

Just-in-Time Clouds

Uma abordagem para Federação de Clouds Privadas

Edigley Fraga¹, Jonathan Brilhante¹, Rostand Costa¹, Francisco Brasileiro¹, Pedro Bignatto², Diego Desani², Hermes Senger², Airtton Pereira³, Vinícius Garcia³, Rodrigo Assad³, Fernando Trinta⁷, Ana Cristina Oliveira⁵, Henryson Chagas⁵, Aleciano Ferreira⁵, Marco Spohn¹, Reinaldo Gomes¹, Philippe Navaux⁴, Eduardo Roloff⁴, Otávio Carvalho⁴, Marcos Barreto⁷, Raimundo Macêdo⁷, Alírio Sá⁷

¹Laboratório de Sistemas Distribuídos – Departamento de Sistemas e Computação –
Universidade Federal de Campina Grande (UFCG)
Av. Aprígio Veloso, s/n, Bloco CO, Bodocongó,
58.429-900, Campina Grande, PB

²Universidade Federal de São Carlos (UFSCar)
São Carlos – SP

³Universidade Federal de Pernambuco (UFPE)
Recife – PE

⁴Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre – RS

⁵Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB)
Campina Grande – PB

⁶Universidade Federal do Ceará
Fortaleza – CE

⁷Universidade Federal da Bahia
Salvador – BA

{edigley,jonathan,rostand.costa,fubica}@lsd.ufcg.edu.br,
ana.oliveira@ifpb.edu.br, {reinaldo,maspohn}@dsc.ufcg.edu.br,
{senger.hermes,junaobignatto,ddesani,aleciano,balrou}@gmail.com,
{faps,vcg}@cin.ufpe.br, fernando.trinta@lia.ufc.br,
{navaux,eroloff,omcarvalho}@inf.ufrgs.br,
{marcoseb,macedo,aliriosa}@ufba.br

Abstract. *This paper describes the Just-in-Time (JiT) Cloud approach for private clouds federation. A JiT Cloud aggregates several geographically distributed JiT Data Centres (JiT DCs), providing on-demand Infrastructure-as-a-Service. The system is comprised of three distributed software components (client, cloud and dc) for the management of the federated infrastructure, the cloud users and admins, the virtual machine images (templates), and the life-cycle of virtual machines. Currently there is a JiT Cloud deployment comprised of six JiT DCs distributed among three Brazilian states (Paraíba, São Paulo e Rio Grande do Sul).*

Resumo. Este artigo descreve a solução *Just-in-Time (JiT) Cloud* para a federação de clouds privadas. Uma *JiT Cloud* consiste na agregação de diversos *JiT Data Centres (JiT DCs)* geograficamente dispersos e provê infraestrutura computacional como serviço (*IaaS*) por meio da alocação sob demanda de máquinas virtuais. A solução é composta por três módulos de software que permitem a gerência da infraestrutura por parte de um administrador e de máquinas virtuais por parte de usuários clientes. Atualmente existe uma *JiT Cloud* em produção composta por seis *JiT DCs* distribuídos em três estados brasileiros (Paraíba, São Paulo e Rio G. do Sul).

1. Introdução

O paradigma de computação na nuvem (*cloud computing*) permite a provisão de infraestrutura de Tecnologia da Informação sob a forma de um serviço (*infrastructure-as-a-service* ou *IaaS*) adquirido sob demanda. Aplicações científicas que processam grandes cargas de trabalho podem potencialmente obter benefícios a partir da elasticidade de *IaaS* para chegar a seus resultados mais rapidamente. Infelizmente, os provedores públicos atuais de *IaaS* precisam limitar a quantidade de máquinas virtuais simultaneamente alocadas para um único usuário [Costa 2011]. Para lidar com esta limitação, Costa propôs o conceito de *Just in Time (JIT) Clouds* [Costa 2013], cujos provedores incorrem em custos de propriedade somente quando os recursos são demandados pelos clientes. Este artigo apresenta um *middleware* para suporte ao conceito de *JiT Clouds*.

