

**UNIVERSIDADE DO ESTADO DE SANTA CATARINA - UDESC  
CENTRO DE CIÊNCIAS TECNOLÓGICAS - CCT  
MESTRADO EM COMPUTAÇÃO APLICADA - MCA**

**GIL ANDRIANI**

**SINCRONIZAÇÃO DE ARQUIVOS ENTRE NUVENS DE  
ARMAZENAMENTO E REPOSITÓRIOS GEOGRAFICAMENTE  
DISTRIBUÍDOS**

**JOINVILLE  
2016**



**UNIVERSIDADE DO ESTADO DE SANTA CATARINA - UDESC  
CENTRO DE CIÊNCIAS TECNOLÓGICAS - CCT  
MESTRADO EM COMPUTAÇÃO APLICADA - MCA**

**GIL ANDRIANI**

**SINCRONIZAÇÃO DE ARQUIVOS ENTRE NUVENS DE  
ARMAZENAMENTO E REPOSITÓRIOS GEOGRAFICAMENTE  
DISTRIBUÍDOS**

Dissertação submetida ao Programa  
de Pós-Graduação em Computação  
Aplicada do Centro de Ciências Tec-  
nológicas da Universidade do Es-  
tado de Santa Catarina, para a ob-  
tenção do grau de Mestre em Com-  
putação Aplicada.

**Orientador:**

Dr. Maurício Aronne Pillon

**JOINVILLE**

**2016**

A573s Andriani, Gil

Sincronização de arquivos entre nuvens de armazenamento e repositórios  
geograficamente distribuídos/

Gil Andriani. – 2016.

125 p. : il. ; 21 cm

Orientador: Maurício Aronne Pillon

Bibliografia: 119-123 p.

Dissertação (mestrado) – Universidade do Estado de Santa Catarina, Centro de  
Ciências Tecnológicas,

Programa de Pós-Graduação em Computação Aplicada, Joinville, 2016.

1. Armazenamento de dados. 2. Sincronização. 3. Memória virtual(computação).  
I. Pillon, Maurício Aronne. II. Universidade do Estado de Santa Catarina. Programa  
de Pós-Graduação em  
Computação Aplicada. III. Título.

CDD: 004.5 - 23. ed.

**Gil Andriani**

**Sincronização de Arquivos entre Nunvens de  
Armazenamento e Repositórios Geograficamente  
Distribuídos**

Dissertação apresentada ao Curso de Mestrado Acadêmico em Computação Aplicada como requisito parcial para obtenção do título de Mestre em Computação Aplicada na área de concentração “Ciência da Computação”.

**Banca Examinadora**

Orientador:

Prof. Dr. Mauricio Aronne Pillon  
CCT/UDESC

**Membros**

Prof. Dr. Rafael Rodrigues Obelheiro  
CCT/UDESC

Prof. Dr. Rodrigo da Rosa Righi  
UNISINOS

Joinville, SC, 23 de setembro de 2016.

*Dedico esse estudo ao meu avô, Marco Aurélio de Oliveira, que com sua  
inesgotável dialética esteve presente em todos os momentos dessa  
trajetória e continua a inspirar a todos que o conheceram.*

## **AGRADECIMENTOS**

Gostaria de agradecer a Márcia de Oliveira Andriani, minha mãe, que não poupou esforços nas infindáveis correções de meus escritos. Para uma professora de português, mestre na educação de adultos, virar uma especialista em nuvem computacional é algo que me surpreende. Agradeço Márcio Mesquita por me desafiar a voltar estudar, Vicente Secchin e Max Polastri por fornecerem prontamente todos os recursos necessários durante o andamento dos estudos. Rafael Rodrigues Obelheiro e Guilherme Piegas Koslovski por participarem com contribuições e críticas ao trabalho. Por fim e não menos importante, agradeço à paciência de meu orientador Maurício Aronne Pillon que tornou esta caminhada a mais produtiva possível e contribuiu muito para realização desse objetivo.



*"Se vi mais longe foi por estar de pé sobre ombros de gigantes."*  
*Isaac Newton*



## RESUMO

Provedores de nuvens computacionais difundiram sistemas e soluções para armazenamento dinâmico de arquivos, entregues aos usuários finais como serviços acessíveis sob-demanda. Especificamente, o armazenamento de arquivos alcançou popularidade em ambientes domésticos através da difusão de clientes para sincronização com servidores remotos. Tais aplicativos introduziram um acesso facilitado aos arquivos pessoais, independente do dispositivo utilizado. Entretanto, a utilização das ferramentas de sincronização, e consequentemente do armazenamento em nuvem, enfrenta dificuldades em organizações empresariais. Uma barreira existente é relacionada com os requisitos de acesso e as expectativas de utilização, combinada com a existência de sistemas legados de armazenamento. Por exemplo, é comum a concentração de usuários em um mesmo local, enquanto diversas organizações possuem sistemas legados de armazenamento interconectados por enlaces privados de comunicação. Em suma, o cenário de execução e os requisitos das organizações não são atendidos por aplicativos populares para sincronização de arquivos entre provedores de nuvem e dispositivos locais. Assim, este trabalho apresenta CHSAN, uma arquitetura para sincronização de arquivos entre armazenamento na nuvem e repositórios geograficamente distribuídos. CHSAN é uma arquitetura agnóstica em relação ao provedor de nuvem, número de usuários na organização e sistema legado de armazenamento, considerando em seu gerenciamento a existência de canais privados de comunicação, bem como a transferência de arquivos pela Internet. Os resultados qualitativos e quantitativos obtidos com um protótipo operacional de CHSAN indicam uma aplicação promissora em ambientes com usuários colaborativos geograficamente distribuídos.

**Palavras-chaves:** Armazenamento em Nuvem. Sincronização de Arquivos. CHSAN.



## ABSTRACT

Cloud Computing has spread the dynamic storage provisioning, delivered to end users as on-demand services. Indeed, file storage and sharing achieved popularity among domestic users through the use of synchronization tools. However, the effective implementation of cloud storage on organizations faces a set of obstacles related to access requirements, performance expectations and usage characteristics. Usually, the usage profile of organizations comprehends an intensive file sharing between local-placed users, sustained by legacy storage tools. Moreover, data transfer is commonly performed atop dedicated and private communication links. Although file storage and share achieved popularity among domestic users, existing synchronization tools lacks on organizations requirements. In this context, this dissertation introduces CHSAN, an architecture for synchronizing files between cloud storage and repositories geographically distributed. The qualitative and quantitative results obtained from an operational prototype of CHSAN indicate a promising application in collaborative environments with geographically distributed users.

**Key-words:** Cloud Storage, Cloud Sync, CHSAN.



## LISTA DE FIGURAS

Figura 2.1 –A aplicação dos modelos de serviço ao Armazenamento em Nuvem. Figura extraída de (HÖFER; KARAGIAN-NIS, 2011) e (ZHANG, 2010) e adaptada pelo autor. . . . .	32
Figura 2.2 –Interação do servidor de arquivos tradicional e os modelos de Serviço de nuvem. . . . .	35
Figura 2.3 –Deduplicação e versionamento em funcionamento nos sistemas distribuídos, inspirado na patente do <i>DropBox</i> (HOUSTON; FERDOWSI, 2014) . . . . .	37
Figura 2.4 –Ambiente organizacional com múltiplos usuários colaborativos. . . . .	40
Figura 3.1 –Arquitetura do serviço e seus componentes . . . . .	51
Figura 3.2 –Arquitetura comum do cliente de sincronização de arquivos, com fluxo de envio e recebimento para a nuvem	55
Figura 3.3 –Arquitetura do cliente <i>DropBox</i> extraído de (HOUSTON; FERDOWSI, 2014) e adaptada pelo autor . . . . .	66
Figura 3.4 –Fluxo de escrita ou leitura do <i>DropBox</i> Fonte: (HOUSTON; FERDOWSI, 2014) e Adaptado pelo autor. . . . .	68
Figura 3.5 –Arquitetura do cliente Google Drive. Fonte: (BESSEN, 2015) Adaptada pelo autor. . . . .	70
Figura 3.6 –Fluxo de escrita ou leitura <i>Google Drive</i> fonte: (BESSEN, 2015) e adaptado pelo autor . . . . .	72
Figura 4.1 –Arquitetura do cliente híbrido de sincronização de arquivos CHSAN. . . . .	80
Figura 4.2 –Visão detalhada do módulo de armazenamento CHSAN.	82
Figura 4.3 –Exemplos de fluxo de leitura e escrita usando CHSAN. .	88
Figura 5.1 –Operações em arquivos localizados no armazenamento externo (CHSAN vs. <i>OneDrive</i> ) . . . . .	92

Figura 5.2 –Arquitetura CHSAN com hachura para identificação das áreas de teste . . . . .	95
Figura 5.3 –Ambientes de testes . . . . .	98
Figura 5.4 –Tempo de execução (em segundos) dos sistemas analisados. . . . .	103
Figura 5.5 –Operações em arquivos localizados no armazenamento externo (CHSAN vs. <i>OneDrive</i> ). . . . .	105

## LISTA DE TABELAS

Tabela 3.1 –Comparação das características da camada de aplicação, do serviço de armazenamento, acesso a arquivos em nuvem . . . . .	64
Tabela 3.2 –Comparação de características dos principais serviços de sincronização de arquivos. . . . .	74
Tabela 4.1 –Lista de operações e abrangência. . . . .	85
Tabela 5.1 –Comparação das características dos clientes de sincronização de acesso a arquivos em nuvem e CHSAN. . .	93
Tabela 5.2 –Vazão alcançada pelos sistemas analisados (em MB/s). 102	



## LISTA DE ABREVIATURAS E SIGLAS

IaaS	Infrastructure as a Service
PaaS	Platform as a Service
SaaS	Software as a Service
NFS	Network File System
CIFS	Common Internet Filesystem
SMB	Server Message Block
CHSAN	Cliente Híbrido para Sincronização de Arquivos em Nuvem
DAS	Direct Attached Storage
SAN	Storage Area Network
NAS	Network Area Storage
FUSE	Filesystem in Userspace
VFS	Virtual Filesystem
SLA	Service Level Agreement
API	Application Program Interface
REST	Representational State Transfer
LAN	Local Area Network



## LISTA DE SÍMBOLOS

$\alpha$	Letra grega alfa
$v$	Volume
$n$	Número de repetições
$\leq$	Menor ou igual
$b_e$	Bytes escritos
$=$	Igual



# SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>23</b>
1.1	Objetivo do Trabalho	26
1.2	Principais Contribuições	26
1.3	Organização do Texto	27
<b>2</b>	<b>Fundamentação Teórica</b>	<b>29</b>
2.1	Computação na Nuvem	29
2.2	Classificação do Armazenamento na Nuvem	30
2.2.1	Conceitos e Definições	34
2.3	Computação nas Organizações	39
2.4	Armazenamento nas Organizações	40
2.4.1	Modelos de Serviço Focados nas Organizações	42
2.5	Trabalhos Relacionados	45
2.6	Considerações Parciais	47
<b>3</b>	<b>Serviço de Acesso a Arquivos em Nuvem</b>	<b>49</b>
3.1	Usuários Domésticos versus Usuários de Organizações	49
3.2	Arquitetura do Serviço de Armazenamento na Nuvem	50
3.2.1	Armazenamento Distribuído e Gerenciamento	51
3.2.2	Camada de Aplicação	53
3.2.3	Comparação dos Serviços da Camada de Aplicação	62
3.3	Provedores do Serviço de Acesso a Arquivos em Nuvem	65
3.3.1	DropBox	65
3.3.2	Google Drive	70
3.3.3	OneDrive	73
3.4	Comparação de Características e Funcionalidades	73
3.5	Considerações Parciais	76
<b>4</b>	<b>Cliente Híbrido para Sincronização de Arquivos na Nuvem</b>	<b>77</b>
4.1	Cenário de Execução	77

4.2	Arquitetura do CHSAN . . . . .	80
4.2.1	Armazenamento . . . . .	81
4.2.2	Módulos Observador e Transporte . . . . .	83
4.2.3	Repositório de Metadados . . . . .	83
4.2.4	Operações . . . . .	84
4.3	Considerações Parciais . . . . .	89
<b>5</b>	<b>Experimentação e Análise dos Resultados . . . . .</b>	<b>91</b>
5.1	Análise Qualitativa da Arquitetura CHSAN . . . . .	91
5.2	Avaliação Experimental da Arquitetura CHSAN . . . . .	94
5.2.1	Descrição do Ambiente de Testes . . . . .	94
5.2.2	Ferramentas de Monitoração e Carga de Trabalho .	98
5.2.3	Experimentação e Resultados . . . . .	99
5.2.4	Análise dos Resultados Experimentais . . . . .	105
5.3	Análise da Escalabilidade da Arquitetura CHSAN . . . . .	108
5.4	Considerações Parciais . . . . .	110
<b>6</b>	<b>Conclusão e Trabalhos Futuros . . . . .</b>	<b>113</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS . . . . .</b>		<b>117</b>

## 1 INTRODUÇÃO

As nuvens computacionais revolucionaram a utilização de recursos de processamento, comunicação e armazenamento, entregando essas funcionalidades aos usuários como serviços sob demanda (ZHANG, 2010). A adoção de armazenamento na nuvem é uma realidade para usuários domésticos: o armazenamento, a edição e o compartilhamento de arquivos foram popularizados pela difusão de clientes para sincronização (e.g., *DropBox*<sup>1</sup>, *Google Drive*<sup>2</sup> e *OneDrive*<sup>3</sup>) de dispositivos locais com repositórios hospedados em nuvens computacionais.

O armazenamento na nuvem, segundo NIST (Instituto Nacional dos Estados Unidos de Padrões e Tecnologia) (MELL; GRANCE, 2009), apresenta características essenciais que otimizam o capital, equipamentos e pessoas, fornecendo aos usuários maior disponibilidade aos dados (DRAGO, 2012) em três modelos de serviço: IaaS - *Infrastructure as a service*; PaaS - *Plataform as a Service*; e SaaS - *Software as a Service*. O foco deste trabalho está no armazenamento na nuvem com o modelo de serviço SaaS. O serviço de acesso a arquivos na nuvem é, basicamente, constituído por: provedores, que hospedam e fornecem serviços de acesso remoto aos dados, e clientes, que manipulam estes dados localmente e interagem com provedores sincronizando dados locais e remotos.

A caracterização do perfil de usuários de clientes de sincronização de arquivos na nuvem é objeto de pesquisa recente. Gonçalves e seus colegas (GONÇALVES, 2014) identificaram o perfil desses usuários em campus universitário, apontando o tamanho médio dos repositórios como 4,23 GB, sendo que a maioria dos usuários tende a armazenar

<sup>1</sup> DropBox, disponível em <<https://www.dropbox.com/>>.

<sup>2</sup> Google Drive, disponível em <<https://www.google.com/drive/>>.

<sup>3</sup> Microsoft Onedrive, disponível em <<https://onedrive.live.com/>>.

muitos arquivos (mais de 1.000 arquivos), incluindo usuários com mais de 20.000 arquivos armazenados. Referente à dinâmica de atualização, foi observado que cerca de 82% das operações de atualização carregam até 1 MB, caracterizando uma concentração de uso sobre arquivos pequenos. Ainda sobre a dinâmica os autores identificaram que quanto maior o número de usuários que compartilham, maior é o tráfego gerado por cada cliente. Este perfil foi traçado através da monitoração do tráfego gerado pelos clientes de sincronização de arquivos mais populares, que foram concebidos com foco em usuários domésticos (GONÇALVES, 2016). O compartilhamento de arquivos entre grupos com grande número de usuários não é uma realidade para cenários domésticos e acadêmicos.

Embora populares, os clientes de sincronização de arquivos na nuvem não atendem aos requisitos específicos de organizações com múltiplos usuários colaborativos, geograficamente distribuídos, concentrados em redes privadas e sistemas legados de armazenamento (ANDRIANI, 2016; GONÇALVES, 2014). Organizações são subdivididas em grupos de usuários colaborativos dinâmicos e temporários, formados espontaneamente segundo interesses profissionais. Quanto a localização geográfica, estes grupos de usuários colaborativos tendem a estar concentrados em redes privadas e geograficamente distribuídas. Em geral, grupos de usuários colaborativos possuem um maior conjunto de arquivos compartilhados quando comparados com usuários domésticos. Consequentemente, a sincronização gera maior tráfego em cada cliente. Embora estes usuários estejam, na sua maioria, em redes privadas, os clientes de sincronização populares ignoram a existência de repositórios locais integrando, a cada operação, diretamente com o provedor na nuvem.

Armazenamento distribuído não é uma novidade nas organizações. Soluções aplicáveis em redes privadas, tais como, *Network File System (NFS)*, *Common Internet File System (CIFS)* ou *Server Message Block (SMB)* são comumente aplicadas para o gerenciamento interno em

cenários com múltiplos usuários. A existência destes recursos é fator limitante para a migração de soluções para a nuvem, sendo parcialmente responsável pelo fato de 45% das organizações não terem previsão para migração de qualquer serviço para nuvens computacionais (SMITH, 2013). A latência no acesso aos arquivos é um ponto crítico para múltiplos usuários colaborativos, tornando, neste quesito, os clientes de sincronização na nuvem menos interativos que as soluções atualmente implantadas.

Os clientes de sincronização mantêm o conteúdo do usuário sincronizado com o armazenamento na nuvem, realizando cópias do conteúdo compartilhado a cada acesso para cada dispositivo. Este não é um fator limitante para os usuários domésticos, pois usualmente a capacidade de seu espaço na nuvem é semelhante (ou inferior) ao espaço em seus dispositivos pessoais. No entanto, o conteúdo compartilhado entre equipes de uma organização facilmente supera o espaço individual das estações de trabalho. Os repositórios de dados disponíveis nessas organizações, normalmente, possuem capacidade de armazenamento superior à capacidade do disco local das estações de trabalho (DOUCEUR; BOLOSKY, 1999). No caso do armazenamento em nuvem, o acesso à totalidade dos dados fica restrito a disponibilidade de armazenamento do dispositivo com menor capacidade. Assim, a colaboração entre os usuários do grupo depende dos arquivos selecionados para sincronização, atualmente sob responsabilidade dos próprios usuários. Quando colaborando, os usuários devem reconfigurar os repositórios locais para fazer a sincronização seletiva.

Ainda, o amplo acesso aos arquivos através da Internet e a concentração de profissionais em um mesmo local são fatores conflitantes. Por um lado, a disponibilidade de acesso aos dados de qualquer lugar permite maior interação e colaboração entre as equipes, por outro lado, a concentração de profissionais em um mesmo local acessando o mesmo volume de dados na nuvem pode sobrecarregar os pontos de acesso à

Internet das organizações, além de aumentar o tempo de acesso aos arquivos.

## 1.1 Objetivo do Trabalho

Este trabalho está focado em organizações com múltiplos usuários colaborativos, geograficamente distribuídos, concentrados em redes privadas e com soluções de armazenamento distribuído implantadas, usualmente interconectadas por canais dedicados de comunicação. Assim, o objetivo principal é o desenvolvimento de uma arquitetura que permita a sincronização de arquivos entre nuvens de armazenamento e repositórios locais, geograficamente distribuídos.

