

Proposição de um Modelo de Consumo de Energia para Aplicações Elásticas em Nuvem

Gustavo Rostirolla¹, Ivam Guilherme Wendt¹, Rodrigo da Rosa Righi¹ e Cristiano André da Costa¹

¹Programa Interdisciplinar de Pós-Graduação em Computação Aplicada, Universidade do Vale do Rio dos Sinos (Unisinos) - RS - Brasil

{grostirolla1, mail.ivam}@gmail.com, {rrrighi, cac}@unisinos.br

Abstract. *One of the main characteristics of cloud computing is elasticity, which refers to the capacity of on-the-fly changing the number of resources to support the execution of a task. One of the main challenges in this scope is how to measure its effectiveness, because of elasticity enables high performance computing by reducing the application time, but an infeasible amount of resource and/or energy can be paid to accomplish this. Particularly, the state-of-the-art does not present an energy consumption model that fits a malleable number of resources, but only a fixed and predefined number of them. In this context, this article proposes an elastic energy consumption model called EME. The results revealed a median accuracy of 97.15% when analyzing data from the model against real energy consumption data. In addition, we also proposed an empirical energy model, and evaluated both with AutoElastic, a middleware designed to provide transparent elasticity for HPC applications. Our results provide a detailed profile of how much energy is spent on each virtual machine (VM) configuration allowing us to better manage the energy consumption.*

Resumo. *Uma das principais características da computação em nuvem é a elasticidade, que se refere à capacidade de alterar a quantidade de recursos em tempo real a fim de otimizar a execução de uma tarefa. Um dos principais desafios é como medir a sua eficácia. Utilizando elasticidade em aplicações HPC é possível reduzir o tempo de aplicação, mas consumindo uma grande quantidade de recursos e/ou energia para concluir a tarefa. Particularmente, observa-se que o estado da arte não apresenta um modelo de consumo de energia que contempla um número maleável de recursos, mas apenas um número fixo e predefinido deles. Neste contexto, propõe-se um modelo de consumo de energia elástico chamado EME. Os resultados revelaram uma acurácia média de 97,15% ao comparar os dados do modelo com os dados de consumo de energia real. Além disso, também é avaliado um modelo de energia empírico, ambos utilizando AutoElastic, um middleware projetado para fornecer elasticidade de forma transparente para aplicativos HPC. Os resultados fornecem um perfil detalhado de quanta energia é gasta em cada configuração de máquina virtual (VM), o que permite uma melhor gerencia do consumo de energia.*

1. Introdução

Uma das principais características da computação em nuvem é a elasticidade, na qual os usuários podem escalar seus recursos computacionais a qualquer momento, de acordo

com a demanda ou o tempo de resposta desejado [Lorido-Botran et al. 2014]. Considerando uma aplicação paralela de longa execução, um usuário pode querer aumentar o número de instâncias para tentar reduzir o tempo de conclusão da tarefa. Logicamente, o sucesso deste processo vai depender tanto do grão quanto da modelagem da aplicação. Por outro lado, se a tarefa não escala de forma linear ou perto de uma forma linear, e se o utilizador é flexível com respeito ao tempo de conclusão, o número de instâncias pode ser reduzida. Isso resulta em uma menor quantidade nós \times horas, e portanto, em um custo mais baixo e melhor uso da energia. Graças aos avanços na área de virtualização [Petrides et al. 2012], a elasticidade em computação em nuvem pode ser uma alternativa viável para obter economia de custo significativa quando comparado com o método tradicional de manter uma infra-estrutura de TI baseada em *cluster*. Normalmente, neste último caso, há um dimensionamento para o uso de pico, sendo subutilizada quando observamos toda a execução do aplicativo ou ainda, ao analisar o uso real da infra-estrutura.

Elasticidade pode ser uma faca de dois gumes envolvendo desempenho e o consumo de energia. Ambos são diretamente relacionados ao consumo de recursos, o que também pode ajudar a medir a qualidade elasticidade. Embora elasticidade permita que os aplicativos aloquem e liberem recursos de forma dinâmica, ajustando às demandas da aplicação, estabelecer limites apropriados, medir o desempenho e consumo de energia com precisão neste ambiente não são tarefas fáceis [Lorido-Botran et al. 2014]. Desta forma, um utilizador pode conseguir um bom desempenho considerando o tempo para executar a sua aplicação, mas utilizando uma grande quantidade de recursos, resultando em um desperdício de energia. A ideia de apenas obter um melhor desempenho da aplicação com uma execução elástica, em alguns casos, não é suficiente para usuários e administradores da nuvem. Os usuários acabam pagando por um maior número de recursos, não efetivamente utilizados, de acordo com o paradigma *pay-as-you-go*. A medição do consumo de energia de tais sistemas elásticos não é uma tarefa fácil. Muitos trabalhos se concentram em medição e como estimar o consumo de energia em *data centers*, no entanto, essas tarefas são desafios ao considerar sistemas elásticos.

Este artigo apresenta EME, que é um passo rumo a modelagem do consumo de energia de aplicações HPC elásticas. O modelo leva em conta diversos indicadores, tais como consumo de energia, consumo de recursos e desempenho, para fornecer uma métrica de consumo de energia precisa para infraestruturas de nuvem. EME é capaz de modelar o consumo de energia das aplicações mesmo sob diferentes cargas de trabalho e mudanças na estrutura da nuvem (elasticidade). A contribuição científica do modelo EME aparece é um conjunto de equações que fazem uso de parâmetros da infraestruturas de nuvem elásticas retornando o consumo de energia de uma determinada aplicação de forma detalhada. Além do modelo EME, está presente também um modelo energético empírico. Este modelo apresenta o consumo de energia com base na alocação de recursos ao longo do tempo, permitindo assim uma análise da correlação entre o provisionamento de recursos e do consumo de energia.

