

Sistema de Arquivos Distribuídos: Comparativo Entre Ambiente Local e em Nuvem

Fábio Rogério Baenas Lopes, Henrique Pachioni Martins, Vinicius Santos Andrade.

Faculdade de Tecnologia de Bauru – Redes de Computadores (FATEC-Bauru)
CEP 17.015-171 – Bauru, SP – Brasil

fabio.baenas@gmail.com, hermartins@gmail.com, viniciusarmani@gmail.com

***Abstract.** Having an information not available twenty four hours a day uprightly is the same as not having. This study aims to present the tool options to manage and keep information available and fully. Being an option in physical environment and other in cloud, where will be done the comparatives in relation of security, accessibility, implementation, among others, of the addressed systems. It follows that the cloud growth is imminent, nevertheless, due to some technological obstacles, it will take a while for the physical system to stop being used.*

***Resumo.** Ter uma informação não disponível 24 horas por dia de forma integra é o mesmo que não ter. Este estudo tem como objetivo apresentar opções de ferramentas para gerir e manter a informação disponível e integra. Sendo uma opção em meio físico e o outro em nuvem, onde será feito comparativos em relação a segurança, acessibilidade, implementação, entre outros, dos sistemas abordados. Conclui-se que o crescimento da cloud é eminente, porém, devido alguns empecilhos tecnológicos, levará um certo tempo para o sistema físico deixar de ser utilizado.*

1. Introdução

A informação é a base de tudo, é com base nelas que conseguimos tomar nossas decisões. Por ser tão importante, perder uma informação pode custar muito para uma organização. Sendo assim, com o passar dos anos, as organizações vem investindo na tecnologia da informação, com o intuito de proteger, compartilhar e tornar-la disponível, sempre considerando o fator custo x benefício. Não adianta ter informação, se ela não estiver disponível 24 horas por dia, pois nunca sabemos quando ela será necessária. Por isso as organizações passaram a investir mais em novas tecnologias que possibilitem além da segurança, a disponibilidade, pois elas são a matéria prima de muitas empresas.

A cada dia, surgem novas tecnologias, e fica difícil para as organizações definirem qual é a mais adequada para suas necessidades, levando em consideração, que tais necessidades podem mudar com o passar do tempo, assim como a estrutura da organização, ou até mesmo o segmento, além de suprir as necessidades da organização, a tecnologia deve ser flexível.

O compartilhamento de informações se mostra um assunto muito amplo, pois, diversos fatores, sejam eles internos ou externos, influenciam no meio em que a informação deve ser compartilhada.

No cenário atual, possuímos basicamente duas formas de compartilhar a informação:

- a) Compartilhamento por meio físico;
- b) Compartilhamento em nuvem.

Ambos os modos de compartilhamento citados acima, possuem suas vantagens e desvantagens. Com tanta opção de tecnologia no mercado, fica difícil para o usuário escolher qual lhe suprirá suas necessidades de forma simples e eficiente.

Queremos demonstrar através dessa pesquisa, as principais diferenças, vantagens e desvantagens entre o sistema de arquivos distribuído físico e o sistema de arquivos em nuvem.

Este estudo apresenta na seção 2, a definição de sistemas distribuídos e suas características. Na seção 3, está descrito o sistema de arquivos distribuídos e suas características, na seção 4 é abordado o conceito de computação em nuvem. Na seção 5, estão relatados os materiais e métodos utilizados, bem como as ferramentas que são utilizadas para a criação de um sistema de arquivos distribuídos local e em nuvem, explicando o funcionamento de cada um deles, e os resultados obtidos, fazendo a comparação entre os dois. Na seção 6 são descritas as considerações finais, e ao final as referências.

2. Sistema distribuído

2.1 Definição de um sistema distribuído

Segundo Coulouris, Dollimore e Kindberg (2007), um sistema distribuído é formado por um conjunto de software e hardware, localizados em computadores distintos que estão interligados através de uma rede. Esses computadores não precisam necessariamente estar no mesmo meio físico, em certos casos, os computadores que fazem parte do mesmo sistema distribuído podem estar em continentes diferentes.

De acordo com Tanenbaun e Steen (2007), apesar do sistema distribuído ser na maioria das vezes, composto por vários computadores, ele aparece para o usuário como um único sistema. Uma particularidade interessante é que as diferenças entre os computadores e a maneira como eles se comunicam estão ocultas ao usuário e da mesma maneira ocorre com a organização interna do sistema distribuído.

Para Coulouris, Dollimore e Kindberg (2007), um sistema de arquivos distribuído permite que os programas armazenem e acessem arquivos a partir de qualquer computador em uma rede, permitindo assim que o armazenamento ocorra em servidores reduzindo a necessidade de armazenamento no disco local e facilitando *obackup* dos dados importantes de uma organização.

Com base nos autores, pode-se concluir a definição de sistema distribuído, levando em conta certas consequências descritas a seguir:

- a) Concorrência: Em uma rede de computadores, a execução concorrente de programas sempre existirá. A capacidade do sistema de manipular recursos compartilhados pode ser ampliada sempre que necessário, precisa-se apenas adicionar a rede novos recursos.

b) Inexistência de relógio global: a coordenação de programa que cooperam entre si é feita através da troca de mensagens. Todavia, existem limites para precisão na qual os computadores podem sincronizar seus relógios em uma rede – não existindo uma noção única do tempo correto, tornando-se assim, uma consequência direta do fato que a única comunicação se dá pelo envio de mensagens em rede.

c) Falhas independentes: todos os sistemas estão suscetíveis a falhas. Nos sistemas distribuídos, a falha de um computador ou a falha na rede, não resulta em um travamento total do sistema, cada componente do sistema pode falhar independentemente, deixando os outros ainda em funcionamento.

O principal motivo de se construir um sistema distribuído é o fato do mesmo proporcionar de forma simples e eficaz o compartilhamento de recursos. Podemos caracterizar como recursos tudo aquilo que está dentro da rede, isto é, desde os componentes de *software* e *hardware* até os dados inseridos pelos usuários.

2.2 Metas

De acordo com Tanenbaun e Steen (2007), o fato de se poder montar um sistema distribuído, não significa que em todos os casos, ele será a opção ideal para suprir suas necessidades. Um sistema distribuído deve oferecer fácil acesso a seus recursos, deve ocultar o fato de que seus recursos são distribuídos por uma rede e deve ser flexível.

2.2.1 Acesso a recursos

Para Tanenbaun e Steen (2007), conectar os usuários aos recursos facilita a colaboração de troca de informações, por isso a principal meta de um sistema distribuído deve ser facilitar aos usuários o acesso a recursos remoto e seu compartilhamento de maneira controlada e eficiente. Dentre os vários motivos para justificar o compartilhamento de recursos, podemos citar como principal a economia.

Por exemplo, em uma empresa de pequeno porte, que possui uma ampla sala com diversos computadores, é mais fácil comprar uma impressora e compartilhá-la com todos os computadores (que serão acessados pelos usuários) do que comprar uma impressora para cada computador.

2.2.2 Transparência na distribuição

Segundo Tanenbaun e Steen (2007), uma meta importante do sistema distribuído, é ocultar o fato de que os processos e recursos estão sendo distribuídos por computadores distintos. A capacidade de um sistema distribuído apresentar suas aplicações como se fosse um único sistema é denominado transparente.