## 1.2 Principais Contribuições

As principais contribuições são: categorização dos serviços de acesso a nuvem vinculados aos modelos de serviços (IaaS, PaaS e SaaS); definição de uma arquitetura comum do serviço de armazenamento na nuvem; e concepção de uma arquitetura híbrida para sincronização de arquivos na nuvem – CHSAN – focada nas necessidades da organização.

Em nuvens computacionais, usuários estabelecem contratos com fornecedores especificando os requisitos que desejam: tipo de serviço, a forma de acesso aos dados; e pacotes de recursos mais adequados as suas necessidades naquele momento. Soluções de armazenamento são encontradas nos três modelos, porém, embora possuam diferentes formas de acesso aos dados, não foram categorizadas. Neste trabalho, estabeleceu-se a forma de acesso aos dados como critério de classificação dos serviços. Com isso, pode-se identificar que o modelo de serviço IaaS fornece o acesso aos dados através de blocos de armazenamento; o PaaS, disponibiliza o acesso através de objetos encapsulando o armazenamento com maior abstração, quando comparado com o acesso a

bloco de dados; e, finalmente, o SaaS o acesso aos dados baseia-se em acesso direto a arquivos.

A definição da arquitetura comum para suporte ao serviço de armazenamento na nuvem baseou-se nas patentes dos três principais clientes de sincronização de arquivos para nuvem, *DropBox*, *Google Drive* e *OneDrive*. Esta arquitetura é composta por três elementos: camada de aplicação, gerenciamento e armazenamento distribuído. As arquiteturas e distribuições diferem na forma de comunicação entre eles elementos e na disponibilidade de serviços habilitados no módulo de gerenciamento.

O Cliente Híbrido para Sincronização de Arquivos em Nuvem (CHSAN) aplica técnicas já consolidadas em armazenamento distribuído no contexto de sincronização com nuvens de armazenamento, porém contorna as limitações de capacidade com o uso de armazenamento local, hierárquico e seletivo. O CHSAN permite a indexação de todo o sistema de arquivos da nuvem, além de permitir a consulta por seus utilizadores de todo o conteúdo armazenado na nuvem sem a necessidade de ter conteúdo dos arquivos no disco local. O conteúdo do arquivo pode estar no disco local, em um sistema distribuído na rede interna ou na nuvem. A localização da melhor forma de restituir os dados do arquivo é feita pelo CHSAN de forma transparente ao usuário e ao sistema operacional. Ainda, CHSAN coexiste com os clientes atuais de forma transparente.

### 1.3 Organização do Texto

O restante deste trabalho está organizado da seguinte forma: O Capítulo 2 apresenta a fundamentação teórica, revisando conceitos associados à nuvem computacional, computação nas organizações e a categorização dos serviços de acesso a nuvem. O Capítulo 3 descreve a arquitetura comum do serviço de armazenamento na nuvem e a comparação entre os principais provedores. A arquitetura proposta, Cliente Híbrido para Sincronização de Arquivos em Nuvem (CHSAN), é descrita no Capítulo 4. A análise qualitativa e experimental é objeto do Capítulo 5,

comparando CHSAN a arquitetura comum (discussão qualitativa), e experimentalmente, apresentando os resultados obtidos em dois cenários: (i) impacto no armazenamento híbrido no desempenho de operações sobre arquivos no acesso ao repositório de metadados comparado a outros meios de armazenamento e (ii) impacto da transferência híbrida que analisa os impactos na latência da arquitetura CHSAN comparada a do cliente *OneDrive*. O Capítulo 6 apresenta as considerações finais e trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os conceitos básicos, relacionados à computação na nuvem, um tema recente e popular. A classificação do armazenamento com o devido relacionamento entre os modelos de serviço de nuvem, também faz parte deste estudo, assim como o vínculo entre os conceitos e definições de organizações e o relacionamento destes com os serviços de armazenamento de dados. O serviço de acesso a arquivos na nuvem é foco principal deste trabalho e em especial deste capítulo.

### 2.1 Computação na Nuvem

A computação em nuvem baseia-se no princípio de alocação de recursos sob demanda. Esta filosofia permite a otimização dos recursos físicos de *data centers* (THENG; HANDE, 2012). A forma de provisionamento dos recursos é descrita através de modelos de serviço.

Soluções em nuvem são encontradas nos três modelos de serviços definidos por Zhang (ZHANG, 2010) em 2010 e reafirmados em 2011 por (HÖFER; KARAGIANNIS, 2011) e (MELL; GRANCE, 2009). Os três modelos de serviços são: IaaS - *Infrastructure as a service*; PaaS - *Platform as a Service*; e SaaS - *Software as a Service*. O modelo IaaS é a abstração da infraestrutura física, baseado na virtualização, que fornece ao usuário o acesso direto ao *hardware* virtualizado contratado. No PaaS, o usuário interage com recursos contratados através de plataformas de desenvolvimento. SaaS, nível mais alto de abstração, o acesso é efetuado via aplicação. Alguns exemplos de aplicações SaaS comuns são editores de texto e ferramentas para edição de vídeos. O armazenamento na nuvem é transversal aos três modelos de serviço, dispondo de maneiras distintas de acesso ao serviço, conforme o modelo enquadrado.

No que concerne aos serviços de computação na nuvem, NIST apresenta cinco características essenciais (MELL; GRANCE, 2009). São elas: auto-serviço sob demanda, amplo acesso por rede, agrupamento de recursos, elasticidade e serviço mensurado. A administração de organizações modernas foca suas ações em planejamento, ordenação e controle sobre os recursos das organizações (CHIAVENATO, 2005), e as características de NIST facilitam as ações de administração, uma vez que optimizam o capital, equipamentos e pessoas. Um serviço elástico e mensurado facilita a dinâmica de planejamento, assim como o controle dos custos, uma vez que existem regras estabelecidas em função do consumo. Mesmo assim, 45% das organizações não pensam na adoção de soluções em nuvem computacional (SMITH, 2013), devido a ausência de serviços focados nos interesses dessas organizações.

O armazenamento para as organizações que não se utilizam de nuvem usa sistemas legados associados a tecnologias acessíveis em redes privadas, locais ou metropolitanas. Esses armazenamentos são desenvolvidos sobre soluções *Network File System (NFS)*, *Common Internet File System (CIFS)* ou *Server Message Block (SMB)* e serão denominadas doravante como *NAS (Network Attached Storage)* (GIBSON; METER, 2000). O armazenamento em nuvem modifica esses acessos fazendo com que o fluxo passe a depender de conectividade com à Internet, além de modificar os protocolos de acesso ao dado, à maneira de como esses dados são compartilhados e à forma pela qual o controle de concorrência a tais dados é estabelecida. As organizações desejam estender suas formas de armazenamento agregando características da computação em nuvem importantes para seu planejamento e mantendo os sistemas legados ativos.

## 2.2 Classificação do Armazenamento na Nuvem

Em nuvens computacionais, especificamente no provisionamento de serviços de armazenamento, usuários estabelecem contratos com for-

necedores, usualmente através de menus de auto-serviço, especificando os requisitos que desejam: o tipo de serviço; a forma de acesso aos dados, em dispositivos móveis ou computadores; o pacote de recursos mais adequado às suas necessidades naquele momento. Diversos pacotes de recursos são oferecidos pelos provedores, permitindo que os contratantes aumentem ou diminuam a configuração do serviço, adequando o provisionamento às necessidades das aplicações finais. O modelo de negócio proposto por provedores de armazenamento em nuvem contempla serviços de degustação, franquia livre ou franquia com capacidade limitada (exemplo: vazão máxima, volume trafegado e espaço de armazenamento). Soluções de armazenamento são encontradas nos três modelos de serviços: IaaS - *Infrastructure as a service*; PaaS - *Platform as a Service*; e SaaS - *Software as a Service* (ZHANG, 2010) (MELL; GRANCE, 2009), conforme descrito na Figura 2.1. Da esquerda para a direita, têm-se: a identificação dos usuários finais, os recursos gerenciados em cada nível, exemplos de serviços e de armazenamento. As três primeiras colunas estabelecem o vínculo dos modelos de serviço de nuvem com cada nível, gerenciamento de recursos e exemplos de serviços, respectivamente. A última coluna vincula os serviços de armazenamento em nuvem às classificações estabelecidas.

O principal critério de classificação para estes serviços de armazenamento é a forma de acesso aos dados. Em IaaS, o acesso é realizado através de blocos de armazenamento, como por exemplo os serviços *Amazon Elastic Block Store* (EBS) e *Rackspace Cloud Block Storage* (CBS). Neste modelo, o usuário da nuvem tem a gerência total do hardware virtualizado e o acesso aos dados localizados nas unidades de armazenamento é idêntico ao de um dispositivo local. Em PaaS, o acesso aos dados é efetuado através de objetos representativos encapsulando o armazenamento com maior abstração, quando comparado com acesso a bloco de dados. Neste modelo de serviço, os dados estão fortemente acoplados a interfaces, métodos e ações, sendo disponibilizados através de *frameworks* de desenvolvimento. *Amazon S3*, *Google Cloud*

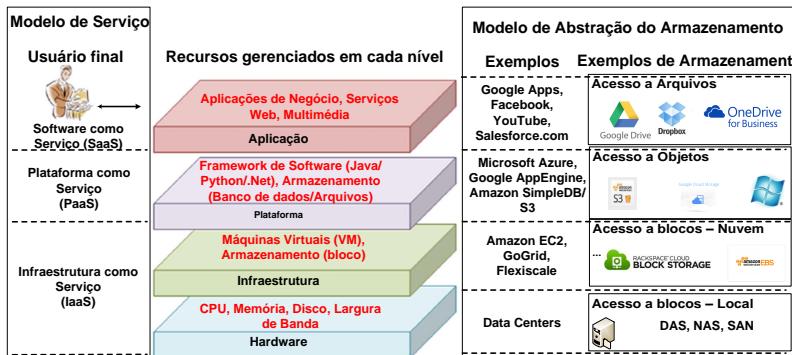


Figura 2.1: A aplicação dos modelos de serviço ao Armazenamento em Nuvem. Figura extraída de (HÖFER; KARAGIANNIS, 2011) e (ZHANG, 2010) e adaptada pelo autor.

*Storage* e *Microsoft Azure* representam exemplos dessa categoria. Finalmente, para o modelo de serviço SaaS, o acesso aos dados baseia-se em acesso direto a arquivos, sendo independente da plataforma de armazenamento e infraestrutura computacional. Os principais exemplos do serviço de acesso a arquivos são *DropBox*, *Google Drive* e *OneDrive*. Os aplicativos de sincronização deste modelo de serviço de armazenamento tratam seus dados como arquivos, que são armazenados na nuvem e estão acessíveis de qualquer lugar por qualquer dispositivo.

Analizando exclusivamente armazenamento de dados, PaaS possui a maior abstração quando comparado a SaaS e IaaS. Este tipo de serviço, permite o acesso aos dados através de requisições a objetos, e pode ser parte compositora dos serviços de SaaS e IaaS (DRAGO, 2012) quando apoiados em armazenamento distribuído. SaaS, por sua vez, possui interseção com serviços de armazenamento distribuído, apoiando-se em ferramentas e protocolos já existentes de sincronização de arquivos (e.g., rsync<sup>1</sup>). Finalmente, IaaS fornece o acesso compartilhado, com virtualização. Os servidores de arquivos apoiam-se no modelo de acesso a

<sup>1</sup> <https://rsync.samba.org>

bloco ou acesso a arquivo. Equipamentos dedicados exclusivamente para armazenamento, como NAS (*Network-Attached Storage*), compõem um *data center* e este, por sua vez, fornece infraestrutura para computação na nuvem. Especificamente, IaaS é suportado por ferramentas, protocolos e tecnologias de sistemas de arquivos distribuídos em *data center*, como por exemplo SAN - *Storage Area Network*, iSCSI *Internet Small Computer System Interface* e DAS *Directed Attached Storage*. As principais diferenças entre estas tecnologias são: forma de conexão, protocolos de transporte de dados e/ou infraestrutura (TATE, 2005).

O modo de acesso aos dados determina o modelo de serviço de armazenamento em nuvem mais adequado aos objetivos da organização. As peculiaridades, sejam elas de usuários ou de organizações, estão diretamente relacionadas ao modelo de serviço de armazenamento contratado. A classificação de armazenamento em nuvem proposta por este trabalho facilita a identificação de modelo de serviço e o vínculo ao armazenamento escolhido.

O conceito de computação na nuvem surgiu em meados de 2009, porém a maioria dos protocolos e ferramentas de sistemas de arquivos foram concebidos antes disso. A compreensão da interdependência destas ferramentas e protocolos com os serviços de armazenamento em nuvem, vinculados a modelos de serviços (IaaS, PaaS e SaaS), são representados na (Figura 2.2). O armazenamento na nuvem está representado na elipse à esquerda, em tons de azul, e as ferramentas e protocolos na elipse à direita, em tons de laranja. Pode-se observar que PaaS possui o conceito mais abstrato dos três serviços, englobando as características de armazenamento na nuvem de SaaS e IaaS. Este tipo de serviço, em que o acesso aos dados se dá através de requisições a objetos pode fornecer espaço de armazenamento para os serviços de SaaS e IaaS (DRAGO, 2012), quando apoiado em armazenamento distribuído. SaaS, por sua vez, possui interseção com serviços de armazenamento

distribuído. Os serviços fornecidos em SaaS apoiam-se em ferramentas e protocolos já existentes de sincronização de arquivos, tais como *rsync* (TRIDGELL, 1996). Estes serviços podem ou não ser hospedados em uma infraestrutura virtual (IaaS). Finalmente, IaaS fornece o acesso compartilhado, com virtualização, a ferramentas e protocolos de servidores de arquivos. Os servidores de arquivos, elipse à direita, apoiam-se no modelo de acesso a bloco ou acesso a arquivo, mesmo princípio utilizado por IaaS. Equipamentos dedicados exclusivamente a armazenamento, NAS (*Network-Attached Storage*), compõem um *data center* e este, por sua vez, fornece infraestrutura para computação na nuvem. Esta interdependência é representada pela pequena elipse laranja contida na elipse azulada IaaS, à esquerda. Protocolos, ferramentas e tecnologias de sistemas de arquivos distribuídos em *data center* com NAS são representados na elipse laranja à direita (SAN - *Storage Area Network*, iSCSI *Internet Small Computer System Interface* e DAS *Directed Attached Storage*). As principais diferenças entre estas tecnologias é a forma de conexão, protocolos de transporte de dados e/ou infraestrutura (TATE, 2005). Nesse ambiente, o acesso aos dados é feito por acesso a blocos, utilizando-se SAN, iSCSI ou DAS.

### 2.2.1 Conceitos e Definições

O armazenamento em nuvem trata problemas conhecidos e estudados por sistemas de arquivos locais e remotos em redes privadas. Dentre os conceitos, para a compreensão deste trabalho, destacam-se : NAS versus SAN, *cache*, versionamento, deduplicação, sistema de arquivos distribuído e FUSE.

- **NAS - (*Network Attached Storage*) versus SAN (*Storage Area Network*):** NAS e SAN oferecem diferentes níveis de abstração para acesso aos dados. NAS oferece acesso a arquivos, via rede e independente da localização física dos dados, enquanto SAN oferece acesso a blocos dentro de uma geometria rígida que repre-

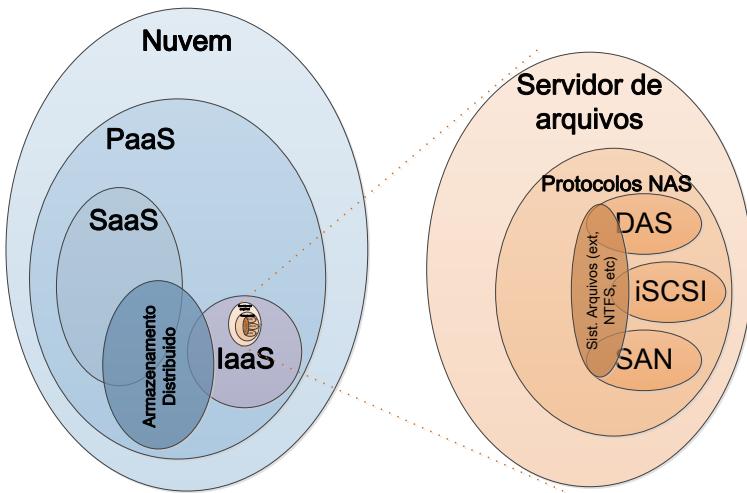


Figura 2.2: Interação do servidor de arquivos tradicional e os modelos de Serviço de nuvem.

sente um disco rígido, mesmo que lógico, de tamanho finito. Na figura 2.2 é possível observar que NAS exporta um sistema de arquivos, fornecendo ao cliente arquivos, via protocolos de rede. O SAN é uma das tecnologias para fornecer blocos a um sistema de arquivos (GIBSON; METER, 2000; TATE, 2005).

- **Cache em sistemas de arquivos:** A técnica de *cache* se caracteriza por armazenar temporariamente uma informação com finalidade de otimizar o tempo de acesso ou garantir-o mesmo quando uma fonte primária não está disponível (TANENBAUM; WOODHULL, 2009). O tema *cache* é antigo e utilizado em conjunção com sistemas de arquivos e dispositivos de armazenamento. Trabalhos como (MUTHITACHAROEN, 2001; ANNAPUREDDY, 2005; SRINIVASAN; MOGUL, 1989; GHEMAWAT, 2003) utilizam *cache* como forma de otimização reduzindo o tempo de latência. (SRINIVASAN; MOGUL, 1989) explora ainda a questão da consistência e coerência de *ca-*

*che*. No contexto do serviço de acesso a arquivos em nuvem SaaS, a coerência é garantida pelo serviço que trata os conflitos não sobrepondo os dados. No contexto de armazenamento na nuvem nas organizações, (GONÇALVES, 2015; VRABLE, 2012) propõem o uso de *cache* para redução do consumo dos enlaces de internet. O *cache* mesmo sendo um armazenamento temporário é utilizado como possibilidade de acesso *off-line* a conteúdos que tenham sido previamente baixados (BHAT, 2007) e Microsoft<sup>2</sup>.

