

Uni4Cloud – Uma Abordagem para Implantação de Aplicações sobre Múltiplas Nuvens de Infra-Estrutura

**Américo Sampaio, Matheus Cunha, Nabor Mendonça, Marcelo Barros,
Thiago Leite, Roberto Costa, Michel Vasconcelos**

Mestrado em Informática Aplicada (MIA) – Universidade de Fortaleza (UNIFOR)
Av. Washington Soares, 1321, Bloco J, Sala 30, CEP 60811-905 Fortaleza, CE, Brazil

{americo.sampaio, mathcunha, nabor, marcelobarrosmia,
robertocostalima, michelav}@unifor.br, thiago.carvalho@serpro.gov.br

Abstract. *Cloud Computing is a computing paradigm that aims at providing on demand access, over the Internet, to a pool of resources (e.g., virtual machines, storage, networks and applications) that can be rapidly provisioned. Clouds offer high scalability and flexibility for using these computing resources which are normally provisioned as services (offered via an API) and charged as utilities such as electricity where users pay for the resources they actually consume (e.g., CPU-Hours used and data stored). This paper presents an approach that allows to model, deploy and configure complex web applications in multiple infrastructure clouds.*

Resumo. *A computação em nuvem é um modelo computacional que permite o acesso, sob demanda e através da Internet, a um “pool” de recursos computacionais (redes, servidores virtuais ou físicos, armazenamento, aplicações e serviços) que podem ser rapidamente provisionados. As nuvens oferecem alta escalabilidade e flexibilidade de uso destes recursos, acopladas a um novo modelo de negócio onde os usuários dessas plataformas pagam apenas pelos recursos (por exemplo, tempo de processamento e volume de dados armazenados) que de fato consumirem (modelo de pagar pelo uso como serviços de Energia Elétrica). Este artigo apresenta uma abordagem que facilita a modelagem, implantação, configuração e distribuição de aplicações web complexas em múltiplas nuvens que oferecem servidores virtuais como recurso, chamadas de nuvens de infra-estrutura (IaaS).*

1. Introdução

A computação em nuvem é um novo paradigma de desenvolvimento, distribuição e implantação de aplicações na Internet que tem recentemente atraído bastante interesse da indústria de software e da comunidade acadêmica [Armbrust et al. (2009), Leavitt 2009, Galán et al. 2009]. Empresas de consultoria renomadas como Merrill Lynch [Lynch 2008] prevêem que o mercado de computação em nuvem terá um potencial de aplicação de 160 bilhões de dólares em 2011. A Gartner [Pring and Desisto 2008, Smith and Austin 2009] por sua vez estima que em 2011 75% dos serviços de software (SaaS) utilizarão algum tipo de infra-estrutura de nuvem. Já a empresa de marketing IDC aponta que o gasto diretamente ligado a investimentos em soluções de computação em

nuvem deve crescer de 16 bilhões de dólares em 2008 para 42 bilhões de dólares em 2012 [Gens 2008].

O interesse da indústria em adotar a computação em nuvem se dá principalmente pelo fato deste modelo permitir utilizar os recursos computacionais de forma mais econômica e otimizada, possibilitando uma redução de custos operacionais (menos máquinas, manutenção simplificada, redução de gastos com energia e resfriamento) [Armbrust et al. (2009)], o que representa um fator extremamente relevante considerando-se o atual cenário de crise econômica mundial e redução de gastos com tecnologia da informação (TI). Esta economia ocorre devido ao fato que empresas clientes das nuvens não precisam investir para adquirir (comprando hardware, coolers, alugando espaço físico) e administrar (contratando pessoal) sua própria infra-estrutura. Ao invés disso, estas empresas podem optar por alugar a infra-estrutura necessária de outras empresas que possuem enormes *data centers* (como Google e Amazon) e, com isso, conseguem atingir economias de escala oferecendo tais serviços a um baixo custo a qualquer cliente. Um exemplo ilustrativo de como este modelo pode ser benéfico é descrito em [Derek 2007]. Este exemplo mostra que o departamento de TI do jornal New York times conseguiu cumprir (usando um software de processamento em alto desempenho baseado no Hadoop executado sobre a plataforma de nuvem da Amazon EC2 [EC2 2009]) uma tarefa de converter parte do seu acervo de reportagens digitalizadas em imagens em formato escaneado “.tiff” para formato PDF, para que fosse integrado e disponibilizado no seu web site. A tarefa foi cumprida em menos de 24 horas e custou milhares de dólares a menos (e precisou de muito menos tempo) do que custaria se o desenvolvimento fosse feito internamente no seu departamento de TI. Caso a equipe de TI do NY times não tivesse usado a plataforma EC2 seria necessário adquirir uma grande quantidade de servidores que depois ficariam ociosos.

Este novo paradigma vem também atraindo atenção de diversos pesquisadores que apontam a computação em nuvem como uma relevante e desafiante área de pesquisa que evolui conceitos já consolidados como virtualização, computação em grade e computação utilitária. Além disto, pesquisadores apontam que a computação em nuvem traz benefícios como escalabilidade de aplicações complexas, maior disponibilidade e tolerância a falhas devido à alta capacidade da infra-estrutura e da sua adaptação dinâmica e sob demanda.

Um dos grandes desafios [Galán et al. 2009, VMware 2009] do atual cenário de computação em nuvem é possibilitar um mecanismo para implantar aplicações complexas que podem ser compostas de vários componentes (e.x. banco de dados, servidor *web*, servidor de aplicação, balanceador de carga) e serem “empacotadas” em uma ou várias máquinas virtuais que podem utilizar infra-estruturas de nuvens distintas de forma fácil e automatizada. Normalmente cada nuvem oferece uma forma proprietária de “empacotar” uma aplicação além de oferecer APIs também proprietárias para realizar a implantação da aplicação.