De acordo com os autores, os aspectos mais importantes de transparência em um sistema distribuído que devem ser levados em consideração são:

a) Transparência de Acesso: oculta a diferença na representação de dados e no modo de acesso a um recurso.

b) Localização: oculta o local em que um recurso está localizado, isto é, o usuário não sabe qual a localização física de um recurso no sistema.

c) Migração: oculta que um recurso pode ser movido para outro local sem afetar seu modo de acesso;

d) Realocação: oculta que um recurso pode ser movido para outro local, mesmo quando está em uso;

e) Replicação: oculta que recursos são replicados. A replicação é feita em locais que estão próximos do verdadeiro arquivo e com o mesmo nome do arquivo original;

f) Concorrência: oculta que um recurso pode ser compartilhado por usuários concorrentes;

g) Falha: oculta a falha e a recuperação de recursos.

Apesar da transparência de distribuição ser indicada para qualquer sistema distribuído, há situações em que ocultar completamente dos usuários todos os aspectos da distribuição não é uma boa ideia, completa os autores.

2.2.3 Ciladas

De acordo com Tanenbaun e Steen (2007), existe N fatores que devem ser levados em consideração ao se desenvolver um sistema distribuído, devido a isso, Peter Deutsch, que trabalhava na *Sun Microsystems*, formulou uma lista de falsas premissas que todos adotam ao desenvolver um sistema distribuído:

- a) A rede é confiável;
- b) A rede é segura;
- c) A rede é homogênea;
- d) A topologia não muda
- e) A latência é zero;
- f) A largura da banda é infinita;
- g) O custo de transporte é zero
- h) Há só um administrador.

2.2.4 Heterogeneidade

Segundo Coulouris, Dollimore e Kindberg (2007), tanto a rede quanto os computadores que fazem parte dela são sistema heterogêneo. Tais aspectos se aplica a heterogeneidade:

- a) Redes;
- b) *Hardware* de computador;
- c) Linguagens de programação;
- d) Implementações de diferentes desenvolvedores;

Para os autores, programas escritos por diferentes desenvolvedores não podem se comunicar, a não ser que utilizem padrões comuns, por exemplo, para realizar a comunicação via rede e usar uma mesma rede, tais padrões devem ser estabelecidos.

2.2.5 Sistema abertos

De acordo com Coulouris, Dollimore e Kindberg (2007), um sistema aberto é aquele que pode ser estendido e reimplementado de várias maneiras, tanto a parte de *software* quanto a de *hardware*. O que determina se um sistema distribuído é aberto ou não, é dado principalmente pelo grau com que novos serviços podem ser adicionados e disponibilizados para uso por uma variedade de programas e usuários.

Segundo os autores, a característica de um sistema aberto é obtida a partir do momento que suas principais interfaces são publicadas. Esse processo é semelhante àquele realizado por organizações de padronizações, porém este ignora os procedimentos oficiais, que em sua maioria são pesados e lentos.

Para os autores, os projetistas dos protocolos *Internet* elaboram uma série de documentos, denominado *Requests For Comments* (RFCs), onde cada um era identificado por um número. Porém RFCs não são os únicos meios de publicação. Sistemas projetados de acordos com tais padrões são denominados sistemas distribuídos abertos.

2.2.6 Segurança

De acordo com Coulouris, Dollimore e Kindberg (2007), o *firewall* não garante o uso dos recursos dentro da *internet*, nem o uso apropriado da mesma.

Segundo o exemplo utilizado pelos autores, dois usuários trocam informações sigilosas utilizando uma rede. A segurança não é apenas a questão de ocultar o conteúdo da informação, mas também garantir que seja identificado o agente ou usuário de cada mensagem, e garantir que tal identificação esteja correta.

2.2.7 Escalabilidade

De acordo com Neuman (1994, apud TANENBAUN; STEEN 2007), no mínimo três dimensões diferentes devem ser consideradas ao medir a escalabilidade de um sistema. Em primeiro lugar, se o sistema é escalável em relação a seu tamanho, isto é, se é fácil adicionar mais usuários e recursos ao sistema. Em segundo lugar, o sistema escalável em termos geográfico é o sistema que permite que usuários e recursos estejam longe uns dos outros. Em terceiro lugar, o sistema pode ser escalável em termos administrativo, garantindo assim o fácil gerenciamento independente da dimensão e heterogeneidade do sistema.

Um sistema distribuído escalável é aquele que permanece eficiente quando há um aumento significativo no número de recursos e no número dos usuários, completa Coulouris, Dollimore e Kindberg (2007).

De acordo com os autores, um sistema distribuído escalável apresenta os seguintes desafios:

- a) Controlar o custo dos recursos físicos;
- b) Controlar a perda de desempenho;
- c) Impedir que os recursos de *software* se esgotem;
- d) Evitar gargalos de desempenho;

2.3 Tipos de Sistemas Distribuídos

A seguir iremos diferenciar os três tipos de sistema distribuído, sistemas de computação distribuído, sistemas de informação distribuídos e sistemas embutidos distribuídos.

2.3.1 Sistemas de computação distribuídos

De acordo com Tanenbaun e Steen (2007) em sistemas de computação distribuídos podemos fazer uma distinção ente dois subgrupos: computação de *cluster* e computação em grade.

2.3.1.1 Sistema de computação de *cluster*

Conforme Tanenbaun e Steen (2007) sistema de computação de *cluster* é utilizada em programação paralela na qual um único programa é executado em paralelo em varias maquinas, esse sistema tornou-se popular no momento em que os computadores ficaram mais baratos e com melhor desempenho. Um bom exemplo de computador de *clusters* é formado por *cluster Beowulf* baseado no sistema operacional *Linux* conforme a figura 1.

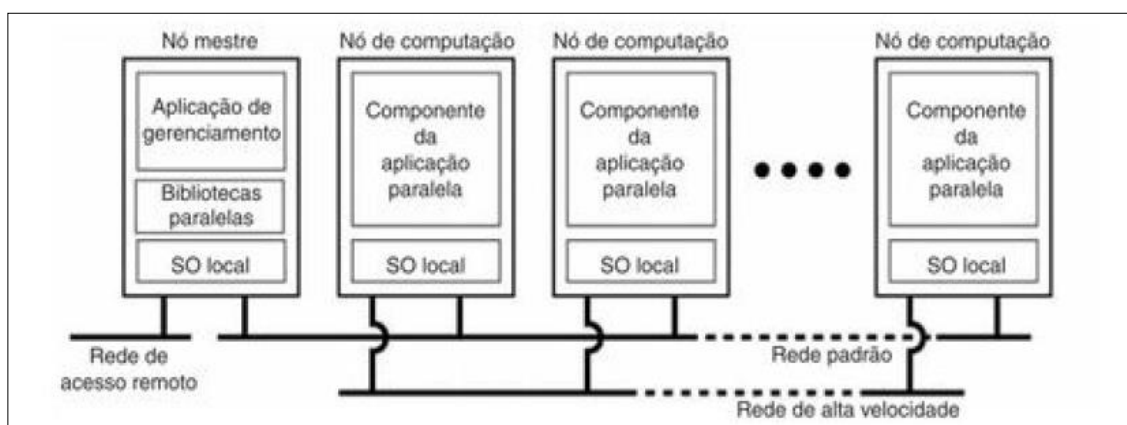


Figura 1. Exemplo de Computação em Cluster

Fonte: TANENBAUM e STEEN (2007)

2.3.1.2 Sistemas de computação em grade

Segundo Tanenbaun e Steen (2007) a principal característica desse sistema é a heterogeneidade, ou seja, nenhum princípio é aplicado com *hardwares*, sistemas operacionais, redes, etc. Nesse sistema recursos de várias organizações são combinados para assim, permitir a participação de diferentes grupos de pessoas ou organizações. Essa participação é chamada de organização virtual, grande parte do *software* é desenvolvida para prover o acesso a distintos domínios administrativos, e apenas para usuários e aplicações que pertençam a uma organização virtual específica. Um exemplo dessa arquitetura é mostrado na figura2.

