**Atvdd\_AOC\_01**

**Atividade de Pesquisa – Cloud Computing**

**Colocar Nome e R.A.**

Efetuar uma pesquisa - **Cloud Computing** ou Computação em Nuvens.

**Apresentar:**

* Surgimento;
* Características;
* Segurança;
* Arquitetura;
* Vantagens e Desvantagens.

**Apresentar as Fontes de Consulta.**

# Introdução:

Este trabalho tem como objetivo apresentar uma breve atividade de pesquisa sobre *Cloud Computing* (Computação em Nuvem) tendo em conta os tópicos: surgimento do conceito de *Cloud Computing*, características, segurança, arquitetura além de suas vantagens e desvantagens.

*Cloud Computing* é considerado um conjunto de serviços de TI (Tecnologia da Informação) fornecidos ao cliente através de uma rede juntamente com a habilidade de se reduzir os requisitos de serviço desejado. A Computação em Nuvem tem o potencial de eliminar os requisitos de infraestrutura com alto custo para serviços e soluções baseadas em TI usados pela indústria, prometendo oferecer uma arquitetura de TI flexível e acessível através da *Internet* (Rede Mundial de Computadores) e dispositivos portáteis.

A Computação em Nuvem é fornecida por acesso sob demanda, através da Internet, para utilização de recursos computacionais, de aplicações, servidores (servidores físicos e virtuais), armazenamento de dados, ferramentas de desenvolvimento, capacidade de rede com hospedagem em Data Centers remotos gerenciados por um provedor de serviços (CSP – Cloud Services Provider). Os CSPs tornam estes recursos disponíveis mediante uma taxa de assinatura mensal ou cobrança de acordo com o uso.

# Breve histórico do surgimento da Computação em Nuvem

A ideia atual de Computação em Nuvem data dos anos 1950 quando *mainframes* de larga escala eram disponibilizados para universidades e corporações. A infraestrutura colossal de *hardware* de um *mainframe* era instalada no que era chamada de Sala do Servidor (já que esta sala era capaz de comportar apenas um *mainframe*). Múltiplos usuários eram capazes de acessar os recursos do *mainframe* através dos “*dumb-terminals*” (terminais-burros) que eram estações com a única funcionalidade de facilitar o acesso ao *mainframe*.

Devido ao alto custo de se comprar e manter um *mainframe*, as organizações não eram capazes de oferecer um dispositivo para cada usuário. Se tornou prática permitir que múltiplos usuários compartilhassem acesso a mesma camada de armazenamento de dados e recursos de CPU, ou seja, poder de processamento, de qualquer estação. Com a disponibilização de um *mainframe* de acesso compartilhado, uma oprganização poderia ter um melhor retorno de investimento feito em seu equipamento.

O cientista da computação norte-americano, John McCarthy , inventor do termo “Inteligência Artificial”, defendeu a proposta de “time-sharing” ou computação por tempo compartilhado. McCarthy, disse que a computação poderia permitir que um computador fosse utilizado simultaneamente por dois ou mais usuários. Desta forma, as pessoas poderiam realizar tarefas, aproveitando o período de tempo disponível dos recursos. Na visão do cientista, o compartilhamento, além de reduzir gastos, permitiria pagamento somente pelo período utilizado.

Esse modelo foi apresentado por McCarthy durante um discurso no Massachusetts Institute of Technology (MIT), nos EUA, em 1961. Ele sugeriu a criação da “Utility Computing” ou computação como serviço de utilidade pública, no mesmo sentido do fornecimento de água, luz ou telefone, que chega às residências ou empresas, sem que as pessoas saibam de onde vem.

Em 1963, DARPA (Defense Advanced Research Projects Agency – Agência de Projetos de Pesquisa Avançada de Defesa), presenteou o MIT (Massachussets Institute of Technology) com 2 milhões de dólares para o Projeto MAC (MIT Project on Mathematics and Computation). O fundo incluía um requerimento para que o MIT desenvolvesse uma tecnologia que permitisse o uso de um computador por duas ou mais pessoas simultâneamente. O projeto definiu-se inicialmnente como uma forma primitiva de Computação em Nuvem, com apenas duas ou três possoas realizando acesso simultâneo.

O Projeto MAC inicialmente foi dirigido pelo Cientista da Computação Robert M.Fano, tendo o Cientista da Computação Fernando José Corbató como membro fundador. O termo Projeto foi usado no lugar de Laboratório para inspirar o pessoal do MIT a juntar-se à iniciativa sem desfiliar-se dos seus laboratórios à época. Uma das primeiras contribuições do projeto foi expandir e fornecer o hardware para o software CTSS (*Corbató’s 1961 Compatible Time-Sharing System*) o que permitiu com que múltiplos usuários em terminais espalhados pelos departamentos do MIT rodassem programas centralizados em uma única máquina *mainframe*.

Em 1969, C. R. Licklider (Joeph Carl Robnett Licklider), um Psicólogo e Cientista da Computação estado-unidense, ajudou a desenvolver a ARPANET (Rede de Agências para Projetos de Pesquisa Avançadas - *Advanced Research Projects Agency Network*) o embrião da rede mundial de computadores atualmente conhecida como Internet, e promoveu a ideia de “*Intergalactic Computer Network*” (Rede Intergalática de Computadores) todas as pessoas do planeta estariam iterconectadas por computadores capazes de acessar informação de qualquer lugar. J. C. R. Licklider acreditava que o CTSS facilitaria grandemente a eficiência, redução de custos, a ganharia tempo ao permitir que vários usuários compartilhassem dos mesmos recursos computacionais de um enorme mainframe em vez de empregar pequenas máquinas individuais.

Dentro de seis meses da criação do Projeto MAC, 200 usuários estavam aptos a acessar o sistema de 10 departamentos diferentes no MIT. Em 1969, *Project MAC*, *Bell Laboratories*, e *General Electric* juntamente desenvolveram o Multics (*Multiplexed Information and Computing Service*) que evoluiu de computador de tempo compartilhado para um sistema de computador online e incorporou funcionalidades como compartilhamento de arquivos, gerenciamento e sistema de segurança em seu *design*. O complexo sistema agora poderia suportar 300 conexões simultâneas em 1000 terminais do MIT e levou a Bell Labs a empregar uma forma mais simples do sistema operacional UNIX.

A palavra “virtualização” começou a ser usada para descrever esta situação, embora seu significado tenha mudado posteriormente para algo mais amplo.

O termo “*Cloud*” (Nuvem) e sua primeira referência data dos anos 1990 e origina-se no mundo das telecomunicações, onde historicamente estas companhias ofereciam apenas uma conexão dedicada de dados ponto a ponto. Estas companhias começaram a oferecer o serviço de Rede Virtual Privada (VPN – *Virtual Private Network*). Em vez de construir uma infraestrutura física para permitir que mais usuários tivessem suas próprias conexões, as empresas de telecomunicações forneceram aos usuários acesso compartilhado à mesma infraestrutura física. Essa mudança permitiu que as empresas de telecomunicações redirecionassem o tráfego de rede em tempo real conforme necessário, levando a um melhor equilíbrio da rede e maior controle sobre o uso da largura de banda por um custo mais baixo. O termo “*Cloud*” em *Cloud Computing* refere-se, portanto, a *Internet* e sua infraestrutura.

O significado de virtualização começou a ser difundido nos anos 1970, e hoje é definida como a criação de uma Máquina Virtual, que atua como um computador real com um sistema operacional totalmente funcional. O conceito de virtualização desenvolve-se juntamente com a Internet, a partir do momento que as VPNs se tornaram um negócio rentável levando ao desenvolvimento da Infraestrutura de Computação em Nuvem Moderna.

