

Migração de máquinas virtuais para economia de energia*

Leonardo Pais Cardoso, Lino Henrique G. Ferraz, Otto Carlos M. B. Duarte

¹Grupo de Teleinformática e Automação
Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro – RJ – Brasil

{cardoso, lino, otto}@gta.ufrj.br

Resumo. *A computação em nuvem oferece um novo modelo de negócios que provê serviços de infraestrutura flexíveis por demanda. O cliente paga apenas pelo que realmente necessita ou usa recursos de processamento, memória e banda sob demanda, podendo crescer ou retrair sem custos extras. Logo, os provedores de infraestrutura devem prover recursos de forma dinâmica, a custo reduzido e também com baixo consumo de energia elétrica, para diminuir a emissão de carbono e melhorar a sua imagem junto a seus clientes. Este artigo propõe um mecanismo que emprega técnicas de otimização para gerenciar a migração de máquinas virtuais entre máquinas físicas com o objetivo de reduzir os recursos ociosos e também obter menor consumo energético através do desligamento de máquinas físicas. O estado das máquinas e as decisões são baseadas no monitoramento dos perfis de uso de recursos. A heurística implementada procura minimizar o número de máquinas físicas em funcionamento ao realocar máquinas virtuais. Foram realizados experimentos em um protótipo desenvolvido e implantado no Future Internet Testbed with Security (FITS). Os resultados obtidos mostram a eficácia da proposta na redução do consumo energético.*

Abstract. *Cloud computing introduced a new business model that offers flexible on demand infrastructure services. The client gets resources on demand such as processing, memory and bandwidth, which can grow or shrink without extra costs and he pays just for what he needs or uses. Therefore, the infrastructure providers must offer resources dynamically with low costs and low energy consumption in order to decrease carbon emission levels and improve the image of the company in the consumer marketplace. This paper proposes a virtual machine management mechanism that implements optimization techniques to manage the migration of virtual machines between physical machines. The mechanism aims to decrease power consumption and idle resources by turning off physical machines. The machine state and the decisions taken are based on the monitoring of the resource usage profiles. The developed heuristic tries to minimize the number of active physical machines by redistributing virtual machines. A prototype was developed and deployed in Future Internet Testbed with Security (FITS). The results show the effectiveness of the mechanism reducing power consumption.*

*Este trabalho foi realizado com recursos do CNPq, CAPES, FINEP, FAPERJ e FUNTTEL.

1. Introdução

A computação em nuvem permite que o provedor de infraestrutura ofereça recursos computacionais como processamento, memória e rede de forma flexível e dinâmica de acordo com a demanda de seus clientes. Além de serem evitados custos operacionais de manutenção, configuração e reparo, o cliente também pode obter os recursos que necessita naquele momento, evitando sobre ou sub valores investidos em infraestrutura. Uma técnica utilizada para isto é a virtualização [Fernandes et al. 2010] que amplia a flexibilidade em alocar recursos ao abstrair o *hardware*, possibilitando que mais de um usuário compartilhe recursos de uma mesma máquina sem interferir nos processos de outro usuário. Para evitar sobrecarga, é comum distribuir os recursos pelas máquinas físicas interconectadas em rede ou em aglomerados. A migração de máquinas virtuais é uma forma eficaz de redistribuir dinamicamente os recursos pelas máquinas físicas existentes. A migração ao vivo [Clark et al. 2005] permite que as máquinas virtuais continuem em funcionamento durante o processo de migração, garantindo alta disponibilidade do serviço.

Alocar recursos computacionais é um importante desafio em computação em nuvem, pois determina se o provedor consegue atender aos Acordos de Nível de Serviço (*Service Level Agreements* - SLAs) sem comprometer os seus lucros. Uma alocação eficiente de recursos deve atender a todos os clientes com a qualidade de serviço por eles requerida e com o menor uso de recursos físicos para diminuir o número de computadores ativos, reduzindo assim o consumo de energia elétrica. No entanto, as plataformas de virtualização como o Xen [Egi et al. 2008] não possuem nativamente um mecanismo de gerência que possibilite alocar os recursos de forma eficiente. Além disso, alocar diferentes recursos em máquinas que possuem restrições de capacidade é um Problema Generalizado de Atribuição (*Generalized Assignment Problem* - GAP) e portanto NP-Difícil [Kundakcioglu e Alizamir 2009]. A medida que o número de recursos e máquinas aumenta, o tempo para calcular a solução cresce exponencialmente. Assim, encontrar uma solução exata não é escalável.

Este artigo propõe um mecanismo automático de gerência baseado na migração de máquinas virtuais para minimizar o consumo de energia elétrica e reduzir a ociosidade de recursos. A minimização do consumo de energia e da ociosidade é feita com base no monitoramento e análise dos perfis de uso de recursos das máquinas físicas e virtuais. A partir dessa análise é aplicada uma heurística que realoca recursos através da funcionalidade de migração de máquinas virtuais de forma a minimizar o número de máquinas físicas em funcionamento. O algoritmo provê um mapeamento das máquinas virtuais em máquinas físicas que objetiva um número reduzido de máquinas físicas. O mecanismo considera ainda a quantidade de memória a ser transferida pela rede. As máquinas virtuais são então migradas uma a uma para evitar violações dos Acordos de Nível de Serviço que poderiam ser causadas pela sobrecarga no tráfego na rede que as transferências de dados da migração requerem. As máquinas físicas em estado inativo são então desligadas com a finalidade de reduzir os recursos ociosos e o consumo de energia. As máquinas são mantidas desligadas até que a demanda dos clientes seja superior a quantidade de recursos oferecidos e, neste caso, novas máquinas físicas são ativadas para atender a demanda.

O mecanismo de gerência foi implementado e testado no FITS. Dois testes foram realizados: Um experimento para comprovar o funcionamento do mecanismo e um estudo

do algoritmo de otimização. Os resultados mostram que o mecanismo é capaz de encontrar soluções de menor custo, migrar as máquinas para essa solução, desligar máquinas ociosas e ligar máquinas quando a demanda é maior que a oferta. O estudo mostra que o Arrefecimento Simulado é capaz de melhorar a solução de outras heurísticas.

