

Uma arquitetura para a Gerência de Migração de Máquinas Virtuais

Alexandre Veloso de Matos^{1,2}, Carlos A. Maziero¹

¹Departamento de Informática (DInf)
Universidade Federal do Paraná (UFPR)
Curitiba, Paraná – Brasil

²Centro de Educação do Planalto Norte
Universidade do Estado de Santa Catarina (UDESC)
São Bento do Sul, SC – Brasil

{avmatos, maziero}@inf.ufpr.br

Abstract. *The cloud infrastructure is provided by Cloud Providers (CP) enabling Cloud Services (CS) being supplied by Cloud Service Providers (CSP). In order to guarantee elasticity to CSs, Virtual Machine migrations have been adopted within the dynamic resource management. However, the workloads dynamism becomes a drawback to Cloud Management Platforms (CMP). Solutions in that scenario are challenging, concerning the diversity of service integration strategies performed by CMPs. As a consequence, migrations tend to be poorly planned, increasing the risks of failures and delays. This work proposes an architecture that allows integrating different sources of information, aiding in the automation and management of VM migrations. The proposal was compared with a solution that does not employ migration management features, in a scenario involving inter-DC communication. The evaluations show the effectiveness of the proposal, diminishing unnecessary migrations as well as the frequency of overload situations.*

Resumo. *Provedores de Infraestrutura (Cloud Provider - CP) fornecem recursos físicos para a provisão de Serviços Baseados em Nuvem (Cloud Service - CS) que são coordenados por um Provedor de CSs (Cloud Service Provider - CSP). Para garantir a elasticidade de CSs, a migração de VMs tem sido adotada no gerenciamento dinâmico de recursos. Contudo, o dinamismo que as cargas de trabalho (workload) impõe às VMs tem sido um obstáculo para as Plataformas de Gerência de Nuvens (Cloud Management Platforms - CMP). Propor soluções nesse cenário é desafiador, considerando a diversidade de abordagens de integração de serviços dos CMPs. Neste cenário, migrações tendem a ser mal planejadas, aumentando o risco de falhas e atrasos. Este trabalho propõe uma arquitetura que possibilita a integração das diferentes fontes de informação, contribuindo para a automatização e o gerenciamento das migrações de VMs. A proposta foi comparada com uma solução que não emprega recursos de gerenciamento, em um cenário de migrações entre Datacenters (DCs). A avaliação mostra a efetividade da abordagem proposta, diminuindo a quantidade de migrações desnecessárias e a frequência de sobrecargas pontuais.*

1. Introdução

Máquinas Virtuais (VMs) são a tecnologia chave para o gerenciamento de recursos em ambientes de nuvem. Uma VM encapsula um sistema de software e os recursos a ela providos, facilitando a alocação e o gerenciamento de recursos disponíveis em DCs. Tecnologias atuais de virtualização permitem a migração de uma VM entre *hosts* físicos, sob a coordenação de *hipervisores* em ambos os *hosts* [Zhang et al. 2018]. Apesar das reconhecidas contribuições, as migrações de VMs podem apresentar limitações que tornam-se obstáculos para sua aceitação, tais como: a indisponibilidade de destinos; a perda de dados; a longa duração de algumas migrações e o impacto de uma migração no trabalho de outras VMs. As soluções para tais impasses podem ser enquadradas em duas dimensões. Na primeira, têm-se algoritmos, técnicas e heurísticas que melhoram o desempenho da execução da migração e na outra, estratégias que visam tornar o planejamento de migrações mais eficiente.

O impacto que uma migração gera no consumo de recursos é designado como o custo de uma migração [Beloglazov and Buyya 2012, Dargie 2014]. Sua redução é importante para se atingir diferentes objetivos no gerenciamento de recursos, tais como: a otimização de largura de banda, a redução de migrações desnecessárias e a diminuição do gasto de energia despendido por uma migração [Moghaddam et al. 2020, Meneguette and Boukerche 2019]. O custo de uma migração pode envolver diferentes métricas, o que estimula a diversidade de algoritmos atualmente desenvolvidos [Silva Filho et al. 2018]. Uma abordagem que tem sido adotada para lidar com essa limitação é a Otimização Multi-Objetivo (*Multi-Objective Optimization - MOO*), como descrito nos trabalhos de [Xu and Fortes 2011, Beloglazov and Buyya 2012, Chen et al. 2018]. A principal crítica à adoção de MOO é o custo computacional da aplicação de tais técnicas, que são usualmente baseadas em análises estatísticas ou em algoritmos de computação evolucionária.

Migrações que não ocasionem novas necessidades de ajustes de recursos estão diretamente associadas à criação de planos de migração eficientes. Pode-se afirmar que tal eficiência é consequência de duas ações: a coordenação de diferentes fontes de informação, impedindo que os planos baseiem-se em dados ultrapassados ou que ignore dados importantes; e a diminuição da necessidade de intervenção humana.

Dentre os distintos ambientes que contribuem para o gerenciamento de nuvens, os CMPs destacam-se por três características: pela flexibilidade em atender diferentes modelos de nuvens (públicas, privadas e híbridas), por ser um intermediário (*broker*) para consumidores de recursos de infraestrutura e ainda pela integração de diferentes serviços. No entanto, a diversidade de serviços e, conseqüentemente, de interfaces e padrões, dificulta a obtenção de informações atualizadas seja no âmbito da infraestrutura como das aplicações. Em decorrência, é razoável considerar a fragilidade dos planos de migração gerados.