# Características fundamentais da Computação em Nuvem

Plataformas de Computação em Nuvem oferecem fácil acesso, escalabilidade, confiabilidade, reconfigurabilidade e alta performance provenientes de seus recursos sobre a Internet sem necessitar do gerenciamento de infraestruturas complexas pelos seus clientes.

A computação em nuvem é a distribuição de um enorme poder computacional acessível pela Internet, e não nas máquinas locais. As organizações com seus centros de dados privados também trabalham com um princípio semelhante. A computação em nuvem permite que as organizações movam seus recursos para onde precisam de mais poder de processamento para seus aplicativos, em vez de desperdiçar os recursos que não são utilizados em todo o seu potencial. O poder de processamento tradicional é transferido para o processamento centralizado em *Datacenters,* a mudança permite computação ou poder de processamento como uma mercadoria que pode ser negociada pela Internet.

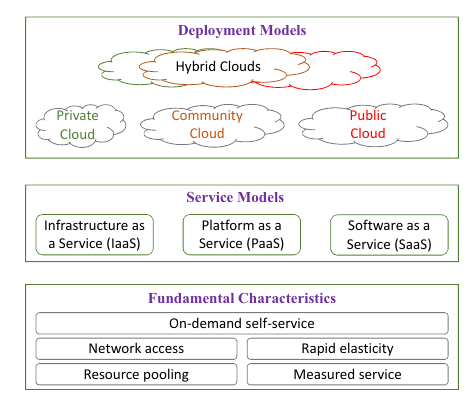
A computação em nuvem é definida de várias maneiras, mas o Instituto Nacional de Padrões e Tecnologia (*National Institute of Standards and Technology* - NIST) define que “a computação em nuvem é um modelo para permitir acesso de rede onipresente, conveniente e sob demanda a um conjunto compartilhado de recursos de computação configuráveis (por exemplo, redes , servidores, armazenamento, aplicativos e serviços) que podem ser rapidamente provisionados e liberados com mínimo esforço de gerenciamento ou interação com o provedor de serviços”.

Computação em grade (*Grid Computing*): é o processo de resolver os problemas massivos de computação em partes de problemas menores e resolvê-los em máquinas ou computadores simples ou de baixo desempenho para obter o resultado final para grandes problemas, é um sistema de computação distribuído onde um grupo de computadores são acoplados para formar uma máquina virtual, distribuindo as tarefas entre várias máquinas da grade.

A computação utilitária (*Utility Computing*): por volta de 1960, os preços do poder de processamento ou computação eram altos para qualquer finalidade, então surgiu a ideia de compartilhar recursos de computação. O objetivo é integrar servidores, sistemas de armazenamento e aplicativos distribuídos ao redor do mundo para compartilhar com vários usuários. O compartilhamento permitiria que os usuários usassem e compartilhassem os recursos do computador e os clientes que usam os recursos pudessem pagar pelos serviços usados apenas pelo período em que usaram os serviços.

A computação autônoma é um sistema que possui recursos de autogerenciamento.

Os recursos de computação em nuvem dependem dos clusters implantados (computação em grade) com várias funcionalidades de computação utilitária e autônoma.

As características comuns da computação em nuvem são: grande escala, virtualização, baixo custo, geograficamente distribuída, orientada a serviços, computação resiliente e segurança avançada para serviços. Além das características acima, a nuvem também deve ter as seguintes características fundamentais:

1) ***On-demand self-service*** *(*Autoatendimento sob demanda): usuários finais com a necessidade de usar os recursos de computação em um determinado momento (por exemplo, tempo de CPU, armazenamento de rede, software e assim por diante) de forma automática e conveniente, iniciam e param de usá-los sem qualquer interferência humana.

2) ***Network access*** (Acesso à rede): Os recursos de computação fornecidos pela Internet e podem ser usados por vários aplicativos de diferentes tipos de dispositivos, como laptops, desktops e dispositivos móveis, de acordo com os requisitos e disponibilidade dos usuários finais.

3) **Resource pooling** (Pool de recursos): os provedores de serviços em nuvem agrupam (grupos) todos os seus recursos de computação para atender a vários usuários finais usando o modelo de multialocação “com diferentes recursos físicos e virtuais atribuídos e reatribuídos dinamicamente de acordo com a demanda do consumidor”. O usuário final deve ser capaz de usar recursos independentemente de sua localização para oferecer suporte ao pool de recursos independente de localização.

4) ***Physical transparency or Rapid elasticity*** (Transparência física ou elasticidade rápida): os usuários finais podem alterar seus recursos de recursos automaticamente para escalar sempre que quiserem usar mais recursos e liberá-los assim que terminarem de usar os serviços para escalar para baixo. Para usuários finais, os recursos estão disponíveis para configuração com etapas simples para escalar suas operações e vice-versa. Esses recursos não se limitam aos usuários finais; eles aumentam o uso de serviços para atender às suas necessidades de pico a qualquer momento.

5) ***Pay per use or Measured Service*** (Pagamento por uso ou serviço medido): embora todos os recursos sejam agrupados e compartilhados entre vários locatários, os usuários finais precisam ser cobrados apenas pelos serviços em nuvem que usaram. Isso deve ser cuidado com um mecanismo adequado para medir os serviços utilizados por cada cliente.

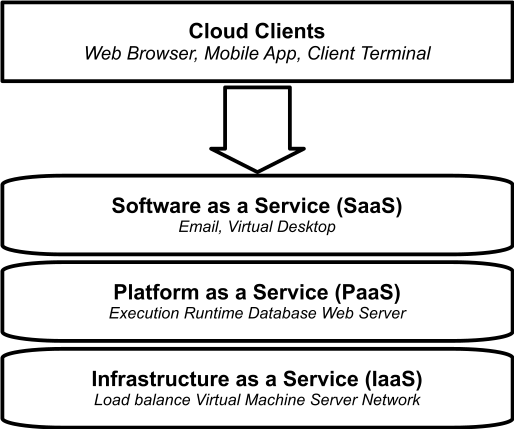
6) **Serviços móveis em nuvem**: Como o uso de dispositivos móveis aumentou na última década, os dados gerados por indivíduos também aumentaram tremendamente. Para conquistar mais clientes, muitos provedores de serviços em nuvem começaram a oferecer suporte a dispositivos móveis para usar ou acessar serviços em nuvem. Usando esses aplicativos ou interfaces móveis, os clientes podem armazenar seus dados ou acessar os serviços fornecidos pelos provedores de serviços em nuvem.

7) **Segurança na Nuvem**: Com a evolução da tecnologia e dos serviços em nuvem, muitos dos usuários estão usando os serviços em nuvem, mas ainda há um problema essencial na nuvem que precisa ser resolvido, como a segurança dos dados na nuvem. Existem muitas técnicas de criptografia e protocolos de segurança para proteger os dados, mas com o rápido crescimento da tecnologia e do poder de processamento disponível para os invasores, torna-se fácil quebrar algumas das técnicas de criptografia existentes. Portanto, ainda há espaço para novas técnicas de criptografia e protocolos de segurança para operações futuras seguras de computação em nuvem.

8) **Cloud Design**: O desenvolvimento de serviços em nuvem levou à rápida adoção dos serviços mesmo em mercados tradicionais. Os mercados centrados na nuvem têm vantagens sobre os mercados convencionais porque é conveniente para a geração mais jovem usar os serviços. Os mercados baseados em nuvem atraem mais novos clientes sem estar fisicamente presente em muitos locais com várias lojas em cada cidade.