O restante deste artigo está organizado da seguinte forma. Na Seção 2 os trabalhos relacionados são descritos e comparados com o mecanismo proposto. A Seção 3 apresenta a solução proposta, descrevendo a coleta de dados, a heurística utilizada e a arquitetura do mecanismo. Os resultados são apresentados na Seção 4. A Seção 5 apresenta a conclusão e direções futuras deste trabalho.

2. Trabalhos Relacionados

O desenvolvimento de mecanismos que visem a minimização do consumo de energia é um tema atual e também um desafio tecnológico. Diversos trabalhos abordam a alocação de elementos virtuais sem levar em conta o consumo de energia elétrica.

Esta seção se restringe aos trabalhos de alocação de recursos para redução de consumo de energia elétrica. Uma forma de resolução do problema de otimização é o uso de heurísticas como o *First Fit Decreasing* (FFD) [Johnson et al. 1974] e de meta-heurísticas como o Arrefecimento Simulado. O FFD é um algoritmo guloso que ordena as máquinas virtuais de forma decrescente de demanda e tenta alocar a máquina virtual na máquina física de menor índice com capacidade disponível. Caso não consiga alocar a quantidade de recursos demandados em nenhuma máquina física ativa, uma nova máquina física é ativada com um índice maior que o da última ativada. A técnica de Arrefecimento Simulado é uma meta-heurística capaz de encontrar uma solução ótima para o problema de alocação através de uma função de aceitação. Essa função permite a aceitação de estados de maior custo, e assim possibilita a exploração de diferentes soluções para fugir de mínimos locais.

Wu *et al.* comparam as heurísticas FFD e Arrefecimento Simulado para a realocação de máquinas virtuais e economia de energia [Wu et al. 2012]. Wu *et al.* realizam simulações variando o número de máquinas físicas e virtuais bem como a capacidade e o uso de recursos de cada uma delas. A minimização é feita sobre uma função de consumo de energia elétrica dependente de parâmetros como o uso de processamento e memória e do *hardware* utilizado. Eles concluíram que o uso em conjunto do Arrefecimento Simulado e do FFD encontra soluções que podem economizar mais energia que o uso somente do FFD ou do Arrefecimento Simulado. Porém, os autores focam na proposta e simulação do algoritmo, assim o trabalho carece de um desenvolvimento prático que lide com os problemas relacionados de gerenciamento efetivo de um ambiente virtualizado. Apenas verificou-se a possibilidade de alocar as máquinas virtuais e o tempo que o algoritmo leva para atingir a solução, não considerando o tempo necessário para o sistema convergir após as requisições de migração, o que pode ser crítico em uma aplicação real.

Rodriguez *et al.* implementam uma heurística de *branch and cut* [Mitchell 2002] baseada em um algoritmo de programação linear 0-1 cujo objetivo é encontrar um mapeamento de roteadores e enlaces virtuais em roteadores e enlaces físicos. O artigo avalia o compromisso entre minimizar o consumo de energia e o uso de banda [Rodriguez et al. 2013]. Os autores definem pesos para o uso de banda e consumo de energia e executam duas formulações de programação linear 0-1 de forma sequencial.

A primeira formulação mapeia as redes virtuais no substrato a fim de minimizar o consumo de energia e a largura de banda por requisição da nova rede, a segunda visa encontrar um caminho que minimize o tempo necessário para transferir a imagem do roteador virtual e carregar o sistema operacional. Os autores apresentam melhores resultados quando aplicam o mesmo peso para banda e economia de energia, apresentando uma diminuição de 30% da largura de banda e um aumento de 10% do consumo quando comparado a minimização apenas do consumo de energia. A proposta de Rodriguez *et al.* não é orientada a computação em nuvem e considera apenas o problema de chegada de redes virtuais em uma plataforma pluralista para prover infraestrutura de rede. Em um cenário de nuvem não se pode limitar a instanciação de redes, sendo necessário manter a economia de energia dos servidores que entregam recursos sob demanda para as máquinas virtuais.

Este artigo, ao contrário dos trabalhos acima relacionados, propõe a solução de um mecanismo de gerência capaz de reduzir o número de máquinas em funcionamento utilizando heurísticas que levam em conta parâmetros que influem substancialmente no tempo de migração. O mecanismo proposto utiliza a técnica de migração ao vivo para reduzir a inatividade das máquinas virtuais enquanto são migradas. Com isto, quando ocorre a migração, apenas a memória da máquina virtual é copiada pela rede. Essa técnica permite que a máquina virtual permaneça em operação durante a migração e entre em estado suspenso por um curto período de tempo. O mecanismo também evita a sobrecarga da rede causada pela migração e a sobrecarga dos recursos de memória e processamento das máquinas físicas. A sobrecarga da rede é tratada através da escolha de soluções que migrem máquinas virtuais com menor quantidade de memória. A sobrecarga de memória e processamento é evitada pela utilização do protocolo *Wake on Lan* que liga máquinas pela rede. No mecanismo, o *Wake on Lan* é acionado quando a demanda por recursos é maior que a disponível, ligando uma das máquinas físicas ociosas para satisfazer a necessidade dos clientes. Os autores Rodriguez *et al.* procuram otimizar a instanciação de roteadores virtuais na criação da rede. Em contrapartida, este artigo aborda a migração de máquinas virtuais já instanciadas e em uso por clientes que possuem acordos de nível de serviço pré-definidos com o provedor de infraestrutura.

A verificação do comportamento do mecanismo proposto foi realizada no testbed FITS, que provê um ambiente virtualizado em que se pode testar soluções para a Internet do Futuro [Mattos et al. 2012, Moraes et al. 2014]. O FITS é uma rede de testes inter-universitária baseada nas tecnologias Xen e OpenFlow, contando com parceiros no Brasil e na Europa. O FITS adota o paradigma pluralista e permite a execução de sistemas operacionais e aplicações distintas nas redes virtuais. Dessa forma, o FITS provê um ambiente propício a avaliação do desempenho e da viabilidade de novas tecnologias para a Internet do Futuro. A arquitetura do FITS pode ser vista na Figura 1.