Este artigo apresenta a abordagem Uni4Cloud que visa facilitar a implantação de aplicações em múltiplas nuvens através da criação de ferramentas e componentes de software implementados com base em padrões abertos e soluções “open source” existentes. Dentre as principais características da abordagem, destacam-se:

- Permite criar um modelo que contém informações sobre os componentes de software que fazem parte de uma aplicação (ex: servidor *web*, servidor de aplicação, banco de dados) e que são distribuídos como uma *virtual appliance* (baseado no padrão *Open Virtualization Format – OVF* [DMTF 2009]) em nuvens híbridas. *Virtual Appliance* corresponde a uma nova forma de “empacotar” um serviço de software através de conjunto de máquinas virtuais (cada uma contendo sua pilha de software específica) e metadados que descrevem diversas informações como, por exemplo: (i) a capacidade de cada máquina virtual (CPU, memória, disco, rede com a qual sua interface está conectada, etc.); (ii) seqüência de startup das máquinas virtuais; (iii) conjunto de pares propriedades-valor a serem configurados em tempo de *boot* das máquinas virtuais. O Padrão OVF é uma especificação aberta (baseada em XML e XML Schema) para criar *virtual appliances* que está sendo suportada pelas grandes empresas na área de computação em nuvem e virtualização como VMware, Citrix, IBM, Sun, Cisco, Microsoft, dentre outras.
- A distribuição de cada componente da aplicação se faz de forma automática, através de chamadas às APIs baseadas em um padrão genérico de API para nuvens chamado *Open Cloud Computing Interface - OCCI* [OCCI 2009]. A implementação funciona como um adaptador que encapsula chamadas correspondentes às APIs proprietárias de nuvens de infra-estrutura distintas.
- Em tempo de inicialização de cada componente, ou seja, quando sua máquina virtual é inicializada, a aplicação tem o poder de se auto-configurar através de scripts (ex: servidor de aplicação configura o IP do banco de dados ao qual ele referencia).

O restante deste artigo é organizado da seguinte forma: a seção 2 apresenta uma visão geral sobre o estado da arte da computação em nuvem; a seção 3 detalha a abordagem Uni4Cloud e seus componentes arquiteturais, os quais são validados através de um exemplo discutido na seção 4; a seção 5 apresenta os trabalhos; por fim, a conclusão e trabalhos futuros são apresentados na seção 6.

2. Visão geral sobre computação em nuvem

A computação em nuvem é um modelo computacional que permite o acesso, sob demanda e através da rede (geralmente na internet), a um *pool* de recursos computacionais (redes, servidores virtuais ou físicos, armazenamento, aplicações e serviços) que podem ser rapidamente provisionados. Algumas características do modelo de nuvem são [Leavitt 2009, Mell and Grance 2009]:

- Atendimento *self-service* e sob demanda. Um consumidor de um serviço de nuvem pode requisitar automaticamente (normalmente usando-se de interfaces de programação ou APIs) um recurso computacional (por exemplo, um servidor virtual ou espaço em um servidor de armazenamento em rede).
- Elasticidade. O consumidor do serviço pode requisitar dinamicamente mais ou menos recursos para sua aplicação para se adaptar à demanda dos seus usuários. Por exemplo, em períodos de pico o consumidor solicita à nuvem mais recursos

computacionais (como, por exemplo, servidores adicionais), podendo depois liberar tais recursos, quando os mesmos não forem mais necessários.

- Pagamento pelo Uso e Garantias de serviço (Service Level Agreements – SLAs). Os consumidores pagam aos provedores de serviço de nuvem de acordo com o consumo efetuado (modelo de pagamento pelo uso semelhante a utilidades como energia e gás). A nuvem possui mecanismos para permitir o compartilhamento dos recursos existentes de forma eficiente para os diversos clientes e monitorar as aplicações e os recursos de forma a respeitar as garantias de serviço oferecidas como, por exemplo, disponibilidade de 99.9%.
- Acesso ubíquo através da rede. Os recursos computacionais podem ser acessados através de padrões (como APIs REST baseadas em HTTP ou SOAP) por diversos tipos de clientes (*browsers*, PDAs, celulares) e seus detalhes de funcionamento e complexidades ficam “escondidos” pela nuvem.

Existem atualmente diversos tipos (ou modelos de entrega [Mell and Grance 2009]) de nuvens. A EC2 [EC2 2009], da Amazon, é caracterizada como infra-estrutura como serviço (IaaS), oferecendo máquinas virtuais de capacidades diferentes a preços variados. Por exemplo, a menor máquina virtual oferecida pela EC2 (*Small Instance*) possui uma CPU de aproximadamente 1,2GHz, memória RAM de 1,7 GB e 160 GB de armazenamento em disco, e custa 10 centavos de dólares por hora. Outros tipos de nuvens conhecidos são plataforma como serviço (PaaS) e software como serviço (SaaS), que oferecem, respectivamente, plataformas de desenvolvimento e aplicações como serviço. Exemplos comerciais de PaaS são as plataformas Google App Engine [Appengine 2009] e Microsoft Azure [Azure 2009], que oferecem ambientes de desenvolvimento de aplicações em nuvem. Um exemplo bem conhecido de SaaS é Salesforce.com [Salesforce 2009], que oferece diversas aplicações (software comerciais como *CRM*, *ERP*, etc.) como serviço na sua própria infra-estrutura de nuvem.

