1

Desmistificando a Computação em Nuvem

Alexandre Carissimi - asc@inf.ufrgs.br[1]

Resumo:

A computação em nuvem fornece poder computacional como um serviço público, análogo ao que existe para telefonia e energia elétrica. Os recursos computacionais são alocados por demanda e os usuários são tarifados apenas pelo o que consomem. Este trabalho apresenta os principais conceitos, os modelos de serviços (IaaS, PaaS, SaaS) e os modelos de implantação (publica, privada, comunitária e híbrida) da computação em nuvem. Plataformas reais como Amazon, Google, Azure, Eucalyptus, OpenStack e CloudStack são usadas para mostrar a aplicação desses conceitos. Por fim, são discutidos os principais riscos e benefícios da computação em nuvem, assim como alguns dos eixos de pesquisa e desenvolvimento nessa área.

[1]Alexandre da Silva Carissimi é Doutor em Informática pelo Institut National Polytechnique de Grenoble (INPG), França (1999), Mestre em Ciência da Computação pelo Programa de Pós-Graduação em Computação da UFRGS (1989) e formado em Engenheira Elétrica pela UFRGS (1985). Atua como Professor no Departamento de Informática Aplicada da UFRGS desde 1989, tendo sido, no decorrer desse período, responsável por ministrar diversas disciplinas nas áreas de introdução à informática, arquitetura de computadores, sistemas operacionais, sistemas distribuídos e redes de computadores. É instrutor da Escola Superior de Redes (ESR), iniciativa da Rede Nacional de Pesquisa (RNP), que mantém convênio com a UFRGS, onde ministra cursos relacionados a administração de redes de computadores, sistemas operacionais e virtualização. É coautor de três livros publicados pela Editora Bookman, dentro da Série Didática da UFRGS: Sistemas Operacionais (vol. 11), Sistemas operacionais e Programação Concorrente (vol. 14) e Redes de computadores (vol. 20). Já apresentou diversos minicursos e palestras em congressos e escolas (ERAD, ERRC, SBRC, WSCAD, CLEI), além de várias participações em semanas acadêmicas de universidades do Rio Grande do Sul (UFRGS, UFSM e UFPel). Com relação a áreas de pesquisa, atua na área de processamento paralelo e distribuído, em especial no estudo e desenvolvimento de suportes de execução (runtimes) e middlewares.

1.1. Introdução

Em 1961, o cientista americano John McCarthy, do MIT, criador do termo inteligência artificial, previu o que hoje conhecemos como computação em nuvem: "*se os computadores, da forma como eu imagino, se tornarem os computadores do futuro, então a computação poderá ser organizada como um serviço público, assim como a telefonia é... Cada assinante pagará apenas pelos recursos que ele realmente utilizará, mas ele terá acesso a todos os recursos oferecidos pelas linguagens de programação de um grande sistema... Alguns assinantes poderão oferecer serviços a outros assinantes... A computação como um serviço público poderá ser base de uma nova e importante indústria*".

Quase quatro décadas depois, em meados dos anos 90, a computação em grade, ou *grid computing* [Foster, Kesselman, 1998], começava a colocar em prática a visão de McCarthy. O termo *grid computing* é uma metáfora para dizer que o acesso a um poder computacional pode ser obtido com a mesma facilidade que se tem acesso à energia elétrica em nossas residências: basta plugar um equipamento às tomadas para usar a energia disponibilizada por uma concessionária de distribuição de energia (*power grid*, em inglês) e pagarmos de acordo com nosso consumo. De maneira mais formal, a computação em grade pode ser definida como um sistema distribuído composto por uma grande quantidade de recursos computacionais heterogêneos, fracamente acoplados e geograficamente dispersos em diferentes domínios administrativos. O objetivo da computação em grade é fornecer ao usuário a ilusão de que ele tem um computador virtual de grande porte à sua disposição. Essa ilusão é proporcionada através de *middlewares* desenvolvidos para alocar e gerenciar os recursos computacionais disponíveis e fornecer acesso à grade computacional aos usuários.

Entretanto, na prática, o modelo da infraestrutura física das grades computacionais resultou em alguns inconvenientes. Primeiro, como os recursos são interconectados através de redes de longa distância, a latência e a banda passante disponíveis se tornaram um problema para aplicações interativas, ou que exigiam comunicação de um grande volume de dados. Apesar disso, várias aplicações científicas como simulações aerodinâmicas, desdobramento de proteínas, modelos de mercado financeiro e de climatologia, entre outras, por serem computacionalmente intensivas, tiraram proveito das grades computacionais. Segundo, como as grades computacionais são formadas por recursos heterogêneos (hardware e software), o desenvolvimento de aplicações portáveis, assim como o escalonamento de tarefas, balanceamento de carga, e otimizações da utilização dos recursos são complexas de serem feitas e se tornaram desafios de pesquisa. Por fim, o fato dos recursos estarem em domínios administrativos diferentes trouxe problemas relacionados com segurança da informação como a gerência dos recursos computacionais, as questões de autenticação e de autorização, entre outros. Por isso, o uso e a disseminação das grades computacionais acabaram ficando restritos a centros de pesquisas e universidades.

Entretanto, as lições aprendidas com as grades computacionais foram importantes para a evolução da ideia de computação como um serviço público. Aliado a essas lições, houve o avanço tecnológico na área de comunicação de dados, com o aumento da capacidade de banda passante e da capilaridade da rede, e da redução de custos e de latência. No entanto, o avanço tecnológico mais importante se deu na área da virtualização. Esses fatores lançaram os pilares do que hoje conhecemos como computação em nuvem (*cloud computing*).

A computação em nuvem é a implantação do conceito de computação como um

serviço público e, em relação às grades computacionais, ela apresenta duas novidades. Inicialmente, a computação em nuvem é um modelo de negócio onde um usuário paga apenas pelo o que consome de recursos (modelo *pay-as-you-go*) e o provedor do serviço mantém uma infraestrutura física (*data center*) que ele amortiza os custos de manutenção e investimentos compartilhando e alugando seus recursos a diferentes usuários. Segundo, a virtualização do hardware, dos ambientes de desenvolvimento e da execução de serviços. O emprego da virtualização o uso de recursos por demanda permitindo que um usuário, se necessário, solicite, e pague, por mais recursos. Da mesma forma, um usuário pode devolver os recursos em excesso e não pagar mais pelo seu uso. Essa flexibilidade é denominada de elasticidade e representa uma resiliência às aplicações computacionais, isso é, uma capacidade de adaptação às modificações de carga computacional.

O objetivo principal deste minicurso é desmistificar a computação em nuvem. Para isso, este minicurso está organizado em oito seções com esta introdução. A seção 1.2 discute a ideia fundamental da computação em nuvem que, na prática, é um modelo de negócio baseado no conceito de elasticidade. Assim, a computação em nuvem provê serviços computacionais em uma infraestrutura dedicada para tal. Para melhor estruturá-los, o NIST (*National Institut of Standards and Technology*) define três modelos básicos de serviços em nuvem (*IaaS*, *PaaS* e *SaaS*) e quatro arquiteturas possíveis para a sua implantação (pública, privada, comunitária e híbrida). Essas definições são abordadas na seção 1.3. A seção 1.4 apresenta a arquitetura genérica existente em uma solução de computação em nuvem, sob o ponto de vista de camadas de software e de hardware, e suas tecnologias subjacentes. A seção 1.5 é composta por estudos de caso, onde se busca mostrar soluções existentes de computação em nuvem e os modelos serviços oferecidos. São citados os casos da Amazon, do Google, Microsoft Azure e de soluções em software livre (Eucalytpus, CloudStack e OpenStack). A computação em nuvem, no entanto, traz consigo novos desafios que devem ser avaliados antes de ser empregada. Então, a seção 1.6, discute alguns dos riscos e benefícios existentes no emprego dessa tecnologia, e a seção 1.7 fornece, em grandes linhas, os principais eixos de pesquisa e desenvolvimento na área da computação em nuvem. Por fim, a seção 1.8, conclusão, que resume os principais pontos apresentados neste minicurso.

1.2. As bases da computação em nuvem

A computação em nuvem é, basicamente, a utilização massiva da virtualização para a criação de um modelo de negócio. A inovação da computação em nuvem está em como as tecnologias já existentes, e comuns no nosso dia a dia, são empregadas para prover um novo modelo de acesso a recursos computacionais. Isso foi, em parte, consequência natural da evolução da computação.

Inicialmente, o acesso ao poder computacional passou por vários ciclos entre sistemas centralizados e sistemas descentralizados. No início dos anos 60, o modelo existente era aquele de um grande e único mainframe que centralizava o poder de cálculo computacional e de armazenamento de dados. O acesso a esses recursos se fazia através de terminais. Na sequência, surgiram os computadores pessoais (PCs) que levaram a uma descentralização do poder de processamento e de armazenamento, permitindo aos usuários executarem e armazenarem dados em seus próprios computadores. A interconexão em rede levou os computadores pessoais a compartilharem recursos como os servidores

de impressão e de arquivos, entre outros. Surgiu então a ideia de oferecer serviços de forma remota som um modelo cliente-servidor, onde o servidor executa, de forma centralizada, um serviço que é oferecido aos clientes via rede. Esse é o modelo empregado nos serviços Internet como o correio eletrônico, web, DNS, entre outros.

