negociação entre as partes envolvidas e, ao seu final, um SLA será produzido;
- *Analisador de Serviço:* Entidade cuja a responsabilidade é aceitar ou rejeitar requisições feitas por consumidores. De posse do SLA definido para o serviço e informações oriundas de outros componentes: Gerente de VM e Monitor de Energia. O Analisador de Serviço dará o seu aval se o consumidor pode, ou não, usurfruir do serviço;
- *Perfil de Consumidor:* Responsável por implementar um esquema de privilégios entre os consumidores. Melhor explicando, pode ser implementado um esquema que crie classes de usuários, onde algumas classes tem prioridades sobre as outras. Este componente é responsável por definir a qual classe o consumidor pertence e se ele terá preferência sobre algum outro consumidor;
- *Tarifador:* Componente responsável por monitorar o uso de recursos de cada consumidor e, junto com os valores definidos através do SLA, tarifar o serviço prestado. Segundo o mesmo material, devido a este monitoramento, ele se torna uma ferramenta importante para a alocação de consumidores prioritários;
- *Monitor de Energia:* Observa o nível de carga de cada máquina física e determina quais delas merecem ser ligadas ou desligadas;
- *Escalonador de Serviço:* Instancia e aloca os recursos requisitados de uma Máquina Virtual, além de entregar a VM recém criada ao consumidor a fim de executar o serviço. Esta entidade

também é responsável por remover máquinas virtuais ociosas com o intuito de liberar recursos e, por consequência, aumentar oferta;

  - *Gerente de VM:* Realiza o monitoramento das máquinas virtuais com o intuito de garantir que os recursos oferecidos a elas estão compatíveis com o acordado anteriormente e manter as máquinas virtuais sempre disponíveis. Também é incubida de coordenar a migração de VMs entre servidores;
  - *Contador:* Guarda as informações atuais e o histórico de consumo do recursos do *data center*. Estes dados podem ser úteis para auxiliar políticas de seleção de máquinas virtuais para migração, mecanismos de escolha de servidores destinos para receber VMs migradas e, até mesmo, técnicas para montar perfis de consumo úteis para fazer predição da utilização de recursos.

- **Máquinas Virtuais:** Instâncias lógicas de máquinas físicas responsáveis por executar serviços requisitados pelos consumidores. Múltiplas VMs podem coexistir em um mesmo servidor. Para isso, cada uma delas recebe componentes virtualizados que representam partições dos recursos reais. Oferecem mecanismos de realocação dinâmica de forma proporcional a suas necessidades por *hardware*. Estas entidades também podem ser migradas visando uma maior ou menor distribuição de carga de trabalho entre os servidores;
- **Máquinas Físicas:** Compostas por servidores responsáveis por prover a infraestrutura de *hardware* necessária para hospedar máquinas virtuais e executação de serviços.

2.4 Migração de máquinas virtuais em computação nas nuvens

Em um ambiente de Computação em Nuvens, os clientes utilizam-se da infraestrutura dos provedores de serviço para realizar suas computações. O provedor de serviço, por sua vez, visando aumentar o nível de utilização de sua infraestrutura, cria máquinas virtuais para atender às requisições dos clientes e as alocam em algum servidor de seu *data center*, onde elas irão compartilhar os recursos físicos deste servidor para realizar os seus processamentos.

Devido a característica de elasticidade, onde as máquinas virtuais podem obter ou liberar recursos de acordo com suas necessidades, a alocação inicial pode se tornar ineficiente com o passar do tempo, pois um aumento na demanda pode criar condições de disputa pelo *hardware* entre as VMs. Essa competição pode levar ao mal funcionamento das máquinas virtuais e comprometer a qualidade do serviço prestado. No pior caso, pode acarretar no pagamento de multas aos clientes prejudicados. Dessa forma, é necessário um mecanismo que evite a ocorrência de tal problema.

A técnica aconselhada é a migração de máquinas virtuais, onde é possível tirar uma VM de seu servidor origem e enviá-la, através da rede, para uma máquina física destino com maior disponibilidade de recursos sem a necessidade de desligar a máquina virtual. Isso proporciona uma melhor distribuição das cargas de trabalho por todo o *data center* e evita a sobrecarga de seus servidores que, por consequência, reduz a probabilidade de condições de disputa entre as VMs. O problema central da migração de máquinas virtuais está na escolha da VM a ser migrada, pois este procedimento tem de ser feito de forma inteligente para reduzir o número de migrações necessárias para o balanceamento de carga entre os servidores. Uma grande quantidade de migrações pode culminar em uma sobrecarga na rede e perdas de desempenho das máquinas virtuais dos clientes. Trabalhos como (BELOGLAZOV;

BUYYA, 2012), (BUYYA; BELOGLAZOV; ABAWAJY, 2010), (DO et al., 2011) e (VERMA et al., 2009) propõe mecanismos que auxiliam na escolha de máquinas virtuais para migração. Tais trabalhos serão explicados com detalhes no Capítulo 3.

O nível ideal de carga de trabalho dos servidores é divergente se observado do ponto de vista do desempenho e da energia. De uma forma geral, um grande número de servidores sobrecarregados representa uma maior consolidação das máquinas virtuais. Isto culmina em economia de energia para o *data center*, uma vez que, com uma maior concentração de VMs em um grupo de servidores, os demais estarão livres para serem postos em modo de hibernação e, com isso, requisitar menos energia. Conforme visto anteriormente, servidores sobrecarregados ocasionam perda de perfomace para as máquinas virtuais neles hospedadas e, em alguns casos, violações de SLA, enquanto servidores subutilizados representam uma grande quantidade de recursos disponíveis para alocação e uso, o que pode representar um aumento de desempenho das VMs significativo. Resumindo, máquinas sobrecarregadsas são boas para a energia e ruins para a performace (SLA), enquanto o contrário acontece com máquinas subtutilizadas.

Este trabalho visa abordar o problema da escolha de máquinas virtuais para migração em um ambiente de Computação em Nuvens tendo, como objetivos, a economia de energia do *data center*, sem comprometer o desempenho das máquinas virtuais envolvidas significativamente. Como técnica para solucionar este problema, será utilizado o modelo de Regressão Linear Múltipla na decisão de qual máquina virtual tem maior contribuição na energia do seu servidor hospedeiro. Tal modelo será explicado nas seções subsequentes.

2.5 Regressão

Segundo Jain (1991), o modelo de regressão permite realizar estimativas ou predições de uma variável aleatória em função de diversas outras variáveis, desde que exista um relacionamento entre as variáveis envolvidas. Técnicas de regressão são bastante versáteis, pois podem ser aplicadas em grupos de duas ou mais variáveis (desde que exista apenas uma a ser estimada), variáveis com relacionamento linear ou não linear e etc. Entretanto, esta praticidade exige um cuidado maior durante a fase de modelagem do problema para que não conduza o pesquisador a análises erradas. O modelo de regressão pode ser classificado em:

- **Regressão Linear Simples:** Recebe duas variáveis de entrada, sendo uma dependente e a outra independente, com um relacionamento linear entre elas;
- **Regressão Linear Múltipla:** Recebe três ou mais variáveis de entrada, sendo uma dependente e as demais independentes, com um relacionamento linear entre elas;
- **Regressão Não Linear Simples:** Recebe duas variáveis de entrada, sendo uma dependente e a outra independente, com um relacionamento não linear entre elas;
- **Regressão Não Linear Múltipla:** Recebe três ou mais variáveis de entrada, sendo uma dependente e as demais independentes, com um relacionamento não linear entre elas;

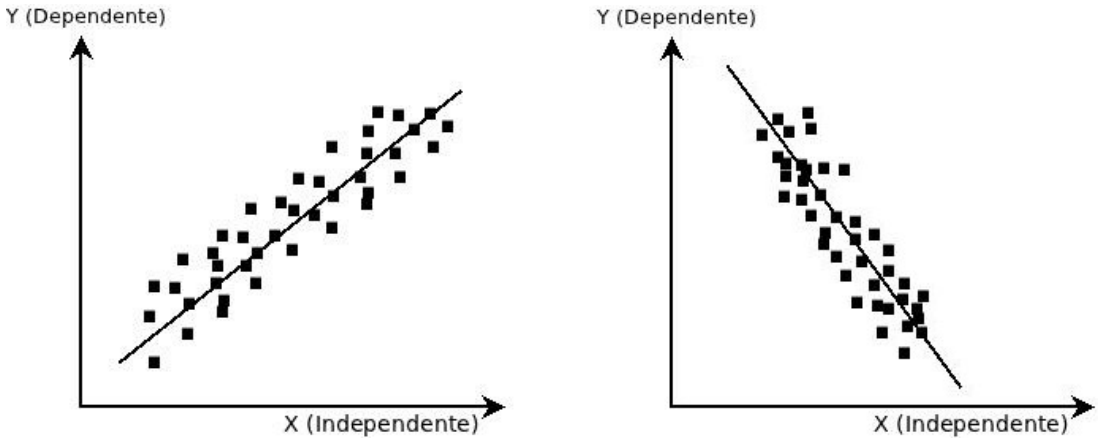
O modelo de Regressão Linear Múltipla (RLM) deverá ser utilizado neste trabalho para determinar, dentre as máquinas virtuais hospedadas em um dado servidor, aquela que, com seu

processamento, contribui mais para o consumo de energia de seu equipamento hospedeiro. Contudo, para facilitar o seu entendimento, será estudado, inicialmente, a técnica de Regressão Linear Simples (RLS) um modelo que, ao contrário da RLM, recebe, como entrada, apenas duas variáveis (uma Variável Dependente e outra Independente). Posteriormente, tomando como base a RLS, será explicado as particularidades da Regressão Linear Múltipla.

2.5.1 Regressão linear simples (RLS)

Este é o modelo mais simples de regressão, pois necessita de apenas duas entradas que se relacionam linearmente. Contudo, apesar de toda esta simplicidade, a RLS ainda possui algumas exigências. Em primeiro lugar, a técnica recebe, como entrada, um grupo com somente duas variáveis: uma usada para a predição e uma a ser estimada. O modelo também assume um relacionamento linear entre as variáveis envolvidas. Por fim, as variáveis devem ser quantitativas (representadas por números) e não necessariamente devem ser da mesma natureza (valores sob a mesma unidade de medida).

Cada variável na Regressão Linear Simples recebe uma nomeclatura especial que representa bem sua função no modelo. A variável usada para a predição é chamada de preditora, fator ou variável independente, enquanto aquela que se deseja estimar é conhecida como variável resposta ou dependente. A relação entre essas variáveis pode ser direta (positiva) ou inversa (negativa). O relacionamento será classificado como direto quando a variável dependente acompanha o crescimento da variável independente, ou seja, à medida que uma cresce, a outra também aumenta. Já em uma relação inversa ocorre o contrário, ou seja, à medida que a variável dependente aumenta, a variável independente diminui. Essa classificação é exemplificada graficamente nas Figuras 7(a) e 7(b).



(a) Relação Direta (b) Relação Inversa

Figura 7 – Diagrama de Dispersão das amostras obtidas

A Figura 7 mostra uma importante ferramenta visual para a análise do relacionamento entre todas as variáveis envolvidas em qualquer modelo de regressão. O Diagrama de Dispersão é composto por um plano cartesiano, onde o eixo das abscissas (X) e o eixo das ordenadas (Y) representam, respectivamente, as variáveis independente e dependente do modelo. Cada ponto no plano representa uma amostra coletada a partir de observações reais realizadas previamente. As variáveis de relacionamento linear, como as das Figuras 7(a) e 7(b), tem, em sua ”nuvem” de pontos, uma tendência

a se aproximar de uma reta, enquanto as não lineares se aproximam de uma curva.

Infelizmente, nem sempre, a reta encontrada que resume o relacionamento entre as variáveis envolvidas (também conhecida como reta de regressão) é 100% precisa. Muitas vezes, os valores previstos pela reta não são compatíveis com aqueles coletados a partir de observações reais. O resíduo ou erro é a diferença entre o valor real coletado e o obtido através do modelo (simbolizado pela linha vertical tracejada na Figura 8). Alguns erros podem ser positivos (valor estimado é menor que o valor observado), enquanto outros são negativos (valor estimado maior que o observado). A presença desses erros significa que:

1. As variações da variável independente não conseguem explicar perfeitamente as variações da variável dependente;
2. As amostras obtidas das variáveis apresentam distorções em relação a realidade;
3. A variável dependente depende de outras variáveis independentes.

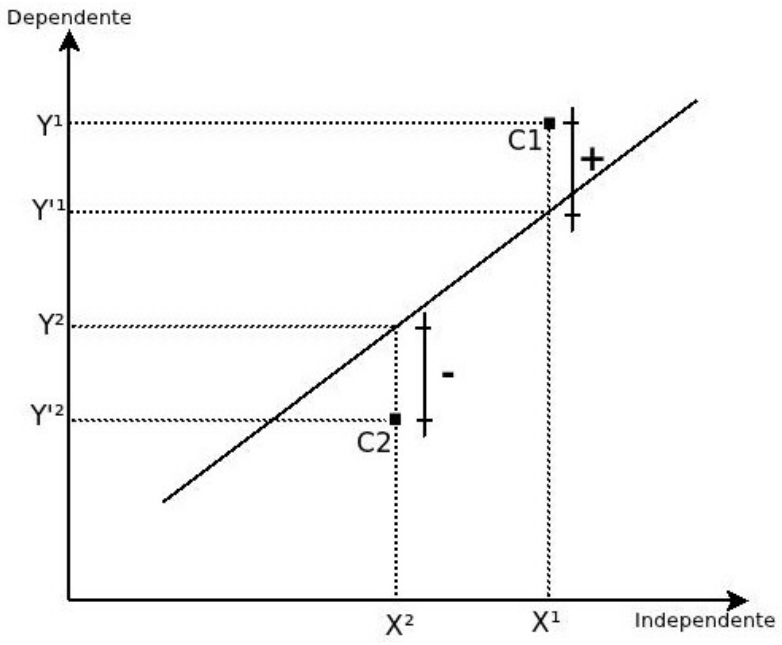


Figura 8 – Erro de estimativa da reta de regressão

A técnica consiste, de uma maneira geral, em receber um conjunto de amostras de dados das variáveis (um da variável independente e outro da dependente) e encontrar a função que melhor descreve o relacionamento entre elas. No caso da RLS, a função encontrada será uma equação de 1º grau e descreverá uma reta no plano cartesiano onde os eixos serão formados pelas duas variáveis de entrada do modelo. A função terá a seguinte fórmula geral:

$$Y = \alpha + \beta X + \epsilon \tag{2.1}$$

Onde:

$Y =$ Variável Dependente

$\alpha, \beta =$ Parâmetros de Regressão

$X =$ Variável Independente

$\epsilon =$ Erro ou Resíduo

Para determinar a equação de 1º grau que melhor descreve o comportamento entre as variáveis, é necessário definir os valores de α e β encontrados na equação 2.1. Comumente, é utilizado o Método dos Mínimos Quadrados. Essa técnica, tem como objetivo, definir uma função que minimiza a distância ao quadrado entre os valores observados e os estimados através do modelo. Essa distância é, justamente, o que foi definido anteriormente como erro ou resíduo. Logo, o Método dos Mínimos Quadrados retorna a função que apresenta a menor margem de erro em relação a todos os valores observados. A formulação matemática desse problema é assim descrita:

$$\textit{Minimizar } \epsilon_{total} \tag{2.2}$$

Onde:

$$\epsilon_{total} = \epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2 + \cdots + \epsilon_n^2$$

$$\epsilon_n = (Y_n - Y'_n)$$

n = nº de amostras coletadas

Onde o Erro ou Resíduo ϵ é dado pela diferença entre o valor real coletado (Y_n) e o valor obtido através do modelo de Regressão (Y'_n). Tais valores serão melhor explicados nos parágrafos conseguintes.

Através do Método dos Mínimos Quadrados é possível encontrar a função que melhor descreve o comportamento das duas variáveis envolvidas, dado um conjunto de amostras. Contudo, nem sempre esta reta de regressão retornada poderá ser útil para uma avaliação da realidade. O fato da equação retornada minimizar os erros não implica que os valores desses resíduos serão pequenos, o que poderia implicar em análises e previsões erradas. Assim, se faz necessário uma forma de mensurar o grau de utilidade da reta de regressão, ou seja, o quão boa é a predição sugerida por ela.