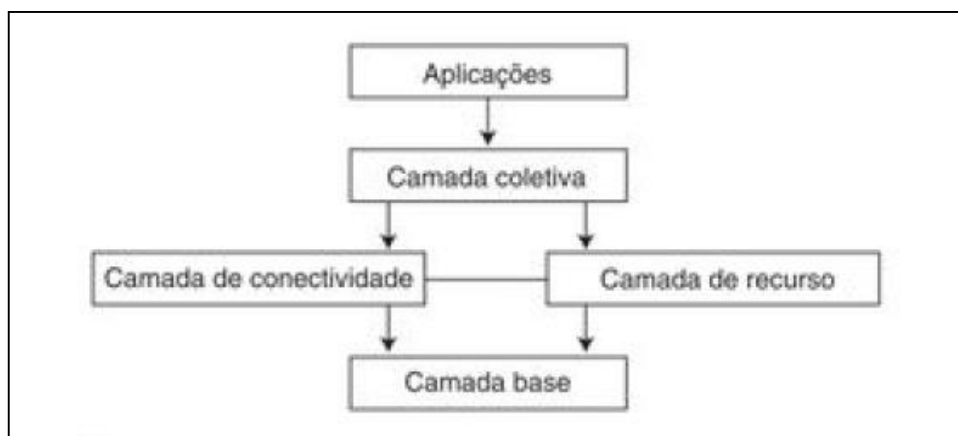


Figura 2. Arquitetura em camadas para sistemas de computação em grade

Fonte: TANENBAUM e STEEN (2007)

2.3.1.2 Sistemas de computação em grade

De acordo com Tanenbaun e Steen (2007) esse sistema distribuído é encontrado em organizações que enfrentaram problemas com o fluxo de aplicações em rede onde a interoperabilidade se mostrou um transtorno.

Segundo os autores esse sistema possui vários níveis diferentes onde ocorreu integração. Um servidor executa uma aplicação usando banco de dados e a disponibiliza para os programas remotos chamados clientes. Os clientes enviam solicitações ao servidor para executar uma operação e depois o servidor devolve a resposta ao cliente. Em níveis mais baixos clientes empacotam várias requisições em uma única requisição maior e as enviam como uma transação distribuída, com isso, todas as requisições são executadas.

De acordo com os autores, conforme as aplicações se tornaram sofisticadas e os componentes foram separados em bancos de dados e processamento, evidentemente deveria ocorrer uma integração para que as aplicações se comunicassem diretamente resultando na integração de aplicações empresariais.

2.3.3 Sistemas distribuídos pervasivos

De acordo com Tanenbaun e Steen (2007) com a introdução de dispositivos de computação móveis encontramos sistemas distribuídos onde a instabilidade é esperada,

esses sistemas são chamados de sistemas distribuídos pervasivos na qual um ponto de vista importante é a falta de controle administrativo humano.

Explica os autores que os dispositivos têm que estar sempre cientes que o seu ambiente mudara o tempo todo, uma dessas mudanças ocorre quando uma rede não está mais disponível, pois o usuário está se movimentando com isso a aplicação deve reagir e provavelmente conectar a outra rede ou tomar as devidas providências.

Para os autores, um fato importante de sistema pervasivos é que os dispositivos se unem ao sistema para acessar informações. A inconstância da conectividade e a frequente mudança dos dispositivos é muito provável que o local onde residam as informações acessíveis estará em constante mudança.

Os autores explicam que existem três tipos de sistemas distribuídos pervasivos:

a) Sistemas domésticos: montados ao redor de redes domésticas, ou seja, são compostos de um ou mais computadores pessoais, o mais importante é a integração de eletrônicos como aparelhos de TV, equipamentos de áudio e vídeo, *smartphones* etc.

b) Sistemas eletrônicos para tratamento de saúde: com o custo cada vez mais elevado do tratamento médico, estão sendo desenvolvidos dispositivos para monitorar a saúde de indivíduos e caso ocorra algo necessário o sistema entra em contato com médicos.

c) Redes de sensores: consiste em vários nós pequenos, cada um com dispositivo de sensoriamento acoplado. As redes de sensores em sua maioria utilizam comunicação sem fio e os nós são alimentados por baterias.

3.Sistema de arquivos distribuídos (SAD)

Os SADs são a base de diversas aplicações distribuídas. Esses sistemas garantem que os dados serão compartilhados de forma que possa ser acessível de qualquer computador conectado à rede de maneira segura e confiável.

3.1 Introdução

Segundo Coulouris, Dollimore e Kindberg (2007), os sistemas de arquivos foram originalmente desenvolvidos para sistemas de computadores centralizados ou *desktop*. Posteriormente, adquiriram atributos como controle de acesso e mecanismos de proteção de arquivos. Os SADs permitem o compartilhamento de informações através de recursos de *hardware* e *software*. Quando bem projetado, um sistema de arquivo dá acesso a arquivos armazenado em um servidor com desempenho e confiabilidade semelhantes aos arquivos armazenados em discos locais.

3.2 Modelos arquiteturais

A maioria dos sistemas são construídos com base na arquitetura cliente-servidor, porém, também existem de sistemas totalmente descentralizados.

Segundo Fernandes (2012), a arquitetura cliente-servidor ou centralizada, é a mais utilizada e menos complexa para implementação e administração comparada a

outras arquiteturas. Em certas ocasiões, o servidor também pode desempenhar papel de cliente em relação a outros servidores.

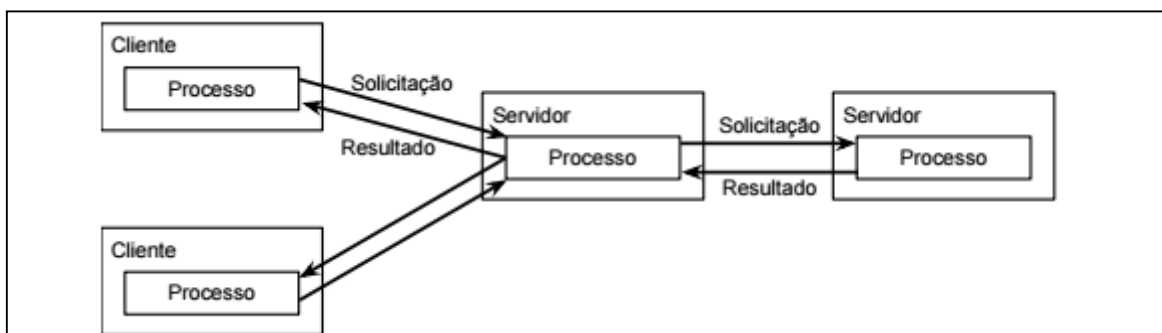


Figura 3. Arquitetura Cliente Servidor

Fonte: COULOURIS, DOLLIMORE E KINDBERG (2007). (apud FERNANDES, 2012)

De acordo com Coulouris, Dollimore e Kindberg (2007), no modelo descentralizado, o exemplo mais comum é o *peer-to-peer* (P2P). Nesse modelo, não existem diferenças entre os processos como ocorrem no cliente-servidor, sendo assim, todos os envolvidos possuem igual relevância.