Com o objetivo de analisar os dois modelos de energia que utiliza-se um trabalho anterior chamado AutoElastic [Righi et al. 2015], um middleware elasticidade reativa que gerencia recursos de nuvem de acordo com a demanda de um aplicativo HPC. EME foi incorporado como um plug-in para o AutoElastic, criando traços e armazenando o consumo

de energia das aplicações em tempo de execução. Os resultados com uma aplicação de uso intensivo de CPU, realizados em diferentes cenários e variando os valores dos limites (*thresholds*) inferior e superior e com diferentes cargas de trabalho (Crescente, Decrescente, Constante e Onda). Os resultados, apontam EME como uma solução viável para avaliar a qualidade da elasticidade, ou seja, a eficácia do middleware de elasticidade ao observar o consumo de energia, alocação de recursos e métricas de desempenho.

O restante deste trabalho apresenta primeiramente os modelos de energia na Seção 2, seguido da metodologia de avaliação utilizada na Seção 3, avaliação dos resultados na Seção 4, trabalhos relacionados na Seção 5 e por fim a conclusão na Seção 6.

2. Modelos de Consumo de Energia para Ambientes de Nuvem com Elasticidade

Esta seção apresenta dois modelos de consumo de energia: (i) EME, que é a principal contribuição deste artigo; (ii) um modelo empírico de energia que explora a relação entre consumo de energia e de recursos. Ambos levam em consideração a elasticidade horizontal, onde VMs são adicionadas *on-the-fly*, ou removidas para dar suporte a execução de uma aplicação HPC. Os modelos de energia são projetados para medir o consumo de energia de sistemas homogêneos, ou seja, sistemas que usam o mesmo modelo para instanciar suas VMs e executam sobre uma mesma arquitetura de hardware. Por uma questão de clareza, os modelos são apresentados em seções separadas, primeiramente o modelo EME, seguido do modelo empírico.

2.1. EME: Modelo de Energia para Aplicações HPC em Nuvem com Elasticidade

Esta seção apresenta o modelo EME, um modelo de consumo de energia para extrair dados sobre o consumo, explorando as relações entre o consumo de energia, consumo de recursos e desempenho. O modelo apresentado leva em consideração uma das principais características da computação em nuvem, a elasticidade, onde a quantidade de recursos muda durante o tempo de execução.

A implantação de sensores de corrente ou Wattímetros pode ser caro se não for feito no momento em que toda a infraestrutura (*i.e.*, *cluster* ou *data center*) é instalada, além de ser custosa tanto em questões financeiras como em tempo conforme a infraestrutura cresce. Uma solução alternativa e menos dispendiosa é a utilização de modelos de energia para estimar o consumo de componentes ou de um *data center* inteiro [Orgerie et al. 2014]. Bons modelos devem ser leves (em relação ao consumo de recursos computacionais) e não interferir no consumo de energia que eles tentam estimar. Tendo em vista estes requisitos, o modelo proposto explora dados de energia capturados em um pequeno conjunto de nós, a fim de formular uma equação que estende os resultados para um conjunto arbitrário de nós homogêneos. Mais precisamente, a metodologia utilizada é similar a de Luo et al. [Luo et al. 2013] que consiste em três etapas:

- (i) Coletar amostras de uso de recursos, bem como o consumo de energia da máquina utilizando um medidor de consumo. Neste caso, utilizou-se um medidor Minipa ET-4090 que coletou mais de 8000 amostras usando uma carga composta que pode consumir diversos tipos de recursos dos nós, a fim de representar aplicações reais em ambiente de nuvem [Chen et al. 2014];

- (ii) Executar métodos de regressão para gerar o modelo de energia a ser utilizada posteriormente;
- (iii) Testar o modelo em um conjunto diferente de dados, coletados com o medidor de diferentes máquinas homogêneas, a fim de validar se o modelo representa corretamente o consumo de energia das demais máquinas.

A fim de analisar a precisão do modelo gerado foram coletados dados de CPU, memória principal e consumo de energia instantâneos, aplicando posteriormente PCR (Regressão de Componentes Principais) em mais de 8000 amostras obtidas a partir de um único nó. Os dados recolhidos estão alinhados com estudos anteriores [Orgerie et al. 2014], que apresentam a CPU como o principal vilão do consumo de energia. Após a geração deste modelo foi realizada a predição da mesma quantidade de amostras de energia baseada em amostras coletadas de CPU e memória de outro nó com mesma configuração de hardware. Comparando estas amostras geradas pela predição de consumo, com as amostras coletadas com o medidor, obteve-se uma precisão média e mediana de 97,15% e 97,72% respectivamente, como pode ser visto na Figura 1.

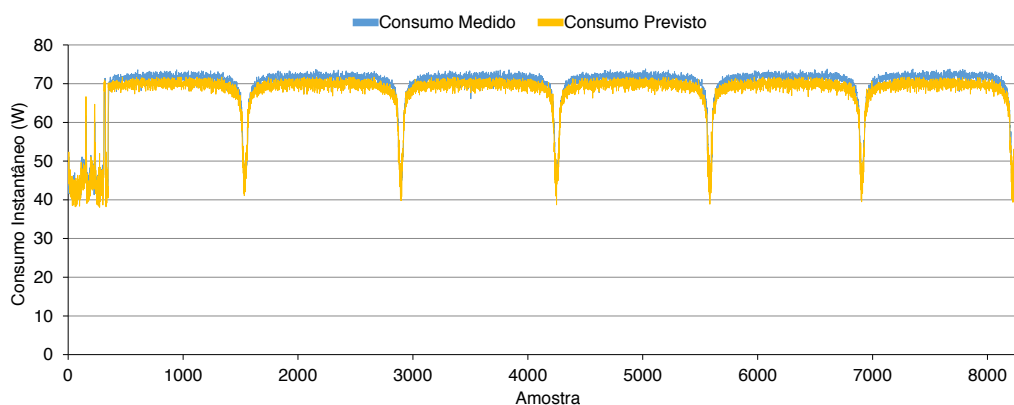


Figura 1. Comparativo do consumo instantâneo entre o consumo previsto e o consumo medido.