Conforme mencionado anteriormente, a computação em nuvem adota alguns conceitos e abordagens de áreas já consolidadas como a computação em grade (*grid computing*); porém, é importante ressaltar que existem algumas diferenças tais como [Foster et al. 2009]:

- Modelo de negócio: o modelo de computação em nuvem é um modelo de pagar pelo uso similar a utilidades como eletricidade. Os provedores de serviços de nuvem se beneficiam de economias de escala para realizar lucros e oferecer seus serviços a um baixo custo para os clientes. A computação em grade é normalmente um modelo mais voltado para a pesquisa que envolve a cooperação de grandes centros de pesquisa do governo e universidades. A ideia da grade (*grid*) também consiste de compartilhar recursos para que os pesquisadores submetam processos (*jobs*) para resolver problemas computacionais complexos (ex: projetos na área genética ou de previsão de tempo, simulação, etc.). Normalmente a computação em grade oferece tais recursos computacionais para os membros participantes de um determinado projeto.
- Arquitetura e distribuição: normalmente em grades os recursos computacionais compartilhados (e.g., máquinas) podem estar em regiões geográficas e pertencer a instituições diferentes. Isto em geral não ocorre na computação em nuvem,

cujas soluções são normalmente oferecidas pela mesma empresa embora também se possa haver distribuição geográfica (a Amazon, por exemplo, oferece seus serviços na costa leste e oeste dos EUA e na Europa). Outra diferença é que a distribuição de aplicações em nuvem dá-se normalmente através de máquinas virtuais, enquanto que em grades estas normalmente consistem de processos submetidos para execução em *batch*.

- Tipos de aplicação: a computação em grade é normalmente voltada para o processamento *batch* de problemas de alto desempenho, enquanto que as aplicações em nuvem são normalmente voltadas para aplicações comerciais na Internet. A computação em nuvem, em princípio, também pode ser uma alternativa interessante para aplicações de alto desempenho como no exemplo da aplicação do *NY Times* descrita anteriormente.

Um grande desafio dos atuais clientes de nuvens de infra-estrutura é de possuir um mecanismo que facilite a modelagem, implantação e configuração de uma aplicação que possa ser usada em nuvens distintas. Normalmente, cada provedor IaaS possui um mecanismo proprietário para empacotar e implantar os componentes da aplicação em máquinas virtuais que são implantadas com chamadas à APIs proprietárias.

Este artigo mostra uma abordagem para lidar com a distribuição de aplicações complexas sobre nuvens de infra-estrutura que se baseia em padrões e soluções open source que permitem a distribuição dos componentes da aplicação em diferentes nuvens de infra-estrutura.

3. Abordagem Uni4Cloud

A abordagem Uni4Cloud (ver Figura 1) visa facilitar a modelagem, implantação, configuração e distribuição de aplicações em nuvens de infra-estrutura distintas. A abordagem considera três principais *stakeholders*:

- *Provedor de infra-estrutura de nuvem*. Oferecem serviços de “aluguel” de infra-estrutura como máquinas virtuais (contendo CPU, memória RAM, disco e acesso a rede). Exemplos são Amazon EC2, GoGrid e também a nuvem acadêmica da Universidade da Califórnia em Santa Barbara. Na Figura 1 existem dois exemplos de provedores de IaaS (Amazon e nuvem da Empresa X). A abordagem Uni4cloud se foca em integrar ferramentas e software para facilitar a distribuição de aplicações sobre IaaS distintas.
- *Provedor de serviços web*. Utiliza-se de um provedor de IaaS para distribuir sua aplicação web visando uma redução de custos com infra-estrutura. O objetivo do provedor de serviço é oferecer seu serviço (e.g., um site de vendas on-line como o Provedor de Serviços 1 ou um site de Cursos Online como o Provedor de serviços 2) da forma mais eficiente e econômica possível para seus clientes web. O Provedor de Serviços 1, por exemplo, possui um site de vendas cujos componentes de software são distribuídos em nuvens diferentes (Componente 1 na sua nuvem privada e Componentes 2 e 3 na Amazon). Isso pode ser feito da forma mais simplificada possível através do auxílio da ferramenta de configuração do serviço web que permite a modelagem, configuração e

implantação dos componentes da aplicação através da criação de uma *virtual appliance* usando a abordagem Uni4Cloud.

- *Cientes web (usuários)*. Acessam os serviços (através da internet usando *browsers*, PDAs, celulares, etc.) do provedor de serviço (e.g., loja online) sem saber detalhes do seu funcionamento e de onde a aplicação está sendo hospedada. Portanto, para os clientes não importa como o serviço está distribuído (e.g., usando uma ou mais nuvens). Note que o site de vendas online é composto de 3 componentes (máquinas virtuais), sendo que 2 destes componentes contém o servidor web (Apache) e de aplicação (JBoss) e o outro componente possui o banco de dados (MySQL Server)¹. Cada máquina virtual pode conter um sistema operacional distinto e ser distribuída (de forma transparente para os clientes web) em qualquer um dos servidores físicos que fazem parte de uma ou mais nuvens.