O restante do artigo está organizado como segue. A Seção 2 dá uma visão geral do sistema. Na Seção 3, são apresentadas as principais funcionalidades providas, enquanto que a Seção 4 contém uma descrição da arquitetura da solução. Por sua vez, a Seção 5 inclui um roteiro de demonstração das principais funcionalidades. A Seção 6 apresenta as considerações finais e os trabalhos futuros a serem realizados.

2. Visão Geral

Na abordagem de *JiT Clouds*, que objetiva a federação de recursos computacionais amortizados pertencentes a diferentes *data centres*, um *provedor* permite aos *clientes* a alocação sob demanda e de forma transparente de *máquinas virtuais (VMs)*, do inglês *virtual machines*) que serão iniciadas em *máquinas físicas* pertencentes aos fornecedores dos recursos. Pelo uso dos recursos, o *provedor* tarifa o *usuário*, que paga pelos recursos utilizados. Por sua vez, o *provedor* realiza um repasse de parte do valor pago para o *fornecedor* dos recursos amortizados. A automação dessas etapas é realizada pelos diversos componentes que integram uma *JiT Cloud*.

Os componentes mínimos necessários são os seguintes:

- ***JiT Cloud CLI:*** Módulo cliente padrão para acesso à *JiT Cloud*. Consiste em um conjunto de comandos que permite a interação com a interface do *JiT Cloud Middleware* para utilizar os serviços providos pela *JiT Cloud*.
- ***JiT Cloud Middleware:*** O *JiT Cloud Middleware* é a implementação de um coordenador de uso e oferta de infraestrutura-como-um-serviço (*IaaS*, do inglês *Infrastructure-as-a-Service*) baseada na federação de recursos computacionais

amortizados. O *JiT Cloud Middleware* interage com cada agrupamento de recursos através de uma abstração chamada *JiT Data Centre (DC)*.

- ***JiT DC Middleware:*** O *JiT DC Middleware* é o responsável por gerenciar os recursos amortizados que estão sendo federados por uma *JiT Cloud*. Um *JiT DC Middleware* é vinculado a um *JiT Cloud Middleware*, que por sua vez interage com vários *JiT DCs Middleware*.

Ao emitir os comandos do módulo cliente, cada usuário tem acesso apenas às máquinas virtuais que lhe pertencem e sua identificação na *JiT Clouds* é assegurada com base em credenciais. Estas credenciais lhe são fornecidas no momento de seu registro.

De forma semelhante, o administrador da infraestrutura se utiliza do mesmo módulo. No entanto, além do acesso à API cliente, também lhe é disponibilizado o acesso à API administrativa. Por meio desta, pode-se registrar usuários, estabelecer políticas de uso e registrar *JiT DCs*. O registro de *data centres* permite a ampliação da infraestrutura resultante, sendo vital para uma *JiT Cloud*.

Todas as operações realizadas via módulo cliente são fornecidas pelas APIs básicas da *JiT Cloud*, disponibilizadas via *web services (WS)* e acessadas programaticamente pelo cliente padrão destas APIs, implementado no módulo *JiT Cloud CLI*. Outras formas de interação com a *JiT Cloud* que não sejam via linha de comando podem ser implementadas, a exemplo de um portal *web* (atualmente em fase de desenvolvimento) que também funcione como cliente das APIs fornecidas.