Após a execução da aplicação, os dados de CPU e memória principal são utilizados como entrada no modelo gerado, a fim de se obter o consumo de energia instantânea, medido em Watts (W). A grande vantagem deste modelo é o fato de considerar a elasticidade da nuvem, em outras palavras, o modelo leva em conta apenas o consumo de energia dos recursos que foram efetivamente utilizados, e não o consumo total do *data center*, ou um nó específico. A quantidade de recursos alocados é coletada de todos os nós durante o tempo de execução da aplicação, e através de um arquivo de log que informa o intervalo de tempo que cada máquina é utilizada, apenas as amostras relativas a execução da aplicação são consideradas para o cálculo do consumo de energia. Este processamento de registro é executado *post-mortem* e permite uma análise mais precisa do consumo de energia da aplicação, e não apenas o consumo de energia de toda a infraestrutura. Esta granularidade mais fina permite a utilização de funções de custo, por exemplo, a fim de determinar a viabilidade da utilização da elasticidade em nuvem para executar uma determinada aplicação.

O modelo de consumo também poderia ser empregado para avaliar os ambientes de computação em nuvem heterogêneos, uma vez que é baseado em um modelo já consolidado [Orgerie et al. 2014] apresentado na Equação 1 onde α representa um consumo

de energia quando o nó está ocioso e β e δ representam o consumo de energia variável determinado pela quantidade de recursos utilizados (neste caso de CPU e de memória) e retornando o consumo de energia instantâneo em Watts. A única adaptação necessária para contemplar ambientes heterogêneos seria a criação de modelos de consumo de energia distintos para cada tipo de máquina presente no data center.

Para complementar esta análise, apresenta-se um conjunto de equações que permitem o cálculo do consumo de energia em ambientes elásticos e também a quantidade de energia gasta por um determinado número de nós. A Equação 2 resulta no consumo de energia de uma máquina m de acordo com o valor de CPU e memória registrados em um instante i , utilizando a Equação 1 como base. A Equação 3 é utilizada para calcular o consumo total de energia de todas as máquinas alocadas em um instante t , ou seja, levando em conta a elasticidade, retornando o consumo em Watts. A Equação 4 calcula o consumo de energia total de um instante 0 a um instante t onde intervalos de tempo são calculados em segundos e utilizando a Equação 3 mencionada anteriormente que já considera a questão elasticidade, este cálculo resulta no consumo de energia em Joules ($W \times segundo$). Finalmente, a Equação 5 apresenta o consumo de energia da aplicação quando utilizando uma quantidade específica de nós representados por z . Este cálculo resulta no consumo total de energia, também representada em Joules, gasto quando utilizando esta quantidade específica de nós.

$$f(CPU, Memoria) = \alpha + \beta \times CPU + \delta \times Memoria \quad (1)$$

$$MC(m, i) = f(CPU(m, i), Memoria(m, i)) \quad (2)$$

$$ETC(t) = \sum_{i=0}^{Maquina} MC(i, t) \times x \begin{cases} x = 0 & \text{se a máquina } i \text{ não está ativa no instante } t; \\ x = 1 & \text{se a máquina } i \text{ está ativa no instante } t. \end{cases} \quad (3)$$

$$TC(t) = \sum_{i=0}^t ETC(i) \{ 0 \leq t \leq TempoTotalAplicacao \} \quad (4)$$

$$NEC(z) = \sum_{i=0}^{TempoApp} ETC(i) \times y \begin{cases} y = 0 & \text{se no instante } i \text{ o total de máquinas ativas } \neq z; \\ y = 1 & \text{se no instante } i \text{ o total de máquinas ativas } = z. \end{cases} \quad (5)$$

2.2. Modelo de Energia Empírico

Para melhorar a reprodutibilidade dos resultados e fornecer uma maneira mais fácil de medir a energia consumida durante a execução da aplicação, também está presente nesta seção um modelo energético empírico que baseia-se na relação entre o consumo de energia e recursos [Orgerie et al. 2014]. Para tal, utiliza-se a Equação 6 para criar um índice do uso de recursos, sendo i o número de máquinas virtuais e $T(i)$ a quantidade de tempo que o aplicativo foi executado com este conjunto de máquinas virtuais. Este modelo é

similar ao modelo de cobrança utilizado por provedores de nuvem como Amazon AWS e Microsoft Azure, onde é considerado o número de VMs em cada unidade de tempo, que é normalmente definida como uma hora [Roloff et al. 2012].

$$EnergiaEmpirica = \sum_{i=1}^n (i \times T(i)) \quad (6)$$

A Equação 6 apresenta como é calculada a energia empírica. Esta equação explora a relação entre a alocação de recursos e o tempo quando se utiliza uma configuração particular. Portanto, a unidade de tempo depende da medida de $T(i)$ (em minutos e segundos ou milésimos de segundo, e assim por diante), entretanto, esta unidade de tempo não é relevante para fins de comparação, desde que todas as execuções utilizem a mesma unidade. Por exemplo, considerando um cenário com uma unidade de tempo em segundos e um tempo de conclusão de aplicação de 1200 segundos, é possível obter-se o seguinte cálculo: 100 segundos (2 VMs), 200 segundos (4 VMs), 200 segundos (2 VMs), 200 segundos (4 VMs), 200 segundos (6 VMs), 200 segundos (4 VMs) e 100 segundos (2 VMs).