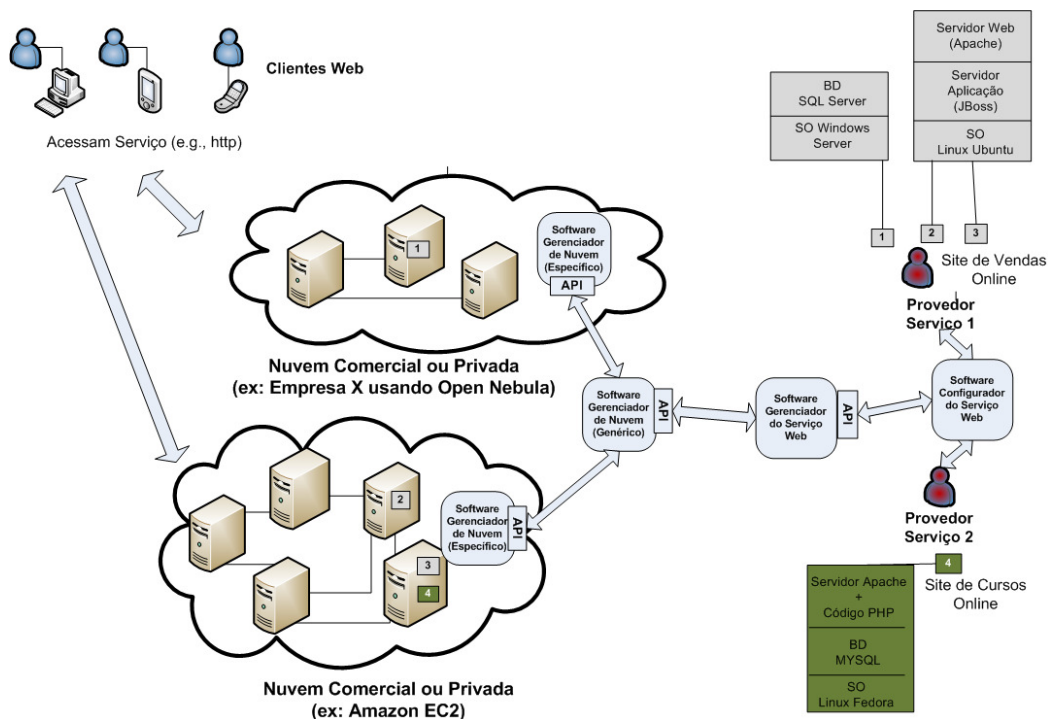


Figura 1 – Abordagem Uni4Cloud

Para dar o suporte mencionado acima, a Abordagem Uni4cloud provê os seguintes componentes² da sua arquitetura:

- *Software Configurador do serviço web*. Esta camada é responsável pela especificação do serviço web e das máquinas virtuais que o compõem. Uma ferramenta visual permite a modelagem da *virtual appliance* especificando as máquinas virtuais necessárias e quais componentes de software estão contidos

¹ O site de vendas online neste exemplo distribui a máquina virtual contendo o servidor de banco de dados na sua nuvem interna por razões de segurança. Ele também distribui duas máquinas virtuais contendo o servidor web e de Aplicação dentro da nuvem da Amazon EC2 e pode se beneficiar do conceito de elasticidade para lidar com a demanda dos usuários balanceando a carga para estes servidores replicados.

² A tecnologia usada para implementar estas camadas de software segue a filosofia de Web Services (REST ou SOAP) e se utiliza de padrões e soluções open source já existentes.

nelas além das suas propriedades de configuração. Esta ferramenta é uma extensão do software *open source* Jasmine Design [Jasmine 2009] que gera a especificação do serviço web baseada no padrão OVF [DMTF 2009] à partir do modelo visual montado da *Virtual Appliance*.

- *Software gerenciador do serviço web*. Esta camada oferece mecanismos para ler a especificação do serviço fornecida pelo configurador e fazer uma “tradução” (i.e., o parsing) para chamadas específicas do software gerenciador de nuvem de forma a distribuir os componentes de software em uma ou mais nuvens de acordo com a especificação. Além disso, esse software permite monitorar o serviço web, ou seja, a aplicação do cliente que é executada em uma ou mais máquinas virtuais da nuvem. Este serviço também oferecerá mecanismos que poderão auxiliar a aplicação a tomar decisões dinamicamente, como, por exemplo, fazer o *start up* de mais um servidor de aplicação devido ao aumento da carga do sistema (devido a um pico de uso, por exemplo).
- *Software gerenciador de nuvem (genérico)*. Este software funciona como um adaptador que se baseia na API genérica de software gerenciador de nuvem OCCI e traduz as chamadas feitas para o gerenciador de nuvem específico correspondente. Na prática, são implementados vários adaptadores (um adaptador para cada nuvem IaaS distinta). A vantagem de usar estes adaptadores é tornar o software gerenciador de serviço web fracamente acoplado a uma nuvem IaaS específica.
- *Software gerenciador de nuvem (específico)*. Este software gerencia e controla a alocação das máquinas virtuais nos servidores físicos pertencentes à nuvem IaaS. É importante ressaltar que este software **não é** um componente implementado pela Uni4Cloud, mas sim por um provedor IaaS qualquer (ex: Amazon EC2) que oferece uma API proprietária com diversas funcionalidades de gerenciamento do ciclo de vida das máquinas virtuais (*start, shutdown, describe, etc.*).

A Figura 2 mostra uma visão geral de um cenário básico de implantação de um serviço web em uma ou mais nuvens com suporte da abordagem Uni4Cloud.

O *stakeholder* provedor do serviço web utiliza-se do **configurador de serviço web** para modelar e configurar a *Virtual Appliance* conforme mostrado na Figura 3. O configurador irá permitir montar a *Virtual Appliance* através de componentes reutilizáveis (e.g., uma aplicação baseada em um cluster de servidores de aplicação J2EE contendo dois servidores JOnAS e um servidor web Apache configurado também para balanceamento de carga e o banco de dados) e gerar a especificação OVF correspondente (um arquivo XML que pode conter referências a outros arquivos como os arquivos de imagem de cada máquina virtual).

