

# Um Monitor de Consumo Energético para Computação em Nuvem na Ferramenta OpenNebula

Raul Dias Leiria<sup>1</sup>, Claudio Schepke<sup>1</sup>, Aline Vieira de Mello<sup>1</sup>, Dalvan Griebler<sup>2</sup>

<sup>1</sup>Universidade Federal do Pampa (UNIPAMPA)  
Alegrete – RS – Brasil

<sup>2</sup>Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)  
Porto Alegre – RS – Brasil

rdleiria@gmail.com, {claudioschepke, alinemello}@unipampa.edu.br  
dalvan.griebler@acad.pucrs.br

**Abstract.** *Computational clouds consume a lot of energy and are responsible for causing the global emission of at least 2% of carbon dioxide. Current cloud management tools do not have resources for monitoring the energy consumption of their infrastructures as well as any information on electricity demand, which is an integral part of the cloud's maintenance cost. Therefore, our paper proposes a model for monitoring the electrical consumption in computational clouds. We created an add-on named Monitor Energético (ME) for monitoring energy consumption in data centers virtualized with Kernel-based Virtual Machine and managed by OpenNebula. The experiments were performed using Sysbench tool to stress our environment, where results proved our tool works well and has an intuitive monitoring visualization.*

**Resumo.** *Nuvens computacionais consomem grandes quantidades de energia elétrica, sendo responsáveis pela emissão de pelo menos 2% do dióxido de carbono mundial. Por outro lado, as ferramentas de gerenciamento de nuvens atuais não dispõem de recursos para monitorar o consumo energético de suas infraestruturas, bem como não oferecem informações sobre a demanda de energia elétrica, sendo este uma parte integrante do custo de manutenção da nuvem. Portanto, este artigo propõe um modelo de monitoramento energético para nuvens computacionais. Foi criado um addon denominado Monitor Energético (ME) para monitorar o consumo energético em data centers virtualizados, através de Kernel-based Virtual Machine e geridos por OpenNebula. Os experimentos foram realizados com a ferramenta Sysbench para o estresse proposital do ambiente de testes, aonde os resultados comprovaram o bom funcionamento da nossa ferramenta e que ela possui uma visualização intuitiva do monitoramento.*

## 1. Introdução

A computação em nuvem visa o acesso, a hospedagem e gerência remota de infraestruturas e serviços independente da localidade geográfica dos envolvidos. Sua popularidade é consequência da boa consolidação de seus modelos de serviço (IaaS, PaaS, SaaS) e de seus modelos de implantação (nuvens públicas, híbridas, comunitárias e privadas)

[Vogel et al. 2016]. As nuvens computacionais em sua maioria são compostas por sistemas virtualizados, o que facilita a manutenção, migração, expansão e níveis de abstração diferenciados para quem as utiliza.

Há três possíveis perspectivas para a forma de interação com as nuvens computacionais. Na primeira tem-se o usuário que acessa o recurso *online*, na segunda a empresa que contrata ou que é dona da nuvem e por fim, tem-se a perspectiva do administrador do *data center* (nuvem). Das diversas atribuições do administrador, destaca-se para este trabalho a de monitorar os recursos computacionais [Roveda et al. 2015].

A virtualização é um recurso de *software* que permite executar computadores virtuais em computadores físicos, e por essa característica também contribui na quantidade de energia elétrica consumida pelo *data center*. Enquanto há *hardware* específico para monitorar o consumo energético de computadores físicos, não há nenhum recurso para monitorar o de computadores virtuais (máquinas virtuais) [Bourdon et al. 2011]. Uma solução para isso é utilizar modelos energéticos (matemáticos) que estimem o consumo energético das máquinas virtuais através de dados coletados do kernel do sistema operacional.

Na literatura não foram encontrados trabalhos que abordassem formas de monitorar o consumo energético de máquinas virtuais. Em vista disso, o objetivo deste artigo é propor um modelo de monitoramento de consumo energético. Foi desenvolvido um *addon* (estensor de funcionalidades) que supre essa necessidade e exibe os respectivos dados energéticos na interface *Web* do OpenNebula *Sunstone* versão 4.14.2. Para isso cada servidor hospedeiro possui um coletor de dados energéticos, onde os dados são enviados para o gerenciador de nuvem via sistema distribuído e posteriormente exibidos graficamente.