A estratégia de analisar o tempo de execução parcial sobre cada tamanho de infraestrutura é relevante por duas razões: (i) para fins de comparação entre os diferentes execuções com elasticidade habilitada; e (ii) investigar a relação entre as curvas de consumo obtidas através do modelo EME e as obtidas com o modelo de energia empírica.

3. Metodologia de Avaliação

Esta seção apresenta os detalhes técnicos sobre como foi conduzida a avaliação dos modelos. Primeiramente apresenta-se na subseção 3.1 a infra-estrutura e o cenário de teste, e por fim, os padrões carga da aplicação na subseção 3.2.

3.1. Infraestrutura, Middleware de Elasticidade e Ambiente de Testes

Os experimentos foram conduzidos utilizando uma nuvem privada OpenNebula com 6 (1 FrontEnd e 5 nós) nós homogêneos. Cada nó possui um processador de 2,9 GHz dual-core com 4 GB de memória RAM e uma rede de interconexão de 100 Mbps. Tanto o modelo EME quanto o modelo de energia empírico foram desenvolvidos como plug-ins para um middleware de elasticidade chamado AutoElastic [Righi et al. 2015]. AutoElastic, funciona com elasticidade horizontal e reativa, fornecendo alocação e consolidação de nós de computação e máquinas virtuais que executam uma aplicação paralela iterativa [Righi et al. 2015]. AutoElastic usa a métrica CPU para fornecer elasticidade reativa baseado em *thresholds* de forma transparente e sem esforço por parte do usuário, que não precisa escrever regras de elasticidade para a reconfiguração de recursos. Neste caso os *thresholds* utilizados foram 30 e 50 para os valores inferiores e 70 e 90 para os valores superiores, considerados valores representativos de acordo com trabalhos anteriores [Righi et al. 2015].

3.2. Aplicação Paralela

A aplicação utilizada nos testes calcula a aproximação para a integral do polinômio $f(x)$ num intervalo fechado $[a, b]$. Para tal, foi implementado o método de Newton-Cotes para

intervalos fechados conhecido como Regra dos Trapézios Repetida [Comanescu 2012]. A fórmula de Newton-Cotes pode ser útil se o valor do integrando é dada em pontos igualmente espaçados.

A carga de trabalho recebida pelo processo mestre consiste em uma lista de equações e seus parâmetros, enquanto o retorno é a mesma quantidade de valores de integração numérica. Buscando analisar o impacto de diferentes configurações de *thresholds* na aplicação paralela com diferentes padrões de carga. Foram definidos quatro padrões de carga: Constante, Crescente, Decrescente e Onda. Cargas sintéticas de trabalho foram escolhidas por serem consideradas uma forma representativa de avaliação de elasticidade de nuvens computacionais [Islam et al. 2012]. A Figura 2 apresenta graficamente uma representação de cada padrão de carga. O eixo *x* expressa a iteração (cada iteração representa uma equação que será calculada, dividida e distribuída pelo processo mestre), enquanto o eixo *y* representa a respectiva carga de processamento para aquela iteração.

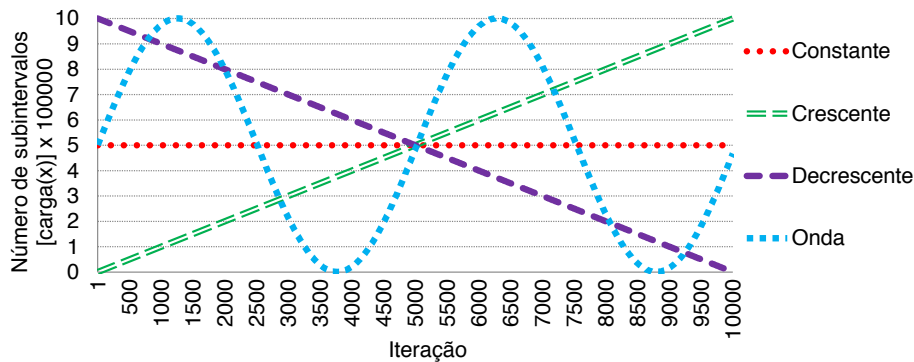


Figura 2. Visão gráfica dos padrões de carga.

4. Avaliação dos Resultados

Para melhor compreensão a avaliação dos resultados foi dividida em três momentos. O primeiro apresenta os resultados relativos ao consumo de energia durante a execução de uma aplicação na nuvem sem a utilização de elasticidade. O segundo momento analisa os resultados de uma execução em nuvem com elasticidade, variando os *thresholds* superiores e inferiores. Estes dois momentos apresentam resultados referentes ao modelo EME e ao modelo energético empírico. Por fim apresenta-se uma análise de correlação entre os modelos.

4.1. Consumo de Energia Sem Elasticidade

A Figura 3 ilustra o consumo de energia em Watts de acordo com o modelo EME quando as ações de elasticidade estão desativadas. Neste contexto, um único nó com duas VMs está sendo usado para executar os processos escravos. Observa-se que o simples fato de ligar o nó (com Sistema Operacional Ubuntu Linux) consome cerca de 40 Watts. Qualquer atividade de cálculo provoca uma elevação desse índice ao intervalo entre 40 e 71 Watts. Embora a função Crescente tenha um crescimento lento com relação a carga da CPU, o consumo de energia aumenta rapidamente até o limite superior do intervalo mencionado. O mesmo comportamento aparece nas funções Decrescente e Onda. Na função de Onda,

especialmente, por possuir uma queda de processamento na metade de sua execução de acordo com uma função senoidal. O modelo energético empírico, por sua vez, apresenta um índice aproximado de 8600 para todos os padrões de carga. Este índice se refere à área do consumo execução e recurso: quando arredonda-se uma multiplicação de aproximadamente 4296 segundos por 2 máquinas virtuais.