**Característica conceitual - orientada para o serviço**

O conceito “orientado a serviços” é semelhante, mas mais prático do que o conceito de Arquitetura Orientada a Serviços (SOA) na computação em grade. Através da virtualização e outras tecnologias, a arquitetura subjacente é abstraída sem expor muito ao usuário. A abstração reduz a necessidade de os usuários aprenderem os detalhes da arquitetura da nuvem e o limite do desenvolvimento de aplicativos. Ao mesmo tempo, os principais elementos da arquitetura subjacente podem ser acessados de forma simples pelos usuários que podem consumir toda a capacidade facilmente explorando os parâmetros do sistema, como desempenho de processamento e capacidade de armazenamento. Em geral, de acordo com o tipo de capacidade fornecida, os serviços de computação em nuvem são amplamente divididos em três categorias: Infraestrutura como serviçoou*Infrastructure-as-a-Service* (IaaS), Plataforma como serviço ou *Platform-as-a-Service (PaaS)* e Software como serviço Serviço ou *Software-as-a-Service (SaaS)*.



***Infrastructure-as-a-Service*** ***(IaaS)*** é a entrega de enormes recursos de computação, como capacidade de processamento, armazenamento e rede. Tomando como exemplo o armazenamento, quando um usuário utiliza o serviço de armazenamento de computação em nuvem, ele paga apenas pela parte que consome, sem comprar nenhum disco ou mesmo saber a localização dos dados com os quais lida. Às vezes, o IaaS também é chamado de Hardware-as-a-Service (HaaS).

***Platform-as-a-Service (PaaS)***geralmente abstrai as infraestruturas e suporta um conjunto de interfaces de programas de aplicativos para aplicativos em nuvem. É a ponte intermediária entre o hardware e o aplicativo. Devido à importância das plataformas, muitas grandes empresas querem agarrar a chance de dominar a plataforma de computação em nuvem como a Microsoft faz na época do computador pessoal. Exemplos bem conhecidos são o Google App Engine e a Plataforma de Serviços Azure da Microsoft.

***Software-as-a-Service (SaaS)*** visa substituir os aplicativos executados no PC. Não há necessidade de instalar e executar o software especial em seu computador se você usar o SaaS. Em vez de comprar o software por um preço relativamente mais alto, basta seguir o padrão de pagamento conforme o uso, o que pode reduzir o custo total. O conceito de SaaS é atraente e alguns softwares rodam bem como computação em nuvem, mas a demora da rede é fatal para aplicações em tempo real ou semi-real como jogos 3D online.

Também existem muitos recursos de nuvem que não podem ser classificados em infraestrutura, plataforma ou software. A *App Store* da *Apple* é uma computação em nuvem criativa e famosa na área sem fio. Os serviços de software são vendidos no estilo pay-per-use. Porém rodar em terminais como telefones 3/4/5G em vez de grandes centros de dados é diferente de SaaS.

Na área de jogos online, poderosos servidores de jogos fornecem as interações de milhões de jogadores. Os jogadores usam a capacidade de computação em nuvem sem realmente perceber essa tecnologia. Os recursos de *e-books* na Amazon também são serviços em computação em nuvem. Esses serviços dificilmente têm substituição e construir outro EC2 (*Amazon Elastic Compute Cloud*) é muito mais fácil do que possuir tantos recursos computacionais. Todos esses serviços são tão importantes quanto IaaS, PaaS e SaaS e devem ser totalmente estudados.

**Característica técnica 1) – Baixo Acoplamento ou Acoplamento Fraco (*Loose Coupling*)**

O acoplamento fraco é o fundamento técnico da computação em nuvem e vai além do método de acoplamento fraco da interação do aplicativo. Através da virtualização ou outras tecnologias, as infraestruturas são separadas em lógica ou física. O comportamento de uma parte dificilmente afeta outras partes.

Por exemplo, a plataforma é uma camada abstrata que pode isolar diferentes aplicativos em execução nela. O mais importante de tudo é que toda a computação em nuvem é executada em um modelo cliente-servidor. Os clientes ou usuários da nuvem se conectam livremente com servidores ou provedores de nuvem. Todos os usuários quase não têm dependência de dados ou controle. Mas a dependência de dados desempenha um papel fundamental em HPC (*High Performance Computing*). Podemos ter apenas formalizações:

Os usuários compõem os conjuntos de usuários *Uset1, Uset2, ..., Usetm (m ≥ 1)*.

Os provedores compõem os conjuntos de provedores *Pset1, Pset2, ..., Psetn (n ≥ 1)*.

O conjunto do usuário Uset*i* desacoplado do conjunto do provedor *Psetj* é descrito como *Set (Useti, Psetj)*.

Três propriedades são mostradas a seguir:

Os conjuntos do usuário são independentes:

*Useti ∩ Usetj = φ (0 ≤ i, j ≤ m, i ≠ j)*.

Conjuntos provedores são independentes:

*Pseti ∩ Psetj = φ (0 ≤ i, j ≤ m, i ≠ j)*.

Os conjuntos de baixo acoplamento (onde o usuário se conecta ao provedor) são independentes:

*Set (Useti1, Psetj1) ∩ Set (Useti2, Psetj2) = φ*.

Tomando como um exemplo simples, supondo que os provedores são *Google*, *Yahoo!* e *Bing*. Os usuários do serviço de pesquisa não podem usar três mecanismos de pesquisa ao mesmo tempo (em termos de tempo absoluto) e podem ser divididos em conjuntos de usuários independentes:

*UsetGoogle ∩ UsetYahoo! ∩ UsetBing = φ*.

Os centros de dados por trás de interfaces de busca simples são independentes entre os três provedores:

*PsetGoogle ∩ PsetYahoo! ∩ PsetBing = φ*.

Os conjuntos de baixo acoplamento são independentes:

*Set (UsetGoogle, PsetGoogle) ∩ Set (UsetYahoo!, PsetYahoo! ) ∩ Set (UsetBing, PsetBing) = φ.*

Usuários de nuvem independentes podem induzir muitos outros recursos, como ausência de estado, escalabilidade e forte tolerância a falhas. Um exemplo oposto é o forte acoplamento de sistemas HPC que se concentra na resolução de problemas científicos.

Normalmente, existem muitas dependências de dados ou sincronizações globais em diferentes iterações para suportar o alto atraso entre os nós de computação. Este tipo de sistema utiliza uma rede de alta velocidade, por ex. *InfiniBand* em vez de *Ethernet* padrão industrial, que é muito mais barato e amplamente suportado.

**Característica técnica 2) - Forte tolerância a falhas**

Para a Computação Paralela, existem muitos métodos tolerantes a falhas. Em baixo nível, sempre existem alguns mecanismos de correção de falhas com hardware específico, em alto nível, muitas aplicações específicas são estudadas com métodos voltados para algoritmos.

O ponto de verificação é um dos métodos mais eficazes no nível médio. Em sistemas de computadores paralelos de grande porte, o intervalo de duas falhas pode ser menor que o tempo de execução do aplicativo, alguns aplicativos de computação científica podem ser executados por semanas, no entanto, podem ocorrer vários erros triviais ou fatais durante todo o tempo de execução.

A tecnologia tolerante a falhas torna-se crítica nesta condição, caso contrário, o sistema teria menos chances de concluir tarefas computacionais mais demoradas. Como um erro mínimo é inaceitável e reprocessar a tarefa custa muito tempo em muitas aplicações científicas, todos os estados de computação que são salvos periodicamente no armazenamento estável serão revertidos para um ponto de verificação especial se ocorrer um erro.