Uma vez que o serviço esteja configurado, o usuário (no caso o provedor de serviços web da Figura 1) pode distribuir seu serviço invocando a API de distribuição de serviço e que recebe a especificação OVF do serviço como parâmetro para o **gerenciador de serviços web**. Este irá fazer o parsing da especificação OVF (do arquivo XML) e traduzir isso em chamadas correspondentes à inicialização das máquinas virtuais de acordo com a sua especificação OVF. Isso será feito para todas as máquinas virtuais contidas na especificação OVF e se traduzirá em chamadas a API do Gerenciador de Nuvem genérico.

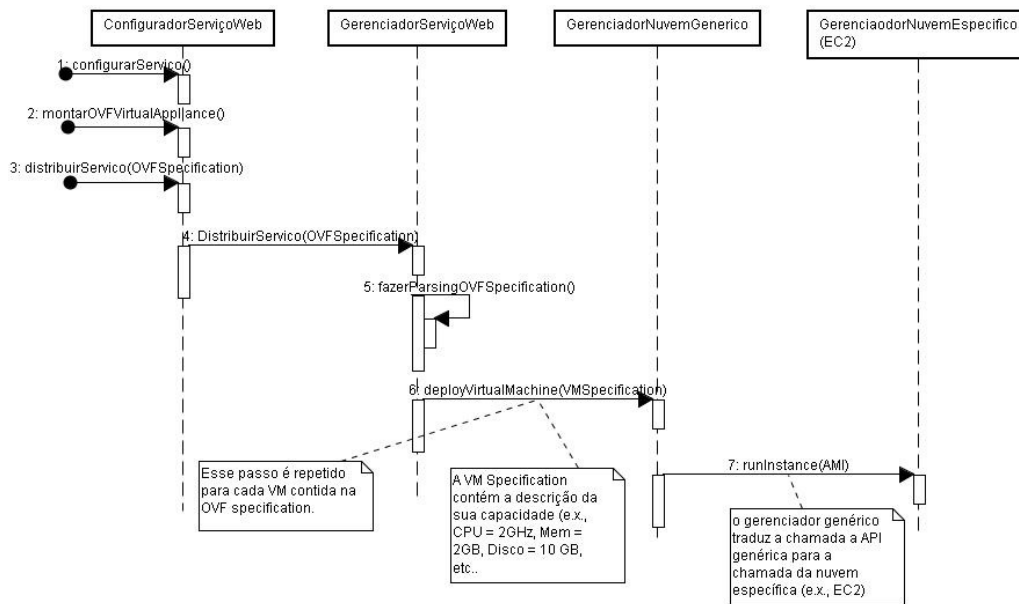


Figura 2 - Visão geral da implantação de um serviço web

O gerenciador de nuvem genérico se baseia na especificação do padrão OCCI [OCCI 2009] e representa um adaptador genérico para traduzir as chamadas para uma determinada nuvem específica em tempo de execução. Isto é importante, pois as máquinas virtuais contidas dentro da especificação OVF podem ser distribuídas em nuvens distintas. Uma vez que o gerenciador de nuvem genérico obtenha a informação de qual(is) nuvem(s) específica(s) será(ão) usada(s), ele irá fazer a chamada correspondente como, por exemplo, chamar o serviço *runInstance()* do gerenciador de nuvem específico da EC2 (ver Figura 2).

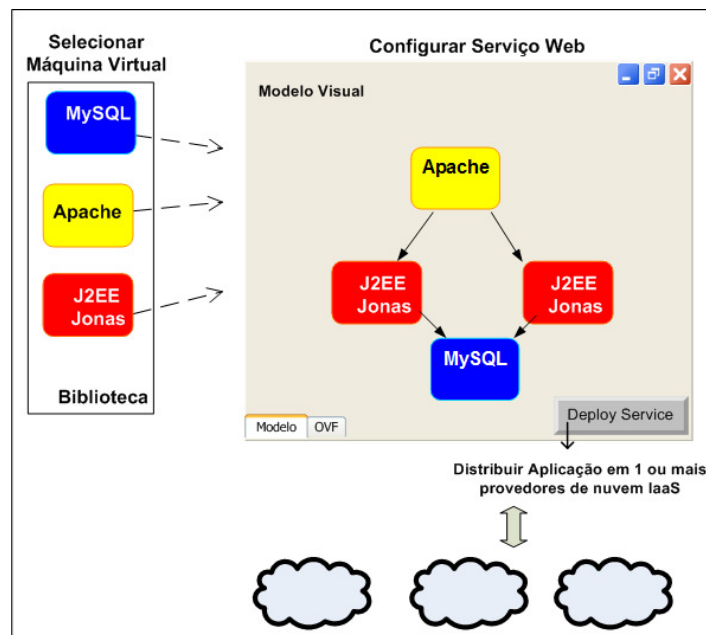
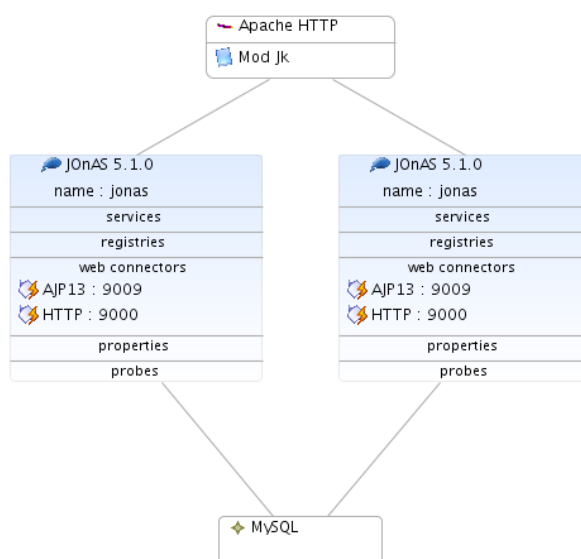