Segundo Tanenbaun e Steen (2007), arquiteturas híbridas compreendem consiste na união da arquitetura centralizada com a arquitetura descentralizada. É mais usada para dar início a um serviço e em seguida a aplicação possa se estabelecer com outras partes distribuídas. Um bom exemplo dessa arquitetura são serviços de *torrents*.

3.3 Estudo de Casos

3.3.1 *Networking file system* (NFS)

Conforme Coulouris, Dollimore e Kindberg (2007), o *Networking File System* (NFS) foi projetado para atender as metas citadas no capítulo 2, em especial a transparência no acesso aos arquivos. Foi o primeiro SAD desenvolvido como produto final, inicialmente, utilizado apenas em ambiente *UNIX*. Depois de passar por várias adaptações, tornou-se compatível com uma gama maior de *hardware* e *software*.

Segundo os autores o NFS proporciona transparência de localização e de acesso se o serviço de montagem for usado certamente para produzir espaços de nomes semelhantes em todos os clientes. O NFS suporta *hardware* e sistemas operacionais diferentes. A implementação de um servidor NFS permite que clientes e servidores recuperem sua execução após uma falha sem a necessidade de qualquer procedimento de recuperação. A migração de arquivos, ou de sistemas de arquivos, deve ser feita manualmente para a reconfigurar as diretivas de montagem. Com o uso de *cache* para blocos de arquivos em cada computador cliente permite uma melhora de desempenho.

A seguir será apresentado os outros objetivos de projeto do NFS de acordo com Coulouris, Dollimore e Kindberg (2007).

a) Transparência de acesso: o módulo cliente do NFS oferece uma interface de programação de aplicativos para processos locais parecidos com a interface do sistema operacional local

b) Transparência de localização: cada cliente designa um espaço de nomes de arquivo, adicionando diretórios montados em sistemas de arquivos remotos em seu espaço de nomes local. O NFS não impõe um único espaço de nomes de arquivo em nível de rede cada cliente enxerga um agregado de sistemas de arquivos remotos de acordo com a montagem usada. Com isso os sistemas de arquivos podem ter diferentes nomes de caminhos em clientes diferentes.

c) Transparência de mobilidade: os sistemas de arquivos podem ser movidos entre os servidores porem as tabelas de montagem deve ser atualizada separadamente, por isso a transparência da migração não é obtida por inteiro pelo NFS.

d) Escalabilidade: os servidores NFS podem ter seu desempenho aumentando adicionando mais processadores, discos e controladores, no momento que o limite for atingido devem ser instalados servidores adicionais e o sistema de arquivos deve ser realocado. Isso ocorre, pois, vários arquivos são acessados com bastante frequência atingindo o limite do servidor.

e) Replicação de arquivos: Meios de armazenamento de arquivos somente leitura podem ser replicados em diversos servidores NFS, porem o NFS não permite a replicação de arquivos com atualizações para isso é necessário um serviço separado chamado de *Network Information Service* (NIS).

f) Heterogeneidade de *hardware* e de sistema operacional: o NFS é suportado por uma variedade de sistemas de arquivos e pode ser implementado em quase todos os sistemas operacionais e plataformas de *hardwares*.

g) Tolerância à falha: quando um servidor falha o serviço fornecido por ele é parado até a sua reinicialização, mas os processos clientes no nível do usuário continuam a partir do ponto em que o serviço foi interrompido, os clientes não têm o conhecimento da falha do servidor.

h) Consistência: ele apresenta uma forte afinidade com a semântica da cópia única e atende as exigências da maioria dos aplicativos, contudo não é aconselhado o uso de compartilhamento de arquivos por meio NFS para comunicação, ou forte coordenação entre processos em computadores diferentes.

i) Segurança: a necessidade de segurança no NFS apareceu com a conexão de várias *intranets* com a *internet* com isso a integração do *Kerberos* foi um grande passo. Outros desenvolvimentos incluem a opção de usar uma implementação de RPC-SEG para autenticação, privacidade e segurança dos dados. As instalações que não implantarem esses mecanismos acabam se tornando inseguras.

j) Eficiência: o desempenho medido de várias implementações NFS, e sua adesão para o uso em situações de cargas de trabalhos muito pesadas, são indícios claros da eficiência do NFS.

A figura 4 exemplifica a arquitetura NFS:

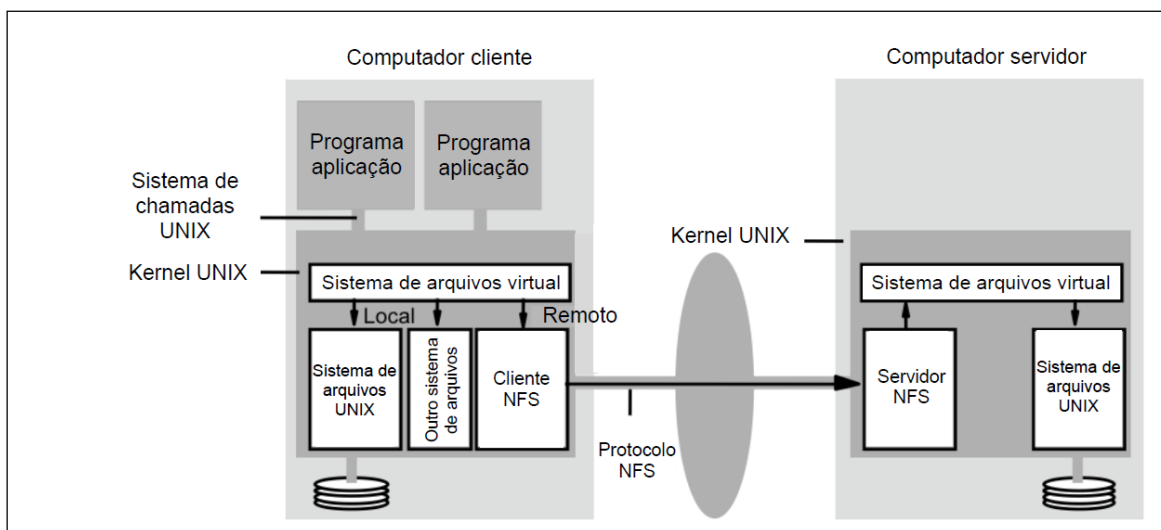


Figura 4. Arquitetura NFS

Fonte: COULOURIS, DOLLIMORE E KINDBERG (2007)

3.3.2 Andrew file system (AFS)

De acordo com Coulouris, Dollimore e Kindberg o AFS igual ao NFS fornece acesso transparente a arquivos remotos compartilhados para programas *UNIX* executado em estações de trabalho. O AFS é compatível com o NFS e difere em seu projeto e implantação, as diferenças estão relacionadas principalmente na identificação da escalabilidade. O AFS funciona melhor com um grande número de usuários ativos do que outros SADs. A principal Estratégia para obter a escalabilidade é o uso de *cache* que são capazes de armazenar arquivos nos nós clientes.

