

<https://periodicos.utfpr.edu.br/recit>

## Ambiente para a criação automatizada de clusters virtuais

### RESUMO

O uso da virtualização, especialmente em ambientes de alto desempenho, tem-se tornado comum devido as suas diversas vantagens, seja no custo para manutenção, ou na facilidade de gerenciamento. Porém, configurar um ambiente desta natureza pode não ser algo muito trivial, principalmente aos usuários com pouca experiência. Neste trabalho, é proposto um ambiente que visa facilitar esse processo, permitindo que usuários iniciantes possam também usufruir desta tecnologia. O ambiente apresentou um comportamento estável, disponibilizando no final do processo um conjunto de máquinas virtuais funcionais para o usuário.

**PALAVRAS-CHAVE:** Clusters Virtuais, Virtualização, Aplicações.

## INTRODUÇÃO

As grande quantidade de aplicações desenvolvidas nos últimos anos trouxe à tona problemas de compatibilidade e dependências para que sua execução seja possível. Muitas vezes, uma única aplicação pode requerer um leque muito grande de bibliotecas ou até mesmo outras aplicações para sua execução. Desta forma, como o usuário nem sempre utiliza de um sistema que contém todas as dependências necessárias à execução de uma determinada aplicação, configurar um ambiente funcional para estas pode demandar horas de trabalho, que poderiam ser dedicadas a outras tarefas.

Com o surgimento das tecnologias de virtualização juntamente com a evolução dos ambientes e ferramentas disponíveis para a Computação em Nuvem, tem surgido diversas soluções no que diz respeito à abstração dos recursos de hardware. Com o uso mecanismos de virtualização presentes nos ambientes de nuvem, a pilha de software, o sistema operacional, as bibliotecas, as aplicações e outros dados do ambiente virtual podem ser encapsulados em uma única imagem de máquina virtual. Estas imagens podem ser facilmente transferidas para outras máquinas físicas, salvas para uso posterior ou copiadas para permitir a execução de réplicas de um ambiente (KELLER, 2011).

Neste contexto, o objetivo deste trabalho é criar um processo automatizado que, dado determinados requisitos e características introduzidos pelo usuário, crie uma máquina virtual, ou um conjunto de máquinas virtuais, capaz de proporcionar um ambiente funcional e pronto para uso. Para tal, foi implementado um ambiente onde o usuário, por meio de um arquivo de configuração, informa os parâmetros desejados, e utilizando *scripts* e algumas ferramentas de virtualização, são criadas as máquinas virtuais desejadas. Este ambiente desenvolvido mostrou-se estável nos testes realizados, disponibilizando no fim de suas etapas de execução um conjunto de máquinas virtuais totalmente funcionais ao usuário.

Este trabalho está organizado da seguinte forma. Na Seção 2 apresenta-se as técnicas de virtualização e seu funcionamento; Na Seção 3 contextualiza-se o paradigma dos clusters aliados à Computação em Nuvem; na Seção 4, apresenta-se o ambiente proposto, assim como seu funcionamento; Na Seção 5, são dados alguns detalhes da fase de implementação, assim como as ferramentas utilizadas para tal; Na Seção 6, é apresenta-se os testes realizados e o comportamento do

ambiente proposto; A Seção 7 conclui o trabalho e apresenta as possibilidades de trabalhos futuros envolvendo o ambiente desenvolvido.

## **VIRTUALIZAÇÃO**

A computação em nuvem surgiu como um novo paradigma de computação que visa fornecer ambientes de computação seguros, personalizáveis e sempre disponíveis. Isso se deve principalmente ao fato de que a Computação em Nuvem simplifica muitos problemas de gerenciamento, reduz custos com manutenção e pode muitas vezes resolver problemas de versionamento de software (WANG, 2008). Outra vantagem que pode ser citada é a possibilidade de se ofertar infraestrutura como serviço, provendo grande capacidade de processamento por um determinado período. Esta alternativa tem atraído a atenção de empresas, universidades e instituições de pesquisa, pois desta forma, não se torna necessário um investimento em estrutura física de alto desempenho, o que muitas vezes apresenta um alto custo financeiro (OLIVEIRA, 2010).