A implementação desta proposta se serve de módulos desenvolvidos anteriormente no Grupo de Teleinformática e Automação referentes ao VOLTAIC [Carvalho e Duarte 2012] e ao sistema de gerência AMAS proposto por Bezerra *et al.* [Bezerra et al. 2014]. O VOLTAIC é um sistema de gerência de recursos para a computação em nuvem que provê qualidade de serviço e evita desperdício de recursos. Bezerra *et al.* propuseram uma ferramenta de gerência que evita a sobrecarga dos nós físicos e violações de acordo de nível de serviço. Este trabalho, assim como o VOLTAIC, se baseia na técnica de perfis de uso de recursos das máquinas físicas para analisar

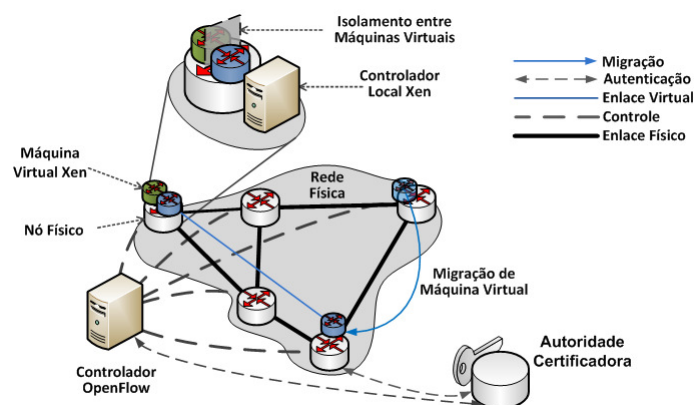


Figura 1. Arquitetura do FITS destacando a migração de uma máquina virtual.
Adaptado de [Figueiredo et al. 2013] [Mattos et al. 2012, Moraes et al. 2014].

as capacidades e, assim como a ferramenta de gerência desenvolvida, redistribui a carga de trabalho através da migração de máquinas virtuais caso um limiar de consumo seja atingido.

3. O Esquema Proposto

A proposta deste artigo visa minimizar o consumo de energia elétrica e os recursos ociosos. Fan *et al.* afirmam que um servidor em modo inativo não consome menos que 50% da energia elétrica que consumiria em fase de pico [Fan et al. 2007]. Portanto, para reduzir de forma significativa o consumo de energia deve-se desligar por completo a máquina física. Assim, a ideia-chave é migrar as máquinas virtuais para um número reduzido de máquinas físicas e desligar as máquinas físicas que não possuem máquinas virtuais ativas. O mecanismo desenvolvido possui quatro módulos principais: o Monitor de Recursos, o Otimizador, o Orquestrador de Migração e o Gerenciador de Energia, que são detalhados na Seção 3.3. Esses módulos geram perfis, calculam um estado que economize energia, migram máquinas virtuais e desligam e ligam as máquinas físicas.

A quantidade de memória de cada máquina virtual a ser migrada influencia no tempo de convergência do sistema, porque ao migrar uma máquina virtual é necessário transferir a sua memória pela rede para a máquina física que a hospedará. Para reduzir o tempo de migração, o algoritmo de Arrefecimento Simulado implementado considera, na condição de duas ou mais soluções de mesmo custo em termos de máquinas físicas, escolher a solução que transfere menor quantidade de memória. Logo, sempre que uma solução de mesmo custo que a última armazenada é encontrada, verifica-se quais máquinas virtuais precisam migrar, ou seja, trocaram de posição em relação a disposição inicial, e também se verifica o quanto de memória essas máquinas possuem. Após o cálculo escolhe-se e armazena-se o plano de migração com menor quantidade de memória para que o procedimento de migração provoque a menor sobrecarga possível na rede.

3.1. Formulação para minimização do número de máquinas em funcionamento

Para a otimização são definidos os seguintes parâmetros:

- V - conjunto de máquinas virtuais instanciadas;
- F - conjunto de máquinas físicas ativas;

- $C_v, v \in V$ - processamento utilizado pela máquina virtual v ;
- $C_f, f \in F$ - processamento utilizado pelo Domínio-0 da máquina física f ;
- $T_{C|f}, f \in F$ - limiar de uso de processamento da máquina física f ;
- $M_v, v \in V$ - memória da máquina virtual v ;
- $M_f, f \in F$ - memória utilizada pelo Domínio-0 da máquina física f ;
- $T_{M|f}, f \in F$ - limiar de uso de memória da máquina física f ;
- $N_v, v \in V$ - consumo de rede da máquina virtual v ;
- $N_f, f \in F$ - rede utilizada pelo Domínio-0 da máquina física f ;
- $T_{N|f}, f \in F$ - limiar de uso de rede da máquina física f ;
- $X(f, v) = \begin{cases} 1, & \text{se } v \in V \text{ está instanciada na máquina física } f \in F \\ 0, & \text{senão} \end{cases}$
- $X(f) = \begin{cases} 1, & \text{se } \sum_{v \in V} X(f, v) \geq 1 \\ 0, & \text{senão} \end{cases}$

O algoritmo de minimização segue o procedimento de

Minimizar

$$\sum_{f \in F} X(f) \quad (1)$$

sujeito às restrições:

$$\forall f \in F, \left(\sum_{v \in V} C_v * X(f, v) \right) + C_f \leq T_{C|f} \quad (R1)$$

$$\forall f \in F, \left(\sum_{v \in V} M_v * X(f, v) \right) + M_f \leq T_{M|f} \quad (R2)$$

$$\forall f \in F, \left(\sum_{v \in V} N_v * X(f, v) \right) + N_f \leq T_{N|f} \quad (R3)$$

$$\forall v \in V, \sum_{f \in F} X(f, v) = 1 \quad (R4)$$

$$\{V, F\} \subset \mathbb{Z}_{>0} \quad (R5)$$

A Equação 1 é a função objetivo do problema que nesse artigo é a minimização do número de máquinas físicas em funcionamento $X(f)$, ou seja, o número total de máquinas físicas ativas.