O *JiT DC Middleware* é o módulo responsável por lidar com os recursos físicos nos quais as máquinas virtuais são instanciadas, por meio de uma abstração para a *cloud* privada do fornecedor. A comunicação com a *cloud* privada é feita via API *Amazon AWS EC2* [AWS 2013], padrão *de facto* para provisão de *IaaS*. Por sua vez, a *cloud* privada é implantada com base no *middleware* de *cloud Eucalyptus* [Eucalyptus 2013], utilizado como um gerente de recursos dentro de um *JiT DC*. É de sua responsabilidade a abstração de recursos físicos em máquinas virtuais. As soluções de virtualização utilizadas são KVM (*Kernel-based Virtual Machine*) [Kivity 2007] e XEN [Barham 2003]. Embora estes monitores de *VMs* suportem máquinas com sistema operacional Windows, apenas sistemas *Unix-like* são considerados na *JiT Cloud* atualmente.

De forma similar ao *JiT Cloud Middleware*, o *JiT DC Middleware* também expõe duas APIs, uma cliente e outra com funções administrativas. A *JiT DC Client API* é utilizada para as funções básicas de provisionamento de máquinas virtuais. A comunicação entre *JiT Cloud* e *JiT DC* se dá via chamada remota de procedimentos e a autenticidade da troca de mensagens é assegurada via troca de certificados de segurança.

3. Principais Funcionalidades

Nesta seção, são apresentados os casos de uso mais significativos para a operação de uma *JiT Cloud*. Os casos de uso são agrupados em *gerência de infraestrutura*, *registro de usuários*, *registro de imagens de máquinas virtuais* e *gerência do ciclo de vida de máquinas virtuais*. É importante frisar que, no caso da *JiT Cloud*, todas as operações são voltadas para um ambiente federado, envolvendo diferentes domínios administrativos.

3.1. Gerência de infraestrutura

Para gerência da infraestrutura, é vital poder realizar o registro de novos *JiT DCs*, situação necessária para a expansão da nuvem federada. Deve ser possível também realizar a remoção de um *JiT DC*, fazendo com que o *data centre* não seja mais considerado para escalonamentos de instâncias. Para a remoção, são observadas diversas pré-condições, a exemplo de não ter *VMs* em execução.

```
# Para listar os JiT Data Centers que fazem parte da JiT Cloud:
sh bin/jit-describe-dcs

# Para adicionar um novo JiT Data Center à JiT Cloud:
sh bin/jit-register-dc -n <jit_dc_name> \
-e http://<ip>:<port>/services/JiTDCUserAPIService \
-a http://<ip>:<port>/services/JiTDCAdminAPIService

# Para remover um JiT DC da JiT Cloud:
sh bin/jit-remove-dc -n <jit_dc_name>
```

Figura 1 Comandos para gerência da infraestrutura

As operações de registro e remoção são realizadas apenas pelo administrador da *JiT Cloud*, no entanto, a listagem dos *JiT DCs* pode ser realizada por qualquer usuário, pois ele poderá utilizar a informação resultante para especificar em qual *JiT DC* deseja que suas máquinas virtuais sejam iniciadas.

3.2. Registro de usuários

De forma similar à gerência de infraestrutura, as operações envolvendo usuários são tipicamente administrativas. Do ponto de vista funcional, para o registro de um novo usuário basta um identificador (nome) e um *e-mail*. Já do ponto de vista de negócio, outros dados seriam necessários, a exemplo de número de cartão de crédito para posterior tarifação de recursos consumidos. Como resultado, é gerado um par de credenciais para o usuário (*accessKey* e *secretAccessKey*) a serem utilizadas na configuração do módulo cliente.

```
# Para listar todos os usuários da JiT Cloud:
sh bin/jit-describe-clients

# Para registrar um usuário na JiT Cloud:
sh bin/jit-register-client -n <user_name> -e <user_e-mail> [-ssh]

# Para deregistrar um usuário na JiT Cloud:
sh bin/jit-remove-client -n <user_name> -e <user_e-mail>
```

Figura 2 Comandos para gerência de usuários

Também é possível a inativação de usuário (caso em que os registros de uso e de faturação são mantidos, mas o usuário não pode mais iniciar novas *VMs*) e a listagem dos usuários cadastrados. É importante notar que a inativação pode ser um procedimento realizado automaticamente, por exemplo, na adoção de um modelo de tarifação pré-pago.