Figura 3 - Ferramenta de configuração e modelagem do serviço web

4. Utilização da Abordagem

A abordagem Uni4Cloud foi utilizada na prática com o objetivo de investigar como ela pode dar suporte à implantação automática de aplicações web em nuvens distintas. A aplicação web escolhida foi uma *Virtual Appliance* para comércio eletrônico, composta por quatro máquinas virtuais: uma com o servidor web *Apache*, que também foi configurado como balanceador de carga, duas executando o servidor de aplicação *Java EE JOnAS 5* em *cluster* e a última como servidor de banco de dados *MySQL*, conforme ilustrado na Figura 4. A razão da escolha desta arquitetura se deve ao fato de ser um perfil de configuração bastante comum para aplicações web em cluster como sites de comércio eletrônico. O mesmo cenário de validação poderia ter sido realizado com outras tecnologias (ex: servidor de aplicação JBoss, banco de dados Oracle, etc.).

Figura 4 – Virtual Appliance do cluster Jonas



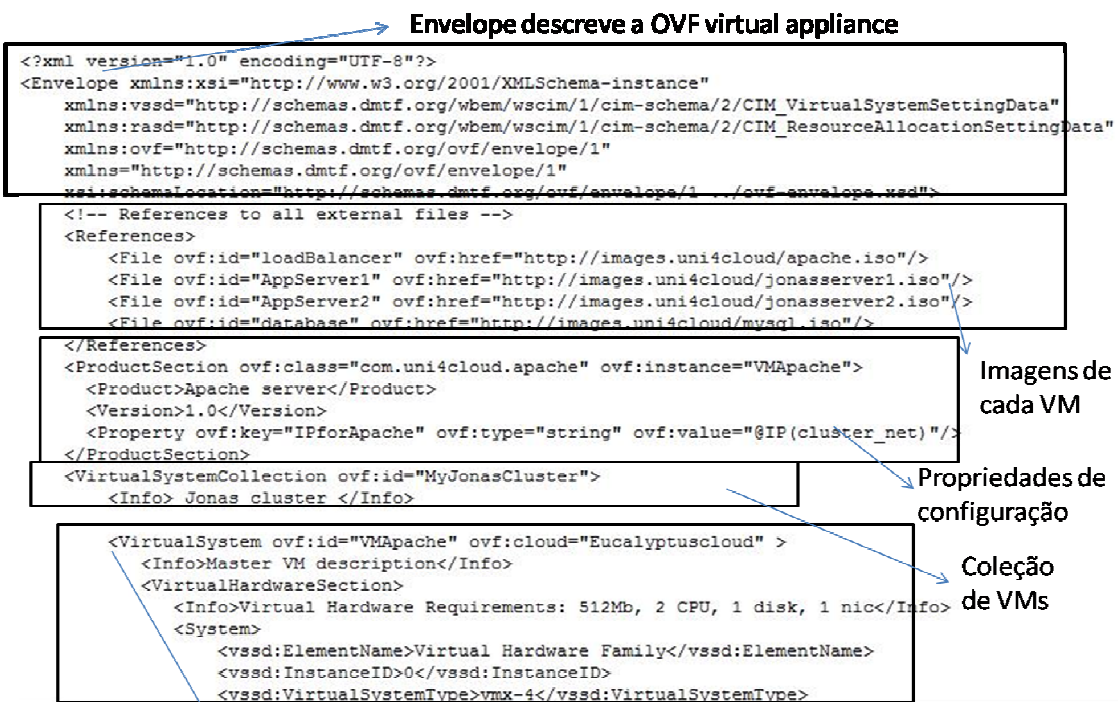
Para implantar a *virtual appliance* da Figura 4 em um conjunto de nuvens de infraestrutura os passos seguintes são realizados.

(1) Modelagem da *virtual appliance* pelo configurador de serviço web

Nesta etapa um modelo é criado para representar as principais propriedades de cada componente da aplicação. Por exemplo, na Figura 4 cada elemento da figura representa uma máquina virtual que conterá propriedades associadas como: imagem de disco (neste caso, uma imagem Xen de cada VM), nuvem na qual a VM será implantada (neste exemplo usamos duas nuvens privadas configuradas no nosso laboratório, baseadas em Eucalyptus [Eucalyptus 2009] e Open Nebula [Nebula 2009]) além de portas na qual o componente executa. O software configurador do serviço web é uma extensão da ferramenta *Jasmine Design*, a qual incorpora algumas propriedades referentes à computação em nuvem como especificar em qual nuvem o componente irá ser implantado.

(2) Conversão do modelo em uma especificação do Padrão OVF

A Figura 5 mostra parte do arquivo *OVF descriptor* que contém a especificação da aplicação e é gerada automaticamente baseada no cluster de Jonas modelado na Figura 4. O arquivo OVF contém as máquinas virtuais citadas acima, quatro imagens contendo o sistema operacional *Debian GNU/Linux Etch*, além das configurações de rede. A Figura 5 mostra a seção *<References>* que faz referência às quatro imagens utilizadas que foram armazenadas em um repositório de imagens no nosso laboratório. A seção *<ProductSection>* mostra o exemplo de uma propriedade que é o endereço IP da VM que contém o Apache e que é gerada dinamicamente em tempo de implantação por DHCP. A seção *<VirtualSystemCollection>* contém as especificações de cada máquina virtual *<VirtualSystem>* como nós filhos. Por exemplo, a *virtual system* que representa a máquina virtual que contém o Apache é parcialmente mostrada na parte inferior da figura especificando o hardware daquela VM (CPU de 2 cores, 512 Mb de RAM, disco e interface de rede).