Esse paradigma é proveniente do aperfeiçoamento das técnicas de virtualização. A virtualização é a tecnologia que permite executar dois ou mais sistemas operacionais simultaneamente em um único PC, por meio de máquinas virtuais (NATIONAL INSTRUMENTS, 2011). Sendo assim, é adotado um software que cria uma camada de abstração e funciona como Monitor de Máquina Virtual (MMV) ou *hypervisor*. Esse software, traduz as instruções do sistema convidado para o sistema hospedeiro. Na Figura 1, é feita uma comparação entre um sistema virtualizado (direita) a um não-virtualizado (esquerda). Note que no sistema virtualizado, existe uma camada de abstração entre o hardware e as máquinas virtuais.

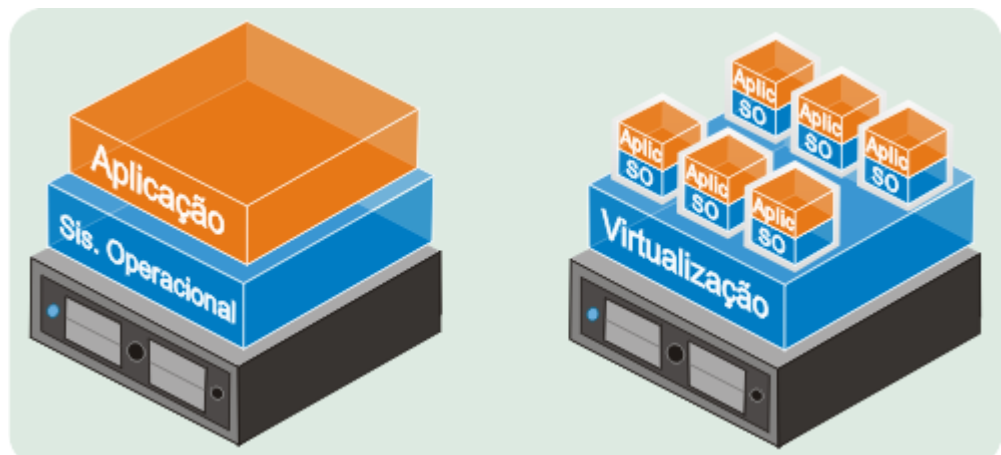
Outro ponto que deve ser ressaltado é a questão de como o *hypervisor* é implementado em relação ao sistema operacional hospedeiro e ao hardware. Na literatura em geral, são consideradas duas diferentes formas de virtualização com relação a este aspecto: a virtualização total e a paravirtualização.

A virtualização total consiste em prover uma réplica (virtual) do hardware subjacente de tal forma que o sistema operacional e as aplicações possam executar como se estivessem diretamente sobre o hardware original (CARISSIMI, 2009).

Uma das vantagens desta forma de virtualização é que o sistema hóspede não necessita de alterações para que seja executado sobre a camada de abstração.

Já na paravirtualização, se tem o objetivo de melhorar o desempenho das máquinas virtuais executadas. Desta forma, “remove-se” de certa forma o sistema operacional hospedeiro, substituindo-o por um *hypervisor* que roda diretamente sobre o hardware. Nessa abordagem, o sistema hóspede deve ser modificado para chamar a máquina virtual sempre que for executar uma instrução ou ação considerada sensível. Na prática isso traduz pela necessidade de alterar todas as instruções de sistema do hóspede por chamadas a máquina virtual para que ela interprete e emule essas ações de forma adequada. Isso é a principal desvantagem da paravirtualização (CARISSIMI, 2009).

Figura 1 – Esquema de Funcionamento das MVs.



Fonte: DevMedia (2016)

### CLUSTERS VIRTUAIS E APLICAÇÕES PRÁTICAS

A Computação de Alto Desempenho (HPC, do inglês *High Performance Computing*) tem buscado alternativas que visam solucionar o problema de se processar grandes quantidades de dados de forma rápida e menos custosa economicamente. Utilizando o paradigma da Computação em Nuvem e sua possibilidade de abstrair recursos de hardware e encapsular softwares em imagens de máquina virtual, vários trabalhos têm estudado os benefícios de empregar esta técnica para solucionar os problemas da HPC. Porém, utilizando unicamente a virtualização, as opções ficam limitadas ao hardware de uma única máquina física.

Em um cenário real, depara-se com situações onde os recursos de processamento estão disponíveis em um conjunto de máquinas diferentes. Neste contexto, como forma de ampliar os recursos utilizáveis, pode-se utilizar as diversas máquinas para a construção de *clusters*.