3.3. Registro de imagens de máquinas virtuais

Antes de se iniciar uma máquina virtual é preciso haver um arquivo *template*, que servirá como o sistema de arquivos básico para a *VM*. Assim como em outras soluções de *IaaS*, numa *JiT Cloud* esses *templates* são denominados *imagens de VMs*. Por exemplo, é o *template* que define se a *VM* será *Debian*, *Ubuntu* ou *CentOS*. Além do sistema de arquivos, outros arquivos são necessários, a exemplo do *kernel* que será utilizado.

```
# Para listar as imagens de máquinas virtuais registradas:
sh bin/jit-describe-images

# Para realizar o upload de uma nova imagem para o JiT Cloud Storage
sh bin/jit-upload-image -i <vm_img> -k <kvm_kernel> -r <kvm-initrd>
                        -a <vm_image_architecture>

# Para registrar na JiT Cloud uma imagem presente no JiT Cloud Storage
sh bin/jit-register-image -e <vm_image_id> -s <vm_image_size> -o <vm_image_os> \
                        -d <vm_image_description> -a <vm_image_architecture>

# Para remover uma imagem da JiT Cloud:
sh bin/jit-remove-image -i <vm_image_id>
```

Figura 3 Comandos para gerência de imagens de vms

Assim sendo, é possível realizar *upload* e registro de imagens de máquinas virtuais. Embora seja uma operação comumente administrativa, nada impede que o próprio usuário realize *upload* e registro de uma imagem para a *JiT Cloud*. Posteriores transferências dessa imagem para os *DCs* participantes, quando necessárias, são realizadas automaticamente pelos componentes *JiT Clouds*. Qualquer usuário pode realizar a listagem das imagens públicas presentes no catálogo de imagens e, no caso de suas próprias, realizar a remoção dessas.

3.4. Gerência do ciclo de vida de máquinas virtuais

Dado que existe infraestrutura disponível, usuário e imagens de *VMs*, pode-se partir para a provisão de recursos computacionais. Para tanto, é possível a instanciação de máquinas virtuais por usuários ativos e devidamente autenticados na *JiT Cloud*. Na requisição, é informada a imagem base, o tipo de *VM* (de acordo com a configuração de *memória*, *disco* e *cpu*) e a quantidade de instâncias a serem provisionadas. Opcionalmente, pode ser informado o *JiT DC* alvo das alocações ou a estratégia de escalonamento. As instâncias são alocadas de acordo com as restrições estabelecidas, podendo ocorrer alocações em diferentes *JiT DCs*. Ao final, são retornados os identificadores de todas as instâncias alocadas.

```
# Para listar instâncias do usuário que estejam executando na JiT Cloud:
sh bin/jit-describe-instances

# Para iniciar novas instâncias na JiT Cloud:
sh bin/jit-run-instances -t <type> -i <vm_image_id> -n <number_of_instances>

# Para terminar instâncias que estejam executando na JiT Cloud:
sh bin/jit-terminate-instances -i <instance_ids>

# Para criar um snapshot de uma máquina virtual:
sh bin/jit-create-snapshot -v <vm_id> -i <instance_ip>
```

Figura 4 Comandos para gerência do ciclo de vida de máquinas virtuais

De posse dos identificadores, que também podem ser obtidos pela listagem de instâncias do usuário, pode-se emitir um comando para encerrá-las ou, ainda, para a criação de um *snapshot*. O *snapshot* (cópia do sistema de arquivo principal) poderá ser utilizado para iniciar uma nova instância com o estado idêntico ao da instância que lhe deu origem.

4. Arquitetura da Solução

A Figura 5 apresenta os componentes e agentes de uma *JiT Cloud*. A arquitetura é composta por três macros componentes: *JiT Cloud Client (Broker e Toolkit)*, *JiT Cloud (Middleware)* e *JiT DC (Middleware)*. O primeiro é responsável pelo acesso do cliente e do administrador à infraestrutura. O macro componente *JiT Cloud* permite a federação de recursos computacionais amortizados acessados por meio de implantações do componente *JiT DC*.