A evolução do modelo cliente-servidor foi permitir aos computadores serem clientes e servidores ao mesmo tempo, surgindo assim os sistemas *peer-to-peer* (P2P), um novo ciclo descentralizado. Uma desvantagem dos sistemas P2P foi a sua dependência tecnológica a um tipo solução P2P, ou seja, não era portável. O uso de padrões sempre se impôs como uma forma de permitir portabilidade e tratar a heterogeneidade de sistemas e isso aconteceu com a disseminação dos serviços web (*web services*), do protocolo SOAP, do uso de XML, iniciando um novo ciclo centralizado, baseado no modelo cliente-servidor. Essa é a época do modelo de hospedagem de aplicações com as soluções ASP, .Net, JavaBeans.

Entretanto, o modelo de hospedagem de aplicações possui uma arquitetura e pilha tecnológica customizada para um determinado nicho de mercado. Um tipo de arquitetura bem disseminada é conhecida pelo acrônimo LAMP (Linux, Apache, MySQL e PHP) que permite a hospedagem de aplicações, mas a um custo de implantação e manutenção. Simultaneamente, com o advento do conceito da Web 2.0, por Tim Oreilly, em 2005, a Internet aparece como plataforma computacional para prover serviços como *blogs*, *wikis*, redes sociais, compartilhamento de fotos e vídeos, sistemas de trocas de mensagens etc. Na prática, nasce a ideia de oferecer software como um serviço, mas há ainda um problema: sobre qual plataforma executar esses serviços? A solução é também oferecer a plataforma como um serviço. E sobre qual infraestrutura física se executaria essa plataforma? A resposta, novamente, é oferecer essa infraestrutura física como um recurso. Esse é o berço da computação em nuvem, isso é, oferecer acesso a software (aplicações), a plataformas e a hardware como serviços disponíveis na Internet.

A capacidade de prover uma infraestrutura de computação como serviço foi beneficiada pela evolução e amadurecimento do conceito de virtualização. Inicialmente, as máquinas virtuais permitem a criação de um ambiente de execução, possivelmente com um conjunto de bibliotecas e sistemas operacionais próprios, onde uma aplicação pode executar sobre uma plataforma de hardware comum junto com outras máquinas virtuais. Dessa forma, a virtualização proporciona um alto grau de portabilidade e de flexibilidade permitindo que várias aplicações, de sistemas operacionais diferentes, executem em um mesmo hardware. Mas, a virtualização não se resume apenas as máquinas virtuais, é possível criar e definir redes de interconexão e sistemas de armazenamento virtuais.

A execução de múltiplas instâncias de máquinas virtuais sobre um mesmo hardware físico é denominada de consolidação de servidores [Smith, Nair, 2005], é especialmente interessante em provedores de hospedagem e *data centers*. Assim, ao invés de alocar uma máquina física para cada cliente hospedado e sua solução particular, o provedor de hospedagem pode instanciar uma máquina virtual. Em um *data center*, se os serviços são executados em máquinas virtuais, na situação em que uma máquina física está sob baixa carga, é possível migrar as máquinas virtuais para outras máquinas, balanceando a carga e desligando as de baixa carga. Essas ações reduzem os custos de instalação e manutenção da estrutura física tais como espaço, energia elétrica, cabeamento, refrigeração, suporte e manutenção a vários sistemas, etc. Essa economia de energia faz com que a virtualização e, por consequência, a computação em nuvem, sejam mecanismos para atingir o que se denomina computação verde (*green computing*).

A flexibilidade de instanciar e migrar máquinas virtuais não é proveitosa apenas

para o provedor de computação em nuvem. O cliente também se beneficia disso de duas formas. A primeira é o reflexo da redução do custo do provedor que pode ser repassado ao cliente. A segunda, e mais importante, é que, se durante a execução de uma aplicação for atingido um pico de utilização, provocando gargalos ou a degradação de desempenho, o cliente pode solicitar mais máquinas virtuais para auxiliar o processamento durante esse pico de demanda. Após a normalização da situação, essas máquinas virtuais podem ser liberadas. O cliente paga apenas pela utilização efetiva dessas máquinas. Essa capacidade de alocar recursos conforme a demanda é denominada de elasticidade.

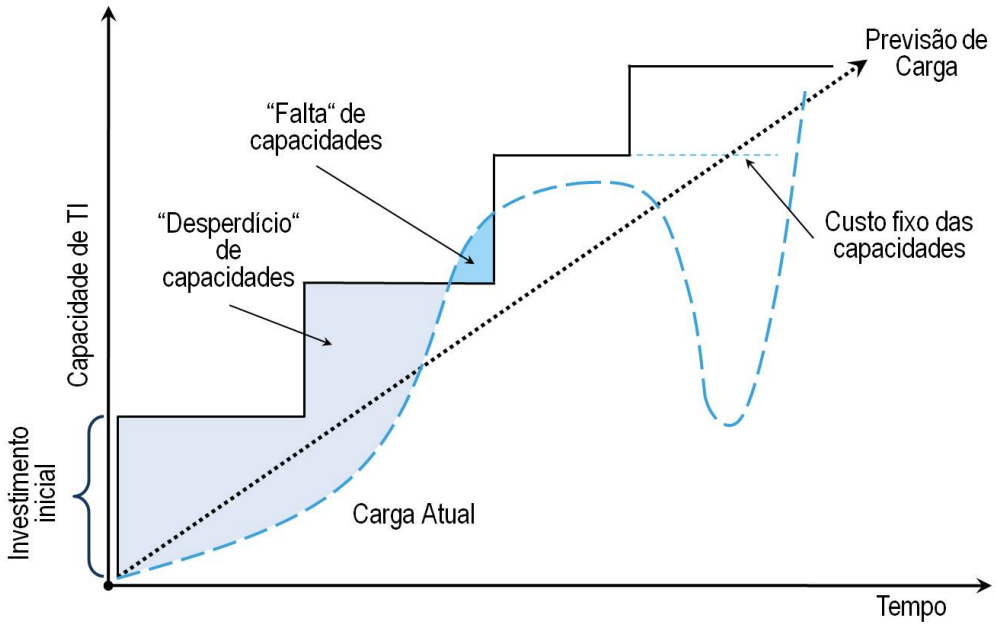

Figura 1.1: Alocação de recursos: previsão, investimento e elasticidade; Fonte: Coelho, 2010.

A elasticidade é o conceito fundamental no modelo de negócios da computação em nuvem. Para melhor compreendê-lo é possível imaginar o exemplo de uma loja de comércio eletrônico. Em um modelo de computação clássico, essa loja deveria adquirir capacidade de processamento para atender sua previsão de crescimento (carga no sistema). Isso é representado pela linha pontilhada no gráfico da figura 1.1. Para não perder clientes, a loja deve ser capaz de atender todos os clientes que desejam fazer compras sem apresentar lentidão. Isso é obtido investindo em equipamentos que atendam a carga prevista. A capacidade computacional decorrente desse investimento é representada pela linha em formato de escada (figura 1.1). A carga atual é aquela gerada pelos clientes efetivos da loja e está representada pela linha tracejada. Nesse caso, é possível ver que há situações em que os recursos computacionais estão superdimensionados em relação à carga atual e, em outras, subdimensionados. Ainda, o comércio eletrônico está sujeito a sazonalidades típicas de mercado como Natal, dia das mães, dias dos namorados, etc. Assim, por exemplo, para atender a demanda de Natal, ou se investe em equipamentos que ficarão ociosos o resto do ano, ou não se investe e se convive com a eventual perda e ou insatisfação dos clientes. No entanto, se essa loja adotar uma solução em computação em nuvem, ela pode "alugar os recursos necessários" conforme a demanda de seus clientes. Assim, além de continuar oferecendo bons serviços nos picos de demanda ela

pode economizar recursos nos períodos de baixa procura. A alocação e a liberação dos recursos devem ser feitas para manter os recursos computacionais suficientes para atender a demanda da carga atual.

O exemplo da loja de comércio eletrônico destaca duas características da computação em nuvem. Primeira, em uma solução baseada em computação em nuvem, o valor e o investimento estão na informação e no negócio e não na propriedade. O mais importante para loja de comércio eletrônico é o negócio em si e não o parque tecnológico necessário. A filosofia é "melhor alugar, quando necessário, que possuir" elimina os investimentos em infraestrutura própria. Segunda, o modelo "pague o quanto usa" (*pay-as-you-go*), isso é, o cliente paga apenas pelos recursos que ele efetivamente utiliza. Assim, a computação em nuvem pode ser vista como a disponibilidade de um conjunto de recursos computacionais (virtualmente) infinito, oferecido por um terceiro, em um modo *self-service*, via um modelo *pay-as-you-go*. Por modo *self-service* entende-se o fato que, idealmente, o cliente deve ser capaz de alocar os recursos da nuvem de forma simples, através da Internet, sem a intervenção do provedor do serviço.