Um *cluster* é um conjunto de máquinas, que geridas por um frontend, tem o objetivo de se comunicar entre si, compartilhando poder de processamento, para assim concluir uma tarefa num espaço de tempo menor do que se essa mesma tarefa fosse realizada por uma única máquina. Desta forma, torna-se possível alocar máquinas virtuais em diferentes *hosts* físicos mantendo uma comunicação entre cada uma dessas máquinas, distribuindo as tarefas a serem realizadas entre os nodos criados. Esse tipo de *cluster* é chamado de *cluster* virtual.

A viabilidade, benefícios e desempenho dos *clusters* virtuais, bem como suas aplicações práticas é abordada em diversos trabalhos na literatura. Em (MAUCH ET AL, 2011), é feito uma revisão do estado-da-arte da Computação em Nuvem aliada à HPC, apresentando o modelo HPC2, que oferece o controle do usuário sobre seus dados de forma segura, mesmo em uma nuvem pública, uma característica importante para empresas e instituições que mantêm propriedade intelectual. Já em (MONTERO ET AL, 2009), é apresentada uma arquitetura para ambientes em nuvem visando a elasticidade de máquinas em ambientes HTC (*High Throughput Computing*), onde se tem grande fluxo de dados. Além disso, esta arquitetura permite a criação de *clusters* virtuais com nodos alocados em máquinas locais e em nuvens públicas, como a *Amazon EC2*. Um outro exemplo de ferramenta é a *Viteraas*, que permite a criação de *clusters* virtuais utilizando recursos locais e nuvens públicas, identificando e utilizando os recursos ociosos. Esse ambiente, apresentado em (DOELITZSCHER ET AL, 2011), tem seu princípio de funcionamento baseado no modelo de PaaS (*Plataform as a Service*), podendo assim ser comercialmente utilizável.

Assim como visto no último trabalho citado, os *clusters* virtuais tem-se tornado também um atrativo no meio comercial. Por meio dela, empresas que utilizam de recursos computacionais mas não veem necessidade de adquirir máquinas físicas de melhor desempenho, o que pode se tornar um investimento de alto custo, pagam para serviços que oferecem esses recursos, cobrando por tempo de uso. Um exemplo deste tipo de negócio é a *Amazon EC2*. Nesta nuvem pública, são oferecidos recursos de processamento por um determinado preço, o

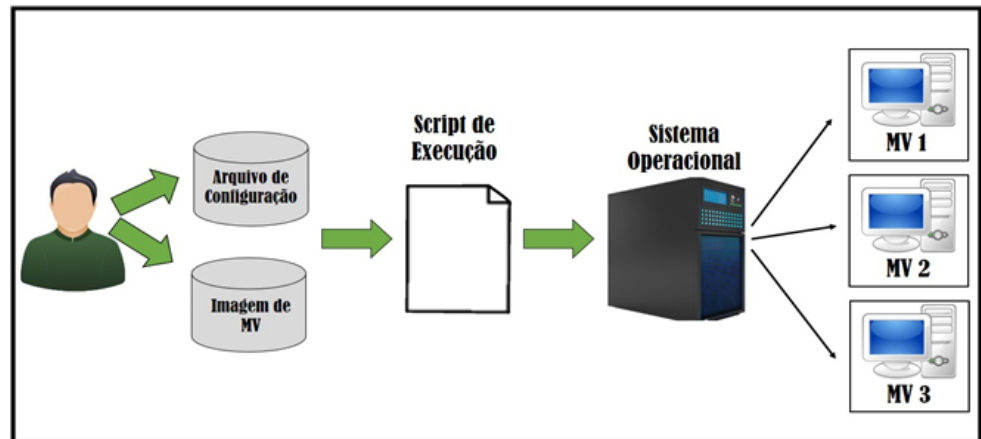
qual depende do tempo de utilização e do tipo de instância escolhidos pelo cliente (AMAZON, 2016).

### **SOLUÇÃO PROPOSTA**

Mesmo com as facilidades oferecidas pela Computação em Nuvem, criar e configurar um ambiente de máquinas virtuais pode demandar, além de conhecimento das ferramentas utilizadas para tal fim, um tempo de configuração que inclui várias horas de trabalho. Neste contexto, automatizar o processo de criação mantendo o controle do usuário sobre os recursos que serão disponibilizados ao término do processo se torna altamente atrativo aos olhos de desenvolvedores que trabalham com estas tecnologias.

