

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/237644729>

# Computação em Nuvem: Conceitos, Tecnologias, Aplicações e Desafios

## Article

CITATIONS

28

READS

46,015

### 3 authors:



[Flávio R. C. Sousa](#)

Universidade Federal do Ceará

38 PUBLICATIONS 402 CITATIONS

[SEE PROFILE](#)



[Leonardo Moreira](#)

Universidade Federal do Ceará

62 PUBLICATIONS 227 CITATIONS

[SEE PROFILE](#)



[Javam Machado](#)

Universidade Federal do Ceará

101 PUBLICATIONS 491 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



RING Project [View project](#)



Traffic orchestration based on clusters in urban road networks [View project](#)

## Capítulo

# 7

## Computação em Nuvem: Conceitos, Tecnologias, Aplicações e Desafios<sup>1 2</sup>

Flávio R. C. Sousa, Leonardo O. Moreira e Javam C. Machado

Universidade Federal do Ceará (UFC)

### *Abstract*

*Platforms and software are available as services, both being used in Cloud Computing environments. This has improved flexibility, reducing total business cost and providing on-demand services. Several companies already use cloud computing to distribute its software, and it is believed that a constant migration to these environments will take place in the upcoming years. This paper will introduce the main cloud computing concepts and technologies, its architecture, service and deployment models, as well as applications that can be executed in such environments. Finally, challenges and opportunities in the cloud computing field will be presented.*

### *Resumo*

*Plataformas e software estão disponibilizados como serviços, sendo estes disponibilizados por ambientes de Computação em Nuvem. Isso tem melhorado a flexibilidade, reduzindo o custo total dos negócios e provendo serviços sob demanda. Muitas empresas já utilizam computação em nuvem para distribuir seus sistemas de software e acredita-se que ocorrerá uma migração constante para estes ambientes. Esse artigo tem como objetivo apresentar os principais conceitos e tecnologias de computação em nuvem, sua arquitetura e modelos de serviços e implantação, bem como, aplicações executadas nestes ambientes. Por fim, desafios e oportunidades em computação em nuvem são apresentados.*

---

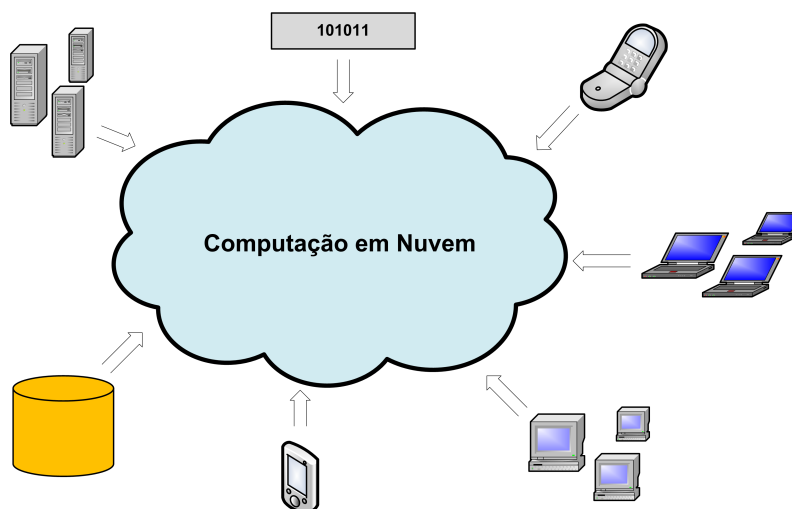
<sup>1</sup>Publicado no ERCMAPI 2009. Todos os direitos reservados a EDUFPI.

<sup>2</sup>Versão revisada e estendida em Setembro de 2010.

## 7.1. Introdução

Com o avanço da sociedade humana moderna, serviços básicos e essenciais são quase todos entregues de uma forma completamente transparente. Serviços de utilidade pública como água, eletricidade, telefone e gás tornaram-se fundamentais para nossa vida diária e são explorados por meio do modelo de pagamento baseado no uso [Vecchiola et al. 2009]. As infraestruturas existentes permitem entregar tais serviços em qualquer lugar e a qualquer hora, de forma que possamos simplesmente acender a luz, abrir a torneira ou usar o fogão. O uso desses serviços é, então, cobrado de acordo com as diferentes políticas de tarifação para o usuário final. Recentemente, a mesma ideia de utilidade tem sido aplicada no contexto da informática e uma mudança consistente neste sentido tem sido feita com a disseminação de *Cloud Computing* ou Computação em Nuvem.

Computação em nuvem é uma tendência recente de tecnologia cujo objetivo é proporcionar serviços de Tecnologia da Informação (TI) sob demanda com pagamento baseado no uso. Tendências anteriores à computação em nuvem foram limitadas a uma determinada classe de usuários ou focadas em tornar disponível uma demanda específica de recursos de TI, principalmente de informática [Buyya et al. 2009b]. Computação em nuvem pretende ser global e prover serviços para as massas que vão desde o usuário final que hospeda seus documentos pessoais na Internet até empresas que terceirizam toda infraestrutura de TI para outras empresas. Nunca uma abordagem para a utilização real foi tão global e completa: não apenas recursos de computação e armazenamento são entregues sob demanda, mas toda a pilha de computação pode ser aproveitada na nuvem. A Figura 7.1 mostra uma visão geral de uma nuvem computacional.



**Figura 7.1. Visão geral de uma nuvem computacional**

Com isso, os usuários estão movendo seus dados e aplicações para a nuvem e assim acessá-los de forma simples e de qualquer local. Isso é novamente um caso de utilização de processamento centralizado. Cenário semelhante ocorreu há aproximadamente 50 anos: um servidor de tempo compartilhado acessado por vários usuários. Contudo, nas últimas décadas, quando os computadores pessoais surgiram, os dados e as aplicações começaram a serem utilizados localmente. Certamente o paradigma de computação em

nuvem não é uma repetição da história. Há 50 anos, os servidores de tempo compartilhado foram adaptados por questões de limitação de recursos.

A computação em nuvem surge da necessidade de construir infraestruturas de TI complexas, onde os usuários têm que realizar instalação, configuração e atualização de sistemas de software. Em geral, os recursos de computação e hardware são propensos a ficarem obsoletos rapidamente e a utilização de plataformas computacionais de terceiros é uma solução inteligente para os usuários lidarem com a infraestrutura de TI. Na computação em nuvem os recursos de TI são fornecidos como um serviço, permitindo que os usuários o acessem sem a necessidade de conhecimento sobre a tecnologia utilizada. Assim, os usuários e as empresas passaram a acessar os serviços sob demanda e independente de localização, o que aumentou a quantidade de serviços disponíveis.

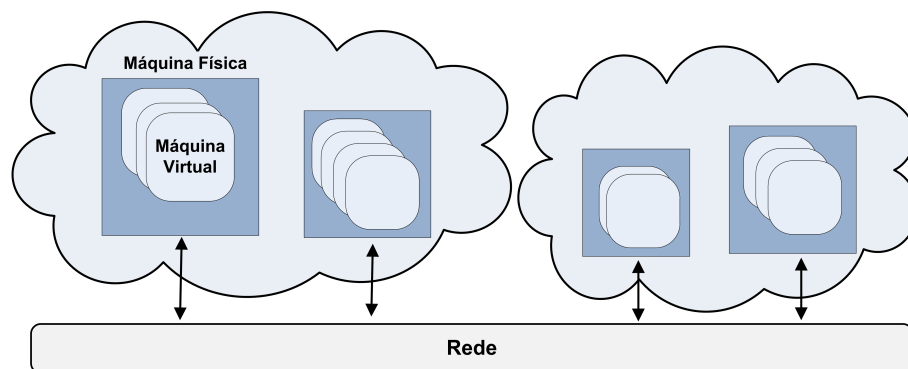
Na computação em nuvem, os recursos de TI são fornecidos como um serviço, permitindo aos usuários acessarem os serviços sem a necessidade de conhecimento sobre a tecnologia utilizada. Assim, os usuários e empresas passaram a acessar os serviços sob demanda e independente de localização, o que aumentou a quantidade de serviços disponíveis. Este trabalho aborda a computação em nuvem e está organizado da seguinte forma: a seção 7.2 apresenta os principais conceitos de computação em nuvem. A seção 7.3 apresenta as tecnologias atuais de computação em nuvem. A seção 7.4 apresenta algumas aplicações para computação em nuvem. A seção 7.5 apresenta os principais desafios em computação em nuvem e finalmente a seção 7.6 apresenta algumas conclusões.

## **7.2. Computação em Nuvem**

A computação em nuvem está se tornando uma das palavras chaves da indústria de TI. A nuvem é uma metáfora para a Internet ou infraestrutura de comunicação entre os componentes arquiteturais, baseada em uma abstração que oculta à complexidade de infraestrutura. Cada parte desta infraestrutura é provida como um serviço e, estes são normalmente alocados em centros de dados, utilizando hardware compartilhado para computação e armazenamento [Buyya et al. 2009b].

Para utilizarem os serviços, os usuários necessitam apenas ter em suas máquinas um sistema operacional, um navegador e acesso a Internet. Todos os recursos computacionais estão disponíveis na nuvem e as máquinas dos usuários não necessitam ter altos recursos computacionais, diminuindo o custo na aquisição de máquinas. Todo hardware pode ser utilizado para realizar alguma tarefa que seja adequada ao seu poder de processamento. Novos recursos de hardware podem ser adicionados a fim de aumentar o poder de processamento e cooperar com os recursos existentes.