Este artigo está dividido em 7 seções. A Seção 2 discute os trabalhos relacionados. Na Seção 3 é apresentado o modelo proposto. A implementação do modelo está descrita na Seção 4. As funcionalidades do Monitor Energético e sua implantação são apresentadas na Seção 5. A seção seguinte traz a validação do *addon* implementado. Por fim, a última seção apresenta as conclusões e trabalhos futuros.

## 2. Trabalhos Relacionados

Os trabalhos relacionados que constam, num primeiro momento, foram encontrados na biblioteca digital Google Scholar, utilizando as seguintes palavras-chave: *data center*, *power*, *virtual machine*, *energy*, *cloud*, *metric*, *Kernel-based Virtual Machine*, *OpenNebula*. Num segundo momento, com o propósito de refinar a escolha dos trabalhos em relação às técnicas ou ferramentas, foram priorizados os seguintes critérios:

1. Realizar de forma específica medições em máquinas virtuais, processos ou nuvem;
2. Independente de arquitetura computacional;
3. Ser *open source* e estar disponível para *download*;
4. Possuir compatibilidade com o Kernel-based Virtual Machine (KVM);
5. Haver praticidade na instalação e execução, de modo que haja independência de *softwares* extras nos sistemas operacionais hóspedes;
6. Possuir uma maneira de totalizar os consumos energéticos dos hospedeiros, Máquinas Virtuais (VMs) e *data center*;
7. Possuir integração com alguma ferramenta de gerenciamento de nuvem, preferencialmente como o OpenNebula [OpenNebula Community 2016];

## 8. Disponibilizar uma forma de exportar os dados estimados.

Nem sempre foi possível encontrar em todos os trabalhos as especificidades demarcadas para a pesquisa bibliográfica. Por isso, a Tabela 1 apresenta os trabalhos selecionados e indica os critérios atendidos pelos mesmos, onde ‘+’ representa atendimento ao critério, ‘NA’ indica o não atendimento e ‘NC’ quando nada consta sobre o mesmo. O resumo de cada trabalho é apresentado a seguir.

[Kansal et al. 2010] apresenta a ferramenta Joulemeter para sistemas operacionais hospedeiros Microsoft Windows. A ferramenta é genérica e possibilita medir o consumo energético de processos e máquinas virtuais. A obtenção dos dados de consumo de energia ocorre por meio de modelos energéticos, que utilizam como parâmetro as estatísticas de uso dos recursos do sistema; cada consumo pode ser atribuído a sua respectiva VM ou a um processo. A Joulemeter está disponível para *download*, independe de *hardware* e não é *open source*.

[Bertran et al. 2012] propõe uma metodologia que estima o consumo energético de máquinas virtuais e é específica para *hosts* Linux. Não há necessidade de especificidade de *hardware*. Os dados energéticos são obtidos por meio da ferramenta Linux *pfmon*, que lê os *performance counters* referentes à Unidade de Processamento Central (CPU) e memória, e os trabalha em um modelo energético.

[Marcu et al. 2011] traz um método aplicado à solução de virtualização de VMware [VMWare Inc. 2015]. Os dados energéticos são obtidos por um computador extra que está ligado via Universal Serial Bus (USB) a um medidor de consumo de energia chamado Watts up [Watts up? 2015]. Essa dependência de *hardware* externo torna a proposta inviável em larga escala.

[Wen et al. 2013] apresenta uma solução que é específica para o hipervisor L4 [Fiasco 2015]. Os consumos de energia das VMs são estimados por um modelo energético que infere os dados a partir de *performance counters*. Estes coletam as estatísticas de uso da memória e da CPU em baixo nível. Para isso utilizam a ferramenta Linux *perf*. Segundo o autor do artigo, o consumo energético de disco e de rede para a maioria dos servidores é praticamente pequeno e estático, não sendo tão considerável em uma estimativa final.