Este trabalho reconhece que os desafios que envolvem as migrações de VMs têm sido abordados de maneira fragmentada, como se as vantagens de uma migração na provisão dinâmica de recursos pudessem ser alcançadas pela solução de objetivos particulares. Por tal motivo, é apresentada uma arquitetura que possibilita gerenciar migrações de VMs em um ambiente de nuvem, cujo principal propósito é contribuir para que administradores de ambientes de nuvens possam arquitetar planos de migração que produzam efetivas contribuições e, portanto, dirimindo as limitações mencionadas anteriormente. Um geren-

ciador de migração implementando a arquitetura proposta foi modelado e avaliado em um cenário envolvendo migrações inter-DC. Os resultados demonstraram que uma solução de gerência de migração reduz efetivamente falhas, atrasos e migrações mal planejadas.

Este texto está organizado da seguinte forma: a Seção 2 apresenta informações gerais sobre migração de VMs, seu potencial para contribuir na provisão de recursos e as tarefas envolvidas. Algumas questões relevantes na gerência da migração de VMs também são apresentadas. Na Seção 3, alguns trabalhos relacionados são discutidos. A Seção 4 descreve nosso modelo para a gerência de migrações de VM e a arquitetura correspondente. Uma avaliação da proposta é apresentada na Seção 5, incluindo métricas aplicadas para avaliá-la e o resultados dos experimentos. Finalmente, a Seção 6 conclui este artigo.

2. A Migração de Máquinas Virtuais

A migração de VMs é considerada a abordagem *de facto* para lidar com diferentes objetivos na gerência de recursos: uso eficiente de energia, diminuição da dissipação de calor, eliminação de *hot/cold spots* (ou seja, de *hosts* sobre/sub utilizados), balanceamento de carga, consolidação de servidores e operações de manutenção [Choudhary et al. 2017]. Ao longo do tempo, como VMs são continuamente criadas e encerradas, o uso de recursos em DCs e *hosts* pode se tornar desequilibrado, alguns *hosts* estando sobrecarregados e outros subutilizados. Quando essa condição de desequilíbrio é, de fato, constatada, VMs são migradas para *hosts* adequados.

2.1. O Ciclo de Vida da Migração de uma VM

Independente do objetivo ao qual a migração esteja associada, planejar uma migração demanda três passos: a) detectar quando uma migração é necessária; b) identificar VMs que deveriam ser migradas e c) encontrar *hosts* de destino adequados a receber as VMs migrantes [Mishra et al. 2012]. No primeiro passo, detecta-se uma condição (associada a um ou mais objetivos) que denota a necessidade de migração. Nos passos seguintes são feitas duas escolhas, sucessivamente: da VM migrante e do *host* de destino adequado. Tais escolhas podem corresponder a diferentes objetivos, tais como: desempenho dos serviços, da infraestrutura ofertada pelo provedor e os relativos aos consumidores dos serviços [Jennings and Stadler 2015]. Essa diversidade de metas pode demandar a formulação de um Problema de Otimização Multi-Objetivo (*Multi-Objective Optimization Problem - MOP*) [Xu and Fortes 2011]. Outra estratégia é alternar entre diferentes objetivos conforme o estado do DC em de onde ocorre a migração [Foster et al. 2013]. Os passos do ciclo de vida servem como um guia para desenvolvimento de quaisquer soluções baseadas em migração, requerendo também que sejam consideradas: as restrições, a sobrecarga que a própria migração causa, a duração do impacto da migração nas outras aplicações afetadas e se ela, de fato, traz uma contribuição na utilização dos recursos.

2.2. Gerência da Migração de VMs

O esforço de padronização descrito por [Council 2017] destaca que as migrações de VMs estão inseridas em uma área funcional dos CMPs destinada ao gerenciamento de recursos. Em geral, CMPs possuem um console de gerenciamento de onde operadores coordenam as diferentes etapas da migração. Embora abordagens para a automação do planejamento de migrações já tenham sido apresentadas anteriormente por [Wood et al. 2007], [Xu and Fortes 2011] e [Baccarelli et al. 2015], é possível constatar que as ferramentas que

automatizam o controle de VMs em nuvens, tais como o *Cisco CloudCenter* [Cisco 2019] lidam apenas com aspectos da infraestrutura.

Pode-se considerar as seguintes funcionalidades como mínimas para a gerência de migrações: possibilitar o controle e o monitoramento de diferentes fontes de informação; permitir a construção de planos de migração e prover meios de monitoramento das migrações em curso. As fontes de informação que contribuem para o planejamento de uma migração advêm de entidades da infraestrutura tais como: hipervisores, equipamentos de interconexão de redes e do *hardware* associado aos *hosts*. E também de informações disponíveis ao nível de aplicação, tais como: a quantidade atual de VMs migrantes, última migração da qual uma VM participou, número de cargas de trabalho em execução em cada VM e os objetivos e restrições do CS associado à VM. A coleta dos dados mencionados envolve a aplicação de técnicas que permitam extrair informações atualizadas para o instante em que a avaliação do uso dos recursos é executada.

Dois escopos de migração de VMs podem ser identificados: intra-DC e inter-DC. No primeiro, uma migração pode ser reduzida à transferência da memória usada pela VM, considerando que uma unidade compartilhada de armazenamento esteja disponível e que os endereços de rede usados pela VM sejam facilmente reencaminhados. O segundo escopo engloba dois cenários: intra-nuvem e inter-nuvem. Em ambos, a migração pode ser prejudicada por questões de heterogeneidade: segmentos distintos de rede, administrações de DCs, a falta de um armazenamento compartilhado e a coordenação de diferentes hipervisores. Migrações são ainda mais desafiadoras em cenários envolvendo múltiplos DC, envolvendo várias requisições simultâneas, demandando que decisões sejam tomadas ao mesmo tempo.

Os serviços que contribuem para o planejamento de migrações usam interfaces distintas de comunicação e geram padrões de informação variáveis, exigindo dos CMPs variações constantes na estrutura usada para planejar migrações. O gerenciamento de migrações proposto neste trabalho envolve a estruturação de tal integração, estabelecendo interfaces de comunicação que possibilitam destituir do operador humano a necessidade de integrar distintas fontes de informação.