A infraestrutura do ambiente de computação em nuvem normalmente é composta por um grande número, centenas ou milhares de máquinas físicas ou nós físicos de baixo custo, conectadas por meio de uma rede como ilustra a Figura 7.2. Cada máquina física tem as mesmas configurações de software, mas pode ter variação na capacidade de hardware em termos de CPU, memória e armazenamento em disco [Soror et al. 2010]. Dentro de cada máquina física existe um número variável de máquinas virtuais (VM) ou nós virtuais em execução, de acordo com a capacidade do hardware disponível na máquina física.



**Figura 7.2. Ambiente de Computação em Nuvem**

O modelo de computação em nuvem foi desenvolvido com o objetivo de fornecer serviços de fácil acesso, baixo custo e com garantias de disponibilidade e escalabilidade. Este modelo visa fornecer, basicamente, três benefícios. O primeiro benefício é reduzir o custo na aquisição e composição de toda infraestrutura requerida para atender as necessidades das empresas, podendo essa infraestrutura ser composta sob demanda e com recursos heterogêneos e de menor custo. O segundo é a flexibilidade que esse modelo oferece no que diz respeito à adição e substituição de recursos computacionais, podendo escalar tanto em nível de recursos de hardware quanto software para atender as necessidades das empresas e usuários. O último benefício é prover uma abstração e facilidade de acesso aos usuários destes serviços. Neste sentido, os usuários dos serviços não precisam conhecer aspectos de localização física e de entrega dos resultados destes serviços.

A computação em nuvem é uma evolução dos serviços e produtos de tecnologia da informação sob demanda, também chamada de *Utility Computing* [Brantner et al. 2008]. O objetivo da *Utility Computing* é fornecer componentes básicos como armazenamento, processamento e largura de banda de uma rede como uma “mercadoria” através de provedores especializados com um baixo custo por unidade utilizada. Usuários de serviços baseados em *Utility Computing* não precisam se preocupar com escalabilidade, pois a capacidade de armazenamento fornecida é praticamente infinita. A *Utility Computing* propõe fornecer disponibilidade total, isto é, os usuários podem ler e gravar dados a qualquer tempo, sem nunca serem bloqueados; os tempos de resposta são quase constantes e não dependem do número de usuários simultâneos, do tamanho do banco de dados ou de qualquer parâmetro do sistema. Os usuários não precisam se preocupar com *backups*, pois se os componentes falharem, o provedor é responsável por substituí-los e tornar os dados disponíveis em tempo hábil por meio de réplicas [Brantner et al. 2008].

Uma razão importante para a construção de novos serviços baseados em *Utility Computing* é que provedores de serviços que utilizam serviços de terceiros pagam apenas pelos recursos que recebem, ou seja, pagam pelo uso. Não são necessários investimentos iniciais em TI e o custo cresce de forma linear e previsível com o uso. Dependendo do modelo do negócio, é possível que o provedor de serviços repasse o custo de armazenagem, computação e de rede para os usuários finais, já que é realizada a contabilização do uso.

Com o objetivo de auxiliar na compreensão da *Utility Computing*, considere

o seguinte exemplo. Suponha que você tenha um requisito para operar 100 servidores por três anos. Uma opção seria alugar estes servidores por \$ 0.40 por instância/horas. Isso custaria aproximadamente  $100 \text{ servidores} * \$ 0.40 \text{ por instância/horas} * 3 \text{ anos} * 8760 \text{ horas/ano} = \$ 1.051.200$ .

Outra opção seria comprar os servidores e administrá-los. Suponha que o custo para comprar cada servidor seja de \$ 750 e que sejam necessários dois funcionários para administrá-los pagando \$ 100.000 por ano. Suponha ainda que os servidores consomem 150 *watts* cada e o custo da eletricidade é de \$ 0.10 por *quilowatt-hora*. Assim, o custo anual para operar os 100 servidores seria de \$ 13.140 e a opção de comprar e administrá-los custaria aproximadamente  $100 \text{ servidores} * \$ 750 + 3 \text{ anos} * \$ 13.140 \text{ eletricidade/ano} + 3 \text{ anos} * 2 \text{ funcionários} * \$ 100.000 \text{ salários/ano} = \$ 714.420$ .

Portanto, se a utilização dos servidores fosse de 100%, a opção de comprar 100 servidores seria mais barato. Por outro lado, se a utilização dos servidores fosse de 68% ou menos, a opção de alugar um serviço seria mais interessante. Mesmo considerando que os números apresentados acima são apenas estimativas e que nem todos os custos foram considerados, pode-se verificar que modelo de *Utility Computing* é preferível em muitos casos.

O *National Institute of Standards and Technology* (NIST) [Mell and Grance 2009] define computação em nuvem como um paradigma em evolução. Suas definições, casos de uso, tecnologias, problemas, riscos e benefícios serão redefinidos em debates entre os setores público e privado e essas definições, atributos e características evoluirão com o tempo.

Tratando-se especificamente da definição, ainda não se tem uma definição amplamente aceita. O NIST apresenta a seguinte definição para computação em nuvem: “*Computação em nuvem é um modelo que possibilita acesso, de modo conveniente e sob demanda, a um conjunto de recursos computacionais configuráveis (por exemplo, redes, servidores, armazenamento, aplicações e serviços) que podem ser rapidamente adquiridos e liberados com mínimo esforço gerencial ou interação com o provedor de serviços*”. [Armbrust et al. 2009] propõem a seguinte definição: “A computação em nuvem é um conjunto de serviços de rede ativados, proporcionando escalabilidade, qualidade de serviço, infraestrutura barata de computação sob demanda e que pode ser acessada de uma forma simples e pervasiva”.

Um levantamento com algumas propostas de definição para computação em nuvem pode ser encontrado em [Vaquero et al. 2009]. Para este trabalho, considerou-se a visão do NIST, que descreve que o modelo de computação em nuvem é composto por cinco características essenciais, três modelos de serviço e quatro modelos de implantação, detalhados a seguir.

### 7.2.1. Características Essenciais

As características essenciais são vantagens que as soluções de computação em nuvem oferecem. Algumas destas características, em conjunto, definem exclusivamente a computação em nuvem e faz a distinção com outros paradigmas. Por exemplo, a elasticidade rápida de recursos, amplo acesso e medição de serviço são características básicas para

compor uma solução de computação em nuvem.

### **Self-service sob demanda**

O usuário pode adquirir unilateralmente recurso computacional, como tempo de processamento no servidor ou armazenamento na rede, na medida em que necessite e sem precisar de interação humana com os provedores de cada serviço. O hardware e o software dentro de uma nuvem podem ser automaticamente reconfigurados, orquestrados e estas modificações são apresentadas de forma transparente para os usuários, que possuem perfis diferentes e assim podem personalizar os seus ambientes computacionais, por exemplo, instalação de software e configuração de rede para a definição de determinados privilégios.

### **Amplo acesso**

Recursos são disponibilizados por meio da rede e acessados através de mecanismos padronizados que possibilitam o uso por plataformas *thin* ou *thin client*, tais como celulares, *laptops* e PDAs. A interface de acesso à nuvem não obriga os usuários a mudar suas condições e ambientes de trabalho, como por exemplo, linguagens de programação e sistema operacional. Já os sistemas de software clientes instalados localmente para o acesso à nuvem são leves, como um navegador de Internet.

### **Pooling de recursos**

Os recursos computacionais do provedor são organizados em um *pool* para servir múltiplos usuários usando um modelo *multi-tenant* ou multi-inquilino [Jacobs and Aulbach 2007], com diferentes recursos físicos e virtuais, dinamicamente atribuídos e ajustados de acordo com a demanda dos usuários. Estes usuários não precisam ter conhecimento da localização física dos recursos computacionais, podendo somente especificar a localização em um nível mais alto de abstração, tais como o país, estado ou centro de dados.

### **Elasticidade rápida**

Recursos podem ser adquiridos de forma rápida e elástica, em alguns casos automaticamente, caso haja a necessidade de escalar com o aumento da demanda, e liberados, na retração dessa demanda. Para os usuários, os recursos disponíveis para uso parecem ser ilimitados e podem ser adquiridos em qualquer quantidade e a qualquer momento. A virtualização auxilia a elasticidade rápida na computação nuvem, criando várias instâncias de recursos requisitados utilizando um único recurso real [Abounaga et al. 2009]. Além disso, a virtualização é uma maneira de abstrair características físicas de uma plataforma computacional dos usuários, exibindo outro hardware virtual e emulando um ou mais ambientes que podem ser independentes ou não.

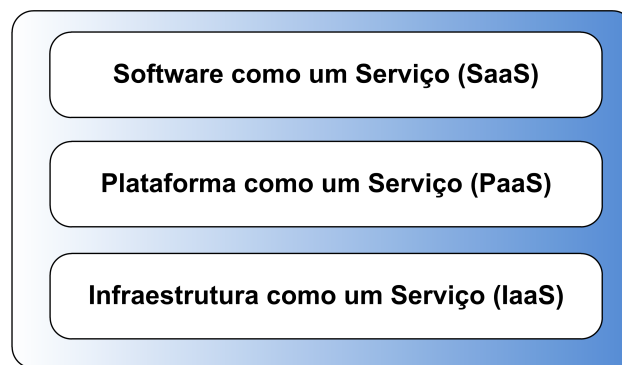
### **Serviço medido**

Sistemas em nuvem automaticamente controlam e otimizam o uso de recursos por meio

de uma capacidade de medição. A automação é realizada em algum nível de abstração apropriado para o tipo de serviço, tais como armazenamento, processamento, largura de banda e contas dos usuários ativos. O uso de recursos pode ser monitorado e controlado, possibilitando transparência para o provedor e o usuário do serviço utilizado. Para garantir a qualidade do serviço ou *Quality of Service* (QoS), pode-se utilizar a abordagem baseada em acordo de nível de serviço ou *Services Level Agreement* (SLA). O SLA fornece informações sobre os níveis de disponibilidade, funcionalidade, desempenho ou outros atributos do serviço como o faturamento e até mesmo penalidades em caso de violação destes níveis.