[Bourdon et al. 2011] apresenta PowerAPI [Spirals Research Group 2016], uma ferramenta que mede o consumo energético de processos do sistema operacional hospedeiro Linux. Apesar da sua não especificidade para ambientes virtualizados, é possível utilizá-la para medir o consumo de energia de máquinas virtuais de KVM. Os dados para geração do consumo energético podem vir de meios externos como medidores de

**Tabela 1. Comparativo dos Trabalhos Relacionados**

Trabalho	1	2	3	4	5	6	7	8
[Bertran et al. 2012]	+	+	NC	+	NC	NC	NC	NC
[Marcu et al. 2011]	+	+	NC	NA	NC	NC	NC	NC
[Wen et al. 2013]	+	NC	NC	NA	NC	NC	NC	NC
[Kansal et al. 2010]	+	NC	NC	NA	NC	NC	NC	NC
[Bourdon et al. 2011]	+	+	+	+	+	NA	NA	+

potência que interceptam o consumo de energia do servidor físico, como também a partir dos *performance counters*. Os dados coletados são aplicados no modelo energético, que irá gerar o consumo de energia do processo, ou de um grupo de processos, com base nas informações de disco, memória, rede e CPU. A ferramenta em questão está disponível para *download* e utiliza uma abordagem modular, facilitando possíveis novas implementações. A interoperabilidade da ferramenta possibilita utilizá-la via shell ou como uma biblioteca de funções. Diferentemente de outras ferramentas, que precisam de intervenção para calibrar os modelos energéticos, a PowerAPI possui uma autocalibração, fazendo com que essa solução seja ideal em larga escala.

PowerAPI, dentre as ferramentas analisadas nos trabalhos realizados, é a que menos possui limitações e é a que mais está apta a ser implantada em cenários reais. Para tanto é necessário apenas estender algumas de suas funcionalidades, para que a mesma seja utilizável em ambientes virtualizados e possa interagir com o gerenciador de nuvem OpenNebula. A próxima seção mostra como PowerAPI se encaixa no modelo proposto.

### 3. Modelo Proposto

A maioria dos recursos para estimar o consumo energético em *data centers* está focado no *hardware*, desconsiderando os cenários reais onde a virtualização é predominante. A arquitetura de software proposta neste artigo está exposta na Figura 1.

De acordo com o modelo, os servidores *hosts* hospedam máquinas virtuais que podem ou não conter diferentes sistemas operacionais hóspedes. Cada máquina virtual é monitorada através de seu Process Identifier (PID) pela ferramenta PowerAPI. Com isso, a ferramenta retorna dentro de intervalos de tempo o respectivo consumo energético em  $\mu\text{Wh}$ . O sistema de arquivos de rede em conjunto com o sistema proposto neste artigo encarrega-se de tornar disponíveis os dados energéticos dos *hosts* para o gerenciador de nuvem, onde os dados são processados e exibidos na interface *Web* de OpenNebula.

A próxima seção descreve como foi feita a implementação para avaliar a viabilidade do modelo.

### 4. Implantação do Modelo

Os servidores hospedeiros utilizados são máquinas Dell com 8GB de Random Access Memory (RAM) e processador Intel i3. Antes de qualquer instalação, foi necessário