- **Deduplicação:** Consiste em uma técnica especializada em compressão de dados que permite a eliminação de blocos duplicados. Em sistemas de arquivos distribuídos o uso do conceito de *chunk* é corrente. Arquivos são divididos em unidades menores de tamanhos fixos, denominados de *chunk*. O uso da técnica de deduplicação em sistemas de arquivo pode ser aplicado à unidade *chunk*. Na Figura 2.3, apresenta-se um exemplo de deduplicação com o uso de *chunk*. Dois arquivos são armazenados em um mesmo sistema de arquivos, Foo.bin e Foo\_V1.bin. Eles possuem 16 MB cada e somente um *chunk* (4 MB) os diferem. Associado a cada um tem-se um identificador único (*hash*). Portanto, os *chunk* duplicados, ABC, GHI e JKL são armazenados uma única vez mesmo que eles estejam associados a arquivos distintos, neste caso, provendo uma redução de 12 MB (HOUSTON; FERDOWSI, 2014).
- **Versionamento:** Consiste na técnica de controlar a versão de um arquivo. O sistema permite a manipulação, tanto de leitura quanto de escrita, de arquivos sem bloqueio. A técnica de deduplicação otimiza de tal forma o armazenamento que os provedores de armazenamento PaaS e SaaS, parametrizam seus sistemas de arquivo, para sempre criar uma nova versão, armazenando efetivamente apenas os *chunks* que possuam conteúdo diferente, mantendo a rastreabilidade do arquivo, informação do conjunto de *chunk* que pertenciam ao arquivo em um dado momento, junto aos metada-

<sup>2</sup> <https://technet.microsoft.com/library/ff183315.aspx>

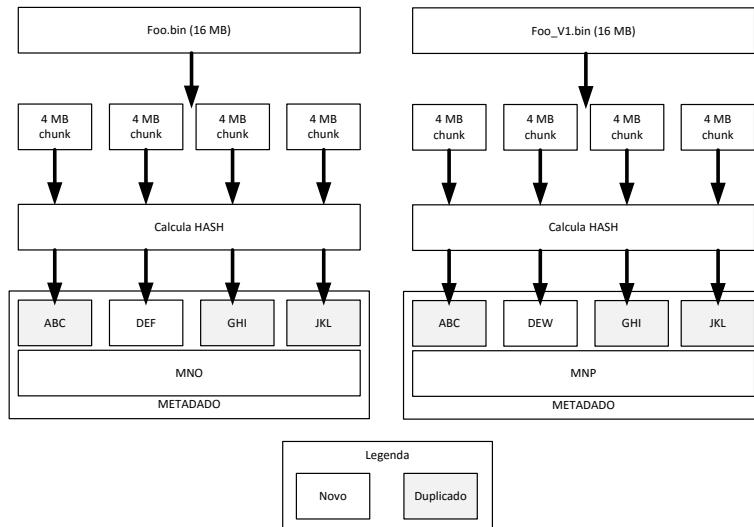


Figura 2.3: Deduplicação e versionamento em funcionamento nos sistema distribuídos, inspirado na patente do *DropBox* (HOUSTON; FERDOWSI, 2014)

dos. Isso possibilita também que em caso de conflito, o sistema de arquivo crie uma cópia do arquivo, sem sobrepor o original e com um custo de armazenamento baixo.

- **Sistema de arquivos distribuídos e FUSE:** Sistemas de arquivos distribuídos, como Amazon S3 S3FS (MUNISWAMY-REDDY, 2010), Google FileSystem (GHEMAWAT, 2003), Hadoop FS (SHVACHKO, 2010) têm como características principais a escalabilidade e o acesso a objetos denominados *chunk* via protocolo HTTPS apoiados em metadados. Outras características encontradas nos sistemas distribuídos são: versionamento dos arquivos e deduplicação de dados (BHAGWAT, 2009; DRAGO, 2012).

O sistema de arquivos é a parte de um sistema operacional res-

ponsável por gerir os arquivos, na forma como são estruturados, nomeados, acessados e utilizados. As operações com arquivos, leitura, escrita, atribuição de propriedades, nomeação e endereçamento são comandadas pelo núcleo do sistema operacional. O núcleo do sistema operacional tem embutido a funcionalidade para operar com um ou mais sistemas de arquivo. O VFS (Virtual File System) (PATIL; GIBSON, 2011) é um sistema de arquivos cujo objetivo é abstrair chamadas de sistema, uniformizando o acesso a arquivos, independente se este arquivo encontra-se armazenado no disco local, em uma arquitetura NAS ou na nuvem. O VFS é implementado junto ao núcleo, e devido a isso perde em flexibilidade e ganha em desempenho. A flexibilidade é perdida, pois uma vez embutido no núcleo, qualquer modificação necessita de uma nova versão do núcleo. O projeto FUSE - *Filesystem in Userspace*<sup>3</sup> permite características semelhantes ao VFS no que se refere a acesso aos arquivos. No entanto, a sua abstração é feita no nível do usuário, ou seja sem modificação do núcleo (PATIL; GIBSON, 2011; YU, 2012).

Esta característica permite que FUSE seja a base para acesso a arquivos em arquiteturas de sistemas de arquivos distribuídos com alta capacidade como: GIGA+ (PATIL; GIBSON, 2011); CEPH (WEIL, 2006); GlusterFS (NORONHA; PANDA, 2008) e MooseFS (YU, 2012). Estes sistemas de arquivos distribuídos têm em comum o uso do FUSE e a utilização de *cache*. FUSE para esses projetos permite criar um novo sistema de arquivos que possibilite o acesso as operações E/S de forma transparente para os sistemas de arquivos distribuídos.

---

<sup>3</sup> <http://fuse.sourceforge.net/>

## 2.3 Computação nas Organizações

Uma organização pode ser formada por um ou mais membros que colaboram por um objetivo comum. Organizações de um único membro podem colaborar com seus clientes e fornecedores. As organizações podem ser centralizadas, descentralizadas com ou sem fins lucrativos (WEIL, 2006). O indivíduo difere da organização, mesmo quando a organização é de um único indivíduo, pelos objetivos que apresentam. Usuários domésticos têm interesses distintos entre si e tendem a colaborar pouco (GONÇALVES, 2015), quando comparados com usuários das organizações, pois os usuários das organizações são exatamente o oposto, têm interesses comuns e tendências de colaboração. Os ambientes organizacionais, alvo deste trabalho, são constituídos por múltiplos usuários que colaboram em projetos comuns, muitas vezes em ambientes geographicamente distribuídos, concentrados em redes privadas. Na Figura 2.4, tem-se um cenário exemplificando uma organização que possui dois grupos: os usuários em vermelho e os em verde, situados fisicamente nas cidades de Nova York, Paris, Joinville e Praga. Esta organização tem dois escritórios físicos, localizados em Nova York e Paris, onde se concentram o maior número de colaboradores que interagem por meio de redes privadas interligadas. Em Joinville e Praga, encontram-se dois usuários geographicamente distantes dos escritórios que colaboram com seus grupos via Internet.

O compartilhamento de arquivos entre estes usuários pode ser efetuado via sistemas de armazenamento privado, desenvolvidos sobre soluções NFS, CIFS ou SMB. Estas soluções são usualmente empregadas em redes privadas locais ou ambientes geographicamente distribuídos, interconectados com enlaces de comunicação dedicados. Os sistemas de armazenamento privado contemplariam a colaboração entre os usuários de Paris e de Nova York, no entanto, esta escolha não permitiria a integração dos usuários de Joinville e Praga, pois eles estão fora da rede privada com uma conexão Internet. Para estes usuários, uma alternativa

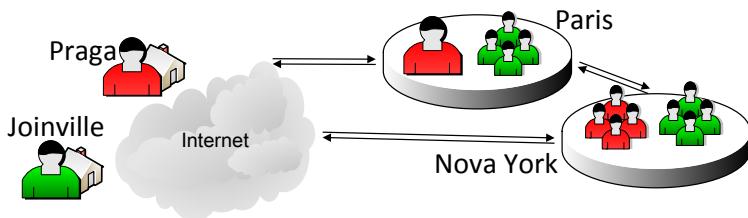


Figura 2.4: Ambiente organizacional com múltiplos usuários colaborativos.

viável remete ao armazenamento em nuvem computacional.

A alternativa de nuvem viabiliza a colaboração dos usuários remotos, como Joinville e Praga, porém poderia degradar a qualidade do acesso aos serviços dos demais colaboradores que se encontram concentrados geograficamente.

## 2.4 Armazenamento nas Organizações

No contexto das organizações, o legado de armazenamento em rede NAS implantado se faz presente, uma vez que como mencionado anteriormente, 45% das empresas ainda não têm previsão de migração de qualquer serviço para a nuvem. Entender as características que auxiliem as organizações a quantificar o impacto, positivo ou negativo, de cada modelo de armazenamento na infraestrutura é essencial. As características essenciais definidas pelo NIST(MELL; GRANCE, 2009) permitem a comparação entre os modelos de serviço (IaaS, PaaS, SaaS) e o armazenamento legado que existe nas organizações. As cinco características do NIST, identificadas como características essenciais para comparação de serviços de armazenamento, são:

1. **Auto-serviço sob demanda:** com essa característica o cliente pode provisionar por conta própria recursos de armazenamento, espaço e velocidade de acesso, sem a necessidade de intervenção humana dos provedores de serviços. No contexto da organização, o mapeamento de atribuições é importante, pois o cliente que utiliza o serviço pode ser diferente do cliente que contrata/paga.
2. **Amplio acesso por rede:** aos dispositivos de armazenamento, incluindo dispositivos móveis. A integração entre células ou setores de uma organização é facilitada pelo acesso rápido e consistente à informação. Possuir um sistema de armazenamento integrado, acessível via rede interna ou externa à organização, com atualização simultânea das informações é um diferencial.
3. **Agrupamento de recursos:** gerenciado pelo provedor de forma que atenda a múltiplos usuários em modalidade multi-inquilinos, com recursos físicos ou virtuais diferentes. Visto pelo cliente, os recursos estão simplesmente disponíveis em qualquer lugar e a qualquer momento. O provedor, por sua vez, atribui a localização física das informações, definindo país, estado ou *data center* onde as informações se encontram. A transparência de localização, via agrupamento de recursos, gera uma preocupação adicional no contexto organizacional: a confidencialidade dos dados. O contrato de serviço (SLA - *Service Level Agreement*) permite a inclusão de cláusulas que restrinjam o armazenamento de dados em determinados países ou continentes (KHAJEH-HOSSEINI, 2010).
4. **Elasticidade rápida:** característica de armazenamento que permite o incremento ou decremento do provisionamento de espaço ou velocidade de acesso a um dispositivo de armazenamento. O cliente tem a impressão de que os recursos são ilimitados e que podem ser alocados em qualquer quantidade a qualquer tempo. Esta flexibilidade permite à organização trabalhar em projetos com exigências sazonais, em que por um determinado mês, por exemplo, a capacidade de armazenamento para o desenvolvimento de um

projeto triplique a média anual. O espaço extra de armazenamento para o projeto nesse mês determinado é contratado e no mês seguinte reduzido.

5. **Serviço mensurado:** disponibilizado aos clientes da nuvem através de ferramentas para o controle e otimização dos recursos alocados. No que se refere ao serviço de armazenamento para organizações, o controle de espaço utilizado por um membro ou a comparação entre a quantidade contratada e a utilizada pode gerar economia para a organização.

As características apresentadas acabam por ter herança em tecnologias que constituem o armazenamento. Um serviço de nuvem, como de acesso a arquivos, herda características do serviço de acesso a objetos, assim como este herda do sistema de arquivo distribuído e de sua infraestrutura de acesso a blocos. Tal relacionamento existe entre os modelos de serviço, mas também é intrínseco às diferentes formas de armazenamento.

#### 2.4.1 Modelos de Serviço Focados nas Organizações

As características essenciais de NIST estão alinhadas com as organizações na busca da otimização e da flexibilidade necessária para acompanhar as constantes evoluções tecnológicas. Cada modelo de serviço (SaaS, PaaS, IaaS) proposto para as organizações atualmente tem suas limitações e impõe novos desafios, como uma maior dependência dos enlaces com à Internet e a mudança no perfil do usuário. Mesmo o armazenamento em NAS, amplamente difundido, não atende mais às necessidades atuais das organizações, uma vez que o NAS não propicia a mobilidade necessária ao armazenamento de dados. No contexto em que os serviços NAS foram desenvolvidos, a Internet, os dispositivos móveis, o armazenamento de áudio visual não eram tão difundidos. O serviço de armazenamento de acesso a blocos está vinculado ao modelo de serviço

da nuvem IaaS, Infraestrutura como Serviço.

O modelo de serviço de armazenamento com acesso a blocos é implantado em rede local ou na nuvem. Em rede local, os protocolos e ferramentas (DAS, NAS e SAN) são os mesmos utilizados em *data centers* tradicionais que se utilizam de ambientes virtualizados ou não (TATE, 2005). O acesso a blocos de dados na nuvem é provido por protocolos e por ferramentas específicas que, variam de provedor para provedor, tais como *Block Storage*, da *Rackspace Cloud*,<sup>4</sup> e (*EBS*), da *Amazon EC2*<sup>5</sup>.

O acesso a objetos, modelo de serviço de armazenamento associado a PaaS, Plataforma como Serviço, encapsula o armazenamento ao conceito de objetos. O nível de abstração é maior, se comparado com acesso a bloco de dados. O cliente do serviço da nuvem PaaS interage com a plataforma através de um *framework* de desenvolvimento. Neste modelo de serviço, os dados estão fortemente acoplados a interfaces, métodos e ações, justificando a transferência direta e armazenamento de objetos. Os protocolos de comunicação e armazenamento são HTTP e HTTPS, independentes de plataforma, porém os *frameworks* de manipulação e armazenamento dos objetos diferem de provedor para provedor.

O modelo de serviço de armazenamento e acesso a arquivos está associado ao modelo de serviço na nuvem SaaS, Software como Serviço. Este modelo de serviço de armazenamento é independente de plataforma e de infraestrutura. O serviço é composto por módulos dedicados, que desempenham tarefas como notificação, indexação e interface de operações. Entre os módulos dedicados estão o armazenamento distribuído PaaS. Os serviços de acesso a arquivos são caracterizados por estes módulos e um cliente de acesso a arquivos (DRAGO, 2012). O sistema de armazenamento distribuído está implantado na nuvem de um

<sup>4</sup> [http://www.rackspace.com/managed\\_hosting/services/storage](http://www.rackspace.com/managed_hosting/services/storage)

<sup>5</sup> <http://aws.amazon.com/ec2/>

provedor de serviço PaaS e provê o acesso aos arquivos através da rede.

Os módulos dedicados controlam e gerenciam os metadados; notificam os clientes sobre atualizações e identificam a localização dos *chunk* com as respectivas URL no servidor de armazenamento. Os clientes de acesso a arquivos são disponibilizados por seus desenvolvedores, em várias plataformas, inclusive para dispositivos móveis. Este serviço tornou-se rapidamente popular, principalmente, devido às suas características de heterogeneidade, fácil utilização, disponibilidade e modelo de negócio com franquia gratuita (NALDI; MASTROENI, 2013).

O cliente deste modelo de serviço de armazenamento trata seus dados como arquivos, que são armazenados na nuvem e estão acessíveis de qualquer lugar por qualquer dispositivo. O tipo de negócio deste modelo prevê dois perfis de usuários: **básicos**, com capacidade de armazenamento e velocidade de acesso reduzidas, sem pagamento de mensalidades e **avançados**, cuja a capacidade e a velocidade de acesso podem ser adequadas às necessidades dos usuários (KATZER; CRAWFORD, 2013) e (DRAGO, 2012). Os clientes de acesso a arquivos ainda preveem o modelo de negócio empresarial (Google drive<sup>6</sup>, DropBox<sup>7</sup> e Onedrive<sup>8</sup>) em que o conceito de colaboração para projetos é incorporado à capacidade de armazenamento e à velocidade de acesso adequadas a um maior número de usuários (DRAGO, 2013). O ambiente organizacional utiliza os dispositivos NAS para prover velocidade e capacidade de armazenamento em rede local, o modelo de serviço que mais se assemelha ao NAS é o IaaS, último modelo que esta seção pretende descrever.

O modelo de acesso a blocos IaaS não difere do NAS, apenas transporta o NAS para a nuvem. O acesso a este tipo de armazenamento

<sup>6</sup> <https://www.google.com/work/apps/business/products/drive/>

<sup>7</sup> <https://www.dropbox.com/business>

<sup>8</sup> <https://onedrive.live.com/about/business/>

sobre enlaces de Internet tem problemas de desempenho, além disso a alocação de espaço de armazenamento é rígida, pois é necessário alojar uma capacidade de disco estática e o aumento desta capacidade depende de intervenções manuais dos administradores. A escalabilidade do IaaS é inferior aos demais modelos de serviço de nuvem. O serviço de acesso a objetos tem a maior escalabilidade dentro dos modelos de serviço apresentados. O acesso a objetos muda completamente a forma de acesso aos arquivos, tornando-se incompatível com as aplicações legadas, por não ser transparente ao sistema operacional. O modelo de acesso a arquivos destaca-se como o modelo mais aderente a substituir o NAS por manter a compatibilidade com os sistemas legados, uma vez que se utiliza do cliente de acesso a arquivos para abstrair as características de acesso a objetos herdadas do acesso a objetos. O presente trabalho foca no acesso a arquivos como sucessor do NAS para as organizações.

## 2.5 Trabalhos Relacionados

No contexto das organizações, os problemas arquiteturais de clientes de sincronização identificados têm sido fontes de pesquisas científicas e de negócios em ambientes empresariais.

No que se refere a objetivos, as arquiteturas mais próximas ao CHSAN são BlueSky (VRABLE, 2012) e o SCFS (BESSANI, 2014). BlueSky fornece um sistema de arquivo em rede baseado no armazenamento em nuvem computacional, suporta múltiplos protocolos (NFS/CIFS) e está integrada com diferentes fornecedores de nuvem (Amazon S3 e Windows Azure). SCFS disponibiliza o acesso a serviços de armazenamento em nuvem seguindo a semântica near-POSIX com o auxílio do FUSE-J. No âmbito comercial, tem-se soluções como Nasuni (UNIFS..., 2015) e Panzura (PANZURA..., 2008) que fornecem "virtual NAS appliance" e gateway CIFS/NFS, respectivamente, vinculado a fornecedores de nuvem computacional. Todas as soluções apresentadas atacam cloud-backed file systems. As comerciais, por serem fechadas, não dispõem de

documentação pública detalhada que permita comparações. Finalmente, o cenário de uso do BlueSky, Nasuni e Panzura contempla organizações geograficamente concentradas ignorando a existência de colaboradores em home office, uma prática em expansão na atualidade, ou o modelo de organizações geograficamente distribuídas.

Soluções de sistemas de arquivos com alta disponibilidade e escalabilidade, aderentes ao modelo de organizações geograficamente distribuídas, são disponibilizadas, diretamente, por fornecedores de nuvem computacional. GFS (GHEMAWAT, 2003), S3FS (MUNISWAMY-REDDY, 2010), CSTORE (DUAN, 2015) e Hadoop (SHVACHKO, 2010) (non-POSIX) não suportam sistemas de arquivos legados disponíveis em redes privadas das organizações. Desta forma, o cliente tem que escolher entre a manutenção de soluções legadas e o incremento de funcionalidade advinda com sistemas de arquivos vinculados a nuvem computacional.