1.3. Modelo NIST para computação em nuvem

Atualmente, há muito marketing e abuso no uso no emprego da terminologia da computação em nuvem em função do interesse comercial de muitos provedores desse tipo de serviço. Na seção anterior, foi dito que a computação em nuvem é um modelo que pretende prover a computação como um serviço, ou seja, prover recursos computacionais – processamento e armazenamento – por demanda. Esses recursos devem estar disponíveis para os usuários sem muita, idealmente, nenhuma, interação com o provedor de serviços, de forma semelhante ao que acontece com os serviços de telefonia e energia elétrica. Para identificar claramente o que é a computação em nuvem e fugir das armadilhas do marketing, o *National Institut of Standards and Technology* (NIST) [Mell, Grace, 2011], definiu um modelo para computação em nuvem com três camadas: atributos de serviços ou características essenciais, modelo de serviços e modelos de implantação. O modelo NIST para definição de computação em nuvem é ilustrado na figura 1.2 e será detalhado nas próximas seções.

1.3.1. Atributos de serviços

A definição do NIST elencou cinco características essenciais, denominados de atributos de serviços, que a computação em nuvem deve ter: serviços por demanda na forma *self-service*, acesso universal (ubíquo), conjunto de recursos (*pool*), contabilização de recursos e elasticidade.

O uso de serviços por demanda significa que um cliente de computação em nuvem deve ser capaz de alocar os recursos computacionais, a medida do necessário, sem precisar da intervenção e interação direta com equipes de tecnologia da informação do provedor de computação em nuvem.

Os recursos virtuais da nuvem devem estar disponíveis na rede através do uso de protocolos e métodos para permitir o acesso independente de plataforma. O objetivo é fornecer acesso universal e ubíquo a diversas plataformas computacionais, incluindo diferentes sistemas operacionais e equipamentos (notebooks, desktops, dispositivos móveis).

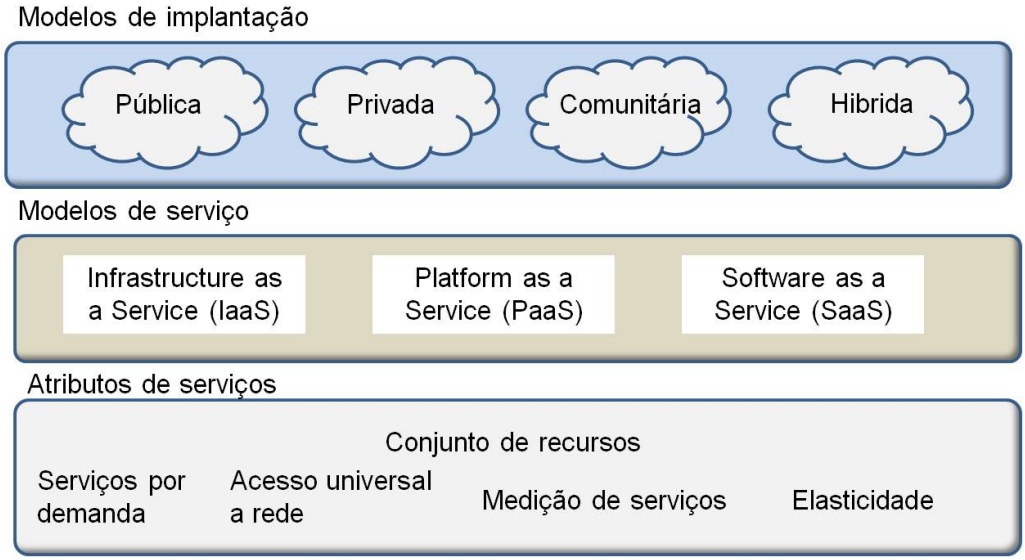

Figura 1.2: Modelo NIST para computação em nuvem; Fonte: Sosinsky, 2011.

O provedor de computação em nuvem deve fornecer um conjunto de recursos (*pool*) capaz de ser empregado por vários clientes simultaneamente. Os recursos físicos e virtuais são alocados e liberados de forma dinâmica. O *pool* de recursos deve abstrair, para o cliente final, os aspectos de máquinas virtuais, processamento, memória, armazenamento, banda passante de rede e interconexão.

Os recursos da nuvem devem ser disponibilizados rapidamente e, conforme a demanda, adicionando mais recursos do mesmo tipo (*scale out*) ou mais poder computacional (*scale up*). Essa adição de recursos pode ser feita de forma automática ou manual. Ainda, sob o ponto de vista do cliente, ele deve ter a impressão que tem acesso a recursos ilimitados que podem ser comprados a qualquer momento, em qualquer quantidade. Como visto anteriormente, essa característica é denominada de elasticidade.

Por fim, um sistema de computação em nuvem deve ser capaz de medir e disponibilizar relatórios de uso a seus clientes de acordo com alguma métrica. Essas métricas podem variar entre tempo de processamento, quantidade de máquinas virtuais alocadas, uso de banda passante da rede, capacidade de memória e armazenamento usados, entre outras. Ainda, o sistema deve prever possibilidade de ter o uso, o consumo e a tarifação de recursos auditados.

1.3.2. Modelos de serviços

Os modelos de serviços definidos pelo NIST são três: software como serviço (*Software as a Service – SaaS*), Plataforma como serviços (*Platform as a Service – PaaS*) e infraestrutura como serviço (*Infrastructure as a Service – IaaS*). De uma forma genérica, o que diferencia os tipos de serviços entre si é o tipo de cliente final ao qual cada um se destina.

O modelo *SaaS* provê serviços de computação para o usuário final. Ele consiste basicamente no aluguel de uma aplicação já operacional sem a necessidade que os usuários passem por fases de aquisição, configuração e manutenção dessa aplicação. No *SaaS*,

o software está pronto para a ser usado. Os serviços oferecidos executam em plataformas computacionais concebidas para serem utilizadas por um grande número de usuários, das mais diferentes organizações e localizações geográficas. A forma de acesso ao serviço é através da Internet, usualmente a partir de um navegador. São exemplos desses serviços o Google Drive, o Gmail, Hotmail, Dropbox, SQL Azure e Oracle on Demand. Além disso, existem softwares disponíveis para e-Science onde cientistas, a partir de navegados web, submetem seus dados a simulações e análise de serviços. Nesse modelo, o usuário enxerga apenas o software que precisa usar e não tem conhecimento de onde, realmente, estão localizados os recursos empregados, nem quais linguagens de programação foram usadas no desenvolvimento do serviço, nem o sistema operacional e o hardware sobre o qual a aplicação executa.

No modelo *PaaS*, o cliente final são os desenvolvedores de aplicações em software. Esse modelo fornece a seus clientes um ambiente completo composto por todos os recursos necessários para o desenvolvimento de software em uma ou mais linguagens de programação tais como compiladores, depuradores, bibliotecas e um sistema operacional. É importante salientar que o ambiente de desenvolvimento pode ter limitações quanto às linguagens de programação, gerenciadores de banco de dados, sistema operacional, etc, ou seja, ele não é uma plataforma completa genérica, mas sim uma plataforma completa para uma determinada finalidade. Além do desenvolvimento de aplicações, no modelo *PaaS*, a plataforma pode ser "alugada" para hospedar sites web ou para prover serviços do tipo SaaS. São exemplos de *PaaS*, o Windows Azure Platform, Force.com, Google AppEngine, entre outros.

Por fim, o *IaaS*, que é um modelo de provimento de serviços destinado a uma equipe de tecnologia da informação (TI). Aqui é feito o fornecimento de um sistema computacional composto por processadores, memória e armazenamento. Nesse caso, é necessário instalar e configurar, por conta própria, todos os recursos necessários a utilização desse sistema, tais como, compiladores, banco de dados e, inclusive, o próprio sistema operacional. Em relação aos modelos de serviços anteriores, a tarifação no *IaaS* considera a quantidade de recurso que é destinado ao cliente, durante um certo período de tempo, sem considerar se está ou não sendo efetivamente empregado. Alguns exemplos de provedores *IaaS* são o Windows Azure, Amazon Elastic Compute Cloud (EC2), Citrix, Eucalyptus, e o Rackspace Cloud.

É interessante comentar que o marketing define muitos outros serviços, mas que, na verdade, nada mais são que especializações do modelo SaaS. Por exemplo, é possível citar serviços como *Information as a Service*, *Integration as a Service*, *Security as a Service*, *Testing as a Service* e até mesmo *Fax as a Service* para o envio de faxes. Tal criatividade do marketing cunhou o modelo *XaaS* (*x as a service*), onde x significa "alguma coisa".

1.3.3. Modelos de Implementação

O NIST define ainda quatro modelos para a implantação de nuvens, a saber: pública, privativa, comunitária e híbrida. Essencialmente, dois critérios são usados para essa classificação, o objetivo ao qual a nuvem se propõe e a natureza da localização de seus recursos computacionais.

As nuvens públicas são aquelas que seguem o conceito tradicional da computação em nuvem, isso é, fornecer acesso aos recursos computacionais por demanda. Dessa forma, nas nuvens públicas, os recursos são provisionados através da Internet e com uma

granularidade fina, ou seja, se aloca e usa apenas o necessário. Tipicamente, os recursos são externos a infraestrutura do usuário e a tarifação é feita em função da utilização dos recursos. É importante salientar que o termo "público" não significa acesso gratuito aos recursos da nuvem, mas sim que qualquer público pode usá-la mediante o respeito às suas regras de utilização e tarifação. Normalmente, as nuvens públicas oferecem os três modelos de serviços (SaaS, PaaS e IaaS). Os recursos computacionais, como processadores, memória e armazenamento, são localizados dentro de um ou mais *data centers*, pertencentes a uma única organização que fornece o serviço de computação em nuvem. Os provedores de computação em nuvem, como o Google, Windows Azure, seguem esse modelo de implantação.