As restrições R1, R2 e R3 garantem que o uso de recursos não ultrapasse um limiar. Assim, o uso de recursos para qualquer máquina física $f \in F$ em funcionamento, é necessariamente menor ou igual ao somatório dos recursos utilizados pelas máquinas virtuais $v \in V$ pertencentes a máquina f , o que é representado por $X(f, v)$, e pelo uso de recursos do Domínio-0 da máquina f . A Equação R4 restringe uma máquina virtual v pertencer a apenas uma máquina física f . Dessa maneira, o somatório das máquinas físicas f as quais possuem a máquina virtual v deve ser igual a 1. Essa restrição impede que uma mesma máquina virtual pertencente ao conjunto de máquinas virtuais instanciadas V seja instanciada em duas máquinas físicas ao mesmo tempo e garante que a máquina virtual v terá uma máquina física f como destino. A restrição R5 limita o conjunto de máquinas virtuais V e o de máquinas físicas F ao domínio dos inteiros maiores que zero.

3.2. Arrefecimento Simulado

A meta-heurística de Arrefecimento Simulado implementada considera um estado que evidencia a distribuição das máquinas virtuais sobre as máquinas físicas. Inicialmente fixa-se uma temperatura inicial, gera-se aleatoriamente um estado inicial e calcula-se o custo associado definido pelo número de máquinas físicas ligadas. Em seguida, gera-se um novo estado a partir de uma perturbação do estado inicial. Essa perturbação consiste em escolher uma das máquinas virtuais e instanciá-la em uma máquina física diferente, aleatoriamente. Caso o novo estado possua um custo menor que o anterior ele é aceito. Do contrário, a aceitação dependerá dos custos do estado anterior e do gerado, e da temperatura. A função de aceitação é expressa por

$$P = \exp\left(-\frac{j_i - j_{i-1}}{\tau_i}\right), \quad (3)$$

onde j_i e τ_i são custo e temperatura na iteração i , respectivamente.

A Equação 3 representa a probabilidade de considerar uma solução de maior custo, para isso geralmente compara-se o valor da função com um valor aleatório no intervalo $[0, 1]$. Se o custo diminui ou se mantém a expressão é sempre igual a 1 e o estado é aceito. Para temperaturas elevadas o valor da função de aceitação tem maior probabilidade de ser maior que o valor aleatório. Dessa forma, o algoritmo tem maior probabilidade de sair de um mínimo local. As soluções aceitas são utilizadas para gerar novos estados. O algoritmo armazena a solução aceita como a solução candidata se o custo diminuir ou o custo se mantiver e a memória das máquinas a migrar diminuir. A escolha da solução de menor memória é feita para evitar sobrecarregar a rede com a transferência das máquinas virtuais. A medida que a temperatura diminui a função de aceitação retorna valores menores e o algoritmo converge para uma solução. Ao final, a solução do problema será a última solução candidata. O pseudocódigo do algoritmo está em Algoritmo 1.

```
Dados:  $\tau$ ; distribuicao, menor_custo, ultimo_custo,  
ultimo_estado, custo_memoria  
enquanto  $\tau > 0$  faça  
    nova_distribuicao = gerar_estado(ultimo_estado)  
    novo_custo = custo(nova_distribuicao)  
    se aceitacao(ultimo_custo, novo_custo, tau)  $\geq$  random() então  
        ultimo_estado = nova_distribuicao  
        se novo_custo < menor_custo ou (novo_custo = menor_custo e  
        custo_memoria(nova_distribuicao) < custo_memoria) então  
            distribuicao = nova_distribuicao  
            menor_custo = novo_custo  
            custo_memoria = custo_memoria(nova_distribuicao)  
        fim  
        ultimo_custo = novo_custo  
    fim  
     $\tau = \tau - 1$   
fim  
retorna distribuicao
```

Algoritmo 1: Arrefecimento Simulado

3.3. Arquitetura do Mecanismo

O mecanismo possui quatro módulos principais como pode ser visto na Figura 2. O Monitor de Recursos é responsável por coletar o uso de recursos das máquinas físicas e virtuais, o Otimizador executa a heurística para minimizar o número de máquinas físicas em funcionamento, o Orquestrador de Migração faz a redistribuição das máquinas virtuais entre as máquinas físicas e o Gerenciador de Energia desliga e liga máquinas físicas de acordo com a demanda. O Monitor de Recursos e o Orquestrador de Migração foram desenvolvidos no trabalho de Bezerra *et al.* [Bezerra et al. 2014] e integrados aos módulos Otimizador e Gerenciador de Energia desenvolvidos nesse artigo. O mecanismo foi desenvolvido em Python para facilitar a integração com o Monitor de Recursos e o Orquestrador de Migração que foram desenvolvidos nessa linguagem.

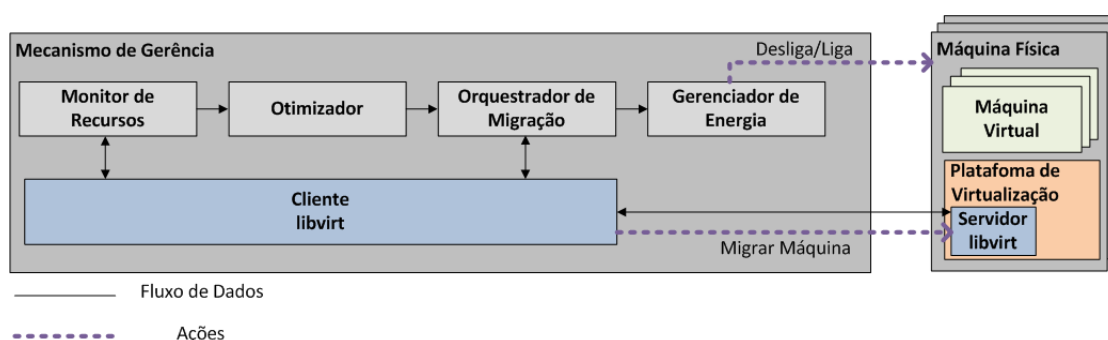


Figura 2. Arquitetura do mecanismo de gerência proposto e do esquema de migração automática de máquinas virtuais.