Os clientes de sincronização em nuvem computacional populares, *DropBox*, *OneDrive* e *GoogleDrive*, baseiam-se no princípio da cópia local de todos os arquivos compartilhados. Este princípio é conveniente, no contexto de usuários domésticos, onde a restrição da capacidade de sincronização encontra-se no lado do fornecedor e o enlace à Internet, normalmente, não é compartilhado com outros colaboradores. A caracterização do tráfego de dados destes clientes em ambientes organizacionais é tema de recentes publicações (DRAGO, 2012; DRAGO, 2013; GONÇALVES, 2014; GONÇALVES, 2015) e (GONÇALVES, ). Drago (DRAGO, 2012) e (DRAGO, 2013) identificaram o tráfego de dados da ferramenta *DropBox* em um ambiente universitário, indicando que o uso de recursos de rede para sincronização de arquivos alcançou um terço do tráfego para visualização de vídeos no cenário estudado. Estes trabalhos caracterizaram o comportamento dos usuários contabilizando o volume de dados, tamanho médio dos arquivos e atividades de sincronização. Eles também forneceram uma perspectiva sobre o uso de clientes de sincronização em usuários domésticos, embasando a realização da presente proposta, porém, não se preocuparam com ambientes organizacionais com múltiplos

usuários. Nestes ambientes, a flexibilidade e a heterogeneidade são características essenciais para um sistema de arquivos (TATE, 2005) (SHEPLER, 2003) (Microsoft Corporation, 2014) (FRENCH; TEAM, 2007), enquanto a escalabilidade, a alta disponibilidade e a segurança são desejáveis (TATE, 2005). CHSAN atende às características essenciais de sistemas de arquivos e ainda se preocupa com a escalabilidade e a alta disponibilidade, combinando três aspectos: o acesso ao disco local, aos sistemas de armazenamento privado, e ao armazenamento na nuvem.

Quanto às propostas para otimização do desempenho de clientes de sincronização, (WANG, 2012) e (LI, 2013) apresentaram técnicas para diminuir o tráfego total, tempo de acesso aos arquivos e *overuse*, no qual o tráfego de dados é maior que a quantidade de dados efetivos. O cliente de sincronização *Dropbox* destaca-se neste quesito, pois é o único que fornece a possibilidade de sincronização direta entre clientes localizados em uma mesma rede lógica por meio do protocolo *LanSync*. Nessa linha, CHSAN inova ao permitir a sincronização entre clientes concentrados em redes privadas, estando na mesma rede lógica ou não, utilizando-se de hierarquia de cache de dois níveis (local e na rede privada).

A aplicação de técnicas de cache, em sistemas de arquivos distribuídos, com o intuito de acelerar a manipulação de arquivos é utilizada nos trabalhos (GRUENWALD III, 2015) (VRABLE, 2012) e (BESSANI, 2014). BlueSky (VRABLE, 2012) possui somente um nível de cache e usa o algoritmo de coerência de cache write-back. SCFS (BESSANI, 2014) dispõe dois níveis de cache gerenciados pelo SCFS Agente, o primeiro em disco com capacidade na unidade de GB e o segundo em memória (centenas de MB). O algoritmo LRU de substituição de páginas é o utilizado pelo SCFS Agente.

## 2.6 Considerações Parciais

A computação na nuvem tem um impacto positivo nas organizações, pois as características essenciais de NIST estão alinhadas com as necessidades das organizações. A computação nas organizações difere

daquela de uso doméstico, essencialmente pelo propósito; usuários domésticos têm objetivos difusos, enquanto organizações têm objetivos específicos. Os modelos de serviço de armazenamento em nuvem, acesso a arquivos, acesso a objetos e acesso a blocos são uma classificação proposta pelo autor para identificar serviços de um mesmo grupo, que se destina explicitar as necessidades das organizações. Este trabalho tem foco em organizações com múltiplos usuários, geograficamente distribuídos, que tenham interesse em utilizar o armazenamento em nuvem no modelo de serviço de acesso a arquivos. Os clientes de acesso a arquivos atuais se utilizam dos discos locais para terem transparência as suas aplicações. O sistema de arquivos virtual (VFS) é o responsável por esta transparência que é implementada na camada do núcleo e que em conjunto com o módulo FUSE propicia utilização de sistemas de arquivos novos em modo usuário. Técnicas de deduplicação e versionamento baseadas em *chunk* e *hash* são empregadas para otimização dos sistemas de armazenamento distribuídos, utilizados pelos provedores de acesso a arquivos, bem como na transferência de arquivos entre o cliente de acesso a arquivos e o armazenamento na nuvem em seu modelo de serviço.

### 3 SERVIÇO DE ACESSO A ARQUIVOS EM NUVEM

Este capítulo descreve o serviço de acesso a arquivos em nuvem, suas características e identifica os principais componentes deste serviço. A arquitetura do serviço é dividida em camada de aplicação, gerenciamento e armazenamento distribuído. Detalha-se ainda as arquiteturas dos clientes *DropBox*, *Google Drive* e *OneDrive* que são os mais populares. Os provedores de serviço são comparados e analisados, apresentando-se um resumo de características e funcionalidades.

#### 3.1 Usuários Domésticos versus Usuários de Organizações

Em nuvem computacional, o acesso aos dados armazenados pode ser efetuado em diversas camadas. O acesso via SaaS, vinculado à camada com mais alta abstração, pode ocorrer através de uma console *web*, aplicações dedicadas ou clientes com serviço de sincronização. Clientes de sincronização de arquivos na nuvem são objeto de estudo da comunidade científica (DRAGO, 2012; DRAGO, 2013; LI, 2013; GONÇALVES, 2015), principalmente, pelo fato do consumo do enlace de Internet ser próximo dos maiores consumidores de rede tal qual fornecedores de vídeo sobre demanda.

Este trabalho tem como objetivo o estudo dos sistemas de armazenamento na nuvem disponibilizados via SaaS com foco nas organizações. Esses serviços de acesso tiveram em sua origem um foco de desenvolvimento centrado no usuário doméstico. As características de uso destes indivíduos são diferentes daquelas verificadas nas organizações (GONÇALVES, 2016). Entre essas diferenças destacam-se: a quantidade de compartilhamentos e a quantidade de participantes desses compartilhamentos. Tais quantidades influem diretamente na quantidade de tráfego gerado: quanto maiores as quantidades, maior o tráfego.

Um grupo de colaboração é formado por usuários do serviço de acesso a arquivos que possuem arquivos ou diretórios compartilhados com um ou mais usuários desse mesmo serviço. Usuários domésticos tendem a compartilhar menos arquivos ou diretórios e construir grupos de colaboração menores do que usuários de organizações (GONÇALVES, 2016). A natureza da organização constitui-se em grande número de pessoas e projetos vinculados a objetivos comuns, favorecendo a colaboração.

O uso do serviço de sincronização como facilitador do avanço destes projetos é desejado pelas organizações. A diferença entre usuários domésticos e da organização tornam, por alguns aspectos, inadequada a arquitetura atualmente utilizada por serviços de acesso a arquivos em nuvem pelas organizações.

## 3.2 Arquitetura do Serviço de Armazenamento na Nuvem

O serviço de acesso a arquivos na nuvem é constituído de uma parte localizada na própria nuvem e outra no dispositivo de acesso remoto. Este dispositivo pode ser um computador, um tablet, *smartphone* entre outros. Entre o serviço de acesso a arquivos disponíveis atualmente, pode-se identificar alguns componentes comuns. A arquitetura comum destes serviços, como pode ser observado na Figura 3.1, é constituída de três camadas: armazenamento distribuído, gerenciamento e camada de aplicação. A camada de aplicação é conectada aos outros dois através da API de comunicação, REST API.

- **Gerenciamento:** A camada de gerenciamento recai nos serviços disponibilizados na camada de aplicação: concessão e remoção de permissões de compartilhamento sobre arquivos e diretórios; con-

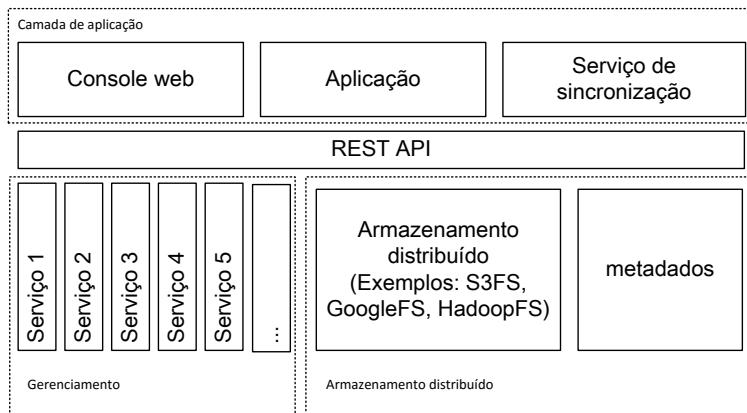


Figura 3.1: Arquitetura do serviço e seus componentes

cessão e remoção de compartilhamento de arquivos e diretórios; elasticidade, aumentando ou reduzindo a área de armazenamento.

- **Armazenamento Distribuído:** Na camada de armazenamento distribuído uma base de dados nomeada de metadados é utilizada, característica presente nos sistemas de arquivos distribuídos para manipulação de dados. Internamente o sistema de arquivos distribuído faz acesso a blocos e externamente manipula os dados via requisições HTTP. São exemplos de sistemas de armazenamento distribuído utilizados no serviço de acesso a arquivos na nuvem: S3FS (MUNISWAMY-REDDY, 2010), Google FileSystem (GHEMAWAT, 2003) e Hadoop FS (SHVACHKO, 2010).

### 3.2.1 Armazenamento Distribuído e Gerenciamento

Sistemas de arquivos distribuídos foram concebidos para a manipulação de grandes massas de dados, da ordem de petabytes, de forma eficiente e sob a ótica da escalabilidade. Eles possuem técnicas de armazenamento comuns, baseado na segmentação dos arquivos em *chunk*,

aproveitando-se da deduplicação para otimizar o armazenamento de partes comuns. Um *chunk* só é replicado para garantir a redundância física dos dados, portanto, se uma centena de usuários armazenar um mesmo arquivo *Foo.bin*, de 1 GB, este arquivo ocupará somente 1 GB de espaço no armazenamento disponível. Porém, estes 1 GB serão descontados da área de armazenamento de cada usuário. O acesso aos clientes é possível via *HTTPS* e *REST API* (BHAGWAT, 2009; DRAGO, 2012).

No contexto de armazenamento na nuvem, as camadas de armazenamento distribuído e gerenciamento são implantadas junto ao provedor de nuvem. Por definição, o provedor da nuvem encontra-se fora da rede interna das organizações. As funcionalidades destas camadas são:

- **Administração:** Consiste na operacionalização das tarefas: concessão e remoção de permissões de compartilhamento sobre arquivos e diretórios; concessão e remoção de compartilhamento de arquivos e diretórios; elasticidade, aumentando ou reduzindo a área de armazenamento; entre outras.
- **Versionamento:** Permite o gerenciamento, por parte do usuário, de diferentes versões de um mesmo arquivo. São mantidas versões dos arquivos à medida que os arquivos são alterados. Por meio do gerenciamento, o usuário pode consultar ou restaurar versões anteriores, inclusive as que tenham sido removidas.
- **Notificação:** Mantém os diferentes clientes informados sobre alterações que ocorram no sistema de armazenamento, informando o estado da conexão, incluindo alterações em arquivos e diretórios.
- **Autenticação:** O serviço de autenticação, por sua vez, resume-se a validação de credenciais, vinculando usuários autorizados à áreas e permissões, tanto do usuário autenticado quanto as de compartilhamento do mesmo.

### 3.2.2 Camada de Aplicação

A camada de aplicação é a mais próxima do usuário, cabe a ela a responsabilidade de prover acesso aos dados e às funcionalidades a vários dispositivos e plataformas com condições de acesso distintas. Os clientes de acesso ao armazenamento devem adequar-se ao dispositivo com capacidades de armazenamento na ordem de gigabytes conectados à Internet, como computadores pessoais conectados via ADSL e a dispositivos com capacidade de armazenamento na ordem de megabytes e à Internet, tais como smartphones via 3G.

A escolha do cliente mais adequado recai ao próprio usuário. A camada de aplicação é constituída de três modos de acesso distintos, exatamente para dispor de flexibilidade. São eles: console web, aplicação e cliente de sincronização.

- **Console web:** o uso de console web para o acesso ao serviço de armazenamento a arquivos na nuvem não exige a instalação de nenhuma aplicação específica. O acesso é possível através de navegador web compatível, sendo que as versões atuais mais populares (Internet Explorer, Firefox, Chrome e Opera) possibilitam tal acesso. Essa forma de acesso não necessita de espaço de armazenamento no disco local e efetua todas as operações diretamente na nuvem, gerando tráfego no enlace de acesso à Internet a cada operação. Portanto, a interação depende exclusivamente do desempenho do enlace de Internet.

Em computadores pessoais, esse modo de acesso não está integrado ao sistema de arquivos local, dificultando o uso de aplicações que se baseiam na estrutura local de arquivos.

Esse modo de acesso é adequado a interações em que o limitador não seja o enlace ou para acessos esporádicos.

- **Aplicação:** O modo de acesso aplicação tem por objetivo o desenvolvimento de operações específicas integradas à camada de armazenamento. Cada provedor dispõe de uma interface proprietária de desenvolvimento (API). A forma de acesso e as necessidades estão ligadas aos dispositivos, alvo desta aplicação.

Os clientes móveis de acesso a armazenamento na nuvem são desenvolvidos com estas API. Atualmente o mercado dispõem de vários dispositivos (*tablete*, *smartphone*, etc) e plataformas distintas (Apple IOS, Android, Windows e etc). Cada plataforma exige um cliente móvel desenvolvido especificamente para ela. Todavia, as características e desafios são semelhantes: dispositivos com pouca capacidade de armazenamento local; acesso à Internet com conexão que varia em latência, largura de banda, taxa de ruído e rede de pacotes.

No quesito enlace, o dispositivo pode estar conectado via WiFi com enlace de 100 Mbps durante 10 minutos e passar a conexão via rede móvel a 128 Kbps no minuto seguinte. Na questão armazenamento em dispositivos móveis, há a limitação de espaço para armazenamento, o que inviabiliza a sincronização com cópia total dos dados. Por outro lado, a arquitetura dos clientes móveis pode fazer uso do armazenamento local como *cache* e priorizar as operações de entrada e saída de forma direta com a nuvem, via protocolo HTTPS, por exemplo. Clientes utilizam-se da API para obter acesso ao armazenamento distribuído na nuvem, aplicando restrições como: tamanho máximo de arquivos; controle de versão; regras de nomeação e etc. Além dos serviços corriqueiros de todos clientes, os clientes móveis dispõem do serviço de cópia de segurança, uma necessidade bem específica desse tipo de dispositivo.

- **Serviço de sincronização:** A última forma de acesso disponível na camada de aplicação é comumente conhecida por clientes de sincronização de arquivos. A exemplo dos demais componentes da camada de aplicação, o cliente de sincronização também está disponível para várias plataformas, porém os dispositivos foco de desenvolvimento são computadores pessoais. Estes clientes são destaques no consumo de dados em organizações e universidades (GONÇALVES, 2015).

O princípio básico de funcionamento está na sincronização de arquivos entre o cliente, um computador pessoal, e o servidor, um fornecedor de armazenamento na nuvem. Ele permite que o acesso aos arquivos seja efetuado mesmo que o cliente esteja temporariamente sem conexão com à Internet. Todavia ele consome espaço em disco local de cada cliente, efetuando cópia na íntegra dos arquivos selecionados para sincronização. A arquitetura comum desses clientes de sincronização de arquivos é constituída por quatro componentes (Figura 3.2):

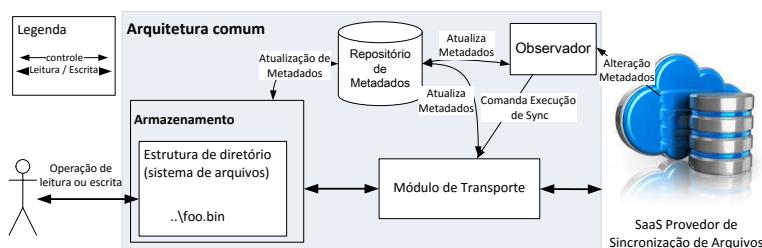


Figura 3.2: Arquitetura comum do cliente de sincronização de arquivos, com fluxo de envio e recebimento para a nuvem

- **Repositório de metadados:** Local de armazenamento persistente dos metadados. As informações que compõem a estru-

tura do metadado variam de fornecedor para fornecedor, porém é comum um metadado ter informações como: endereço dos dados; datas de: acesso; criação e modificação; autoria; estado e outros. Os metadados contêm informações sobre todos os objetos locais ou remotos e seus estados de sincronização.

- **Módulo de Transporte:** Responsável pela transferência dos dados entre o cliente de sincronização, que se encontra executando na máquina local do usuário, e o serviço de nuvem, onde o dado é efetivamente armazenado e consolidado. Esta transferência pode ser implementada por protocolos específicos de cada fornecedor, porém a unidade de transferência é o *chunk*. A transferência ocorre nos dois sentidos, atualizando tanto o disco local da máquina cliente, quanto o disco remoto do serviço de armazenamento. Para que a sincronização tenha êxito, o disco local e o remoto devem obrigatoriamente dispor de espaço para todo o conteúdo sincronizado.
- **Observador:** Responsável pelo controle e sincronização das modificações efetuadas tanto nos clientes quanto na nuvem. Ele identifica as divergências ocorridas no intervalo de tempo entre duas solicitações de sincronização. O intervalo pode ser regular, automaticamente invocado pelo cliente de sincronização quando a máquina cliente estiver continuamente ligada à Internet ou esporádico, realizado no momento da identificação da conexão. Havendo divergência nas informações, ele notifica o módulo de transporte para que se realizem as transferências de dados e atualiza o repositório de metadados local, conforme a necessidade.
- **Armazenamento:** Responsável pelo armazenamento dos dados no disco local, cabe a esse componente a organização da

estrutura de diretório e integração transparente com o sistema de arquivos local. O usuário acessa somente os arquivos e diretórios que se encontram no disco local, ou seja, uma réplica dos dados armazenados no serviço de acesso a arquivos em nuvem.

Na configuração do cliente pelo usuário, é estabelecido o vínculo entre a conta de usuário e os seus dados armazenados. Os procedimentos de autenticação e vinculação permitem que o usuário determine um diretório vazio no disco local. Este diretório será o espelho dos dados contidos na nuvem no computador onde o cliente está instalado. O componente observador é que mantém o repositório de metadados local em sincronização com o serviço de armazenamento distribuído da nuvem. Ele verifica se o conteúdo do diretório está idêntico ao que consta no repositório de metadados local. Esse processo se dá através da varredura de todos os arquivos do diretório espelho. Nesta varredura é analisado o *hash* de cada arquivo, que é comparado com o repositório de metadados, identificando o que precisa ser sincronizado.