O Coeficiente de Determinação (R^2) mensura a qualidade das previsões encontradas pelo modelo de regressão em relação aos dados reais obeservados. Mas antes de defini-lo formalmente, faz-se necessário apresentar algumas métricas que irão facilitar o seu entendimento. Veja a Figura 9.

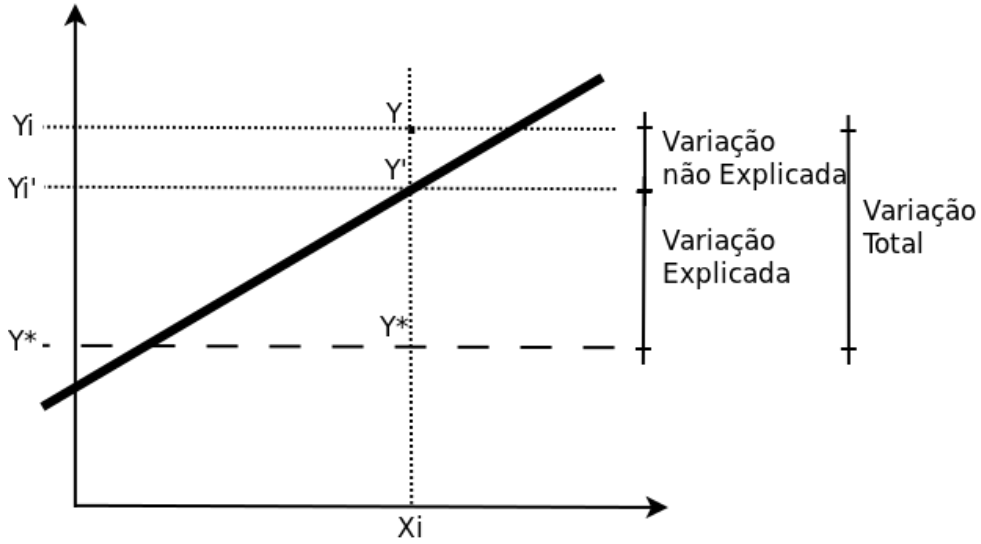


Figura 9 – Métricas de Variação de Dados em RLS

A pior forma de estimativa da variável resposta é através da média (Y*) dos seus dados coletados na amostra, pois embora seja um valor fácil de ser calculado, dificilmente será condizente com a realidade para todos os valores da variável preditora. Contudo, tal estimativa é usada como base para

analisar a qualidade da função encontrada pela RLS. O objetivo da técnica de regressão é melhorar o máximo possível esta estimativa base, aproximando o resultado previsto (Y') dos valores coletados (Y).

Contudo, conforme ressaltado anteriormente, a técnica de regressão não consegue ser 100% precisa em todos os casos. Muitas vezes existirá uma diferença entre o valor observado (Y) e o valor estipulado pelo modelo (Y') como o caso mostrado na Figura 9. Esse resíduo representa uma parte do valor real observado que, infelizmente, a regressão não foi capaz de prever, mas que também não desqualifica o seu uso (se em pequenas proporções).

Com base nas considerações acima e na Figura 9, define-se:

- **Variação Total:** Somatório ao quadrado da diferença entre os valores reais observados Y e o valor médio amostral Y*. Quantifica o total de erros encontrados se a média amostral fosse usada como estimativa;

$$Var_{Total} = \sum_{k=0}^{N} (Y_k - Y*_k)^2, \text{ onde } N = \text{nº de amostras} \tag{2.3}$$

- **Variação Explicada:** Somatório ao quadrado da diferença entre os valores Y' previstos através do modelo e o valor médio amostral Y*. Quantifica a melhoria geral que a RLS realizou na pior forma de estimativa;

$$Var_{Exp} = \sum_{k=0}^{N} (Y'_k - Y*_k)^2, \text{ onde } N = \text{nº de amostras} \tag{2.4}$$

- **Variação Não Explicada:** Somatório ao quadrado da diferença entre os valores reais coletados Y e os valores estimados Y'. Representa o total de resíduos encontrados quando a função obtida no modelo é aplicada.

$$Var_{NExp} = \sum_{k=0}^{N} (Y_k - Y'_k)^2, \text{ onde } N = \text{nº de amostras} \tag{2.5}$$

Conforme ressaltado anteriormente, o Coeficiente de Determinação representa a qualidade da função encontrada via Regressão Linear Simples. Trata-se de uma medida normalizada (varia de 0 a 1) que quanto maior a precisão da função (ou seja, quanto menor for sua taxa de erros), mais próximo a 1 se encontrará o seu valor. Uma reta de boa qualidade consegue explicar uma boa parte da diferença entre o valor real observado e a média amostral. Dessa forma, o Coeficiente de Determinação representa o pedaço da Variação Total (Equação 2.3) que a função encontrada conseguiu explicar (Equação 2.4) e dada pela seguinte fórmula:

$$R = \frac{Var_{Exp}}{Var_{Total}} \tag{2.6}$$

2.5.2 Regressão linear múltipla

Na seção anterior foi estudado a Regressão Linear Simples, um modelo de regressão que recebe, como entrada, amostras de duas variáveis (sendo uma independente e a outra dependente) e retorna uma reta (chamada reta de regressão) que melhor descreve a relação entre as variáveis envolvidas. O fato da reta de regressão representar a melhor solução encontrada não implica que ela

seja uma ferramenta útil para análises e predições em todos os casos, pois os erros de estimativa podem ser significativamente grandes e conduzir o pesquisador a conclusões erradas.

Também foi apontado que um dos motivos para a existência desses resíduos é a dependência da variável resposta a outras variáveis preditoras que não foram consideradas na modelagem do problema. Ou seja, a única variável dependente está sujeita a mais de uma variável independente e foi justamente a falta desses outros dados que levou a regressão a uma solução tão fraca. Assim, se faz necessária uma técnica que também considere todas as demais variáveis identificadas no problema.

A Regressão Linear Múltipla vem para atender esta necessidade. Assim, o modelo da RLM é composto por duas ou mais variáveis preditoras e apenas uma variável resposta. O modelo em questão herda muitas definições da Regressão Linear Simples (JAIN, 1991), contudo apresenta esses mesmos conceitos com um ganho em complexidade. Por exemplo, o Diagrama de Dispersão na RLS (Figura 7) é um plano cartesiano comum formado por dois eixos. Já o Diagrama de Dispersão na RLM, é composto por *n* eixos, onde *n* é o número de variáveis envolvidas. A Figura 10 mostra um Diagrama de um modelo com duas variáveis independentes (X1 e X2).

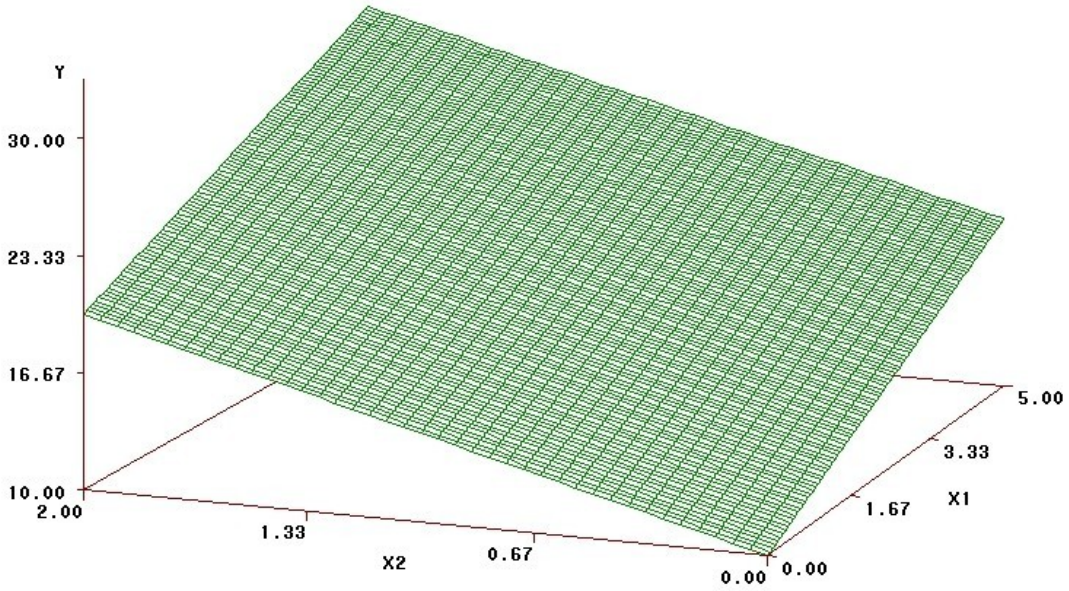


Figura 10 – Métricas de Variação de Dados em RLS

A fórmula geral do modelo de Regressão Linear Múltipla é muito semelhante à sua antecessora (vista na Equação 2.7), entretanto ela recebe alguns termos a mais que representam as novas variáveis independentes inseridas. Logo, a função geral da RLM é:

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \epsilon \tag{2.7}$$

Onde:

$Y =$ Variável Dependente

$\alpha, \beta_1, \ldots, \beta_n =$ Parâmetros de Regressão

$X_n =$ n-ésima Variável Independente

$\epsilon =$ Erro ou Resíduo

Assim como a Regressão Linear Simples, a técnica de RLM consiste em receber um conjunto de amostras colhidas apartir de observações reais e encontrar os Parâmetros de Regressão

$(\alpha, \beta_1, \ldots, \beta_n)$ de uma função que melhor descreve o relacionamento entre as variáveis independentes e a dependente. Dado um conjunto com *m* amostras o problema consistirá em resolver o seguinte sistema de equações:

$$\begin{cases} y_1 = \alpha + \beta_1 x_{11} + \beta_2 x_{12} + \cdots + \beta_n x_{1n} + \epsilon_1 \\ y_2 = \alpha + \beta_1 x_{21} + \beta_2 x_{22} + \cdots + \beta_n x_{2n} + \epsilon_2 \\ y_3 = \alpha + \beta_1 x_{31} + \beta_2 x_{32} + \cdots + \beta_n x_{3n} + \epsilon_3 \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \ldots \\ y_m = \alpha + \beta_1 x_{m1} + \beta_2 x_{m2} + \cdots + \beta_n x_{mn} + \epsilon_m \end{cases} \tag{2.8}$$

O sistema apresentado na Equação 2.8 também pode ser visto em um formato matricial. Assim:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & x_{13} & \ldots & x_{1n} \\ 1 & x_{21} & x_{22} & x_{23} & \ldots & x_{2n} \\ 1 & x_{31} & x_{32} & x_{33} & \ldots & x_{3n} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{m1} & x_{m2} & x_{m3} & \ldots & x_{mn} \end{bmatrix} \times \begin{bmatrix} \alpha \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \beta_m \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \vdots \\ \epsilon_5 \end{bmatrix}$$

É importante ressaltar que, por se tratar de um sistema de equações, se faz importante observar a forma como o sistema é classificado antes do sistema ser solucionado. Existem três tipos (VINCENZO; VISSOTO; LAUREANO, 1994):

- **Sistema Impossível:** Não é possível encontrar soluções para todas as incógnitas;
- **Sistema Possível e Indeterminado:** Existem infinitas soluções para as variáveis;
- **Sistema Possível e Determinado:** Existe apenas uma única solução;

Do ponto de vista da Regressão Linear Múltipla, apenas Sistemas Possíveis e Determinados são interessantes, pois Sistemas Impossíveis não apresentam soluções e Sistemas Possíveis e Indeterminados retornam soluções que ainda dependem de uma ou mais variáveis, o que, no caso da RLM, representaria a indeterminação de um ou mais Parâmetros de Regressão.

2.6 Sumário do capítulo

Neste capítulo, foram apresentados os conceitos relacionados à Computação em Nuvens, à Virtualização, à *Green Cloud Computing* e ao modelo de Regressão Linear Múltipla, onde foram abordadas questões relevantes ao desenvolvimento e entendimento do algoritmo proposto neste trabalho.

No capítulo a seguir, será apresentada uma fundamentação sobre os trabalhos relacionados com a alocação de máquinas virtuais, bem como técnicas voltadas para detectar a sobrecarga de servidores e auxiliar na escolha de VMs para migração.

3 TRABALHOS RELACIONADOS

Este capítulo apresenta uma revisão de alguns trabalhos importantes sobre o problema de alocação e realocação de máquinas virtuais em um ambiente de *data center*. Esses trabalhos apresentam técnicas voltadas tanto para minimizar o consumo energético do *data center*, como, principalmente, otimizar o desempenho das máquinas virtuais e evitar violações de SLA por carência de recursos físicos a serem oferecidos pelos servidores.

3.1 Utilizando o problema de múltiplas mochilas para modelar o problema de alocação de máquinas virtuais em computação em nuvens

Amarante (2013) afirma que o problema de alocação de máquinas virtuais é um dos principais pontos para a utilização de Computação em Nuvens. Além de ser classificado como um problema difícil (*NP-Hard*) (BELOGLAZOV; BUYYA, 2012), a decisão de distribuição de VMs em servidores deve ser feita de uma maneira rápida, para não prejudicar a prestação do serviço. No trabalho em questão, a alocação de máquinas virtuais foi mapeada como um problema de múltiplas mochilas (*multiple knapsack problem*) (MARTELLO; TOTH, 1990) e (KELLERER; PFERSCHY; PISINGER, 2004).

O problema das múltiplas mochilas (Figura 11) se assemelha bastante ao problema da mochila simples. A diferença básica entre eles está na presença de não apenas uma, mais de duas ou mais mochilas. O problema consiste em colocar a quantidade mais lucrativa de objetos, cada qual com um determinado peso e valor, dentro das mochilas, que suportam um peso máximo definido, desde que nenhuma mochila receba mais objetos suportados por ela e que um item só pode ser posto em uma mochila (MARTELLO; TOTH, 1990). Existem diversas heurísticas para resolver tal problema. Em (AMARANTE, 2013), foi escolhido o algoritmo baseado em colônia de formigas (DORIGO; BIRATTARI; STüTZLE, 2006).

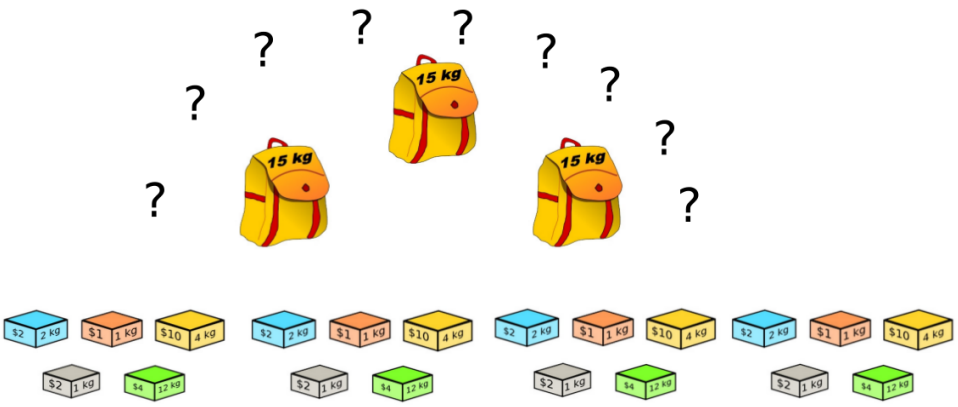


Figura 11 – Problema das Múltiplas Mochilas

O mapeamento entre o problema das múltiplas mochilas e a alocação de máquinas virtuais é feita da seguinte forma:

- Os servidores representarão as mochilas e seus pesos definidos serão suas respectivas capacida-

des máximas de CPU;

- As máquinas virtuais serão os objetos e seus pesos a quantidade de processamento definida pelo SLA;
- Ao contrário do problema original, a alocação de máquinas virtuais objetivará a minimização do consumo de energia e não a maximização do lucro, ao colocarmos os objetos.

