

Uma Abordagem Baseada no Consumo de CPU e RAM para a Eficiência Energética em Centros de Dados para Computação em Nuvem

Pedro H. P. Castro, Sand Corrêa, Kleber V. Cardoso
 Instituto de Informática (INF)
 Universidade Federal de Goiás (UFG)
 Goiânia, Brasil
 {pedrocastro, sand, kleber}@inf.ufg.br

Resumo—A computação em nuvem tem levado os sistemas distribuídos a um novo patamar, oferecendo recursos computacionais de forma virtualizada, flexível, robusta e escalável. Essas vantagens, no entanto, surgem juntamente com um alto consumo de energia nos centros de dados, ambientes que podem ter até centenas de milhares de servidores. Existem muitas propostas para alcançar eficiência energética em centros de dados voltados para computação em nuvem. Entretanto, muitas propostas consideram apenas o consumo proveniente do uso de CPU, além de empregarem definições de SLA dependentes do domínio da aplicação. Neste trabalho, propomos duas novas abordagens para melhorar a eficiência energética desses ambientes. As abordagens se baseiam no consumo proveniente do uso de CPU e de memória conjuntamente. Implementamos e validamos ambas as propostas no simulador CloudSim e comparamos os resultados com outras soluções consideradas estado da arte. Nossas propostas reduzem o consumo de energia em até 33% quando comparadas com outras abordagens. Elas também diminuem a violação de SLA em até 90%, mesmo quando este é definido de forma genérica.

Index Terms—computação em nuvem; centro de dados; eficiência energética; cumprimento de SLA; consolidação de máquinas virtuais.

I. INTRODUÇÃO

A computação em nuvem está mudando a forma como os recursos computacionais são tradicionalmente provisionados e alocados, estabelecendo um modelo onde a própria infraestrutura de computação é fornecida como um serviço a seus usuários. Atualmente, através de uma nuvem, é possível ter acesso remoto a uma coleção de recursos virtualizados que inclui processamento, memória, armazenamento e comunicação de dados [1]. Esses recursos são provisionados dinamicamente, obedecendo acordos de níveis de serviços (*Service Level Agreements* – SLAs) estabelecidos entre os provedores e os consumidores da infraestrutura [2]. Esse modelo de computação tem se mostrado adequado para atender diversos tipos de necessidades, desde um usuário doméstico que apenas precisa de um pequeno disco virtual até um usuário corporativo que terceiriza processamento e armazenamento em larga escala.

Por um lado, a computação em nuvem tem levado o conceito de sistemas distribuídos a um novo patamar, oferecendo recursos computacionais com flexibilidade, robustez e escalabilidade sem precedentes. Por outro lado, a demanda energética da computação em nuvem exige atenção. A base dos sistemas de computação em nuvem são os centros de dados (*data centers*) que podem ter até centenas de milhares

de servidores em uma única infraestrutura. Além dos equipamentos de computação, como servidores, *storages* e *switches*, os centros de dados contam também com enormes sistemas de refrigeração que podem chegar a consumir metade de toda a energia da infraestrutura [3]. Nesse contexto, a eficiência energética no uso dos recursos computacionais pode implicar em uma sensível redução dos custos de operação de um centro de dados. A redução no consumo de energia elétrica também tem impacto ambiental positivo, o qual é relevante na escala que envolve os centros de dados. De acordo com um relatório do Greenpeace [4], em 2011, os centros de dados consumiam 2% da energia elétrica global e o consumo crescia a uma taxa de 12% ao ano.

O problema energético nos centros de dados pode ser resumido em dois fatores principais: 1) a quantidade de recursos computacionais ligados e a ineficiência energética de seus componentes físicos (*hardware*); e 2) o uso ineficiente dos recursos computacionais (*software*). Estudos mostram que mesmo ociosos, os servidores atuais consomem cerca de 70% da energia que é necessária durante um período de pico de trabalho [5]. Há também trabalhos que revelam que os servidores de centros de dados operam abaixo de 50% de sua capacidade total [6], consumindo energia e gerando calor desnecessariamente. A ineficiência energética de componentes físicos tem sido tratada por soluções em *hardware* e *firmware*, usando técnicas como desativação de componentes em tempo de execução [7], [8] e escalonamento de desempenho proporcional à energia consumida [9], [10]. A subutilização dos recursos computacionais tem sido tratada no contexto de virtualização [11], [12], [13], [14]. Basicamente, essa abordagem consiste em criar máquinas virtuais (*Virtual Machines* ou VMs) que atendam a demanda efetiva do centro de dados e consolidá-las em um número mínimo de servidores físicos, sem, no entanto, violar o SLA acordado com os clientes. Os servidores ociosos são então desligados ou colocados em estados de dormência, nos quais consomem pouca energia. Como a demanda por recursos em infraestruturas de computação em nuvem é elástica [2], é importante que o processo de consolidação de VMs seja dinâmico com o intuito de atender as necessidades correntes.

Consolidação de VMs com foco em eficiência energética é, portanto, um problema de alocação de recursos computacionais em que se deseja minimizar o consumo de energia e, ao mesmo tempo, atender os requisitos de qualidade de serviço

definidos em termos de SLA. A abordagem tradicionalmente utilizada consiste em minimizar o consumo de energia de um único recurso: a CPU. No entanto, a proliferação de arquiteturas com múltiplos núcleos (*multi-core*) e a própria virtualização têm exigido servidores equipados com grandes quantidades de memória (RAM¹), tornando representativo o consumo de energia por parte desse componente. Em 2009, [15] já constatava que o consumo de energia proveniente do uso de RAM correspondia a até 25% do consumo total de um servidor.