As nuvens privativas possuem uma infraestrutura que é operacionalizada e utilizada por uma única organização. A gestão e a localização dos recursos computacionais podem ser tanto da própria organização como terceirizado. Na realidade, esse tipo de nuvem é bastante polêmico, já que não se beneficia do modelo de negócio clássico onde o usuário da nuvem paga por utilização. Aqui a organização arca com o custo total da plataforma, incluindo a infraestrutura física de energia, refrigeração, etc, e de sua manutenção e gerenciamento, ou se, for o caso, o custo de terceirização de todos esses elementos.

Já as nuvens comunitárias possuem uma infraestrutura física, isso é, seus recursos, compartilhadas por organismos que possuem um mesmo objetivo comum e que, normalmente, estão sujeitas a um mesmo tipo de restrição legal ou de segurança. Um exemplo desse tipo de nuvem são as iniciativas governamentais onde os recursos computacionais pertencentes a órgãos como ministérios, ou autarquias, podem ser compartilhados para prover serviços de governança e cidadania. Nesse caso, os recursos podem pertencer a esses organismos, ou serem locados de terceiros, usando provedores de serviços de hospedagem ou de computação em nuvem (PaaS ou IaaS). Dessa forma, a localização física dos recursos pode ser interna ou externa as organizações e os custos podem ser compartilhados entre os membros da comunidade. O modelo de serviço mais comumente encontrado nas nuvens comunitárias é o SaaS, na forma de portais web.

Por fim, as nuvens híbridas, que são as nuvens compostas por duas ou mais nuvens de outros tipos. As nuvens híbridas nada mais são que múltiplas nuvens conectadas entre si.

1.4. Arquitetura de computação em nuvem

A tecnologia de base da computação em nuvem é a virtualização. Assim, o componente fundamental é a noção de máquina virtual que permite a definição de servidores para executar aplicações em geral. Os servidores virtuais são criados com base em uma capacidade de processamento (número de processadores), memória, armazenamento, interfaces de rede e o software que executam. O software a ser executado forma a imagem da máquina virtual e possui todos os aplicativos e programas necessários, incluindo o próprio sistema operacional. Essa imagem é denominada de *virtual appliance*. Uma *virtual appliance* nada mais é que a imagem de um sistema operacional e todo software instalado para ser executado em uma máquina virtual em um determinado hipervisor. Por exemplo, é possível ter uma imagem do Windows 7, com vários aplicativos, para executar em uma máquina virtual definida no VirtualBox.

No caso da computação em nuvem, o termo *virtual appliance* assume uma defini-

ção um pouco mais ampla. Uma *virtual appliance* é o objeto de implantação (*deployment*) na nuvem que é composto por uma imagem de uma máquina virtual e todo o software necessário para executar uma determinada tarefa. Um exemplo dessa generalização seria uma *virtual appliance* para fazer com que uma máquina virtual aja como um servidor web ou como um servidor de banco de dados. O ponto a ressaltar é que uma *appliance* já possui todas as configurações necessárias para executar uma determinada tarefa, sem que seja necessário instalar, configurar e mesmo dar manutenção e atualizações a esse serviço. É comum que o sistema operacional da *appliance* seja altamente customizado para realizar apenas o serviço a que ela se destina, com isso, muitas funcionalidades podem ser suprimidas em relação a um sistema operacional de propósito geral. Assim, pode-se simplificar dizendo que a *virtual appliance* é o software que executa em servidor virtual. Observe que, dependendo do que será executado, uma *virtual appliance* pode atender os três modelos de serviços da computação em nuvem: ao fornecer apenas uma máquina virtual com seu sistema operacional básico, estamos diante de *IaaS*; ao fornecer uma plataforma de desenvolvimento, temos um *PaaS* e; por fim, se prover um serviço final é o modelo SaaS.

As máquinas virtuais, e suas *virtual applicances*, executam sobre máquinas reais usando um software de virtualização denominado de hipervisor. É necessário criar (instanciar), configurar e gerenciar essas máquinas virtuais. Esse gerenciamento inclui tarefas como migração de máquina virtuais, balanceamento de carga, alocação de recursos reais para atender critérios de SLAs (*Service Level Agreements*) e QoS (*Quality of Service*), entre outras. Há ainda a possibilidade de alocar capacidades de armazenamento (*storage*) a máquina virtual e de configurá-la em rede. A figura 1.3 ilustra a arquitetura em camadas típica de um sistema de computação em nuvem.

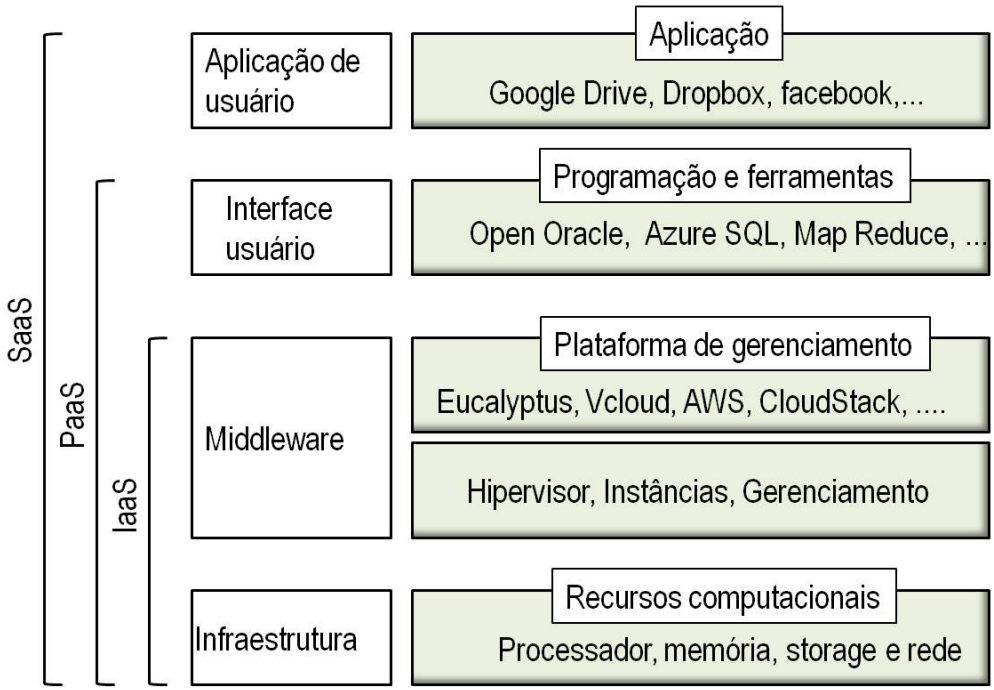

Figura 1.3: Arquitetura em camadas da computação em nuvem.

Observando a figura 1.3, é possível notar que a primeira camada é a infraestrutura

física, composta por todos os recursos físicos que compõem a nuvem. Esses recursos são classificados como processadores, rede e armazenamento (*storage*) localizados, tipicamente, em *data centers*, podendo estar localmente ou geograficamente dispersos. Da mesma forma, esses recursos podem ser organizados em clusters ou mesmo em grades computacionais. Sobre a camada de infraestrutura há a primeira camada de software, genericamente chamada de *middleware IaaS*. É essa camada que prove as capacidades básicas de virtualização e de gerenciamento dos recursos da nuvem. É possível subdividir essa camada em duas subcamadas funcionais, onde a primeira oferece um determinado hipervisor com imagens de máquinas virtuais e suas funcionalidades de configuração, implantação (instanciação) e gerenciamento, incluindo aqui, as capacidades de migração das máquinas virtuais, de *backups*, de balanceamento de carga etc. São exemplos dessa subcamada funcional o Xen, VM Sphere, Hyper-V, Citrix Xen, Oracle, entre outros.

A segunda subcamada funcional é o *middleware* de computação de nuvem, responsável por, entre outros, permitir a solicitação dos recursos virtuais, configurá-los, e autenticar usuários. Essa camada, tipicamente, oferece aos clientes da computação em nuvem uma API através de linha de comando (CLI – *Comand Line Interface*), ou de primitivas de chamadas de função usando REST[2] ou Web Services,ou ainda, via navegadores web para se ter acesso aos serviços da nuvens. Os middlewares de nuvem Apache Cloud Stack, OpenStack e Eucalyptus, que serão apresentados na seção 1.5, se enquadram aqui.

A camada seguinte é formada pela plataforma de computação em nuvem. Uma plataforma é composta pelo software usado para criar serviços de mais alto nível, variando bastante de uma para outra em função de seu objetivo. Uma plataforma oferece um "pacote" com todo o software e hardware necessários para desenvolver e implantar aplicações web, ou outros serviços criados por desenvolvedores, dentro de um contexto e da gama de possibilidades que a plataforma oferece. Uma plataforma é entregue como uma *virtual appliance* pronta a ser usada pelo cliente da computação em nuvem. O acesso à plataforma é disponibilizado por uma API que inclui interfaces web, primitivas REST, e linha de comando (ssh). Fazem parte, entre outros, o suporte a linguagens de programação, seus compiladores e bibliotecas, como Java, Rails, Python, .NET; bancos de dados, Postgres, NoSQL, MySQL, MongoDB, etc; e *frameworks* em geral como o RabbitMQ e Hadoop. São exemplos dessas plataformas: Windows Azure Platform, Google AppEngine, OpenShift, AppScale, entre outras.