O AFS tem duas características incomuns como mostra os autores.

a) Servir arquivos inteiros: o conteúdo inteiro de diretórios e de arquivos é transmitido pelos servidores AFS para os computadores clientes.

b) *Cache* de arquivos inteiros quando uma cópia de arquivo é transferida para um computador cliente ela é armazenada em uma *cache* de disco local esse *cache* contém vários arquivos usados recentemente no computador. Quando possível são usadas copias locais dos arquivos para atender as requisições dos clientes.

Segundo os autores o AFS é implementado usando como base dois componentes de *softwares* que existem como processos *UNIX* chamados *Vice* e *Venus* a figura 5 mostra a distribuição desses processos. *Vice* é o *software* servidor executado em cada computador servidor e *Venus* é o processo executado em cada computador cliente. Os arquivos compartilhados são armazenados em servidores e suas cópias são armazenadas na *cache* dos discos locais das estações de trabalho. O espaço de nomes usa uma hierarquia de diretório *UNIX*, com uma sub-árvore específica com todos os arquivos compartilhados, isso dá certa perda de transparência, porém dificilmente em percebido para os usuários que não sejam os administradores.

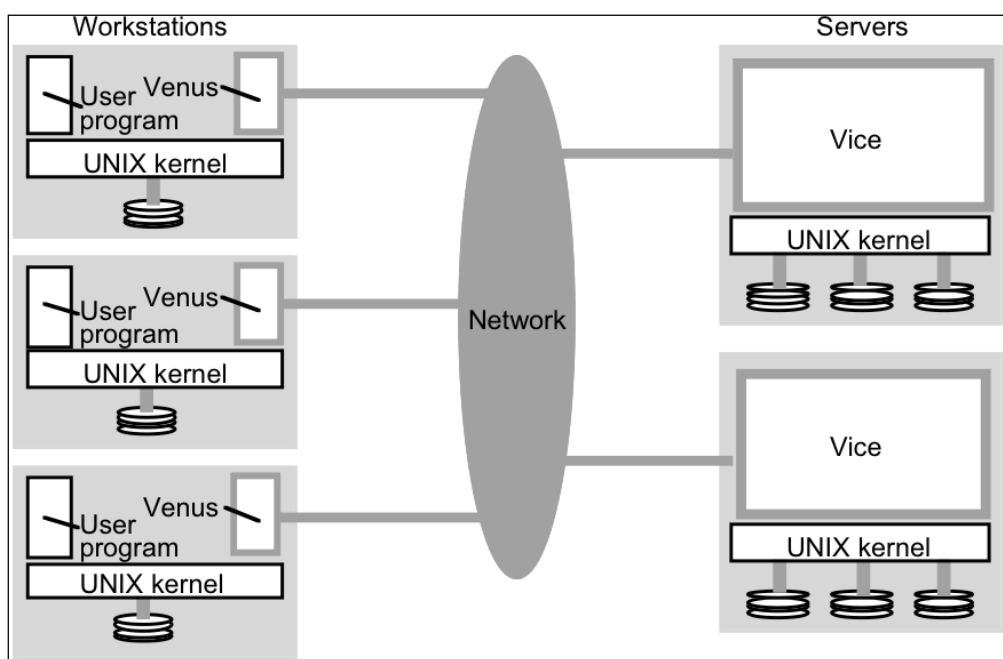


Figura 5. Arquitetura AFS

Fonte: COULOURIS, DOLLIMORE E KINDBERG (2007)

4. Computação em nuvem (*cloud computing*)

4.1 Definição de computação em nuvem

Segundo Veras (2012) computação em nuvem foi desenvolvido com a finalidade de oferecer serviços de baixo custo, fácil acesso e garantir características tais como disponibilidade e escalabilidade. Ela garante uma maneira eficaz de potencializar e flexibilizar os recursos computacionais, sendo assim um ambiente redundante que é capaz de continuar a funcionar corretamente com o mau funcionamento de um ou mais componentes, completa Taurion (2009). A Figura 6 mostra uma visão geral de uma nuvem computacional.

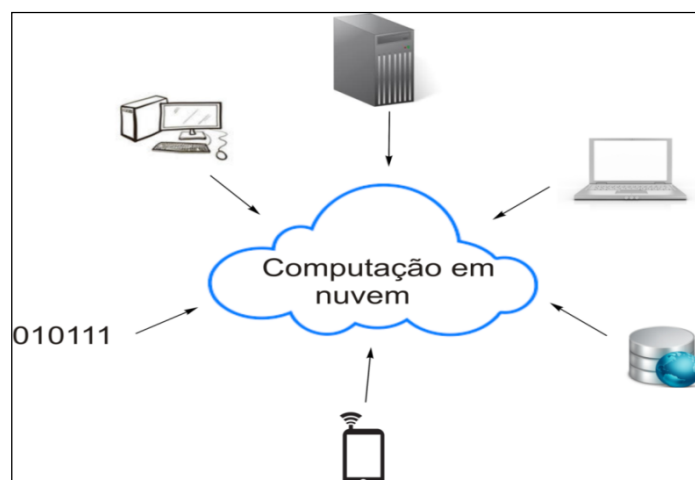


Figura 6. Visão geral de uma nuvem computacional

Fonte :Elaborado pelos autores (2015)

De acordo com Veras (2012) a Figura 7 ilustra a mudança do modelo, onde a capacidade do sistema acompanha a demanda, promovendo assim a otimização dos recursos. A Figura 7(A) ilustra a situação comum, onde a capacidade está sempre sobrando, e a Figura 7(B) ilustra a situação ideal.

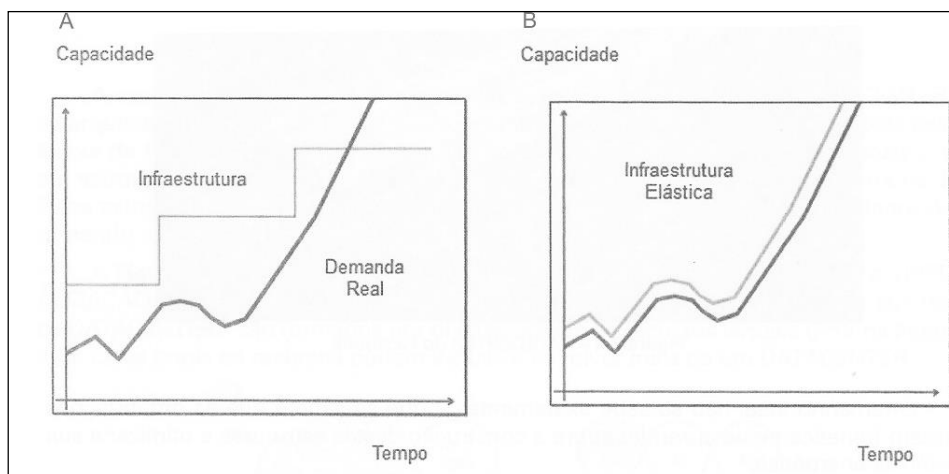


Figura 7. Mudança com Cloud Computing

Fonte: VERAS (2012)

4.2 Características essenciais

São as características essenciais que, quando agrupados, diferem a computação em nuvem dos demais sistemas.

4.2.1 Self-service sob demanda

Segundo Mell e Grance (2011) o usuário pode unilateralmente prever a capacidade computacional, como o desempenho do servidor ou seu armazenamento, sem a necessidade de interação humana.