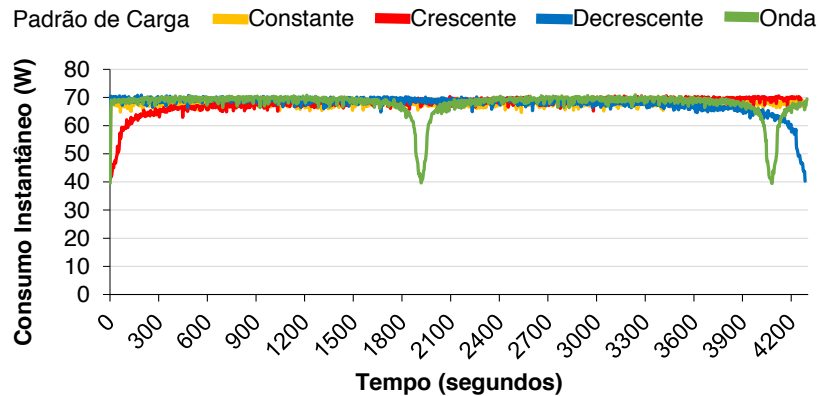


Figura 3. Consumo de energia (em Watts) dos diferentes padrões de carga, sem elasticidade.

4.2. Consumo de Energia Com Elasticidade Ativa

Quando a elasticidade está ativa é possível observar uma grande variação no número de VMs alocadas para executar a aplicação paralela, impactando diretamente sobre o tempo e o consumo de energia da execução. A Figura 4 apresenta um perfil do consumo de energia da aplicação utilizando o modelo EME, considerando os quatro padrões de carga. Em particular, este perfil apresenta os resultados da Equação 5, demonstrando mais claramente a distribuição e uso de VMs para cada combinação de *thresholds*. Observou-se que o consumo de energia cresce juntamente com o número de máquinas virtuais utilizadas para cada carga, confirmando a relação entre a utilização de recursos e consumo de energia. Iniciando a execução com único nó (2 VMs), a aplicação com carga Crescente, Decrescente e Onda alocam até 5 nós (10 VMs), enquanto a Constante utiliza no máximo 4 nós (8 VMs).

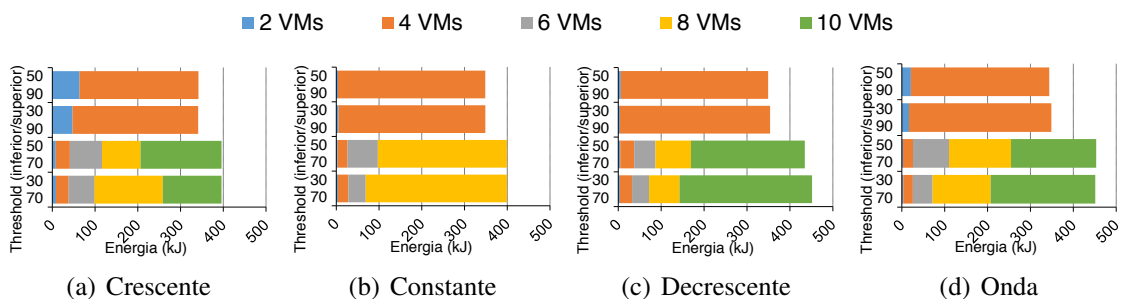


Figura 4. Consumo energético para diferentes quantidades de máquinas virtuais e cargas de trabalho variando os thresholds inferior e superior.

Na Figura 4 também é possível observar que a variação do *threshold* superior impacta diretamente no consumo total de energia. O valor de 70% implica no desencadeamento de alocações de VMs de forma mais reativa, uma vez que a carga do sistema excede

este limite com maior frequência. A mesma situação sob uma perspectiva diferente por ser observada com um *threshold* superior maior, o que aumenta o tempo de execução, mas fornece um melhor consumo de energia. Em outras palavras, um valor próximo de 100% para o limite superior adia a reconfiguração recurso, mantendo o estado de sobrecarga por mais tempo. Por exemplo, o valor de 70% e 90% são responsáveis pela alocação de 10 e 4 VMs no padrão de carga Crescente, respectivamente. Apesar de alocar 6 VMs a mais no primeiro caso, a diferença no consumo de energia não segue esta proporção: mesmo executando com menos VMs, o segundo caso mantém a CPU sobrecarregada (cerca de 90%) por mais tempo, impactando diretamente sobre o consumo energético. O limite inferior não apresenta impacto significativo sobre a execução.