3. Trabalhos Relacionados

Trabalhos associados ao gerenciamento de migrações podem ser reduzidos a dois grupos: gerência intra-DC e inter-DC. Dentro dessa classificação destacam-se iniciativas para atender objetivos específicos de gerenciamento e para a criação de arquiteturas para o controle da migração [Cao et al. 2018]. Os trabalhos relevantes de cada classificação são discutidos nessa Seção.

3.1. Gerência da Migração Intra-DC

De nosso conhecimento, [Wood et al. 2007] apresentou a primeira abordagem que se aproxima das funcionalidades de gerência da migração de VMs, automatizando planos de migrações intra-DC. Uma arquitetura centralizada foi idealizada, considerando que os *hosts* de um DC devam informar periodicamente sobre o uso dos recursos, delegando à unidade central decidir se um *hot spot* ocorreu e ainda, possibilitando estabelecer um plano para mitigá-lo, por meio de migrações. No entanto, foi constatado que dados coletados são propensos a variações, causando um fenômeno designado *thrashing* de

migrações, ou seja, o disparo constante de migrações em decorrência de mau planejamento. Por essa razão, os perfis de uso dos recursos são coletados em janelas deslizantes de tempo, permitindo constatar se há condição de efetivo desequilíbrio. Posteriormente, VMs são classificadas conforme o desequilíbrio identificado, permitindo a escolha de VMs migrantes. Distribuições de probabilidade também são úteis na previsão do uso de recursos das VMs migrantes, assim como para definir se a solução para um *hot/cold spot* é uma migração ou apenas o redimensionamento da VM.

Em [Xu and Fortes 2011], migrações são planejadas consoante a primeira ocorrência de uma das três condições: emergência térmica, baixa eficiência energética e contenção de recursos. A cada condição de detecção é associada uma prioridade, ou seja, escolher uma VM migrante implica favorecer alguma condição. Para todos os casos, sensores fazem a coleta periódica dos dados reportando-os a um controlador central. Para a confirmação de uma necessidade de migração, dois níveis são aplicados. No primeiro nível, verifica-se a transiência, ou seja, se uma condição de migração é um alarme falso (pico momentâneo). No segundo nível, é feita uma análise de tendência que verifica se o comportamento do *workload* tende a retornar ao estado normal. Para ambos os casos existem diferentes *thresholds* e o planejamento de uma migração só inicia se ambos forem ultrapassados.

Para [Beloglazov and Buyya 2012], aumentar o tempo médio entre migrações diminui a complexidade de solucionar violações de SLA. Por isso, para a detecção de condições de migração foi proposta uma heurística em que a coleta de dados seja feita dentro de um intervalo de tempo de duração variável e com *thresholds* adaptativos. Além disso, Cadeias de Markov são usadas para que o comportamento entre migrações possa ser otimizado. Na escolha de VMs migrantes, [Beloglazov and Buyya 2012] destacam três políticas: Política do Tempo Mínimo de Migração (*Minimum Migration Time Policy* - MMT), Escolha Aleatória e Política da Máxima Correlação (*Maximum Correlation Policy* - MC). Todas são aplicadas iterativamente sempre que um *host* é considerado sobrecarregado. Após a seleção de uma VM migrante, o *host* novamente é checado e, se continuar sobrecarregado, as políticas são aplicadas novamente para a busca de outra VM. Para alcançar eficiência energética, o objetivo de migração é a consolidação. Por esse motivo, a seleção de um destino ocorre em dois passos: ordenação das VMs a migrar em ordem de decrescente de uso da CPU e identificação do *host* que provê o menor aumento no consumo de energia.

O principal propósito do trabalho de [Baccarelli et al. 2015] é a redução do consumo de energia, enquanto para [Cao et al. 2018] é a diminuição do *downtime* percebido pelo usuário. Ambos propõem métricas que ajudam a alcançar os respectivos objetivos. No trabalho de [Baccarelli et al. 2015], dois modelos suportam as decisões para medir o consumo de energia durante uma migração e para detectar o consumo de largura de banda, permitindo que um gerenciador de largura de banda otimize o tráfego. No trabalho de [Cao et al. 2018], um Modelo de Predição do Comportamento do Usuário (*User Behavior Prediction Model* - UBP) é usado para classificar VMs de *hosts* que apresentam condições de migração. O modelo UBP usa o valor de predição do uso de memória e do período da sessão de acesso durante a migração de VM para calcular o peso de cada VM.

Do trabalhos anteriores, é possível depreender que embora uma arquitetura de referência para CDCs envolva entidades que contribuem na formação de planos de migração, ainda são necessários mecanismos que contribuam no atendimento a objetivos no

gerenciamento de recursos. Muitos trabalhos visam lidar com questões específicas, tais como a eficiência energética, impedindo que soluções abrangentes sejam consideradas.

A escolha do valor para um *threshold* é uma questão pouco analisada. Trabalhos anteriores consideraram *thresholds* para o consumo seguro de recursos variando de 70% a 90% [Wood et al. 2007, Xu and Fortes 2011, Kollberg et al. 2020]. Conforme [Xu and Fortes 2011], é importante considerar que *thresholds* altos indicam um esquema de detecção mais permissivo, propenso a aceitar potenciais violações de SLAs. E que o contrário implica em um monitoramento mais agressivo, indicando avaliações da necessidade de migrar mais frequentes. A escolha dos valores, nos trabalhos anteriores, foi empírica, considerando que o valor produziria resultados satisfatórios aos propósitos de cada trabalho, o que pode ser confirmado nas respectivas avaliações. Considerando esse panorama, neste trabalho o valor de *threshold* escolhido foi um valor menos propenso à permissividade de violações de SLAs do que pode ser observado nos trabalhos relacionados: 70%.