Os componentes observador e módulo de transporte interagem até que não existam diferenças de conteúdo entre o diretório vinculado e os dados constantes no serviço de armazenamento de arquivos em nuvem. Este processo é bidirecional, ora o módulo de transporte necessita enviar informação do diretório local para nuvem, ora recebe os dados da nuvem para o diretório local. A interação pode ser melhor descrita pelo entendimento de três processos: inicialização; alteração e desvinculação descritos a seguir:

- **Inicialização:** Processo que vincula o cliente de sincronização a uma credencial e a um diretório base. Esse diretório será mantido em sincronia com a nuvem. O processo de inicializa-

ção se dá pela validação de uma credencial junto ao serviço de autenticação na nuvem e, após tal validação, o usuário faz a escolha de um diretório no sistema de arquivos local. Ao término deste processo fica estabelecido ao cliente de sincronização um vínculo de sincronia entre o diretório base e a estrutura de dados na nuvem.

- **Alterações:** O processo de alterações pode ser iniciado pela criação, remoção ou modificação de arquivos ou diretórios da estrutura vinculada. As alterações podem acontecer tanto na nuvem, quanto no disco local. O módulo observador é o responsável pela detecção de alteração (BHAGWAT, 2009) e acionamento do módulo de transporte. O módulo de transporte executará uma lista de tarefas (lista de atividades de recebimento ou envio). A lista de tarefas pode conter operações de alterações dos atributos de arquivos ou diretórios, ou a necessidade de transporte de dados de arquivos modificados. As modificações de dados de arquivo se dão por *chunk*. As tarefas podem ser de envio e ou de recebimento dos dados de cada *chunk*. A responsabilidade por alterações nos metadados da nuvem é do serviço de armazenamento distribuído e as alterações de metadados locais são do observador.

A remoção de arquivos ou diretórios é detectada da mesma forma pelo observador. O observador instrui o módulo de transporte para que envie o comando de remoção do arquivo ou diretório no disco local ou na nuvem, conforme necessidade.

- **Desvinculação:** A desvinculação de um cliente é apenas o processo de remoção das informações sobre credenciais. O diretório base não é removido, ficando sua remoção a encargo do usuário. Um cliente de sincronização desvinculado, desativa o componente observador.

O funcionamento básico dos clientes de sincronização de arquivo foi apresentado, porém os clientes disponíveis atualmente possuem outras características que serão discutidas mais adiante. Entre as funcionalidades disponíveis aos clientes de sincronização *DropBox*, *Google Drive* e *OneDrive* pode se elencar as seguintes técnicas:

- **Sincronização na LAN:** É uma estratégia de redução da latência de sincronização, ao mesmo tempo que reduz o tráfego do enlace de Internet. Um cliente de sincronização que utilize essa técnica precisa estar na mesma sub-rede em uma rede local. Transformando os clientes de sincronização desta sub-rede em repositório temporário para as demais estações que compartilhem os mesmos diretórios. As estações realizarão a transferência dos dados a partir do repositório temporário, recebendo arquivos através de uma rede ponto a ponto entre o cliente que já tem os dados sincronizados e o cliente que deseja os dados do arquivo (GONÇALVES, 2015).
- **Sincronização incremental:** Tem os mesmos objetivos de Sincronização na LAN, porém a técnica envia ou recebe da nuvem somente os dados dos *chunks* modificados do arquivo (DRAGO, 2013).
- **Compressão:** Com os mesmos objetivos das técnicas anteriores, na compressão os *chunks* são compactados antes da transferência, reduzindo o volume e o tempo necessário para transmissão (DRAGO, 2013; BESEN, 2015).
- **Sincronização por streaming**<sup>1</sup>: Objetiva reduzir o tempo total de sincronização de arquivos com múltiplos *chunk* a serem

---

<sup>1</sup> <https://www.dropbox.com/business/sync-performance>

sincronizados entre múltiplos clientes. A técnica é permitir o recebimento dos *chunks* por outros clientes mesmo antes do envio completo do arquivo. Desta forma a sincronização é realizada a cada *chunk*.

- **Criptografia:** Nesta técnica o foco é a segurança dos dados, garantindo a confidencialidade da informação. Os dados trafegados são criptografados antes da transmissão. A criptografia SSL é utilizada no protocolo de envio e recebimento HTTPS (DRAGO, 2013). Além disso a técnica pode ser utilizada em conjunção com a compressão dos dados, compactando e criptografando cada *chunk* com o certificado emitido ao cliente de sincronização.

Nem todas as características descritas estão disponíveis em todos os clientes de sincronização. Uma tabela com as características disponíveis em cada cliente estudado é apresentada na Seção 3.4. A arquitetura comum dos clientes de sincronização impõe barreiras no seu uso por algumas organizações. São elas: ausência de integração com sistemas de armazenamento privado; necessidade de cópia local de todos os arquivos compartilhados e latência de sincronização entre arquivos da nuvem e repositórios locais:

- **Ausência de integração com sistemas de armazenamento privado:** Aplicações legadas se utilizam do controle de *lock* no NAS como forma de garantir o acesso ao mesmo arquivo por múltiplos usuários, permitindo a colaboração de forma síncrona. Os clientes de sincronização se utilizam do disco local (HOUSTON; FERDOWSI, 2014) (BESEN, 2015), onde o *lock* da aplicação legada não pode ser detectado pelos demais clientes que estão colaborando.

- **Necessidade de cópia local de todos os arquivos compartilhados:** O conteúdo compartilhado na nuvem é sincronizado nos discos locais de cada cliente. Para que a sincronização ocorra, é necessário que a quantidade de dados envolvida seja inferior ao menor espaço de armazenamento, ou seja, ela está limitada em função do cliente ou em função da nuvem. No contexto das organizações, normalmente, essa limitação de capacidade encontra-se nas estações de trabalho. Neste ambiente é comum estações de trabalho que possuem unidades de armazenamento na ordem de centenas de gigabytes, enquanto o armazenamento em nuvem está na ordem de centenas de terabytes (KATZER; CRAWFORD, 2013). Os clientes de sincronização permitem ao usuário definir, se necessário, quais diretórios disponíveis na nuvem serão sincronizados em um determinado cliente, gerando réplicas locais parciais em função de um contexto. Todavia, esse processo é manual e fere o princípio de NIST quanto ao amplo acesso à informação.
- **Latência de sincronização entre arquivos da nuvem e repositórios locais:** As organizações que compartilham o mesmo ponto de acesso à Internet por diversos usuários em uma mesma localidade, porém em redes locais distintas, estão sujeitas a impactos oriundos da arquitetura. A arquitetura, conforme apresentada, converge as operações E/S para o serviço da nuvem. Usuários que compartilham as mesmas informações irão receber atualizações dos dados sempre que estes forem modificados. Os *chunks* modificados serão enviados à nuvem pelo cliente que alterou o arquivo ou diretório. Os demais clientes, cada qual no seu momento oportuno, atualizam suas réplicas locais, utilizando-se do mesmo enlace de Internet. Portanto, um mesmo conjunto de *chunks* é trafegado entre os clientes e a nuvem várias vezes pelo mesmo enlace

de Internet, caracterizando um desperdício deste recurso. A velocidade desta atualização vai depender da velocidade disponível do enlace de Internet bem como o tamanho do grupo sobre o mesmo enlace de Internet.

Os servidores NAS são implementados em rede local, com latência inferior a 1ms, permitindo que as operações E/S sejam realizadas de forma síncrona. A escolha do armazenamento na nuvem para cenários atendidos atualmente por NAS implicaria no aumento da latência ao acesso de arquivos compartilhados entre usuários de uma mesma localidade. Características como o modelo de sincronização explorado pelos clientes, concentração de usuários com os mesmos documentos compartilhados e existência de canais dedicados faz com que as ferramentas de sincronização em nuvem, para organizações, tenham percentuais de adoção inferiores aos observados com usuários domésticos (SMITH, 2013).

Os clientes de sincronização de arquivos diferem de provedor para provedor. No contexto das organizações, onde os dados são armazenados em NAS, a transferência se restringe à rede local. No entanto, seguem uma arquitetura que não atende às necessidades dos usuários das organizações.

### 3.2.3 Comparação dos Serviços da Camada de Aplicação

No ambiente das organizações, o uso de sistema de arquivos distribuídos é essencial. Todavia, na maioria dos casos, o usuário deseja incrementar as funcionalidades sem abdicar de outras já disponíveis. Nesta seção, busca-se identificar as funcionalidades que um usuário de organização deseja em um serviço de armazenamento de nuvem. A lista de funcionalidades não engloba todos os pontos, mas serve ao objeto deste trabalho. São elas:

- **Acesso transparente:** Para o usuário, o acesso transparente é a possibilidade de utilizar os aplicativos, ler e salvar arquivos, com as ferramentas de sistema operacional. Tecnicamente o acesso transparente implica que o sistema operacional realize operações de entrada e saída da mesma forma que o faz com um disco local ou um compartilhamento de rede. Neste caso as operações de leitura e escrita são mediadas pelo VFS (TANENBAUM; WOODHULL, 2009).
- **Edição colaborativa:** É a edição de um mesmo documento por múltiplos usuários, permitindo a construção colaborativa do documento. Em nuvem, a edição colaborativa é provida pela camada de aplicação com editores de texto e planilhas. Exemplo desses aplicativos são as plataformas Google Docs e Office 365 que são integradas aos seus respectivos serviços de armazenamento.
- **Acesso desconectado:** Acesso aos arquivos mesmo sem conectividade com a Internet e posterior sincronização.
- **Recuperação e versionamento:** O usuário obtém através destas funcionalidades a capacidade de acompanhar e recuperar diferentes versões de um mesmo documento. Protege o usuário de remoções acidentais e o rastreamento de alterações em um documento. Esta funcionalidade é disponibilizada no serviço de nuvem na console web (HOUSTON; FERDOWSI, 2014) (BESEN, 2015). O versionamento está relacionado ao sistema de armazenamento distribuído. Este trabalha com *chunk* de forma incremental. As alterações são mantidas, escrevendo no repositório de metadados a trilha de alterações.
- **Desenvolvimento de aplicações:** É a possibilidade de criar integrações para gravação e acesso aos dados. Esta funcionalidade é

o componente Aplicação da camada de aplicação, Figura 3.1 chamada de API e os provedores disponibilizam ferramentas de desenvolvimento que possibilitam os desenvolvedores de criarem aplicações novas.

- **Otimização de transferência:** Funcionalidades que reduzam o consumo de recursos do enlace de Internet ou menor tempo de sincronização. O conjunto de funcionalidades de otimização da transferência pode ser apresentado de diferentes maneiras: sincronização na LAN; sincronização incremental; sincronização por *streaming* e a compressão.

Os modos de acesso aos serviços analisados foram: console web, aplicação (API), cliente móvel (um exemplo popular de aplicação) e cliente de sincronização. Sob a ótica das características especificadas, gerou-se a Tabela 3.1 onde os modos de acesso são comparados com as características desejadas por usuários das organizações.

Tabela 3.1: Comparação das características da camada de aplicação, do serviço de armazenamento, acesso a arquivos em nuvem

Características	Console web	Aplicação (API)	Cliente Móvel	Serviço de sincronização
Acesso transparente	não	sim	não	Local
Edição colaborativa	Na nuvem	sim	não	não
Acesso desconectado	não	sim	Cache	Local
Recuperação e versões	Na Nuvem	sim	não	não
Desenvolvimento	não	sim	não	não
Otimização de transferência	não	sim	não	sim

A única forma de acesso que atende a todas as características é a aplicação (API) isto se dá exatamente por sua natureza. Por ser uma biblioteca, ela não fornece um serviço ou aplicação e sim uma plataforma de desenvolvimento que dispõe de funções que permitem manipular os dados e o acesso ao serviço. Um exemplo de aplicação que tem base na plataforma de desenvolvimento são os aplicativos de clientes móveis. Embora possam utilizar estas funções, os clientes móveis são projetados

para dispositivos com recursos limitados como *smartphones* e *tablets*, e não atendem a nenhuma característica completamente. A única exceção é o acesso desconectado que é possível com a intervenção manual do usuário e ainda limitado ao espaço do dispositivo. Os modos console web e cliente de sincronização possuem características semelhantes. A console web se destaca atendendo às características de edição colaborativa e recuperação de versões, mas não permite acesso transparente e nem otimização de transferência.

Todos os modelos de acesso podem ser usados simultaneamente e, certamente, são complementares. A escolha entre um e outro depende das circunstâncias que o usuário está no momento. Da mesma forma que a escolha do provedor do serviço é feita em função de características técnico comerciais.

### 3.3 Provedores do Serviço de Acesso a Arquivos em Nuvem

*DropBox*, *Google Drive* e *OneDrive* estão entre os provedores de serviço mais populares de acesso a arquivos na nuvem (DRAGO, 2013), correspondendo em média a um terço dos acessos ao *Youtube* (DRAGO, 2012). Além de populares esses clientes servem ao propósito deste trabalho por terem documentações públicas de suas arquiteturas (HOUSTON; FERDOWSI, 2014) e (BESEN, 2015). Ainda esses provedores de serviço seguem a arquitetura comum apresentada no início deste capítulo e possuem todos os serviços da camada de aplicação.

#### 3.3.1 DropBox

O *DropBox* é o nome de um serviço de armazenamento de arquivos em nuvem, da empresa *DropBox, Inc.* O serviço de armazenamento iniciou a sua comercialização em 2008 e em maio de 2016 contava

com 500 milhões de usuários<sup>2</sup>. O *DropBox* apoia-se na plataforma PaaS da Amazon EC2 com armazenamento apenas no continente americano (DRAGO, 2012).

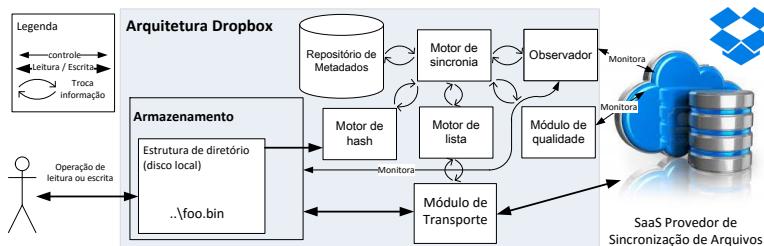


Figura 3.3: Arquitetura do cliente *DropBox* extraído de (HOUSTON; FERDOWSKI, 2014) e adaptada pelo autor

A arquitetura do *DropBox* estende os componentes da arquitetura comum. Possui os componentes repositório de metadados, observador e módulo de transporte. O repositório de metadados armazena as informações de arquivos e diretórios, tais como os *hashes* dos *chunks* que os compõem. O observador é responsável por verificar as modificações ocorridas localmente e na nuvem e o módulo de transporte se encarrega das transferências dos arquivos. Além dos componentes comuns, vistos anteriormente e detalhado na Seção 3.2, a arquitetura conta com os componentes motores de sincronia, *hash*, lista e um módulo de qualidade.

- **Motor de sincronia:** Responsável por centralizar as requisições de sincronização, interagindo com os motores de *hash*, lista e qualidade, bem como com o repositório de metadados e observador. Esse motor é acionado pelo observador sob demanda de alteração de um arquivo, local ou na nuvem. Cabe a ele o processamento da solicitação e o encaminhamento dela aos demais componentes

<sup>2</sup> <https://www.dropbox.com/about>

necessários. Ele também é responsável pela atualização do repositório de metadados.

A etapa de sincronização é encerrada após a verificação do conjunto de *hashes*, individuais de cada *chunks*, e do *hash* total do arquivo. A verificação é feita através da comparação dos *hashes* gerados pelo motor de *hash* e verificado pelo motor de qualidade.

- **Motor de *hash*:** Responsável pelo cálculo do *hashes* dos arquivos e diretórios. É acionado pelo motor de sincronização que passa como parâmetro o caminho de um arquivo ou diretório.
- **Motor de lista:** Responsável por gerar uma lista de tarefas para execução. O motor de sincronização interage com o motor de lista, passando a lista de *chunk* e operações que precisam ser realizadas. O motor de lista verifica a interdependência dessas operações, passando ao módulo de transporte apenas as tarefas que precisam ser executadas e na ordem que devem ser executadas.

Algumas operações precisam de uma sequência determinada. Um arquivo só pode ser sincronizado para um diretório uma vez que o diretório exista. Assim, a criação do diretório precede a operação com o arquivo. No *DropBox*, a sincronização de um único arquivo pode ocasionar inúmeras operações no módulo de transporte, sendo uma para cada *chunk*. Operações como a remoção de um diretório podem cancelar tarefas pendentes no motor de lista.

- **Módulo de qualidade:** Responsável por garantir integridade dos dados transferidos, tanto no sentido cliente-nuvem quanto nuvem-cliente. Cabe ao motor de qualidade a monitoração dos *chunks* envolvidos e a revalidação dos *hashes*, garantindo a integridade dos

dados.

O armazenamento distribuído calcula o *hash* de cada arquivo e diretório existente e os disponibiliza no repositório de metadados da nuvem. O módulo de qualidade compara as informações de *hash* da nuvem com o cálculo obtido pelo motor de *hash*.

O cliente *DropBox* otimiza o uso do enlace à Internet utilizando-se da técnica de deduplicação na origem. A técnica é aplicada no cliente, fazendo que a transferência de *chunks* entre cliente e nuvem restrinja-se a *chunks* únicos (DRAGO, 2012). Exemplificando: Se um usuário A faz *upload* para nuvem de um artigo no formato PDF e um usuário B faz *upload* posterior deste mesmo PDF o usuário B não vai enviar este PDF para a nuvem, pois ele já existe no sistema de armazenamento distribuído. Os *chunks* trafegados entre o cliente e a nuvem são criptografados e comprimidos garantindo segurança e reduzindo o tráfego, respectivamente (DRAGO, 2012).

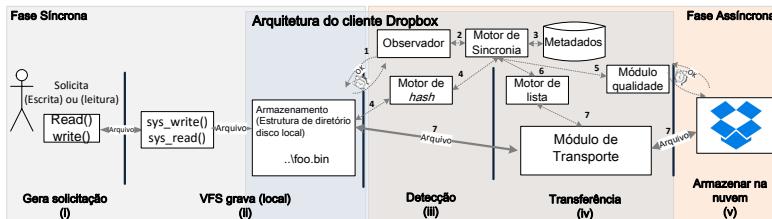


Figura 3.4: Fluxo de escrita ou leitura do *DropBox* Fonte: (HOUSTON; FERDOWSKI, 2014) e Adaptado pelo autor.

A execução do cliente de sincronização *DropBox* resume-se a disponibilizar operações sobre arquivos e diretórios. As operações de leitura e escrita são as operações mais complexas. Na Figura 3.4, ilustra-se o fluxo de um processo de escrita do *DropBox* até que um arquivo che-

gue à nuvem. O processo completo é composto por cinco etapas, (i) gera solicitação; (ii) VFS grava local; (iii) detecção; (iv) transferência; (v) armazenar na nuvem. As etapas (i) e (ii) são executadas internamente ao sistema operacional e não fazem parte intrinsecamente da arquitetura do *DropBox*, apenas o local de gravação passa a fazer parte da arquitetura. Na etapa (iii), detecção, o cliente percebe modificações e altera o sistema de arquivo local. A etapa (iv), transferência, é dedicada a transferir os dados do arquivo. A etapa (v), por fim, é a gravação do armazenamento local para o armazenamento na nuvem.