O algoritmo proposto recebe, como entrada, o conjunto de máquinas virtuais a serem alocadas e uma lista com as capacidades disponíveis de cada servidor. Também são definidas, previamente, a quantidade de iterações e de formigas (soluções) geradas por ciclo de execução. Após a criação das formigas, o primeiro passo é definir o conjunto de VMs aptas a serem alocadas em um dado servidor. Entenda esta aptidão como as Máquinas Virtuais com seus recursos suportados pelo servidor em questão. Contudo, apenas uma dessas VMs pode ser alocada. Essa escolha é feita através de um mecanismo de roleta: cada máquina virtual candidata tem uma probabilidade de seleção que, após serem somadas e o resultado encontrado, dividir cada uma destas probabilidades, gera uma porcentagem vinculada a cada VM. Então, gera-se um número aleatório entre 0 e 1 que escolherá a máquina virtual referente à faixa onde ele caiu.

Escolhida a máquina virtual, ela é retirada do conjunto de máquinas virtuais a serem alocadas e inserida em um conjunto solução. O servidor, por sua vez, tem seu nível de recursos disponíveis reduzido. Este procedimento se repete enquanto houver máquinas virtuais a serem alocadas. Construída uma solução, ela é então comparada, se existir, com a melhor já encontrada. Se a atual for a primeira ou a mais atrativa, ela passará a ser a melhor solução e uma nova formiga será criada. Atingido o número máximo de formigas, tem-se fim a etapa de construção da solução.

O próximo passo é a etapa de atualização do feromônio. Os feromônios referentes a cada um dos pares VM e servidor determinado na melhor solução são reforçados, enquanto os demais sofrem uma evaporação. Há ainda um último passo nesta etapa para manter os níveis de feromônios dentro de uma faixa pré-estabelecida contendo um limiar máximo e um limiar mínimo. Finalizada esta fase, uma nova iteração começa e todo este procedimento é reiniciado.

Embora apresente excelentes resultados e se trate de uma boa proposta para realizar a alocação de máquinas virtuais, a ideia defendida nesse trabalho apresenta uma grave limitação quando utilizada para a realocação de VMs, pois ela analisa, assim como outras propostas anteriores, apenas o consumo de momento das máquinas virtuais e não o seu histórico, já disponível no caso de uma realocação. O algoritmo também é um pouco lento, podendo levar a degradação de performace no caso de migrações de VMs.

3.2 Utilizando perfis de aplicações para alocação de máquinas virtuais em nuvens

Além de técnicas voltadas para a economia de energia em servidores e, por consequência, em todo o *data center*, existem algumas propostas concebidas com o intuito de melhorar as questões de performace das Máquinas Virtuais. Quanto maior o nível de trabalho em um servidor, maiores são as probabilidades de falhas no sistema e de degradação no desempenho das VMs, visto que aumentará a disputa entre elas pela utilização dos recursos físicos reais. Assim, faz-se importante o desenvolvimento

de técnicas, preferencialmente proativas, ou seja, tomam medidas antecipadas para evitar sobrecarga de servidores, de balanceamento de carga entre os equipamentos constituintes de um *data center*.

Uma das técnicas adotadas para auxiliar estes mecanismos proativos é a Análise de Correlação Canônica (*Canonical Correlation Analysis* ou CCA). De forma resumida, a CCA consiste em observar o comportamento do sistema por um tempo determinado e, com base em alguns dados coletados, extrair, através de um modelo matemático, informações como o nível de relacionamento entre as variáveis observadas e, em alguns casos, a predição do comportamento do sistema observado. Esse processo se repete periodicamente. Para mais informações sobre a CCA consultar (JAIN, 1991) e (CLARK, 1975).

Durante a alocação ou, até mesmo, a realocação de máquinas virtuais em servidores, é de suma importância que os SLAs envolvidos sejam obedecidos. Estes SLAs compreendem tanto aquele relacionado à Máquina Virtual a ser alocada no servidor destino, como também as das VMs já hospedadas nesse servidor. Novas VMs, ao serem alocadas em servidores que já hospedam máquinas virtuais, podem prejudicar o funcionamento das demais e, inclusive, levar a falhas de SLA.

A proposta de Do et al. (2011) busca aplicar o método de Análise de Correlação Canônica de duas formas: 1) Encontrar perfis de utilização dos recursos e usá-los para prever o comportamento das VMs; 2) Usar como ferramenta para auxiliar na escolha de servidores destino para máquinas virtuais que estão sofrendo migração. A primeira opção é a mais relevante para este trabalho, pois pode ser utilizada na escolha de máquinas virtuais para migração. Assim, somente o procedimento deste item será explicado aqui.

O primeiro passo na proposta de (DO et al., 2011) é selecionar e separar as variáveis do sistema em dois grupos principais: o primeiro voltado para as métricas de performace da aplicação tratado como a variável independente no modelo CCA, enquanto o segundo responderá pelas métricas de performace do sistema e, por sua vez, será vinculado a variável dependente. A Tabela 1 ilustra este primeiro passo.

Tabela 1 – Tabela simplifada da proposta de (DO et al., 2011)

Métricas da aplicação	Métricas do sistema

As variáveis relacionadas à aplicação são específicas para cada programa e podem ser utilizadas como métricas de análise da performace do serviço prestado definidas pelo SLA. Por exemplo, uma máquina virtual que executa um servidor Apache pode ter, como variáveis de aplicação, o número de requisições HTTP processadas por segundo e/ou o período necessário para atender uma ou mais requisições HTTP.

O segundo grupo representa as métricas que sintetizam o desempenho do sistema durante o período de observação. No trabalho em questão as variáveis escolhidas são: Níveis de utilização de CPU (CPU) e memória (MEM), Velocidades de leitura (I/O$_R$) e escrita (I/O$_W$) de I/O. Este conjunto é formado por dois subgrupos: o primeiro refere-se às métricas de performace observadas na máquina virtual dona da aplicação cujos valores se encontram na variável independente do modelo CCA, chamada de *foreground metrics*, enquanto o segundo representa o somatório dos valores coletados por todas as demais VMs, chamada de *background metrics*. Cada um desses grupos possui suas próprias variáveis CPU, MEM, I/O$_R$ e I/O$_W$ totalizando oito variáveis no conjunto dependente. De posse das informações apresentadas nos dois parágrafos anteriores, a Tabela 1 pode ser expandida para a Tabela 2.

Definidas as métricas, a proposta pode ser, então, executada. Os dois primeiros passos compreendem em coletar informações, tanto das métricas referentes à aplicação, como aquelas re-

Tabela 2 – Tabela completa da proposta de (DO et al., 2011)

Métricas da aplicação			Métricas da aplicação							
App_1	...	App_N	CPU_F	MEM_F	I/O_{RF}	I/O_{WF}	CPU_B	MEM_B	I/O_{RB}	I/O_{WB}

lacionadas ao sistema. Por um determinado período de tempo, esses dados são registrados e, ao seu final, submetidos a CCA que, por sua vez, retorna todos os padrões de correlação identificados entre as variáveis envolvidas. Apenas o padrão detentor da maior raiz canônica, mais próxima a um, é escolhido, pois ele representa o maior nível de correlação. De posse dessa única raiz, são encontrados os vetores e pesos canônicos a serem utilizados para, dado uma entrada de métricas de aplicação, prever a performace do sistema ou vice versa.

A principal desvantagem dessa proposta é não analisar questões energéticas durante o processo de escolha de VMs para migração. Conforme visto na Tabela 2, apenas dados sobre as métricas relacionadas ao desempenho são analisados. Nenhuma informação ligada a energia é ponderada ou sequer coletada durante o procedimento da seleção. Esta proposta, claramente, está voltada para garantir o cumprimento do SLA e não se preocupa se a solução encontrada é a melhor do ponto de vista energético.

3.3 Alocação baseada em correlação e alocação baseada em clusterização de picos de demanda

A proposta de Verma et al. (2009) começa segmentando a consolidação de servidores em três tipos: estática, semi estática e dinâmica. As consolidações estática e semi estática consistem em alocar as máquinas virtuais em servidores físicos por longos períodos: meses para a consolidação estática ou dias para a semi estática. Ambas não realizam migrações automáticas, cabendo aos administradores do ambiente virtualizado tal tarefa. Já a consolidação dinâmica aloca VMs em servidores por horas e suportam migrações automáticas. Segundo o mesmo trabalho, por questões de confiança, a maioria dos administradores não usa a consolidação dinâmica, preferindo adotar propostas estáticas e semi estáticas que, dessa maneira, serão o seu foco.

Com base em estudos próprios, os pesquisadores concluíram que o nível de correlação entre máquinas virtuais está diretamente relacionado ao número de ocorrências de violações no SLA. Ou seja, VMs correlacionadas não podem ser alocadas em um mesmo servidor, pois, em algum momento de seus processamentos, uma irá interferir no desempenho da outra e poderá ocasionar uma, ou mais, violações de SLA. Tal observação deve ser levada em consideração durante o processo de alocação ou migração de máquinas virtuais.

O trabalho em questão propõe duas técnicas para alocação efetiva de máquinas virtuais: a *Correlation Based Placement* (CBP) e a *Peak Clustering Based Placement* (PCP). A primeira é voltada para ambientes virtualizados onde VMs, além de apresentarem baixo correlacionamento, possuem alta defasagem entre seu valor de pico e seu comportamento habitual. Já a segunda foi concebida para amenizar as limitações da primeira. A PCP trabalha em cima da análise de correlacionamento entre os valores de pico das Máquinas Virtuais.

O funcionamento do CBP é baseado na técnica pMapper (VERMA; AHUJA; NEOGI, 2008) que, por sua vez, desenvolve adaptações do algoritmo *First-Fit Decreasing* (FFD) para resolver questões

como a escolha de servidores objetivando minimizar a energia total consumida. O CBP, contudo, possui algumas diferenças em relação à sua técnica base: 1) Não mensura as necessidades da máquina virtual com seu valor de pico, mas sim com alguma medida estatística como, por exemplo, o 90-percentil; 2) Insere restrições referentes ao nível de correlação das VMs. Esta técnica, contudo, apresenta duas limitações importantes: Em primeiro lugar, ela provoca uma sobrecarga significativa no momento de inserção da análise da correlação entre as VMs. Em segundo lugar, ela se torna ineficiente, no ponto de vista da energia, em ambientes onde existem muitas máquinas virtuais correlacionadas, pois, cada uma delas deverá ser alocada, mesmo sozinha, em um servidor, gerando mais consumo de energia.

Já a técnica PCP até aceita Máquinas Virtuais com picos correlacionados alocadas em um mesmo servidor. Contudo, evita ao máximo designar duas ou mais VMs com picos de demanda que ocorram ao mesmo tempo. Essa técnica consiste em criar *clusters* de aplicações com picos correlacionados utilizando uma função de similaridade que mensura o nível de semelhança entre conjuntos de dados passados como entrada. Esses conjuntos referem-se aos dados dos recursos monitorados de cada VM. Identificado os *clusters*, os servidores são ordenados de acordo com suas eficiências energéticas, os mais eficientes primeiro, e, por fim, *clusters* completos ou subgrupos deles são destinados para cada servidor. Não existem garantias que VMs correlacionadas são livres de terem picos de demanda ao mesmo tempo. Para contornar esta adversidade, a proposta determina recursos de reserva para cada máquina virtual. Essa reserva é proporcional aos picos de cada VM.

Ambas as propostas, CBP e PCP, apresentam alguns problemas destacáveis. A CBP, tem como principal ponto falho, a ineficiência energética quando aplicada em máquinas virtuais com um alto nível de correlação. A PCP, por sua vez, falha quando aloca recursos de reserva para cada máquina virtual em virtude de possíveis picos de demanda que possam vir a acontecer ao mesmo tempo. Isto, pode culminar em desperdícios de recursos físicos. Acima de tudo, a proposta é voltada para consolidações de longo prazo e não há garantias sobre sua eficiência quando aplicada em consolidação dinâmica de VMs.

3.4 Escolha aleatória, correlação máxima, tempo mínimo de migração e maior potencial de crescimento

Buyya, Beloglazov e Abawajy (2010) buscam minimizar o consumo de energia geral de um *data center*. O trabalho relacionado em questão é segmentado em duas partes: a primeira voltada para a alocação de máquinas virtuais, enquanto a segundo diz respeito a mecanismos de realocação de VMs otimizando sua localização. A primeira parte é mapeada como um problema da mochila (*bin packing*) e resolvida utilizando uma modificação própria do algoritmo *Best Fit Decreasing* (BFD). Esta solução foi denominada *Modified Best Fit Decreasing* (MBFD).

O MBFD (BELOGLAZOV; BUYYA, 2010) toma, como base, o consumo de CPU corrente de servidores e máquinas virtuais. No caso de VMs em processo de alocação, será considerada o nível desejável de processamento definido via SLA. O primeiro passo consiste em listar e ordenar as VMs de forma decrescente. Feito isso, a lista deverá ser percorrida e observado cada caso possível de alocação entre máquina virtual e servidores. Será ponderado, em primeiro lugar, se existem recursos suficientes para recebê-la. Em caso positivo, uma avaliação de energia pós-alocação será feita. O servidor com a menor estimativa será escolhido.

A segunda fase, por sua vez, é dividida em outros dois passos: As escolhas da Máquina Virtual a ser migrada e do servidor destino a receber tal VM. A definição da máquina física destino também se utiliza do algoritmo MBFD. Em relação as políticas de escolha de VMs, inicialmente, são definidos dois limiares: um inferior e um superior referentes ao nível de utilização de recursos dos servidores (ver a Figura 12).

Figura 12 – Estados disponíveis de servidores

Conforme observado, cada servidor pode, em um determinado momento, se encontrar em um dos estados vistos. A região em azul, abaixo do limite inferior, representa o servidor em questão operando abaixo de sua capacidade mínima ideal e, por isso, deverá ser posto em modo de hibernação. Mas antes disso, todas as Máquinas Virtuais por ele hospedadas deverão ser migradas para outros aparelhos. Neste caso, as políticas de seleção não são aplicadas, pois não há necessidade de escolher uma VM para migração. A região em vermelho, acima do limite superior, informa o servidor sobrecarregado e que, dessa forma, precisa livrar-se de alguma carga de trabalho. Neste caso, as políticas de seleção são chamadas até o servidor estar em seu estado normal de operação, ou seja, na região verde.

No trabalho em questão, são propostas três heurísticas como políticas de seleção de máquinas virtuais:

- **Escolha Aleatória:** Escolhe, aleatoriamente, cada VM a ser migrada de um servidor sobrecarregado. Essa é a proposta mais simples das políticas sugeridas;
- **Minimização de Migrações:** O objetivo principal dessa política é minimizar o número de migração. Ela parte do princípio que um menor número de migrações irá minimizar a sobrecarga na rede. Essa proposta apresentou um melhor desempenho em um ambiente simulado;
- **Maior Potencial de Crescimento:** Defende a idéia de escolher as Máquinas Virtuais com menor nível de uso de processador. Tal escolha se deve a uma tentativa de minimizar a probabilidade de violações no SLA.

Apesar de apresentar bons resultados para os cenários testados, essa abordagem apresenta algumas limitações importantes em ambas as fases definidas. Na primeira fase a escolha dos servidores é feita através da análise do incremento de energia provocado pela alocação de novas máquinas virtuais. Não é levado em consideração a eficiência energética, ou seja, quantidade de processamento dividido pelo consumo de energia, e pode levar servidores com baixa eficiência energética a receber muitas Máquinas Virtuais e ocasionar um consumo de energia maior do que o previsto, pois a VM precisará de mais tempo para ser totalmente executada.

Já em relação à segunda fase, durante o procedimento de escolha de Máquinas Virtuais para migração, o principal problema está na base de dados escolhidas para a decisão tomada. Todas as propostas baseiam-se em medidas de momento para definir sua escolha. Em casos onde as máquinas virtuais apresentam uma alta instabilidade de consumo, esse processo pode definir valores esperados de forma errônea. Por exemplo, suponha que, somente naquele momento, uma das VMs hospedadas estava em modo de hibernação. Essa VM, potencialmente, seria escolhida por três das quatro propostas acima definidas, com excessão da política de escolha aleatória. Seria escolhido um servidor destino,

com base nesse consumo de momento, que poderia, quando a VM for reativada, não ser mais capaz de suportá-la gerando a necessidade de uma nova migração.

Ao contrário das propostas de Buyya, Beloglazov e Abawajy (2010), esta dissertação focará em buscar uma solução baseada no histórico de consumo das máquinas virtuais e servidores.