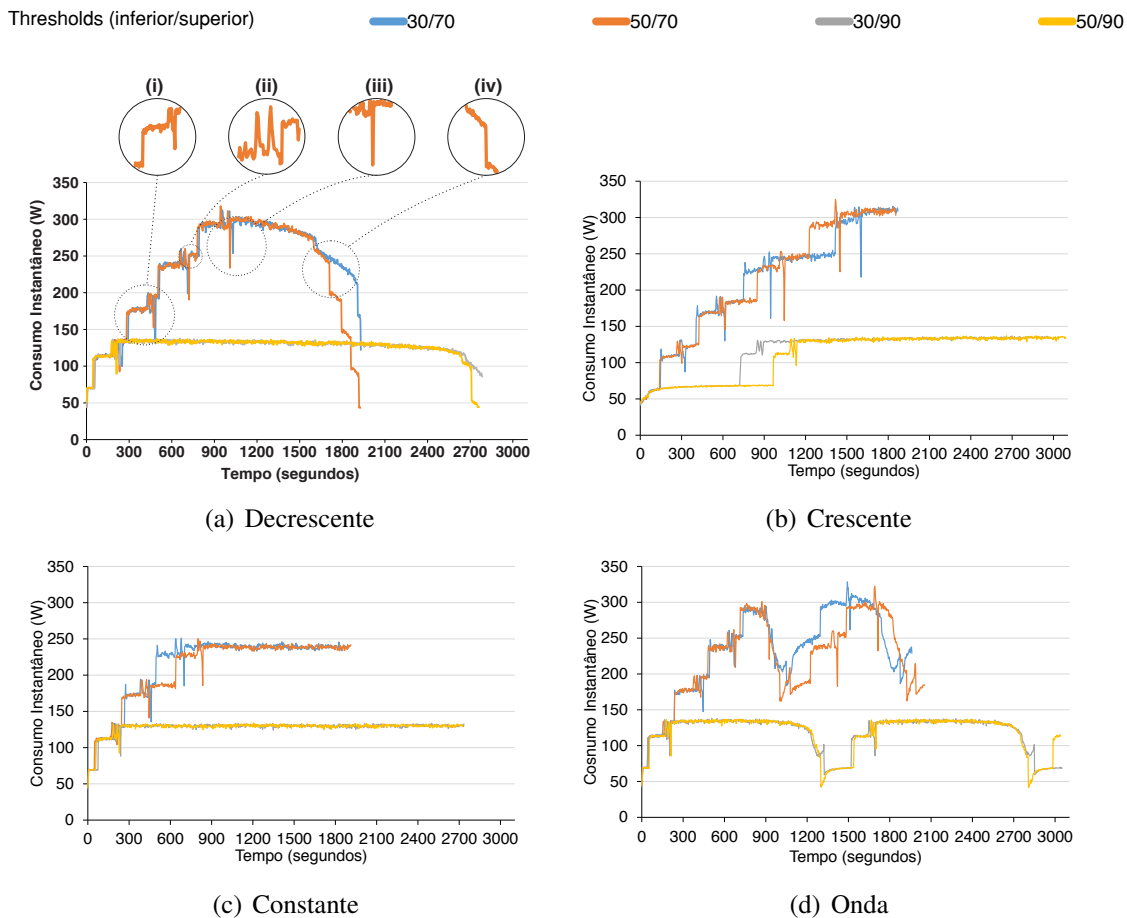


Figura 5. Comportamento do consumo energético das diferentes cargas de trabalho variando os thresholds inferior e superior. Em (a) resalta-se (i) alocação de host; (ii) inicialização de uma máquina virtual; (iii) parada de processamento para incorporar novos recursos; (iv) desalocação de host.

A Figura 5 apresenta os gráficos de execução da aplicação, mais especificamente a Figura 5 (a) destaca picos e quedas bruscas de consumo de energia quando se analisa o consumo de energia de forma elástica, utilizando a Equação 3 durante o tempo total de execução da aplicação. Neste gráfico podemos observar alocação e desalocação de *hosts*, além de oscilações durante a inicialização das VMs. Os gráficos apresentam as vantagens em analisar a aplicação utilizando um modelo elástico, pois considera apenas o consumo de energia das máquinas que executam computação, e representa de forma mais fiel o consumo energético de uma aplicação que faz uso da elasticidade.

Além dos resultados EME ilustrados na Figura 5, a Tabela 1 apresenta os resultados do modelo de energia empírico com e sem elasticidade na nuvem. Durante a execução da aplicação sem elasticidade os resultados apontam um menor consumo de energia, e um maior tempo de execução. No entanto, no âmbito de aplicações HPC, o objetivo geralmente é o de reduzir o tempo de execução para resolver um determinado problema. Considerando o tempo de execução, esta tabela mostra que, quanto menor o valor do limite superior, melhor o tempo de execução da aplicação. O limite inferior não têm um impacto significativo na execução da aplicação: a exceção ocorre no padrão de carga Decrescente. Nesta situação, o uso de 30% é responsável pela consolidação de recursos de modo menos reativo, fazendo com que a aplicação seja executada com a mesma quantidade de recursos por mais tempo. Por outro lado, o valor de 50% responsável por antecipar a desalocação de VM, eleva a carga da CPU nas demais instâncias já que a demanda de processamento é distribuída entre um menor número de nós.

Tabela 1. Tempo e consumo de energia dos dois modelos com e sem elasticidade variando as cargas de trabalho e *thresholds*.

Carga	Elasticidade	Threshold		Tempo	Energia	
		Superior	Inferior		EME	Empírico
Crescente	x	x	x	4261	289.43	8522
	✓	70	30	1869	395.88	13428
	✓		50	1858	395.96	13284
	✓	90	30	2965	341.28	10404
	✓		50	3088	341.93	10418
Constante	x	x	x	4277	291.37	8554
	✓	70	30	1883	399.28	13422
	✓		50	1914	399.21	13440
	✓	90	30	2730	348.53	10794
	✓		50	2737	348.79	10802
Decrescente	x	x	x	4286	290.86	8572
	✓	70	30	1929	451.49	15930
	✓		50	2787	353.75	11032
	✓	90	30	1926	434.64	15132
	✓		50	2761	349.19	10844
Onda	x	x	x	4296	291.30	8592
	✓	70	30	1959	451.17	15888
	✓		50	2053	453.26	15914
	✓	90	30	3050	362.04	11312
	✓		50	3037	359.48	11188

4.3. Análise de Correlação Entre os Modelos de Energia

Tanto o modelo EME quanto o modelo de energia empírica seguem as mesmas tendências de consumo de energia como ilustrado na Figura 6. Embora apresentem diferentes unidades de medida, os comportamentos dos gráficos são semelhantes. Os resultados mostram que o consumo de energia em Joules (obtido com modelo EME) e o modelo empírico (Equação 6) são altamente correlacionados no âmbito das aplicações HPC.