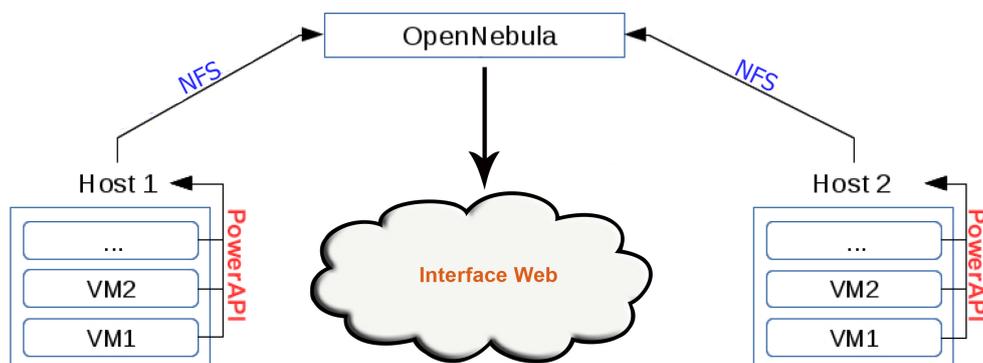


Figura 1. Esquema de Funcionamento do Addon de Monitoramento Energético

habilitar a tecnologia de virtualização na Basic Input and Output System (BIOS) dos servidores, visto que essa é uma prática necessária para o funcionamento do KVM. Para garantir a confiabilidade somada à rapidez de acesso aos dados, deve ser configurado, via utilitário da placa mãe, o Redundant Array of Independent Disks (RAID) na combinação 1+0.

Após as configurações iniciais do *host*, é possível passar para a instalação do sistema operacional hospedeiro.

#### **4.1. Instalação do Sistema Operacional Hospedeiro**

Por questões de estabilidade e vasta documentação, foi escolhido o sistema operacional Linux para o servidor hospedeiro. Durante a instalação, além dos pacotes *default* do sistema, deve-se optar um ambiente sem interface gráfica. A partir do momento em que a instalação do sistema operacional hospedeiro está concluída, é possível avançar para a parte referente às aplicações.

#### **4.2. Instalações, Configurações e Execuções das Aplicações**

Na etapa das aplicações, deve ser instalada a PowerAPI para coleta dos consumos energéticos, o OpenNebula para gerir a nuvem e o libvirt para efetuar a comunicação do OpenNebula com o KVM. O sistema distribuído para comunicação entre os *hosts* e o *front end* é a própria arquitetura Network File System (NFS) provida pelo OpenNebula.

#### **4.3. Viabilidade do Trabalho**

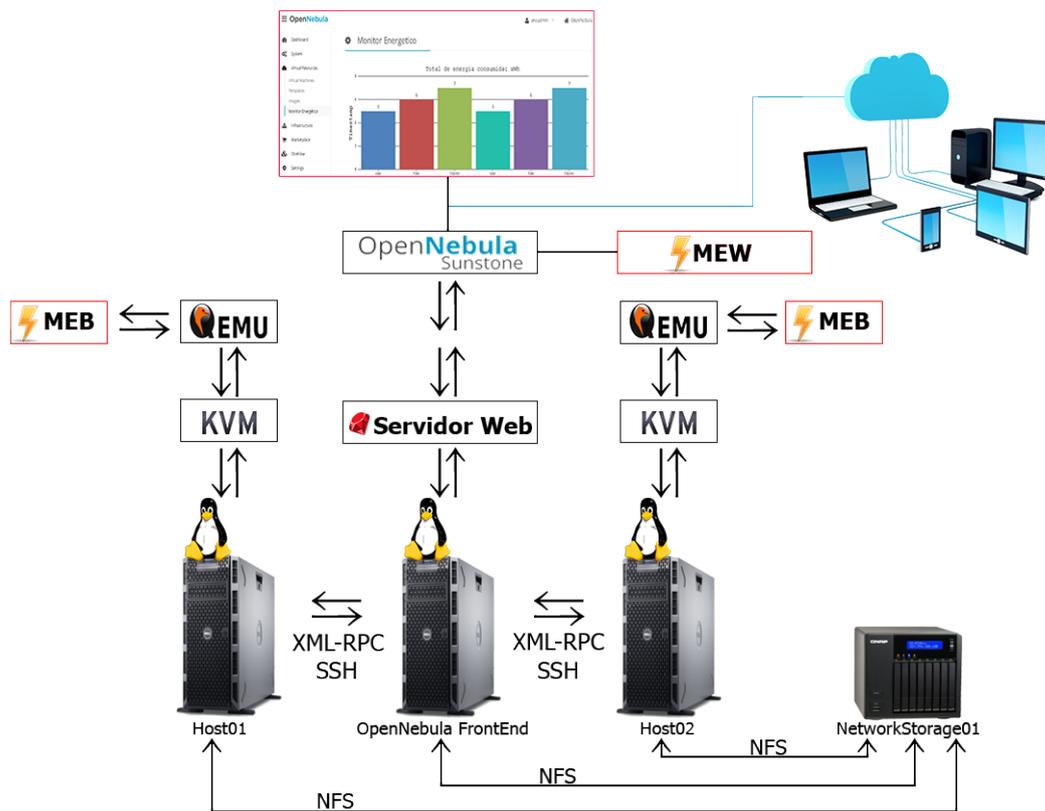
Para verificar a viabilidade técnica do trabalho, foram realizados testes preliminares com as ferramentas KVM, Quick Emulator (QEMU) e PowerAPI em um computador Dell Optiplex 3010, com 4GB de RAM e processador Intel i3. A instalação e execução do KVM e do QEMU foram bem sucedidas, e em razão disso, foi possível instalar uma máquina virtual com sistema hóspede Microsoft Windows XP, e monitorá-la por meio de seu PID.