Por fim, A camada de aplicação do usuário que, tipicamente, através de portais web oferece a execução de um serviço ao cliente da computação em nuvem. Há vários tipos de serviços disponíveis variando desde simulações científicas em diversas áreas passando por sistemas de produtividade, como o Google Drive e o Dropbox, até as redes sociais (twitter, facebook, picasa, etc).

1.5. Estudos de caso

A computação em nuvem se popularizou e há um grande número de provedores oferecendo os diferentes modelos de serviços: *SaaS*, *PaaS* e *IaaS*. O modelo *SaaS*, por exemplo, está bem mais próximo de nós do que imaginamos através das ferramentas de

[2]*Representation State Transfer* é um protocolo para leitura, atualização e remoção de recursos na web. REST utiliza as funções CRUD (*Create*, *Read*, *Update*, *Delete*) disponíveis através de métodos HTTP, identificando os objetos através de URLs. Por vezes é mencionado como arquitetura orientada a recursos (ROA, *resource oriented architecture*)

trabalho colaborativo. Já os modelos *PaaS* e *IaaS* são mais comuns em ambientes coorporativos. Esta seção apresenta alguns provedores de computação em nuvem e suas ofertas de serviços privilegiando uma retrospectiva histórica através da Amazon; uma análise de duas importantes e conhecidas empresas de computação, a Google e a Microsoft; e mostrando as principais plataformas de código aberto para a computação em nuvem.

1.5.1. Computação em nuvem na Amazon

No final da década de 90, início dos anos 2000, a Amazon se notabilizou como empresa de comércio eletrônico oferecendo uma variada gama de mercadorias como livros, CDs e DVDs, artigos de beleza, eletrodomésticos etc. Para realizar esse serviço de forma satisfatória, a Amazon investiu pesadamente em um parque computacional e, posteriormente, em decorrência de períodos de ociosidade, se deu conta que poderia aproveitar essa infraestrutura para oferecer serviços de computação para empresas e usuários domésticos. Nascia assim, a computação em nuvem na Amazon.

A entrada da Amazon na computação em nuvem se deu através do *Amazon Web Services*(AWS) [Amazon, 2015] em um modelo IaaS e, na sequência, adotou, também, o modelo PaaS. O AWS é dividido em uma série de serviços onde se destacam o Elastic *Cloud Computing* (EC2) e o *Simple Storage Service* (S3). O modelo de implantação adotado é o de nuvem pública.

O EC2 é um serviço web, baseado em mensagens SOAP, através do qual um usuário pode criar instâncias de máquinas virtuais usando imagens (*appliances*) de diferentes sistemas operacionais, como várias distribuições Linux; Microsoft Windows Server 2003, 2008 e 2012; OpenSolaris e FreeBSD; ou, criando, do zero, sua própria configuração de sistema. As imagens predefinidas são denominadas de *Amazon Machine Image* (AMI). Essas imagens são caracterizadas em função da potência do processador, memória e espaço de armazenamento disponíveis. O EC2 emprega o Xen como ferramenta de virtualização e o custo de utilização é tarifado por hora por recursos alocados, sejam eles efetivamente usados ou não.

O S3 é um serviço de armazenamento de dados persistente concebido para guardar grandes objetos de dados. As AMIs são mantidas nesse tipo de espaço de armazenamento. Os objetos são armazenados em uma abstração denominada de *bucket* com uma capacidade que varia de 1 byte a 5 Terabytes. Cada *bucket* tem um identificador único associado a um usuário. O espaço de nomes usado para identificar os *buckets* são globais e protegidos por mecanismos de autenticação. Os *buckets* podem ser públicos e também é possível definir direitos de acesso a outros usuários. Há três primitivas básicas para a manipulação de objetos no S3, PUT, GET e DELETE, não sendo possível que objetos sejam copiados, renomeados e movidos entre *buckets*. Além do S3, há o *Amazon Elastic Block Store* (EBS) que também é um serviço de armazenamento persistente de dados, mas de alto desempenho. O seu custo, se comparado com o do S3, é superior. O EBS aparece para o usuário com uma partição (volume) de um disco rígido, não formatado, e confiável.

Além do EC2, do S3 e do EBS, os serviços do AWS oferecem a possibilidade de se criar túneis IPsec entre o site cliente e a nuvem Amazon através do *Virtual Private Cloud* (VPC); o *Route* 53, que oferece a configuração de DNS; e o serviço *AutoScaling* que permite automatizar o processo de elasticidade. Todos esses serviços são considerados como ofertas *IaaS*.

No modelo PaaS, a Amazon oferece suporte a banco de dados, uma infraestrutura para gerenciamento da nuvem, um *framework* para troca de mensagens, e um serviço de

cache. O *Simple* DB é um repositório de dados persistente do tipo NoSQL que permite aos usuários escrever e consultar dados através de requisições de serviços web. Há ainda o *Relational Database Service* (RDS) que segue um tipo SQL. O *Elastic Beanstalk* possibilita a interação entre os serviços AWS com vistas a automatização dos processos de instanciação de novas máquinas, balanceamento de carga e monitoramento de recursos. Esse serviço é disponibilizado aos usuários através de APIs na plataforma Java, PHP e .Net. O *Simple Queue Message Servic*e (SQS) é uma oferta de *Message Oriented Middleware* (MOM) que permite a comunicação entre o AWS e outros componentes da nuvem de forma assíncrona. Por fim, o *CloudFront* que é serviço de cache distribuído que tem por objetivo, através da replicação de dados, reduzir a latência do tempo de acesso.

1.5.2. A Google e a computação em nuvem

A computação em nuvem na Google é mais conhecida pelos serviços oferecidos através do modelo SaaS, onde se destacam o *Gmail*, *Google Drive*, *Google Calendar*, *Picasa*, *YouTube*, *Google Earth*, *GoogleMaps* e o *Google Groups*, que dispensam maiores apresentações. Esses serviços são gratuitos para usuários individuais e disponíveis às organizações mediante o pagamento de uma taxa. Entretanto, a Google também tem uma atuação importante no modelo *PaaS* através do *App Engine*. Como a Amazon, o modelo de implantação adotado é o da nuvem pública.

O *Google App Engine* (GAE) [Google, 2015] é uma plataforma de desenvolvimento que disponibiliza uma grande capacidade de armazenamento e processamento. A infraestrutura da Google não diferencia as aplicações do grande público daquelas desenvolvidas por organizações: todas usam a mesma estrutura lógica e física, o que confere uma série de facilidades de gerenciamento, de disponibilidade e de recursos. Por outro lado, dada essa unificação, é necessário que se tenha um relativo controle do que é executado para evitar que o mau funcionamento de um componente comprometa, de alguma forma, o restante. Por isso, há uma série de restrições que são impostas às aplicações *AppEngine*. Ao mesmo tempo, reconhecendo as necessidades diferentes entre os usuários grande público e organizações, a Google tem investido em melhorias em uma plataforma denominada de *App Engine for Business*.

O ambiente de desenvolvimento disponível é chamado de *Google App Engine* SDK e permite o desenvolvimento de aplicações em várias linguagens, onde se destaca o uso de Python e Java, e a posterior implantação dessa aplicação na infraestrutura do Google. Em relação ao armazenamento persistente de dados, a *App Engine* oferece o *DataStore*, que é uma base NoSQL que recebe consultas usando a *Google Query Language* (GQL). Para dar uma noção da capacidade de armazenamento do *DataStore*, é sobre essa base de dados que estão disponíveis as informações do *Google Earth* e do *YouTube*.

1.5.3. Microsoft e a computação em Nuvem

A Microsoft oferece os modelos *SaaS*, *PaaS* e *IaaS* na forma de uma nuvem pública. Os modelos *SaaS* e *PaaS* são ofertados, respectivamente, pelo *Online Services* e pelo *Azure* [Azure 2015] e o modelo IaaS pela disponibilização de imagens de máquinas virtuais. O Azure é baseado no hipervisor Hyper V onde executam máquinas virtuais com versões do Microsoft Windows Server, do *Internet Information Service* (IIS) e .NET.

O modelo *SaaS* do Azure é disponível através Windows Live que inclui aplicações como Windows Mail, Windows Photo Gallery, Windows Movie Maker, entre outros. No

modelo *PaaS*, o desenvolvimento de aplicações do Azure é através do Visual Studio, bem difundido entre aqueles que adotam tecnologia Microsoft, tornando o ambiente bastante familiar. No entanto, nem sempre é possível executar no Azure código já existente sem que sejam feitas algumas adaptações. Essas adaptações vêm, em parte, devido a diferenças de bibliotecas (DLLs). Os principais serviços Azure são:

- *Azure storage*: um sistema de armazenamento não relacional, organizado em blobs (arquivos binários), tabelas e mensagens;
- *SQL Azure*: sistema de armazenamento relacional baseado no Microsoft SQL Server;
- *AppFabric Service Bus* e *AppFabric Acess Control*: empregado para a hospedagem de aplicações desenvolvidas no Azure. Atua como uma espécie de repositório de aplicações que são disponibilizadas aos usuários via URIs;
- *Azure Connect*: uma solução baseada em IPsec para estabelecer comunicação segura entre o site cliente e a nuvem Azure.
- Azure CDN (*Content Delivery Network*): um sistema de cachê distribuído geograficamente

Além desses, como um modelo *IaaS*, o Azure oferece uma série de imagens de máquinas virtuais sobre as quais é possível configurar e executar pacotes de aplicativos. Há, a partir do que é denominado de VM *Role*, a possibilidade de obter uma máquina virtual e customizá-la completamente, executando, inclusive, sistemas operacionais que não pertencem a família Microsoft.