Neste trabalho, apresentamos duas abordagens para aperfeiçoar a consolidação de VMs com foco em eficiência energética e manutenção de SLA de centros de dados voltados para computação em nuvem. Inicialmente, propomos uma heurística chamada CREW (*CPU and RAM Energy aWare*), a qual utiliza um modelo de energia que considera conjuntamente o consumo proveniente do uso de CPU e de RAM. Apresentamos também um novo algoritmo para detecção de nós subutilizados, chamado CREW-UD (*CREW with improved Underload Detection*), que se baseia no histórico de utilização de CPU para estabelecer um limiar mínimo para desligamento ou adormecimento de servidores.

Este trabalho está organizado da seguinte forma. A Seção II discute importantes trabalhos propostos com foco em eficiência energética em centro de dados. A Seção III apresenta o modelo de sistema considerado neste trabalho. A Seção IV apresenta uma descrição detalhada da nossa proposta para consolidação de VMs. A avaliação da proposta e os resultados das simulações realizadas são o foco da Seção V. Por fim, a Seção VI apresenta as conclusões e os trabalhos futuros.

II. TRABALHOS RELACIONADOS

Nos últimos anos, diversas abordagens têm sido usadas para formular o problema de gerenciamento de energia em centros de dados que provêm recursos virtualizados. Em [16], políticas locais e globais, são usadas de forma coordenada para alcançar esse objetivo. Enquanto políticas locais – por exemplo, políticas aplicadas no nível do sistema operacional – controlam o consumo de energia em cada máquina física, políticas globais utilizam o conhecimento sobre características da infraestrutura para consolidação de VMs. No entanto, os autores não detalham nenhuma política global específica.

O aprovisionamento dinâmico e ciente de energia de VMs em centros de dados também tem sido tratado como um problema de otimização. Em [11], é proposto um modelo para ambientes em que clientes pagam por serviços providos pela infraestrutura e são ressarcidos em caso de violação de SLA. O objetivo do modelo é maximizar os ganhos do provedor de serviços, minimizando o consumo energético e o número de violações. Para tratar esse problema, os autores utilizam teoria de controle. No entanto, o modelo proposto requer uma fase de treinamento que é dependente de aplicação, o que dificulta sua implantação em ambientes de computação em nuvem,

¹Neste trabalho, usaremos RAM como sinônimo de memória principal do servidor, independente da tecnologia utilizada.

visto que a carga de trabalho nestes ambientes é variada. Em [12], a consolidação dinâmica de VMs é tratada usando uma heurística para o problema *bin packing*. No modelo proposto, os contêineres ou *bins* representam servidores físicos cujas capacidades são medidas pela disponibilidade de CPU. VMs são os itens a serem alocados nos contêineres e o custo de cada contêiner é representado pelo consumo de energia do servidor. O algoritmo proposto, no entanto, não suporta definição de SLA e a consolidação de VMs pode degradar o desempenho das aplicações. Em [11] e [12], o modelo de energia proposto considera apenas o consumo de CPU.

Uma abordagem diferente, baseada em classificação de tarefas, é apresentada em [13]. Os autores propõem um arcabouço para caracterizar a carga de trabalho em centro de dados e a informação capturada em tempo de execução é usada para controlar o número de VMs a serem alocadas, a fim de obter um equilíbrio entre consumo de energia e qualidade de serviço. Os autores utilizam o algoritmo *K-means* para caracterizar um conjunto de carga de trabalho em classes de tarefas, enquanto a composição da carga para cada classe é capturada em termos da taxa de chegada. No entanto, capturar adequadamente a heterogeneidade das tarefas que são executadas em infraestruturas de computação em nuvem de grande escala exige um grande esforço computacional e de monitoramento. Além disso, como os trabalhos citados anteriormente, o trabalho proposto em [13] busca otimizar apenas o consumo de energia de CPU.

Recentemente, os autores de [14] propuseram um arcabouço para o problema de consolidação de VMs com foco em eficiência energética. Como em [12], o arcabouço proposto avalia diferentes heurísticas para o problema *bin packing* com tamanho e custo variados de contêineres. No entanto, diferentemente de [12], a abordagem proposta em [14] suporta a definição de SLA, sendo essa expressa de forma independente de aplicação. As heurísticas propostas também podem tratar heterogeneidade da infraestrutura e das tarefas, sem, no entanto, assumir qualquer conhecimento prévio das aplicações que executam nas VMs. No entanto, como os trabalhos anteriores, o modelo de energia proposto busca minimizar apenas o consumo de CPU.

Neste trabalho, propomos adaptar o arcabouço proposto em [14] para considerar também o consumo de energia proveniente do uso de RAM. Para quantificar esse consumo, usamos um modelo analítico proposto em [17]. Esse último trabalho é focado na *hardware* e, portanto, não leva em consideração o problema de consolidação de VMs em centros de dados. Adicionalmente, nosso trabalho aperfeiçoa a detecção de nós subutilizados, um componente importante do arcabouço proposto em [14].

III. MODELO DE SISTEMA

Neste trabalho, consideramos um ambiente de computação em nuvem do tipo IaaS (*Infrastructure as a Service*) composto por centenas ou milhares de servidores físicos. Os servidores podem ser heterogêneos em termos de arquitetura, velocidade de processamento, propriedades do *hardware* e consumo de

energia. Nenhum servidor possui disco local e o serviço de armazenamento é provido por um equipamento especializado em armazenamento (*storage*) que está conectado aos servidores através de uma rede de alta velocidade. Em um dado instante de tempo, múltiplas VMs coexistem no sistema de computação em nuvem. Essas VMs podem possuir requisitos de qualidade de serviços heterogêneos, ou seja, podem possuir diferentes demandas por capacidade de processamento, quantidade de memória e largura de banda. Vários tipos de aplicações podem ser executados pelos diferentes usuários do ambiente e não existe nenhum conhecimento prévio da carga de trabalho à qual o sistema é submetido. A qualidade de serviço acordada entre o provedor da infraestrutura e os usuários é definida na forma de SLAs (definidos, por exemplo, em termos de vazão mínima ou tempo máximo de resposta).