Na Figura 6 (a) a (d) pode-se observar que os modelos diferem em magnitude, mas os picos e as quedas são semelhantes, o fator de suavização é atribuído à granularidade de cada modelo. Também foi avaliada a correlação entre os modelos usando o coeficiente de Pearson [Benesty et al. 2009] que varia de -1 quando há uma correlação negativa perfeita, 0 quando as variáveis não são dependentes e 1 quando existe uma correlação positiva perfeita. Os valores obtidos para todas as cargas analisadas variam em torno de 0,99, indicando uma correlação positiva.

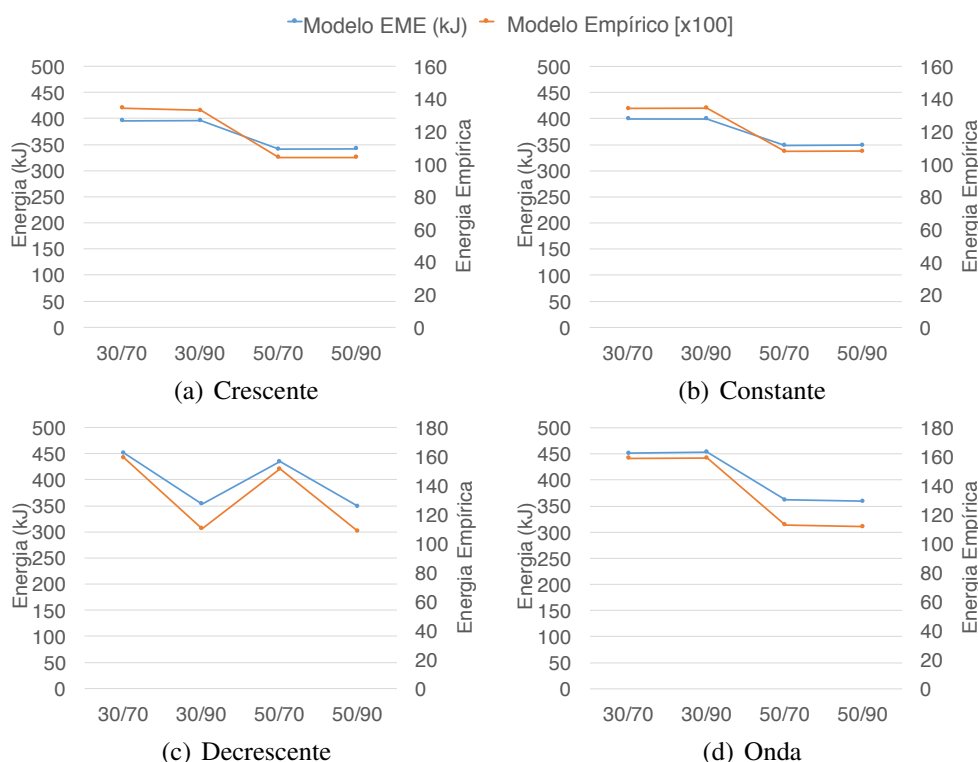


Figura 6. Análise das curvas de consumo do modelo EME e do modelo de energia empírico.

5. Trabalhos Relacionados

Alguns trabalhos concentram-se em modelos para estimar o consumo de energia em ambientes de nuvem, no entanto, estes trabalhos não levam em conta a elasticidade de tais sistemas. Luo et al. [Luo et al. 2013] apresenta um algoritmo de gestão de recursos que considera tanto requisitos de consumo de energia como QoS (Qualidade de Serviço). O artigo apresenta um modelo para prever o consumo de energia dentro de uma única máquina, além de uma estrutura simulada para avaliar algoritmos de escalonamento de recursos que leva em consideração o consumo de energia. Os autores afirmam que na maioria dos estudos de energia de computação em nuvem existentes são utilizados modelos lineares para estimar o consumo de energia, descrevendo a relação entre consumo de energia e utilização de recursos. Garg et al. [Garg et al. 2011] apresenta um modelo de energia do data center com base nos dados de CPU. O modelo apresentado considera todas as CPUs no data center sem considerar a variação dos recursos disponíveis para a aplicação. Com relação a métricas específicas para estimar o consumo de energia, em [Zikos and Karatza 2011], os autores utilizam a seguinte equação para medir a energia: $E = P \times T$. A quantidade de energia utilizada depende da potência e o tempo no qual é utilizada. Assim, E , P e T , denotam energia, potência e tempo, respectivamente. A unidade padrão para a energia é o joule (J), assumindo que a energia é medida em watts (W) e o tempo em segundos (s).

Considerando a análise do consumo de energia, algumas obras focam em definir perfis de energia [Chen et al. 2014], avaliação de custo e desempenho energético [Tsfatsion et al. 2014]. Feifei et al. [Chen et al. 2014] propõe a StressCloud:

uma ferramenta de análise de desempenho e consumo de energia e análise de sistemas em nuvem. Os resultados experimentais demonstram a relação entre o desempenho e o consumo de energia dos sistemas de nuvem com diferentes estratégias de alocação de recursos e cargas de trabalho. No entanto, os autores não abordam nem aplicações paralelas nem elasticidade em nuvem.