### 7.2.2. Modelos de Serviços

O ambiente de computação em nuvem é composto de três modelos de serviços. Estes modelos são importantes, pois eles definem um padrão arquitetural para soluções de computação em nuvem. A Figura 7.3 exibe estes modelos de serviços [Armbrust et al. 2009].



**Figura 7.3. Modelos de Serviços**

#### Software como um Serviço (SaaS)

O modelo de SaaS proporciona sistemas de software com propósitos específicos que estão disponíveis para os usuários através da Internet. Os sistemas de software são acessíveis a partir de vários dispositivos do usuário por meio de uma interface *thin client* como um navegador Web. No SaaS, o usuário não administra ou controla a infraestrutura subjacente, incluindo rede, servidores, sistemas operacionais, armazenamento ou mesmo as características individuais da aplicação, exceto configurações específicas. Com isso, os desenvolvedores se concentram em inovação e não na infraestrutura, levando ao desenvolvimento rápido de sistemas de software.

Como o software está na Web, ele pode ser acessado pelos usuários de qualquer lugar e a qualquer momento, permitindo maior integração entre unidades de uma mesma empresa ou outros serviços de software. Assim, novos recursos podem ser incorporados automaticamente aos sistemas de software sem que os usuários percebam estas ações, tornando transparente a evolução e atualização dos sistemas. O SaaS reduz os custos, pois é dispensada a aquisição de licenças de sistemas de softwares. Como exemplos de SaaS podemos destacar os serviços de *Customer Relationship Management* (CRM) da Salesforce [Salesforce 2010] e o Google Docs [Ciurana 2009].



## Plataforma como um Serviço (PaaS)

A PaaS oferece uma infraestrutura de alto nível de integração para implementar e testar aplicações na nuvem. O usuário não administra ou controla a infraestrutura subjacente, incluindo rede, servidores, sistemas operacionais ou armazenamento, mas tem controle sobre as aplicações implantadas e, possivelmente, as configurações das aplicações hospedadas nesta infraestrutura. A PaaS fornece um sistema operacional, linguagens de programação e ambientes de desenvolvimento para as aplicações, auxiliando a implementação de sistemas de software, já que contém ferramentas de desenvolvimento e colaboração entre desenvolvedores.

Em geral, os desenvolvedores dispõem de ambientes escaláveis, mas eles têm que aceitar algumas restrições sobre o tipo de software que se pode desenvolver, desde limitações que o ambiente impõe na concepção das aplicações até a utilização de sistemas de gerenciamento de banco de dados (SGBDs) do tipo chave-valor, ao invés de SGBDs relacionais. Do ponto de vista do negócio, a PaaS permitirá aos usuários utilizarem serviços de terceiros, aumentando o uso do modelo de suporte no qual os usuários se inscrevem para solicitações de serviços de TI ou para resoluções de problemas pela Web. Com isso, pode-se melhorar o gerenciamento do trabalho e as responsabilidades das equipes de TI das empresas. Como exemplos de SaaS podemos destacar as PaaS *Google App Engine* [Ciurana 2009] e *Aneka* [Vecchiola et al. 2009].

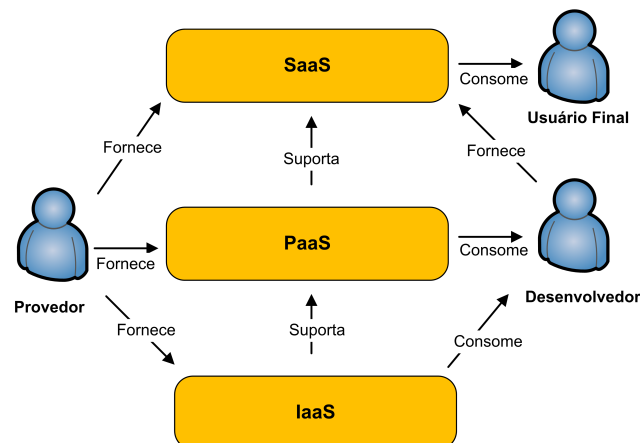
## infraestrutura como um Serviço (IaaS)

O IaaS é a parte responsável por prover toda a infraestrutura necessária para a PaaS e o SaaS. O principal objetivo do IaaS é tornar mais fácil e acessível o fornecimento de recursos, tais como servidores, rede, armazenamento e outros recursos de computação fundamentais para construir um ambiente sob demanda, que podem incluir sistemas operacionais e aplicativos. A IaaS possui algumas características, tais como uma *interface* única para administração da infraestrutura, *Application Programming Interface* (API) para interação com *hosts*, *switches*, balanceadores, roteadores e o suporte para a adição de novos equipamentos de forma simples e transparente. Em geral, o usuário não administra ou controla a infraestrutura da nuvem, mas tem controle sobre os sistemas operacionais, armazenamento e aplicativos implantados, e, eventualmente, seleciona componentes de rede, tais como *firewalls*.

O termo IaaS se refere a uma infraestrutura computacional baseada em técnicas de virtualização de recursos de computação. Esta infraestrutura pode escalar dinamicamente, aumentando ou diminuindo os recursos de acordo com as necessidades das aplicações. Do ponto de vista de economia e aproveitamento do legado, ao invés de comprar novos servidores e equipamentos de rede para a ampliação de serviços, pode-se aproveitar os recursos disponíveis e adicionar novos servidores virtuais à infraestrutura existente de forma dinâmica. O Amazon *Elastic Cloud Computing* (EC2) [Robinson 2008] e o *Elastic Utility Computing Architecture Linking Your Programs To Useful Systems* (Eucalyptus) [Liu et al. 2007] são exemplos de IaaS.

## Papéis na Computação em Nuvem

Os papéis são importantes para definir responsabilidades, acesso e perfil para os diferentes usuários que fazem parte e estão envolvidos em uma solução de computação em nuvem. Para entender melhor a computação em nuvem, pode-se classificar os atores dos modelos de acordo com os papéis desempenhados [Marinos and Briscoe 2009]. A Figura 7.4 destaca estes papéis.



**Figura 7.4. Papéis na Computação em Nuvem**

O provedor é responsável por disponibilizar, gerenciar e monitorar toda a estrutura para a solução de computação em nuvem, deixando o desenvolvedor e o usuário final sem esse tipo de responsabilidade e fornecendo serviços nos três modelos de serviços. Os desenvolvedores utilizam os recursos fornecidos e disponibilizam serviços para os usuários finais. Esta organização em papéis ajuda a definir os atores e os seus diferentes interesses. Os atores podem assumir vários papéis ao mesmo tempo de acordo com os interesses, sendo que apenas o provedor fornece suporte a todos os modelos de serviços.

### 7.2.3. Modelo de Implantação

Tratando-se do acesso e disponibilidade de ambientes de computação em nuvem, têm-se diferentes tipos de modelos de implantação. A restrição ou abertura de acesso depende do processo de negócio, do tipo de informação e do nível de visão. Pode-se perceber que certas empresas não desejam que todos os usuários possam acessar e utilizar determinados recursos no seu ambiente de computação em nuvem. Neste sentido, surge a necessidade de ambientes mais restritos, onde somente alguns usuários devidamente autorizados possam utilizar os serviços providos. Os modelos de implantação da computação em nuvem podem ser divididos em nuvem pública, privada, comunidade e híbrida [Mell and Grance 2009].

#### Nuvem Privada

No modelo de implantação de nuvem privada, a infraestrutura de nuvem é utilizada exclusivamente para uma organização, sendo esta nuvem local ou remota e administrada pela própria empresa ou por terceiros. Neste modelo de implantação são empregados políticas

de acesso aos serviços. As técnicas utilizadas para prover tais características podem ser em nível de gerenciamento de redes, configurações dos provedores de serviços e a utilização de tecnologias de autenticação e autorização.

### **Nuvem Pública**

No modelo de implantação de nuvem pública, a infraestrutura de nuvens é disponibilizada para o público em geral, sendo acessado por qualquer usuário que conheça a localização do serviço. Neste modelo de implantação não podem ser aplicadas restrições de acesso quanto ao gerenciamento de redes, e menos ainda, utilizar técnicas para autenticação e autorização.

### **Nuvem Comunidade**

No modelo de implantação de nuvem comunidade ocorre o compartilhamento por diversas empresas de uma nuvem, sendo esta suportada por uma comunidade específica que partilhou seus interesses, tais como a missão, os requisitos de segurança, política e considerações sobre flexibilidade. Este tipo de modelo de implantação pode existir localmente ou remotamente e geralmente é administrado por alguma empresa da comunidade ou por terceiros.

### **Nuvem Híbrida**

No modelo de implantação de nuvem híbrida, existe uma composição de duas ou mais nuvens, que podem ser privadas, comunidade ou pública e que permanecem como entidades únicas, ligadas por uma tecnologia padronizada ou proprietária que permite a portabilidade de dados e aplicações.

#### **7.2.4. Arquitetura da Computação em Nuvem**

A arquitetura de computação em nuvem é baseada em camadas, sendo que cada uma destas trata de uma particularidade na disponibilização de recursos para as aplicações [Buyya et al. 2009b]. Uma camada é uma divisão lógica de componentes de hardware e software. Alguns destes recursos computacionais podem ser agrupados e organizados para realizar uma determinada tarefa do sistema como um todo. Cada camada pode ter seu gerenciamento ou monitoramento de forma independente das outras camadas, melhorando a flexibilidade, reuso e escalabilidade no tocante a substituição ou adição de recursos computacionais sem afetar as outras camadas. A Figura 7.5 exhibe essas camadas e suas respectivas associações.