Assim como em [16], assumimos uma camada de *software* composta por um gerente global e múltiplos gerentes locais. Cada gerente local reside em um servidor físico e é responsável por monitorar o uso de CPU e RAM nesse servidor. É também responsabilidade do gerente local decidir quando e quais VMs devem ser migradas para outro servidor, além de mudar o modo de energia do servidor, ou seja, desligá-lo ou adormecê-lo. Periodicamente, o gerente global coleta informações dos gerentes locais para decidir como as VMs devem ser realocadas de forma a minimizar o consumo total de energia e também as violações de SLA. A realocação de VMs ocorre em tempo de execução através de migração das VMs entre os servidores disponíveis. Na seção seguinte, descrevemos o modelo de energia usado neste trabalho. Posteriormente, apresentamos também métricas para calcular o custo de migração e quantificar as violações de SLA.

A. Modelo de energia

Neste trabalho, assumimos que o consumo total de energia de um servidor, E_{TOTAL} , é determinado pela energia consumida pelo uso de CPU, E_{CPU} , e de RAM, E_{RAM} :

$$E_{TOTAL} = E_{CPU} + E_{RAM}. \quad (1)$$

Conforme proposto em [14], o consumo de energia proveniente do uso de CPU pode ser obtido a partir de dados reais fornecidos pelo SPECpower [18]. A Tabela I ilustra o consumo energético da CPU em função da porcentagem de utilização para os dois servidores utilizados neste trabalho: HP ProLiant ML110 G4 (Intel Xeon 3040, 2 núcleos de 1860 MHz, 32 GB) e HP ProLiant ML110 G5 (Intel Xeon 3075, 2 núcleos de 2660 MHz, 32 GB).

O consumo de energia proveniente do uso de RAM é calculado usando o modelo analítico proposto em [17]. Nesse modelo, a energia consumida pela RAM é dividida em duas partes: 1) consumo de fundo (*background*), o qual não depende da quantidade e nem do tipo dos comandos executados, e 2) consumo de operação, proveniente da execução de um comando de leitura ou escrita quando o dispositivo de memória está ativo.

O consumo de fundo depende apenas do estado e da frequência de operação da memória. Em [17], são descritos

sete estados em que a memória pode estar: 1) de prontidão (*standby*) ativo, 2) de prontidão com pré-carga, 3) desligado (*powerdown*) ativo, 4) desligado com pré-carga rápida, 5) desligado com pré-carga lenta, 6) auto-atualizado (*self-refresh*) e 7) auto-atualizado com registradores desligados. Neste trabalho, consideramos que a memória sempre está no estado de prontidão ativo. Embora esse estado seja o que consome mais energia, ele possui a vantagem de receber comandos imediatamente, sem qualquer penalidade de latência. Essa escolha também é coerente com a estratégia fundamental do nosso trabalho que consiste em manter a maior quantidade de VMs por servidor dadas as restrições de SLA.

O consumo de operação é o produto da largura de banda consumida e a energia necessária para realizar uma operação específica por unidade de largura de banda. Assim, o consumo de energia da RAM é descrito pela seguinte expressão:

$$E_{RAM} = E_F + (E_{LB,l} \times LB_l + E_{LB,e} \times LB_e), \quad (2)$$

onde E_F representa o consumo de fundo; $E_{LB,l}$ e $E_{LB,e}$ são respectivamente a energia requerida para leitura e escrita por GB/s; e LB_l e LB_e indicam a largura de banda de leitura e escrita, respectivamente. Utilizamos os mesmos valores descritos em [17] para os parâmetros da Equação 2. Assim, para cada 4 GB de DRAM do tipo DDR3 DIMM, temos: $E_F = 5,36W$, $E_{LB,l} = 0,939W/(GB/s)$, $E_{LB,e} = 1,023W/(GB/s)$. Dada a pequena diferença de consumo entre as operações de leitura e escrita, neste trabalho, assumimos que metade da largura de banda é usada para operações de leitura e metade para operações de escrita.

B. Métricas

Nesta seção, descrevemos as principais métricas usadas para mensurar o custo de migração de VMs e as violações de SLA. Essas métricas foram definidas originalmente por [14] e as escolhemos por dois motivos. Primeiro, as heurísticas propostas em [14] são comparadas com as nossas propostas e, portanto, o uso das mesmas métricas facilita a comparação. Segundo, o conjunto de métricas criado por [14] para medir as violações, apesar de simples, é uma maneira genérica de avaliar o SLA.

Custo de Migração

Neste trabalho, utilizamos a definição de migração de VM proposta em [19], a qual consiste no processo de transferir, em tempo de execução, as páginas de memória de uma máquina virtual de um servidor para outro. Nesse processo, o hipervisor inicialmente realiza uma cópia prévia das páginas de memória que são, posteriormente, transferidas em várias rodadas. Para a cópia das páginas mais utilizadas, a execução da VM é interrompida, ocasionando um período de indisponibilidade (*downtime*). Assim que todas as páginas de memória são transferidas para o novo servidor, a VM retorna ao estado de execução. Seja j uma VM que utiliza uma quantidade de memória M_j e seja C a capacidade efetiva

Tabela I: Consumo de energia, em Watts, por percentual de utilização da CPU.