Considerando o esquema ilustrado na Figura 3.4, os componentes de arquitetura do cliente *DropBox* são invocados nas etapas (iii), detecção, e (iv), transferência. O passo 1 é executado periodicamente pelo observador. Caso não identifique nenhuma alteração, ele programa sua inspeção para o próximo período e aguarda. Se identificar modificação, ele aciona o motor de sincronia (passo 2), passando como parâmetro a localização do arquivo ou diretório observado. O motor de sincronia com as informações recebidas no passo 4 envia a lista de *chunks* e identificadores ao módulo de qualidade (passo 5). O módulo de qualidade verifica a lista de identificadores únicos junto ao servidor de armazenamento em nuvem e descobre quais *chunks* precisam ser enviados, repassando esta informação ao motor de sincronia.

O módulo de qualidade continua monitorando o armazenamento em nuvem durante as etapas (iv) e (v) retornando as informações ao motor de sincronia até que nenhum dado esteja pendente. O motor de sincronia repassa para o motor de lista (passo 5) informações de *chunks* pendentes com os respectivos endereços. O módulo de lista envia e orquestra a execução paralelizando os trabalhos do módulo de transporte como tarefas (passo 7) (HOUSTON; FERDOWSI, 2014).

O fluxo anterior segue a sincronização cliente-nuvem, as etapas

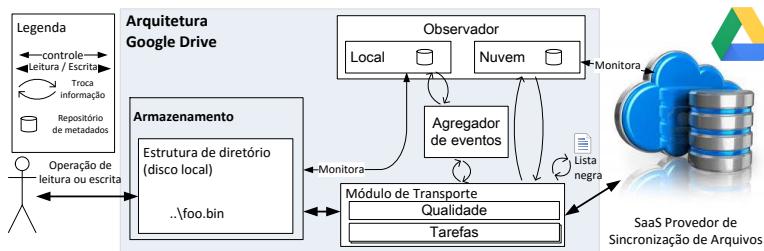


Figura 3.5: Arquitetura do cliente Google Drive. Fonte: (BESEN, 2015) Adaptada pelo autor.

(v) a (i) são invertidas no fluxo nuvem-cliente, com exceção da etapa (iii) detecção, que acontece primeiro. O observador, em vez de detectar a mudança no disco local, detecta a mudança na nuvem. Pela natureza assíncrona do processo as etapas (i) e (ii) só podem acontecer se as etapas (iii),(v) e (iv) foram executadas previamente pelas execuções cíclicas do observador (HOUSTON; FERDOWSI, 2014).

### 3.3.2 Google Drive

*Google Drive* é o nome de um serviço de armazenamento de arquivos em nuvem, da empresa Alphabet. O serviço de armazenamento iniciou em 2006 e conta com mais de 1 bilhão de usuários<sup>3</sup>. O *Google Drive* é parte da plataforma PaaS da Google. *DropBox* e *Google Drive* estende a arquitetura comum apresentada na Seção 3.2.

A arquitetura do *Google Drive* apresentada na Figura 3.5 possui os mesmos componentes que a arquitetura comum: armazenamento, observador e módulo de transporte. Entretanto, na visão da arquitetura do Google, o módulo de transporte é subdividido em local e remoto, assim como o módulo de transporte incorpora submódulos de nome, qualidade

<sup>3</sup> <https://abc.xyz/>

e tarefas. Em adição à arquitetura comum, o *Google Drive* conta com um agregador de eventos (BESEN, 2015), descrito a seguir:

**Agregador de eventos:** O agregador de eventos é responsável por gerar uma lista de tarefas para execução pelo módulo de transporte. O observador interage com o agregador de eventos passando a lista de arquivos e operações que precisam ser executadas (localmente ou na nuvem). O agregador de eventos verifica a interdependência destas operações, passando ao módulo de transporte apenas as tarefas que precisam ser executadas e na ordem que devem ser executadas. Um exemplo de operação que precisa da correta sequência é que um arquivo só pode ser copiado para um diretório, uma vez que o diretório exista. Desta forma a criação do diretório precisa ser enviada ao módulo de transporte antes da cópia do arquivo. Operações como a exclusão de um diretório, podem cancelar a sincronização de um ou mais arquivos pendentes no agregador de eventos.

Na estratégia do *Google Drive* existe um observador local e um observador de nuvem e a base de metadados é referenciada como um grafo, existindo um para cada contexto (nuvem e local). O objetivo do cliente *Google Drive* não difere dos demais. É um cliente de sincronização com o serviço de acesso a arquivos na nuvem e as operações de leitura e escrita passam por seus componentes, conforme ilustrado na Figura 3.6. A escrita ou leitura de um arquivo ou diretório entre o *Google Drive* e a nuvem é composta por cinco etapas, (i) gera solicitação; (ii) VFS grava local; (iii) detecção; (iv) transferência; (v) armazenar na nuvem. As etapas (i) e (ii) são executadas internamente ao sistema operacional e nos casos de leitura dependem da existência prévia do arquivo obtido pelo processo de sincronização. O local de gravação faz parte da arquitetura do *Google Drive*. Na etapa (iii) detecção, o cliente *Google Drive* percebe modificações no sistema de arquivos local ou da nuvem. Na etapa (iv) transferência que é dedicada a transferir os dados do arquivo. A etapa (v), por fim, é a gravação ou leitura no armazenamento na nuvem.



Figura 3.6: Fluxo de escrita ou leitura *Google Drive* fonte: (BESEN, 2015) e adaptado pelo autor

Detectada a criação de um novo arquivo pelo observador local (passo 1), ele cria um arquivo cópia de nome N para N+1 em um diretório temporário. Após isso, remove o arquivo N, original e move o arquivo N+1 para N, passando como parâmetro ao agregador de eventos (passo 2). O agregador de eventos faz o processo inverso, esta técnica é descrita pelo Google em sua patente (BESEN, 2015), batizada como "bloquear e combinar". O objetivo principal da técnica é permitir que o módulo de transporte tenha uma cópia íntegra em um determinado momento no tempo, enquanto o arquivo será sincronizado com a nuvem. Ao mesmo tempo, o arquivo original é liberado para que continue sendo utilizado (escrito e lido), pelo sistema operacional.

O agregador de eventos passa como parâmetro ao módulo de transporte (passo 3) a localização do arquivo temporário. Além disso, o agregador de eventos verifica se existe algum evento pendente em fila que invalide outros eventos, tal como a remoção de um diretório, que invalidará os objetos abaixo dele. O módulo de transporte paralleliza o envio dos arquivos, quebrando-os em *chunks* (passo 4).

A cópia de um arquivo da nuvem para o local segue a seguinte ordem das etapas: (iii) detecção, (iv) transferência. O observador detecta a alteração na nuvem (passo 1), enviado os endereços de sincroniza-

ção, que precisam ser copiados, ao agregador de eventos (passo 2). O agregador de eventos envia a lista ao módulo de transporte (passo 3). O módulo de transporte realiza a cópia dos arquivos diretamente para o diretório local (passo 4).

### 3.3.3 OneDrive

O *OneDrive* é o nome do serviço de armazenamento de arquivo em nuvem da empresa Microsoft. O *OneDrive* surgiu em 2008 com o nome de SkyDrive, em 2014 migrou para o nome atual. A Microsoft conta com 1,2 bilhões de possíveis usuários de sua plataforma Office, que inclui o *OneDrive*<sup>4</sup>. A patente (WAUTIER, 2015) descreve um ambiente de sincronização de arquivos em nuvem, porém foca no caso de uso do usuário doméstico com múltiplos dispositivos e como contornar os conflitos de sincronização na nuvem, apresentando de forma macro a arquitetura da nuvem e do cliente, impossibilitando a obtenção de detalhes de sua arquitetura.

A patente (YANG, 2013) trata da interface do usuário para serviço de armazenamento externo, focando na integração do *OneDrive* com o Windows no que tange a aparência e localização de ícones e locais de salvamento para plataforma Office. Além disso, os documentos de desenvolvimento da camada de aplicação (API)<sup>5</sup> indicam as funcionalidades disponíveis pelo serviço, que não se destacam em relação a arquitetura comum. Os métodos existentes nas API e as funcionalidades observáveis colocam o cliente *OneDrive* como um cliente básico, tal qual apresentado na Seção 3.2.2.

## 3.4 Comparação de Características e Funcionalidades

Esta seção tem por objetivo comparar os três provedores *DropBox*, *Google Drive* e *OneDrive* descritos na Seção 3.3. As características es-

<sup>4</sup> <http://news.microsoft.com/bythenumbers/>

<sup>5</sup> <https://dev.onedrive.com/>

colhidas e agrupadas na Tabela 3.2 não são exaustivas, mas são as principais identificadas nos estudos e que permitem entender as diferenças entre os provedores de serviço.

Tabela 3.2: Comparação de características dos principais serviços de sincronização de arquivos.

Item	Característica	Dropbox	Google Drive	OneDrive
1	Sincronização na LAN (Lansync)	Sim	Não	Não
2	Sincronização incremental.	Sim	Não	Não
3	Deduplicação do lado do cliente	Sim	Não	Não
4	Sincronização por Streaming	Sim	Não	Não
5	Compressão	Sim	Sim	Não
6	Criptografia	Sim	Não	Não
7	Sistemas Operacionais suportados	Windows, Mac, Linux, Android, iOS, Blackberry, Kindle Fire	Windows, Mac, Android, iOS	Windows, Mac, Android, iOS, Blackberry
8	Versionamento	Sim	Sim	Não
9	Edição de documento de forma colaborativa	Não	Sim	Sim
10	Quota de arquivos compartilhados é única?	Não	Sim	Sim
11	Sistema de arquivo distribuído	S3FS	GFS	Hadoop FS

Os itens de 1 a 7 da Tabela 3.2 são características relativas ao cliente e de 8 a 11 ao serviço de nuvem. As características de 1 a 5 formam um conjunto que objetiva minimizar o tráfego com a nuvem e maximizar a velocidade de sincronização entre os clientes.

**DropBox:** O *DropBox* é o cliente que atende ao maior número de características. Ele possui todas as características de otimização de tráfego: sincronização na LAN (Lansync); sincronização incremental; deduplicação do lado do cliente; sincronização por streaming e compressão. Além disso o *DropBox* possui a característica de criptografia que confere segurança. O *DropBox* suporta 7 sistemas operacionais: Windows, Mac, Linux, Android, iOS, Blackberry, Kindle Fire. O Versionamento está disponível via console web, mas o *DropBox* não conta com a edição colaborativa de documentos e a quota é contabilizada em duplicidade. A plataforma PaaS que suporta é a da Amazon EC2 (DRAGO, 2012) e consequentemente o sistema de arquivos distribuído é o S3FS (MUNISWAMY-REDDY, 2010). Uma questão interessante é que o *DropBox* mantém o

armazenamento dos dados unicamente nos Estados Unidos (DRAGO, 2013).

**Google Drive:** É um cliente de sincronização intermediário se comparado aos outros dois. A única característica de otimização de tráfego que o *Google Drive* tem disponível é a compressão dos arquivos antes do envio. Esta disponível para apenas 4 sistemas operacionais: Windows, Mac, Android, iOS, entretanto a plataforma PaaS Google Cloud Service possui todas as funcionalidades: versionamento; edição colaborativa de documentos (*Google Docs*), quota compartilhada e o sistema de arquivos distribuído é o GFS, Google FileSystem (GHEMAWAT, 2003).

**OneDrive:** O *OneDrive* não conta com nenhum dos recursos de otimização de tráfego e nem de criptografia dos arquivos armazenados. Por este motivo o *OneDrive* é o cliente mais básico entre os três. Está disponível para 5 sistemas operacionais: Windows, Mac, Android, iOS, Blackberry. O versionamento do *OneDrive* é exclusivamente para arquivos identificados como Office e por isso considerado como não disponível. Ele se utiliza da plataforma PaaS da Microsoft Azure e conta com a edição online de documentos do Office 365 (KATZER; CRAWFORD, 2013) que é integrada. A quota é compartilhada entre os usuários e o sistema de arquivo distribuído é o Hadoop FS (SHVACHKO, 2010).

A arquitetura do *DropBox* apresenta um maior número de características disponíveis em relação aos concorrentes e demonstra uma preocupação com a otimização de tráfego do enlace de Internet. Sobre os pontos avaliados o *DropBox* apresenta o melhor cliente. *OneDrive* e *Google Drive* integraram com o serviço de armazenamento as plataformas Office 365 e *Google Docs*. Porém, as características dos clientes não se distanciam da arquitetura comum e todas as ferramentas de sincronização carregam a mesma limitação implícita na escolha de manter a sincronização de todo conteúdo da nuvem com o cliente. Esta arquitetura

proporciona tanto uma limitação de armazenamento em função do cliente, quanto um desperdício de recursos de rede ao sincronizar todos os arquivos independente do uso. Este último problema é agravado à medida que se aumenta a quantidade de usuários compartilhando um mesmo arquivo (GONÇALVES, 2016). Os pontos fracos desta arquitetura são motivadores para uma nova proposta de arquitetura.

### 3.5 Considerações Parciais

O serviço de armazenamento de acesso a arquivos na nuvem encontra uma diferenciação no uso relacionada a indivíduos de organizações, pois a utilização difere em densidade versus localização e tamanho dos grupos de colaboração. A arquitetura atual do serviço não reflete tal diferença e está organizada em blocos: uma camada de aplicação, um serviço de gerenciamento e um armazenamento distribuído interligados por uma API REST. A camada de aplicação é composta por uma console *web*, o serviço de aplicação e o serviço de sincronização, cujo aprimoramento da arquitetura é o foco deste trabalho. *DropBox* é o cliente de sincronização mais completo e mesmo assim não se afasta da arquitetura comum. Mas é preciso destacar que a camada de aplicação dos clientes *OneDrive* e *Google Drive* são mais completas. A arquitetura dos clientes limita o armazenamento em função do cliente e gera desperdício de recursos de rede ao sincronizar todos os arquivos independente do uso. Esses problemas são agravados à medida que se aumenta a quantidade de usuários compartilhando de um mesmo arquivo (GONÇALVES, 2016). Os pontos fracos da arquitetura do serviço de sincronização são motivadores para uma nova proposta de arquitetura, que atenda as necessidades das organizações.

## 4 CLIENTE HÍBRIDO PARA SINCRONIZAÇÃO DE ARQUIVOS NA NUVEM

Um dos principais componentes do armazenamento de arquivos na nuvem é o cliente de sincronização de arquivos (GONÇALVES, 2015). Nos capítulos anteriores apresentou-se as principais características desses clientes e formas de acesso a arquivos. Sob a ótica do uso de armazenamento nas organizações, os clientes atuais deixam a desejar, em um ou mais aspectos desse processo. Neste capítulo será apresentado o CHSAN, Cliente Híbrido para Sincronização de Arquivos na Nuvem, sua arquitetura e descrição de seus módulos.

### 4.1 Cenário de Execução

O cenário de nuvens computacionais, armazenamento e compartilhamento de arquivos, compreende dois atores principais: os repositórios de armazenamento e os clientes de sincronização. O repositório é usualmente composto por módulos para notificação, indexação e interface de operação, sendo implementado como um sistema de arquivos distribuído que assegure disponibilidade, integridade e confidencialidade. É comum a disponibilização de serviços para controle de versionamento, quotas de compartilhamento e edição colaborativa.

Por sua vez, um cliente de sincronização de arquivos é responsável por manter uma determinada estrutura de diretórios e de arquivos em sincronia com o repositório da nuvem. Em suma, o controle é realizado por módulos de sincronização e módulos observadores, que atuam sobre os metadados relacionados aos arquivos. Existem dois observadores (local e nuvem), responsáveis por identificar alterações nos arquivos. Quando necessário, o módulo de transporte realiza a transferência dos arquivos (parcial ou total) entre os repositórios. Relacionado a transferência

algumas funcionalidades comuns são encontradas em clientes de sincronização tais como: sincronização incremental, criptografia e compressão de dados. Essas estratégias de otimização são descritas na Seção 3.4 e têm como objetivo diminuir o volume de dados trafegados entre os repositórios de armazenamento e os clientes de sincronização.

Há alguns anos, clientes de sincronização de arquivos na nuvem estão difundidos entre usuários domésticos, atendendo satisfatoriamente suas necessidades. Recentemente, provedores de armazenamento na nuvem passaram a oferecer serviços voltados para organizações, que se resumem ao oferecimento de maior capacidade de armazenamento e melhor capacidade de gerenciamento que os pacotes oferecidos aos usuários domésticos (NALDI; MASTROENI, 2013). No entanto, ambientes organizacionais são constituídos por múltiplos usuários que colaboram em projetos comuns, muitas vezes em ambientes geograficamente distribuídos concentrados em redes privadas.

Nesse contexto colaborativo e distribuído, onde estão os usuários de organizações, uma alternativa viável de armazenamento remete ao armazenamento em nuvem computacional. O conteúdo compartilhado na nuvem é sincronizado nos discos locais de cada cliente. Para que a sincronização ocorra, é necessário que a quantidade de dados a ser copiada seja inferior à menor unidade de armazenamento envolvida: limitado em função do espaço de armazenamento local do cliente ou na nuvem. Nas organizações, essa limitação de capacidade encontra-se, na maioria das vezes, nas estações de trabalho. É comum a existência de organizações em que uma estação de trabalho possui unidades de armazenamento na ordem de centenas de gigabytes, enquanto o armazenamento em nuvem está na ordem de centenas de terabytes (KATZER; CRAWFORD, 2013).

Uma forma de minimizar esse problema encontrado pelos clientes de sincronização atuais é repassar a responsabilidade de escolha de

diretórios a serem sincronizados aos próprios usuários. Entretanto, com múltiplos arquivos e usuários, a elaboração e manutenção de uma hierarquia de compartilhamento seletiva não é uma tarefa trivial. Ainda, a sincronização seletiva, da forma como foi implementada pelos clientes de sincronização, fere o amplo acesso difundido pelo armazenamento em nuvem (MELL; GRANCE, 2009): o recurso permanece disponível no repositório de armazenamento, mas as funcionalidades de gerenciamento e utilização locais não podem ser aplicadas.

Finalmente, quanto à distribuição geográfica dos usuários, as organizações compartilham o mesmo ponto de acesso à Internet com diversos usuários internos. Em ambientes onde existe uma concentração de usuários, a utilização do serviço pode ser limitada em função da quantidade de dados a ser escrito na nuvem frente à quantidade de clientes que necessitam compartilhar a mesma informação. Nos ambientes organizacionais, onde o armazenamento é realizado sobre NAS, é comum que conjuntos de arquivos sejam compartilhados e, usualmente, existe um canal de comunicação dedicado para tráfego de informações sobre redes metropolitanas. A escolha do armazenamento na nuvem em substituição ao NAS implicaria o aumento da latência no acesso aos arquivos compartilhados entre usuários.