Sendo assim, o ambiente proposto neste trabalho realiza esta automatização, utilizando como entrada para criação das máquinas virtuais um arquivo de configuração, que determinará as características físicas destas. Também como alternativa pode ser usado como entrada uma imagem pré-configurada de máquina virtual, que será replicada para instanciar os nodos no final do processo. Este arquivo de configuração ou imagem de máquina virtual é manipulado por um script de execução, que faz a criação e alocação do espaço das máquinas virtuais. Por fim, depois do término da execução deste script, estas máquinas são disponibilizadas para o usuário. Na Figura 2, mostra-se o esquema de funcionamento do ambiente.

Figura 2 – Etapas de funcionamento do ambiente.



Fonte: Autoria Própria.

Como pode ser visto na Figura 2, o usuário deve entrar com um arquivo de configuração, além da possibilidade de utilizar uma imagem de MV para que seja replicada. Após essa etapa, é executado o *script*, o qual envia as informações inseridas pelo usuário para o sistema hospedeiro, que por sua vez, cria e prepara as MVs desejadas, que após o período de criação, ficam disponíveis ao usuário.

## IMPLEMENTAÇÃO

Para a implementação do ambiente, utilizou-se um conjunto de ferramentas. Como *hypervisor* foi escolhido o KVM (*Kernel Virtual Machine*). Este é o software que permite a abstração de recursos e fornece às máquinas virtuais interfaces similares à interface real de hardware (ARAÚJO, 2015).

Para manipulação das máquinas virtuais, foi escolhido a API LibVirt. O LibVirt é uma biblioteca que constrói uma camada de abstração no que diz respeito ao gerenciamento de máquinas virtuais para vários *hypervisors*. Foi escrito em linguagem C e tinha inicialmente o foco na administração através do XEN, posteriormente foi estendido para diversos *hypervisors*, sendo um deles o KVM. É incluso no LibVirt uma biblioteca de drivers, um *daemon* (libvirtd), e um utilitário de linha de comando (virsh) (CARVALHO, 2014).

Com a finalidade de automatizar o processo de entrada dos dados, foi utilizado um arquivo de configuração (.INI). Esse arquivo tem como principal vantagem a organização dos dados por blocos. Logo, foi desenvolvido um script

que, de acordo com cada campo informado no arquivo de configuração, cria a MV com as configurações desejadas. Na Figura 3 temos um exemplo de entrada do arquivo de configuração. Note que cada campo especifica uma determinada característica da MV. É importante ressaltar ainda que foi usado, para reconhecimento dos campos do arquivo de configuração no script, um *parser* chamado de Bash INI File Parser<sup>1</sup>.

Figura 3 - Exemplo de entrada do arquivo de configuração.

```
1  [Virtual_Machine]
2  NAME= Teste
3  RAM= 512
4  VCPU= 2
5  CPuset= auto
6  CDROM= /home/user/Downloads/ubuntu.iso
7  OSTYPE= linux
8  BOOT= cdrom,hd,menu=on
9  DISK= path=/home/user/Documentos/
10 DISK_SIZE= ,size=8
11 GRAPHICS= vnc
12 CONNECT= qemu:///system
13 VIRTTYPE= qemu
14 DESCRIPTION= VMcriadaparateste
15  [Cluster]
16  nodes= 2
17
```

Fonte: Autoria Própria.

Com relação ao processamento dos dados e envio das instruções ao sistema operacional hospedeiro, o script de execução implementado utilizou-se de uma aplicação denominada Virt-Tools, que fornece algumas ferramentas para facilitar a criação e visualização das máquinas virtuais. Mais detalhes sobre essa aplicação podem ser vistos em (IBM, 2016). Quando escolhida a utilização de entrada dos dados de hardware da máquina virtual no formato .INI, o ambiente executa um script que passa os parâmetros informados à uma funcionalidade disponível no Virt-Tools chamada Virt-Install, que aloca e disponibiliza as máquinas virtuais ao usuário, como pode ser visto na Figura 4. Já quando o usuário escolhe a opção de entrar com uma imagem existente de máquina virtual, são utilizados comandos do utilitário de console do LibVirt, o virsh, para replicar e instanciar estas réplicas, disponibilizando-as ao usuário ao fim do processo.