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

do enlace de rede que interliga o servidor de origem com o servidor de destino. O tempo necessário para migrar a VM j através desse enlace é definido por:

$$T_{M_j} = \frac{M_j}{C}. \quad (3)$$

Devido ao período de indisponibilidade, a migração tem impacto negativo no desempenho das aplicações que executam na VM. Em [14], a degradação de desempenho experimentada por uma VM j em virtude de migrações é estimada como 10% da utilização de CPU durante todas as suas migrações. Utilizamos a mesma abordagem em nosso trabalho.

Violação de SLA

Os autores de [14] definem três métricas para quantificar o nível do serviço experimentado pelas VMs e, portanto, pelas aplicações. A primeira métrica é chamada *SLATAH* (*SLA violation Time per Active Host*) e representa o percentual do tempo durante o qual os servidores ativos atingiram a utilização de 100% de CPU. Quando um servidor atinge sua capacidade máxima de processamento, o desempenho das aplicações que executam nas VMs fica limitado pela capacidade do servidor. Como consequência, é assumido que as VMs passam a não estar provisionadas com o nível de desempenho requerido. Formalmente, seja N o número de servidores, T_{s_i} o tempo total em que o servidor i experimentou uma utilização de 100% de CPU levando a uma violação de SLA e T_{a_i} o tempo total que o servidor i esteve ativo. Temos então a métrica *SLATAH* definida da seguinte maneira:

$$SLATAH = \frac{1}{N} \sum_{i=1}^N \frac{T_{s_i}}{T_{a_i}}. \quad (4)$$

A segunda métrica, denominada *PDM* (*Performance Degradation due to Migrations*), estima o total de degradação de desempenho causado pelas migrações. Seja M o número de VMs, C_{d_j} a estimativa de degradação de desempenho experimentada pela VM j em virtude de migrações e C_{r_j} a capacidade total de CPU requisitada pela VM j em todo o seu tempo de vida. A métrica *PDM* é então expressa por:

$$PDM = \frac{1}{M} \sum_{j=1}^M \frac{C_{d_j}}{C_{r_j}}. \quad (5)$$

A terceira métrica é chamada *SLAV* (*SLA Violation*) e consiste na combinação das duas métricas anteriores através da seguinte expressão:

$$SLAV = SLATAH \times PDM. \quad (6)$$

IV. PROPOSTA

O problema de consolidação de VMs pode ser dividido em quatro etapas, conforme sugerido em [14], as quais consistem em: 1) determinar quando um servidor deve ser considerado sobrecarregado, o que gera a necessidade de migração de uma ou mais VMs para outro equipamento; 2) determinar quais VMs deverão ser migradas do servidor identificado como sobrecarregado; 3) determinar para qual servidor as VMs devem ser migradas; e 4) determinar quando um servidor está subutilizado e, portanto, é, sob o ponto de vista energético, interessante realizar a migração de todas as VMs desse equipamento para seu posterior desligamento ou adormecimento.

Neste trabalho, apresentamos novas soluções para tratar as duas últimas etapas, ou seja, para alocação de VMs e para detecção de servidores subutilizados. A seguir, fornecemos informações adicionais sobre cada etapa, em especial a respeito das duas etapas nas quais introduzimos nossas soluções.

A. Detecção de servidores sobrecarregados

Os autores de [14] propõem e avaliam quatro algoritmos para realizar a detecção de servidores sobrecarregados. Eles verificaram que o método LR (*Local Regression*) apresenta o melhor resultado no geral. Em nossos testes, confirmamos esse comportamento, mas observamos que o algoritmo baseado em IQR (*Inter Quartil Range*) também apresenta resultados promissores quando é combinado com as nossas propostas. Portanto, apresentaremos avaliações com esses dois algoritmos, tanto de maneira isolada como em combinação com as nossas soluções.

Tanto o LR quanto o IQR são baseados em métodos estatísticos. O LR consiste em estabelecer modelos simples para subconjuntos de dados para construir uma curva que se aproxime dos dados reais. O IQR é uma medida de dispersão estatística obtida a partir da diferença entre o primeiro e o terceiro quartil.

B. Seleção de VMs

Após identificar um servidor como sobrecarregado, é necessário selecionar uma ou mais VMs desse servidor para migração. Esse processo de seleção de VMs para migração é repetido até que o servidor não esteja mais sobrecarregado. Em [14], os autores avaliaram três abordagens e verificaram que a MMT (*Minimum Migration Time*) apresenta o melhor desempenho. Em condições normais, o tempo de migração é função apenas do tamanho da VM, ou seja, a menor VM possui o menor tempo de migração. Neste trabalho, também utilizamos a abordagem MMT.

C. Alocação de VMs

Conforme comentamos anteriormente, a consolidação dinâmica de VMs pode ser tratada como um problema *bin packing*, com preços e tamanho de contêiner variáveis [14]. Em nosso problema, o contêiner é o servidor, os itens são as VMs, o tamanho do contêiner é a combinação da CPU e da RAM disponíveis no servidor e o preço é o custo energético da CPU e da RAM causado pela alocação das VMs. Adaptamos o algoritmo MBFD (*Modified Best Fit Decreasing*), proposto em [14], para realizar a alocação das VMs. Em sua versão original, o algoritmo ordena todas as VMs de maneira decrescente de uso de CPU e cada VM é alocada no servidor cuja alocação cause o menor aumento no consumo de energia. Em nossa modificação, denominada CREW (*CPU and RAM Energy aWare*), alteramos a verificação de consumo de energia causado pela alocação das VMs para levar em conta também a quantidade de RAM utilizada no servidor. Esse consumo é descrito pelo modelo de energia que apresentamos na Seção III-A.

D. Detecção de servidores subutilizados

Em [14], os autores utilizam uma abordagem simples para identificar os servidores subutilizados. Basicamente, após cada ciclo de execução das etapas 1) a 3), a etapa 4) consiste em escolher o servidor com a menor utilização naquele instante e tentar migrar suas VMs sem causar sobrecarga nos demais equipamentos. Conforme mostraremos na Seção V, a utilização instantânea não reflete adequadamente uma situação estável do servidor e pode levar a escolhas inadequadas, degradando o desempenho.