A camada de mais baixo nível é a de infraestrutura física, que contém centros de dados, *clusters*, *desktops* e outros recursos de hardware, podendo ter recursos heterogêneos. Com isso, fornece certa flexibilidade e facilidade de agregação de novos recursos à medida que se tornem necessários. Uma camada de *middleware* é responsável por gerenciar a infraestrutura física e tem por objetivo fornecer um núcleo lógico de uma nuvem. Estes serviços contêm negociações de QoS, gerenciamento dos SLAs, serviços de cobrança, serviços para verificar aceitação de requisições baseado no QoS e preço, serviços



**Figura 7.5. Arquitetura da Computação em Nuvem [Vecchiola et al. 2009]**

para cálculo, serviços de gerenciamento de virtualização, entre outros.

No nível acima da camada de *middleware*, encontra-se a camada responsável por prover suporte para a construção de aplicações e que contem ferramentas ou ambientes de desenvolvimento. Estes ambientes possuem *interfaces* Web 2.0, *marshups*, componentes, recursos de programação concorrente e distribuída, suporte a *workflows*, bibliotecas de programação e linguagens de programação. Esta camada de desenvolvimento não é utilizada pelos usuários finais, e sim, pelos usuários mais experientes, aqueles que desenvolvem as soluções para computação em nuvem. Por fim, encontra-se a camada das aplicações de computação em nuvem. Esta camada é de interesse do usuário, pois é por meio dela que eles utilizam os aplicativos. As camadas abaixo desta são responsáveis pelas características de escalabilidade, disponibilidade, ilusão de recursos infinitos e alto desempenho. Algumas soluções de arquitetura podem incluir uma camada de gerenciamento de adaptações sendo esta responsável por fornecer adaptação a estas soluções. Essas adaptações ocorrem de forma automática ou semi-automática e com isso, diminui os esforços humanos para gerenciar arquiteturas de computação em nuvem.

### 7.3. Tecnologias para Computação em Nuvem

A computação em nuvem envolve uma grande quantidade de conceitos e tecnologias. Empresas e corporações como a Amazon, Google e Microsoft estão publicando serviços computacionais seguindo a lógica da infraestrutura de computação em nuvem, sendo a Amazon a pioneira em disponibilizar e comercializar infraestrutura deste tipo.

A comunidade acadêmica também tem apresentado interesse em computação em nuvem, sendo que alguns trabalhos tem sido direcionados para melhorias de aspectos de desempenho, segurança, implementação e confiabilidade do sistema como um todo [Armbrust et al. 2009]. Outros trabalhos têm desenvolvido novas técnicas para compor infraestruturas adequadas a cada contexto dos ambientes de computação em nuvem, dentre os quais podemos destacar o projeto Eucalyptus [Liu et al. 2007], desenvolvido pela Universidade da Califórnia. A seguir apresentamos algumas tecnologias, destacando um modelo de programação, infraestruturas e plataformas para computação em nuvem.

## MapReduce/Hadoop

O MapReduce é um modelo de programação que visa o processamento de grandes volumes de dados, onde o usuário especifica sua aplicação através da sequência de operações MapReduce [Dean and Ghemawat 2004]. As tarefas de paralelismo, tolerância a falhas, distribuição dos dados e balanceamento de carga são deixadas a cargo do sistema MapReduce, simplificando o processo de desenvolvimento. Do ponto de vista de sistemas distribuídos, o MapReduce oferece transparência de replicação, distribuição e sincronização.

No modelo MapReduce cada operação é composta por duas funções. A primeira chamada de função de Mapeamento recebe uma porção do arquivo de entrada e de acordo com a especificação do usuário emite um conjunto de tuplas intermediárias no formato chave-valor. A segunda função, chamada Redução, recebe um conjunto de valores associados a cada chave, chamados de blocos. O processamento, definido pelo usuário, é realizado sobre cada bloco. Por fim, cada função de redução emite um conjunto de tuplas que são armazenadas em arquivos de saída. A Figura 7.6 ilustra as fases do MapReduce.

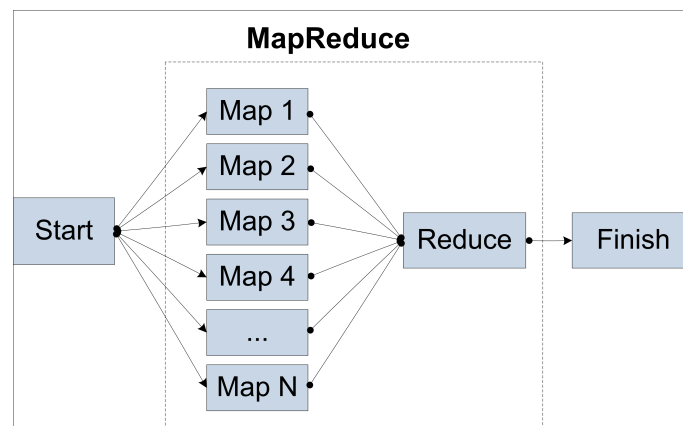


Figura 7.6. Fases do MapReduce

O sistema MapReduce gerencia o processamento através de um processo *master*, cuja função é orquestrar o processamento, gerenciar o processo de agrupamento de registros e distribuir os blocos de forma equilibrada. O MapReduce foi implementado utilizando a linguagem C++ e possui interfaces para Java e Python. O MapReduce foi desenvolvido pelo Google mas existem algumas implementações de código livre, dentre as quais destaca-se o Hadoop [Hadoop 2010].

O Hadoop é um *framework* de código livre desenvolvido em Java para rodar aplicações que manipulem uma grande quantidade de dados em ambientes distribuídos. O Hadoop é composto pelo sistema de arquivos *Hadoop Distributed File System* (HDFS) e um ambiente de execução paralela. Dentro deste ambiente, ou melhor, do Hadoop *framework*, pode-se encontrar vários subprojetos como, por exemplo, a implementação do MapReduce, o sistema de gerenciamento de dados distribuído denominado *HBase* e a linguagem para fluxo de dados e estrutura de execução para computação paralela denominada *Pig* [Olston et al. 2008].

O Hadoop possui como principais características: sistemas de armazenamento distribuído, arquivos particionados em grandes blocos e distribuídos nos nós do sistema, blocos replicados para lidar com falha de hardware e um local para dados temporários. Diferentemente de outras abordagens de sistemas de arquivos distribuídos, o armazenamento e processamento do HDFS é feito em cada nó do sistema. Dessa forma, usando o MapReduce ou o Hadoop, é relativamente fácil para um projeto de computação trabalhar com 10 TB de dados e com mais de 1000 nós.

### **Amazon Web Services (AWS)**

O Amazon AWS é um ambiente de computação em nuvem com características de escalabilidade, disponibilidade, elasticidade e desempenho para aplicações executadas neste ambiente. O Amazon AWS disponibiliza uma infraestrutura completa para computação em diversos níveis de processamento, desde tarefas simples até de alto desempenho e possui uma gerência eficaz dos recursos. O Amazon Web Services é composto por um conjunto de sistemas, dentre os quais podemos destacar:

- Execução: *Elastic Compute Cloud (EC2)*.
- Armazenamento: *Simple Storage Service (S3)*, *SimpleDB* e *Relational Database Service (RDS)*.
- Programação: *Simple Queue Service (SQS)* e *Elastic MapReduce*.
- Monitorização: *Cloudfront*.

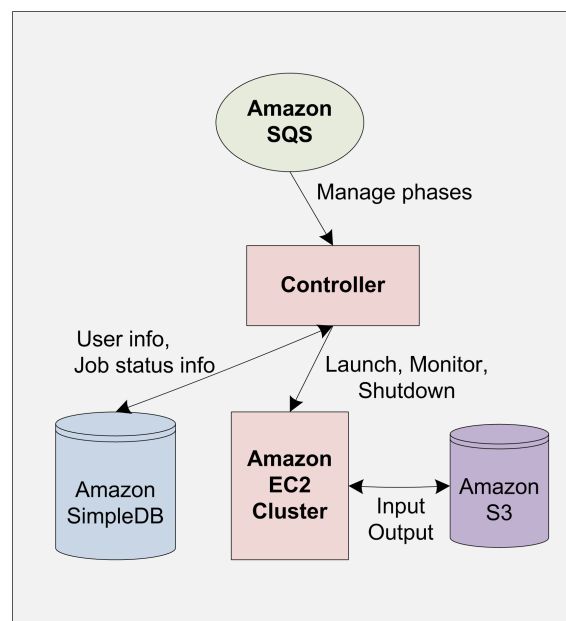
O EC2 é um sistema responsável pelo gerenciamento da execução de aplicações na infraestrutura da Amazon. O EC2 permite um controle completo das instâncias dos sistemas, sendo possível acessar e interagir com cada uma destas, de forma similar a máquinas convencionais. Também é possível escolher as características de cada instância, tais como sistema operacional, pacotes de software e as configurações das máquinas, como CPU, memória e armazenamento. Para garantir a segurança, o EC2 utiliza *firewall* para controlar o acesso às instâncias, criando ambientes virtuais privados. Para utilizar o EC2, primeiro é necessário criar uma imagem de máquina para executar as aplicações, chamada de *Amazon Machine Image (AMI)*, que contém os aplicativos, bibliotecas, dados e configurações associadas. Esta imagem é armazenada em um repositório seguro, rápido e confiável. O desempenho de CPU é medido em *EC2 Compute Unit (ECU)*, uma unidade empírica da Amazon que corresponde a aproximadamente entre 1.0 e 1.2 GHz AMD Opteron. As máquinas virtuais podem não corresponder a máquinas físicas mas possuem características bem definidas, por exemplo:

- *Small*: 32-bit, 1 ECU, 1,7GB RAM, 150 GB HD temporário.
- *Large*: 64-bit, 4 ECU, 7,5 GB RAM, 840 GB HD temporário.
- *XL*: 64-bit, 8 ECU, 15 GB RAM, 1680 GB HD temporário.