1 VM da coleção será 1 servidor Apache que será implantado na nuvem privada do Eucalyptus. Esta VM tem hardware: CPU = 2 Cores; RAM = 512 MB, etc...

Figura 5 – *Virtual Appliance* do cluster JONAS no Formato OVF

(3) Distribuição automática da aplicação

Uma vez que o arquivo OVF é gerado pelo **configurador de serviço web** pode-se solicitar a implantação da virtual appliance o que resulta na invocação do **gerenciador de nuvem** que irá interpretar o arquivo *OVF descriptor* da Figura 5 para implantar as VMs nas nuvens IaaS escolhidas. Neste exemplo, o gerenciador genérico de nuvem tinha registrado dois adaptadores para os **gerenciadores de nuvem específicos para OpenNebula e para Eucalyptus**, que são responsáveis por fazer as chamadas às APIs específicas de cada nuvem IaaS (ex: chamar o serviço *runInstance()* do Eucalyptus para

implantar a VM do Apache na nuvem do Eucalyptus). Ao final do processo, o novo serviço é disponibilizado, distribuindo cada máquina virtual na sua devida nuvem, com todos os componentes configurados (cada VM ao fazer o *boot* pela primeira vez pode realizar configurações locais baseadas em scripts como, por exemplo, configurar o IP do banco de dados ao inicializar as VMs de cada servidor JOnAS).

A Abordagem Uni4Cloud propicia os seguintes benefícios:

- Possibilidade de trocar o provedor IaaS de forma simples bastando alterar o modelo e realizar uma nova implantação. Isto beneficia empresas que tenham interesse em disponibilizar sua *virtual appliance* em diversas nuvens distintas dependendo de qual lhe traga mais benefícios (ex: custo menor, maior segurança, etc.). Isto também ajuda a minimizar o problema de ficar preso a um provedor IaaS específico (*vendor lock in*).
- Utilização de padrões da área de *cloud computing*. O padrão OVF está sendo adotado em soluções de *cloud computing* de grandes empresas como VMWare IBM, Microsoft, Sun dentre outras para construção de *virtual appliances*. Basear-se neste padrão facilita integrar a abordagem Uni4Cloud com soluções destas empresas.
- Facilitar a implantação automática de aplicações web. O exemplo baseado no cluster de JOnAS descrito anteriormente mostra como a Uni4Cloud pode facilitar a implantação de uma aplicação relativamente complexa em nuvens de IaaS. Tal cenário poderia ser bastante útil na prática, pois empresas consomem bastante tempo (e dinheiro) para instalar e configurar aplicações com este perfil em servidores físicos em seus *data centers* locais. A Uni4Cloud provê uma forma simples de distribuir a aplicação automaticamente em nuvens IaaS distintas, abstraindo detalhes da instalação física e simplificando consideravelmente o processo de implantação.

5. Trabalhos Relacionados

A abordagem Uni4Cloud facilita a modelagem e implantação de serviços web sobre nuvens híbridas através do empacotamento da aplicação em uma virtual appliance baseada no padrão OVF. A especificação OVF é interpretada e os componentes da aplicação são implantados automaticamente em nuvens IaaS distintas.

O projeto Europeu Reservoir [Rochwerger et al. 2009] se baseia em um conceito de federação de nuvens onde vários provedores IaaS (que são membros da federação Reservoir) implementam a mesma camada de middleware (ex: algo similar ao software gerenciador de nuvem específico) que dá suporte ao funcionamento da sua nuvem IaaS. Estes provedores são capazes de se comunicar entre si e cooperar (ex: migrar uma máquina virtual de um provedor para outro) pois se baseiam na mesma tecnologia do middleware de nuvem Reservoir. No entanto, o projeto Reservoir ainda não suporta a distribuição de um serviço web de forma integrada com outros provedores IaaS (como GoGrid, Amazon EC2, etc..) e também não facilita a modelagem do serviço para criação automatizada das *Virtual Appliances*. Portanto, a abordagem Uni4Cloud oferece funcionalidades complementares ao Reservoir, podendo escolher o Reservoir como uma

potencial nuvem de IaaS na qual um determinado componente de software pode ser distribuído.

O projeto Altocumulus, da IBM [Maximilien et al. 2009], propõe uma plataforma de middleware que oferece uma API uniforme para implantar e gerenciar recursos e aplicações em nuvens heterogêneas. Os serviços oferecidos pela API do Altocumulus são mapeados para os serviços equivalentes oferecidos por diferentes provedores de nuvem através de adaptadores criados especificamente para cada nuvem. Uma limitação do Altocumulus é que ele define a sua própria API genérica, o que pode vir a desestimular a sua adoção por empresas e usuários que não queiram ficar presos a uma API proprietária (*API lock-in*). Ao contrário, a abordagem Uni4Cloud se baseia no padrão OCCI que está sendo suportado por diversos provedores de nuvem IaaS. Outra limitação é que o Altocumulus atualmente não suporta a modelagem da virtual appliance e sua implantação de forma híbrida, ou seja, contendo diferentes componentes da mesma virtual appliance em nuvens distintas (no Altocumulus todas as máquinas virtuais da mesma *virtual appliance* são implantadas na mesma nuvem IaaS).