3.2. Gerência da Migração Inter-DC

Até onde pode-se observar, *Reservoir* [Rochwerger et al. 2011] é a primeira abordagem a lidar com a gerência da integração de DCs, em um ambiente de múltiplas nuvens. Entidades denominadas *Virtual Execution Environment Host* (VEEH) são responsáveis por gerenciar *hipervisores* locais. VEEHs comunicam com um *Virtual Execution Environment Manager* (VEEM). Diferentes VEEM se interligam através de interfaces designadas *VEE Management Interface* (VMI) que podem estabelecer canais privados que possibilitam a execução de migrações.

Desafios na migração de VMs em redes de longa distância, tais como: a manutenção do endereçamento de rede, a transferência de quantidades massivas de memória e a heterogeneidade dos distintos ambientes por onde a VM migrante deve passar, diminuem o potencial de que a integração de nuvens possa atender aos objetivos do gerenciamento dinâmico de recursos. Para lidar com tais desafios, na iniciativa *Reservoir* foi proposto o *Transfer Proxy* (TP). Cada *host* implementa um ou mais TPs, cujo objetivo é abstrair os dados de VMs, através de estruturas designadas como *Transfer Token* (TT). Cada TT mantém um mapa relacionando o identificador de uma VM ao seu hospedeiro, sendo utilizado na execução das migrações. Como dados sensíveis de uma VM não estão disponíveis em um TT, cada *host* intermediário que participa de uma migração não tem conhecimento senão da origem da VM. Uma migração, ao final, envolve a transferência dos dados originais de uma VM para um TP distinto.

InterCloud [Calheiros et al. 2012] é outro *framework* projetado para apoiar na gerência de recursos em uma federação de nuvens. Não há uma entidade que centralize o planejamento de migrações, no entanto, cada nuvem possui um *Cloud Coordinator* (CC) que implementa um *Mobility Manager* que dá suporte a decisões de migração intra-nuvem, baseadas em dados de sensores que periodicamente verificam se *thresholds* não foram ultrapassados.

O trabalho de [Kollberg et al. 2020] apresenta uma abordagem para migração entre distintos DCs, baseados na proposta *ACTiCLOUD* (<https://www.acticloud.eu/>). O propósito central do trabalho é a integração de múltiplas nuvens, baseada na infraestrutura fornecida pelo Sistema Operacional de Nuvens *OpenStack*. Um componente central gerencia diferentes controladores de carga que identificam sobrecargas nos *hosts* de cada

DC. No entanto, as migrações inter-DC só podem ocorrer com VMs sem um estado associado. Ou seja, apenas a imagem da VM é transferida.

A virtualização de funções de rede (*Virtualized Network Function* - VNF) tem assumido um papel essencial, possibilitando a simplificação de necessidades decorrentes da integração de DCs. Por exemplo, a reorganização de endereçamento IP para VMs que migraram e a previsão de largura de banda são ações que têm sido facilitadas por conta de VNF, especialmente em novos modelos como Computação em Névoa (*Fog Computing*) e Computação de Borda (*Edge Computing*). Trabalhos como os de [Hong and Varghese 2019, Casellas et al. 2016] consideram que abstrair a estrutura do DC contribui na execução de operações inter-DC. No entanto, a dificuldade de se estabelecer mecanismos para o gerenciamento dos ambientes integrados é uma lacuna para a adoção definitiva de VNF.

4. Proposta de Gerência da Migração de VMs

Esta Seção discute os requisitos relevantes para um gerenciamento de migrações e posteriormente apresenta um modelo assim como a arquitetura distribuída que o implementa.

4.1. Requisitos de Gerenciamento

A fim de evitar que a detecção de condições de migração decorra de dados instáveis, toda verificação de condição deve ser baseada em uma amostra de dados coletada em intervalos periódicos de tempo. A seleção de VMs migrantes e de *hosts* de destino deve ser suportada por dados das outras VMs, assim como das migrações que estejam em curso. Ainda, para evitar que uma migração resulte em mais migrações, deve-se antecipar o comportamento dos recursos da VM migrante no destino.

No âmbito de uma nuvem, migrações podem ocorrer no escopo intra-DC ou inter-DC. O cenário intra-DC limita-se a um grupo de *hosts* coordenados por um gerente de DC. Em migrações inter-DC, um gerenciador de nuvens é responsável pela coordenação de migrações entre diferentes DCs. Mesmo quando todos DCs pertencem à mesma nuvem, cada DC pode operar em um segmento de rede diferente. Neste caso, deve-se considerar que existam diferentes políticas de segurança e gerenciamento, além de configurações técnicas distintas.

4.2. Modelo para a Gerência de Migrações

A sobrecarga de um *host* é a condição de migração considerada neste trabalho. No tocante à seleção de VMs migrantes, neste trabalho é adotada uma política inspirada na métrica *Volume-Size Ratio* ($VSR(m)$) que é a relação entre o volume $V(m)$ de uma VM m (Equação 2) e seu tamanho $S(m)$ (o uso de memória RAM por m)(Equação 1) [Wood et al. 2007]:

$$VSR(m) = \frac{V(m)}{S(m)} = \frac{V(m)}{U(m, ram)} \quad (1)$$

$$V(m) = \frac{1}{1 - U(m, cpu)} \times \frac{1}{1 - U(m, ram)} \times \frac{1}{1 - U(m, bw)} \quad (2)$$

onde *cpu*, *ram* e *bw* são respectivamente o processador, a memória e a banda de rede e $U(h, r)$ denota o uso atual do recurso r pelo *host* h .