A junção dessas características (modelo de sincronização explorado pelos clientes, concentração de usuários com os mesmos documentos compartilhados e existência de canais dedicados) faz com que as ferramentas de sincronização em nuvem, nesses ambientes, tenham percentuais de adoção inferiores aos observados com usuários domésticos (SMITH, 2013). Portanto, verifica-se que a aplicação de clientes de sincronização desenvolvidos para usuários domésticos não atende às necessidades de ambientes organizacionais. As principais barreiras arquiteturais encontradas são: (i) ausência de integração com sistemas de armazenamento privado; (ii) necessidade de cópia local de todos os arquivos

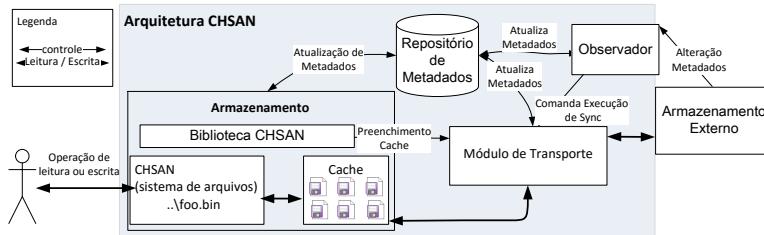


Figura 4.1: Arquitetura do cliente híbrido de sincronização de arquivos CHSAN.

compartilhados; e (iii) latência de sincronização entre arquivos da nuvem e repositórios locais.

## 4.2 Arquitetura do CHSAN

A arquitetura do Cliente Híbrido para Sincronização de Arquivos em Nuvem (CHSAN), resumida na Figura 4.1, está alinhada com os principais clientes de sincronização de arquivos na nuvem. Os módulos comuns em suas arquiteturas (armazenamento, transporte, observador e repositório de metadados) (HOUSTON; FERDOWSI, 2014) (BESEN, 2015) foram estendidos por CHSAN.

Ao módulo de armazenamento, incorporou-se um sistema de arquivos virtual e *cache* seletiva, enquanto o módulo de transporte foi adaptado para encaminhar dados a um armazenamento externo legado (além dos repositórios da nuvem). O sistema de arquivos virtual permite que o cliente tenha acesso à totalidade dos índices dos arquivos armazenados na nuvem, mesmo sem nunca tê-los sincronizado em sua integralidade. Assim, embora o volume de armazenamento disponível seja inferior ao total necessário, o cliente ainda tem conhecimento sobre a existência de todos os arquivos e informações referentes às alterações realizadas. Esse ponto diferencia CHSAN dos clientes tradicionais que informam a existência de arquivos somente quando sincronizados com o repositório

principal. A *cache* seletiva agiliza o acesso aos dados mais usados pelo cliente (atuando no requisito de latência de acesso). Dispositivos legados, como *Network Attached Storage* (NAS), recebem informações do módulo de transporte, como opção para minimização do uso do enlace à Internet e alternativa para comunicação em enlaces dedicados.

Para aplicativos finais, a existência de CHSAN é transparente (aplicativos utilizam a interface padrão do sistema operacional). Os módulos e o fluxo de dados são individualmente discutidos na sequência.

### 4.2.1 Armazenamento

O módulo de armazenamento do CHSAN é responsável pela apresentação do sistema de arquivos bem como pelas operações de leitura e gravação, combinando sistema de arquivos virtuais (VFS – *Virtual File System*) com FUSE<sup>1</sup> para criar um sistema de arquivos customizado em espaço de usuário, sem a necessidade de intervenção no núcleo. O módulo é constituído de: (i) espaço de armazenamento local; *cache*, alimentada de forma seletiva; (ii) biblioteca de acesso, que disponibiliza funções para consulta, escrita e leitura de arquivos; e (iii) sistema de arquivos CHSAN, que exporta para o sistema operacional uma estrutura de diretórios e arquivos como uma unidade de disco local. Tal exportação é análoga, do ponto de vista de funcionalidade e compatibilidade, com a estrutura de diretórios dos clientes de sincronização de arquivos em nuvem.

Na Figura 4.2, é representado o fluxo de uma operação submetida por um aplicativo usuário. Na parte inferior da Figura, modo núcleo, tem-se os módulos VFS e FUSE e, na parte superior, área do usuário, o aplicativo do cliente, a biblioteca *glibc* e o CHSAN. O fluxo é iniciado pelo aplicativo do cliente que gera a operação em um arquivo. Esta solicitação, ainda na área do usuário, utiliza a biblioteca padrão do sistema para acesso a arquivos locais, neste caso, *glibc*. Através desta biblioteca, a solicitação é encaminhada ao modo núcleo e processada pelos módulos

<sup>1</sup> FUSE, disponível em <<http://fuse.sourceforge.net/>>.

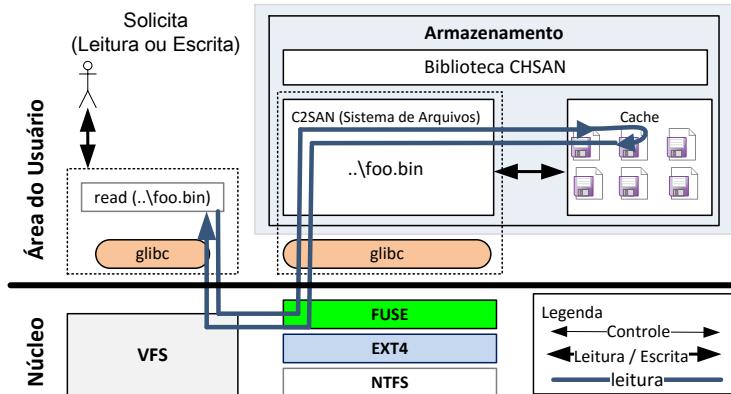


Figura 4.2: Visão detalhada do módulo de armazenamento CHSAN.

VFS e FUSE, assim como acontece com operações de outros sistemas de arquivos (e.g., EXT4 e NTFS). O módulo de armazenamento CHSAN, quando acionado, interage com o repositório de metadados por meio desta biblioteca e identifica a localização do arquivo solicitado, caso ele exista, ou provoca a criação de um novo índice no repositório de metadados. O fluxo de retorno do conteúdo somente é iniciado se o arquivo estiver na cache local. Caso contrário, o módulo de armazenamento aguarda até que o conteúdo seja copiado localmente. A transferência do conteúdo solicitado à cache local é responsabilidade do módulo de transporte auxiliado pelo módulo observador. O módulo de armazenamento desta arquitetura, pode ser alterado de forma que quando o armazenamento externo NAS esteja disponível, esse se utilize do NAS como cache local. Tal alteração pode permitir a edição colaborativa para aplicativos legados, bem como a sincronização por *streaming*, porém essa modificação é objeto de trabalho futuro.

### 4.2.2 Módulos Observador e Transporte

O módulo observador tem como objetivo a execução regular de tarefas assíncronas para atualizar a base local de metadados frente às alterações detectadas nos repositórios de armazenamento da nuvem. Essas atualizações são realizadas nos índices do sistema de arquivos do CHSAN, invalidando, quando necessário, as entradas na *cache*. Além disso, o módulo observador solicita o *upload* para o armazenamento externo (NAS ou repositório da nuvem) quando existe alguma entrada pendente no repositório de metadados. É importante ressaltar que CHSAN não foi concebido para ser utilizado sem o armazenamento em nuvem, uma vez que esse é base para a composição e manutenção dos metadados.

O módulo de transporte do CHSAN diferencia-se das arquiteturas tradicionais de sincronização, pois, uma vez parametrizado, pode interagir com diferentes repositórios externos (provedores de nuvem ou repositórios legados (NAS)). Cabe a ele a identificação do repositório com menor latência e a transferência dos arquivos remotos para a *cache* local. Assim, o usuário pode estar conectado na rede privada da organização e recuperar o arquivo de um repositório privado, ou estar conectado à Internet e recuperar o arquivo diretamente da nuvem. Dessa forma, o CHSAN minimiza o tráfego de dados para acesso aos arquivos compartilhados e não está limitado à capacidade de armazenamento do dispositivo local ou a restrições de seleções manuais de sincronização. Técnicas de otimização de transferência como as descritas na Seção 3.2.2, em especial sincronização incremental e compressão, podem ser implementadas no módulo de transporte, desde que o provedor de serviço disponibilize a respectiva API e métodos para execução da mesma.

### 4.2.3 Re却itório de Metadados

O re却itório de metadados gerido pelo módulo observador é o responsável pela indexação da totalidade dos arquivos compartilhados na nuvem. Cada arquivo possui somente um índice no re却itório de me-

tadados, mas pode estar replicado em *cache* local, sistema de armazenamento privado e na nuvem. Os metadados são concentrados em um banco de dados, contendo informações como identificadores, caminho do objeto (hierarquia), tipo do objeto (arquivo ou diretório), tamanho desse objeto, as datas de criação e alteração, bem como um controle de operações realizadas e pendentes. De acordo com a situação atual da informação sobre os objetos (sincronizado com a nuvem, criado ou pendente de sincronização), o módulo observador aciona a sincronização. Em suma, CHSAN baseia-se nos metadados para montar o sistema de arquivos virtual disponibilizado aos usuários, permitindo que o usuário visualize a totalidade do sistema de arquivos compartilhados, independente da localização física deste arquivo no momento da requisição.

#### 4.2.4 Operações

O CHSAN por ser uma arquitetura híbrida possui operações que ocorrem no contexto local e outras que acontecem no contexto de interação com o serviço de armazenamento externo, que pode ser em nuvem ou no NAS. Existem dois grupos de operações no módulo de armazenamento:(i) as operações que são escritas ou lidas somente no repositório de metadados (operações de banco de dados);(ii) e as que são escrita ou lidas no *cache* e consequentemente disco local (operações com arquivos).

Dentre as operações que são realizadas em banco de dados estão operações que envolvem apenas o índice, tais como criação e alteração de atributos tanto de diretórios quanto arquivos. Para criar um diretório ou um arquivo vazio o CHSAN apenas inclui uma entrada em banco de dados que fica pendente de sincronização. A alteração de atributos inclui a troca de nome, que segue a mesma lógica. As operações com arquivos só acontecem quando há demanda por leitura ou alteração dos dados contidos no arquivo. A Tabela 4.1 descreve as operações prototipadas no CHSAN e a abrangência das operações entre banco de dados;

Operações com arquivos locais (*cache*) e Armazenamento externo (Externo).

Tabela 4.1: Lista de operações e abrangência.

Função OS	Tipo	Descrição	Banco de dados	Cache	Externo
FindFiles()	Volume	Busca um arquivo ou diretório pelo nome	✓		
GetFileInformation()	Volume	Busca os atributos de um arquivo	✓		
Mount()	Volume	Monta o volume	✓		
Unmount()	Volume	Desmonta o volume	✓		
CreateDirectory()	Diretório	Cria um diretório	✓		✓
DeleteDirectory()	Diretório	Remove um diretório	✓		✓
OpenDirectory()	Diretório	Lê um diretório	✓		
CloseFile()	Arquivo	Fecha um arquivo			
CreateFile()	Arquivo	Cria arquivo	✓		✓
DeleteFile()	Arquivo	Remove um arquivo	✓	✓	✓
OpenFile()	Arquivo	Abre o arquivo	✓		
ReadFile()	Arquivo	Lê dados em um arquivo		✓	✓
SetFileAttributes()	Arquivo	Escreve os atributos de um arquivo	✓	✓	✓
WriteFile()	Arquivo	Escreve os dados em um arquivo	✓	✓	✓

Cada operação tem um tipo: volume, diretório ou arquivo. O tipo volume engloba operações cujo objeto é o sistema de arquivo virtual do CHSAN, obtendo informações tais como o espaço disponível e espaço utilizado na nuvem. Essas informações são obtidas da nuvem e armazenadas no banco de dados junto ao registro de metadados do diretório raiz. Somente os metadados são acessados em modo consulta para operações do tipo volume. As funções *FindFiles()* e *GetFileInformation()* servem tanto para diretórios quanto para arquivos, sendo que a *FindFiles()* localiza arquivos em um diretório pelo caminho e a função *GetFileInformation()* traz as informações de atributos de um determinado arquivo ou diretório. A função de *Mount()* conecta ao banco de dados e executa a função *FindFiles()* para varrer o diretório raiz, bem como a *GetFileInformation()* para obter as informações de cada objeto que forma o volume. Finalmente, a função *Unmount()* apenas desfaz a conexão com o banco de dados.

O segundo tipo, diretório, caracteriza-se pela execução completa em dois tempos. Assim que invocada, as operações deste tipo atualizam

o banco de dados de forma síncrona. O segundo passo, atualização do armazenamento externo, é feito de forma assíncrona quando identificado pelo observador. O observador solicita atualização nos armazenamentos externos (NAS e nuvem). As operações deste tipo são: *CreateDirectory()* e *DeleteDirectory()*. A operação *OpenDirectory()* somente verifica a existência do diretório no banco de dados local e retorna o sucesso ou falha da operação.

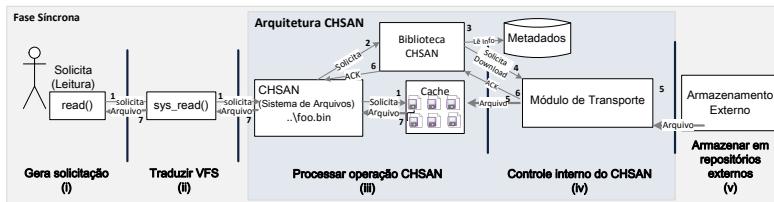
Finalmente, o terceiro e último tipo é o arquivo, em que se encontra o maior número de operações. As operações *OpenFile()* e *CloseFile()* atuam somente na base de dados, sincronamente. As operações *CreateFile()*, *DeleteFile()* e *SetFileAttributes()* executam parte de sua ação de forma síncrona no banco de dados e codificam uma pendência de atualização. Quando identificada essa pendência pelo observador, a segunda etapa é concretizada atualizando os repositórios *cache* e armazenamento externo (NAS e nuvem).

Embora as operações de leitura e escrita sejam tratadas de formas diferentes, elas seguem um fluxo similar de execução, dividido em cinco etapas: (i) gerar solicitação; (ii) traduzir VFS; (iii) processar operação CHSAN; (iv) controle interno do CHSAN; (v) armazenar em repositórios externos. A etapa (i) é executada na área do usuário e iniciada pela aplicação. No modo núcleo, a etapa (ii) recupera as informações da solicitação advindas da aplicação do usuário e traduz para VFS. No CHSAN, novamente no modo usuário, a etapa (iii) representa a operação do módulo de armazenamento da arquitetura. A etapa (iv) é de responsabilidade dos módulos observador e transporte, enquanto a etapa (v) é caracterizada pela sincronização dos dados entre a *cache* e os repositórios.

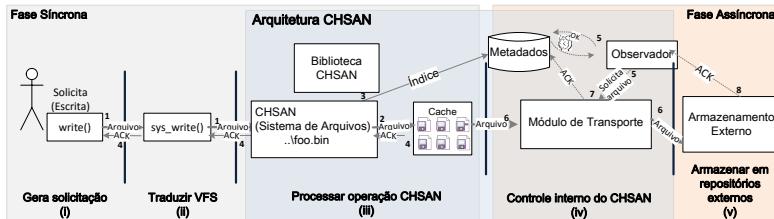
Na Figura 4.3, ilustra-se o fluxo de processamento das operações de leitura e escrita (representando em passos). As etapas (i)-(v) são representados pelo fluxo de execução (da esquerda para a direita).

Na Figura 4.3(a), tem-se o fluxo de uma operação de leitura, em que um aplicativo gera a solicitação e a envia (identificada no passo 1). Se o arquivo solicitado já existe na *cache* local e está coerente com a informação existente no índice de arquivos (tamanho e data de escrita são consultados no conjunto de metadados), ele é imediatamente retornado (passo 7) à aplicação. Neste caso, a resposta ao usuário é possível com o processamento das etapas (i) e (ii) simplesmente. Se o arquivo solicitado não estiver na *cache*, a etapa (iii) é iniciada. O sistema de arquivos do CHSAN solicita ao módulo de transporte o *download* do arquivo do repositório externo (servidor NAS ou repositório da nuvem). No módulo de transporte, os passos 2 (solicitação do arquivo à biblioteca CHSAN), passo 3 (leitura dos metadados) e passo 4 (efetua *download*) são processados. O fluxo segue na etapa (iv), desta vez, sob responsabilidade do módulo de transporte. A leitura do armazenamento NAS é prioritária em relação ao repositório na nuvem, portanto, se o arquivo estiver atualizado no NAS, os passos 5, 6 e 7 são executados e o arquivo entregue à aplicação. Finalmente, se o arquivo só estiver na nuvem, o módulo de transporte atualiza o arquivo no armazenamento NAS (passo 5), caracterizando a etapa (v), e o fluxo de retorno do arquivo é concluído (passos 6 e 7).

Enquanto o processo de leitura acontece de forma síncrona, o processo de escrita Figura 4.3(b) possui uma fase assíncrona. Uma escrita usando CHSAN ocorre inicialmente na *cache* da arquitetura. Os blocos de dados são redirecionados (passo 1) à *cache* local, passando pelo núcleo. O índice do arquivo é atualizado na base de metadados e uma confirmação é enviada (passo 4) ao sistema operacional, finalizando a operação. A fase assíncrona (passo 5) é executada pelo observador em intervalos regulares para verificação de pendências de sincronização na base de metadados. Quando necessário, o módulo de transporte é acionado para realizar o *upload* aos repositórios de armazenamento externo (NAS e provedor de nuvem). Ao final do passo 6, os armazenamentos externos e o metadado local são atualizados.



(a) Fluxo para leitura de um arquivo usando CHSAN.



(b) Fluxo para escrita de um arquivo usando CHSAN.

Figura 4.3: Exemplos de fluxo de leitura e escrita usando CHSAN.

A arquitetura apresentada bem como as operações descritas indicam que os objetivos de superar as limitações locais e de reduzir os impactos causados pelo desperdício dos enlaces de Internet podem ser alcançados com a implementação desta proposta. Além disso é possível implementar uma alteração no fluxo de escrita, tornando viável a edição colaborativa. Tal alteração consiste de que uma vez um CHSAN esteja conectado ao armazenamento externo NAS, este se utilize do NAS como cache. Essa opção compromete o acesso isolado, mas permite utilizar os controles do NAS para edição simultânea de arquivos, como já é feito atualmente por aplicações legadas, tais como bancos de dados em rede. Funcionalidade que será objeto de trabalho futuro.

### 4.3 Considerações Parciais

O armazenamento em nuvem dos provedores do tipo acesso a arquivos possuem barreiras para adoção de seus clientes de sincronização por organizações. Essas barreiras são: ausência de integração com NAS; necessidade de cópia local de todos os dados compartilhados e latência de sincronização. O CHSAN é uma proposta de arquitetura viável para superar essas barreiras. A arquitetura do CHSAN conta com um módulo de armazenamento que possibilita a montagem de um sistema de arquivos virtual na área de usuário capaz de viabilizar a navegação completa pelos dados contidos no armazenamento externo e sem a necessidade de sincronizar o conteúdo dos arquivos. A utilização de um *cache* local gerenciado habilita o CHSAN a manter o uso do espaço em disco dentro de um tamanho especificado. O módulo de transporte interage com o armazenamento externo, orquestrado pelo módulo observador, possibilitando reduzir a latência. O CHSAN reescreve o fluxo das operações sobre arquivos e diretórios, atuando em algumas destas fases síncrona e em outras assíncrona. Como estudo de caso, o protótipo CHSAN foi desenvolvido na plataforma Microsoft com o auxílio e inspiração de outros projetos. O próximo capítulo apresenta a análise experimental do protótipo desenvolvido. Embora prototipado com soluções Microsoft, CHSAN é agnóstico a plataforma.