É desnecessário manter todos os estados dos sistemas no caso da computação em nuvem, pois quase não há dependência entre duas transações. A falha de uma transação não afeta a outra e a falha parcial do sistema não causará uma reação em cadeia.

Existem principalmente quatro locais onde podem ocorrer falhas na computação em nuvem: provedor interno, provedor transversal, provedor usuário e usuário transversal.

Se ocorrer uma falha no provedor, o backup ou redundância do provedor substituirá a parte que falhou. Interromper serviços e reiniciar são outro método comum se os serviços não forem pontuais ou urgentes. A natureza de acoplamento frouxo do provedor, por exemplo O data center faz com que esse tipo de falha não seja difícil de lidar.

Se ocorrer uma falha entre os provedores, a transação entre provedores será cancelada e retornará com uma informação de erro. O redirecionamento para outros provedores é um método universal que envolve o balanceamento de carga de todo o sistema de nuvem.

Felizmente, apenas algumas poucas transações são causadas pelo gerenciamento em segundo plano envolvendo mais de um provedor que pode ser executado vez por dia ou mesmo por semana.

Existem muitos motivos, como congestionamento de rede, colapso do navegador, tempo limite de solicitação, provedor ocupado e ataque de hackers que podem causar falhas entre o provedor e o usuário. Se não envolver alguns elementos-chave, essas falhas são omitidas e o usuário pode tentar na próxima vez. Os algoritmos *tolerantes a falhas bizantinos* (BFT) são muito importantes no aspecto da tecnologia porque exploração maliciosa de falhas provedor-usuário são cada vez mais comuns e podem fazer com que os nós defeituosos exibam um comportamento arbitrário difícil de lidar. Se envolver elementos-chave que causam perda real ao usuário, tais como perda financeira, operações adicionais são necessárias para garantir a segurança da transação e ao mesmo tempo, o *log* do sistema e o crédito do provedor podem lidar com esses incidentes.

O usuário não se conecta apenas com o provedor, mas também com outros usuários. Muitos usuários participam de atividades e compartilham vários recursos críticos, o acesso inseguro a recursos críticos pode causar caos em sistemas de computação em nuvem. Existem métodos de nível de hardware, nível de sistema operacional e nível de software para proteger recursos críticos onde o provedor faz a mediação para arbitrar a disputa entre os usuários.

Tanto o amadurecimento da tecnologia quanto a sociedade garantem a forte característica tolerante a falhas da computação em nuvem.

**Característica econômica - Modelo de negócios**

O modelo de negócios é a característica chave para distinguir computação em grade e computação em nuvem. Enquanto a Computação em grade é fortemente suportada por governo e instituições acadêmicas em sua maior parte, a computação em nuvem é suportada principalmente por gigantescas empresas de TI, que planejam os investimentos em computação em nuvem afim de obter retorno sobre o investimento (ROI) a curto prazo ou superar os concorrentes de mercado no longo prazo.

Existem muitos modelos de negócios, especialmente modelos de como pagar na computação em nuvem. No caso de Pay-per-use, aproveita-se o conceito de computação utilitária, a capacidade de processamento, armazenamento e rede na computação em nuvem é um serviço de utilidade e esses serviços podem estar disponíveis sempre que o usuário os solicitar. Os usuários pagam aos provedores de serviços com base no uso desses recursos.

Os serviços em nuvem são fins em si mesmos para o usuário final, portanto existem duas categorias de usuários da nuvem: usuário final e usuário mediano.

O usuário final consome serviços de nuvem para uso próprio. O usuário mediano consome serviços em nuvem e fornece serviços profissionais a terceiros de forma econômica. Às vezes, o usuário final não paga pelos serviços de nuvem diretamente.

Por exemplo, os jogadores de jogos online pagam pelo jogo específico de acordo com o tempo que ficam online, então é cobrada uma liquidação parcial para manter o funcionamento do sistema em nuvem. Este processo não é transparente para o usuário final.

O usuário mediano geralmente paga diretamente pelos serviços de nuvem consumidos, não é necessário gerenciar hardwares e softwares complexos, aprender a usar ferramentas e ganhar experiência com a tecnologia de computação em nuvem. O negócio da computação em nuvem vai muito além disso, existem muitos acessos gratuitos à computação em nuvem como por exemplo, é grátis pesquisar com o Google, enviar e-mail com o Hotmail e encontrar novos amigos com o Facebook onde a “economia de atenção” pode ser a ideia-chave.

**Característica da experiência do usuário - Facilidade de uso**

Na computação em nuvem, a experiência do usuário melhora muito comparada a seus antecessores, como a computação em grade. O serviço de nuvem é um meio de fornecer uma boa experiência e os serviços valiosos são facilmente acessados pelo usuário.

Existem três razões pelas quais a computação em nuvem deve ser fácil de usar:

Primeiro, a maioria dos provedores de nuvem oferece interfaces baseadas na Internet que são mais simples do que outras interfaces de programa de aplicativo (API). Essas interfaces são simples e elegantes o suficiente para ocultar o processamento de negócios por trás. As interfaces podem permanecer as mesmas ignorando se o processamento de negócios mudou ou não.

Em segundo lugar, para a experiência do usuário de aplicativos da web as interfaces são independentes do conteúdo. O desenvolvimento de um aplicativo da Web tem um fluxo completo que pode ser dividido em três estágios, incluindo análise das necessidades do usuário, design de função e implementação do programa. No método top-down, o design da experiência do usuário é o fundamento de todo o design funcional. As facetas da experiência do usuário são úteis, estudáveis, utilizáveis, valiosas, desejáveis, encontráveis, confiáveis e acessíveis.

Em terceiro lugar, a web 2.0 aumenta as interações entre usuários e provedores da web. A web foi originalmente projetada para transportar hipertexto à medida que surgem os rápidos e ricos desenvolvimentos de conteúdos cada vez mais sofisticados, a web é naturalmente utilizada como uma interface remota de software. A web 2.0 deve ser o continuum da experiência do usuário e confunde a linha entre o software e a Internet.

**Outras características**

Existem outras características importantes, como base TCP/IP, virtualização e alta segurança. O TCP/IP fornece entrega confiável, um serviço orientado à conexão entre aplicativos remotos e é amplamente utilizado na computação em nuvem. Embora os protocolos de rede possam ser privados no back-end do data center, a maioria dos usuários da nuvem se conecta aos provedores por meio de TCP/IP.

Os recursos de nuvem geralmente são virtualizados como um serviço pela Internet. Até o presente, muitas infraestruturas de computação em nuvem consistem em data centers que utilizam tecnologias de virtualização abstraindo a comunidade da infraestrutura em diferentes níveis.

A alta segurança da computação em nuvem é alcançada principalmente por meio de três maneiras: O baixo acoplamento, faz com que o sistema de computação em nuvem funcione bem quando parte dele é destruído. Em segundo lugar, a abstração, a virtualização e a privação do provedor de nuvem evitam expor os detalhes das implementações correspondentes. Em terceiro lugar, a tecnologia que coopera com a lei é a proteção da computação em nuvem.

**Segurança:**

A segurança na nuvem evoluiu com a adoção da computação em nuvem. O conceito de segurança é mais crítico com a adoção de serviços em nuvem por mais usuários. Com muitos usuários espalhados mundialmente, se os serviços em nuvem não forem protegidos adequadamente, grandes quantidades de dados de clientes ficarão vulneráveis a invasores de todo o mundo.

A segurança na nuvem pode ser obtida de várias formas, como proteção contra ataques de rede e ataques de software, detecção de intrusão, controle de acesso, análise de comportamento anormal, análise de vírus, análise de *malware*, análise de Cavalos de Tróia (*Trojans*) e assim por diante.