O *VSR* de cada VM é usado para classificar o impacto de sua migração no desempenho do sistema. A VM com o maior *VSR* é a melhor candidata a migrar. Em caso de falha na migração, deve-se tentar migrar as VMs subsequentes. Filas de requisições de migração baseadas nas propostas de [Xu and Fortes 2011, Baccarelli et al. 2015], são empregadas.

A *capacidade* do recurso r no *host* h , ou seja, a quantidade de r presente em h , é dada por $C(h, r)$. Sendo $U(m, r)$ a quantidade do recurso r usado pela VM m e $M(h)$ o conjunto de VMs no *host* h , o uso atual de um recurso r no *host* h é definido como:

$$U(h, r) = \sum U(m, r) \forall m \in M(h) \quad (3)$$

Como um *host* não pode prover mais recursos do que possui, $0 \leq U(h, r) \leq C(h, r)$. A *disponibilidade* de um recurso r em h , ou seja, a quantidade do recurso r disponível para novas VMs em h , é dada por $A(h, r) = C(h, r) - U(h, r)$. Por fim, a *demand*a do recurso r pela VM m é dada por $D(m, r)$. Em situações normais, $D(m, r) = U(m, r)$. Neste trabalho, ambas a disponibilidade e a demanda estão baseadas no uso do processador. Como o consumo do processador é o recurso mais relevante nas decisões de alocação e migração, pode-se simplificar a notação e afirmar que $C(h) \triangleq C(h, \text{cpu})$, $U(h) \triangleq U(h, \text{cpu})$, e assim por diante.

Um *host* de destino h é considerado adequado a receber uma VM m se a disponibilidade atual de processador em h é maior que a demanda da VM m , ou seja $A(h) \geq D(m)$. Todos os *hosts* que satisfaçam a essa restrição são incluídos em uma fila ordenada pela sua disponibilidade do processador $A(h)$. Tentativas de migrar são feitas nesses *hosts*, sendo sempre escolhidos o primeiro disponível na fila. Como *workloads* são dinâmicos, um *host* pode facilmente tornar sobrecarregado se sua disponibilidade de recursos se tornar muito baixa. Para impedir esse efeito, definimos *thresholds* T para o uso de recursos, limitando a capacidade do *host*. A capacidade real de um recurso r em um dado *host* h deve então ser considerada como $C(h, r) \times T(h, r)$, com $0 < T \leq 1$. Ou seja, a disponibilidade de um recurso é reduzida e a restrição a ser respeitada torna-se:

$$(C(h) \times T(h) - U(h)) \geq D(m) \quad (4)$$

O perfil de uma VM e a carga resultante são coletados durante uma janela deslizante de tempo, sendo usado para prever o uso de recursos em um *host* de destino. Para evitar alarmes falsos, e evitar que migrações sejam disparadas indevidamente, [Wood et al. 2007] define que um alarme de sobrecarga somente é confirmado se for observado em duas ou mais rodadas sucessivas de coleta de dados. Essa estratégia também é adotada neste trabalho.

Recursos precisam ser reservados no *host* destino antes que uma migração ocorra. Diferentes abordagens para reserva de recursos em ambientes de nuvem têm sido propostas [Biran et al. 2018]. Neste trabalho, a abordagem descrita por [Zhang et al. 2018] é usada: antes que uma migração inicie, a mesma imagem é instanciada no destino. Enquanto a migração progride, há duas instâncias da mesma VM, uma está em redução e a outra em expansão. Portanto, no intuito de disseminar informação atualizada, a variação das capacidades precisa estar disponível para todas as entidades no início de cada janela deslizante de tempo.

4.3. Arquitetura Proposta

A arquitetura de gerência é uma abordagem descentralizada e distribuída, baseada nas seguintes entidades: Gerenciador da Nuvem (*Cloud Manager - CM*), Gerenciador do DC (*DC Manager - DCM*) e Controlador do Host (*Host Controllers - HC*) (Figura 1). As migrações são planejadas e controladas em ambos os domínios, intra-DC e inter-DC.

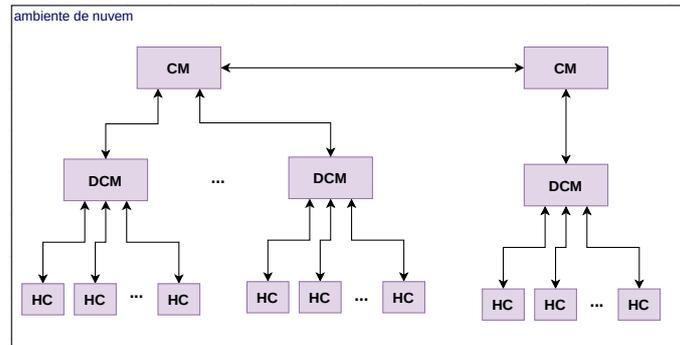


Figura 1. Entidades da arquitetura de gerência de migrações.

Cada *host* possui um HC que executa ações em dois níveis: infraestrutura (virtualização) e estratégico (migração). O mapeamento e a reserva de recursos são ações executadas no nível de infraestrutura. Como descrito na Figura 2(b), um HC monitora o estado de cada recurso (físico e virtual), interage com o hipervisor local e controla as VMs. Condições que possibilitam o disparo de migrações são inferidas no nível estratégico.

Dois tráfegos de controle podem ocorrer entre DCM e HCs: de recursos e de migrações. O HC pode investigar constantemente se cada recurso r mantém-se sob um *threshold* pré-definido $T(h, r)$. Qualquer inconsistência é comunicada ao DCM, que avalia se o evento demanda uma migração. No intuito de impedir migrações desnecessárias, o DCM inspeciona se o evento é transiente e também determina em que ordem as migrações devem ser executadas.