4.2.2 Amplo acesso

De acordo com Mell e Grance (2011) recursos estão disponíveis através da rede e são acessados por meio de mecanismos padrão que promovem o uso de plataformas de cliente, seja ele moderado ou demasiado (por exemplo, celulares, laptops e estações de trabalho).

4.2.3 Pooling de recursos

Conforme Mell e Grance (2011) recursos de computação do provedor estão reunidos para servir a múltiplos usuários, com diferentes recursos físicos e virtuais atribuídos e realocados dinamicamente de acordo com a demanda. O usuário geralmente não tem controle ou conhecimento sobre a localização exata dos recursos disponibilizados.

4.2.4 Elasticidade rápida

Mell e Grance (2011) explica que recursos podem ser elasticamente provisionados e liberados, em certos casos, automaticamente, para escalar de maneira mais eficiente conforme a demanda.

4.2.5 Serviço medido

Para Mell e Grance (2011) sistemas em nuvem controlam e otimizam o uso dos recursos, como por exemplo a capacidade de processamento, utilização da memória, armazenamento e etc. O uso dos recursos pode ser monitorado, controlado e relatado, proporcionando transparência para o provedor e consumidor do serviço utilizado.

4.3 Modelos de Serviços

Segundo Veras (2012), existem três principais modelos de serviços para *Cloud Computing*, descritas a seguir.

4.3.1 Infraestrutura como um serviço (*Infrastructure as a Service - IaaS*)

De acordo com Veras (2012) o usuário não possui o controle na unidade física, contudo, através de mecanismos de virtualização, possui controle sobre armazenamento, máquinas virtuais, aplicativos instalados e um controle limitado dos recursos de rede. Um exemplo de IaaS é o serviço E2C, oferecido pela *Amazon*.

4.3.2 Plataforma como um serviço (*Platform as a Service - PaaS*)

Segundo Veras (2012) é aquela que do suporte à instalação e execução de aplicativos na nuvem. Exemplos de PaaS são a *AppEngine* do *Google* e o *Windows Azure* da *Microsoft*.

4.3.3 Software como um serviço (*Software as a Service - SaaS*)

Para Vera (2012) é viável para um grande número de clientes que passam a ser hospedados na nuvem. Os aplicativos são oferecidos como serviços. Todo o controle e gerenciamento da rede, sistemas operacionais, servidores e armazenamento é feito pelo provedor de serviço. Como exemplos de SaaS podemos citar o *Google Apps* e *SalesForce.com*.

4.3.4 Colocation

A empresa contrata o espaço físico e a infraestrutura dos *Racks*, mas o gerenciamento, monitoramento e suporte técnico são de responsabilidade da organização contratante.

4.3.5 Hosting

Usado em organizações que queiram maior investimento em hardware e software. Neste modelo, a organização tem a opção de contratar os serviços de infraestrutura do

DATACENTER (servidores, *storage*, unidades de *backup*, etc), além de contar com os profissionais do provedor de serviços para suporte.

Analisando a imagem citada anteriormente, fica claro que *Cloud Computing* é uma proposta muito mais sofisticada do que a simples oferta de serviços de *Hosting* e *Colocation*.

Segundo Veras (2012), a Figura 8 menciona os papéis desempenhados na arquitetura baseada em nuvem e destaca quem fornece serviço e quem consome. O modelo IaaS suporta o PaaS, que suporta o modelo SaaS.

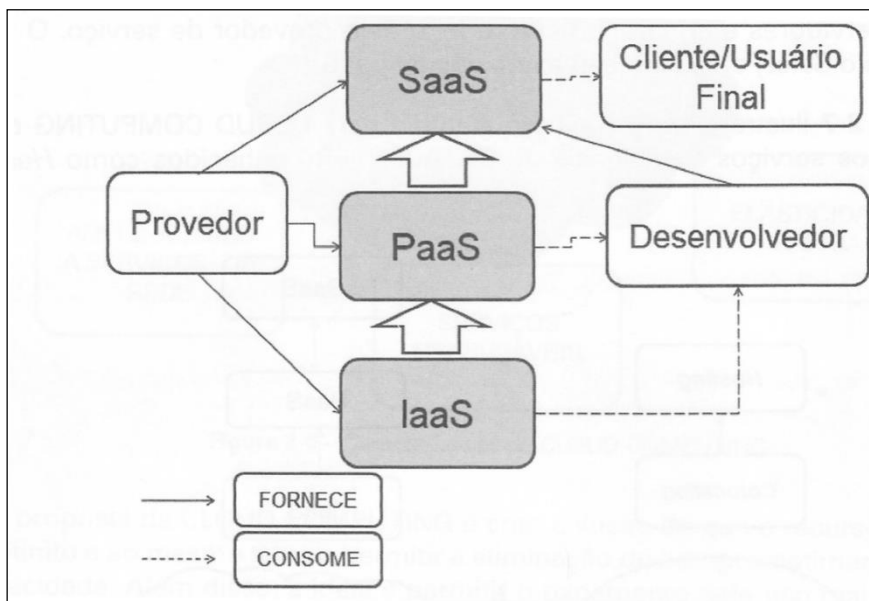


Figura 8. Papéis em Cloud Computing

Fonte: VERAS (2012)

De acordo com Veras (2012), o provedor de serviços considerado ideal é aquele responsável por disponibilizar e gerenciar toda estrutura de *Cloud Computing*, fazendo com que os usuários finais, não tenham preocupações com esse tipo de problema. Para isso o provedor fornece o serviço dentre uma das três modalidades. O desenvolvedor utiliza os recursos para gerar serviços para o usuário final, que por sua vez, paga a conta.

4.4 Modelos de implementação

Segundo Veras (2012), existem quatro principais modelos de implantação de *Cloud Computing*, descritos abaixo:

a) *Nuvem Privada (Private Cloud)*: é aquela que quase sempre é operada e gerenciada pela organização cliente. Os serviços são utilizados apenas pela organização, isto é, não estão publicamente disponíveis. Em alguns casos pode ser gerenciado por terceiros.

As nuvens privadas são basicamente subdivididas em:

- Nuvem privada, hospedada pela empresa. Aconselha-se este modelo quando os aspectos fortes em regulamentação e controle precisam ser considerados
- Nuvem privada, hospedada em provedor de serviço. Aconselha-se este modelo para aplicações de forma geral e aplicações de missão crítica.

b) Nuvem Pública (*Public Cloud*): é oferecida de modo pague-por-uso, é utilizada em organizações públicas ou por grandes corporações onde é necessária ampla capacidade de processamento e armazenamento.

c) Nuvem Comunitária (*Community Cloud*): esta nuvem é compartilhada por várias organizações com interesses em comum. A nuvem comunitária pode ser gerenciada tanto pelas organizações que fazem parte da comunidade ou por terceiros.

d) Nuvem Híbrida (*Hybrid Cloud*): é composta por duas ou mais nuvens, dentre os tipos citados anteriormente, que continuam a ser entidades únicas, porém, conectadas de uma nuvem computacional.

5. Materiais e métodos

A pesquisa feita foi do tipo laboratorial, onde foi criado um ambiente virtualizado com a ferramenta *Oracle VM VirtualBox*, foi instalado os sistemas operacionais *Windows 2008 Server R2* (três ambientes) e *Windows 7* (dois ambientes). Duas máquinas *Server* receberam o sistema *Distributed File System (DFS)* do *Windows* e a ferramenta *Active Directory*. A primeira é o servidor principal, onde foram armazenados todos os arquivos, a segunda é o servidor secundário, no qual replicará os arquivos da principal. Uma máquina *Windows 7* (cliente), para fazer testes em ambos servidores.