Beloglazov e Buyya (2012) apresentam uma versão mais completa do trabalho discutido anteriormente. Nesse artigo, são propostas novas técnicas relacionadas às fases de detecção de sobrecarga de servidor, bem como novos mecanismos que auxiliam na escolha de máquinas virtuais a serem migradas.

A primeira novidade está nas heurísticas voltadas para, dinamicamente, determinar os limiares inferiores e superiores dos servidores com base nos seus históricos de consumo de CPU. De uma maneira geral o procedimento consiste em, periodicamente, armazenar os dados de consumo dos servidores e, com base no valor encontrado através de uma análise estatística, encontrar o seu limiar superior. Este processo, através de um parâmetro de entrada, pode priorizar, ou não, a consolidação das máquinas virtuais, pois um valor baixo de tal entrada representa um alto limiar superior que pode aumentar a quantidade de VMs alocadas em um único servidor.

Por fim, uma nova política de seleção de máquinas virtuais também é proposta. A política de Correlação Máxima é baseada na idéia proposta por Verma et al. (2009), onde máquinas virtuais apresentam um elevado índice de correlação, se colocadas juntas, com altas probabilidades de sobrecarregar o servidor. Funciona da seguinte forma: Os históricos das VMs servirão como entrada para o método de correlação. Cada histórico será, em um dado momento, a variável independente do modelo, enquanto os demais farão parte do conjunto que representa as variáveis dependentes. De cada uma destas análises será obtido o coeficiente de correlação que representa a qualidade do relacionamento entre os dois grupos. A VM que apresentar um maior coeficiente quando confrontado com as demais será escolhida para migração.

Ao contrário das heurísticas definidas no trabalho anterior, a política de Correlação Máxima analisa o histórico de processamento das VMs e não apenas a sua situação de momento. Contudo, esta política não livra o servidor de ocorrências de violações de SLA, pois ela não migra todas as máquinas virtuais com um alto nível de correlação, como defendido no artigo original, apenas o suficiente para que o servidor deixe o estado de sobrecarregamento. Além disso, observando os resultados referentes ao consumo de energia, percebe-se que, na maioria dos casos, esta técnica não representou ganhou algum em termos de economia e obteve comportamento semelhante a técnicas mais simples como a de escolha aleatória.

4 ALGORITMO PROPOSTO: VBALANCE

Este capítulo apresenta uma política de seleção de máquinas virtuais, denominada VBalance, voltada para maximizar a economia de energia em um *data center*. De uma forma geral, o VBalance consiste em coletar informações a respeito do consumo de CPU das máquinas virtuais e o gasto energético dos servidores para estimar, utilizando o modelo de Regressão Linear Múltipla, qual a VM que mais contribui energeticamente com o servidor sobrecarregado. Nas seções posteriores serão apresentadas, respectivamente, a formulação matemática da proposta, bem como o seu algoritmo.

4.1 Formulação matemática

De maneira formal, a técnica VBalance é definida da seguinte maneira: Um *data center* (D) é composto por *k* servidores (H), que, por sua vez, podem conter de zero a *m* máquinas virtuais (V). Matematicamente:

$$D = \{H_1, H_2, H_3, H_4, H_5, H_6, \ldots, H_k\}, \qquad \text{com } k \geq 1$$
$$H = \{V_1, V_2, V_3, V_4, V_5, V_6, \ldots, V_m\}, \qquad \text{com } m \geq 0$$

Periodicamente, serão coletadas algumas informações dos servidores e das máquinas virtuais. Dos servidores, serão resgatados e armazenados dados referentes ao consumo de energia no instante *t*, enquanto, de cada uma das máquinas virtuais, será colhida o consumo de CPU também no dado momento *t*. Dessa maneira, será definida uma função H^i_{status} que receberá, como entrada, um tempo *t* e retornará uma tupla contendo os dados, daquele momento, de energia do servidor *i* (Y^t) e o consumo de CPU de cada VM *m* nele alocada (X^t_m). Ou seja:

$H^i_{status}(t) = \mathbb{N} \rightarrow R^{m+1}, \qquad m = ||H||$ e $1 \leq i \leq ||D||$. Ou seja:

$$H^i_{status}(t) = \{Y^t_0, X^t_1, X^t_2, X^t_3 \ldots, X^t_m\}$$

No instante em que um servidor é classificado como sobrecarregado, a política de seleção de máquinas virtuais VBalance será chamada para escolher, de acordo com os seus critérios, a melhor VM para migração. O critério adotado pela proposta é o nível de participação da CPU de cada VM que se encontra no respectivo servidor sobrecarregado em relação a energia do mesmo. Para mensurar esse relacionamento, será usado o modelo de Regressão Linear Múltipla (RLM) que recebe como entrada duas variáveis: uma Dependente e outra Independente. A Variável Dependente será representada pela Energia do servidor sobrecarregado. Já a Variável Independente, que, no caso da Regressão Múltipla, representa um conjunto de duas ou mais variáveis, será composta pelos níveis de consumo de CPU das máquinas virtuais hospedadas no servidor em questão. Essas duas variáveis serão definidas da seguinte forma:

IND^i_{mxn}, com $m = ||H||$ e $n =$ nº de coletas de H^i_{status} excluindo o primeiro elemento da tupla, tal que $n \geq m$. Ou seja:

$$IND^i = \begin{pmatrix} X^1_1 & X^1_2 & X^1_3 & \dots & X^1_m \\ X^2_1 & X^2_2 & X^2_3 & \dots & X^2_m \\ X^3_1 & X^3_2 & X^3_3 & \dots & X^3_m \\ \dots & \dots & \dots & & \dots \\ X^n_1 & X^n_2 & X^n_3 & \dots & X^n_m \end{pmatrix}$$

DEP^i_{1xn}, com $n =$ nº de coletas de H^i_{status} considerando apenas o primeiro elemento da tupla. Ou seja:

$$DEP_i = \begin{pmatrix} X^1_0 & X^2_0 & X^3_0 & \dots & X^n_0 \end{pmatrix}$$

O último passo será submeter as Variáveis Independente e Dependente à Regressão Linear Múltipla. A RLM consiste em resolver um sistema de equações onde as incógnitas serão os índices da função que descrevem o relacionamento entre as variáveis dependente e independente. A solução do sistema representa o nível de influência da CPU de cada VM na energia do servidor mais um índice referente ao termo independente (que deverá ser desprezado). Por fim, de posse desses dados, a VM que apresentar o maior peso será escolhida para migração. Ou seja:

$RLM(IND^i, DEP^i)$: $\beta \times IND^i = DEP^i$, onde $\beta = \beta_0, \beta_1, \dots, \beta_m$

select(β'): $\beta \rightarrow H$, onde $\beta' = \beta - \beta_0$

Resumindo, a proposta VBalance é matematicamente definida abaixo:

$$D = \{H_1, H_2, H_3, H_4, H_5, H_6, \dots, H_k\}, \qquad \text{com } k \geq 1 \tag{4.1}$$

$$H = \{V_1, V_2, V_3, V_4, V_5, V_6, \dots, V_m\}, \qquad \text{com } m \geq 0 \tag{4.2}$$

$$H^i_{status}(t) = \mathbb{N} \rightarrow R^{m+1}, \qquad m = ||H||, \text{ e } 1 \leq i \leq ||D|| \tag{4.3}$$

IND^i_{mxn}, com $m = ||H||$ e $n =$ nº de coletas de H^i_{status} excluindo o primeiro elemento da tupla, tal que $n \geq m$ (4.4)

DEP^i_{1xn}, com $n =$ nº de coletas de H^i_{status} considerando, apenas, o primeiro elemento da tuplas (4.5)

$RLM(IND^i, DEP^i)$: $\beta \times IND^i = DEP^i$, onde $\beta = \beta_0, \beta_1, \dots, \beta_m$ (4.6)

select(β'): $\beta \rightarrow H$, onde $\beta' = \beta - \beta_0$ (4.7)

4.2 VBalance

A política de seleção de máquinas virtuais VBalance pode ser segmentada em duas fases principais: um período de coleta de informações e um instante referente à tomada de decisão de qual VM migrar. A primeira fase compreende em colher e armazenar os dados utilizados durante a fase de decisão. Enquanto a segunda fase consiste em submeter essas coletas ao modelo de Regressão Linear Múltipla e, de posse dos valores retornados, decidir se uma máquina virtual deve ser migrada ou não. A fase de coleta de informações é cíclica (por um período *T* a ser definido pelo usuário), intermitente e iniciada junto com a política VBalance. Já a segunda fase, por sua vez, tem início sempre que um servidor é qualificado como sobrecarregado e é finalizada com a escolha (ou não) de uma VM para migração. Tais fases serão explicadas com maiores detalhes nos parágrafos que seguem.

Na primeira etapa da proposta, a cada período *T*, são coletados dados de momento referentes à energia de todos os servidores do *data center*, bem como o consumo de CPU de cada uma das máquinas virtuais envolvidas. As informações recebidas são armazenadas para uso futuro. A Figura 13 exemplifica o comportamento desta fase em um cenário simples onde o *data center* é formado por um único servidor que hospeda, somente, uma máquina virtual.

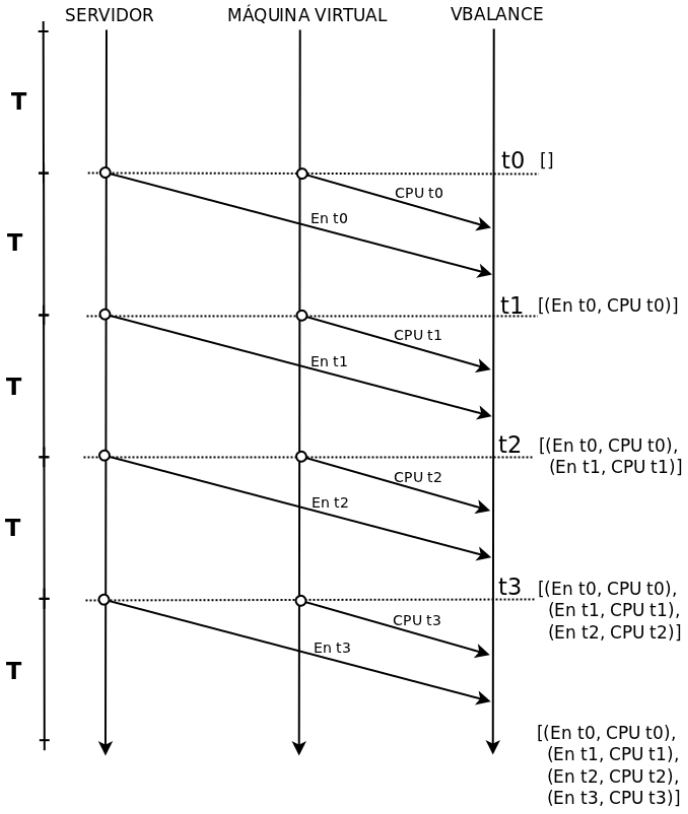


Figura 13 – Diagrama de Sequência referente a fase de coleta de dados

No instante *t0*, o consumo de energia atual (*En t0*) e a quantidade de processamento alocada (*CPU t0*) são capturados e enviados para a instância VBalance que, por sua vez, salva essas informações em uma estrutura de dados. O procedimento se repete pelos próximos três períodos (*t1*, *t2* e *t3*) e, ao final do diagrama, o VBalance conta com quatro amostras do comportamento energético do servidor e da utilização de processamento por parte da máquina virtual.

Esta etapa é intermitente em relação aos servidores, pois, mesmo quando eles se encontram

em modo de hibernação, as informações sobre o consumo de energia são coletadas. Por razões óbvias, apenas quando o servidor estiver desligado não serão produzidos tais dados. Já do ponto de vista da máquina virtual, esta fase ocorre durante todo o seu ciclo de vida. Mesmo quando a VM é migrada as informações continuarão sendo capturadas, mas com uma pequena observação: Ao ser finalizado o processo de migração, o histórico de consumo de CPU, da referida máquina virtual, deverá ser reinicializado, pois, caso contrário, esses valores dificultariam a identificação de um relacionamento entre as variáveis envolvidas.

A segunda etapa da proposta, ao contrário da anterior, não é periódica, nem intermitente. Na verdade, ela só é requisitada no instante em que um servidor é rotulado como sobrecarregado pela política de alocação de máquinas virtuais empregada em conjunto com o VBalance. O primeiro passo verifica se o servidor em questão contém apenas uma máquina virtual hospedada, pois, em caso positivo, o VBalance se nega a escolhê-la. Tal ação é motivada pela seguinte idéia: Se a VM em questão, sozinha, está conduzindo um servidor à sobrecarga de trabalho, muito provavelmente ela, ao ser alocada em outro servidor que já possui VMs hospedadas, também o levará ao estado sobrecarregado. Assim, seguindo este pensamento, decidiu-se adotar tal medida.

Conforme dito anteriormente, máquinas virtuais tem o seu histórico reiniciado quando completam sua migração para os servidores destino. Assim, os servidores destino serão formados por dois grupos de máquinas virtuais: as nativas e as recém-migradas. Foi definido que uma máquina virtual é classificada como nativa de um servidor quando ela está a, pelo menos, seis interações presente no equipamento físico, caso contrário a VM é rotulada como recém-migrada. Tal valor foi obtido através de experimentos próprios, via simulação, onde buscou-se o número mínimo de interações que proporcionasse economia de energia ao *data center* simulado. Para que o VBalance continue a trabalhar, ele exige que existam, pelo menos, três máquinas virtuais nativas dentro do servidor sobrecarregado. Caso tal condição não seja obedecida, a técnica não retorna nenhuma VM para migração.

Conforme observado na seção 2.5.2, a Regressão Linear Múltipla exige que o número de linhas (tempo da coleta) seja maior ou igual ao número de colunas (variáveis envolvidas) e, mesmo as máquinas virtuais nativas possuem diferentes tamanhos de histórico de processamento (VMs nativas mais antigas terão mais coletas do que as mais novas). Dessa maneira, faz-se necessário um nivelamento nos históricos, onde todos eles serão reduzidos ao tamanho do menor histórico dentre as VMs nativas. Essa redução pode acarretar em uma incompatibilidade entre o número de linhas e colunas o que resultaria em erro durante o processo de regressão. Assim, faz-se necessário que algumas VMs sejam excluídas da análise. Este corte é feito com base na mediana dos valores de CPU coletados. Feitos esses cortes, finalmente, as Variáveis Dependente e Independente, que servirão de entrada para a Regressão Linear Múltipla, poderão ser criadas.

Após a aplicação da Regressão Linear Múltipla, duas informações serão requisitadas e utilizadas antes da escolha final de qual máquina virtual será escolhida para migração. A primeira delas é o Coeficiente de Determinação (R^2) da relação encontrada. Foi definido um limiar mínimo para este coeficiente (*r2_min*) que, ao ser confrontado com o R^2 obtido, irá classificar a relação como boa ou ruim. Se o R^2 for maior do que *r2_min*, então trata-se de uma boa função e o processo de escolha pode prosseguir. Caso contrário, o VBalance se nega a fazer uma escolha com base em uma relação tão fraca.

Por fim, a segunda saída da regressão deverá ser analisada: o Vetor Peso referente a Variável Independente. Com base no Vetor Peso, será definida qual a máquina virtual que tem maior influência na energia do servidor sobrecarregado (Variável Dependente). A VM que mais contribui para isso será escolhida e, posteriormente, migrada para um outro servidor.

O fluxograma da Figura 14 resume as etapas descritas nos parágrafos acima.