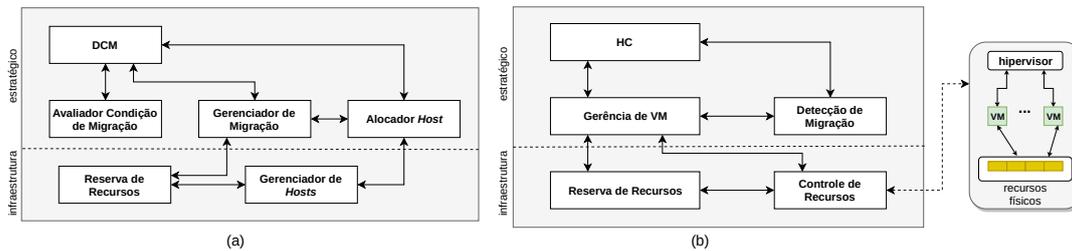


Figura 2. (a) O Gerenciador de DC (DCM) e (b) o Controlador de Host (HC).

O planejamento de uma migração requer duas seleções do DCM: de VMs a migrar e de destinos adequados. O *Avaliador de Condições de Migração* avalia o potencial da migração da VM em duas etapas: primeiro se há disponibilidade, para tanto verifica se $A(h) > D(m)$, posteriormente calcula o uso de recursos em h após a migração de m e inclui essa medida em uma lista. Migrações são disparadas pelo *Gerenciador de Migração* que requer à *Reserva de Recursos* a reserva dos recursos requeridos por m (Figura 2(a)). Cada migração que não pode ser solucionada no âmbito do DC é enviada ao CM, que

mantém informações reportadas periodicamente pelos DCMs. Quaisquer decisões do CM requerem que ambos CM e DCM mantenham o status das migrações em curso. Enquanto o CM está planejando uma migração, o DCM não envia sinalizações de outras migrações não solucionadas.

5. Avaliação

A arquitetura proposta foi avaliada por meio de simulação, usando o simulador *CloudSim* (<http://www.cloudbus.org/cloudsim/>). O simulador foi estendido para que migrações intra-DC e inter-DC fossem possíveis e também para lidar com as ações de gerenciamento.

Foram consideradas migrações inter-DC em ambientes de nuvem já estabelecidos, portanto, cujos CPs e respectivos CMs já participem de acordos previamente estabelecidos.

5.1. Pressupostos Básicos

Dois aspectos são relevantes no modelo aplicado aos experimentos: as características das entidades envolvidas e o que e como está sendo avaliado. Duas entidades são fundamentais para qualquer decisão durante uma migração: VMs e *workloads*. As decisões de migrações são baseadas nos estados de ambas, portanto, dependendo do monitoramento de ambas.

A forma como os recursos são alocados a VMs e seus *workloads* determina o comportamento de realocações. As VMs e os *workloads* podem adotar uma das seguintes políticas de alocação de recursos: *time-shared* e *space-shared*. Na primeira, cada processador pode ser usado por diferentes VMs; no entanto, cada VM pode usar do processador por um *quantum* fixo de tempo, liberando-o com a ocorrência de preempção. Na segunda política, cada VM detém um conjunto de processadores específicos, apenas liberando-os após o *workload* encerrar. Monitorar *workloads* pode se tornar uma tarefa complexa se a política de alocação *time-shared* for considerada, pois a associação entre processadores e os recursos virtuais varia frequentemente, dificultando decisões sobre violações de *thresholds*. Baseando-se nos resultados de [Tani and El Amrani 2016], neste trabalho é considerada apenas a política de alocação *space-shared*.

No intuito de estimular *workloads* com comportamento variável durante a simulação, foi considerado que o uso do processador por um *workload* varia ao longo do tempo. Dado um *workload* w , seu uso de processador no instante t é denotado por $U_t(w, cpu)$. Periodicamente, esse uso varia conforme um fator aleatório no intervalo $\pm 10\%$. No ambiente de simulação, o período escolhido para variação das iterações foi $\tau = 50 ms$ (intervalo de tempo padrão definido no simulador CloudSim). Portanto:

$$U_{t+\tau}(w, cpu) = U_t(w, cpu) \times \text{random}(0.9, 1.1) \quad (5)$$

Considerando a política de alocação *space-shared*, o conjunto de *workloads* associados a uma VM m é definido como $W(m)$. Assim, o uso total do processador por uma VM m em um dado tempo t é definido como a soma de seus *workloads*:

$$U_t(m, cpu) = \sum U_t(w, cpu) \forall w \in W(m) \quad (6)$$

Como discutido na Seção 4.2, as cargas U de *hosts* e de VMs são medidas a cada janela de tempo. Para obter valores mais estáveis, é usada a média de dois valores

consecutivos: $U_m = (U_{t-\tau} + U_t)/2$. Uma migração é disparada apenas se o valor médio U_m viola o *threshold*.

Eventos que podem denotar uma necessidade de migração são identificados, demandando uma decisão rápida. Se uma migração precisar se disparar, duas ações são necessárias: calcular quanto tempo será usado pela migração e notificar o DCM ou o CM sobre as características da migração. O tempo $tm(m)$ necessário para migrar uma VM m pode ser estimado como a relação entre o uso atual de memória $U(m, ram)$ e a largura de banda da rede $A(h, bw)$ disponíveis no *host* h que contém m : $tm(m) = U(m, ram)/A(h, bw)$.

Quando uma migração é efetivamente planejada, um mapa de destinos potenciais é definido. Para iniciar uma migração, o mesmo estado da VM é reservado no destino, sendo mantido estático até que a migração encerre. No ambiente de simulação, a capacidade é então diminuída na fonte em uma porcentagem baseada na largura de banda disponível para receber o tráfego no destino e também pela quantidade de dados transferida.