Em seguida, foi instalado na máquina hospede do *OwnCloud* as ferramentas *WebServer (IIS)* e o *Web Platform* que serve de suporte para criação de um servidor *Apache 2*, *Hypertext Preprocessor (PHP)* e banco de dados *MYSQL*. Após foi instalado a ferramenta *client* do *OwnCloud* na máquina cliente.

5.1 Resultados obtidos

5.1.1 Comparativo entre o sistema em nuvem e sistema local

Existem inúmeras possibilidades de levar uma estrutura ou serviço à nuvem além da ferramenta apresentada no trabalho. Por isso não se deve usar como base apenas as tecnologias que foram estudadas e pesquisadas, por isso, serão apresentadas um comparativo geral e em seguida um específico com base nas ferramentas estudadas.

5.1.2 Comparativo entre sistema e nuvem e sistema local de forma geral

5.1.2.1 Pontos positivos

O sistema em nuvem possui maior flexibilidade e escalabilidade em relação ao tradicional (físico). Traz maior economia de energia, visto que os serviços estarão

hospedados em servidores fora da empresa. O custo com colaboradores também diminui, já que a infraestrutura estará hospedada em um fornecedor especializado que na maioria das vezes fica responsável pelo suporte ao mesmo. Suas atualizações são automáticas, de forma geral, as mais pesadas são semanais e existe uma automanutenção diária. O *backup*, quando feito em nuvem dispensa disco ou qualquer outra forma de armazenamento físico por parte da empresa.

Quando a demanda de serviço é menor, o serviço em nuvem é uma excelente opção, já que na maioria das vezes, alocar ou comprar um servidor acaba ficando mais caro, pois nem sempre encontra-se um servidor com configurações que são compatíveis com a necessidade do cliente, por tanto, o cliente acaba não fazendo uso total do desempenho da máquina ou o desempenho de algum dispositivo da mesma não supri sua necessidade. Por exemplo, o cliente adquire um servidor muito eficiente em questão de processamento, porém deixa a desejar no quesito armazenamento ou o servidor possui alto desempenho em todos os pontos, porém o cliente necessita apenas do espaço em disco mais robusto.

5.1.2.2 Pontos negativos

Uma questão de grande importância para o cliente é a segurança, e uma questão que sempre se levanta é o fato de que as informações do contratante estão armazenadas em um local desconhecido onde a segurança do mesmo não pode ser totalmente gerenciada pela cliente, em certos casos.

A velocidade da *internet* é outro fator importante, já que o acesso sempre será feito de forma externa, gerando assim quantidade de tráfego de dados extremamente alta, que por consequência gera um custo excessivo para o contratante da *cloud computing*, pois a *internet* no Brasil além de não ter a mesma qualidade de países desenvolvidos possui um preço exorbitante.

5.1.3 Comparativos entre os sistemas utilizados

Para estudo foi utilizada o *Distributed File System* da *Microsoft* (sistema local) e o *Owncloud* (sistema em nuvem).

Os tópicos a seguir abordaram as principais vantagens e características de cada sistema, deixando assim de lado a abordagem de informações básicas.

5.1.3.1 Implementação

a) *Owncloud*: é uma ferramenta *open source* que trabalha em conjunto com outros serviços, tais como o *Apache*, banco de dados *MySQL*, entre outras. Também existe a possibilidade de usá-la em conjunto com ferramentas e serviços pagos. Sua estrutura de compartilhamento de arquivos é similar ao *Google drive* da *Google*. Pode-se implementá-lo tanto em ambiente *Windows* quanto *Linux*, um *client* deve ser instalado e configurado na máquina cliente para poder efetuar o acesso a seu “espaço na nuvem”. De modo geral a implementação do *Owncloud* é mais complicada que o sistema de arquivos distribuídos da *Microsoft*.

O usuário tem uma gama de aplicativos, semelhante a um *Google Play* ou *Apple Store*, porém com uma gama muito menor de aplicativos. A Figura 9 mostra a página onde podemos encontrar os aplicativos.

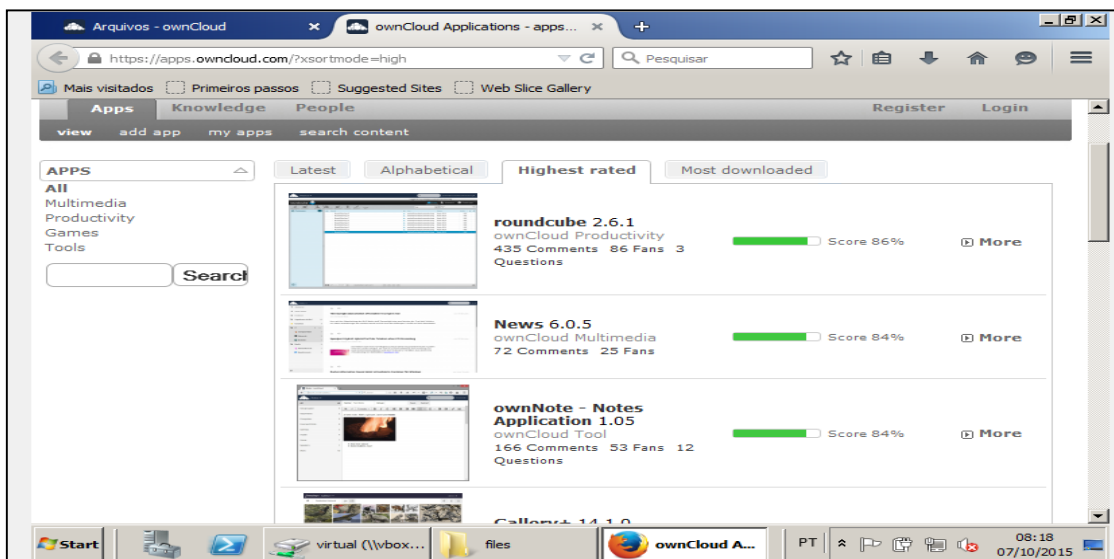


Figura 9. Gerenciamento de arquivos Owncloud

Fonte: Sistema Owncloud (2015)

b) DFS: sua instalação é simples, até mesmo uma pessoa leiga consegue fazer a instalação seguindo um tutorial. A ferramenta faz parte do *Windows Server*, sendo assim, o único gasto será com a compra de uma licença do *Windows*.

5.1.3.2 Segurança em rede

a) *Owncloud*: a segurança do *Owncloud* não depende apenas do sistema em si, já que trabalha em conjunto com outras ferramentas. Sendo assim, se a versão do *apache*, por exemplo, conter falhas de segurança, o *Owncloud* estará automaticamente suscetível aos possíveis problemas que tal falha pode gerar, assim como as informações contidas no servidor.

b) DFS: o *Server* vem apenas com seu firewall como ferramenta nativa, visto que na maioria das vezes é implementada na rede servidores *proxy*, *firewall*, entre outras ferramentas de segurança.

5.1.3.3 Acessibilidade (compartilhamento de arquivos)

a) *Owncloud*: pode ser acessado de qualquer lugar, inclusive *smartphones* ou *tablets* basta ter o *client* instalado e com acesso à internet, outra possibilidade de compartilhamento é via *Uniform Resource Locator* (url), ou seja, um cliente pode apenas mandar a url gerada para outro cliente e o mesmo pode fazer o *download* do arquivo, se por algum motivo a conexão cair os arquivos modificados serão sincronizados com o servidor quando a conexão voltar a sua normalidade.