O Monitor de Recursos coleta o uso de CPU, memória e banda passando das máquinas físicas e virtuais através da biblioteca libvirt para se comunicar com os hipervisores de cada máquina física. O uso de CPU das máquinas virtuais é obtido diretamente pela libvirt. O processamento das máquinas físicas é calculado com a soma do processamento das máquinas virtuais. O perfil de uso de memória é obtido através da quantidade de memória alocada nas máquinas físicas e virtuais. O uso de rede das máquinas virtuais é coletado a partir da quantidade de dados que trafegam pelas interfaces virtuais. Como a solução atual não considera a topologia da rede, o uso de banda das máquinas virtuais contribui apenas no processamento do Domínio-0. O perfil de uso de recursos é gerado através do monitoramento das máquinas físicas e virtuais a cada segundo.

O Otimizador recebe os perfis de uso do Monitor de Recursos e executa a meta-heurística de Arrefecimento Simulado para minimizar o número de máquinas físicas ativas. Ao final, o Otimizador gera uma nova distribuição de máquinas físicas e virtuais. Os dados das máquinas virtuais que serão migradas e os dados das máquinas físicas e virtuais são então enviados para o Orquestrador de Migração.

O Orquestrador de Migração é responsável por gerenciar a migração das máquinas virtuais para a distribuição gerada pelo módulo de otimização. O Orquestrador usa a migração ao vivo do Xen com pré-cópia que se dá em duas fases. Na primeira fase as páginas de memória da máquina virtual são copiadas para a máquina de destino, se uma página é modificada ela é reenviada. A segunda fase inicia quando a taxa de reenvio é menor que a taxa de modificação. A máquina virtual é suspensa na origem e o restante das páginas modificadas é copiado para a máquina de destino. Ao final, a execução da

máquina virtual é retomada na máquina de destino. Dessa forma, o tempo de migração dependerá do tamanho da memória a ser transferida, da taxa de atualização dos dados na memória e do uso de recursos das máquinas físicas envolvidas na migração. Para evitar a sobrecarga do uso de rede devido a migração, uma máquina virtual só pode ser migrada quando a migração da máquina virtual anterior terminar.

O Gerenciador de Energia desliga as máquinas físicas que após a execução das migrações não apresentam máquinas virtuais instanciadas. Esse módulo também liga as máquinas físicas quando não é mais possível atender a todos os clientes devido a um aumento da demanda de recursos. Esse aumento pode ser observado quando o consumo de recursos de uma máquina física atinge determinado limiar em um determinado número de medições, o que caracteriza uma sobrecarga. Caso o algoritmo de otimização não encontre uma solução de menor custo, uma máquina física é religada através do protocolo *Wake on Lan* e as máquinas virtuais são redistribuídas após uma nova execução do algoritmo. O pseudocódigo desse módulo está em Algoritmo 2.

```

Entrada: HostsOciosos, MaquinasDesligadas,
MaquinasFisicas
se sobrecarga(MaquinasFisicas) e HostsOciosos = 0 então
    | acordar(MaquinasDesligadas.pop())
    | Otimizador(MaquinasFisicas)
fim
senão
    | para Maquina em MaquinasFisicas faça
    |   | se MaquinasVirtuais(Maquina) = 0 então
    |   |   | desligar(Maquina)
    |   |   | MaquinasDesligadas.append(Maquina)
    |   |   fim
    |   fim
fim
fim

```

Algoritmo 2: Gerenciador de Energia

Uma máquina virtual é migrada para uma máquina física apenas se a máquina física possui recursos suficientes para alocar a máquina virtual. Do contrário, outra máquina é escolhida como destino. Se antes da migração o uso de recursos da máquina virtual é superior ao da máquina física de destino, a máquina virtual não migra para esse destino. Caso a máquina virtual oscile após a migração e ultrapasse um determinado limiar, a sobrecarga é detectada com base no histórico dos perfis de uso. As máquinas virtuais são então redistribuídas. O mecanismo não considera a topologia da rede para efetuar as migrações. Porém, o escopo de máquinas físicas participantes do processo de migração é configurável pelo administrador que pode estabelecer um grupo máquinas físicas geograficamente próximas. Assim, a latência da máquina virtual em relação ao cliente no novo destino pode ser reduzida.

A coleta dos perfis de uso e a migração é realizada pela biblioteca multiplataforma libvirt. Dessa forma, o mecanismo de gerência pode ser usado em plataformas de virtualização como o Xen e o KVM. Além disso, as medidas para a geração de perfis são

obtidas através da libvirt, o que não requer a modificação de nenhuma máquina virtual, nem a instalação de *softwares*. Assim, preserva o isolamento das máquinas virtuais. O agendamento de execução do mecanismo depende da política do administrador podendo ser mantido em execução contínua ou apenas quando um evento ocorre. Nesse caso, deve ser levado em conta o tempo que o algoritmo leva para encontrar uma solução, o que varia com a quantidade de máquinas físicas e virtuais.

4. Resultados

Dois testes foram realizados: um experimento para a verificação do bom comportamento do mecanismo proposto em migrar as máquinas virtuais e desligar as máquinas físicas e um estudo para a verificação da escalabilidade da solução de otimização.

Para o primeiro teste que verifica o funcionamento do mecanismo proposto foram monitoradas três máquinas da plataforma FITS, a Leblon, Pão de Açúcar e Itanhangá. As máquinas Leblon e Pão de Açúcar possuem processador Intel i7 de 3.2 GHz e 16 GB de RAM, a máquina Itanhangá possui processador Intel i7 de 3.1 GHz e 8 GB de RAM. As máquinas possuem sistema operacional Debian Wheezy e executam a versão 4.1.3 do Xen. As imagens das máquinas virtuais encontram-se em um nó central do FITS, não sendo necessário copiar o disco pela rede. O mecanismo de gerência proposto é executado em uma máquina Intel Core 2 Quad de 2.4 GHz com 3 GB de RAM que é externa à plataforma FITS para não interferir no experimento. O Domínio-0 está configurado para consumir 2 GB de memória.