O ambiente AWS disponibiliza AMIs pré-configuradas que podem ser alteradas pelo usuário, assim como a opção de criar uma nova AMI. Em ambos os casos, a AMI pode ocupar no máximo 10GB. Alguns exemplos de sistemas operacionais disponíveis para execução nas AMI são Red Hat Linux, Windows Server 2003, OpenSolaris, Ubuntu, Debian. Em relação aos sistemas de software podemos destacar o MySQL, Hadoop, Condor, OpenMPI, Apache e IIS. O EC2 suporta os protocolos *Representational State Transfer* (REST) e *Simple Object Access Protocol* (SOAP) e múltiplas interfaces para serviços AWS: Java, C#, Python, extensões do Firefox, ferramentas de linha de comando e o console online *Amazon Management Console*.

O uso do EC2 requer a criação de pares de chaves públicas/privadas e sua associação a uma máquina virtual. Com isso, pode-se executar umas destas instâncias e depois verificar o estado de cada instância e configurar aspectos de rede, tais como DNS. Cada instância possui um par endereço IP, nome DNS e outro par privado e os endereços IP são dinâmicos, o que requer um serviço de DNS dinâmico. Assim, atrasos na atualização do DNS podem fazer parecer que há falhas. Em relação à segurança, o EC2 utiliza grupos de segurança. Estes grupos são conjuntos de regras de acesso por redes a instâncias EC2. O tráfego de saída é livre e o de entrada é limitado. As regras de grupo limitam o acesso externo, acesso a uma porta específica e determinados protocolos.

O auto-escalamento permite escalar automaticamente aplicações Amazon EC2 de acordo com condições definidas pelo usuário. Dessa forma, pode-se aumentar o número de instâncias quando a demanda aumenta e reduzi-la quando a demanda diminui, de modo a minimizar os custos. O EC2 fornece um ambiente altamente confiável, visto que a substituição de instâncias pode ser realizada de forma rápida, garantindo o QoS. A Figura 7.7 ilustra um exemplo de interação do EC2 para fornecer uma solução completa para computação, processamento de consultas e armazenamento.



**Figura 7.7. Amazon EC2 [Robinson 2008]**

O Amazon SQS é um sistema de comunicação confiável e escalável para enfileirar

e armazenar mensagens que estão sendo trocadas entre os recursos computacionais. Os desenvolvedores podem, de maneira simples, mover dados entre os recursos distribuídos das aplicações de diferentes domínios com garantia de entrega. O SQS fornece automação de *workflows* trabalhando em conjunto com o EC2. No Amazon SQS, uma mensagem é um bloco de texto com limite máximo de 8KB e as filas armazenam as mensagens até que estas sejam entregues. Estas filas armazenam as mensagens relacionadas e permitem definir opções de entrega e de controle de acesso. Em relação a coerência, as filas de mensagens são replicadas para garantir a tolerância a falhas e escalabilidade. Quando é feita uma leitura são consultados um quórum das réplicas e por isso podem não ser obtidas todas as mensagens. A entrega das mensagens é pedida pelo receptor e assim não se podem garantir tempos de entrega. O SQS também não garante a ordenação das mensagens e uma mensagem pode ser entregue mais do que uma vez. O SQS não deve ser usado para operações que pressuponham um dado estado da aplicação ou para aplicações temporizadas, como sistemas transacionais.

O S3 é um sistema de arquivos distribuído, utilizado para recuperar e armazenar dados. O S3 fornece um repositório seguro, confiável e rápido para armazenar as imagens AMI, além de armazenar e recuperar os resultados intermediários durante a execução das tarefas de processamento. Durante a execução, as tarefas recuperam os arquivos no S3 e realizam o devido processamento. As soluções EC2 armazenam arquivos como objetos no S3 e todos os metadados relacionados ao objeto no SimpleDB. O SimpleDB fornece as funcionalidades de um sistema banco de dados como armazenamento, indexação e consultas em ambientes de nuvem. A arquitetura do SimpleDB é utilizada para o armazenamento e recuperação dos estados do sistema. A vantagem dessa abordagem é que em caso de falhas, um novo nó pode ser iniciado quase que imediatamente baseada nas mensagens contidas na fila do Amazon SQS e seu estado pelo SimpleDB.

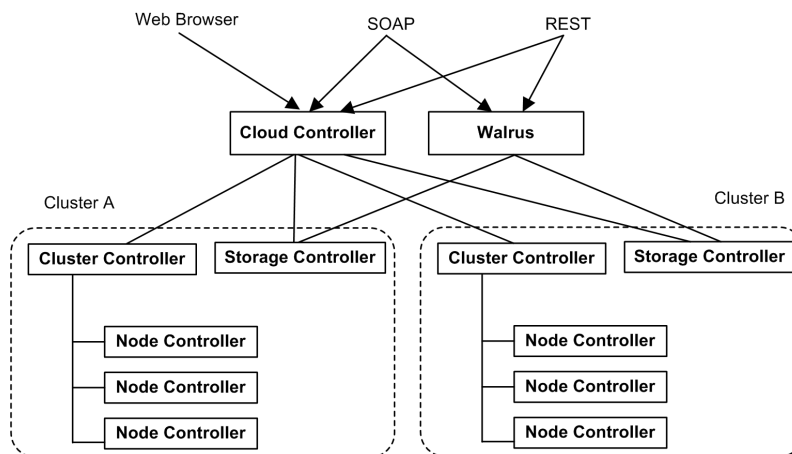
O EC2 Controller é o responsável por executar o processamento sob demanda utilizando em cooperação os componentes descritos anteriormente e faz toda a lógica de orquestração entre os componentes a fim de realizar o processamento distribuído. Ele pode utilizar o Hadoop como abordagem de paralelismo entre tarefas, ou seja, subdivide o processamento entre tarefas, coordena as execuções e reúne os resultados. O EC2 utiliza o Amazon *CloudWatch* para monitorar o estado das aplicações em execução no AWS.

## **Eucalyptus**

O projeto Eucalyptus [Liu et al. 2007] é uma infraestrutura de código aberto que fornece uma interface compatível com o Amazon EC2, S3, *Elastic Block Store* (EBS) e permite aos usuários criarem uma infraestrutura e experimentar a computação em nuvem. A arquitetura do Eucalyptus é simples, flexível e modular e contém uma concepção hierárquica que reflete os recursos comuns do ambiente. A Figura 7.8 mostra a arquitetura do Eucalyptus, composta por quatro partes:

- *Node Controller* (NC): nível inferior. Controla instâncias das máquinas virtuais nos nós.
- *Cluster Controller* (CC): nível intermediário. Ponte de comunicação entre NC e CLC.





**Figura 7.8. Arquitetura do Eucalyptus [Liu et al. 2007]**

- *Storage Controller* (Walrus): nível superior. Gerencia tráfego de dados dentro e fora da nuvem.
- *Cloud Controller* (CLC): nível superior. Controla a nuvem como um todo.

O Eucalyptus tem como objetivo auxiliar a pesquisa e o desenvolvimento de tecnologias para computação em nuvem e possui as seguintes características: interface compatível com o EC2, instalação e implantação simples usando ferramentas de gerenciamento de *clusters*, apresenta um conjunto de políticas de alocação extensível de nuvem, sobreposição de funcionalidade que não requer nenhuma modificação em ambiente Linux, ferramentas para administrar e auxiliar a gestão do sistema e dos usuários e capacidade de configurar vários clusters, cada um com endereços privados de rede interna em uma única nuvem.

Esse sistema permite aos usuários iniciar, controlar o acesso e gerenciar todas as máquinas virtuais utilizando uma emulação do protocolo SOAP do Amazon EC2 e interfaces de consulta. Neste sentido, os usuários interagem com o Eucalyptus utilizando as ferramentas e interfaces exatamente do mesmo modo que eles interagiriam com o Amazon EC2.

## Microsoft Azure

O Microsoft Azure é uma plataforma para a implementação de computação em nuvem que oferece um conjunto específico de serviços para desenvolvedores [Azure 2010]. Esta plataforma pode ser usada por aplicações em execução em nuvem ou fora desta. A plataforma Azure é formada pelo sistema operacional Windows Azure e um conjunto de serviços: *Live Services*, *.NET Services*, *SQL Services*, *SharePoint Services* e *Dynamics CRM Services*. A Figura 7.9 mostra a plataforma Azure.

O Windows Azure é um sistema operacional para serviços na nuvem que é utilizado para o desenvolvimento, hospedagem e gerenciamento dos serviços dentro do ambiente Azure. Microsoft *.NET Services* é um conjunto de serviços escaláveis, orientados



**Figura 7.9. Plataforma Microsoft Azure [Azure 2010]**

ao desenvolvedor e que oferecem os componentes necessários para a maioria das aplicações baseadas em nuvem. Estes serviços possibilitam o desenvolvimento focado na lógica da aplicação ao invés de ter a necessidade de construir e disponibilizar o próprio serviço de infraestrutura na nuvem.

O *Live Services* é um conjunto de componentes dentro do Azure para o tratamento de dados do usuário e recursos da aplicação. *Live Services* possibilita aos desenvolvedores construir aplicações ricas que podem conectar com usuários do *Windows Live*. O *Live Services* inclui as tecnologias do *Live Mesh* para sincronização de dados dos usuários e possibilita a extensão de aplicações Web entre múltiplos dispositivos. O *SQL Services* é um serviço de armazenamento de dados e de processamento de consultas escalável, sendo construído com base na tecnologia do SQL Server. O componente *SharePoint Services* permite colaborar e criar aplicações Intranet e o *Dynamics CRM Services* é um sistema totalmente integrado de CRM.