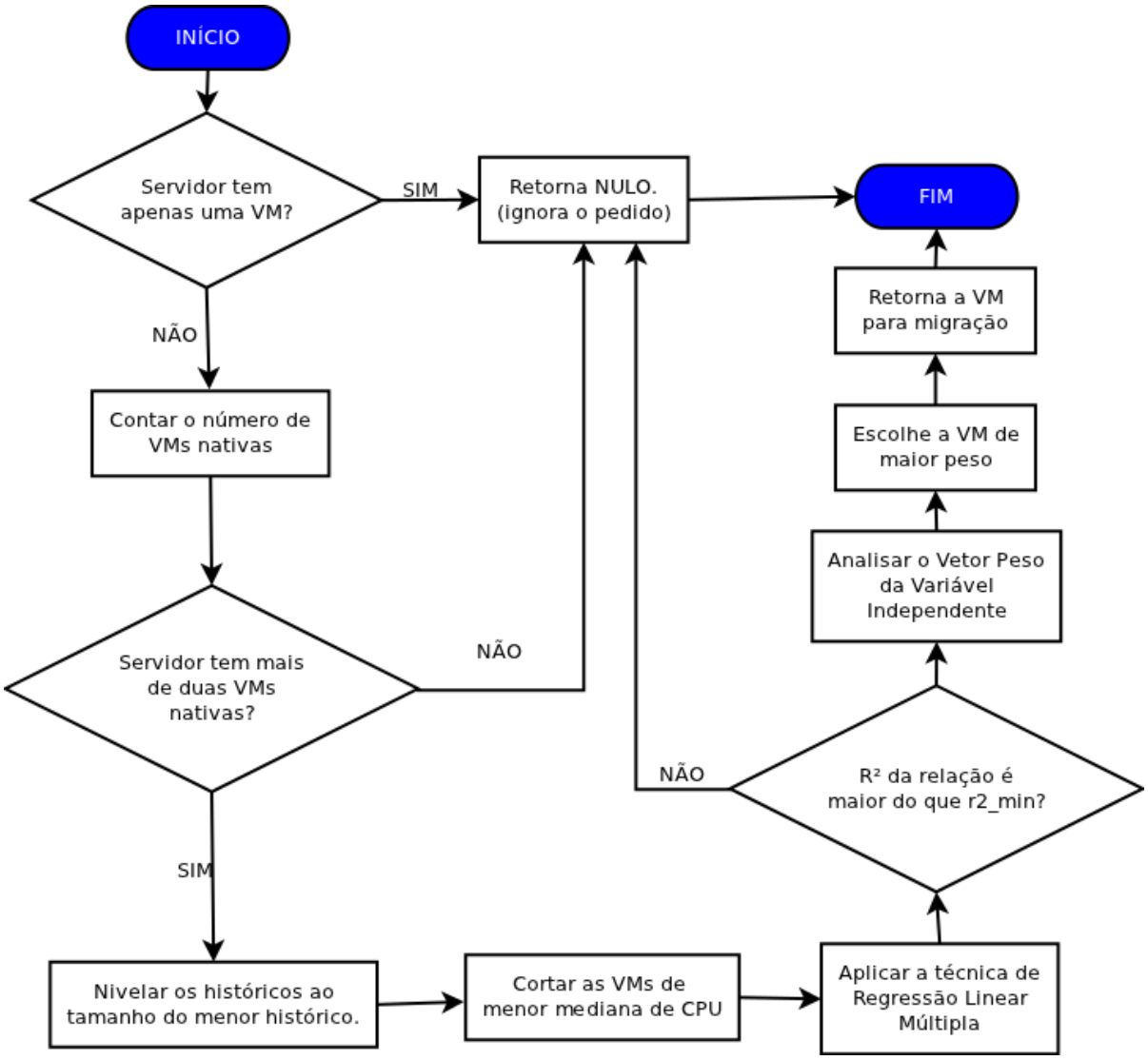


Figura 14 – Fluxograma da proposta VBalance

5 ANÁLISE DE DESEMPENHO

Após a definição da política VBalance sugerida neste trabalho, se faz necessário uma validação da sua eficiência em relação as demais propostas existentes. Neste trabalho foi escolhido o simulador *CloudSim* (CALHEIROS et al., 2011) como ferramenta para tal tarefa. O simulador em questão é capaz de simular os comportamentos e o relacionamento de inúmeros componentes que integram uma arquitetura convencional de Computação em Nuvens como, por exemplo, *data centers*, servidores, máquinas virtuais, etc.

A seguir, além de uma breve explanação sobre a ferramenta de simulação *CloudSim*, serão descritos o cenário e as métricas adotadas para validação deste trabalho. Tanto o cenário, como as métricas utilizadas foram baseadas no trabalho de (BELOGLAZOV; BUYYA, 2012). Ao final deste capítulo, será feita uma análise a respeito dos resultados obtidos através da simulação.

5.1 Por que simulação?

De acordo com (JAIN, 1991), existem três técnicas para validação de desempenho de um sistema ou de uma proposta: modelagem analítica, medição real e simulação. Cada uma destas técnicas possui aspectos positivos e limitações que devem ser ponderados na escolha do procedimento mais adequado para análise da performace de uma nova proposta.

A modelagem analítica, embora rápida e barata, diversas vezes, necessita de muitas simplificações do problema e suposições, o que pode acarretar em uma redução da precisão dos resultados obtidos. Além disso, o autor ainda ressalta que muitas pessoas são céticas em relação aos resultados encontrados através da modelagem analítica.

Já a medição, por se tratar de observações e análises comportamentais de sistemas reais, é a técnica mais confiável dentre as três supracitadas. Contudo, é o método mais caro tanto temporal, como financeiramente e mais complexo para análise, pois é difícil gerenciar os parâmetros de entrada do experimento e, em alguns casos, de repetí-los para uma nova investigação.

No caso de testes em cenários de Computação em Nuvens, a medição real possui mais um empecilho que é o acesso a uma infraestrutura de larga escala para execução dos testes. Muitos pesquisadores tem adotado, como uma alternativa a esta problemática escolha, executar seus testes em um ambiente de menor tamanho (com poucas máquinas físicas) e definir os resultados encontrados como sendo os mesmos de um *data center* convencional. Segundo (AMARANTE, 2013), tal possibilidade não é viável, pois o comportamento de *data centers* maiores podem sofrer variações que um ambiente com poucas máquinas não apresentaria.

A Tabela 3 apresenta um resumo das considerações feitas por (JAIN, 1991). Na referida tabela é possível observar que o método da simulação possui valores intermediários em relação as demais opções durante o estudo. É verdade que as técnicas de modelagem analítica e medição real apresentam aspectos positivos atrativos, contudo suas limitações as tornam complexas e potencialmente problemáticas. A simulação, de uma forma geral, apresentou resultados medianos quando comparada a outras técnicas.

Tabela 3 – Critérios para escolha de uma técnica de validação (JAIN, 1991)

Critério	Modelo Analítico	Simulação	Medição Real
Estágio do sistema	Qualquer	Qualquer	Protótipo
Tempo requerido	Pequeno	Médio	Variante
Ferramentas	Analistas	Linguagens de Computador	Aparelhos
Precisão	Baixa	Moderada	Difícil
Trade-off análise	Fácil	Moderado	Difícil
Custo	Pequeno	Médio	Alto
Confiança	Baixa	Média	Alta

Em face dos impasses constatados e nas conclusões obtidas na Tabela 3, optou-se pelo uso de simulações para validação da proposta. Além disso, durante as pesquisas, encontrou-se um cenário que, segundo (BELOGLAZOV; BUYYA, 2012), é composto por configurações reais de servidores, de máquinas virtuais e de tarefas a serem executadas, ou seja, um cenário que retrata bem um *data center* convencional.

5.1.1 *CloudSim*

O simulador *CloudSim* é uma ferramenta de código aberto, baseada em eventos e desenvolvido na linguagem de programação JAVA, que oferece mecanismos para modelagem e simulação de ambientes baseados em Computação em Nuvens. Através desta ferramenta, é possível criar diversos elementos que compõem uma infraestrutura em Nuvem, simular e observar o funcionamento de tais componentes e coletar resultados visando posterior análise.

Inicialmente, o *CloudSim* utilizava o SimJava (HOWELL; MCNAB, 1998) como *engine* base para suas simulações. Contudo, o SimJava apresenta algumas limitações técnicas especialmente relacionadas a escalabilidade: Segundo (CALHEIROS et al., 2011), para um grande número de entidades, o SimJava demonstra uma degradação na performace da simulação e no seu sistema de *debugging*. Em virtude de tal revés, decidiu-se criar um novo mecanismo de eventos.

O sistema de eventos desenvolvido para o *CloudSim* é composto, basicamente, por duas filas: a primeira delas, chamada de FutureQueue, é responsável por receber os eventos agendados para o futuro, enquanto a segunda, denominada DeferredQueue, lista os eventos a serem tratados naquele momento pela instância da classe, conhecida como CloudSim, que controla toda a execução da simulação. Todas as entidades da simulação estão aptas a enviar mensagens umas para as outras, porém, não se trata de uma comunicação direta, pois essas mensagens são enviadas, inicialmente, para a classe CloudSim e, posteriormente, repassadas para o componente destino.

Os eventos no simulador *CloudSim* são definidos pela classe SimEvent que, dentre outros dados sobre o evento, armazena informações como identificadores das entidades fonte e destino do evento, tempo em que o evento deve ser tratado, dentre outras. O evento também é referenciado por um rótulo, definido na classe CloudSimTags, cuja finalidade é indicar a natureza do evento e informar as ações necessárias para o envio ou recebimento de tal evento.

O *CloudSim*, além de simular características importantes de um cenário em Computação em Nuvens como virtualização e gerenciamento dinâmico de recursos, apresenta algumas funcionalidades extras úteis para uma análise econômica e de performace de soluções por ela simuladas. A ferramenta de

simulação permite a criação de nuvens federadas e mecanismos de balanceamento de cargas entre elas, a modelagem de padrões dinâmicos de consumo de recursos como CPU, memória e largura de banda. O *CloudSim* ainda apresenta um procedimento relacionado a tarifação onde, o usuário, inicialmente, define os custos por unidade de recurso consumido e, ao final da simulação, é disponibilizado o valor total a ser pago por cada cliente do serviço.

O principal módulo extra para o desenvolvimento deste trabalho é aquele relacionado ao consumo de energia do *data center*. O *CloudSim* define uma classe abstrata, denominada PowerModel, que deve ser extendida pelo usuário e vinculada as entidades referentes a CPU na simulação. Assim, durante as etapas da simulação, é possível extrair, com base no nível de consumo do recurso, o total de energia gasto por um servidor naquele instante de tempo. Obviamente, ao final da simulação, é possível obter o total de energia gasta por todo o *data center* para análises futuras. Um diagrama de classes do simulador CloudSim pode ser visto em (CALHEIROS et al., 2011).

A saída do simulador dependerá do que está em teste, cabendo ao usuário programá-la. No caso deste trabalho, os resultados observados são aqueles relacionados a questões de energia e desempenho do sistema com a aplicação das políticas. A performace foi segmentada em três métricas: número total de migrações, percentual de tempo em que os servidores permaneceram com 100% de uso de CPU, e percentual de recursos perdidos em virtude da migração de VMs.

5.2 Cenário

Para simular testes de maneira similar aos ambientes reais, procurou-se montar um *data center* com configurações próximas a um *data center* convencional: servidores e máquinas virtuais com configurações encontradas na atualidade, tarefas com perfis de consumo de recursos físicos equivalentes a realidade etc. Após uma série de pesquisas, foi determinada a utilização do cenário proposto em (BELOGLAZOV; BUYYA, 2012) por dois motivos principais: 1) Atende a ideia inicial de um cenário com atributos reais; 2) Apresenta o cenário usado para validar as propostas concorrentes da VBalance e, teoricamente, onde tais políticas apresentaram os melhores resultados.

O *data center* sugerido é composto por 800 servidores, sendo 400 deles do tipo HP ProLiant G4 e os demais formados por máquinas HP ProLiant G5. Ambos os tipos de servidores são constituídos por um processador de dois núcleos, 4 gigabytes de memória RAM, 1 Gbit/s de largura de banda e 1 terabyte de armazenamento secundário. O que os diferencia é a capacidade de processamento, enquanto o HP ProLiant G4 possui 1860 MIPS por núcleo, o HP ProLiant G5 tem 2660 MIPS por núcleo. A Tabela 4 resume as configurações das duas classes de servidores:

Tabela 4 – Resumo das configurações dos servidores (BELOGLAZOV; BUYYA, 2012)

Recurso	HP ProLiant G4	HP ProLiant G5
Número de núcleos	2	2
Processamento	1860 MIPS	2660 MIPS
Memória Primária	4 GB	4 GB
Memória Secundária	1 TB	1 TB
Largura de Banda	1 Gbit/s	1 Gbit/s
Qnt de servidores	400	400

Ainda são definidos os modelos de consumo de energia de cada um dos tipos de servidores

supracitados. Os gastos energéticos de cada servidor são dados em função do seu nível de carga de trabalho, destinado ao atendimento de requisições de máquinas virtuais, em um determinado tempo. Tal nível é mensurado através do percentual de utilização dos recursos e, dessa forma, vai de 0% à 100% com saltos de 10% (veja a Tabela 5). Os valores intermediários, não representados na tabela em questão, são determinados através de interpolação com os valores piso e teto mais próximos.

Tabela 5 – Consumo de energia de cada servidor (BELOGLAZOV; BUYYA, 2012)

Servidor	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
HP G4 (W)	86	89.4	92.6	96	99.5	102	106	108	112	114	117
HP G5 (W)	93.7	97	101	105	110	116	121	125	129	133	135

Para a simulação de uma tarefa no *CloudSim*, utiliza-se uma entidade, denominada *Cloudlet*, responsável por gerenciar a execução da tarefa a ela incumbida, incluindo suas requisições por recursos para execução. No cenário construído por (BELOGLAZOV; BUYYA, 2012), foram coletados dados relacionados ao consumo de CPU através do, hoje desativado, projeto CoMon (PARK; PAI, 2006). Este projeto consistia em uma grande infraestrutura responsável por colher e armazenar informações estatísticas do PlanetLab (CONSORTIUM, 2013). Foi registrado, a intervalos de 5 minutos, o consumo de CPU de milhares de VMs espalhadas por diversos lugares no mundo. Vale ressaltar que todas as VMs monitoradas não possuíam mais de um núcleo de CPU para processamento. O período de observação foi entre os meses de Março e Abril de 2011, onde, ao seu final, foram escolhidos, aleatoriamente, 10 dias para a realização dos testes. Cada histórico de consumo de CPU coletado deverá ser entregue a uma *Cloudlet* que, por sua vez, será incumbida a uma única máquina virtual para processamento.

Assim como os servidores, as configurações das máquinas virtuais também foram concebidas seguindo modelos reais comercializados. Foram escolhidas quatro instâncias de VMs disponibilizadas pela empresa Amazon através de seu serviço Amazon EC2 (AMAZON, 2013), são elas: Microinstância (*Micro Instance*), Instância pequena de propósito geral (*Small Instance*), Instância extra grande de propósito geral (*Extra Large Instance*) e, por fim, Instância média otimizada para computação (*High-CPU Medium Instance*). Por questões de compatibilidade com as cargas de trabalho obtidas para as *Cloudlets*, foi considerado, quando necessário, apenas um núcleo de processador para cada tipo de VM, o que representa, em termos de simulação, uma segmentação da capacidade de MIPS e da memória total oferecida. A Tabela 6 resume as configurações de cada instância de VM utilizada na construção do cenário para testes.

Tabela 6 – Resumo das configurações das máquinas virtuais (BELOGLAZOV; BUYYA, 2012)

Recurso	High-CPU Medium	Extra Large	Small	Micro
Qnt de núcleos	1	1	1	1
Processamento	2500 MIPS	2000 MIPS	1000 MIPS	500 MIPS
Memória Primária	870 MB	1.7 GB	1.7 GB	613 MB
Tamanho da VM	2.5 GB	2.5 GB	2.5 GB	2.5 GB
Largura de Banda	10 Mbit/s	10 Mbit/s	10 Mbit/s	10 Mbit/s
Qnt de VMs	$\frac{\text{Nº de Cloudlets}}{4}$	$\frac{\text{Nº de Cloudlets}}{4}$	$\frac{\text{Nº de Cloudlets}}{4}$	$\frac{\text{Nº de Cloudlets}}{4}$

A fim de comparar os resultados obtidos com a proposta VBalance, utilizou-se as seguintes políticas de seleção definidas em (BELOGLAZOV; BUYYA, 2010) e (BELOGLAZOV; BUYYA, 2012).