As máquinas virtuais possuem configurações heterogêneas de memória e processamento para avaliar a eficácia do mecanismo proposto. A máquina virtual *lpcvm1* foi configurada com 2 processadores virtuais e 2 GB de memória RAM. As máquinas virtuais *lpcvm2* e *lpcvm3* foram configuradas com 1 processador virtual e com 4 GB e 3 GB de memória respectivamente. O processamento nas máquinas virtuais foi gerado com o programa Stress [Waterland 2003], um programa que permite gerar cargas de processamento de forma controlada. Para evitar problemas de escalonamento nas migrações, foram escolhidas máquinas físicas com configurações semelhantes de processamento.

A Figura 3 mostra a migração da máquina virtual *lpcvm1* da máquina física Itanhangá para a máquina física Pão de Açúcar, Figura 3(a), Figura 3(b), Figura 3(c) e Figura 3(d). Em sequência ocorre a migração da máquina virtual *lpcvm3* que é transferida da máquina física Leblon para a Pão de Açúcar, Figura 4 e, finalmente, as máquinas Itanhangá e Leblon são desligadas. Esse resultado mostra que o mecanismo é capaz de executar o algoritmo de otimização, fazer as migrações necessárias e desligar as máquinas físicas ociosas, atingindo o objetivo de reduzir o consumo de energia elétrica. Além disso, as máquinas migradas são as que possuem a menor quantidade de memória o que reduz a carga na rede durante a migração. Após esse experimento é feito um teste de sobrecarga de recursos. Nesse teste uma quarta máquina virtual é instanciada na máquina Pão de Açúcar. A Pão de Açúcar atinge o limiar de uso de memória e o mecanismo detecta a sobrecarga. Como a oferta de recursos é menor que a demanda o Gerenciador de Energia liga uma máquina física, nesse caso, a Itanhangá. Assim que a máquina é ligada o Otimizador calcula uma nova solução e a máquina virtual *lpcvm1* é transferida para a Itanhangá.

Para o teste do módulo de otimização foram comparadas três técnicas para

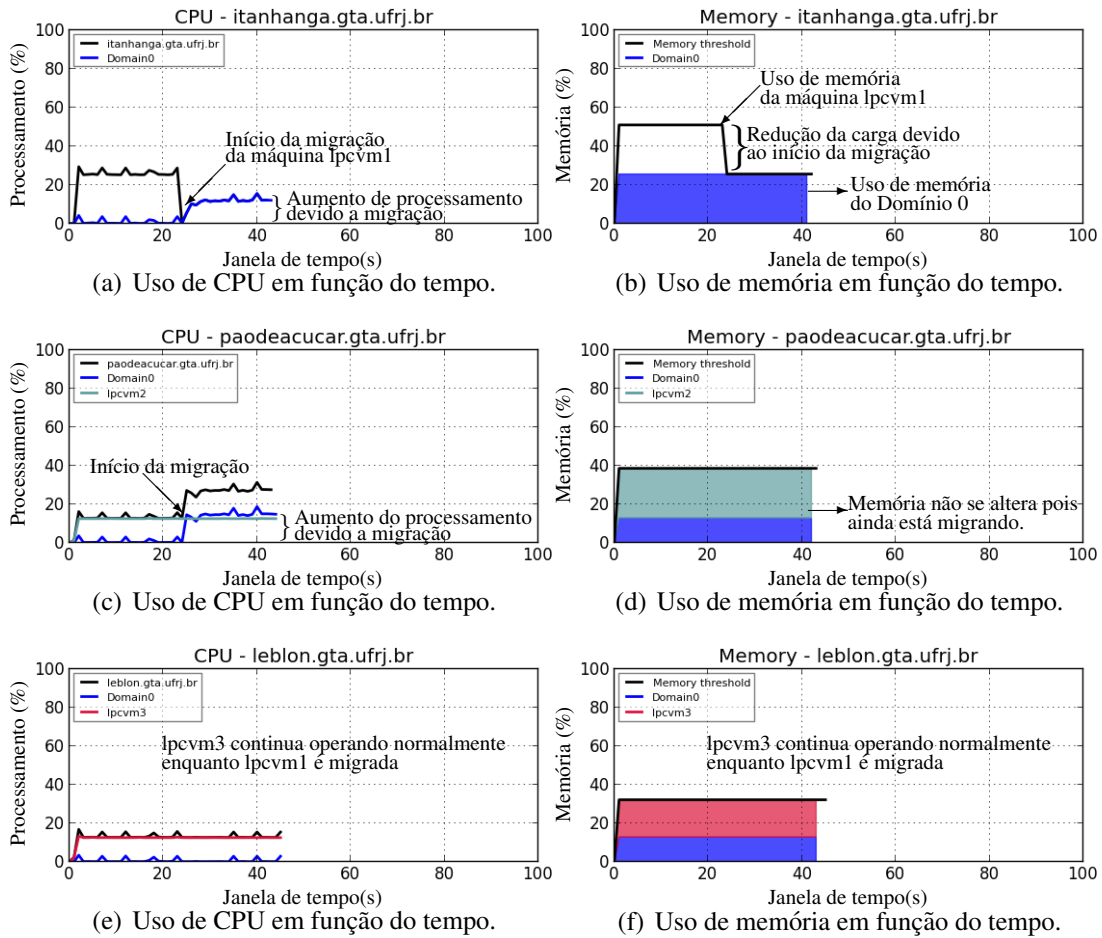


Figura 3. Início da execução do plano de ação de migração das máquinas virtuais determinado como resultado da meta-heurística de Arrefecimento Simulado. Assim que o programa inicia, a meta-heurística calcula uma solução e redistribui as máquinas virtuais. Na figura o mecanismo transfere a máquina lpcvm1 da máquina Itanhanga para a Pão de Açúcar que contém a máquina virtual lpcvm2. A figura também mostra a máquina lpcvm3 instanciada na máquina Leblon.