6. Conclusão e Trabalhos Futuros

Este artigo apresentou a abordagem Uni4Cloud que procura facilitar a modelagem, implantação, configuração e distribuição de aplicações Web complexas em nuvens que oferecem servidores virtuais como recurso, chamadas de nuvens de infra-estrutura (IaaS). O artigo dá uma visão geral da implementação dos componentes de software que fazem parte da Uni4Cloud além de mostrar um exemplo de como a abordagem foi usada na prática para implantar automaticamente uma aplicação de comércio eletrônico baseada em um Cluster do servidor de aplicação JOnAS.

Como trabalho futuro pretendemos evoluir a implementação de todos os componentes da arquitetura do Uni4Cloud além de implementar outros componentes relacionados ao monitoramento e auto-adaptação das aplicações (ex: possibilitar que o tamanho do cluster de servidores de aplicação possa crescer/diminuir dinamicamente) e gerenciamento de SLAs.

Também procuraremos realizar estudos de caso para investigar outros exemplos de aplicações (diferentes de aplicações Web, conforme ilustrado neste artigo) como aplicações de alto desempenho e aplicações de *Grid* podem se beneficiar de infra-estruturas de nuvem. Acreditamos que distribuir os componentes de tais aplicações em nuvens IaaS requer processo semelhante ao que mostramos no nosso estudo de caso. Outros projetos discutidos na seção 5, como o Reservoir, já desenvolveram estudos de caso [Galán et al. 2009] em que uma nuvem de IaaS é usada para distribuir os componentes do *middleware* de *grid* da *Sun Microsystems* (*Sun Grid Engine*) e dar suporte a aplicações em grade executando na nuvem.

Agradecimentos

Este trabalho é parcialmente financiado pela Fundação Edson Queiroz, Universidade de Fortaleza, através do Projeto OW2.

Referências

- Appengine (2009), Google App Engine, Available from: <http://code.google.com/intl/pt-BR/appengine/>, March.
- Armbrust, M., Fox, A. and Griffith, R. (2009), "Above the Clouds: A Berkeley View of Cloud Computing." University of California Berkley Technical Report. <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf>, March.
- Azure (2009), Windows Azure Platform., Available from: <http://www.microsoft.com/windowsazure/>, March.
- DMTF (2009), Open Virtualization Format Specification, Version 1.0.0. Available from: http://www.dmtf.org/standards/published_documents/DSP0243_1.0.0.pdf, March.
- EC2 (2009), Amazon Elastic Compute Cloud (Amazon EC2), Available from: <http://aws.amazon.com/ec2/>, March.
- Eucalyptus (2009), Eucalyptus - Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems, Available from: <http://open.eucalyptus.com/>, March.
- Foster, Ian T., Zhao, I., Raicu, Ioan, et al. (2009) "Cloud Computing and Grid Computing 360-Degree Compared", In: CoRR abs/0901.0131.
- Galán, F., Sampaio, A., Rodero-Merino, L. et al., (2009) "Service Specification in Cloud Environments Based on Extensions to Open Standards". In: Fourth International Conference on COMmunication System softWARE and MiddlewaRE (COMSWARE), Dublin, Ireland: IEEE.
- Gens, Frank (2008), "IDC consulting. IT Cloud Services Forecast - 2008, 2012, In: A Key Driver of New Growth.
- Gottfrid , D., (2007) "Self-service, Prorated Super Computing Fun!", In: New York Times; Available from: <http://open.blogs.nytimes.com/2007/11/01/self-service-prorated-super-computing-fun/>, September.
- Jasmine (2009), Jasmine, Available from: <http://jasmine.ow2.org/>, April.
- Leavitt, Neal, (2009) "Is Cloud Computing Really Ready for Prime Time?" IEEE Computer, vol. 42, no. 1, p. 15-20.
- Lynch, Merrill (2008), "The Cloud Wars: \$100+ billion at stake".
- Maximilien, E. M, Ranabahu, A., Engehausen, R., Anderson, L (2009), "IBM Altocumulus: A Cross-Cloud Middleware and Platform". In: Proc. of the 24th ACM SIGPLAN Conference Companion on Object Oriented Programming Systems, Languages and Applications (OOPSLA'09), pp. 805-806, ACM Press.
- Mell, P. and Grance, T. (2009), "National Institute of Standards and Technology, Information Technology Laboratory (NIST) Working Definition of Cloud ", <http://csrc.nist.gov/groups/SNS/cloud-computing/index.html>.

- Nebula (2009), The Open Source Toolkit for Cloud Computing, Available from: <http://www.opennebula.org/>, March.
- OCCI (2009), OGF Open Cloud Computing Interface Working Group, Open Cloud Computing Interface, Available from: <http://www.occ-wg.org/>, March.
- Pettey, C. (2007), Gartner Estimates ICT Industry Accounts for 2 Percent of Global CO2 Emissions. , Available from: <http://www.gartner.com/it/page.jsp?id=503867>, March.
- Pring, Ben and Desisto, Robert P. (2008), "Gartner report ID Number G00162998, Predicts 2009", In: Software as a Service Gains Traction.
- Rochwerger, B., Caceres, J., Montero, Ruben S. et al.,(2009) "The RESERVOIR Model And Architecture for Open Federated Cloud Computing." In: IBM Systems Journal
- Salesforce (2009), Salesforce.com., Available from: www.salesforce.com, March.
- Smith, David M. and Austin, Tom (2008), "Gartner Report ID Number: G00163522, Predicts 2009", In: Cloud Computing Beckons.
- VMware (2009), VMware hypervisor, Available from: <http://www.vmware.com/>, March.
- Xen (2009), Xen Hypervisor, Available from: <http://www.xen.org/>, March.