A Figura 10 mostra algumas opções disponíveis para o administrador quanto aos arquivos de usuários.

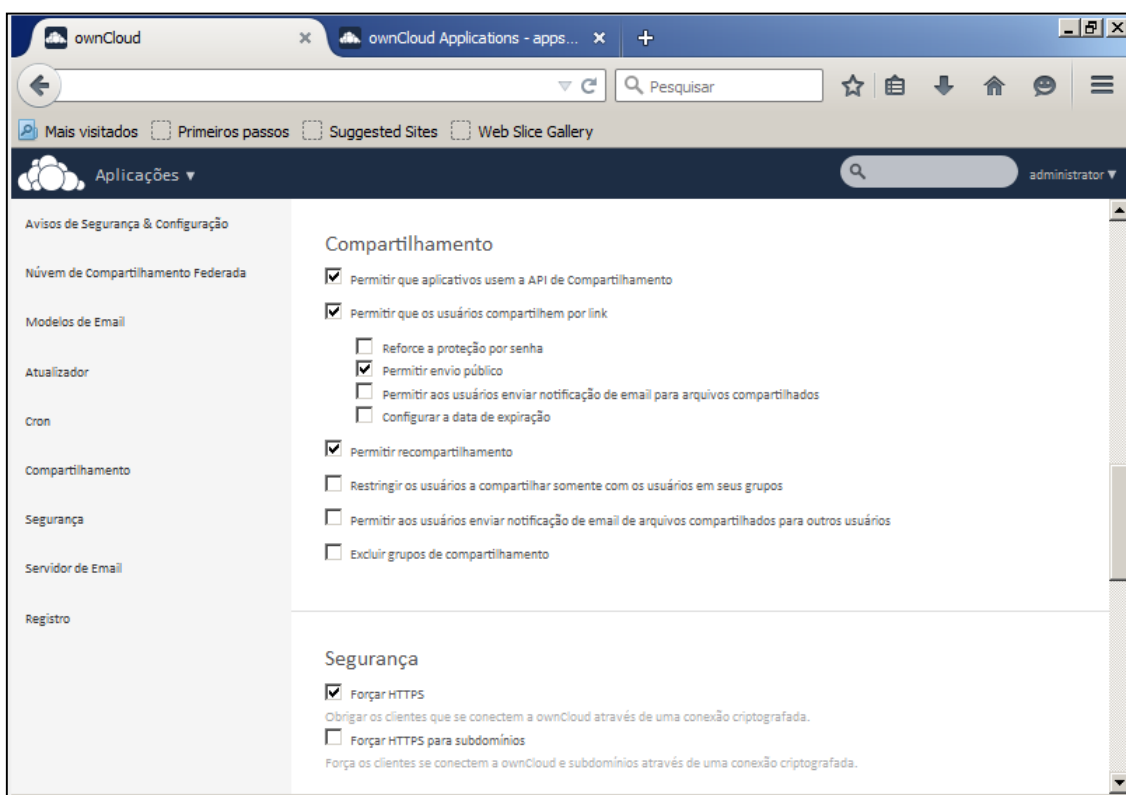


Figura 10. Gerenciamento de arquivos Owncloud

Fonte: Elaborado pelos autores (2015)

b) DFS: traz a possibilidade de replicação de arquivos entre dois servidores, se por algum motivo o servidor principal ficar indisponível o secundário assumirá e assim o usuário passará a utilizar o arquivo replicado, quando o servidor principal voltar a sua normalidade os arquivos modificados do servidor secundário serão replicados ao servidor primário, esse método de replicação ocorre sem o usuário note. Porém para o DFS ser acessado fora da empresa é necessário a utilização de outra tecnologia, por exemplo, uma Rede Virtual Privada (VPN) para conseguir o acesso por meio externo, além do acesso a certos dispositivos como *smartphones* não ser possível, mesmo em meio interno.

5.1.3.4 Gerenciamento

a) *Owncloud*: ambos os sistemas trabalham com base nos seus usuário e criação de grupos gerenciáveis. O sistema conta com ferramentas nativas de gerenciamento, como por exemplo, a emissão de alertar via *e-mail* quando qualquer arquivo ou diretório sofre alguma alteração (exclusão, criação, entre outros). Outro ponto positivo é a possibilidade de os usuários utilizar *smartphone* ou *tablet* para o gerenciamento de arquivos ou pastas, de maneira prática e efetiva.

b) DFS: para ter um melhor gerenciamento é necessário instalar outras ferramentas para auxiliá-lo como, por exemplo, para um controle de usuários e senhas assim como criação de cotas é necessário à utilização do *Active Directory*.

6. Conclusão

Com base nos autores e nas ferramentas estudadas, pode-se concluir que os serviços e servidores em nuvem estão crescendo constantemente graças a seu sistema flexível e de alta disponibilidade. Uma barreira que precisa ser vencida é a da segurança, já que as informações ou serviços específicos é a matéria prima de muitas empresas, o fato de estarem sob a responsabilidade de terceiros ainda não é uma situação aceitável para muitos, e por causa desse e outros fatores, acabam optando pela estrutural tradicional, isto é, estrutura física.

Disponibilidade, flexibilidade e escalabilidade são diferenciais que vem sendo cada vez mais requisitados pelo cliente na hora de implementar um serviço. Tais conceitos são à base da computação em nuvem e é isso que a faz tão superior quando comparada as estruturas físicas.

Ambas ferramentas utilizadas na pesquisa, não foram estudadas em sua integra, por tanto com base nessa pesquisa, podem haver trabalhos futuros, uma sugestão, é utilizar o serviço *Owncloud* em ambiente *Linux*, visto que no trabalho foi utilizado em ambiente *Windows* e o mesmo foi projetado inicialmente para funcionar apenas em ambiente *Linux*, outra sugestão é o estudo do gerenciamento *Owncloud* através da plataforma *mobile*, dentre outros pontos que não foram aprofundados neste trabalho.

7. Referências

- COULORIS, G., DOLLIMORE, J., KINDBERG, T. (2007). *Sistemas Distribuídos: conceitos e projetos*. 4. ed. Porto Alegre: Bookman.
- FERNANDES, S. E. N. (2012); *Sistema de Arquivos Distribuído Flexível e Adaptável*. 72 f. Dissertação (Mestrado em Ciência da Computação) - UNESP. Disponível em: <<http://www.ibilce.unesp.br/Home/Pos-Graduacao475/CienciadaComputacao/dissertacao-mestrado.pdf>>. Acesso em: 29 abr.2015.
- MELL, P.; GRANCE, T. (2011). *The NIST Definition of Cloud Computing: Recommendations of the National Institute of Standards and Technology*. Gaithersburg: NIST Special Publication 800-145.
- TANENABUM, A. S., STEEN, M. (2007). *Sistemas Distribuídos: princípios e paradigmas*. 2 ed. São Paulo: Pearson Education.
- TAURION, C. (2009). *Cloud Computing: computação em nuvem: transformando o mundo da tecnologia da informação*. Rio de Janeiro: Brasport.
- VERAS, M. (2012). *Cloud Computing: Nova Arquitetura da TI*. 1.ed. Rio de Janeiro: Brasport.