5.2. Experimentos

Dois cenários são considerados: gerenciado e não-gerenciado. É esperado que a adoção de uma solução gerenciada traga vantagens em comparação com uma não-gerenciada. Embora ambos os cenários adotem a mesma arquitetura apresentada na Seção 4, o cenário gerenciado é baseado em ações não previstas na solução não-gerenciada: a média do uso dos recursos que evita considerar picos espúrios como violações de *thresholds*, a reserva de recursos no lado destino durante a migração e políticas que definem a seleção de VMs candidatas a migrar e de *hosts* de destino (no cenário não-gerenciado é adotada uma política FCFS para tais escolhas).

Os experimentos consistem de um conjunto de simulações que usam os mesmos parâmetros, mas com valores distintos. As simulações, cujos parâmetros são apresentados na Tabela 1, avaliam o comportamento de ambas as abordagens, conforme a variação na oferta e na demanda de recursos. Cada simulação foi repetida 10 vezes e seus resultados médios são apresentados. A execução de cada simulação se estendeu durante 1000 unidades de tempo do ambiente *CloudSim*. Para confirmar que 10 execuções são suficientes, em ambas as abordagens e para cada simulação foi calculado o Coeficiente de Variação ($CV = SD/M$, onde SD = desvio padrão dos resultados da simulação e M , a média). Consideramos que $CV < 0,6$ indica estabilidade no resultado.

Tabela 1. Parâmetros nas Simulações

Simulação	1	2	3	4	5	6	7	8	9	10
Datacenters	2	2	4	4	8	8	16	16	32	32
Hosts/DC	4	8	4	8	4	8	4	8	4	8
Núcleos/Host	8	8	8	8	8	8	8	8	8	8
Capacidade por núcleo (mips)	4000	4000	4000	4000	4000	4000	4000	4000	4000	4000
VMs	24	24	24	24	24	24	24	24	24	24
Workloads	24	24	24	24	24	24	24	24	24	24
Brokers	4	8	4	8	4	8	4	8	4	8
Núcleos/VM	8	8	8	8	8	8	8	8	8	8
Demanda de CPU por VM (mips)	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000

A Figura 3 exibe a quantidade de migrações em cada simulação, para ambos os cenários. Pode-se observar que, em todas as simulações, o cenário gerenciado apresenta menos migrações. Isso demonstra que o gerenciamento da migração pode reduzir, de

maneira efetiva, a quantidade de migrações de VMs. Nesse experimento, os valores de CV variaram de 0 a 0,56, demonstrando um resultado confiável.

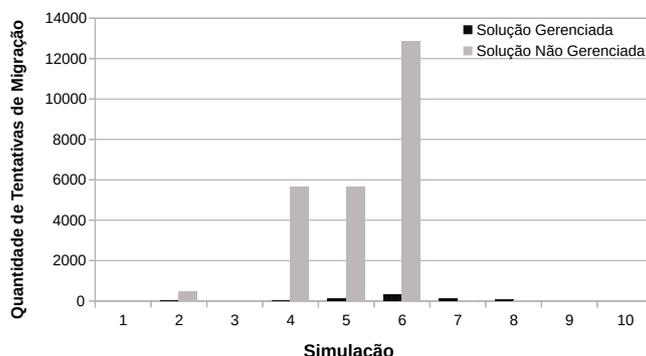
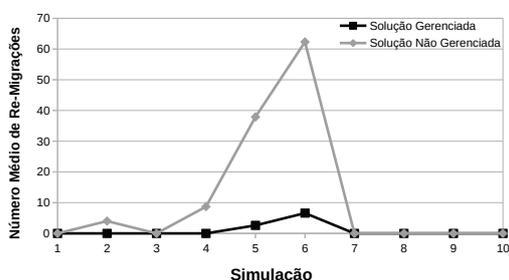
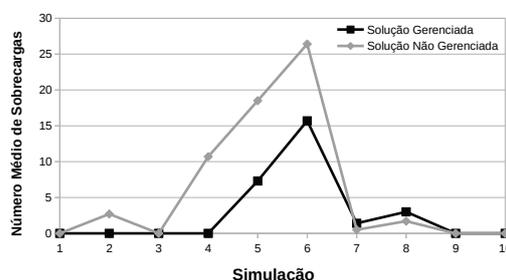


Figura 3. Quantidade de migrações

A Figura 4(a) apresenta as “re-migrações”, ou seja, o número médio de VMs que migraram mais de uma vez em cada simulação. O resultado demonstra que gerenciar migrações traz mais estabilidade para o sistema, pois as VMs tendem a migrar menos. A Figura 4(b) apresenta a quantidade de *hosts* que se tornam sobrecarregados em algum momento. O cenário não-gerenciado apresenta mais sobrecargas em todas as simulações, indicando que planos de migração são mais efetivos para solucionar sobrecargas. Nesse experimento, os valores de CV variaram de 0 a 0,48, demonstrando um resultado confiável.



(a) VMs que migraram mais de uma vez



(b) Quantidade de sobrecarga de *hosts*

Figura 4. Impacto do não-gerenciamento de migrações

6. Conclusão

Soluções atuais de gerenciamento da migração de VMs apresentam problemas como a dificuldade de integrar dados que provêm de distintos níveis de gerenciamento e o monitoramento de várias VMs ao longo dos distintos passos do ciclo de vida de migração. Este trabalho lida com tais problemas, propondo um *framework* distribuído para gerenciar migrações de VM. O modelo de gerência e a implementação de sua arquitetura foram avaliados por meio de experimentos de simulação. Os resultados mostram que o gerenciamento de migrações de VMs conduz a melhores resultados em comparação a uma solução não-gerenciada. Ou seja, ocorrem menos sobrecargas e menos migrações.