1. **Escolha Aleatória (*Random Choice* ou RC):** Escolhe, de maneira aleatória, uma máquina virtual para migração;
2. **Tempo Mínimo de Migração (*Minimum Migration Time* ou MM):** Escolhe a máquina virtual com menor consumo momentâneo de memória primária. Tal política parte do princípio que a VM com

menor quantidade de memória alocada levará menos tempo para ser migrada;

3. **Mínima Utilização de CPU (*Minimum CPU Utilization* ou MU):** Escolhe a máquina virtual com menor consumo momentâneo de CPU. O objetivo desta política é reduzir o potencial crescimento do consumo de CPU do servidor e, por consequência, evitar possíveis violações de SLAs;

4. **Máxima Correlação (*Maximum Correlation* ou MC):** Esta política consiste em analisar o nível de correlação entre as máquinas virtuais presentes em um servidor e migrar aquela que apresenta um maior nível de correlação em relação as demais. É a única política, dentre as quatro, que se utiliza de informações do histórico de consumo das VMs.

Tanto quanto as políticas de seleção definidas acima, é importante definir algumas políticas de alocação para a realização dos testes do VBalance, uma vez que, além de designar máquinas virtuais para servidores físicos, essas políticas determinarão o estado em que se encontra um dado servidor e, no caso de sobrecarregamento, chamará a política de seleção para determinar qual a melhor VM a ser migrada.

A política *Static Threshould* consiste em classificar, como sobrecarregados, todos aqueles servidores que possuem um nível de utilização de CPU superior a um determinado limiar estático. Segundo (BELOGLAZOV; BUYYA, 2012), os melhores resultados foram encontrados com um limiar de 80%. Essa política foi escolhida pois apresenta uma técnica onde o limiar é identificado estaticamente, ou seja, independente do histórico de consumo do recurso, o valor do limiar será sempre o mesmo.

Ainda segundo (BELOGLAZOV; BUYYA, 2012), em virtude das naturezas dinâmicas e imprevisíveis das cargas de trabalhos submetidas aos servidores, se faz necessário o uso de limiares dinâmicos para determinar o estado de um servidor. Dessa maneira, os pesquisadores propuseram um conjunto de técnicas voltadas para definir o limiar superior, aquele que determina se um servidor está sobrecarregado ou não, com base em uma análise estatística do histórico de consumo de CPU do servidor. Após uma análise dos resultados apresentados no artigo, escolheu-se a política de alocação *Median Absolute Deviation* (MAD) para os testes, pois esta apresentou o pior desempenho em termos de energia.

A política de alocação MAD utiliza-se da medida de dispersão estatística *Median Absolute Deviation* (JAIN, 1991) para determinar, com base no histórico de consumo de CPU, o limiar máximo que determina o estado do servidor em análise. O limiar superior é determinado através da Equação 5.1.

$$T_u = 1 - s \times MAD_{value} \tag{5.1}$$

Onde:

$T_u =$ Limiar superior

$s =$ Parâmetro de segurança, com $s \in \mathbb{R}^+$

$MAD_{value} =$ Valor MAD obtido com histórico de CPU

Após calculado o limiar máximo (T_u) na fórmula acima, se o percentual de consumo de CPU atual for maior que o T_u obtido, o servidor será classificado como sobrecarregado e uma política de seleção será chamada para escolher uma ou mais máquinas virtuais para migração. Caso contrário, nada será feito e o servidor continuará funcionando normalmente.

5.2.1 Parâmetros de simulação

Para que a simulação através do *CloudSim* seja possível, faz-se necessário a configuração prévia de alguns parâmetros. Nesta seção serão definidos os valores referentes às políticas de alocação e seleção utilizadas nos testes, bem como alguns parâmetros relacionados a simulação que ainda não foram definidos como, por exemplo, o tempo máximo de simulação.

Em relação às políticas de alocação escolhidas (THR, MAD), precisa-se definir os valores relacionados, no caso da THR, ao limiar estático superior que determina se um servidor está sobrecarregado ou não e, no caso da MAD, os parâmetros de segurança usados no cálculo do limiar superior dinâmico. Com base na análise dos resultados apresentados em (BELOGLAZOV; BUYYA, 2012), determinou-se os valores de 80% para o limiar estático superior da política THR e o valor de 2.5 para o parâmetro de segurança da política MAD.

Quanto às políticas de seleção testadas, apenas as políticas MC e VBalance possuem parâmetros de entrada destacáveis. A política de Máxima Correlação tem, como único parâmetro de entrada, um tamanho máximo de histórico de CPU de 30 coletas, ou seja, somente serão armazenadas as trinta coletas mais recentes de cada máquina virtual. Assim, quando o histórico estiver cheio, os dados mais antigos serão excluídos, sendo substituídos pelas informações mais atuais.

Já a política de seleção de máquinas virtuais VBalance, sugerida neste trabalho, recebe os seguintes parâmetros de entrada para processar corretamente: 1) os tamanhos dos históricos de CPU (máquinas virtuais) e de Energia (servidores); 2) O tamanho mínimo do histórico de CPU usado para classificar as VMs como nativas e não-nativas (veja o Capítulo 4); 3) O tamanho mínimo do Coeficiente de Determinação (R^2) utilizado para mensurar a qualidade da estimativa computada pelo modelo de Regressão Linear Múltipla. Os valores destes parâmetros foram definidos empiricamente com base em observações de resultados obtidos via simulação, onde buscou-se os menores valores possíveis que proporcionasse uma economia mínima de energia ao *data center* simulado. Os melhores resultados foram obtidos com um tamanho de histórico máximo de 18 coletas, tamanho mínimo de histórico de CPU de 6 coletas e um valor mínimo do Coeficiente de Determinação de 80%.

Além das configurações de servidores (Tabela 5) e máquinas virtuais (Tabela 6) definidos neste capítulo, alguns parâmetros referentes a questões temporais são necessários para a simulação através da ferramenta *CloudSim*. Primeiramente, por se tratar de um simulador baseado em eventos, é indispensável um período de *schedule* para atualização do estado dos elementos envolvidos e, caso necessário, a produção de eventos para informá-los sobre os resultados ou ocorrências de determinadas ações como, por exemplo, o ínicio ou fim de uma migração, a atualização do processamento de uma tarefa e etc. Por questões de compatibilidade com os *logs* entregues as *Cloudlets*, esse período de *schedule* foi determinado como de 300 segundos, ou seja, 5 minutos.

Também relacionado aos *logs* de CPU obtidos através do projeto CoMon, foi instituído um tempo máximo de simulação. Como os *logs* se referem a informações de dias específicos, a simulação se encerra no momento em que o dia em questão chega ao fim (86400 segundos após o início), independentemente, se todas as máquinas virtuais foram atendidas ou não. Esta medida é aceitável, pois não há garantias de que o comportamento de uma *Cloudlet* se perpetuará de um dia para o outro. Caso tal ação não fosse tomada, o cenário teria sua característica de proximidade a um *data center* real comprometida.

A Tabela 7 resume todos os parâmetros definidos nesta seção.

Tabela 7 – Resumo das configurações extras para simulação

Parâmetros	Valores
Limiar Estático Superior (THR)	0.8
Parâmetro de Segurança (MAD)	2.5
Tamanho do histórico de CPU (MC)	30
Tamanho do histórico de CPU e Energia (VB)	18
Tamanho do histórico de CPU (VB)	6
Coeficiente de Determinação mínimo (VB)	0.8
Período de *schedule*	300 seg
Tempo máximo de simulação	86400 seg

5.3 Métricas

Para medir a eficiência da política de seleção VBalance foram adotadas quatro métricas baseadas no trabalho de (BELOGLAZOV; BUYYA, 2012). A primeira, e mais intuitiva delas, é a quantidade de energia consumida pelo *data center* durante o dia simulado. Por se tratar de uma proposta voltada para a economia energética de um *data center*, esta será a métrica de maior peso na análise do rendimento das políticas de seleção.

As demais métricas referem-se a questões voltadas para o impacto da aplicação dessas políticas no desempenho geral do *data center*. A primeira delas é o número de migrações de máquinas virtuais que ocorreram durante o dia simulado. As outras duas são definidas como *Performace Degradation due to Migration* (PDM) e *SLA violation Time per Active Host* (SLATAH) e serão melhor explicadas nas próximas seções.

5.3.1 *Performace degradation due to migration*

A ferramenta *CloudSim* atribui, automaticamente, uma degradação de 10% na quantidade de CPU destinada a uma máquina virtual em migração e mantém esse percentual enquanto a VM não for completamente migrada. Essa atenuação é motivada pelo tratamento de eventos e processamento de instruções relacionadas aos algoritmos de migração. Contudo, (BELOGLAZOV; BUYYA, 2012) interpreta tal degradação como uma violação de SLA, uma vez que nem toda a quantidade de CPU destinada a VM é usufruída por ela.

Dentro deste contexto, foi elaborada a métrica *Performace Degradation due to Migration* ou PDM. Esta métrica mensura, em porcentagem média, a quantidade de MIPS não entregues às máquinas virtuais, enquanto essas estiveram em migração. Matematicamente, a PDM é formalizada assim:

$$PDM = \frac{1}{M} \sum_{j=1}^{M} \frac{C_{d_j}}{C_{r_j}} \tag{5.2}$$

Onde:

$M =$ Número de máquinas virtuais

$C_{d_j} =$ CPU perdida devido a migração pela VM j

$C_{r_j} =$ CPU requisitada durante a migração pela VM j

5.3.2 *SLA violation time per active host*

A característica de alocação dinâmica de recursos, encontrada em cenários de Computação em Nuvens, permite que as máquinas virtuais requisitem e usem apenas a quantidade de *hardware* necessário para executar suas tarefas. Em contrapartida, tal peculiaridade pode levar servidores a um estado de subutilização ou sobrecarremento. Beloglazov e Buyya (2012) definiram, como violação de SLA, os casos onde um servidor sobrecarregado está com seu nível de uso de CPU em 100%. Tal afirmação é compreensível, pois, um servidor nesta situação, não pode oferecer mais CPU às VMs por ele hospedadas. Dessa forma, basta que apenas uma máquina virtual requisite um pouco mais de CPU para culminar em uma violação de SLA.

Dentro deste contexto, foi criada a métrica *SLA violation Time per Active Host* ou SLATAH. Esta métrica mensura, em porcentagem média, a quantidade de tempo em que os servidores, enquanto estavam no estado ativo, permaneceram com nível de utilização de CPU igual a 100%. Matematicamente, a SLATAH é definida assim:

$$SLATAH = \frac{1}{N}\sum_{i=1}^{N}\frac{T_{s_i}}{T_{a_i}} \tag{5.3}$$

Onde:

$N =$ Número de servidores

$T_{s_i} =$ Tempo em que o servidor i teve 100% de uso de CPU

$T_{a_i} =$ Tempo do servidor i em estado ativo

5.4 Resultados

Esta seção irá apresentar os resultados obtidos com os testes realizados no simulador CloudSim usando as configurações e métricas apresentadas nas seções anteriores. Os valores escolhidos para cada uma das políticas representam a mediana dos resultados encontrados nos dez dias simulados. Após a simulação dos dez dias, foi escolhido a mediana de cada métrica e, tal resultado, definido como aquele que descreve a política de seleção em questão.

Por questões de simplificação, os nomes das políticas de seleção foram abreviados na construção dos gráficos. A Tabela 8 faz o mapeamento dos nomes reais das políticas de seleção de máquinas virtuais para as siglas definidas.

Tabela 8 – Mapeamento entre siglas e nomes das políticas

Siglas	Nomes reais
MM	Minimização de Migrações
MU	Maior Potencial de Crescimento
VB	VBalance
MC	Correlação Máxima
RS	Escolha Aleatória

5.4.1 Consumo de energia

No quesito economia de energia, percebe-se, através dos gráficos apresentados nas Figuras 15 e 16, que a política VBalance apresentou melhores resultados que as propostas sugeridas por (BUYYA; BELOGLAZOV; ABAWAJY, 2010) e (BELOGLAZOV; BUYYA, 2012). Estima-se que, quando utilizada com a política de alocação *Median Absolute Deviation*, a VBalance apresentou uma economia de 16 kWh em relação à segunda colocada, a política de seleção de Correlação Máxima, e de 42 kWh se comparada com a última colocada , a política de seleção de Minimização de Migrações. Tais resultados comprovam a eficiência da política VBalance em relação a economia de energia.

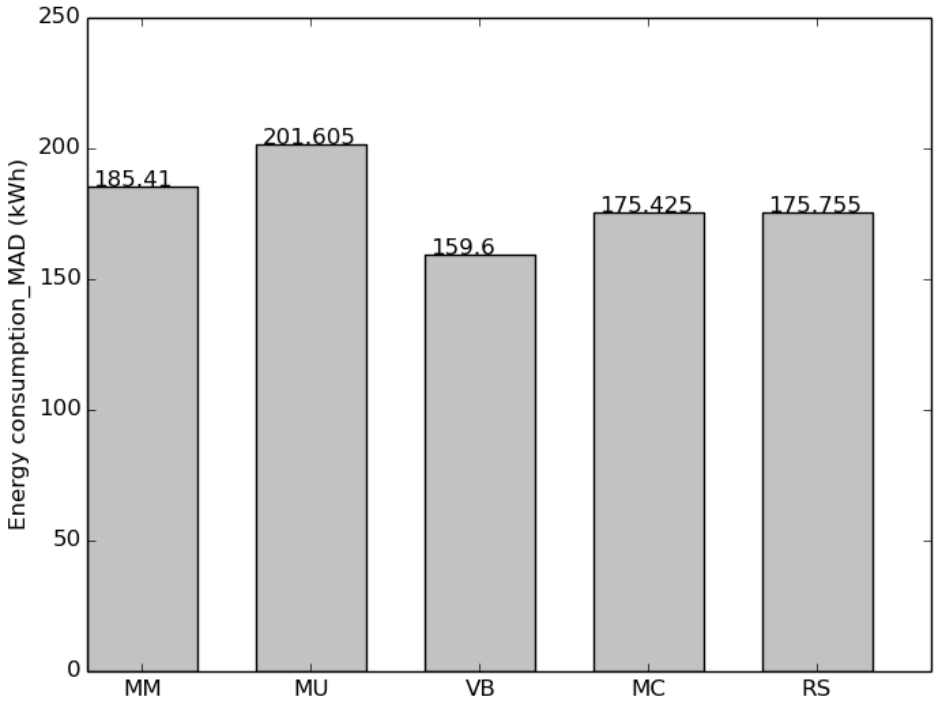


Figura 15 – Consumo de Energia usando política de alocação MAD

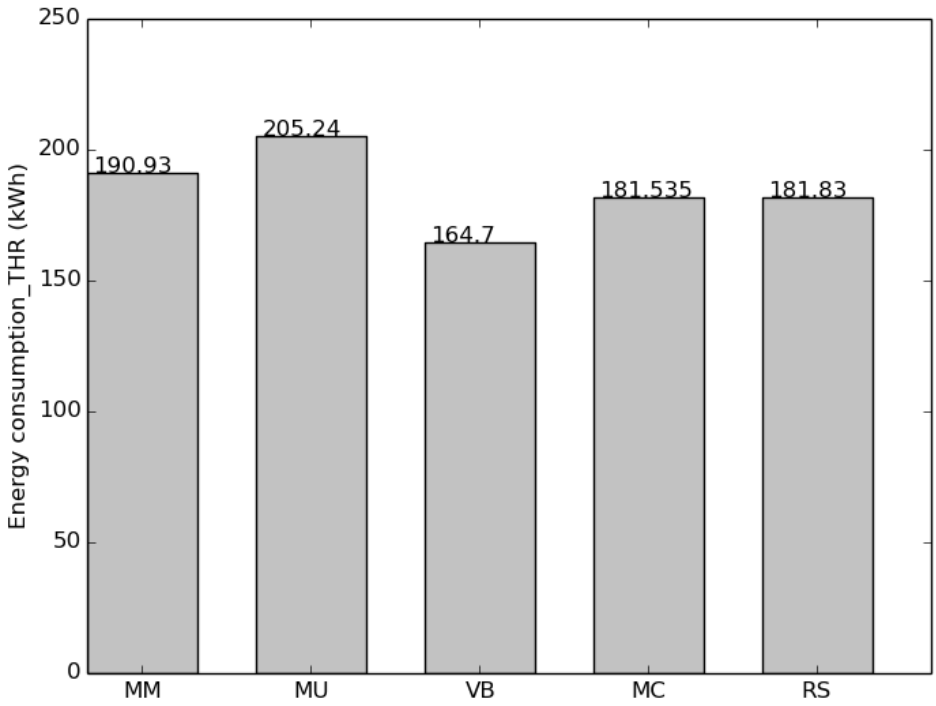


Figura 16 – Consumo de Energia usando política de alocação THR

5.4.2 Número de migrações

Já em relação ao número total de migrações realizadas durante o dia simulado, mais uma vez, política de seleção sugerida nesta dissertação apresentou melhores resultados quando comparada com suas concorrentes. Em ambas as políticas de alocação, MAD e THR, como pode ser observado, respectivamente, nas Figuras 17 e 18, a VBalance reduziu cerca de 3000 migrações em relação à segunda colocada (MC) e 10000 migrações em relação a última (MU). Essa redução acentuada deve-se, principalmente, as ações da VBalance de abortar o processo de migração ao se negar a escolher uma máquina virtual.