## **Google App Engine**

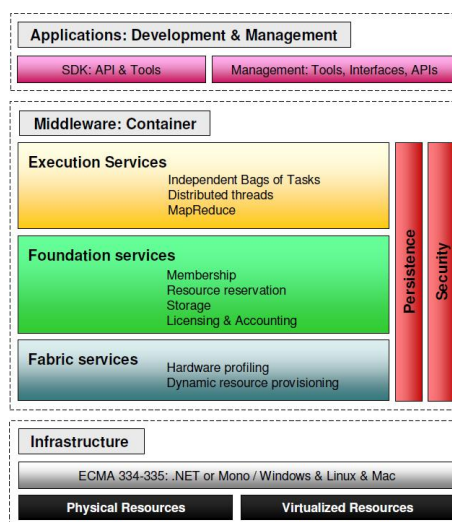
Google App Engine é uma plataforma para o desenvolvimento de aplicações Web escaláveis que são executados na infraestrutura do Google [Ciurana 2009]. Esta plataforma fornece um conjunto de APIs e um modelo de aplicação que permite aos desenvolvedores utilizarem serviços adicionais fornecidos pelo Google, como o e-mail, armazenamento, entre outros.

De acordo com o modelo de aplicação previsto, os desenvolvedores podem criar aplicações Java e Python e utilizar diversos recursos tais como armazenamento, transações, ajuste e balanceamento de carga automáticos, ambiente de desenvolvimento local e tarefas programadas. O Google App Engine possui um serviço de armazenamento baseado no BigTable [Chang et al. 2006], um sistema distribuído de armazenamento de dados em larga escala. As aplicações desenvolvidas para o App Engine serão executadas no Google, que realiza automaticamente, caso necessário, o dimensionamento.

## **Aneka**

O Aneka é uma plataforma para a implementação de aplicações em computação em nuvem baseada em .NET [Vecchiola et al. 2009]. O Aneka fornece serviços de persistência, segurança (autorização, autenticação e auditoria), comunicação e manipulação de mensagens. com isso, o Aneka proporciona flexibilidade e extensibilidade para orquestrar vários serviços. O objetivo central do Aneka é fornecer um ambiente que é implantado em infraestruturas físicas e virtuais e que permite a execução de aplicativos desenvolvidos com modelos de aplicações diferentes.

O Aneka fornece aos desenvolvedores um conjunto de APIs para explorar esses recursos de forma transparente e expressar a lógica de negócio das aplicações usando abstrações de programação. Os desenvolvedores de sistema podem utilizar uma coleção de ferramentas para monitorar e controlar a infraestrutura implantada. O Aneka possui um *Software Development Kit* (SDK) que permite aos desenvolvedores criarem aplicações no contexto de nuvens em qualquer linguagem suportada pelo *.NET runtime* e um conjunto de ferramentas para criação rápida de nuvens, estando disponível para o Windows e sistemas baseados em Linux. A Figura 7.10 apresenta a arquitetura do Aneka.



**Figura 7.10. Arquitetura do Aneka [Vecchiola et al. 2009]**

No nível de aplicação, encontram-se as aplicações disponibilizadas para execução. Neste nível existem os artefatos de desenvolvimento de aplicações, tais como os SDK, APIs, ferramentas para o monitoramento e ajuste das aplicações e do ambiente. Cada um dos recursos no Aneka é instanciado em um *Container* que representa o ambiente onde as aplicações são executadas. Esse *Container* fornece gerenciamento básico de funcionalidades de um nó e *interface* para todos os serviços e operações existentes nele. O Container é composto pelos componentes: *Execution Services*, *Foundation Services*, *Fabric Services* e o *Transversal Services*. Os *Execution Services* são responsáveis pelos escalonamentos, técnicas de processamento como o MapReduce.

Os *Foundation Services* são os serviços de reservas e alocação de recursos, contabilização, armazenamento dos aspectos relacionados às tarefas, configurações, usuários e recursos. Os *Fabric Services* fornecem acesso aos sistemas de provisionamento para recursos da infraestrutura. Os *Transversal Services* tratam das camadas de persistência

e de segurança da infraestrutura. No nível de infraestrutura têm-se os recursos propriamente, tais como recursos físicos de hardware e os recursos virtualizados com o objetivo de que a infraestrutura seja portátil e interoperável. Neste nível é implementado como uma linguagem comum de especificação que suporta o *.NET framework* ou *Mono*.

#### 7.4. Aplicações de Computação em Nuvem

Diversos tipos de aplicações baseadas em computação em nuvem estão disponíveis. Serviços de *webmail*, *sites*, sistemas de software como serviços em geral são alguns exemplos. Outras aplicações estão relacionadas à bioinformática [Deelman et al. 2008] e processamento de imagens. Outros exemplos mais recentes e concretos do uso de computação em nuvem são:

- O jornal NY Times usou EC2 e S3 da Amazon para converter 15 milhões de artigos de notícias para PDF (4 TB), para distribuição *online*, em questão de horas.
- A bolsa de valores eletrônica Nasdaq usa S3 para disponibilizar informações sobre o histórico de ações, sem onerar sua própria infraestrutura de TI
- Financeiras e seguradoras usam a nuvem para executar a avaliação de risco e calcular o custo dos empréstimos.

Neste tópico apresentamos uma aplicação executada em nuvem para detecção de arquivos maliciosos e indesejados.

##### 7.4.1. CloudAV

Aplicativos antivírus são ferramentas muito utilizadas para detectar e bloquear arquivos maliciosos e indesejados. No entanto, a eficácia há longo prazo de antivírus tradicionais é questionável. As aplicações de antivírus podem falhar para detectar muitas ameaças modernas e sua crescente complexidade resulta em vulnerabilidades que estão sendo exploradas por certos vírus. O CloudAV é um novo modelo para detecção de vírus em máquinas baseado no fornecimento de um antivírus como um serviço de computação em nuvem [Oberheide et al. 2008]. Este modelo utiliza uma técnica chamada de *N-version protection*, que permite a identificação de arquivos maliciosos e indesejados por múltiplos mecanismos de detecção em paralelo. Neste sentido, cada arquivo é analisado por diversos aplicativos antivírus, o que fornece uma melhor identificação de arquivos maliciosos.

A Figura 7.11 ilustra a arquitetura proposta pelo CloudAV. O CloudAV possui dois componentes arquiteturais: um agente e um serviço de rede. O agente é hospedado na máquina dos usuários e tem por função verificar a existência de novos arquivos nesta máquina e enviá-los para o serviço de rede. Estes arquivos podem ser originados por diversos meios, tecnologias ou aplicações. Quando um arquivo chega na máquina monitorada, o agente identifica este arquivo e o envia para o serviço de rede. O serviço de rede é composto por outros serviços: serviço de análise e serviço baseado em técnicas forenses. Este serviço de rede tem a responsabilidade de receber os arquivos enviados pelo agente e identificar os arquivos maliciosos ou que sejam dotados de conteúdos suspeitos. Para identificar estes arquivos, o serviço de rede utiliza o serviço de análise, sendo este composto por diversos aplicativos antivírus.

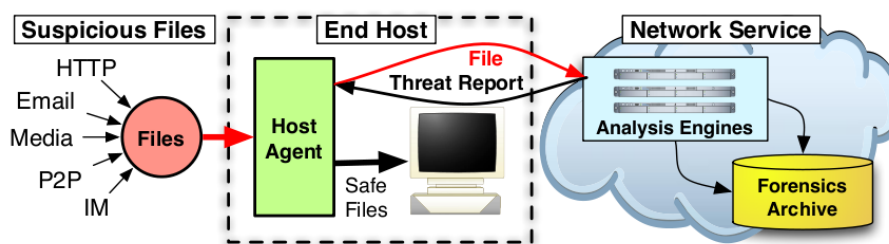


Figura 7.11. Arquitetura do CloudAV [Oberheide et al. 2008]

Após a análise dos arquivos, o serviço de rede notifica o agente sobre o *status* dos arquivos. O serviço baseado em técnicas forenses armazena informações sobre os arquivos analisados e fornece uma *interface* de consulta e de alerta para os administradores do CloudAV. Com base nas informações armazenadas, pode-se eliminar verificações desnecessárias, organizar procedimentos futuros e efetuar otimizações em outros processos de detecção. O CloudAV utiliza técnicas baseadas em *cache* para melhorar o desempenho no fluxo de envio de arquivos para detecção.

O CloudAV é executado em máquinas virtuais, de forma a garantir a escalabilidade. Atualmente o CloudAV pode ser utilizado em ambientes Windows, Linux, FreeBSD, Sendmail/postfix e Nokia Maemo, trabalha com dez aplicativos antivírus: Avast, AVG, BitDefender, ClamAV, F-Prot, F-Secure, Kaspersky, McAfee, Symantec e Trend Micro e os sistemas de software para detecção de comportamento Norman Sandbox e CWSandbox.

Para avaliar o CloudAV, utilizou-se dados reais coletados por um período superior a seis meses e um banco de dados com 7220 amostras de vírus detectados durante o período de um ano. Com base nos resultados, foi possível observar que o CloudAV é 35% mais eficaz na detecção de ameaças recentes em comparação com um único aplicativo antivírus e apresentou uma taxa de 98% de detecção em relação ao conjunto completo de dados analisados. Outros resultados podem ser encontrados em [Oberheide et al. 2008]. Com base nos resultados, percebeu-se que a utilização de um aplicativo antivírus como serviço apresenta alguns benefícios, dentre os quais podemos destacar: (a) o cliente não precisa manter seu antivírus atualizado, pois isso é realizado de forma automática pelo provedor do serviço (b) as máquinas clientes não desperdiçam ciclos de CPU, já que o processo de análise e detecção é efetuado pelo serviço (c) pode-se combinar estratégias para a detecção, melhorando a identificação de arquivos maliciosos.