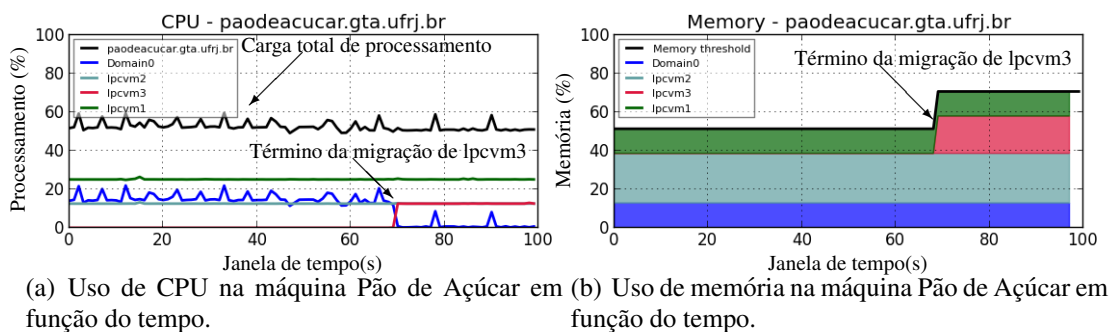
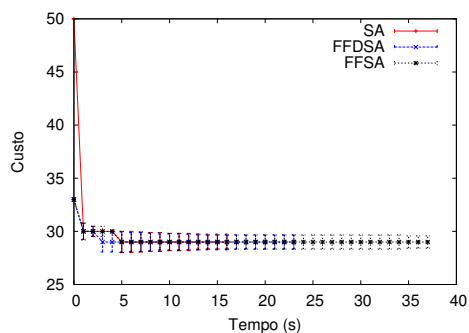


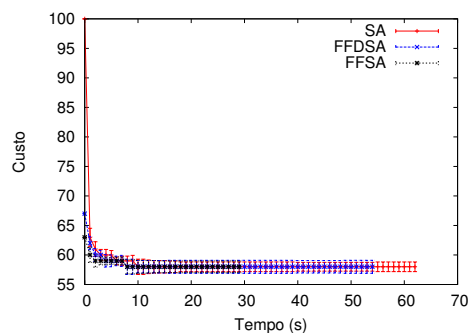
Figura 4. Após a migração da máquina lpcvm1 a máquina virtual lpcvm3 é migrada para a Pão de Açúcar e as máquinas físicas Itanhanga e Leblon são desligadas.

alocação de máquinas virtuais. A meta-heurística de Arrefecimento descrita na Seção 3.2, e as heurísticas *First Fit* (FF) e *First Fit Decreasing* (FF). A heurística *First Fit* tenta alocar as máquinas virtuais na máquina física de menor índice e caso não seja possível aloca na próxima máquina. A *First Fit Decreasing* se diferencia da FF por ordenar as máquinas virtuais de forma decrescente de uso de recursos antes de iniciar a minimização. Para esse teste foi utilizado o FFD padrão que considera as máquinas físicas com a mesma capacidade. O critério de ordenação para a técnica FFD foi estabelecido como o uso de processamento, memória e rede, nessa ordem. Foram utilizadas 50, 100, 500 e 1000 máquinas virtuais inicialmente instanciadas em 50, 100, 500 e 1000 máquinas físicas, respectivamente. Foram consideradas máquinas físicas com capacidade de 100% de processamento e 16 GB de memória.

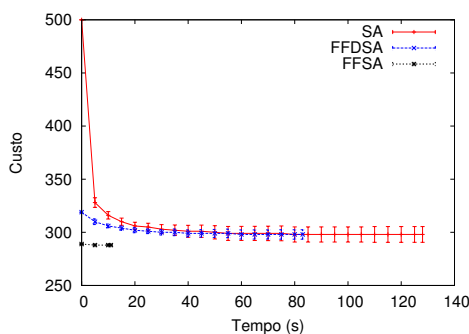
Para criar um ambiente heterogêneo de máquinas virtuais foram gerados recursos de processamento, memória e rede para cada máquina virtual seguindo uma distribuição normal. Os recursos de memória foram gerados entre 0 e 16 GB com média de 8 GB e desvio padrão de 4 GB. Os de CPU e rede entre 0 a 100% da capacidade das máquinas físicas com média e desvio padrão de 50%. Os testes consistiram de 10 rodadas limitadas a 320 segundos de execução e a temperatura do algoritmo de Arrefecimento Simulado foi inicializada em 1 milhão. Assim, o algoritmo para quando a temperatura ou o tempo chega a zero, o que ocorrer primeiro. Esses parâmetros foram configurados para simular um ambiente em que o tempo de reação do algoritmo é limitado.



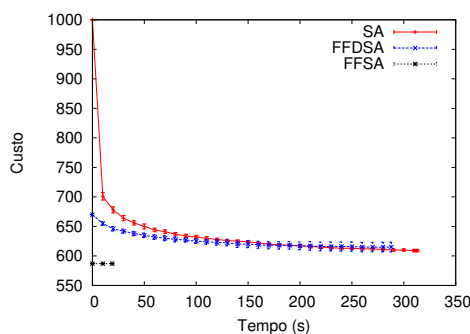
(a) Desempenho dos algoritmos para 50 máquinas virtuais.



(b) Desempenho dos algoritmos para 100 máquinas virtuais.



(c) Desempenho dos algoritmos para 500 máquinas virtuais.



(d) Desempenho dos algoritmos para 1000 máquinas virtuais.

Figura 5. Gráficos Custo x Tempo para os testes de otimização com 50, 100, 500 e 1000 máquinas físicas e virtuais.

Os gráficos da Figura 5 representam o tempo médio para o Arrefecimento Simulado encontrar soluções e melhorar as soluções das heurísticas. As figuras mostram a execução do Arrefecimento Simulado independente de heurísticas (SA), e a partir das soluções obtidas pelas heurísticas *First Fit* (FFSA) e *First Fit Decreasing* (FFDSA). O tempo de cálculo da solução inicial gerada pelo FF e pelo FFD para todos os testes foi menor que 1 segundo e foi desconsiderado do gráfico. Cada ponto representa a média e o desvio padrão para 10 rodadas. Os dados foram tomados sempre que o algoritmo encontrava uma solução de menor custo ou mesmo custo em termos de máquinas físicas. Independentemente do estado inicial, o algoritmo de Arrefecimento Simulado é capaz de reduzir a função custo. Consequentemente, o algoritmo reduz o número de máquinas físicas para cerca de 60% em todas configurações, economizando energia ao desligá-las.