5.4.3 Métrica PDM

Os resultados obtidos na métrica *Performace Degradation due to Migration* também se mostraram favoráveis a política de seleção VBalance. Embora a política em questão não tenha mostrado os melhores resultados das cinco sob teste, ela obteve um comportamento satisfatório. O mau resultado em relação a política de seleção Maior Potencial de Crescimento (MU) já era esperado, uma vez que esta política escolhe, para migração, as VMs de menor consumo de processamento e, por consequência, de menor degradação de performace da CPU que, ao serem somados, produziriam o menor valor da métrica PDM. Já em comparação com a política MM, essa perda de desempenho deve-se ao tempo

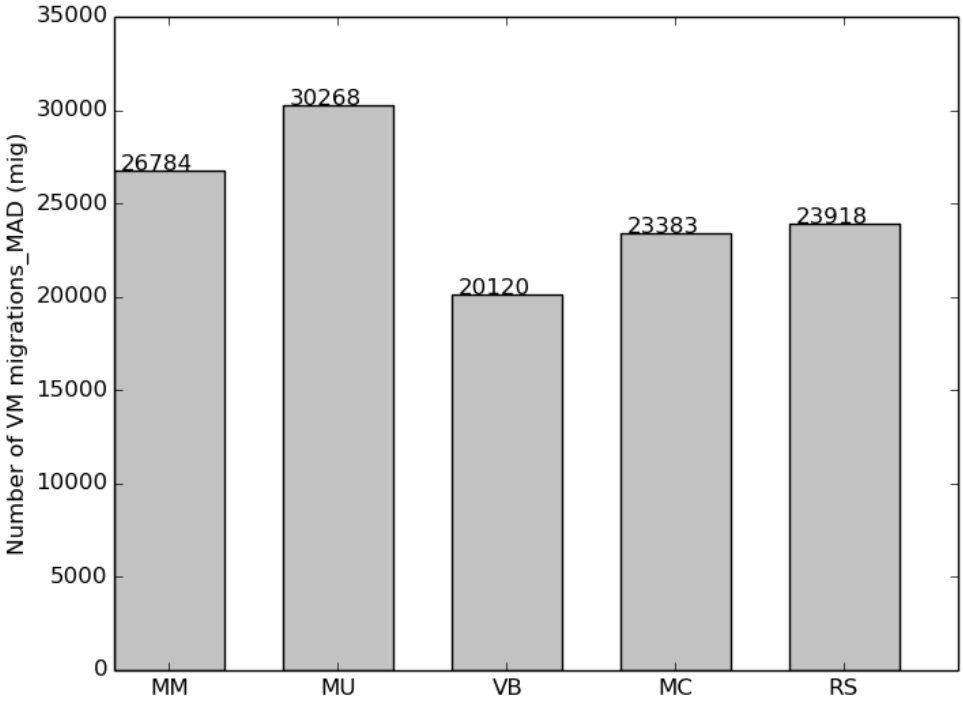


Figura 17 – Número de Migrações usando política de alocação MAD

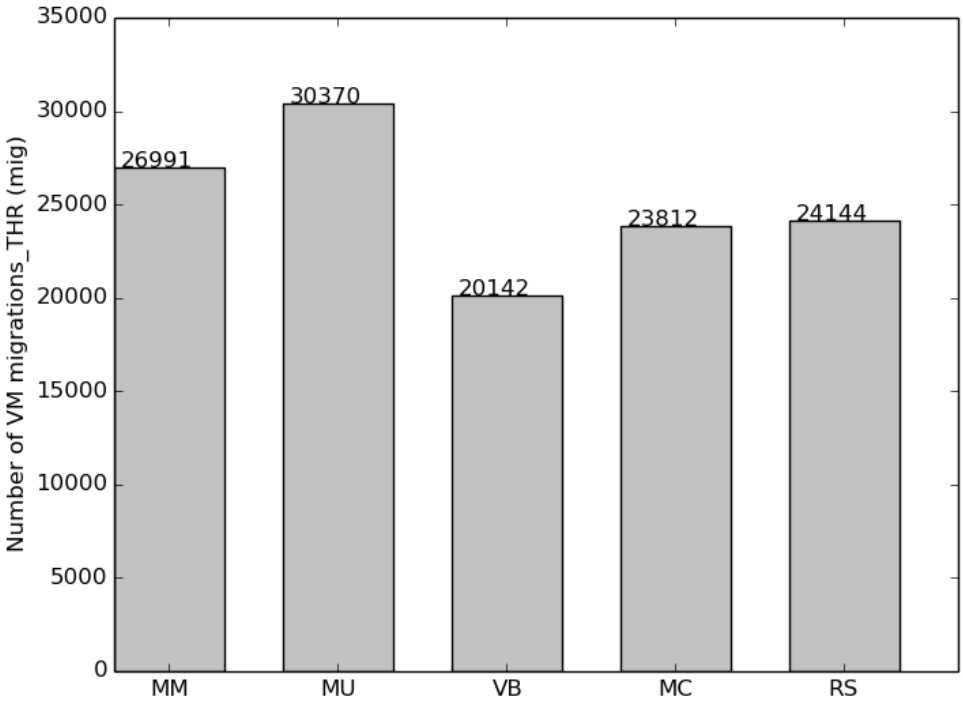


Figura 18 – Número de Migrações usando política de alocação THR

necessário para completar a migração, pois a política de Minimização de Migrações, ao escolher as máquinas virtuais que consomem menos tempo para serem migradas, reduz o período de penalidade sobre a CPU das VMs e da métrica PDM. As Figuras 19 e 20 mostram os resultados encontrados para a métrica em questão.

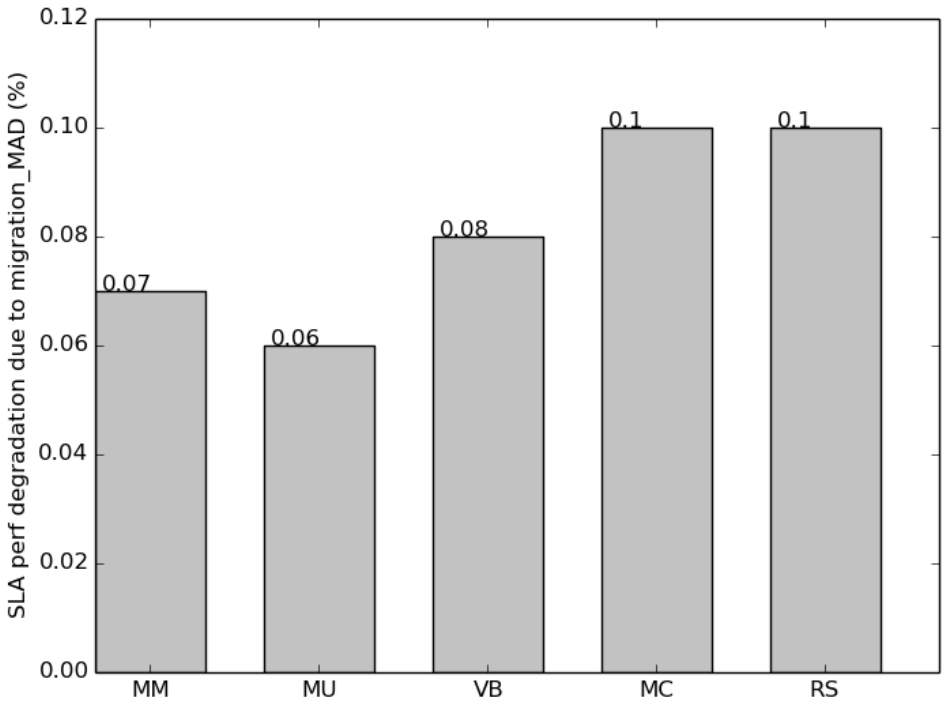


Figura 19 – Métrica PDM usando política de alocação MAD

5.4.4 Métrica SLATAH

Esta métrica avaliadora foi a única, dentre as quatro, que se mostrou completamente desfavorável a política VBalance (Figuras 21 e 22). Dentre as cinco propostas analisadas, a política de seleção sugerida neste trabalho foi a que apresentou o pior desempenho. Foi creditado tal comportamento as ações da VBalance de abortar o processo de migração ao se negar a escolher uma máquina virtual, em especial, o caso de um número mínimo de VMs nativas, pois caso essa quantidade mínima não seja atingida, um servidor precisará esperar, pelo menos, por mais uma interação para decidir qual VM migrar. Se esse servidor estivesse com uso máximo de CPU, esta interação adicional, que não ocorreria nas outras políticas, seria computada na métrica SLATAH da VBalance. A criação de relacionamentos com baixo Coeficiente de Determinação (R^2) também contribui para tais resultados, contudo em menor escala, pois baseado em observações, constatou-se um pequeno número de relacionamentos com R^2 inferior a 80%.

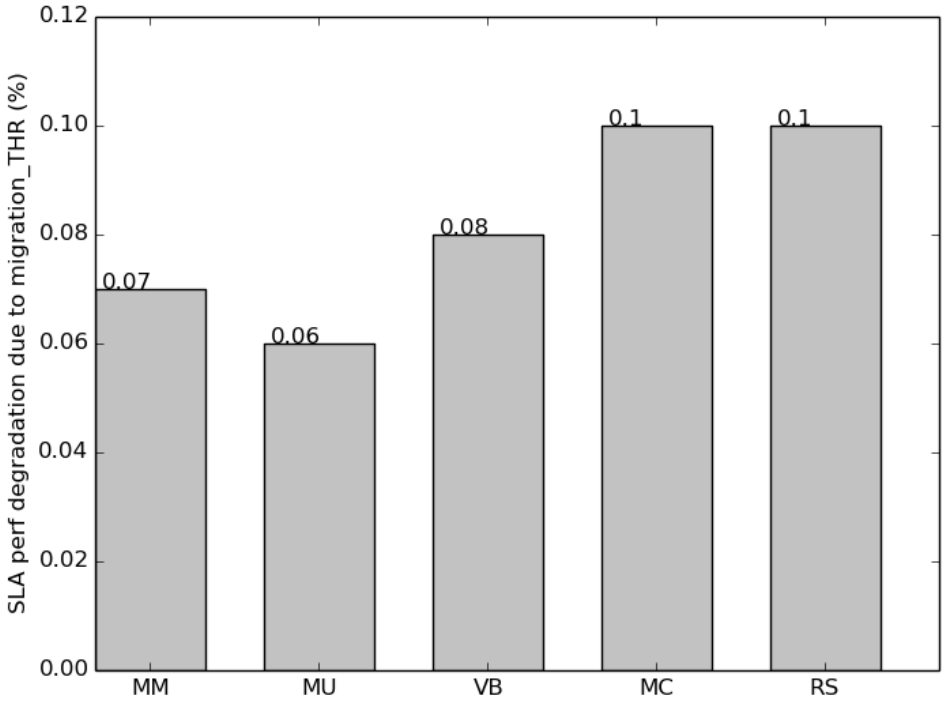


Figura 20 – Métrica PDM usando política de alocação THR

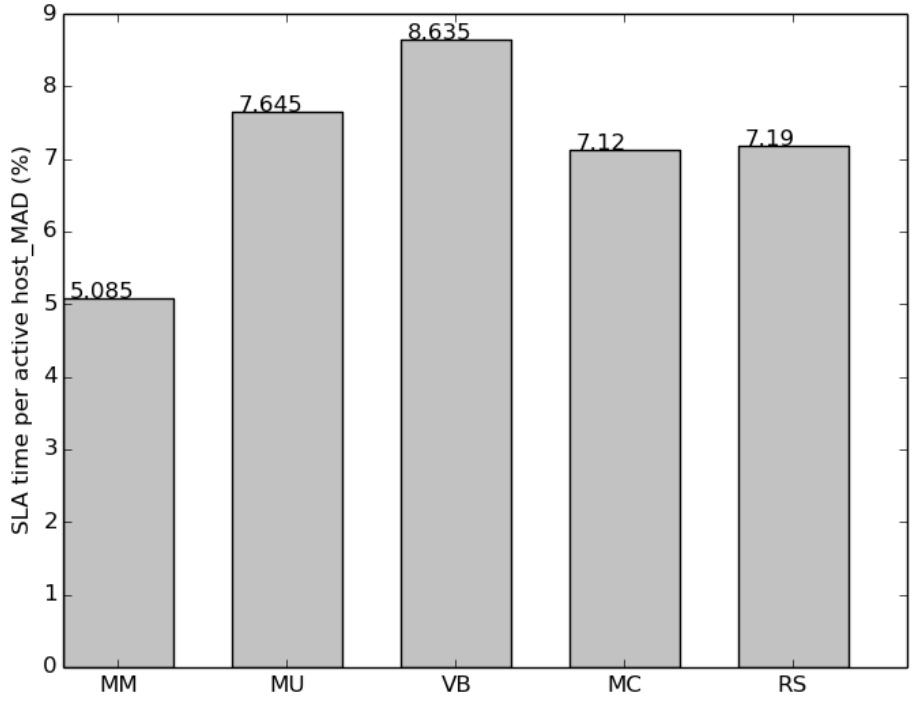


Figura 21 – Métrica SLATAH usando política de alocação MAD

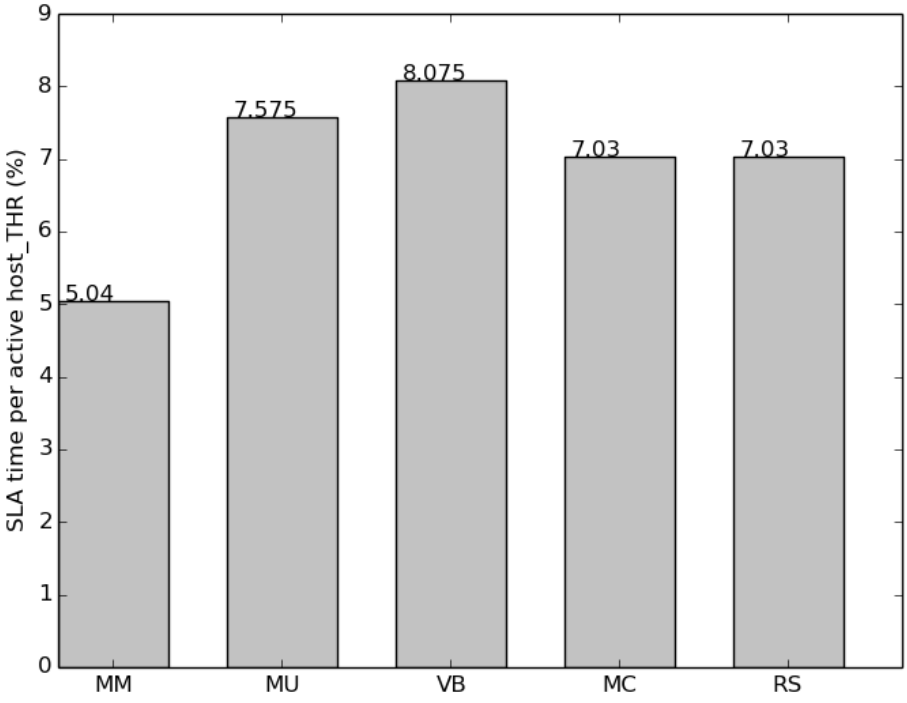


Figura 22 – Métrica SLATAH usando política de alocação THR

6 CONCLUSÃO

Este trabalho consistiu em definir uma política de seleção de máquinas virtuais, denominada VBalance, visando o balanceamento de cargas de trabalho entre servidores de um *data center* em um ambiente de Computação em Nuvens. Ao contrário das demais soluções encontradas no estado da arte, a política VBalance realiza sua escolha analisando o nível de consumo de energia das VMs em um servidor sobrecarregado e escolhendo, para migração, aquela que apresentar o maior grau de contribuição energética. Essa análise energética representa a contribuição dada por esse trabalho.