1.5.4. Plataformas Código Aberto para computação em nuvem

As soluções apresentadas anteriormente, Amazon, Google e Azure, constituem uma solução de nuvens públicas. No entanto, por uma série de motivos, entre eles custo e segurança de dados, as organizações podem optar por manter sua própria nuvem seguindo um modelo de nuvem privada.

Uma nuvem privada possui, basicamente, a mesma estrutura física das nuvens públicas, ou seja, servidores físicos, rede de interconexão, hipervisores executando em máquinas físicas para suportar máquinas virtuais, *virtual appliances*, sistemas de armazenamento e um *front-end* para comunicação com o usuário. Todos esses componentes precisam ser instalados, configurados e a nuvem, em si, deve ser gerenciada. Isso é feito através de uma plataforma de computação em nuvem. As plataformas de código aberto tais como Eucalyptus, OpenStack e CloudStack, entre várias outras, podem ser empregadas para essa finalidade. Todos esses sistemas oferecem suporte ao modelo *IaaS*.

O Eucalyptus (*Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems*) [Eucalyptus 2015] é uma plataforma de código aberto que permite a construção de nuvens privadas e híbridas compatíveis com o AWS (Amazon). De uma certa forma, o Eucalyptus pode ser visto como a versão, em código aberto, do EC2. Cada máquina física é um nó Eucalyptus que pode executar máquinas virtuais sobre diferentes tipos de hipervisores, entre eles, Xen, VMware e KVM. Cada nó possui um controlador (*node controller*) que envia informações sobre a atividade desse nó a um servidor especial, denominado de *cluster controller*. Uma infraestrutura Eucalyptus pode conter um, ou

mais, *clusters controllers* que são gerenciados pelo *cloud controller*. É através do *cloud controller* que os usuários, desenvolvedores e administradores têm acesso aos recursos da nuvem Eucalyptus. Esse acesso é feito através de ferramentas de linhas de comando, compatíveis com o EC2, e de uma interface web. Em relação ao armazenamento de dados, o Eucalyptus oferece o *storage controller* e o *Walrus*. O *storage controller* é empregado para definir discos virtuais para as aplicações e possui uma correspondência com o EBS da Amazon. Por sua vez, o *Walrus* é similar ao S3, ou seja, é um sistema de armazenamento distribuído baseado na noção de *buckets*.

O *OpenStack* [Openstack, 2015] nasceu de uma iniciativa da NASA em colaboração com a empresa Rackspace com o objetivo de desenvolver um sistema operacional para nuvem a ser usado em parques de servidores compostos por máquinas existentes no mercado. A atual versão do sistema oferece uma série de facilidades para gerenciamento de máquinas virtuais (executar, reinicializar, suspender e terminar instâncias); para o controle de acesso aos recursos baseado em papéis (*roles*); para alocar, rastrear e limitar a utilização dos recursos. A administração de uma nuvem OpenStack é feita através de uma aplicação web chamada *Dashboard*, por linha de comando, ou via uma API REST.

O *OpenStack* tem uma arquitetura modular composta por vários elementos. O Nova é responsável pelo provimento de máquinas virtuais e oferece suporte aos principais hipervisores do mercado (Xen, VMware, KVM, Hyper-V). A capacidade de armazenamento é oferecida através do Swift, que é um sistema de arquivos distribuídos redundante, e do Cinder que permite a criação de discos virtuais para as máquinas virtuais instanciadas via OpenStack. A capacidade de interconexão em rede, incluindo a definição de uma infraestrutura de redes própria, a definição de VLANs, o uso de SDN (*Software Defined Networks*), VPNs (*Virtual Private Networks*), assim como a integração com outros serviços de redes, como IDS (*Intrusion Detection Systems*) é feita através do módulo denominado de Neutron. Além desses existem um serviço de autenticação e autorização (Keystone); um serviço de gerenciamento de *virtual appliances* (Glance); bases de dados relacionais e não relacionais (Trove); e serviços para monitoramento e contabilidade (Ceilometer) e para compor aplicações de nuvem a partir de outras já existentes (Heat).

Para concluir, o *CloudStack* [Apache, 2015] que é um plataforma, em código aberto, para criar, gerenciar e implantar infraestrutura de computação em nuvem, tanto públicas, como privadas e híbridas. O *CloudStack* iniciou como um projeto dentro de uma empresa, a VMOPs, que obteve direito de usar o nome *Cloud.com*. Em julho de 2011, a empresa Citrix comprou o *CloudStack* e, posteriomente, doou para a *Apache Software Foundation*, que o considerou como uma de suas prioridades de desenvolvimento. Esse histórico e o envolvimento com a Apache levou o CloudStack a ser compatível com uma grande quantidade de hipervisores, a oferecer uma série de APIs compatíveis com outras soluções em nuvens e a adoção de vários padrões abertos para computação em nuvem. Atualmente, o CloudStack oferece suporte para os provedores comerciais da Citrix, da VMware, Oracle, Xen e KVM. Sua estrutura é baseada em três componentes fundamentais: o *Compute Controller*, que controla as máquinas virtuais; o *Network Controller*, que controla as interconexões em rede; e o *Storage Controller* responsável pelo gerenciamento de um sistema de armazenamento orientado a bloco (disco virtual). Esses componentes constituem a camada inferior de software e tem acesso aos recursos físicos de hardware. O gerenciamento da nuvem em si é feito com o *CloudStack Orchestration Engine* que recebe comandos através de uma API baseada em linha de comandos ou a partir de uma interface web.

1.6. Computação em nuvem: riscos e benefícios

Os riscos e benefícios da computação em nuvem podem ser analisados sob o ponto de vista de organizações (empresas) e dos usuários que a adotam. Inicialmente, para uma organização que é cliente de um serviço em nuvem, o principal benefício é a redução de custos. Essa redução advém de vários fatores como a eliminação da necessidade de investimento em infraestrutura própria com a compra de equipamentos, refrigeração, cabeamento, espaço físico, consumo de energia elétrica, etc. Além disso, há outros fatores correlacionados como a desatualização das máquinas e sua provável renovação, sua depreciação e a necessidade de uma equipe de TI para dar suporte para a manutenção e configuração dessas máquinas.

O uso da computação em nuvem também traz benefícios em relação à segurança física dos dados. Os fornecedores de soluções em nuvem empregam vários sistemas de *backup* e de replicação de forma a manter a integridade física dos dados que estão armazenados na nuvem e a sua disponibilidade. Isso é particularmente interessante para organizações menores, e mesmo para usuários particulares, que tem dificuldades ou falta de disciplina para manter *backups*. A replicação dos dados possibilita que, mesmo na presença de falhas, se tenha acesso aos dados, ou seja, isso agrega disponibilidade a esses dados. Manter os dados na nuvem evita ainda que equipamentos móveis, como notebooks e *tablets*, em caso de perda ou furto, tenham dados sigilosos expostos de alguma maneira. Por outro lado, principalmente para dados sigilosos, há sempre a preocupação de saber se há ou não risco de vazamento das informações e uma certa insegurança psicológica de não saber onde os dados estão armazenados. Esta seção discute alguns dos principais pontos a serem considerados ao se adotar uma solução de computação em nuvem.

1.6.1. Aspectos financeiros: TCO e ROI

A solução em nuvem não é gratuita: é necessário pagar, por demanda, pela utilização dos recursos. Então, principalmente em nível empresarial, se utiliza cálculos de ROI (*Return Of Investiment*) e TCO (*Total Cost of Ownership*) para fornecer aos gestores uma base comparativa para saber se é mais em conta alugar uma infraestrutura que possuí-la. Vários provedores de computação em nuvem fornecem calculadoras de TCO, como o Windows Azure[3].

A análise de TCO é feita levando em conta apenas aspectos financeiros. É claro que, eventualmente, por mais barato que seja uma solução em nuvem, por questões de segurança da informação, pode ser que manter sua própria infraestrutura seja o mais apropriado. Entretanto, justamente como segurança da informLaTeX Error: File 'comment.sty' not found.ação é algo estratégico, os provedores de computação em nuvem seguem normas estritas de segurança e devem ser certificados em vários padrões como, por exemplo, ISO 27001 e SAS 70 Type II.

Além dos aspectos de segurança da informação em si, há questões jurídicas e legais. Por exemplo, o sistema bancário tem uma regulamentação extremamente exigente e particular ou, ainda, as leis que regem acesso a dados pessoais são diferentes entre países. Assim, para provedores como a Google, que tem *datacenters* no mundo inteiro, qual legislação deve ser respeitada? A do local do cliente ou do local de onde os dados estão

[3] `http://www.microsoft.com/brasil/windowsazure/tco`

armazenados (e que podem sofrer migração)? Enfim, a tendência é elaborar contratos fornecendo garantias necessárias ao cliente sem citar leis específicas de países.