Nesse contexto, desenvolvemos um algoritmo chamado CREW-UD (*CREW with improved Underload Detection*) que realiza o procedimento de verificação sobre subutilização de um servidor em dois passos. Primeiramente, é gerada uma lista ordenada contendo os servidores que não estão desligados e também não foram identificados como sobrecarregados. Para cada servidor dessa lista, é calculada a média móvel exponencialmente ponderada (*Exponential Weighted Moving Average* – EWMA) da utilização de CPU, conforme segue:

$$EWMA_t = \alpha Y_t + (1 - \alpha)EWMA_{t-1}, \quad (7)$$

onde $EWMA_{t-1}$ é o cálculo anterior da média, Y_t é a utilização da CPU no tempo t e α é o coeficiente de suavização da EWMA. Neste trabalho, observamos que $\alpha = 0,1$ apresentou um resultado satisfatório para os dados que tínhamos disponíveis. O servidor com a menor utilização de CPU é escolhido para a avaliação que segue.

O segundo passo do nosso algoritmo consiste em verificar se o servidor candidato possui utilização inferior a um percentual ϕ da EWMA. Em caso afirmativo, é feita a verificação da possibilidade de realizar a migração de suas VMs para outros servidores sem que isso cause sobrecarga. Se a migração for possível, ela será realizada e o servidor será desativado. Caso contrário, ele permanecerá ligado. A partir dos dados que tínhamos disponíveis, observamos que $\phi = 0,3$ fornecia

o melhor desempenho para o nosso algoritmo. Realizamos avaliações de diversos valores tanto para ϕ quanto para α , no entanto, por falta de espaço, não apresentaremos os resultados.

V. AVALIAÇÃO

Nesta seção, apresentamos uma avaliação do impacto do consumo de energia dos circuitos eletrônicos de memória, ou RAM, e como esse consumo pode ser levado em conta em soluções para consolidação de VMs. Mostramos também a comparação dos resultados das nossas propostas com os de outras soluções da literatura. Conforme ilustraremos na Seção V-C, CREW e CREW-UD proporcionam sensível redução no consumo de energia e ainda reduzem o nível de violação de SLA.

Utilizamos o simulador CloudSim [20], versão 3.0.2, para realizar a avaliação. Dentre outras características, o CloudSim oferece apoio à modelagem e simulação de: centro de dados de larga escala, servidores virtualizados, consumo de energia dos recursos computacionais e políticas de alocação de VMs em servidores. Ou seja, todos os recursos básicos necessários em nosso trabalho já estão disponíveis no simulador. Naturalmente, algumas modificações foram necessárias, a saber:

- adição de um modelo de consumo de energia para RAM,
- inclusão de heurísticas de migração de VMs que levam em conta o consumo de energia da RAM,
- computação da EWMA de uso de CPU nos servidores.

De maneira similar a [14], simulamos um centro de dados com 800 servidores, sendo metade deles do modelo HP ProLiant ML110 G4 e metade do modelo HP ProLiant ML110 G5. Cada servidor possui a capacidade de transmissão em rede de 1 Gbps. As VMs utilizadas representam tipos de instâncias da nuvem Amazon EC2 [21], porém com configuração apenas para núcleo único. Essa configuração está adequada à carga de trabalho utilizada na simulação. A carga de trabalho, que detalhamos a seguir, foi extraída de um ambiente onde cada VM é executada apenas em um único núcleo do processador. Inicialmente, as VMs são alocadas de acordo com a demanda especificada pelo seu tipo. Ao longo do tempo, as VMs passam a utilizar recursos de acordo com o comportamento descrito pela carga de trabalho e surgem oportunidades para consolidação dinâmica das VMs.

Utilizamos também a mesma metodologia que [14] para geração de carga de trabalho, ou seja, usamos dados fornecidos pelo projeto CoMon [22] que monitora o PlanetLab [23]. Através desses dados, obtivemos o uso de CPU por mais de mil VMs que foram executadas em servidores espalhados por mais de 500 localidades em torno do planeta. Os dados correspondem a 10 dias escolhidos aleatoriamente durante os meses de março e abril de 2011. As medições de uso da CPU são realizadas a cada 5 minutos e, conforme esperado, a utilização média é baixa, não alcançando 13%. A carga de trabalho para a RAM é extraída do tipo de instância atribuído a cada VM instanciada dentre os quatro tipos selecionados: 870, 3.480, 1.740 e 613 MB.

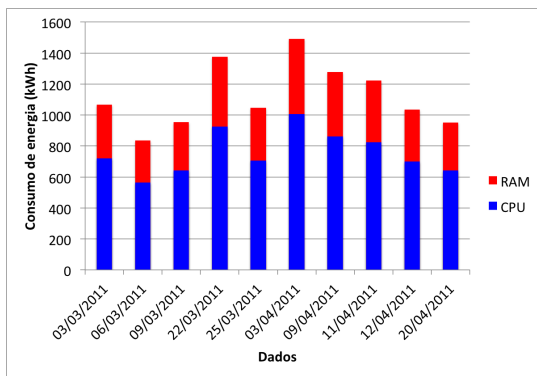


Figura 1: Consumo de energia de CPU e RAM em diferentes conjuntos de dados.