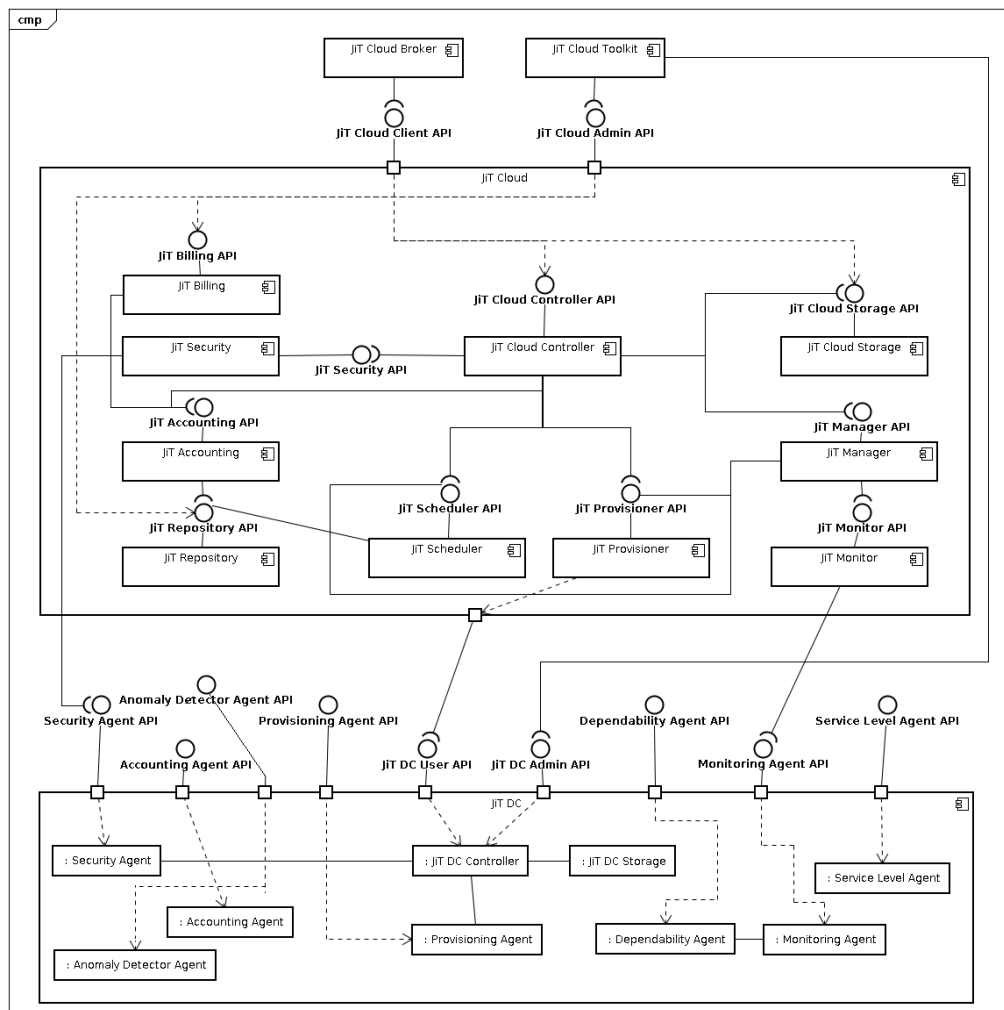


Figura 5 Componentes e Agentes da Arquitetura JiT Clouds

Na *JiT Cloud*, o componente *JiT Cloud Controller* realiza a intermediação entre as solicitações dos usuários e o efetivo cumprimento dessas solicitações, realizado nos recursos físicos existentes nos *JiT DCs*. Para tanto, este controlador se utiliza da funcionalidade especializada provida por componentes plugáveis, a exemplo de componentes de escalonamento, de segurança, tarifação e provisionamento. Por sua vez, em um *JiT DC*, o componente *JiT DC Controller* lida com os recursos físicos e se utiliza da funcionalidade de agentes especializados, tais como agentes de monitoramento, de provisionamento e de qualidade de serviço. Na arquitetura, APIs administrativas lidam com os recursos físicos (servidores), enquanto a API do cliente da *JiT Cloud* trata de entidades lógicas (máquinas virtuais e imagens).

A implementação foi realizada em Java e, para possibilitar uma arquitetura extensível, optou-se por utilizar padrões de integração (EIP - *Enterprise Integration Patterns*), com base no uso da plataforma *opensource Mule ESB* [Mule ESB 2013]. Ainda, para permitir comunicação assíncrona via troca de mensagens entre componentes e agentes, foi utilizado o padrão *AMQP* [AMQP 2013, RabbitMQ 2013]. Embora foque em (*IaaS*), em cima das APIs externas podem ser construídos serviços de alto nível, caracterizados como *Plataforma como Serviço (PaaS)* e *Software como Serviço (SaaS)*.

5. Roteiro de Demonstração das Principais Funcionalidades

O *Cloud Middleware* foi implantado na infraestrutura montada na Universidade Federal de Campina Grande (UFCG), enquanto as implantações do *DC Middleware* foram realizadas na própria UFCG, do Instituto Federal da Paraíba (IFPB), e nas Universidades Federais de São Carlos (UFSCAR) e do Rio Grande do Sul (UFRGS), conforme ilustrado na Figura 6.