## 7.5. Desafios

Computação em nuvem apresenta diversas vantagens, discutidas neste trabalho, mas também possui uma série de desafios a serem superados na utilização desse tipo de ambiente. A seguir destacamos alguns destes desafios.

### 7.5.1. Segurança dos Serviços de Dados

A computação em nuvem é um modelo que utiliza a Internet para disponibilizar seus serviços. Isso se torna mais complexo visto que os recursos computacionais utilizam

diferentes domínios de redes, sistemas operacionais, software, criptografia, políticas de segurança, entre outros. Questões de segurança devem ser consideradas para prover a autenticidade, confidencialidade e integridade. No que diz respeito à confiabilidade e responsabilidade, o provedor deve fornecer recursos confiáveis, especialmente se a computação a ser realizada é crítica e deve existir uma delimitação de responsabilidade entre o provedor e o usuário. Dessa forma, devem-se ter meios para impedir o acesso não autorizado a informações e que os dados sensíveis permaneçam privados, pois estes podem ser processados fora das empresas [Agrawal et al. 2009]. Em geral, cada sistema tem seu próprio modelo de dados e política de privacidade destes dados [Cooper et al. 2009]. Quando ocorre a movimentação de dados entre sistemas, deve-se garantir a privacidade dos dados mesmo com a mudança entre modelo de dados diferente e que aplicações multi-inquilino acessem dados de outras aplicações apenas de acordo com as políticas definidas.

Técnicas de criptografia podem ser utilizadas para garantir a privacidade dos dados. No entanto, estas técnicas têm implicações significativas de desempenho de consultas em SGBDs. Dessa forma, alternativas para a integração de técnicas de criptografia com SGBDs devem ser investigadas e desenvolvidas, já que a complexidade computacional da criptografia de dados aumenta o tempo de resposta da consulta. Em [Agrawal et al. 2009] é apresentado uma abordagem segura e escalonável para preservar a privacidade. Em vez de utilizar a criptografia, que é computacionalmente caro, é utilizada uma estratégia de distribuição dos dados em vários sítios do provedor e técnicas para acessar as informações de forma secreta e compartilhada.

### **7.5.2. Gerenciamento de Dados**

O gerenciamento de dados é considerado um ponto crítico no contexto de computação em nuvem. Os SGBDs relacionais não possuem escalabilidade quando milhares de sítios são considerados [Wei et al. 2009]. Assim, aspectos de armazenamento de dados, processamento de consultas e controle transacional tem sido flexibilizados por algumas abordagens para garantir a escalabilidade, mas ainda não existem soluções que combinem estes aspectos de forma a melhorar o desempenho sem comprometer a consistência dos dados [Abadi 2009]. Existe diversas abordagens para gerenciar dados em nuvens, dentre as quais podemos citar o Microsoft Azure e HBase [Brantner et al. 2008].

Um aspecto importante é o *trade-off* entre funcionalidades e custos operacionais enfrentados pelos provedores de serviços. Os serviços em nuvem para dados oferecem APIs mais restrita do que os SGBD relacionais, com uma linguagem minimalista de consulta e garantia de consistência limitada [Abouzeid et al. 2009]. Isso exige mais esforço de programação dos desenvolvedores, mas permite aos provedores construir serviços mais previsíveis e oferecerem SLA. De acordo com [Armbrust et al. 2009], a criação de um sistema de armazenamento que combina os diversos aspectos de computação em nuvem, de forma a aumentar a escalabilidade, a disponibilidade e consistência dos dados é um problema de pesquisa em aberto.

### **7.5.3. Autonomia**

A computação em nuvem é um sistema autônomo gerenciado de forma transparente para os usuários. Hardware e software dentro de nuvens podem ser automaticamente recon-

figurados, orquestrados e estas modificações são apresentadas ao usuário como uma imagem única. Essa autonomia é importante, pois reduz o custo de equipe de monitoramento do sistema tanto no âmbito centralizado quanto distribuído [Birman et al. 2009]. Comparados com sistemas tradicionais, é possível identificar três fatores complexos: intervenção humana limitada, alta alternância na carga de processamento e uma variedade de infraestruturas compartilhadas.

Na maioria dos casos, não existem administradores de sistemas para ajudar os desenvolvedores que acessam a nuvem, fazendo com que a plataforma seja automatizada ao máximo e os usuários podem variar a carga de trabalho habitual, necessitando de uma infraestrutura de virtualização eficaz. A gerência também é importante no contexto do desenvolvimento de tecnologia de auto sintonia. Assim sendo, técnicas adaptativas e *on-line* deverão ser desenvolvidas para tornar estes sistemas viáveis [Abounnaga et al. 2009].

#### **7.5.4. Disponibilidade de Serviços**

A disponibilidade de serviços permite aos usuários acessar e utilizar a nuvem onde e quando desejarem. Como se trata da Internet podem ocorrer atrasos e sistemas indisponíveis. Os ambientes de computação em nuvem devem prover alta disponibilidade. Para tanto, esses podem utilizar técnicas de balanceamento de carga dinâmico e composição de nuvens de forma a atender as necessidades dos usuários. Por exemplo, podem-se construir aplicações altamente disponíveis com a implantação de duas ofertas de nuvem diferentes. Caso uma das nuvens falhe, a outra nuvem continua a apoiar a disponibilidade das aplicações.

#### **7.5.5. Escalabilidade e Desempenho**

A escalabilidade foi uma das características fundamentais que conduziram ao surgimento da computação em nuvem. As nuvens de serviços e as plataformas oferecidas podem ser dimensionadas por vários fatores, tais como localizações geográficas, desempenho ou configurações. Apesar das limitações de rede e segurança, as soluções de computação em nuvem devem fornecer elevado desempenho, além de ser flexível para se adaptar diante de uma determinada quantidade de requisições. Como os ambientes de computação em nuvem possuem acesso público, é imprevisível e variável a quantidade de requisições realizadas, tornando mais complexo fazer estimativas e garantias de QoS.

#### **7.5.6. Descrição, Descoberta e Composição de Serviços**

Na computação em nuvem, vários modelos evoluíram rapidamente para aproveitar as tecnologias de software, plataformas de programação, armazenamento de dados e infraestrutura de hardware como serviços [Youseff et al. 2008]. Enquanto estes modelos se referem ao núcleo dos serviços de computação em nuvem, suas inter-relações têm sido ambíguas e a viabilidade de sua interoperabilidade é questionável. Além disso, cada serviço da nuvem tem *interfaces* e protocolos diferentes e é complexo para os usuários encontrar e compor serviços, visto que os diversos serviços estão dispersos na Internet e possuem características distintas. Por exemplo, suponha que um usuário necessite de um serviço de processamento e outro de armazenamento para persistir os dados processados. Uma alternativa para o usuário seria fazer uma busca exaustiva. Contudo, como existe uma grande quantidade de serviços, isso pode se tornar inviável. Além disso, ainda seria necessário

compor os serviços de processamento e armazenamento, o que seria outra dificuldade.

Dessa forma, um desafio é desenvolver técnicas eficazes para descrever, descobrir e compor serviços na nuvem de forma a auxiliar os usuários em suas tarefas. Ontologias podem ser utilizadas para a organização do domínio de conhecimento de computação em nuvem, seus componentes e suas relações, ajudando na descrição e descoberta de serviços em nuvem [Youseff et al. 2008], assim como na composição de novos serviços a partir dos serviços existentes. Isso ajudará no projeto de serviços com interoperabilidade entre diferentes provedores, proporcionando melhorias na qualidade dos serviços.

#### **7.5.7. Licenciamento de Software**

Embora pesquisas em computação tenham investigado vários modelos econômicos de infraestrutura computacional durante a última década, a computação em nuvem tem uma abordagem mais aplicada aos negócios e relacionada ao custo. Assim, a computação em nuvem apresenta diversos modelos de preço, sendo estes organizados em três grupos: preço diferenciado, preços por unidade e assinatura de serviços básicos. Preço diferenciado é o modelo adotado pela Amazon, onde os serviços são oferecidos em vários níveis de especificações, tais como alocação de memória e tipo de CPU, informações de SLA e o valor cobrado é um preço específico por unidade de tempo.

Preço por unidade é normalmente aplicado a dados transferidos ou ao uso de memória. Este modelo é mais flexível do que o de preço diferenciado, já que permite aos usuários personalizarem a alocação de memória de seus sistemas baseados nas necessidades de aplicações específicas. O modelo de assinatura de serviços básicos é o modelo de preços mais amplamente utilizado, permitindo aos usuários preverem suas despesas previamente na utilização de um serviço. Contudo, este modelo não tem a precisão em cobrar dos usuários o que eles têm realmente utilizado.

#### **7.5.8. Integração de Serviços**

Com a evolução da computação em nuvem, as empresas necessitam integrar os diferentes ambientes de TI, pois estas empresas utilizam modelos híbridos, nos quais os sistemas instalados possam interagir com diversos provedores. Contudo, não existem padrões de integração de sistemas de computação em nuvem. XML pode ser uma alternativa para mover dados entre ambientes em nuvem, mas os sistemas também precisam gerenciar dados localmente. A utilização de APIs pode auxiliar neste processo de integração. Por exemplo, as APIs da Amazon estão se tornando um padrão de fato para serviços sob demanda [OpenCloud 2010]. Contudo, a quantidade de tecnologias envolvidas é muito grande, tornando-se um desafio padronizar as diversas *interfaces* e serviços, bem como fornecer interoperabilidade entre recursos heterogêneos. Desempenho e a evolução dos serviços são aspectos importantes na integração de nuvem, pois as aplicações possuem requisitos de QoS e as evoluções são constantes. Dessa forma, o uso de tecnologias de integração de dados, serviços e linguagens devem ser utilizadas e adaptadas no contexto da computação em nuvem.