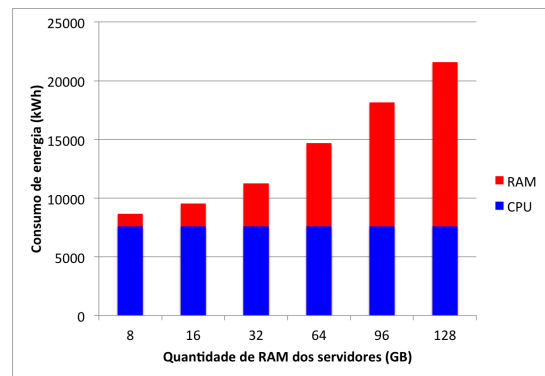


Figura 2: Consumo médio de energia de CPU e RAM em servidores com tamanhos variados de memória.

A. Consumo de energia da RAM

Em [14], os autores comentam sobre o impacto que o uso da RAM tem sobre o consumo de energia. No entanto, não é utilizado nenhum modelo para representar esse impacto e nenhum consumo é considerado nos experimentos realizados naquele trabalho. Conforme descrevemos na Seção IV, adaptamos o modelo proposto em [17] e o implementamos no CloudSim. A seguir, mostramos que há efetivamente impacto sensível do uso da RAM no consumo de energia.

A Figura 1 ilustra o consumo de energia total com CPU e RAM, utilizando a política CPU DVFS. Nesta figura, mostramos os resultados dos diferentes conjuntos de dados utilizados como carga de trabalho. Todos os servidores possuem a mesma quantidade de memória: 32 GB. Nessa configuração, podemos observar que a energia consumida pela memória permanece em torno de 30% do total em todos os conjuntos de dados. Em parte, esse resultado se deve ao consumo de fundo, o qual é constante para um dado tamanho de memória e representa uma parte expressiva do consumo total da energia por uso de RAM, aproximadamente 94% nessa política. Adicionalmente, o consumo por uso de CPU apresenta pequena variação entre os estados de carga mínima e máxima, conforme podemos observar na Tabela I.

Na Figura 2, apresentamos a média do consumo de energia dos diferentes conjuntos de dados em servidores com valores variados de memória. Novamente, os resultados se referem à política de alocação estática CPU DVFS. Conforme esperado, o consumo de energia é diretamente proporcional à quantidade de memória disponível. Esse comportamento é determinado pelo modelo de energia utilizado para memória, o qual descrevemos na Seção III-A. De acordo com o modelo, o consumo de fundo da RAM existe mesmo quando não há operações de leitura ou escrita, ou seja, o simples aumento da capacidade de RAM gera aumento no consumo de energia. Esse consumo pode ser reduzido através de técnicas como DVFS para memória [17]. No entanto, esse tipo de técnica traz pouco benefício quando a memória está sendo efetivamente utilizada. Por exemplo, as estratégias utilizadas neste trabalho buscam manter o mínimo de servidores ligados, mantendo o máximo de VMs em cada servidor dadas as restrições de SLA

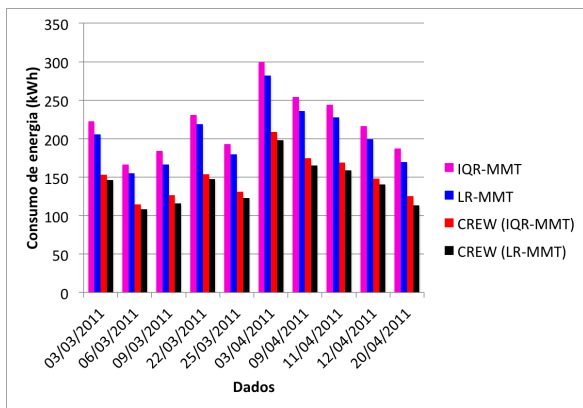
e de memória disponível. Nesse cenário, há uso intensivo da memória, tanto pelas VMs quanto pelo processo de migração.

B. Heurísticas cientes do consumo da RAM

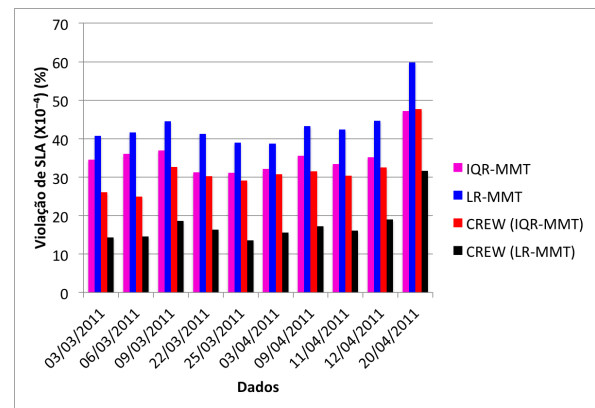
Nesta seção, mostramos a economia de energia que pode ser alcançada ao considerar o consumo da RAM na estratégia de consolidação de VMs. Essa parte da avaliação seguiu uma abordagem semelhante à utilizada em [14], ou seja, utilizamos suas duas métricas principais e também não limitamos a capacidade de memória dos servidores com o intuito de manter o foco nos algoritmos de consolidação de VMs.

A Figura 3a apresenta a energia consumida por cada heurística nos conjuntos de dados analisados. Podemos observar a economia de energia obtida ao incluir o consumo da RAM nas estratégias de consolidação de VMs. Na Figura 3b, é mostrado o nível de violação de SLA ocorrido nos mesmos conjuntos de dados. Além da economia de energia, as heurísticas cientes do consumo da RAM conseguem ainda reduzir o nível de violação do SLA. Esse resultado confirma que a inclusão de uma informação relevante – consumo de energia da RAM – permite que uma heurística melhore seu desempenho mesmo em métricas conflitantes, como economia de energia e violação de SLA. A heurística CREW combinada com a LR-MMT apresenta o melhor desempenho dentre todas as métricas, obtendo uma economia média de energia em torno de 31% em relação à LR-MMT. Além disso, a CREW(LR-MMT) exibe um nível de violação de SLA, em média, 51% menor que a heurística IQR-MMT.