O modelo *pay-as-you-go*, também introduz dificuldades para o mundo empresarial. Esse modelo, em parte, é derivado do comércio eletrônico, onde um cliente paga por um bem de consumo com seu cartão de crédito à medida que adquire bens e serviços. No entanto, no mundo empresarial, não há a mesma flexibilidade de um cartão de crédito pessoal. Mesmo com a existência de um cartão de crédito coorporativo, há uma questão de previsão e controle de custo. É difícil fazer uma programação financeira sem saber quanto se pagará (*pay-as-you-go*) por serviços no decorrer de um mês. Por isso, os fornecedores de computação em nuvem costumam oferecer serviços através de assinaturas e com diferentes planos de consumo.

Sob o ponto de vista de pessoas físicas que utilizam o serviço de computação em nuvem, há diferentes benefícios dependendo do tipo de modelo de serviço explorado (*SaaS*, *PaaS* ou *IaaS*), e do perfil de usuário dessas pessoas. Se considerarmos a questão de desenvolvimento de produtos (software) uma questão interessante é o *time to market*. O ciclo de desenvolvimento de um software passa por etapas como análise, especificação, projeto arquitetural, codificação, testes e manutenção. O uso de soluções do tipo *PaaS* e *IaaS* eliminam a preocupação com o ambiente de desenvolvimento. Não é mais necessário "perder tempo", instalando, configurando e dando manutenção para equipamentos para, só depois, os desenvolvedores poderem iniciar a usar esses recursos adequadamente. Ainda, é possível, com o auxílio de *virtual appliances*, desenvolver, testar e manter versões de softwares para diferentes sistemas operacionais, mesmo legados, sem ter a necessidade de possuir esses sistemas "em casa". Por sua vez, o modelo *SaaS* permite, com o uso de API simples e bem definidas, a integração de diferentes serviços criando *mashups*[4].Dessa forma, o desenvolvimento de novas soluções usando componentes já existentes é facilitado.

1.6.2. Confidencialidade de dados

Outro ponto interessante é a ubiquidade dos dados e das aplicações. Se todas as informações necessárias para desenvolver, ou usar, uma aplicação estão na nuvem, o usuário pode acessá-la de qualquer lugar e de qualquer computador confiável. Isso fornece mobilidade e facilita a migração de um usuário entre máquinas.

Novamente, o maior risco percebido pelos usuários diz respeito à confidencialidade e a segurança de seus dados. No caso de dados de uma organização, a segurança é dada por cláusulas contratuais e pela escolha de um provedor devidamente certificado. Já os dados pessoais como, por exemplo, os e-mails no Gmail, os dados no Facebook, e os arquivos no Dropbox, há sempre a insegurança psicológica de quem pode ter acesso a essas informações. A notar que mesmo nesses serviços há um contrato, que a maioria das pessoas não lêem, e que contém, por vezes, algumas cláusulas abusivas.

[4]Em desenvolvimento web, *mashup* é uma página, ou aplicação web, resultante da utilizado do conteúdo, ou informação, de mais de uma fonte para oferecer um novo serviço. Por exemplo, um site de viagens pode oferecer possibilidades de hotel, de voos e aluguel de carro para determinado período e destino consultando serviços web de reservas de hotéis, das companhias aéreas e de locação de veículos.

1.6.3. Segurança em computação em nuvem

A questão da segurança em computação em nuvem ultrapassa a preocupação com o acesso aos dados. Existem outros pontos que devem ser considerados como autenticação, confidencialidade, integridade, rastreabilidade e disponibilidade.

A administração de serviços *SaaS*, *PaaS* e *IaaS* são feitas, na maior parte das vezes, empregado um nome de usuário e uma senha. A maioria dos acessos aos serviços administrativos é feita usando *ssh*, que oferece um bom nível de segurança, mas cabe lembrar que em caso de exposição de senhas administrativas é possível remover contas de usuários (*SaaS*), remover aplicativos (*PaaS*) e destruir ou modificar máquinas virtuais (*IaaS*).

Por confidencialidade entende-se a garantia de que os dados só serão lidos por aqueles que têm o direito para tal. Essa garantia deve ser dada tanto para os dados armazenados em meio persistentes, isso é, evitando que terceiros leiam diretamente esses dados do meio de armazenamento quanto na transferência dos dados via rede. Para mitigar o risco da confidencialidade, os dados são armazenados usando técnicas de cifragem. No caso da transferência via rede, várias técnicas são empregadas como o VPN, SSL e IPsec. Ainda relacionado com os dados, há a integridade, isso é, a garantia de que os dados não foram modificados de forma intencional ou acidental. Novamente, como no caso da transferência dos dados, são usados protocolos seguros, como o SSL.

A rastreabilidade diz respeito à capacidade de gerar traços (*logs*) que permitam analisar o comportamento e a utilização das aplicações. Esses traços são importantes para, em caso de problemas, realizar auditorias e para monitorar comportamentos suspeitos durante seu uso. Ainda, esses traços podem ser empregados para otimizar o acesso aos recursos da nuvem.

Por fim, a disponibilidade, ou seja, a garantia que se tenha acesso ao serviço sempre que for necessário. A forma mais comum de oferecer disponibilidade é através do estabelecimento de SLAs (*Service Level Agreements*). Um SLA é um contrato de oferecimento de serviço, estabelecido entre um cliente e um provedor, que define um conjunto de métricas a serem respeitadas. As métricas mais comuns são a garantia de acesso ao serviço quando necessário (*uptime*), normalmente fornecido em um percentual, como por exemplo, 99.9% do tempo existente no período de um mês do calendário; tempo de resposta e latência de acesso; e, confiabilidade de componentes (*Medium Time Between Faults* – MTBF). Além dessas, é comum encontrarmos em um SLA as responsabilidades imputadas ao cliente e ao provedor pelo não cumprimento do contrato de SLA e as garantias oferecidas.

1.6.4. Licenciamento de software

O modelo de licenciamento de software é baseado em um acordo denominado de EULA (*End User License Agreement*) que impõe condições para a distribuição e para o uso de um determinado software. No caso da aquisição de um software proprietário, o custo da licença depende do EULA estabelecido que define, entre outros aspectos, se o software pode ser instalado em uma única máquina, ou em múltiplas (quantas?), e se é possível haver usuários remotos e qual é a quantidade de acessos simultâneos permitidos. Esse modelo convencional funcionou muito bem até a disseminação da Internet e da virtualização. Tanto a Internet, quanto a virtualização, trouxeram alguns problemas na forma de contabilizar os acessos e usuários. Por exemplo, ao licenciar o software para

uma máquina específica, mesmo que o software seja sempre usado por apenas uma máquina virtual, se inviabiliza o controle de máquina específica, pois se perde esse controle de onde essa máquina virtual executará.

Com a computação em nuvem, o problema de licenciamento toma uma nova dimensão: não há ainda uma forma aceita de como precificar o software usado na computação em nuvem. Atualmente há modelos de licenciamento, mas todos eles com suas desvantagens. Na prática, os provedores de computação em nuvem tendem a licenciar suas aplicações, ou serviços, baseados em contas de usuários e contas de máquinas. Alguns serviços em nuvem implementam um modelo de assinatura (mensal ou anual) para permitir o acesso aos serviços mediante o pagamento dessa assinatura. No entanto, não há ainda uma forma estabelecida e aceita. Depende do tipo de serviço em si e do modelo de serviço oferecido (SaaS, PaaS, IaaS). De certa forma, o consenso que existe é que o emprego de software livre é um modelo de sucesso para computação em nuvem. Como nesse caso a EULA permite a livre cópia, distribuição e uso de software, não há o que se preocupar. O importante a ressaltar é que se deve prestar atenção a questão de uso de software licenciado ao utilizar e prover serviços em computação em nuvem analisando as eventuais restrições caso a caso.

1.6.5. Padrões abertos

Uma das preocupações da computação em nuvem é oferecer seus serviços sem que haja uma dependência tecnológica ou de fabricante. Isso é possível através da adoção de padrões abertos aceitos pela comunidade em geral. Considerando isso, há muitas entidades preocupadas com a padronização da computação em nuvem abordando tópicos que vão desde conceitos e terminologias a aspectos técnicos como APIs e formatos de arquivos. Uma boa referência é a página wiki da *cloud-standard.org*[5].Entre os vários padrões existentes, este trabalho destaca três: OVF (*Open Virtualization Format*) , OCCI (*Open Cloud Computing Interface*) e CDMI (*Cloud Data Management Interface*).

O OVF é uma especificação para empacotar e distribuir virtual appliances de uma forma eficiente, extensível e independente de fabricante (*hypervisor neutral*). Um dos objetivos do OVF é facilitar o gerenciamento, a integração e automatização para o uso de *virtual appliances*. A ideia fundamental é que seja possível um hipervisor exportar sua máquina virtual para outro hipervisor. Um pacote OVF é um conjunto de vários arquivos sob um mesmo diretório. Entre esses arquivos está um arquivo XML, com a extensão .ovf, que descreve vários atributos (metadados) do pacote como seu nome, requisitos do hardware onde a *appliance* será executada e referências a outros arquivos do pacote OVF. Há ainda um arquivo de manifesto[6] (.mf); um, ou mais, arquivos com as imagens de discos virtuais, que mantém o estado de uma máquina virtual (.vmdk e ou .vhd); e certificados digitais para garantir a integridade e confiabilidade dos arquivos. A forma de distribuir esses arquivos é através de um único arquivo com a extensão .ova (*Open Virtual Appliance*) que nada mais é do que um tar do diretório que contém os arquivos OVF.