### **5. Addon Monitor Energético (ME)**

O Addon Monitor Energético (ME) proposto neste trabalho e representado pela Figura 2, possibilita realizar o monitoramento energético de nuvens computacionais geridas pelo OpenNebula. Para isso, há dois módulos de funcionamento; o módulo Monitor Energético em Bash (MEB) é o responsável por coletar dados energéticos nos *hosts* e transmití-los ao *frontend*; já o módulo Monitor Energético em Web (MEW) é uma interface Web adicionada ao OpenNebula Sunstone para que seja possível visualizar os dados coletados em formato de gráfico. A comunicação dos dois módulos se dá por meio da própria infraestrutura do OpenNebula.

#### **5.1. Instalação do Monitor Energético**

O arquivo ME.bz2 deve ser descompactado no diretório temporário do Linux. Da descompactação irão resultar os arquivos instalador.bash (instalador do Monitor Energético), monitor.bash (Monitor Energético em Bash), sunstone-server.rb (Configuração do servidor Web), monitor.erb (Monitor Energético em Web), main.js (gerador JavaScript da interface Web do OpenNebula Sunstone).



**Figura 2. Monitor Energético (ME)**

A instalação do Monitor Energético se dá pelo *script* `instalador.sh`. Ele deve ser executado no *frontend* do OpenNebula e suas funcionalidades são fazer *download* da PowerAPI, oferecer permissão de execução ao *script* `monitor.sh`, copiá-lo para o diretório `/var/lib/one/remotes/im/kvm-probes.d` e sincronizar o *frontend* com os *hosts* por meio do comando `onehost sync -force`. Esse comando copia remotamente os arquivos do diretório `/var/lib/one/remotes` do *frontend* para o diretório `/var/tmp/one` dos *hosts*.

## 5.2. Monitor Energético em Bash (MEB)

Esse módulo está representado pelo *script* `monitor.bash`. O *script* é executado automaticamente pelo *scheduler* de OpenNebula em intervalos de tempo pré-definidos e a sua saída é escrita no arquivo `/var/lib/one/dados.txt`. O diretório `/var/lib/one` é disponibilizado via NFS em todos os servidores do *cluster* do OpenNebula.

O `monitor.bash` quando está executando cria um arquivo de *lock* para que os valores das estimativas energéticas não sejam eventualmente alterados, já que os *hosts* do *cluster* executam o *script* e escrevem no arquivo `dados.txt` quase simultaneamente. Se o arquivo de *lock* não existe, o *script* continua sua execução e verifica quais máquinas virtuais estão em funcionamento no *host*. As informações referentes aos processos das VMs em execução são escritas no arquivo `/tmp/VMsRODANDO`.

Um laço de repetição lê as informações do arquivo `/tmp/VMsRODANDO` e para cada VM lida (cada iteração), o identificador de processo e o nome da máquina virtual são armazenados nas variáveis `PIDVM` e `NOMEVM`, respectivamente. As duas variáveis posteriormente são passadas por parâmetro para a função `monitorEnergético`. A função

é chamada junto de um `&` para que a cada interação ela seja executada em paralelo sem atrapalhar a continuidade do laço. Se isso não fosse feito, cada máquina virtual lida deveria esperar o tempo de execução da função para que a próxima interação do laço pudesse ser feita; após as chamadas à função `monitorEnergetico` há um `wait` para sincronizá-las.