<sup>1</sup><http://coding.tinternet.info>

Figura 4 – Script de criação da MV.

```

1  #!/bin/bash
2
3  . read_ini.sh
4  read_ini config.ini
5
6  j=1
7  for i in ${0...${INI_Cluster_nodes}};
8  do
9      name="${INI_Virtual_Machine_NAME}$j"
10     let j=$j+1
11     disk=${INI_Virtual_Machine_DISK}$name".img"${INI_Virtual_Machine_DISK_SIZE}
12     virt-install \
13     --name $name \
14     --ram ${INI_Virtual_Machine_RAM} \
15     --vcpu ${INI_Virtual_Machine_VCPU} \
16     --cpuset ${INI_Virtual_Machine_CPUSSET} \
17     --cdrom ${INI_Virtual_Machine_CDROM} \
18     --os-type ${INI_Virtual_Machine_OSTYPE} \
19     --boot ${INI_Virtual_Machine_BOOT} \
20     --disk $disk \
21     --graphics ${INI_Virtual_Machine_GRAPHICS} \
22     --connect ${INI_Virtual_Machine_CONNECT} \
23     --virt-type ${INI_Virtual_Machine_VIRTTYPE} \
24     done
25

```

Fonte: Autoria Própria.

## RESULTADOS E DISCUSSÕES

Para realização dos casos de teste foi utilizado uma máquina física com 2 GB de memória RAM, processador Intel Dual Core com dois núcleos de 1.6 GHz, e HD de 160 GB e sistema operacional Linux Ubuntu 14.04 LTS, com *kernel* 3.13.0-48. É importante ressaltar que para realizar testes com qualquer *host* físico, previamente deve estar instalado na máquina o *hypervisor* (KVM), assim como a API de gerenciamento (LibVirt).

No exemplo exibido na Figuras 3, temos duas máquinas virtuais, cada uma delas contendo 512 MB de memória RAM, processadores de dois núcleos, e disco virtual de 8GB. Ainda, pode ser especificado a arquitetura de processador, hierarquia de execução do *boot*, e modo gráfico a ser utilizado. O processo de criação incluindo o tempo de alocação do disco virtual para a MV, dura cerca de cinco minutos na máquina usada nos testes. Este tempo pode variar dependendo da configuração da máquina física utilizada e do tamanho do disco de máquina virtual.

Na Figura 4, temos como resultado as MVs criadas pelo caso teste. Note que as MVs criadas são numeradas de 1 a  $n$ , onde  $n$  é o número de nodos desejados no *cluster* (Teste1 e Teste 2, no exemplo).

Quanto à forma de acesso do usuário as máquinas criadas, este pode ser feito tanto via interface gráfica utilizando a ferramenta Virt-Viewer, também disponibilizada pelo Virt-Tools, quanto via SSH (*Secure Shell*), visto que o endereço de IP destas, necessário para este tipo de conexão, pode ser facilmente obtido configurando um script de envio deste endereço ao usuário para executar junto a inicialização das máquinas virtuais.

Figura 4 – Caso de Teste.



```
virsh # list --all
Id      Name          State
-----
2       Teste1        running
3       Teste2        running
```

Fonte: Autoria Própria.

## CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Neste trabalho desenvolveu-se um ambiente onde facilita-se o processo de criação de *clusters* virtuais. Nos resultados obtidos, foi possível observar que o processo de criação do ambiente de máquinas virtuais foi reduzido em dois passos: preparar o arquivo de entrada seguindo a normalização necessária ao ambiente ou entrar com a imagem de máquina virtual pré-configurada, e realizar a execução do script que prepara e disponibiliza o conjunto de máquinas virtuais ao usuário. Este processo, se realizado sem a ajuda de tal ambiente, pode demandar várias horas, sendo necessário instalar e configurar as ferramentas de virtualização, além de instanciar e contextualizar cada máquina para se tornar um nodo do *cluster* virtual.

Outro ponto que deve ser lembrado é que o ambiente está em fase de protótipo, necessitando de algumas melhorias e a adição de funcionalidades. Como trabalho futuro, pode ser citada a criação de uma interface gráfica amigável ao usuário, pois o processo atual ainda envolve comandos do terminal *Linux*. Outra possibilidade de continuação é a configuração automática do ambiente para permitir a execução de aplicações MPI (*Message Passing Interface*), além de

possibilitar a alocação de máquinas virtuais em *hosts* físicos remotos, por via de conexão SSH.

# An environment to automatically create virtual clusters

## ABSTRACT

The use of virtualization, especially in high performance computing environments, has become common due to its many advantages, be it at maintenance costs or at the ease of management. However, setting up such environment might not be trivial, especially for beginners. This work proposes an environment that aims to facilitate this process, allowing those with little experience at the area to benefit from this technology. The environment has shown stable behavior, providing, at the end of the process, a set of virtual machines functional to the user.