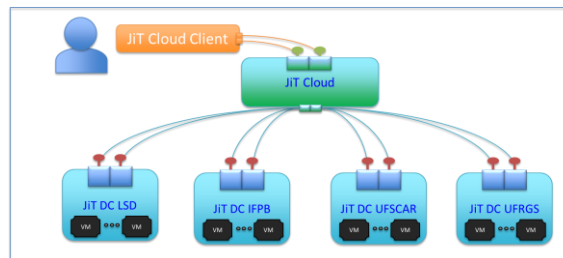


Figura 6 Ilustração da JiT Cloud em Produção

```
wget jitclouds.lsd.ufcg.edu.br/downloads/preconfigured/cloudclient-1.0.zip
unzip cloudclient-1.0-testbed.zip
cd cloudclient-1.0
export JITCLIENT_HOME=`pwd`
cd $JITCLIENT_HOME

sh bin/jit-describe-dcs
jit_dc_lsd      http://150.165.15.131:65082/services/JiTDCUserAPIService
...
jit_dc_ufrgs    http://143.54.85.52:65082/services/JiTDCUserAPIService

sh bin/jit-describe-images
jit-b4420d5c    x86_64      Debian_6      2      OurGrid Worker
...
jit-b4420dfc    i386       Debian_5      2      Snapshot Enabled

sh bin/jit-describe-vm-types
m1.small       1          2          256
...
m1.xlarge      2          20         2048

sh bin/jit-describe-instances
i-33AC41A4     150.165.15.150      jit_dc_lsd      RUNNING        m1.small

sh bin/jit-run-instances -i jit-b4420d5c -t m1.small -n 1 -d jit_dc_ifpb
i-48EE094F

sh bin/jit-describe-instances
i-33AC41A4     150.165.15.150      jit_dc_lsd      RUNNING        m1.small
i-48EE094F     192.168.20.9       jit_dc_ifpb     PENDING        m1.small

sh bin/jit-terminate-instances -i i-33AC41A4

sh bin/jit-describe-instances
i-33AC41A4     150.165.15.150      jit_dc_lsd      SHUTTING_DOWN m1.small
i-48EE094F     192.168.20.9       jit_dc_ifpb     PENDING        m1.small

sh bin/jit-run-instances -i jit-b4420d5c -t m1.small -n 4
i-1CE93B47    i-D6DF3E8A    i-34770623    i-30730629

sh bin/jit-describe-instances
i-33AC41A4     150.165.15.150      jit_dc_lsd      TERMINATED     m1.small
i-48EE094F     192.168.20.9       jit_dc_ifpb     RUNNING        m1.small
i-1CE93B47     150.165.15.148      jit_dc_lsd      PENDING        m1.small
i-D6DF3E8A     150.165.15.145      jit_dc_lsd      PENDING        m1.small
i-34770623     192.168.20.10       jit_dc_ifpb     PENDING        m1.small
i-30730629     192.168.20.12       jit_dc_ifpb     PENDING        m1.small
```