As medidas de segurança, que garantem a segurança da nuvem, são apresentadas a seguir:

1) **Senha**: Para proteger os serviços em nuvem de ataques simples contra os controles de acesso, os usuários são incentivados a usar uma senha exclusiva para acessar os serviços baseados em nuvem. Os clientes não devem usar senhas simples ou reutilizar a senha que já foi usada em alguns outros serviços pela Internet. Os provedores de serviços em nuvem devem garantir que não haja relação direta com nomes de usuário e senhas armazenados em seu banco de dados. Caso haja uma violação no provedor de serviços em nuvem, fica difícil para os invasores combinar os nomes de usuário com as senhas.

2) **Recuperação de Acesso**: Os clientes devem utilizar dados ou questões confidenciais, para recuperar o seu controle de acesso à nuvem caso tenham esquecido a senha. Esta informação pode ser usada para recuperar o acesso à *Cloud*. Os usuários não devem usar as informações que podem ser obtidas usando engenharia social ou apenas verificando algumas informações em seus perfis de redes sociais. Como a maioria dos detalhes pessoais são publicados nos sites de rede. Usando essas informações, os invasores podem facilmente obter acesso à nuvem sem conhecer a pessoa.

3) **Criptografia**: O uso de uma boa técnica de criptografia por provedores de serviços em nuvem sempre protege os dados do cliente, como a Criptografia Homomórfica, porém o uso de uma técnica de criptografia homomórfica ainda não é completamente viável em cenários de tempo real.

4) **Gerenciamento de senhas**: Os usuários não devem reutilizar suas senhas e os provedores de serviços em nuvem devem incentivá-los a usar senhas fortes com caracteres especiais, símbolos do alfabeto e números. É difícil para os usuários se lembrarem de todos os seus nomes de usuário e senhas. Eles precisam ter uma ferramenta de gerenciamento adequada para armazenar seus nomes de usuário e 5 senhas para protegê-los de qualquer pessoa que tenha acesso a eles.

5) **Autenticação multifator**: A autenticação multifator adiciona uma camada extra de segurança à abordagem tradicional para acessar os serviços em nuvem em vez de nome de usuário e senha. Para acessar os serviços em nuvem usando autenticação multifator, os clientes precisam ter dois ou mais fatores para acessar os serviços em nuvem para autenticá-los como um usuário genuíno da nuvem. Esses fatores podem ser baseados em qualquer coisa, como conhecimento (algo conhecido pelo usuário, como uma outra senha), algo que o usuário possui (recursos biométricos) e algo que o usuário possui (chave RSA ou chaves baseadas em USB ou texto aleatório enviado para seu celular).

Os provedores de serviços em nuvem devem oferecer suporte a métodos de autenticação multifator e incentivar os clientes a usar a autenticação multifator em vez de usar autenticação simples usando nome de usuário e senha. Dessa forma, será fácil defender-se contra o acesso não autorizado aos dados dos clientes, mesmo que alguém tenha as credenciais dos clientes; eles não terão acesso a outros fatores.

6) **Monitor de login**: provedores de serviços em nuvem e clientes precisam monitorar dispositivos recentes usados para acessar serviços em nuvem. Com base nessas informações, os usuários podem identificar se alguém fez login com suas credenciais e alterar suas senhas em caso de login suspeito de dispositivos ou locais desconhecidos. Os provedores de serviços em nuvem precisam melhorar as estatísticas de login com detalhes adequados para todos os dispositivos conectados para acessar os serviços em nuvem para todos os clientes.

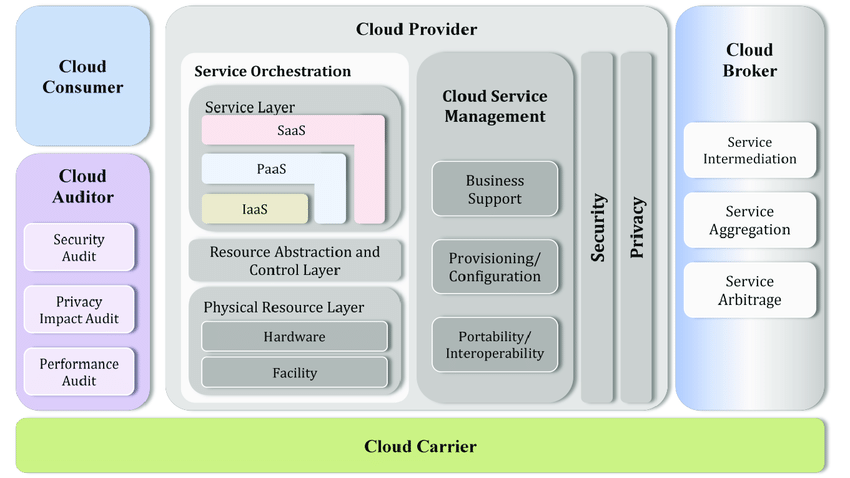
7) **Dispositivos pessoais**: Os clientes devem ter cuidado onde estão fazendo login na nuvem para acessar os serviços. Eles devem evitar usar o dispositivo de outra pessoa, pois podem ter *keyloggers* (um programa que salva todas as teclas pressionadas em um dispositivo, enquanto o programa está em execução). Nesses dispositivos, se tiverem tais aplicativos, os invasores obterão credencial de usuário para a nuvem comprometendo a segurança dos clientes.

8) **Vírus, *Malware* e Cavalos de Tróia**: os clientes devem ter bons aplicativos antivírus e *anti-spyware* em seus dispositivos. Se eles não tiverem proteção adequada de seus dispositivos, que usam para usar os serviços em nuvem, podem ter algum vírus ou *malware* que armazena as credenciais do usuário e obtém acesso aos serviços em nuvem, deixando seus dados pessoais e confidenciais para pessoas não autorizadas ou invasores.

**Arquitetura:**

**A. Descrevendo o modelo NIST**

De acordo com o NIST, cinco atores principais são identificados na computação em nuvem, o diagrama do Modelo de Referência Conceitual do NIST mostra os atores que são discutidos abaixo:



a. ***Cloud Consumer*** - Consumidor de nuvem: uma pessoa ou organização que inicia e mantém uma associação comercial e requer serviços de fornecedores de serviços de nuvem.

b. ***Cloud Provider*** - Provedor de nuvem: uma pessoa ou organização envolvida no fornecimento de serviços de computação em nuvem para pessoas ou organizações interessadas.

c. ***Cloud Auditor*** – Auditor de Nuvem: Uma organização encarregada de conduzir uma avaliação independente da computação em nuvem e determinar a eficácia e segurança dos sistemas.

d. ***Cloud Broker*** – Corretor de Nuvem: uma organização ou indivíduo terceirizado que atua como intermediário entre consumidores e provedores de nuvem. Ele/ela é útil para negociar os termos e condições do contrato de compra de serviços em nuvem.

e. ***Cloud Carrier*** – Operador de Nuvem: Uma pessoa, organização ou entidade intermediária que fornece conectividade e transporte de serviços de nuvem do provedor de nuvem para os consumidores de nuvem. O diagrama que mostra as interações entre os atores na computação em nuvem descreve o relacionamento entre os atores na computação em nuvem. Destacando a relevância de cada parte na atualização da entrega, uso e manutenção de serviços em nuvem. Além disso, o NIST também fornece cinco características da nuvem.

**a. Autoatendimento sob demanda:**

Os recursos de computação são alocados aos usuários com base em seus requisitos predefinidos. Recursos como tempo de processamento do servidor e espaços de armazenamento são alocados automaticamente sem interferência humana.