**KEYWORDS:** Virtual Clusters, Virtualization, Applications.

## REFERÊNCIAS

- ARAÚJO, FRANCISCO RENATO CAVALCANTE. Uso de Ambientes Virtualizados para o Ensino de Infraestrutura de Redes. In: Escola Regional de Informática de Mato Grosso, 6, 2015, Cuiabá.
- BESERRA, DAVID WILLIAMS S.C; SOUTO, SAMUEL CARLOS ROMEIRO AZEVEDO; ANDRADE, MARIEL JOSÉ PIMENTEL DE; ARAÚJO, ALBERTO EINSTEIN PEREIRA DE. Comparativo de desempenho de um cluster virtualizado em relação a um cluster convencional sob condições equipotentes. In: Workshop em Clouds, Grids e Aplicações, 9, 2011, Campo Grande.
- CARVALHO, FABIANO LOSILLA DE; BELLEZI, MARCOS AUGUSTO. Avaliação de Desempenho dos Hypervisors XEN e KVM utilizando Administração Simplificada através do LibVirt. Tecnologias, Infraestrutura e Software. T.I.S. São Carlos, v.3, n.1, p. 88-101, jan-abr 2014.
- FERNANDES, ALMIR DOMINICINI. Avaliação Experimental de Técnicas de Virtualização Através de Balanceamento de Carga em Clusters de Computadores. Rio de Janeiro: COPPE/UFRJ, 2010, p. 1-8.
- KELLER, M.; MEISTER, D.; BRINKMANN, A.; TERBOVEN, C.; BISCHOF, C. eScience cloud infrastructure. In: EUROMICRO Conference on Software Engineering and Advanced Applications, 37, 2011, Oulu, p. 188–195.
- NATIONAL INSTRUMENTS; Introdução à Virtualização; Disponível em: <<http://www.ni.com/white-paper/8708/pt/>>. Acesso em: 18 março 2016.
- OLIVEIRA, DANIEL; OGASAWARA, EDUARDO Is cloud computing the solution for brazilian researchers? International Journal of Computer Applications, 6(8):19–23, 2010.
- WANG, LIZHE; LASZEWSKI, GREGOR VON. Scientific Cloud Computing: Early Definition and Experience. In: IEEE International Conference on High Performance Computing and Communications, 10, 2008, Dalian, China, p. 2-5.
- CARISSIMI, ALEXANDRE. Virtualização: Princípios Básicos e Aplicações. In: Escola Regional de Alto Desempenho, 2009, Caxias do Sul, Brasil, p. 39-69.
- MONTERO, RUBEN S.; MORENO-VOZMEDIANO, RAFAEL; LLORENTE, IGNACIO M. An elasticity model for High Throughput Computing clusters. J. Parallel Distrib. Comput.71, 750-757, 2009.
- MAUCH, VIKTOR; KUNZE, MARCEL; HILLENBRAND, MARIUS. High performance cloud computing. Future Generation Computer Systems 29, 1408–1416, 2011.
- CALATRAVA, AMANDA; MOLTÓ, GERMÁN; ROMERO, ELOY; CABALLER, MIGUEL; DE ALFONSO, CARLOS. Towards Migratable Elastic Virtual Clusters on Hybrid Clouds. In: IEEE 8th International Conference on Cloud Computing, New York, USA, 2015.
- AMAZON; Amazon EC2 – Hospedagem de servidor virtual; Disponível em: <<https://aws.amazon.com/pt/ec2/>>. Acesso em: 05 Agosto 2016.

DOELITZSCHER, FRANK; HELD, MARKUS; REICH, CHRISTOPH. Viteraas: Virtual Cluster as a Service. In: Third IEEE International Conference on Cloud Computing Technology and Science, Athens, Greece, 2011.

IBM; KVM Tools and Utilities; Disponível em: <<http://www.ibm.com/support/knowledgecenter/linuxonibm/laat/laatkvmtools.htm>>. Acesso em: 19 Agosto 2016.

**Recebido:** 20 ago. 2016.

**Aprovado:** 23 nov. 2017.

**DOI:**

**Como citar:** Ambiente para a criação automatizada de clusters virtuais, R. Eletr. Cient. Inov. Tecnol, Medianeira, v. 8, n. 15, 2017. E – 4533.

Disponível em: <<https://periodicos.utfrpr.edu.br/recit>>. Acesso em: XXX.

**Correspondência:**

**Direito autoral:** Este artigo está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.