Em [14], não é dada muita atenção à métrica de violação geral de SLA, que corresponde ao percentual do tempo em que o centro de dados apresentou algum nível de violação ao longo de toda a avaliação. A Tabela II mostra a violação geral de SLA média para as quatro heurísticas avaliadas. Podemos observar que a heurística CREW permanece com algum nível de violação de SLA durante um período maior de tempo. A heurística CREW(LR-MMT), que apresenta o melhor desempenho, ainda permanece em violação geral de SLA durante um período de tempo 12,5% superior ao da IQR-MMT. Essa deficiência motivou o aperfeiçoamento da heurística, dando origem à CREW-UD.



(a) Mediana da energia consumida em diferentes conjuntos de dados.



(b) Mediana da violação do acordo de nível de serviço em diferentes conjuntos de dados.

Figura 3: Impacto ao considerar o consumo de energia da RAM nas estratégias de consolidação de VMs.

Tabela II: Valores médios de violação geral de SLA.

IQR-MMT	LR-MMT	CREW(IQR-MMT)	CREW(LR-MMT)
0,08%	0,11%	0,12%	0,09%

A razão para a heurística CREW apresentar desempenho inferior às demais na métrica violação geral de SLA é a estratégia utilizada para desligamento de servidores subutilizados. Por projeto, a heurística CREW já mantém um número de servidores em operação menor do que as demais. A estratégia original para detecção de servidores subutilizados é baseada no valor instantâneo do uso de CPU e, portanto, servidores críticos para a heurística CREW podem ser incorretamente desligados por uma situação momentânea. A Figura 4 mostra o número de servidores que são mantidos ligados ao longo do tempo por cada heurística. É possível observar que a heurística CREW opera com um número de servidores ligados menor que as demais e, portanto, a escolha do servidor para potencial desativação deve ser criteriosa.

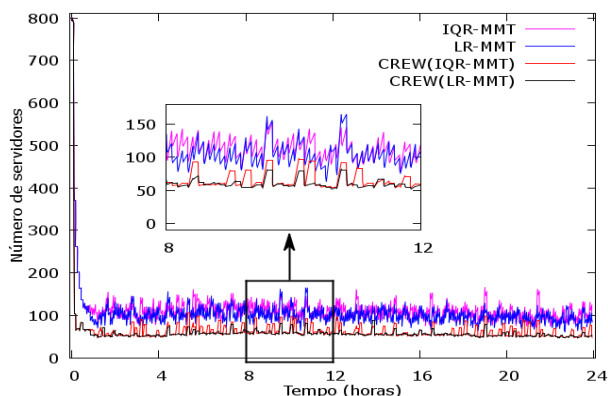


Figura 4: Número de servidores ligados ao longo do tempo.

C. Comparação

Por fim, realizamos uma comparação entre as heurísticas CREW e CREW-UD com abordagens convencionais como NPA (*Non Power Aware*) e CPU DVFS, e também com as heurísticas com melhor desempenho propostas em [14]. Nesta parte da avaliação, utilizamos as métricas anteriores e acrescentamos algumas para auxiliar na compreensão dos resultados obtidos. Além disso, voltamos a utilizar uma quantidade limitada de memória disponível em cada servidor, 32 GB. Logo, todas as heurísticas devem levar em consideração essa restrição ao realizar a consolidação de VMs. Essa abordagem difere da empregada em [14] e também na Seção V-B.

A Tabela III apresenta a mediana de algumas métricas utilizadas para avaliar as heurísticas. Mediana foi a estatística utilizada em [14] e, portanto, permite a comparação direta entre os resultados. Realizamos também avaliação com a média e obtivemos resultados muito semelhantes. A métrica ESV é o produto entre consumo de energia e violação de SLA (SLAV), cujo objetivo é representar de maneira combinada as informações das duas métricas.

Podemos observar que as estratégias NPA e CPU DVFS não geram violação de SLA ao custo de um elevado consumo de energia. Essas abordagens não realizam migrações de VMs e, portanto, perdem a oportunidade de economizar energia através do desligamento de servidores subutilizados. As heurísticas IQR-MMT e LR-MMT, que utilizam migração de VMs, obtêm uma economia de energia sensivelmente superior às estratégias NPA e CPU DVFS. Como consequência, essas heurísticas precisam lidar com a violação de SLA que surge com a redução do número de servidores disponíveis. É importante observar que as métricas SLAV e violação geral de SLA não representam aplicações específicas e nem seus parâmetros de desempenho. Logo, é possível que essas métricas subestimem o impacto efetivo no desempenho final percebido pelos usuários e, portanto, é importante reduzir ao máximo os valores das métricas de violação.

A heurística CREW consegue reduzir consideravelmente

Tabela III: Resultados das heurísticas avaliadas sob diferentes métricas.

	ESV($\times 10^{-3}$)	Energia (kWh)	SLAV ($\times 10^{-5}$)	Violação geral de SLA (%)	Migrações
NPA	0,0000	3.252,735	0,000	0,00	0
CPU DVFS	0,0000	1.057,195	0,000	0,00	0
IQR-MMT	3,0667	294,420	10,416	0,14	38.117
LR-MMT	3,3825	269,995	12,528	0,19	37.841
CREW(IQR-MMT)	0,4749	196,065	2,422	0,10	23.747
CREW(LR-MMT)	0,2982	185,765	1,605	0,08	14.054
CREW-UD(IQR-MMT)	0,0935	232,770	0,402	0,03	13.432
CREW-UD(LR-MMT)	0,0471	235,630	0,200	0,02	6.599

tanto o consumo de energia quanto a violação de SLA. Essa redução é obtida principalmente por dois fatores: 1) menor número de servidores ligados ao longo do tempo, conforme mostramos na Seção V-B; 2) diminuição severa do número de migrações, como podemos observar na Tabela III. A heurística CREW-UD reduz a violação de SLA, tolerando uma pequena elevação no consumo de energia. De acordo com a métrica combinada ESV, a heurística CREW-UD(LR-MMT) apresenta o melhor desempenho, exibindo também os menores valores de violação de SLA.