**b. Amplo acesso à rede:**

*Thick* ou *Thin-clients* (variando de *tablets*, *laptops*, estações de trabalho a telefones celulares) desfrutam de acesso à nuvem usando mecanismos padrão.

**c. Pool de recursos:** os modelos multilocatários agrupam recursos para fornecer serviços a vários consumidores. Esses serviços podem ser customizados para atender a peculiaridade dos requisitos de cada consumidor.

d. Elasticidade rápida: descreve a capacidade dos provedores de nuvem de implantar rapidamente recursos escaláveis a pedido dos consumidores. A arquitetura de nuvem deve ser capaz de aumentar e diminuir continuamente entre cada solicitação dos usuários, garantindo que os recursos pareçam ilimitados para o consumidor.

e. Serviço medido: o provedor de nuvem monitora e controla certos aspectos do serviço de nuvem para garantir o uso eficaz de recursos e o planejamento preditivo geral. Isso é obtido pelo uso de um recurso de medição embutido no sistema.

A. Describing the NIST Model According to NIST, five major actors have been identified in cloud computing [7]. The NIST Conceptual Reference Model diagram in [7] shows the actors which are discussed below.

a. Cloud Consumer: A person or organization that starts and keeps a business association with and requires services from suppliers of cloud services [22].

b. Cloud Provider: A person, organization engaged in supplying cloud computing services to interested persons or organizations.

c. Cloud Auditor: An organisation in charge of conducting independent evaluation of cloud computing, and determining the systems effectiveness and security.

d. Cloud Broker: A third-party organisation or individual that serves as an intermediary between cloud consumers and cloud providers . He/she is useful for negotiating terms and conditions of the contract for the purchase of cloud services.

e. Cloud Carrier: An intermediary person, organisation or entity that provides connectivity and transport of cloud services from cloud provider to cloud consumers. The diagram showing interactions between the actors in cloud computing in [7] describes the relationship amongst actors in cloud computing. Highlighting the relevance of each party in actualising the delivery, use and maintenance of cloud services. In addition, NIST also provides five characteristics of the cloud [11]

a. On-demand self–service:

Computing capabilities are allocated to users based on their predefined requirements. Capabilities such as server processing time and storage spaces are allocated automatically without human interference.

b. Broad Network Access: Thick or thin clients (ranging from tablets, laptops, workstations to mobile phones) enjoy access to the cloud by using standard mechanisms.

c. Resource pooling: Multi-tenant models pool together resources in order to provide services to multiple consumers. These services can be customised to satisfy the peculiarity of each consumers’ requirements.

d. Rapid Elasticity: This describes the cloud providers’ ability to rapidly deploy scalable resources at the request of consumers. The cloud architecture must be able to seamlessly scale up and down amongst each requests of users, ensuring that the capabilities appear unlimited to the consumer.

e. Measured Service: The cloud provider monitors and controls certain aspects of the cloud service to ensure effective resource usage and overall predictive planning. This iachieved by the use of a metering capability embedded within the system.

B. Cloud Computing Services

A cloud architecture can be divided into the back end and front end . The front end is made visible to the user through connections to the Internet, allowing user interactions with the system [11]. The back end comprises the various cloud services models.

1) Software-as-a-Service (SaaS)

The user is offered a hosted set of software running on a platform and infrastructure owned by the cloud provider [2]. Applications are designed and developed to be simultaneously accessed by various cloud consumers over the Internet [11] . The hosted application is managed by the CSP, who maintains and ensures up-to-date running of the system. The hosted application supports multitenancy, it is available on demand and can be scaled up on down [11].

Some SaaS providers run on other cloud provider’s PaaS or IaaS offerings [2]. Examples of SaaS [21]:

a. Email and Office Productivity: Email applications, word editors and processors, spreadsheets applications, presentations applications are typical examples in this category.

b. Billing: There are applications designed to monitor and manage customer billing. This is determined by users’ system usage and subscriptions to products and services.

c. Customer Relationship Management (CRM): CRM are typical call-centre applications.

d. Financials: These are applications useful for tracking and reporting financial activities including processing of expenditure, generating invoices, payroll, and managing taxes.

2) Platform–as-a-Service (PaaS) PaaS is a development service offered to the user through the Internet [2] . The user does not require any software installation or hardware requirements, thereby saving cost. It is a middleware upon which applications are built [11] . PaaS has built–in tools, built–in-security and web service interfaces for the deployed applications [11]. The deployed application can be integrated with other applications on the same platform and interfaced with other applications outside the platform [11]. PaaS has software comprising a database, middleware and development tools [2]. Examples of PaaS [21]:

a. Business Intelligence.

b. Database.

c. Development and Testing.

d. Integration.

e. Application Deployment.

3) Infrastructure–as-a–Service.

This is delivery of servers, storage, network and operating system, as a service [2]. IaaS provides an abstract machine with operating system already installed and configured [11]. IaaS enables data to be stored in different geographical locations . IaaS providers control activities in the cloud data centres while allowing users the flexibility to deploy and manage software services themselves [2]. The user has access to a virtual computer, storage, network infrastructure, computing resources for deploying and running software [6]. The cloud provider only manages the software and hardware, such as servers, storage devices, host OS and hypervisor for virtualization [5]. A typical cloud architecture diagram services available to cloud users is shown in [5]. Examples of IaaS [21]:

a. Content Delivery Networks (CDNs): CDNs record user content and files to improve the system performance such as speed and the cost associated with the delivery content for web-based systems. This is useful for handling diverse kinds of content for delivery to any website or mobile app.

b. Backup and Recovery: This provides ability for seamless backup and restoration of files.

c. Compute: This involves server requirements for maintaining cloud systems that can be configured and provisioned dynamically.

d. Storage: Highly scalable storage ability useful for recording activities of applications, file backups and recovery and storing files are also available.

C. Cloud Computing Deployment Types

a. Private Cloud. A private cloud is exclusively setup and run for a particular enterprise, but third party organizations are given access to manage them on the behalf of the cloud owner [11]. The private cloud can be operated on-premise or off-premise. Private cloud has privacy, security and control. The cost and energy efficiency is also good [11]. Private clouds have limited scalability and are restricted to an area. Public Cloud. They are operated by a CSP, who owns the infrastructure and data centres. The infrastructure is on premise and enterprises can access services on-demand and pay-as-you-go basis [11]. Services are made available to an organizations and users over a public network through a browser [3] [6].

b. Public clouds are location independent, reliable and highly scalable, but less secure and not customizable [11].

c. Community Cloud. Community cloud is hosted by several organizations or institutions sharing common interest. Typical examples are universities using it for learning and research. Organisations may decide to manage the cloud system themselves on site or off site and may also decide to outsource the daily running of the system to a third party organization [11].

d. Hybrid Cloud. Hybrid Cloud represents a combination of either a selection or all cloud deployment types i.e. private, public or community cloud. Core activities are hosted on a private cloud, while less essential services are outsourced to a public cloud. Each of the cloud remains a unique entity, but linked together by standardized technology [11]. Hybrid clouds are subjected to network and security issues.

D. Inter Cloud Inter cloud can be simply referred to as an interconnection of clouds, like the network of networks [11]. This allows connection of multiple cloud infrastructure to make more computing resources available. There are two types of inter-cloud: the federated cloud and Multi cloud.

a. Federation Cloud: A federation cloud is an inter-cloud setup where a set of cloud providers wilfully combine their cloud infrastructure, this enables easy sharing of resources amongst the participating organizations. Every cloud provider voluntarily collaborates to exchange resources.

b. Multi-Cloud: Multi-cloud involves the use multiple independent cloud architecture acting as a single architecture. For example, organisations or enterprises could run a particular cloud activity that requires enormous resources on a private cloud and run other cloud activities requiring lower level of network resources or storage capability on a public cloud.