### 7.5.9. Avaliação de Serviços em Nuvem

Existem muitos serviços disponíveis em nuvem e algumas iniciativas para medir avaliar serviços específicos [Binnig et al. 2009]. Assim, o desenvolvimento de um *benchmark* de propósito geral, que permita avaliar diversos tipos de serviços é um ponto importante. Este *benchmark* deve ser composto de uma ferramenta para gerar cargas de trabalho, monitorar o desempenho e métricas para calcular o custo por usuário em uma determinada unidade de tempo [Yigitbasi et al. 2009]. Outra alternativa para a avaliação de nuvem consiste no desenvolvimento de sistemas de simulação, tais como o CloudSim [Buyya et al. 2009a].

### 7.6. Conclusões

A computação como um serviço está finalmente emergindo e as empresas podem prestar serviços diretamente aos usuários por meio da Internet de acordo com as suas necessidades. Neste contexto, a computação em nuvem é um paradigma que está cada vez mais popular. Diversas empresas apresentaram suas iniciativas na promoção da computação em nuvem. A comunidade científica também tem apresentado algumas iniciativas, principalmente com foco em suas necessidades.

Este trabalho apresentou os principais aspectos de computação em nuvem e alguns conceitos e tecnologias relacionadas com estes ambientes. Foi possível perceber que a computação em nuvem ainda não tem uma definição clara e completa na literatura, mas que existe um grande esforço neste sentido. Aplicações baseadas em computação em nuvem também foram apresentadas.

Foram discutidos alguns desafios de pesquisa importantes, tais como segurança, gerenciamento de dados, disponibilidade, padronização e aspectos da própria utilização da computação em nuvem. É importante ressaltar que, várias soluções, existentes em outros modelos computacionais, que resolvem ou atenuem estes desafios, podem ser aplicadas em ambientes de computação em nuvem. Estes desafios geram oportunidades de pesquisa que devem ser superados, de forma que computação em nuvem seja amplamente aceita e utilizada por todos.

### Referências

- [Abadi 2009] Abadi, D. J. (2009). Data management in the cloud: Limitations and opportunities. *IEEE Data Eng. Bull.*, 32:3–12.
- [Abounaga et al. 2009] Abounaga, A., Salem, K., Soror, A. A., Minhas, U. F., Kokosiellis, P., and Kamath, S. (2009). Deploying database appliances in the cloud. *IEEE Data Eng. Bull.*, 32(1):13–20.
- [Abouzeid et al. 2009] Abouzeid, A., Bajda-Pawlikowski, K., Abadi, D. J., Rasin, A., and Silberschatz, A. (2009). Hadoopdb: An architectural hybrid of mapreduce and dbms technologies for analytical workloads. *PVLDB*, 2(1):922–933.
- [Agrawal et al. 2009] Agrawal, D., Abbadi, A. E., Emekci, F., and Metwally, A. (2009). Database management as a service: Challenges and opportunities. *Data Engineering, International Conference on*, 0:1709–1716.
- [Armbrust et al. 2009] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., Lee, G., Patterson, D. A., Rabkin, A., Stoica, I., and Zaharia, M. (2009). Above the

clouds: A berkeley view of cloud computing. Technical report, EECS Department, University of California, Berkeley.

- [Azure 2010] Azure (2010). *Microsoft Azure*. <http://www.microsoft.com/azure/>.
- [Binnig et al. 2009] Binnig, C., Kossmann, D., Kraska, T., and Loesing, S. (2009). How is the weather tomorrow?: towards a benchmark for the cloud. In *DBTest '09: Proceedings of the Second International Workshop on Testing Database Systems*, pages 1–6, New York, NY, USA. ACM.
- [Birman et al. 2009] Birman, K., Chockler, G., and van Renesse, R. (2009). Toward a cloud computing research agenda. *SIGACT News*, 40(2):68–80.
- [Brantner et al. 2008] Brantner, M., Florescu, D., Graf, D., Kossmann, D., and Kraska, T. (2008). Building a database on s3. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data - SIGMOD '08*, page 251, New York. ACM Press.
- [Buyya et al. 2009a] Buyya, R., Ranjan, R., and Calheiros, R. N. (2009a). Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities. *CoRR*, abs/0907.4878.
- [Buyya et al. 2009b] Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., and Brandic, I. (2009b). Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Gener. Comput. Syst.*, 25(6):599–616.
- [Chang et al. 2006] Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., Chandra, T., Fikes, A., and Gruber, R. E. (2006). Bigtable: a distributed storage system for structured data. In *OSDI '06: Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation*, pages 15–15, Berkeley, CA, USA. USENIX Association.
- [Ciurana 2009] Ciurana, E. (2009). *Developing with Google App Engine*. Apress, Berkely, CA, USA.
- [Cooper et al. 2009] Cooper, B. F., Baldeschwieler, E., Fonseca, R., Kistler, J. J., Narayan, P. P. S., Neerdaels, C., Negrin, T., Ramakrishnan, R., Silberstein, A., Srivastava, U., and Stata, R. (2009). Building a cloud for yahoo! *IEEE Data Eng. Bull.*, 32(1):36–43.
- [Dean and Ghemawat 2004] Dean, J. and Ghemawat, S. (2004). Mapreduce: simplified data processing on large clusters. In *OSDI'04: Proceedings of the 6th conference on Symposium on Operating Systems Design & Implementation*, pages 10–10, Berkeley, CA, USA. USENIX Association.
- [Deelman et al. 2008] Deelman, E., Singh, G., Livny, M., Berriman, B., and Good, J. (2008). The cost of doing science on the cloud: the montage example. In *SC '08: Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, pages 1–12, Piscataway, NJ, USA. IEEE Press.
- [Hadoop 2010] Hadoop (2010). *Apache Hadoop*. <http://hadoop.apache.org>.
- [Jacobs and Aulbach 2007] Jacobs, D. and Aulbach, S. (2007). Ruminations on multi-tenant databases. In *BTW*, volume 103 of *LNI*, pages 514–521. GI.
- [Liu et al. 2007] Liu, S., Liang, Y., and Brooks, M. (2007). Eucalyptus: a web service-enabled e-infrastructure. In *CASCON '07: Proceedings of the 2007 conference of the center for advanced studies on Collaborative research*, pages 1–11, New York, NY, USA. ACM.

- [Marinos and Briscoe 2009] Marinos, A. and Briscoe, G. (2009). Community cloud computing. In *First International Conference Cloud Computing, CloudCom*, volume 5931 of *Lecture Notes in Computer Science*, pages 472–484. Springer.
- [Mell and Grance 2009] Mell, P. and Grance, T. (2009). *Draft NIST Working Definition of Cloud Computing*. National Institute of Standards and Technology. <http://csrc.nist.gov/groups/SNS/cloud-computing>.
- [Oberheide et al. 2008] Oberheide, J., Cooke, E., and Jahanian, F. (2008). Cloudiv: N-version antivirus in the network cloud. In *SS'08: Proceedings of the 17th conference on Security symposium*, pages 91–106, Berkeley, CA, USA. USENIX Association.
- [Olston et al. 2008] Olston, C., Reed, B., Srivastava, U., Kumar, R., and Tomkins, A. (2008). Pig latin: a not-so-foreign language for data processing. In *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1099–1110, New York, NY, USA. ACM.
- [OpenCloud 2010] OpenCloud (2010). *The Open Cloud Manifesto*. <http://www.opencloudmanifesto.org>.
- [Robinson 2008] Robinson, D. (2008). *Amazon Web Services Made Simple: Learn how Amazon EC2, S3, SimpleDB and SQS Web Services enables you to reach business goals faster*. Emereo Pty Ltd, London, UK, UK.
- [Salesforce 2010] Salesforce (2010). *Salesforce*. <http://www.salesforce.com/>.
- [Soror et al. 2010] Soror, A. A., Minhas, U. F., Aboulnaga, A., Salem, K., Kokosielis, P., and Kamath, S. (2010). Automatic virtual machine configuration for database workloads. *ACM Trans. Database Syst.*, 35(1):1–47.
- [Vaquero et al. 2009] Vaquero, L. M., Roderio-Merino, L., Caceres, J., and Lindner, M. (2009). A break in the clouds: towards a cloud definition. *SIGCOMM Comput. Commun. Rev.*, 39(1):50–55.
- [Vecchiola et al. 2009] Vecchiola, C., Chu, X., and Buyya, R. (2009). *Aneka: A Software Platform for .NET-based Cloud Computing*, pages 267–295. In: W. Gentzsch, L. Grandinetti, G. Joubert (Eds.). *High Speed and Large Scale Scientific Computing*. IOS Press, Amsterdam, Netherlands.
- [Wei et al. 2009] Wei, Z., Pierre, G., and Chi, C.-H. (2009). Scalable transactions for web applications in the cloud. In *Euro-Par*, pages 442–453.
- [Yigitbasi et al. 2009] Yigitbasi, N., Iosup, A., Epema, D., and Ostermann, S. (2009). C-meter: A framework for performance analysis of computing clouds. In *CCGRID '09: Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, pages 472–477, Washington, DC, USA. IEEE Computer Society.
- [Youseff et al. 2008] Youseff, L., Butrico, M., and Da Silva, D. (2008). Toward a unified ontology of cloud computing. In *Grid Computing Environments Workshop, 2008. GCE '08*, pages 1–10.