Por fim, Tesfatsion et al. [Tsfatsion et al. 2014] realizar uma análise conjunta de custo e desempenho energético utilizando técnicas como DVFS (*Dynamic Voltage and Frequency Scaling*), a elasticidade horizontal e vertical. Esta abordagem combinada resultou em 34% de economia de energia em comparação com cenários onde cada política é aplicada sozinha. Em relação ao consumo de energia, o método tradicional que leva em conta o consumo instantâneo e o tempo é normalmente utilizado. Desta forma, destaca-se o seguinte a respeito das métricas de avaliação: (i) a avaliação do consumo de energia, considerando um número maleável de recursos; (ii) em ambientes elásticos, há uma falta de análise conjunta do consumo de energia e a utilização de recursos para definir os valores para os limites de *thresholds* inferiores e superiores.

6. Conclusão

Este artigo apresentou e avaliou um modelo elástico de consumo de energia para data centers de computação em nuvem. O modelo proposto estima o consumo de energia com base em amostras de CPU e memória com precisão média e mediana 97,15% e 97,72%, respectivamente. Este modelo foi utilizado em conjunto com o *middleware* AutoElastic, que executa aplicativos HPC, alocando e desalocando recursos de acordo com as demandas da aplicação. Os resultados mostraram que os melhores valores para economia de energia foram obtidos quando se utiliza um limite superior (*threshold*) de cerca de 90%, e os piores valores para essa métrica quando se utiliza 70%. Entretanto, neste último caso obteve-se o melhor desempenho.

Focando na reprodutibilidade dos resultados, está presente um conjunto de equações que permite que outros pesquisadores possam empregar o modelo energético proposto para medir o consumo de energia em suas aplicações elásticas. Além do modelo EME, também foi proposto um modelo energético empírico baseado em alocação dinâmica de recursos para fins de comparação. Os resultados revelaram uma forte correlação entre os modelos, onde a curva de consumo de energia segue a mesma tendência, observada principalmente nas operação de alocação e remoção de recursos.

Por fim, trabalhos futuros compreendem a avaliação de consumo energético de *middlewares* para Internet das Coisas os quais são executados com elasticidade, bem como a análise de aplicações irregulares [Schneider et al. 2009].

Referências

- Benesty, J., Chen, J., Huang, Y., and Cohen, I. (2009). Pearson correlation coefficient. In *Noise Reduction in Speech Processing*, volume 2 of *Springer Topics in Signal Processing*, pages 1–4. Springer Berlin Heidelberg.
- Chen, F., Grundy, J., Schneider, J.-G., Yang, Y., and He, Q. (2014). Automated analysis of performance and energy consumption for cloud applications. In *Proceedings of the 5th ACM/SPEC Int. Conf. on Performance Engineering*, ICPE '14, pages 39–50, New York, NY, USA. ACM.

- Comanescu, M. (2012). Implementation of time-varying observers used in direct field orientation of motor drives by trapezoidal integration. In *Power Electronics, Machines and Drives (PEMD 2012), 6th IET Int. Conf. on*, pages 1–6.
- Garg, S. K., Yeo, C. S., Anandasivam, A., and Buyya, R. (2011). Environment-conscious scheduling of hpc applications on distributed cloud-oriented data centers. *J. Parallel Distrib. Comput.*, 71(6):732–749.
- Islam, S., Lee, K., Fekete, A., and Liu, A. (2012). How a consumer can measure elasticity for cloud platforms. In *Proceedings of the 3rd ACM/SPEC Int. Conf. on Performance Engineering*, ICPE '12, pages 85–96, New York, NY, USA. ACM.
- Lorido-Botran, T., Miguel-Alonso, J., and Lozano, J. (2014). A review of auto-scaling techniques for elastic applications in cloud environments. *Journal of Grid Computing*, 12(4):559–592.
- Luo, L., Wu, W., Tsai, W., Di, D., and Zhang, F. (2013). Simulation of power consumption of cloud data centers. *Simulation Modelling Practice and Theory*, 39(0):152 – 171. S.I.Energy efficiency in grids and clouds.
- Orgerie, A.-C., Assuncao, M. D. D., and Lefevre, L. (2014). A survey on techniques for improving the energy efficiency of large-scale distributed systems. *ACM Computing Surveys*, 46(4):1–31.
- Petrides, P., Nicolaidis, G., and Trancoso, P. (2012). Hpc performance domains on multi-core processors with virtualization. In *Proceedings of the 25th Int. Conf. on Architecture of Computing Systems*, ARCS'12, pages 123–134, Berlin, Heidelberg. Springer-Verlag.
- Righi, R., Rodrigues, V., Andre daCosta, C., Galante, G., Bona, L., and Ferreto, T. (2015). Autoelastic: Automatic resource elasticity for high performance applications in the cloud. *Cloud Computing, IEEE Transactions on*, PP(99):1–1.
- Roloff, E., Birck, F., Diener, M., Carissimi, A., and Navaux, P. (2012). Evaluating high performance computing on the windows azure platform. In *Cloud Computing (CLOUD), 2012 IEEE 5th Int. Conf. on*, pages 803 –810.
- Schneider, J., Gehr, J., Heiss, H. U., Ferreto, T., Rose, C. D., Righi, R., Rodrigues, E. R., Maillard, N., and Navaux, P. (2009). Design of a grid workflow for a climate application. In *Computers and Communications, 2009. ISCC 2009. IEEE Symposium on*, pages 793–799.
- Tesfatsion, S., Wadbro, E., and Tordsson, J. (2014). A combined frequency scaling and application elasticity approach for energy-efficient cloud computing. *Sustainable Computing: Informatics and Systems*, 4(4):205 – 214. Special Issue on Energy Aware Resource Management and Scheduling (EARMS).
- Zikos, S. and Karatza, H. D. (2011). Performance and energy aware cluster-level scheduling of compute-intensive jobs with unknown service times. *Simulation Modelling Practice and Theory*, 19(1):239 – 250. Modeling and Performance Analysis of Networking and Collaborative Systems.