**Service Model**

Cloud computing can be segregated into the following service levels: Infrastructure as a Service, Platform as a Service, and Software as a Service as shown in Fig. 2. Fig. 3 presents he separation between service models with control of cloud service provider and customer of different underlying concepts in each model.

1) Infrastructure as a Service (IaaS) : Customers will get the services for a complete computing infrastructure over the Internet. Example: Amazon EC2 [10] and S3 [20].

2) Platform as a Service (PaaS) : In PaaS, customers will get the platform for the development of software applications. Example: Microsoft Azure [8] and Google AppEngine [9].

3) Software as a Service (SaaS) : Customers will be provided with the Software over the Internet. In this model, users will not get the software; instead, they get the web-based software from the service providers to the intended work. Example: Dropbox [21] and Office365[22].

4) Others-as-a-Service: Backup-as-a-service, storage-as-a-service, logistics-as-a-service, operating systems as a service(OSaaS), framework as a service(FaaS), database as a service(DaaS), network as a service(NaaS) and other models apart from the 3 primary ones were discussed in [19], [9], [23], [29] and [33]. However, only 13% of the papers examined discussed Others-as-a- Service.

**Deployment Model**

Cloud computing services are provided over the following deployment models [2] as shown in Fig. 2:

1) Public cloud : provides the cloud services for end-users by allowing them to access the services from the Internet. So, these cloud services are publicly accessible. The Cloud service provider provides the required infrastructure for end-users.

A cloud is called a "public cloud" when the services are rendered over a network that is open for public use. Public cloud services may be free.[103] Technically there may be little or no difference between public and private cloud architecture, however, security consideration may be substantially different for services (applications, storage, and other resources) that are made available by a service provider for a public audience and when communication is effected over a non-trusted network. Generally, public cloud service providers like Amazon Web Services (AWS), Microsoft and Google own and operate the infrastructure at their data center and access is generally via the Internet. AWS and Microsoft also offer direct connect services called "AWS Direct Connect" and "Azure ExpressRoute" respectively, such connections require customers to purchase or lease a private connection to a peering point offered by the cloud provider.

2) Private cloud : model is used within the organizations to meet the cloud requirements across various levels in the organizations. They will maintain their infrastructure to set up cloud services. This model will reduce the cost of using the cloud services for the organization in the long run with extra security as these private cloud models are deployed in their private networks behind their firewalls. Private cloud model can be accessed inside their network or by authenticating the user at their firewall.

Private cloud is cloud infrastructure operated solely for a single organization, whether managed internally or by a third-party, and hosted either internally or externally. Undertaking a private cloud project requires significant engagement to virtualize the business environment, and requires the organization to reevaluate decisions about existing resources. It can improve business, but every step in the project raises security issues that must be addressed to prevent serious vulnerabilities. Self-run data centers are generally capital intensive. They have a significant physical footprint, requiring allocations of space, hardware, and environmental controls. These assets have to be refreshed periodically, resulting in additional capital expenditures. They have attracted criticism because users "still have to buy, build, and manage them" and thus do not benefit from less hands-on management, essentially "[lacking] the economic model that makes cloud computing such an intriguing concept".

3) Community cloud : model is used to deploy the cloud infrastructure that can be shared between the several organizations with similar goals. It is comparable to a private cloud but shared among some organizations.

4) Hybrid cloud deployment model : model is used where the customers use more than one model to meet the goals of their organizations or end-users them self.

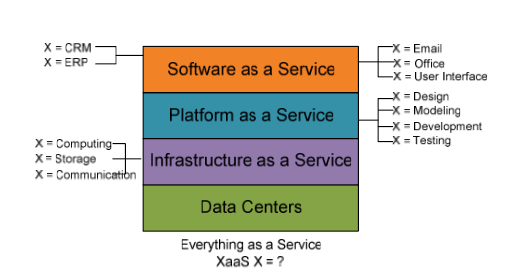
Hybrid cloud is a composition of two or more clouds (private, community or public) that remain distinct entities but are bound together, offering the benefits of multiple deployment models. Hybrid cloud can also mean the ability to connect collocation, managed and/or dedicated services with cloud resources. Gartner, Inc. defines a hybrid cloud service as a cloud computing service that is composed of some combination of private, public and community cloud services, from different service providers. A hybrid cloud service crosses isolation and provider boundaries so that it can't be simply put in one category of private, public, or community cloud service. It allows one to extend either the capacity or the capability of a cloud service, by aggregation, integration or customization with another cloud service.

Varied use cases for hybrid cloud composition exist. For example, an organization may store sensitive client data in house on a private cloud application, but interconnect that application to a business intelligence application provided on a public cloud as a software service. This example of hybrid cloud extends the capabilities of the enterprise to deliver a specific business service through the addition of externally available public cloud services. Hybrid cloud adoption depends on a number of factors such as data security and compliance requirements, level of control needed over data, and the applications an organization uses.

Another example of hybrid cloud is one where IT organizations use public cloud computing resources to meet temporary capacity needs that cannot be met by the private cloud. This capability enables hybrid clouds to employ cloud bursting for scaling across clouds. Cloud bursting is an application deployment model in which an application runs in a private cloud or data center and "bursts" to a public cloud when the demand for computing capacity increases. A primary advantage of cloud bursting and a hybrid cloud model is that an organization pays for extra compute resources only when they are needed. Cloud bursting enables data centers to create an in-house IT infrastructure that supports average workloads, and use cloud resources from public or private clouds, during spikes in processing demands. The specialized model of hybrid cloud, which is built atop heterogeneous hardware, is called "Cross-platform Hybrid Cloud". A cross-platform hybrid cloud is usually powered by different CPU architectures, for example, x86-64 and ARM, underneath. Users can transparently deploy and scale applications without knowledge of the cloud's hardware diversity. This kind of cloud emerges from the raise of ARM-based system-on-chip for server-class computing.

# Deployment models of Cloud Computing

2.1. A Hierarchical View of Cloud Computing Most of the current clouds are built on top of modern data centers. It incorporates Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), and provides these services like utilities, so the end users are billed by how much they used. Figure 1 shows a hierarchical view for cloud computing.

Figure 1: Hierarchical View of Cloud Computing

Data Centers: This is the foundation of cloud computing which provides the hardware the clouds run on. Data centers are usually built in less populated areas with cheaper energy rate and lower probability of natural disasters. Modern data centers usually consist of thousands of inter-connected servers.

Infrastructure as a Service: Built on top of data centers layer, IaaS layer virtualizes computing power, storage and network connectivity of the data centers, and offers it as provisioned services to consumers. Users can scale up and down these computing resources on demand dynamically. Typically, multiple tenants coexist on the same infrastructure resources [1]. Examples of this layer include Amazon EC2, Microsoft Azure

Platform.

Platform as a Service: PaaS, often referred as cloudware, provides a development platform with a set of services to assist application design, development, testing, deployment, monitoring, hosting on the cloud. It usually requires no software download or installation, and supports geographically distributed teams to work on projects collaboratively. Google App Engine, Microsoft Azure, Amazon Map Reduce/Simple Storage Service are among examples of this layer. Software as a Service: In SaaS, Software is presented to the end users as services on demand, usually in a browser. It saves the users from the troubles of software deployment and maintenance. The software is often shared by multiple tenants, automatically updated from the clouds, and no additional license needs to be purchased. Features can be requested on demand, and are rolled out more frequently. Because of its service characteristics, SaaS can often be easily integrated with other mashup applications. An example of SaaS is Google Maps, and its mashups across from the internet. Other examples include Salesforce.com and Zoho productivity and collaboration suite.