Como direções futuras, pretende-se aprofundar a avaliação de estruturas para a gerência inter-nuvem de migrações e propor uma estratégia para gerenciar a capacidade

ociosa em decorrência do encerramento de *workloads*. No tocante à seleção de *hosts* de destino, as políticas adotadas neste trabalho estão principalmente focadas na minimização do tempo de transferência da VM e no balanceamento de carga entre os hosts. Outras políticas podem ser consideradas, tais como a maximização da quantidade de *hosts* ociosos, colocando-os em *standby* para reduzir o consumo de energia. Técnicas de Otimização Multi-Objetivo podem ser empregadas para combinar tais políticas.

Referências

- Baccarelli, E., Amendola, D., and Cordeschi, N. (2015). Minimum-energy bandwidth management for QoS live migration of virtual machines. *Computer Networks*, 93:1–22.
- Beloglazov, A. and Buyya, R. (2012). Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. *Concurrency and Computation: Practice and Experience*, 24(13).
- Biran, O., Breitgand, D., Lorenz, D., Masin, M., Raichstein, E., Weit, A., and Iyoob, I. (2018). Heterogeneous resource reservation. In *IEEE Intl Conference on Cloud Engineering (IC2E)*, pages 141–147. IEEE.
- Calheiros, R. N., Toosi, A. N., Vecchiola, C., and Buyya, R. (2012). A coordinator for scaling elastic applications across multiple clouds. *Future Generation Computer Systems*, 28(8):1350–1362.
- Cao, R., Tang, Z., Li, K., and Li, K. (2018). HMGOWM: a hybrid decision mechanism for automating migration of virtual machines. *IEEE Transactions on Services Computing*.
- Casellas, R., Muñoz, R., Vilalta, R., and Martínez, R. (2016). Orchestration of IT/cloud and networks: From inter-DC interconnection to SDN/NFV 5G services. In *2016 Intl Conference on Optical Network Design and Modeling (ONDM)*, pages 1–6. IEEE.
- Chen, T., Zhu, Y., Gao, X., Kong, L., Chen, G., and Wang, Y. (2018). Improving resource utilization via virtual machine placement in data center networks. *Mobile Networks and Applications*, 23(2):227–238.
- Choudhary, A., Govil, M. C., Singh, G., Awasthi, L. K., Pilli, E. S., and Kapil, D. (2017). A critical survey of live virtual machine migration techniques. *Journal of Cloud Computing*, 6(1):23.
- Cisco (2019). Cisco CloudCenter solution: Architecture overview - white paper. https://www.cisco.com/c/dam/global/es_es/pdfs/Cisco-cloudcenter-architecture-wp-c11-737224.pdf, acessado em 28-03-2020.
- Council, C. S. (2017). Practical guide to cloud management platforms. <https://www.omg.org/cloud/deliverables/CSCC-Practical-Guide-to-Cloud-Management-Platforms.pdf>, acessado em 28-03-2020.
- Dargie, W. (2014). Estimation of the cost of VM migration. In *2014 23rd Intl Conference on Computer Communication and Networks (ICCCN)*, pages 1–8. IEEE.
- Foster, G., Keller, G., Tighe, M., Lutfiyya, H., and Bauer, M. (2013). The right tool for the job: Switching data centre management strategies at runtime. In *2013 IFIP/IEEE Intl Symposium on Integrated Network Management (IM 2013)*, pages 151–159. IEEE.

- Hong, C.-H. and Varghese, B. (2019). Resource management in fog/edge computing: a survey on architectures, infrastructure, and algorithms. *ACM Computing Surveys (CSUR)*, 52(5):1–37.
- Jennings, B. and Stadler, R. (2015). Resource management in clouds: Survey and research challenges. *Journal of Network and Systems Management*, 23(3):567–619.
- Kollberg, S., Lakew, E. B., Svärd, P., Elmroth, E., and Tordsson, J. (2020). Spreading the heat: Multi-cloud controller for failover and cross-site offloading. In *34th Intl Conf on Advanced Information Networking and Applications (AINA-2020)*, pages 1154–1164.
- Meneguette, R. and Boukerche, A. (2019). An efficient green-aware architecture for virtual machine migration in sustainable vehicular clouds. *IEEE Transactions on Sustainable Computing*.
- Mishra, M., Das, A., Kulkarni, P., and Sahoo, A. (2012). Dynamic resource management using virtual machine migrations. *IEEE Communications Magazine*, 50(9):34–40.
- Moghaddam, M. J., Esmailzadeh, A., Ghavipour, M., and Zadeh, A. K. (2020). Minimizing virtual machine migration probability in cloud computing environments. *Cluster Computing*, pages 1–10.
- Rochwerger, B., Breitgand, D., Epstein, A., Hadas, D., Loy, I., Nagin, K., Tordsson, J., Ragusa, C., Villari, M., Clayman, S., et al. (2011). Reservoir – when one cloud is not enough. *Computer*, 44(3):44–51.
- Silva Filho, M. C., Monteiro, C. C., Inacio, P. R., and Freire, M. M. (2018). Approaches for optimizing virtual machine placement and migration in cloud environments: A survey. *Journal of Parallel and Distributed Computing*, 111:222–250.
- Tani, H. G. and El Amrani, C. (2016). Cloud computing CPU allocation and scheduling algorithms using CloudSim simulator. *Intl Journal of Electrical and Computer Engineering*, 6(4):1866.
- Wood, T., Shenoy, P. J., Venkataramani, A., Yousif, M. S., et al. (2007). Black-box and gray-box strategies for virtual machine migration. In *NSDI*, volume 7, pages 17–17.
- Xu, J. and Fortes, J. (2011). A multi-objective approach to virtual machine management in datacenters. In *ACM Intl Conference on Autonomic Computing*, pages 225–234. ACM.
- Zhang, F., Liu, G., Fu, X., and Yahyapour, R. (2018). A survey on virtual machine migration: Challenges, techniques, and open issues. *IEEE Communications Surveys & Tutorials*, 20(2).