5. Conclusão

Este artigo propôs um mecanismo de gerência de máquinas virtuais que minimiza o consumo de energia elétrica ao reduzir o número de máquinas físicas em funcionamento. O mecanismo utiliza técnicas de otimização baseadas em Arrefecimento Simulado para obter as soluções de menor custo, desligando as máquinas ociosas. O mecanismo também ativa máquinas físicas quando a demanda por recursos é maior que a oferta disponibilizada pelas máquinas ligadas.

Uma importante contribuição é o teste e a implementação do mecanismo na plataforma FITS, mostrando que o mecanismo funciona e é capaz de minimizar o consumo de energia em um ambiente real. Foi realizado um experimento para comprovar o funcionamento do mecanismo e um estudo do algoritmo de otimização. Os resultados mostram que o mecanismo é capaz de encontrar soluções de menor custo e migrar as máquinas virtuais para essas soluções, desligando as máquinas físicas que se tornam ociosas nesse processo. Por sua vez, o estudo dos algoritmos de otimização mostrou que o uso do Algoritmo de Arrefecimento Simulado melhora os resultados, sendo a melhor solução aquela que faz uso desse algoritmo com a heurística *First Fit*.

Futuramente pretende-se aprimorar o mecanismo através de um estudo dos principais parâmetros a serem utilizados em diferentes distribuições de máquinas virtuais e também em diferentes topologias de rede. Variáveis tais como número de saltos entre as máquinas físicas, origem e destino da migração e capacidades dos enlaces envolvidos na migração poderiam ser inseridas. Assim, seria possível determinar qual a solução de distribuição de máquinas é a menos custosa em termos de tempo de migração e distância do cliente a sua máquina virtual.

Referências

- [Bezerra et al. 2014] Bezerra, G. M. G., Mattos, D. M. F., Ferraz, L. H. G. e Duarte, O. C. M. B. (2014). Sistema automatizado de gerência de recursos para ambientes virtualizados. Em *a ser publicado em XXXII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC'2014*.
- [Carvalho e Duarte 2012] Carvalho, H. E. T. e Duarte, O. C. M. B. (2012). Voltaic: volume optimization layer to assign cloud resources. Em *Proceedings of the 3rd International Conference on Information and Communication Systems, ICICS '12*, páginas 3:1–3:7, New York, NY, USA. ACM.

- [Clark et al. 2005] Clark, C., Fraser, K., Hand, S., Hansen, J. G., Jul, E., Limpach, C., Pratt, I. e Warfield, A. (2005). Live migration of virtual machines. Em *Proceedings of the 2Nd Conference on Symposium on Networked Systems Design & Implementation - Volume 2*, NSDI'05, páginas 273–286, Berkeley, CA, USA. USENIX Association.
- [Egi et al. 2008] Egi, N., Greenhalgh, A., Handley, M., Hoerd, M., Huici, F. e Mathy, L. (2008). Towards high performance virtual routers on commodity hardware. Em *Proceedings of the 2008 ACM CoNEXT Conference*, páginas 1–12. ACM.
- [Fan et al. 2007] Fan, X., Weber, W.-D. e Barroso, L. A. (2007). Power provisioning for a warehouse-sized computer. Em *Proceedings of the 34th Annual International Symposium on Computer Architecture*, ISCA '07, páginas 13–23, New York, NY, USA. ACM.
- [Fernandes et al. 2010] Fernandes, N., Moreira, M., Moraes, I., Ferraz, L., Couto, R., Carvalho, H., Campista, M., Costa, L. e Duarte, O. (2010). Virtual networks: Isolation, performance, and trends. *Annals of Telecommunications*, 66:339–355.
- [Figueiredo et al. 2013] Figueiredo, U. d. R., Lobato, A. G. P., Mattos, D. M. F., Ferraz, L. H. G. e Duarte, O. C. M. B. (2013). Análise de desempenho de mecanismos de encaminhamento de pacotes em redes virtuais. Em *XXXI Workshop de Gerência e Operação de Redes e Serviços - (WGRS'2013) - SBRC'2013*.
- [Johnson et al. 1974] Johnson, D., Demers, A., Ullman, J., Garey, M. e Graham, R. (1974). Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM Journal on Computing*, 3(4):299–325.
- [Kundakcioglu e Alizamir 2009] Kundakcioglu, O. E. e Alizamir, S. (2009). Generalized assignment problem. Em Floudas, C. A. e Pardalos, P. M., editors, *Encyclopedia of Optimization*, páginas 1153–1162. Springer.
- [Mattos et al. 2012] Mattos, D. M. F., Mauricio, L. H., Cardoso, L. P., Alvarenga, I. D., Ferraz, L. H. G. e Duarte, O. C. M. B. (2012). Uma rede de testes interuniversitária com técnicas de virtualização híbridas. Em *XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC'2012*.
- [Mitchell 2002] Mitchell, J. E. (2002). Branch-and-cut algorithms for combinatorial optimization problems. *Handbook of Applied Optimization*, páginas 65–77.
- [Moraes et al. 2014] Moraes, I. M., Mattos, D. M., Ferraz, L. H. G., Campista, M. E. M., Rubinstein, M. G., Costa, L. H. M., de Amorim, M. D., Velloso, P. B., Duarte, O. C. M. e Pujolle, G. (2014). Fits: A flexible virtual network testbed architecture. *Computer Networks*, 63:221–237. Special Issue on Future Internet Testbeds - Part {II}.
- [Rodriguez et al. 2013] Rodriguez, E., Prado Alkmim, G., Macedo Batista, D. e Saldanha da Fonseca, N. (2013). Trade-off between bandwidth and energy consumption minimization in virtual network mapping. *Latin America Transactions, IEEE (Revista IEEE America Latina)*, 11(3):983–988.
- [Waterland 2003] Waterland, A. (2003). Stress. Disponível em: <http://people.seas.harvard.edu/~apw/stress/>. Acessado em: Março de 2014.
- [Wu et al. 2012] Wu, Y., Tang, M. e Fraser, W. (2012). A simulated annealing algorithm for energy efficient virtual machine placement. Em *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, páginas 1245–1250.