The dividing lines for the four layers are not distinctive. Components and features of one layer can also be considered to be in another layer. For example, data storage service can be considered to be either in as IaaS or PaaS. Figure 1 suggests a hierarchical relationship among the different layers; however, it does not mean the upper layer has to be built on top its immediate lower layer. For example, a SaaS application can be built directly over IaaS, instead of PaaS. In the cloud computing environment, everything can be implemented and treated as a service. Figure 1 shows a few examples of what can be treated as a service in different layers.

3. Existing Cloud Computing Architectures

Both academia and industry have been active on cloud computing research, and several cloud computing architectures have been proposed. In [5], IBM considers current single- providers cloud as limited resource, and the lack of interoperability among cloud providers prevents deployment across different clouds. A cloud computing architecture named Reservoir was proposed to create a federation from multiple cloud providers which acts as a global fabric of resources that can guarantee the required SLA. In Reservoir architecture, the computational resources within a site are partitioned by a virtualization layer into virtual execution environments (VEEs).

A service application is decomposed into a set of software components/services running on VEEs on the same or different VEEs within a site or across from different sites. However, Reservoir architecture does not allow a component/service to run on its duplicates on different VEEs; Moreover, computing resources are abstracted as hosting service which might not be necessarily true for all clouds. In [6], a software platform for .NET based cloud computing named Aneka was introduced.

Aneka is a customizable and extensible service oriented runtime environment that enables developers to build .NET applications with the supports of APIs and multiple programming models. Aneka is a service-oriented, pure PaaS cloud solution. In [7], Rajkumar and his colleagues explained a market-oriented cloud architecture in detail used by Aneka, which regulates the supply and demand of cloud resources to achieve market equilibrium, adds economic incentives for both cloud consumers and providers, and promotes QoS-based resource allocation mechanisms that differentiates service request based on their utility. The key component of this architecture is SLA (Service Level Agreement) Resource Allocator which is consisted of Service Request Examiner and Access Control, VM (Virtual Machines) monitor, Service Request Monitor, and Request Dispatcher. Based on the feedback from VM and Service Request monitors, the dispatcher routes the requests from users/brokers to the cloud resources that can fulfill their QoS requirements. In [8], Huang and her colleagues from IBM described a service oriented cloud computing platform that enables web-delivery of application-based services with a set of common business and operational services. The platform supports multi-tenancy feature by utilizing single application instance model. The isolation among tenants is taken care by the underline design. Other services include subscription management, federated ID management, application firewall, etc.

3.1. Issues with Current Clouds

Current cloud computing has following characteristics:

Users are often tied with one cloud provider: Even though up-front cost for a cloud computing deployment is reduced and long term lease is eliminated, much effort and money is spent on developing the application for a specific cloud platform which makes it difficult to migrate the same application onto a different cloud. Often, migration simply may mean redevelopment. For example, applications deployed on Amazon EC2 cannot be migrated easily due its particular storage framework [9].

Computing components are tightly coupled: This can be clearly explained using an analogy. Suppose one wants a new computer, this person has the choices of either buying a ready-to-use computer from a manufacturer (buying) or purchasing the components separately and building the computer in a DIY style (building). The advantages of building over buying include wider selection of components, flexibility to customize, and cheaper cost [10]. However, as the computing resources over the internet, current cloud implementations do not allow this kind of flexibility. If a customer opts to use Amazon S3 storage service, he is then stuck with other cloud computing services Amazon provides, such as EC2, Elastic Map Reduce.

Lack of SLA supports: Currently, SLA is an obstacle that prevents wide adoption for cloud computing. Cloud computing infrastructure services such as EC2 are not yet able to sign the SLA needed by companies that want to use cloud computing for serious business deployment [11].

Moreover, business is dynamic. Static SLA is not able to adapt to the changes in business needs as cloud computing promises to. Lack of Multi-tenancy supports: Multi-tenancy can support multiple client tenants simultaneously to achieve the goal of cost effectiveness. Currently, one has three types of multi-tenancy enablement approaches: virtualization, mediation and sharing [12]. To achieve the full potential of multi-tenancy, three issues remain to be solved [12]:

1. Resource sharing: To reduce the hardware, software and management cost of each tenant.

2. Security isolation: To prevent the potential invalid access, conflict and interference among tenants.

3. Customization: To support tenant-specific UI, access control, process, data, etc.

Lack of Flexibility for User Interface: UI is an important part of the application, and user experience can be a major evaluation factor for a business application. However,

cloud/SaaS users are limited with UI choices because UI composition frameworks, such as the one proposed in [13], have not been integrated with cloud computing.

Advantages of using Cloud Computing

Cloud computing simplified software development, business process, and accessing the services over the Internet. The traditional way of accessing services has changed with cloud computing. Adoption of the cloud has reduced costs, made an effective business model, and offers a great scale of flexibility for using the services. Many organizations have adopted cloud services and benefited by moving their services to the cloud. With the adoption of cloud, organizations are improving crossplatform collaboration between the developers, allowing them to do more innovations on their IT capabilities, which in turn helps the organizations to grow their business and get more revenue [1].

Referências

Swati I. Bairagi e Ankur O. Bang, “Cloud Computing: History, Architecture, Security Issues”, Special Issue National Conference “CONVERGENCE 2015”, 28 March 2015, International Journal of Advent Research in Computer and Electronics (IJARCE) - (E-ISSN: 2348-5523)

J. Surbiryala e C. Rong, "Cloud Computing: History and Overview," 2019 IEEE Cloud Summit, Washington, DC, USA, 2019, pp. 1-7, doi: 10.1109/CloudSummit47114.2019.00007.

W. -T. Tsai, X. Sun and J. Balasooriya, "Service-Oriented Cloud Computing Architecture," 2010 Seventh International Conference on Information Technology: New Generations, Las Vegas, NV, USA, 2010, pp. 684-689, doi: 10.1109/ITNG.2010.214.

C. Gong, J. Liu, Q. Zhang, H. Chen and Z. Gong, "The Characteristics of Cloud Computing," 2010 39th International Conference on Parallel Processing Workshops, San Diego, CA, USA, 2010, pp. 275-279, doi: 10.1109/ICPPW.2010.45.

M. K. Sasubilli and V. R, "Cloud Computing Security Challenges, Threats and Vulnerabilities," 2021 6th International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 2021, pp. 476-480, doi: 10.1109/ICICT50816.2021.9358709.

I. Odun-Ayo, M. Ananya, F. Agono and R. Goddy-Worlu, "Cloud Computing Architecture: A Critical Analysis," 2018 18th International Conference on Computational Science and Applications (ICCSA), Melbourne, VIC, Australia, 2018, pp. 1-7, doi: 10.1109/ICCSA.2018.8439638.

Deployment models of Cloud Computing [<https://forum.huawei.com/enterprise/en/deployment-models-of-cloud-computing/thread/1010535-893>]

A Brief History of Cloud Computing - IBM [<https://www.ibm.com/cloud/blog/cloud-computing-history>]

# Britannica - Project MAC [<https://www.britannica.com/topic/Project-Mac>]

Dataversity - A Brief History of Cloud Computing [<https://www.dataversity.net/brief-history-cloud-computing/>]

Instituito Brasileiro de Governança Pública [<https://forum.ibgp.net.br/>]