VI. CONCLUSÃO

Neste trabalho, mostramos a importância de considerar a energia consumida pela RAM em estratégias de uso energético eficiente. Desenvolvemos heurísticas que utilizam a informação do consumo de energia da RAM para auxiliar na decisão de migração de VMs em centros de dados. Avaliamos e comparamos, através de simulação, os resultados das nossas propostas com outras soluções da literatura. Verificamos que nossas heurísticas reduzem o consumo de energia e melhoram o SLA oferecido às aplicações que usam os centros de dados.

Como trabalhos futuros, planejamos investigar alternativas para os algoritmos de detecção de servidores sobrecarregados, buscando soluções mais eficientes energética ou computacionalmente. Adicionalmente, pretendemos implementar e avaliar nossas propostas no OpenStack Neat, um arcabouço para consolidação dinâmica de VMs.

AGRADECIMENTOS

Este trabalho foi parcialmente financiado por CAPES, CNPq, FAPEG e RNP.

REFERÊNCIAS

- [1] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: towards a cloud definition," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 50–55, 2008.
- [2] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Gener. Comput. Syst.*, vol. 25, no. 6, pp. 599–616, 2009.
- [3] T. Mukherjee, A. Banerjee, G. Varsamopoulos, S. K. Gupta, and S. Rungta, "Spatio-temporal thermal-aware job scheduling to minimize energy consumption in virtualized heterogeneous data centers," *Computer Networks*, vol. 53, no. 17, pp. 2888–2904, 2009.
- [4] Greenpeace, "How dirty is your data? A Look at the Energy Choices That Power Cloud Computing," <http://www.greenpeace.org/international/en/publications/reports/How-dirty-is-your-data/>, 2011, Último acesso: 29-junho-2013.
- [5] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *Proceedings of the 34th annual international symposium on Computer architecture*, 2007, pp. 13–23.
- [6] L. Barroso and U. Holzle, "The Case for Energy-Proportional Computing," *Computer*, vol. 40, no. 12, pp. 33–37, 2007.
- [7] L. Benini, A. Bogliolo, and G. De Micheli, "A survey of design techniques for system-level dynamic power management," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 8, no. 3, pp. 299–316, 2000.
- [8] C.-H. Hwang and A. C.-H. Wu, "A predictive system shutdown method for energy saving of event-driven computation," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 5, no. 2, pp. 226–241, 2000.
- [9] M. Weiser, B. Welch, A. Demers, and S. Shenker, "Scheduling for Reduced CPU Energy," *Mobile Computing*, vol. 353, pp. 449–471, 1996.
- [10] A. Wierman, L. L. H. Andrew, and A. Tang, "Power-aware speed scaling in processor sharing systems: Optimality and robustness," *Perform. Eval.*, vol. 69, no. 12, pp. 601–622, 2012.
- [11] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang, "Power and performance management of virtualized computing environments via lookahead control," *Cluster Computing*, vol. 12, no. 1, pp. 1–15, 2009.
- [12] A. Verma, P. Ahuja, and A. Neogi, "pMapper: power and migration cost aware application placement in virtualized systems," in *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*, 2008, pp. 243–264.
- [13] Q. Zhang, M. F. Zhani, R. Boutaba, and J. L. Hellerstein, "HARMONY: Dynamic Heterogeneity-Aware Resource Provisioning in the Cloud," in *Proceedings of the 33rd International Conference on Distributed Computing Systems (ICDCS)*, 2013.
- [14] A. Beloglazov and R. Buyya, "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.*, vol. 24, no. 13, pp. 1397–1420, 2012.
- [15] K. Lim, J. Chang, T. Mudge, P. Ranganathan, S. K. Reinhardt, and T. F. Wenisch, "Disaggregated memory for expansion and sharing in blade servers," *SIGARCH Comput. Archit. News*, vol. 37, no. 3, pp. 267–278, 2009.
- [16] R. Nathuji and K. Schwan, "VirtualPower: coordinated power management in virtualized enterprise systems," in *Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles*, 2007, pp. 265–278.
- [17] H. David, C. Fallin, E. Gorbato, U. R. Hanebutte, and O. Mutlu, "Memory power management via dynamic voltage/frequency scaling," in *Proceedings of the 8th ACM international conference on Autonomic computing*, 2011, pp. 31–40.
- [18] SPEC, "SPECpower_ssj2008 benchmark," http://www.spec.org/power_ssj2008/, 2013, Último acesso: 15-julho-2013.
- [19] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*, 2005, pp. 273–286.
- [20] CLOUDSLab, "CloudSim: A Framework For Modeling And Simulation Of Cloud Computing Infrastructures And Services," <http://www.cloudbus.org/cloudsim/>, 2013, Último acesso: 29-junho-2013.
- [21] Amazon, "Amazon EC2 Instance Types," <http://aws.amazon.com/ec2/#instance>, 2013, Último acesso: 29-junho-2013.
- [22] K. Park and V. S. Pai, "CoMon: a mostly-scalable monitoring system for PlanetLab," *SIGOPS Oper. Syst. Rev.*, vol. 40, no. 1, pp. 65–74, 2006.
- [23] PlanetLab, "An open platform for developing, deploying, and accessing planetary-scale services," <http://www.planet-lab.org>, 2013, Último acesso: 29-junho-2013.