A função `monitorEnergetico` primeiramente executa a ferramenta `PowerAPI`. Os parâmetros passados para a ferramenta são definidos em constantes no começo do *script*, sendo elas o módulo de monitoramento `procfs-cpu-simple`, o arquivo temporário de escrita em `/tmp/PIDVM.tmp` e a duração de cerca de 18s. A execução da ferramenta irá gerar uma saída no arquivo temporário a cada segundo, sendo que cada linha contém informações referentes ao *timestamp* de execução e ao consumo energético instantâneo do processo para aquele *timestamp*.

Após coletar os dados referentes ao consumo energético do processo, há um laço de repetição dentro da função `monitorEnergetico` que retém o valor energético de cada linha do arquivo `/tmp/PIDVM.tmp`, o divide por 3600 para obter Wh e o adiciona numa variável para que ocorra o somatório dos valores encontrados em todas as linhas do arquivo. Como os valores são relativamente pequenos, para melhor exibição no gráfico da interface Web eles são convertidos para  $\mu\text{Wh}$ . A função `monitorEnergetico`, antes de chegar no fim de sua execução, chama a função `escreveArquivo` para que o valor obtido seja escrito no arquivo `/var/lib/one/dados.txt`.

A função `escreveArquivo` utiliza o formato JSON para escrever no arquivo `dados.txt` (utilizado pelo `OpenNebula`) e o formato CSV para escrever no arquivo `dados-export.csv` (possibilidade de importação em outros *softwares*). O formato JSON foi adotado ao invés do XML em razão de compatibilidade com o *framework* `CanvasJS` [Fenopix 2016] utilizado para plotar os gráficos na interface Web do `OpenNebula`.

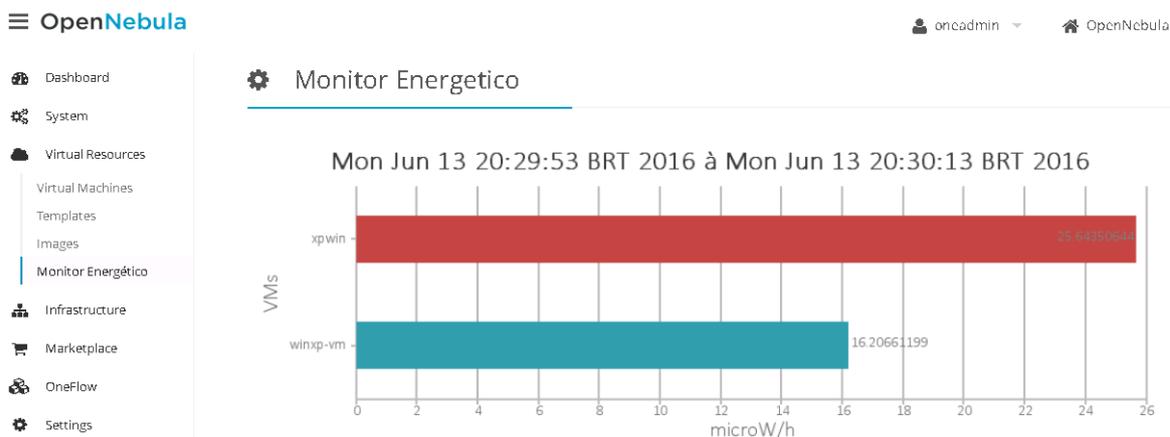
O último passo do *script* `monitor.bash` é remover o arquivo de *lock* para que uma nova instância do mesmo possa ser iniciada. Feito isso, os dados estarão prontos para serem lidos pelo Monitor Energético em Web.