O OCCI é um conjunto de especificações que foram iniciadas no Open Grid Forum e que evoluiram para um protocolo e uma API para as mais diversas tarefas de gerenciamento. Originalmente, em consequência da sua origem, o OCCI definia mecanismos que

[5]http://cloud-standards.org

[6]O termo manifesto, em empacotamento de software, nada mais é que nome dado a um arquivo que contém a listagem do conteúdo do pacote de software; um hash MD5 (opcional) para a conferência da integridade de seu conteúdo; e informações gerais para os usuários.

permitiam a criação e implantação de máquinas virtuais principalmente dentro do modelo *IaaS*. Essas interfaces permitiam, entre outras, tarefas de monitoramento e controle de escalabilidade. A atual especificação OCCI contempla APIs para os três modelos de serviços de computação em nuvem (*IaaS*, *PaaS* e *SaaS*).

Já o CDMI define interfaces para a criação, recuperação, atualização e remoção de dados na nuvem a partir de aplicações de usuários. Esse tratamento sobre dados é feito através do conceito de container que é uma abstração de onde os dados são armazenados e organizados. O CMDI oferece primitivas para manipulação de containers (criação, controle de acesso, monitoramento e contabilização).

1.7. Atividades de pesquisa em computação em nuvem

Baseado em *calls for papers*, e em artigos publicados em várias conferências que tratam do tema de computação em nuvem, é possível identificar cinco grandes linhas de desenvolvimento de trabalhos nessa área, assim classificadas:

- *Provimento de serviço*: trata basicamente da problemática de alocação de recursos virtuais na nuvem para otimizar diferentes aspectos, tais como, redução do consumo de energia, maior banda passante, menor latência, menor tarifa, etc.

  Interfaces para Clientes: desenvolvimento de *brokers* para os usuários alocarem e empregarem os recursos de forma ótima, buscando uma boa relação custo e benefício, sem esquecer a usabilidade da interface.

  Aplicações: desenvolvimento de aplicações para serem executadas em nuvem. O modelo SaaS prevalece nessa categoria. Uma convergência interessante de assuntos de pesquisa é o uso da computação em nuvem para aplicações *big data* e a sua integração com IoT (*Internet of Things*).

- *Implantação* (*Deployment*): mecanismos para instanciar de forma automática os recursos necessários a execução de uma aplicação em nuvem. É nesse item que se enquadra os aspectos de computação autonômica.

Cabe ressaltar que há muita atividade de pesquisa relacionada com tarefas de gerenciamento, monitoramento, contabilização de uso de recursos e faturamento. No entanto, é possível ver essas tarefas como partes integrantes das linhas de provimento de serviço e implantação.

1.8. Conclusão

A computação em nuvem utiliza tecnologias, serviços e aplicações similares aos já existentes na Internet. Na prática, o que distingue a computação em nuvem de aplicações na Internet, e da computação em grade, é basicamente um modelo de negócio. A computação em nuvem disponibiliza o acesso a recursos computacionais de forma similar ao acesso aos serviços públicos de telefonia e de distribuição de energia elétrica, ou seja, é possível utilizar recursos de processamento, de rede e de armazenamento por demanda, conforme a necessidade, e se paga apenas pelo o que for consumido (modelo *pay-as-you-go*). Dentro da noção "por demanda" está embutido o conceito de elasticidade, também

importante, na computação em nuvem, que nada mais é que a capacidade de alocar e liberar recursos conforme necessário.

A facilidade de acesso à Internet permitiu uma proliferação de provedores de computação em nuvem e, com eles, alguns abusos no emprego da terminologia. Por isso, houve a preocupação em se definir claramente os modelos de serviços de computação em nuvem e as formas de implantação da infraestrutura física da nuvem. Os serviços seguem três modelos: (i) *IaaS*, onde a infraestrutura é o serviço oferecido; (ii) *PaaS*, que é a oferta de um ambiente de desenvolvimento completo, a plataforma, para a criação de software e novas soluções, e; (iii) *SaaS*, onde é disponibilizado, via web, um aplicativo para um usuário final. Quanto à implantação, as nuvens são classificadas em função de quem detém os recursos físicos e quem os gerencia em quatro tipos: (i) pública, onde a infraestrutura pertence a uma organização externa; (ii) privativa, a infraestrutura pertence a própria organização e é de seu uso exclusivo; (iii) comunitária, quando organizações afins compartilham recursos físicos, e; (iv) híbrida, quando a nuvem é formado por pelo menos dois tipos diferentes de nuvens.

A computação em nuvem traz uma série de benefícios e, junto com eles, novos desafios. Os benefícios principais estão relacionados com o princípio de que "alugar e mais importante que possuir" e que o valor do negócio, em si, está na informação e no conhecimento e não no parque tecnológico que se possui. Ao terceirizar o parque informático se tem uma redução de custos de aquisição, manutenção, de infraestrutura, em equipes de TI, etc. Além disso, no momento que a infraestrutura deixa de ser um problema, os recursos financeiros e humanos podem ser focados no negócio em si e não em TI. No entanto, essa terceirização traz consigo a preocupação com a privacidade dos dados, com sua integridade, com a disponibilidade de seu acesso. Por fim, o uso de recursos por demanda contribui com a computação verde.

Com base na relação custo versus benefício é importante refletir sobre uma série de pontos antes de se adotar uma solução baseada em computação em nuvem, tais como: (i) esclarecer exatamente a necessidade (*IaaS*, *PaaS* e *SaaS*) e o benefício desejado; (ii) avaliar as ofertas disponíveis dos fornecedores de nuvem em função dessas necessidades; (iii) fazer uma análise de risco de eventuais dados sigilosos; (iv) analisar a perenidade e idoneidade do provedor de serviço, e; (v) analisar os problemas de integração dos serviços e softwares legados com o ambiente em nuvem. Cabe ressaltar que as aplicações estratégicas podem usufruir da computação em nuvem. No entanto, é importante que se tenha uma série de cuidados adicionais com essas aplicações estratégicas por representarem o *savoir faire* da empresa e apresentarem dados confidenciais.

Em relação às atividades de pesquisa e desenvolvimento na área de computação em nuvem pode-se afirmar que há um vasto campo de oportunidades. A computação autonômica, onde os recursos são alocados, gerenciados e liberados, com a mínima intervenção humana agrega uma série de desafios: escalonamento, otimização de recursos, alocação, computação verde, padronização de interfaces, criação de protocolos de gerenciamento, entre tantos outros. Há questões de contabilidade de uso de recursos e do faturamento de serviços em um ambiente largamente distribuído. Tem, ainda, o desenvolvimento de aplicações voltadas a computação em nuvem, tais como as de grande volume de dados (*big data*) e a integração da nuvem com a Internet das Coisas (IoT).

Referências

Amazon EC2. Amazon EC2. Disponível: http://aws.amazon.com/ec2. Acesso: fevereiro, 2015.

Apache. Apache Cloud Stack: Open source cloud computing.
Disponível: http://cloudstack.apache.org. Acesso: fevereiro, 2015.

Azure. Microsoft Azure. Disponível: http://www.microsoft.com/azure/. Acesso: fevereiro, 2015.

Coelho, Otávio P. Cloud Computing and Azure no Contexto da Pesquisa Científica. Apresentação realizada durante o CLCAR 2010, Gramado, RS.

Eucalyptus. Eucalyptus: Open source private cloud software.
Disponível: https://www.eucalyptus.com/eucalyptus-cloud/iaas. Acesso: fevereiro, 2015.

Foster, I.; Kesselman, C. The Grid: Blueprint for a New Computing Infrastructure, I. Foster and C. Kesselman (Eds.), Morgan-Kaufmann, 1998

Google. Google App Engine: Platform as a Service.
Disponível: https://cloud.google.com/appengine/docs. Acesso: fevereiro, 2015.

Gruman, Galen. What cloud computing really means. InfoWorld. 2008.

Marinescu, D. C.; Cloud computing: Theory and Practice. Morgan Kaufmann (Elsevier), Waltahm, USA, 2013.

Mell, P.; Grace T. The NIST definition of Cloud Computing. NIST Special Publication 800-145, September, 2011.

NIST. The NIST definition of Cloud Computing (Draft).
Disponível: http://csrc.nist.gov/groups/SNS/cloud-computing Acesso: fevereiro, 2015.

Openstack. Openstack: The open source cloud operating system. Disponível: http://www.openstack.org/software. Acesso, fevereiro, 2015.

Plouin, Guillaume, Cloud Computing: une rupture décisive pour l'informatique d'enterprise. 2ª edição. Dunod, Paris. 2011.

Smith, J.E, Nair, R. "The architecture of virtual machines". IEEE Computer, v.38, n.5, pp. 32-38, 2005.

Sosinsky, Barrie, Cloud Computing Bible, Wiley & sons, 2011.

Voorsluys, William; Broberg, James; Buyya, Rajkumar. "Introduction to Cloud Computing". In R. Buyya, J. Broberg, A.Goscinski. Cloud Computing: Principles and Paradigms. New York, USA: Wiley Press. pp. 1–44. ISBN 978-0-470-88799-8, February, 2011.