Figura 7 Exemplo de acesso à JiT Cloud

Será disponibilizado um *JiT Cloud Client* já apontando para a infraestrutura da *JiT Cloud* em produção e com credenciais de acesso já configuradas. Para testar o acesso, é necessária uma máquina com sistema operacional *Linux* (ou *MAC OS*) com suporte à plataforma *Java*. A Figura 7 ilustra como o *JiT Cloud Client* pode ser obtido e como pode ser usado para o caso básico do ciclo *run/describe/terminate VMs*.

6. Conclusão e Trabalhos Futuros

Para o desenvolvimento do *JiT Cloud* foram utilizadas tecnologias consolidadas e padrões de projetos de integração, permitindo flexibilidade e extensibilidade por meio de componentes plugáveis em tempo de configuração. Como forma de aumentar a elasticidade da solução, pretende-se evoluir-la para dar suporte a *JiT Clouds* entre-pares (*p2p*), combinando *clouds* federadas e grades entre-pares [OurGrid 2004].

Os produtos de *software* e manuais que fazem parte da Solução *JiT Clouds* podem ser acessados na página <http://jitclouds.lsd.ufcg.edu.br>.

Agradecimentos

Os autores gostariam de agradecer o apoio financeiro do Centro de Pesquisa e Desenvolvimento em Tecnologias Digitais para Informação e Comunicação (CTIC). Hermes Senger agradece à *Fapesp* e ao *CNPQ* pelo apoio recebido.

Referências

- Costa, R., Brasileiro, F., Lemos Filho, G., Sousa, D. (2011) “Sobre a Amplitude da Elasticidade dos Atuais Provedores de Computação na Nuvem”, In: Anais do XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC2011). Campo Grande, MS, Brasil, Maio 2011. Sociedade Brasileira de Computação (SBC).
- Costa, R. (2013) “Just in Time Clouds: Uma Abordagem Baseada em Recursos Terceirizados para a Ampliação da Elasticidade de Provedores de Computação na Nuvem”, Tese de Doutorado, Universidade Federal de Campina Grande, Março 2013.
- Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., and Warfield, A. (2003). Xen and the art of virtualization. In Proceedings of the nineteenth ACM symposium on Operating systems principles, SOSP '03, pages 164–177, New York, NY, USA. ACM.
- AWS (2013). Amazon Web Services (AWS), <http://aws.amazon.com>.
- Eucalyptus (2013). Eucalyptus Home Page, <http://www.eucalyptus.com/>.
- Kivity, A., Kamay, Y., Laor, D., Lublin, U., and Liguori, A. (2007). KVM: The Linux Virtual Machine Monitor. In Ottawa Linux Symposium, pages 225–230.
- OurGrid (2004). OurGrid Community, <http://www.ourgrid.org/>, 2004.
- Mule ESB (2013). Mule Enterprise Service Bus <http://www.mulesoft.org/>, 2013
- RabbitMQ (2013). Rabbit Message Queue, <http://www.rabbitmq.com/>, 2013
- AMQP (2013). Advanced Message Queuing Protocol, <http://www.amqp.org/>, 2013