### 5.3. Monitor Energético em Web (MEW)

O `OpenNebula` possui vários desafios quanto a modificação de sua parte Web. Para começar o servidor Web utilizado é um caso atípico onde ao invés de ser usado o Apache [The Apache Software Foundation 2016] é executado um servidor Web via *script* desenvolvido com a linguagem de programação Ruby. Em razão disso e por restrição do *script* `sunstone-server.rb`, qualquer nova página HTML precisa possuir a extensão `erb` e deve ser previamente liberada para exibição na interface Web.

Além disso, não há execução de um interpretador *server-side* como o PHP [The PHP Group 2016] no `OpenNebula`, quando há a necessidade de execução de uma *query* SQL, por exemplo, é necessário fazer uma chamada ao servidor Web para que o mesmo execute a SQL previamente definida em arquivos do servidor Web. Essa restrição junto ao fato de que há duas possibilidades de uso de sistemas de gerenciamento de banco de dados no `OpenNebula`, induziu a implementação deste sistema a não trabalhar com banco de dados e sim com arquivos de texto.

O *iframe* inserido na página apontada pelo menu garante a exibição de outra página denominada `monitor.erb` conforme mostrado na Figura 3. Essa página



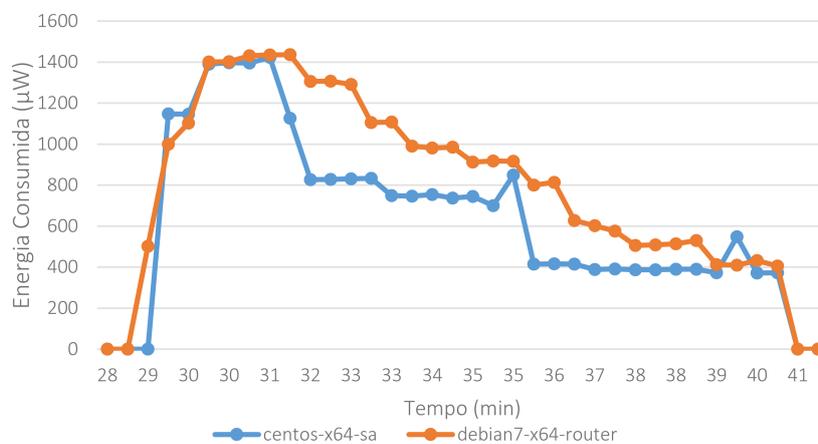
**Figura 3. Entrada referente ao Monitor Energético no menu do Sunstone**

contém códigos HTML e JavaScript que importam os dados energéticos do arquivo `/var/lib/one/dados.txt` e os exibem em um gráfico de barras gerado pelo *framework* CanvasJS. O arquivo `dados.txt` é processado pelo método `parseJSON` do JQuery [jQuery Foundation 2016] e os dados resultantes são inseridos em um vetor temporário, que por sua vez será copiado para o objeto `data` do gráfico. Feito isso, o gráfico pode ser renderizado através do método `chart.render()`.

O gráfico é atualizado via JQuery em intervalos de dois segundos, podendo esse valor ser modificado se necessário.

## 6. Experimento e Validação do Addon Proposto

Para validar o *addon* proposto neste trabalho realizou-se uma estimativa com o Monitor Energético em Bash em duas máquinas virtuais. Estas máquinas executaram por cerca de dez minutos o *benchmark* Sysbench [Alexey Kopytov 2016]. O propósito desse teste foi realizar um estresse no processador para comprovar o aumento do consumo de energia elétrica nos minutos analisados. A Figura 4 apresenta os resultados do monitoramento do consumo de energia.



**Figura 4. Gráfico dos consumos energéticos das máquinas virtuais centos7-x64-sa e debian-x64-router**

É possível observar que a partir da metade do minuto vinte e oito o *benchmark* é inicializado nas duas máquinas virtuais e ocorrem picos no consumo de energia elétrica. Os valores nos minutos seguintes mantêm-se em alta em relação aos valores anteriores ao início da execução do *benchmark*. Desta forma, através da comprovação das alterações energéticas, o modelo foi validado.