Através dos resultados obtidos, foi mostrado um desempenho satisfatório da política proposta quando comparada com outras soluções utilizadas tendo, como principal métrica, o consumo de energia. O fato da VBalance se negar a escolher máquinas virtuais para migração algumas vezes, se mostrou benéfico para o comportamento geral da política, uma vez que proporcionou uma economia de energia para o *data center* e reduziu, significativamente, o número de migrações. Os resultados observados nas métricas de desempenho (SLATAH e PDM) não chegam a ser tão preocupantes, pois apresentaram, no máximo, um acréscimo de 1% em relação a segunda política pior colocada. Desta maneira, concluiu-se como bom o custo-benefício da aplicação da política VBalance.

Como trabalhos futuros, podem ser realizados testes em ambientes emulados visando uma maior credibilidade na eficiência da política VBalance em relação as demais políticas de seleção analisadas. Também merece atenção os parâmetros de entrada da política sugerida, onde pode ser realizado um estudo matemático para determinar os valores ótimos a fim de gerar boas soluções. Outra melhoria destacável, na política de seleção VBalance, seria extendê-la para também ponderar questões de consumo de recursos durante o processo de escolha das máquinas virtuais.

Outro estudo que poderia ser realizado no futuro é o do comportamento da política de seleção VBalance em conjunto com políticas de alocação de máquinas virtuais que averiguam a eficiência energética dos servidores. Pois, uma vez que a VBalance escolhe as máquinas virtuais com maior nível de contribuição energética, seria interessante enviá-las (ou, até mesmo, mantê-las) em servidores com elevado poder computacional, porém que necessitam de pouca energia para funcionar.

REFERÊNCIAS

ADRIEL, Kotviski. *Processador em chamas!* 2009. Acesso em: 20 março 2014. Disponível em: <http://www.tecmundo.com.br/hardware/1766-processador-em-chamas-.htm>. Citado na página 24.

AMARANTE, Silvio Roberto Martins. *Utilizando o problema de múltiplas mochilas para modelar o problema de alocação de máquinas virtuais em Computação nas Nuvens*. Dissertação (Mestrado) — Universidade Estadual do Ceará, Fortaleza, Ceará, 2013. Citado 4 vezes nas páginas 22, 23, 35 e 47.

AMAZON. *Amazon EC2 - Amazon Elastic Compute Cloud*. 2013. Acesso em: 20 março 2014. Disponível em: <http://aws.amazon.com/pt/ec2/>. Citado 2 vezes nas páginas 18 e 50.

ARMBRUST, Michael; FOX, Armando; GRIFFITH, Rean; JOSEPH, Anthony D.; KATZ, Randy H.; KONWINSKI, Andrew; LEE, Gunho; PATTERSON, David A.; RABKIN, Ariel; STOICA, Ion; ZAHARIA, Matei. *Above the Clouds: A Berkeley view of cloud computing*. [S.l.], 2009. Acesso em: 20 março 2014. Disponível em: <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html>. Citado na página 10.

BAGULEY, Joe. How cloud computing is changing the world... without you knowing. *The Guardian*, 2013. Citado na página 14.

BARROSO, Luiz André; HöLZLE, Urs. The case for energy-proportional computing. *Computer*, IEEE Computer Society, 2007. Acesso em: 20 março 2014. Disponível em: <http://impact.asu.edu/cse591sp11/Barroso07_EnergyProp-clean.pdf>. Citado 2 vezes nas páginas 11 e 24.

BELADY, Christian L. *In the data center, power and cooling costs more than the it equipment it supports*. 2007. Acesso em: 20 março 2014. Disponível em: <http://www.electronics-cooling.com/articles/2007/feb/a3/>. Citado na página 24.

BELOGLAZOV, Anton; BUYYA, Rajkumar. Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers. In: *Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science*. New York, NY, USA: ACM, 2010. p. 4|1–4|6. Citado 2 vezes nas páginas 39 e 50.

BELOGLAZOV, Anton; BUYYA, Rajkumar. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. *Concurr. Comput. : Pract. Exper.*, v. 24, n. 13, p. 1397–1420, set. 2012. Acesso em: 20 março 2014. Disponível em: <http://dx.doi.org/10.1002/cpe.1867>. Citado 15 vezes nas páginas 10, 21, 24, 28, 35, 41, 47, 48, 49, 50, 51, 52, 53, 54 e 55.

BUYYA, Rajkumar; BELOGLAZOV, Anton; ABAWAJY, Jemal H. Energy-efficient management of data center resources for cloud computing: a vision, architectural elements, and open challenges. *Computing Research Repository*, 2010. Acesso em: 20 março 2014. Disponível em: <http://dblp.uni-trier.de/db/journals/corr/corr1006.html#abs-1006-0308>. Citado 7 vezes nas páginas 18, 25, 26, 28, 39, 41 e 55.

CALHEIROS, Rodrigo N.; RANJAN, Rajiv; BELOGLAZOV, Anton; ROSE, César A. F. De; BUYYA, Rajkumar. Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Softw. Pract. Exper.*, v. 41, n. 1, p. 23–50, jan. 2011. Acesso em: 20 março 2014. Disponível em: <http://dx.doi.org/10.1002/spe.995>. Citado 3 vezes nas páginas 47, 48 e 49.

C.GUO; LU, G.; WANG, H.; YANG, S.; KONG, C.; SUN, P.; WU, W.; Y.ZHANG. Secondnet: A data center network virtualization architecture with bandwidth guarantees. In: *Co-NEXT '10 Proceedings of the 6th International Conference on emerging Networking Experiments and Technologies*. Philadelphia, USA: ACM, New York, USA, 2010. Citado na página 10.

CLARK, D. *Understanding Canonical Correlation Analysis*. [s.n.], 1975. Acesso em: 20 março 2014. Disponível em: <http://books.google.com.br/books?id=HV5sQgAACAAJ>. Citado na página 37.

CONSORTIUM, PlanetLab. *PlanetLab: An open platform for developing, deploying, and accessing planetary-scale services*. 2013. Acesso em: 20 março 2014. Disponível em: <https://www.planet-lab.org/consortium>. Citado na página 50.

DO, Anh Vu; CHEN, Junliang; WANG, Chen; LEE, Young Choon; ZOMAYA, Albert Y.; ZHOU, Bing Bing. Profiling applications for virtual machine placement in clouds. In: LIU, Ling; PARASHAR, Manish (Ed.). *IEEE CLOUD*. [S.l.]: IEEE, 2011. p. 660–667. Citado 3 vezes nas páginas 28, 37 e 38.

DORIGO, Marco; BIRATTARI, Mauro; STüTZLE, Thomas. Ant colony optimization – artificial ants as a computational intelligence technique. *IEEE COMPUT. INTELL. MAG*, v. 1, p. 28–39, 2006. Citado na página 35.

DUSTIN, Amrhein; SCOTT, Quint. *Cloud computing for the enterprise: Part 1: Capturing the cloud*. [S.l.], 2009. Acessado em: 11 novembro 2013. Disponível em: <http://www.ibm.com/developerworks/websphere/techjournal/0904_amrhein/0904_amrhein.html>. Citado na página 17.

FAN, Xiaobo; WEBER, Wolf-Dietrich; BARROSO, Luiz Andre. Power provisioning for a warehouse-sized computer. In: *Proceedings of the 34th Annual International Symposium on Computer Architecture*. San Diego, USA: ACM, 2007. (ISCA '07), p. 13–23. Citado 2 vezes nas páginas 11 e 24.

FOUNDATION, Linux. *Xen Project*. [S.l.], 2013. Acesso em: 20 março 2014. Disponível em: <http://www.xenproject.org/>. Citado na página 21.

GELINAS, Jacques. *Linux VServer*. 2003. Acesso em: 20 março 2014. Disponível em: <http://linux-vserver.org/>. Citado na página 21.

GOOGLE. *Google App Engine*. 2011. Acesso em: 20 março 2014. Disponível em: <http://code.google.com/appengine/>. Citado na página 18.

GOOGLE. *Google Apps*. 2013. Acesso em: 20 março 2014. Disponível em: <http://www.google.com/Apps/>. Citado na página 15.

GOOGLE. *Google Drive*. 2013. Acesso em: 20 março 2014. Disponível em: <http://www.google.com/drive/>. Citado na página 18.

HASSAN, Qusay F. Demystifying cloud computing. *CrossTalk: The Journal of Defense Software Engineering*, Janeiro 2011. Acesso em: 20 março 2014. Disponível em: <http://www.crosstalkonline.org/storage/issue-archives/2011/201101/201101-Hassan.pdf>. Citado na página 14.

HOWELL, Fred; MCNAB, Ross. Simjava: A discrete event simulation library for java. In: *In International Conference on Web-Based Modeling and Simulation*. [S.l.: s.n.], 1998. p. 51–56. Citado na página 48.

IBM. *IBM Cloud*. 2013. Acesso em: 20 março 2014. Disponível em: <http://www.ibm.com\/Cloud/br>. Citado na página 18.

JAIN, Raj. *The art of computer systems performance analysis - techniques for experimental design, measurement, simulation, and modeling*. [S.l.]: Wiley, 1991. Citado 6 vezes nas páginas 28, 33, 37, 47, 48 e 51.

KELLERER, H.; PFERSCHY, U.; PISINGER, D. *Knapsack Problems*. [S.l.]: Springer, 2004. Citado na página 35.

LAGO, Daniel Guimaraes do; MADEIRA, Edmundo R. M.; BITTENCOURT, Luiz Fernando. Power-aware virtual machine scheduling on clouds using active cooling control and dvfs. In: *Proceedings of the 9th International Workshop on Middleware for Grids, Clouds and e-Science*. Lisbon, Portugal: ACM, 2011. p. 2|1–2|6. Citado na página 24.

MARTELLO, Silvano; TOTH, Paolo. *Knapsack Problems: algorithms and computer implementations*. New York, NY, USA: John Wiley & Sons, Inc., 1990. Citado na página 35.

MATTHEWS, Jeanna N; DOW, Eli M; DESHANE, Todd; HU, Wenjin; BONGIO, Jeremy; WILBUR, Patrick F; JOHNSON, Brendan. *Running Xen: a hands-on Guide to the art of virtualization*. São Francisco, CA, EUA: Prentice Hall, 2008. Citado na página 20.

MICROSOFT. *Microsoft Office 2010*. 2013. Acesso em: 20 março 2014. Disponível em: <http://office.microsoft.com>. Citado na página 18.

MORENO-VOZMEDIANO, Rafael; MONTERO, Ruben S.; LLORENTE, Ignacio M. Key challenges in cloud computing: enabling the future internet of services. *IEEE Internet Computing*, v. 17, n. 4, p. 18–25, 2013. Citado na página 14.

MURUGESAN, San. Harnessing green it: principles and practices. *IT Professional*, Piscataway, NJ, USA, v. 10, n. 1, p. 24–33, jan. 2008. Acesso em: 20 março 2014. Disponível em: <http://dx.doi.org/10.1109/MITP.2008.10>. Citado na página 25.

NANDA, Susanta; CHIUEH, Tzi cker. *A survey on virtualization technologies*. [S.l.: s.n.], 2005. Citado na página 19.

NAYAK, Smitha; YASSIR, Ammar. *Cloud computing as an emerging paradigm*. [S.l.: s.n.], 2012. Citado na página 10.

ORACLE. *VirtualBox*. 2013. Acesso em: 20 março 2014. Disponível em: <https://www.virtualbox.org/>. Citado na página 21.

PARK, KyoungSoo; PAI, Vivek S. Comon: A mostly-scalable monitoring system for planetlab. *SIGOPS Oper. Syst. Rev.*, ACM, v. 40, n. 1, p. 65–74, jan. 2006. Acesso em: 20 março 2014. Disponível em: <http://doi.acm.org/10.1145/1113361.1113374>. Citado na página 50.

PATEL, C. D.; BASH, C. E.; SHARMA, R.; BEITELMAL, M. Smart cooling of data centers. In: *Proceedings of IPACK*. [S.l.: s.n.], 2003. Citado na página 24.

SALESFORCE. *Salesforce.com*. 2013. Acesso em: 20 março 2014. Disponível em: <http://www.salesforce.com/>. Citado na página 18.

SHRIBMAN, Aidan; HUDZIA, Benoit. Pre-copy and post-copy vm live migration for memory intensive applications. In: *Euro-Par Workshops*. [S.l.]: Springer, 2012. p. 539–547. Citado na página 22.

SOARES, Paolo Victor Gonçalves. *Gatekeeper: controle de tráfego distribuído em datacenters virtualizados*. Dissertação (Mestrado) — Universidade Federal de Minas Gerais, Belo Horizonte, Minas Gerais, 2010. Citado na página 20.

SUEUR, Etienne Le; HEISER, Gernot. Dynamic voltage and frequency scaling: the laws of diminishing returns. In: *Proceedings of the 2010 International Conference on Power Aware Computing and Systems*. Vancouver, BC, Canada: [s.n.], 2010. Citado na página 24.

SULLIVAN, Robert F. Alternating cold and hot aisles provides more reliable cooling for server farms. *Site Infrastructure White Paper*, Santa Fe, USA, 2002. Citado na página 25.

TAKEMURA, Chris; CRAWFORD, Luke S. *The Book of Xen: a pratical guide for the system administrator*. Boston, MA, EUA: No Starch Press, 2010. Citado 2 vezes nas páginas 20 e 22.

TANENBAUM, Andrew S; OTEEN, Maarten van. *Sistemas distribuídos: princípios e paradigmas*. 2. ed. [S.l.]: Prentice Hall, 2007. Citado na página 19.

TAURION, Cezar. *Cloud Computing: Computação em Nuvem*. Rio de Janeiro, RJ: BRASPORT, 2009. Citado na página 16.

VERDI, Fábio Luciano; ROTHENBERG, Christian Esteve; PASQUINI, Rafael; MAGALHãES, Mauricio. Novas arquiteturas de data center para cloud computing. In: *SBRC 2010 - Minicursos*. [S.l.: s.n.], 2010. Citado na página 18.

VERMA, Akshat; AHUJA, Puneet; NEOGI, Anindya. Pmapper: power and migration cost aware application placement in virtualized systems. In: *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*. Leuven, Belgium: [s.n.], 2008. p. 243–264. Citado na página 38.

VERMA, Akshat; DASGUPTA, Gargi; NAYAK, Tapan Kumar; DE, Pradipta; KOTHARI, Ravi. Server workload analysis for power minimization using consolidation. In: *Proceedings of the 2009 Conference on USENIX Annual Technical Conference*. San Diego, California: [s.n.], 2009. p. 28–28. Citado 3 vezes nas páginas 28, 38 e 41.

VINCENZO, Bongiovanni; VISSOTO, Olímpio; LAUREANO, José. *Matemática*. [S.l.]: Ática, 1994. Citado na página 34.

VMWARE. *VMWare*. 2013. Acesso em: 20 março 2014. Disponível em: <http://www.vmware.com/br\/>. Citado na página 21.

ZAGE, David; FRANKLIN, Dustin; VINCENT, Urias. What does the future hold for cloud computing? *CrossTalk: The Journal of Defense Software Engineering*, 2013. Acesso em: 20 março 2014. Disponível em: <http://www.crosstalkonline.org/storage/issue-archives/2013/201309/201309-Zage.pdf>. Citado na página 17.

ZHANG, Qi; CHENG, Lu; BOUTABA, Raouf. Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, p. 7–18, Maio 2010. Citado 3 vezes nas páginas 10, 18 e 19.

ZHANG, Ying. *Future wireless networks and information systems*. [S.l.]: Springer Publishing Company, 2012. Citado na página 14.