## 7. Conclusão

Diferentes recursos costumam ser considerados e gerenciados em nuvens computacionais. Um aspecto importante e atualmente não encontrado em ferramentas de gerenciamento de nuvens IaaS *open source* é o consumo de energia elétrica. Diante disso, este trabalho propôs uma solução para o monitoramento do consumo de energia em nuvens computacionais. Foi desenvolvido para este fim, um *addon* para realizar o monitoramento do consumo energético de máquinas virtuais do KVM para a ferramenta OpenNebula. O *addon* serve de auxílio aos administradores de nuvem para mostrar o quanto de energia elétrica está sendo consumida pelas máquinas virtuais.

Através do experimento com a ferramenta Sysbench foi possível constatar o correto funcionamento do Monitor Energético tanto em momentos de uso excessivo de recursos de *hardware* quanto em momentos de pouca utilização. Como trabalhos futuros, pretende-se implementar tomadas de decisões em relação às estimativas energéticas obtidas, podendo ser realizadas através de políticas automatizadas. Pretende-se também estender as funcionalidades do Monitor Energético ao gerenciador de nuvem OpenStack.

## 8. Agradecimentos

Este trabalho foi realizado com recursos de fomento CNPq Edital Universal Processo N. 457684/2014.

## Referências

- Alexey Kopytov (2016). Sysbench in Launchpad. <https://launchpad.net/sysbench>.
- Bertran, R., Becerra, Y., Carrera, D., Beltran, V., González, M., Martorell, X., Navarro, N., Torres, J., and Ayguadé, E. (2012). Energy accounting for shared virtualized environments under dvfs using pmc-based power models. *Future Gener. Comput. Syst.*, 28(2):457–468.
- Bourdon, A., Nouredine, A., Rouvoy, R., and Seinturier, L. (2011). Linux: Understanding Process-Level Power Consumption. Green Computing Middleware (GCM'2011).
- Fenopix (2016). CanvasJS: Beautiful HTML5 JavaScript Charts. <http://canvasjs.com>.
- Fiasco (2015). L4Re – L4 Runtime Environment. <http://l4re.org/>.
- jQuery Foundation (2016). jQuery Foundation. <https://jquery.org/>.
- Kansal, A., Zhao, F., Liu, J., Kothari, N., and Bhattacharya, A. A. (2010). Virtual machine power metering and provisioning. In *Proceedings of the 1st ACM Symposium on Cloud Computing*, SoCC '10, pages 39–50, New York, NY, USA. ACM.

- Marcu, M., Tudor, D., and Fuicu, S. (2011). Power consumption and temperature measurement of virtualization solutions. In *Thermal Investigations of ICs and Systems (THERMINIC), 2011 17th International Workshop on*, pages 1–6.
- OpenNebula Community (2016). OpenNebula, Flexible Enterprise Cloud Made Simple. <http://opennebula.org/>.
- Roveda, D., Vogel, A., and Griebler, D. (2015). Understanding, Discussing and Analyzing the OpenNebula and OpenStack's IaaS Management Layers. *Revista Eletronica Argentina-Brasil de Tecnologias da Informacao e da Comunicacao*, 3(1):15.
- Spirals Research Group (2016). PowerAPI. <http://www.powerapi.org>.
- The Apache Software Foundation (2016). Welcome! - The Apache HTTP Server Project. <https://httpd.apache.org/>.
- The PHP Group (2016). PHP: Hypertext Preprocessor. <http://php.net/>.
- VMWare Inc. (2015). VMware Virtualization for Desktop and Server. <http://www.vmware.com/>.
- Vogel, A., Griebler, D., Maron, C. A. F., Schepke, C., and Fernandes, L. G. (2016). Private IaaS Clouds: A Comparative Analysis of OpenNebula, CloudStack and OpenStack. In *24rd Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pages 672–679, Heraklion Crete, Greece. IEEE.
- Watts up? (2015). Watts up? Pro. <http://www.wattsupmeters.com>.
- Wen, C., Long, X., Yang, Y., Ni, F., and Mu, Y. (2013). System power model and virtual machine power metering for cloud computing pricing. In *Intelligent System Design and Engineering Applications (ISDEA), 2013 Third International Conference on*, pages 1379–1382.