## 5 EXPERIMENTAÇÃO E ANÁLISE DOS RESULTADOS

Nos capítulos anteriores, identificou-se que uma organização necessita de clientes de armazenamento na nuvem que ofereçam a integração entre o armazenamento de arquivos já existentes. CHSAN apresenta-se como esse cliente de armazenamento híbrido para nuvem, pois apoia-se nos conceitos dos clientes de armazenamento em nuvem e permite a integração com serviços legados. Como prova desse conceito, um protótipo da arquitetura foi desenvolvido e analisado, cujos resultados são apresentados neste capítulo.

### 5.1 Análise Qualitativa da Arquitetura CHSAN

Embora as arquiteturas dos clientes de sincronização na nuvem estudados apresentem diferenças, o presente trabalho estabeleceu um núcleo comum de componentes. Tal núcleo foi nomeado como arquitetura comum e servirá de base de comparação para a discussão conduzida nesta seção.

Frente aos componentes e módulos comuns que compõem os clientes, CHSAN não introduziu alterações no repositório de metadados e módulo observador. O diferencial do CHSAN está nas funcionalidades estendidas nos componentes de armazenamento e transporte. A Figura 5.1 permite a identificação desses componentes e comunicações relacionadas, com contorno pontilhado em vermelho. Como os demais componentes permanecem inalterados, a discussão ocorre sobre os componentes estendidos.

A Tabela 5.1 resume os quesitos selecionados para discussão qualitativa. Esses quesitos são oriundos do estudo das necessidades

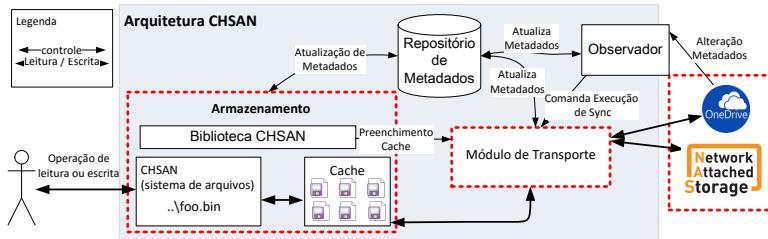


Figura 5.1: Operações em arquivos localizados no armazenamento externo (CHSAN vs. OneDrive)

de armazenamento das organizações descritos no Capítulo 3.2. Inicialmente, é possível observar que o CHSAN dispõe das mesmas peculiaridades dos clientes estudados, permitindo o acesso transparente aos arquivos.

Quanto a edição colaborativa, CHSAN não possui uma restrição arquitetural, ou seja, o módulo de armazenamento pode permitir a edição colaborativa não utilizando-se do cache ao trabalhar com arquivos em NAS. Por sua vez, o acesso desconectado na nuvem do CHSAN é mais abrangente, pois não é limitado aos recursos previamente sincronizados ou disponíveis na rede local, como ocorre em clientes tradicionais. Adicionalmente, CHSAN reconhece arquivos compartilhados via WAN.

A recuperação de versões também pode ser outra função do módulo de armazenamento na arquitetura CHSAN, pois este poderia apresentar em sua base de metadados as diferentes versões disponíveis a um arquivo. Quanto a otimização de transferência, a sincronização na LAN é aplicada no CHSAN no contexto do armazenamento NAS. Ainda, diferentemente dos clientes tradicionais, CHSAN permite a sincronização na LAN não só na mesma sub rede (mesmo domínio de *broadcast*), mas inclusive sobre redes roteadas (e WAN). Tal característica não é disponibilizada em nenhum dos clientes estudados. O *DropBox* com LAN Sync utiliza sincronização ponto a ponto somente em clientes no mesmo domínio.

Tabela 5.1: Comparação das características dos clientes de sincronização de acesso a arquivos em nuvem e CHSAN.

Características	Clientes de sincronização	CHSAN
Acesso transparente	Sim	Sim
Edição colaborativa (acesso a legados)	Não	Sim
Acesso desconectado da Nuvem		
Isolado	Sim	Sim
LAN (mesma sub-rede)	Não	Sim
LAN/WAN (roteada)	Não	Sim
Recuperação e versões	Não	Sim
Otimização de transferência		
Sincronização na LAN	DropBox	Sim
Sincronização na LAN/WAN (roteada)	Não	Sim
Sincronização incremental	DropBox	Dependendo da API do provedor
Compressão	DropBox Google Drive	e Dependendo da API do provedor
Sincronização por <i>streaming</i>	DropBox	Sim

nio de *broadcast*, pouco eficiente em redes segmentadas.

A sincronização incremental e a compressão de dados são características de otimização de transferência que visam sincronizar somente os *chunks* alterados e comprimir os *chunks* antes do envio, respectivamente. Tais características podem ser agregadas ao módulo de transporte desde que a API do serviço permita tal funcionalidade. Por fim a sincronização por *streaming*, que é a capacidade de permitir o recebimento de *chunks* por outros clientes mesmo antes do envio completo do arquivo à nuvem, é disponibilizada no CHSAN via NAS. A atualização direta na porção do arquivo a ser alterado é executada no NAS, permitindo aos demais clientes o acesso ao arquivo, respeitando os controles de *lock* do próprio NAS.

Enquanto os clientes estudados focam na otimização do canal de comunicação com o provedor, CHSAN controla os dados, aproveitando a concentração geográfica dos usuários da organização através da técnica de *cache*. O CHSAN acessa os dados diretamente do NAS disponível internamente na organização. O uso do enlace à Internet só ocorre no caso de atualização (escrita) ou sincronização com repositório externo.

## 5.2 Avaliação Experimental da Arquitetura CHSAN

Embora transparente ao cliente, os componentes da arquitetura CHSAN estendidos da arquitetura comum influenciam no modo de acesso destes clientes aos dados. A análise quantitativa da arquitetura CHSAN detém-se à aplicação de cargas de trabalho controladas e simultâneo monitoramento dos componentes. O objetivo é traçar o comportamento da arquitetura CHSAN sob determinada condição e comparar a mesma condição em um cliente *OneDrive* disponível no mercado.

Para atingir esse objetivo, identificou-se dois pontos de testes, o primeiro, focado no módulo de armazenamento e, o segundo, no módulo de transporte. O experimento de análise do módulo de armazenamento visa quantificar o impacto no tempo de acesso de leitura e escrita de dados locais, portanto sem envio ao provedor. Os componentes da arquitetura CHSAN envolvidos nesse experimento estão representados na Figura 5.2 determinado pela hachura em amarelo. A este experimento deu-se o título de Impacto do Armazenamento Híbrido. O experimento de análise do módulo de transporte foca na quantificação do tempo efetivo de envio dos dados do cliente, armazenados localmente, ao provedor na nuvem e da nuvem ao cliente. Nesse experimento todos os componentes são envolvidos: armazenamento, repositório de metadados, módulo de transporte, observador e armazenamento externo (NAS e *OneDrive*) representados na Figura 5.2 determinado pela hachura diagonal em verde. Este experimento foi intitulado de Impacto da Transferência Híbrida.

### 5.2.1 Descrição do Ambiente de Testes

Os experimentos para análise quantitativa exigem a montagem de ambientes reais porém controlados e adequados às ferramentas de monitoração. Esta seção descreve o hardware, software e máquinas virtuais utilizadas nesses ambientes.

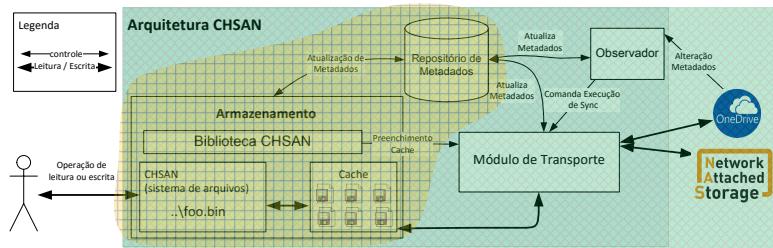


Figura 5.2: Arquitetura CHSAN com hachura para identificação das áreas de teste

O hardware utilizado dispõe de duas categorias, processamento e comunicação, descritos a seguir:

- **Processamento:**

- *Notebook*: Notebooks HP modelo EliteBook 840 G1, processador Core I5 – 4300U de 2.5 GHz, 16 GB RAM, Placa de rede Intel (10/100/1000) e Intel Wireless-N 7260, disco local de 500 GB SATA 5400rpm e sistema operacional Windows 10 64 Bit<sup>1</sup>;
- *Desktops*: Desktops Lenovo 63, processador Core I7 – 4790S de 3.6 GHz, 16 GB RAM, Placa de rede Intel (10/100/1000), disco local de 500 GB SATA 5400rpm e sistema operacional Ubuntu 16.04 LTS 64bit<sup>2</sup>;

- **Comunicação:**

- *Access Point*: D-LINK DWR-922B 802.11 /b/g/n 300 Mbps;
- *Switch*: SMC Networks SMC6128L2 28 Portas UTP 10/100 Mbps;
- *Gateway*: Roteador, NAT e DHCP server Pfsense 2.3<sup>3</sup>;

<sup>1</sup> <https://www.microsoft.com/pt-br/windows/>

<sup>2</sup> <http://www.ubuntu.com/desktop>

<sup>3</sup> <https://www.pfsense.org/>

O conjunto de software é constituído por *benchmarks*, para geração de carga de trabalho controlada, geração de arquivos aleatórios e software para cenário de teste.

- **Software para o cenário de teste:**

- *Benchmark* sistema de arquivos: *Postmark*, Versão 1.5 (KAT-CHER, 1997);
- Gerador de arquivos aleatórios: *Random Data File Creator* (RDFC)<sup>4</sup>, versão 0.1.0.5 (Windows);
- Microsoft OneDrive: Versão 2015, 17.3.6390.0509 (Windows 10);
- FUSE: Dokany FUSE for windows, versão 0.7.4;
- Hipervisor: Oracle Virtual Box 5.0.16 (Linux e Windows)

O protótipo CHSAN foi construído com as especificações para permitir a validação da arquitetura.

- **Protótipo:**

- Linguagem de programação: C# .Net, da plataforma de desenvolvimento
- Repositório de metadados: Microsoft SQLServer 2014;
- Biblioteca de desenvolvimento: OneDrive SDK C#<sup>5</sup>;

Finalmente, o ambiente de teste contou com um conjunto de máquinas virtuais, cuja funcionalidade atende aos atores necessários a cada cenário. São elas:

- **Máquinas Virtuais:**

<sup>4</sup> <http://www.bertel.de/software/rdfc/index-en.html>

<sup>5</sup> <https://dev.onedrive.com>

- MV-CHSAN: Hospedagem do cliente CHSAN com a configuração de hardware virtual, composto de 2 vCPUs, 2 GB RAM e 80GB disco e sistema operacional Windows 10 32 Bits. Softwares adicionais: Microsoft SQLServer 2014; FUSE DokanY for windows e CHSAN versão 0.1.
- MV-*OneDrive*: Hospedagem do cliente *OneDrive* com a mesma configuração de hardware virtual que a anterior.
- MV-NAS: Hospedagem de arquivos com papel de servidor NAS. As mesmas configurações de hardware virtual que as demais, sem nenhum software adicional, apenas com um compartilhamento NTFS habilitado para diretório específico.
- MV-NATIVO: Testes de desempenho com armazenamento nativo, mesma configuração de hardware virtual que as anteriores.
- MV-Gateway: Máquina virtual com 1 vCPU, 512 MB RAM, 1GB HD, com funções de gateway, NAT e DHCP servidor Pf-sense 2.3.

Os hardware, software e máquinas virtuais foram interconectados de quatro formas distintas: isolado, ponto-a-ponto, Wi-Fi e provedor. A Figura 5.3 ilustra os hardware utilizados e a forma de interconexão de cada esquema. O esquema isolado Figura 5.3(a) não tem acesso à rede, o hardware é o notebook com a máquina virtual MV-NATIVO ou MV-CHSAN, conforme necessidade. A Figura 5.3(b) ilustra o esquema ponto-a-ponto constituído por dois *notebooks*, um com a máquina virtual MV-NATIVO e outro com a máquina virtual MV-NAS. A interconexão é direta, ponto-a-ponto, com cabo UTP categoria 6 com velocidade de 1 Gbps. A Figura 5.3(c) descreve componentes de um esquema de rede sem fio, dois *notebooks* e interconexão a 144.4 Mbps. Por último, tem-se o cenário completo, único com acesso à Internet. Ele é composto por quatro *desktops* interconectados por um único *switch* com enlaces internos de 100 Mbps. A conexão à Internet é de 100 Mbps e permite acesso ao provedor de armazenamento na nuvem passando pelo *gateway*.

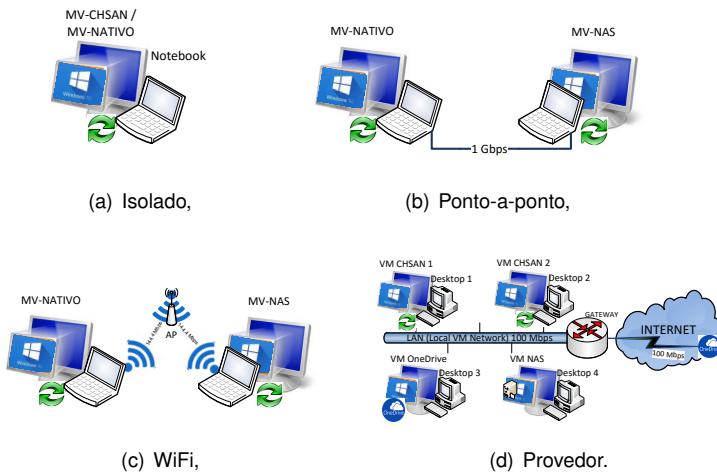


Figura 5.3: Ambientes de testes

### 5.2.2 Ferramentas de Monitoração e Carga de Trabalho

O sucesso dos experimentos depende, entre outras coisas, da precisão na montagem do ambiente e da correta injeção de carga de trabalho. A monitoração do ambiente foi feita através de roteiros automatizados desenvolvidos pelo próprio autor baseado em comandos disponíveis no sistema operacional Windows 10. Os dados monitorados foram tempos de acesso, datas de acesso e atributos dos arquivos observados. Estas informações puderam ser obtidas pelos comandos DATE, TIME e WMIC. As demais informações necessárias foram obtidas diretamente no banco de dados alimentado pelo próprio cliente CHSAN.

A carga de trabalho segue a metodologia de trabalhos anteriores (KANAUJIA; GIRIDHAR, 2012), executando operações de leitura e escrita com diferentes tamanhos de arquivos (1 MB, 50 MB, 100 MB e 200 MB). Pelo trabalho de caracterização apresentado por (GONÇALVES, 2014), 93% dos arquivos são menores que 1 MB e tal tamanho do

ponto de vista de tempo para operação, é inferior a 1 segundo. Com os arquivos de tamanho de 50 MB a 200 MB foi possível estabelecer uma tendência. Os arquivos gerados no teste precisam ser binariamente únicos, evitando a contaminação dos resultados por efeitos de *cache* ou por técnicas de otimização.

A ferramenta *postmark* (KATCHER, 1997), é usada para testes de referência (*benchmark*) e realiza uma sequência de operações de arquivos com base em um arquivo de configuração, medindo o tempo necessário para executar tal configuração em um determinado diretório. O *postmark* foca em operações de sistema de arquivos, medindo os tempos das operações em si e a vazão do fluxo de dados durante as operações.

O aplicativo de criação de binários aleatórios, RDFC<sup>6</sup> (*Random Data File Creator*), é uma ferramenta de linha de comando que cria arquivos com conteúdo aleatório com tamanho customizável. Esta ferramenta auxilia na criação de roteiros automatizados (*scripts*) para criação de arquivos que serão utilizados nos testes de transferência. É importante que os arquivos utilizados a cada teste e a cada transferência para nuvem sejam únicos para evitar a contaminação dos testes por alguma estratégia de otimização.

### 5.2.3 Experimentação e Resultados

Dois cenários são estudados: impacto do armazenamento híbrido e impacto da transferência híbrida.

O primeiro cenário, impacto do armazenamento híbrido, envolve os componentes da arquitetura híbrida hachurada em amarelo na Figura 5.2. O objetivo deste experimento é quantificar a sobrecarga, caso

---

<sup>6</sup> <http://www.bertel.de/software/rdfc/index-en.html>

haja, imposta pelo arquitetura do CHSAN em comparação ao modelo de armazenamento local do cliente de sincronização do *OneDrive* para Windows 10. Esse experimento testou os tempos de escrita e leitura nos quatro primeiros esquemas de conexão, isolado (Figura 5.3(a)), ponto-a-ponto (Figura 5.3(b)) e Wi-Fi (Figura 5.3(c)).

O segundo cenário, impacto da transferência híbrida, envolve os componentes hachurados em verde na Figura 5.2. Neste experimento, busca-se quantificar o impacto no enlace à Internet gerado pelo cliente de sincronização CHSAN comparado com o cliente *OneDrive*. O esquema utilizado foi o provedor descrito na Seção 3.3.3 e ilustrado na Figura 5.3(d). Este cenário é constituído por 5 máquinas físicas tipo Desktop, hospedando 2 MV-CHSAN, 1 MV-ONEDRIVE, 1 MV-NAS e 1 MV-GW.

### **Impacto do armazenamento híbrido**

Os testes de impacto do armazenamento híbrido pretendem estabelecer uma relação de tempo em relação a outras experiências de armazenamento. O software de *benchmark* de sistema de arquivos foi parametrizado para executar 10 operações de arquivos como criar, adicionar dados, ler e apagar. O conjunto de testes foi repetido 10 vezes.

Além do sistema de arquivo CHSAN, os testes foram promovidos com seis conjuntos de condições de armazenamento distintas, formando dois grupos, os sistemas de arquivos Nativo (NTFS e NTFS sobre NAS), que servirão com parâmetro de comparação com o grupo de sistemas de arquivo Dokan (FUSE for Windows) que inclui o CHSAN:

- **Nativo local:** leitura e escrita realizadas no mesmo dispositivo de armazenamento (com NTFS) que o sistema operacional está localizado. Este cenário é utilizado como linha de base (identificado por *Local* nos gráficos).

- **Native LAN cabeada:** operações realizadas em um repositório remoto com sistema de arquivo CIFS (NAS) (sobre NTFS) via mapeamento pelo sistema operacional do compartilhamento de rede. O ambiente de testes utilizado foi ponto-a-ponto Figura 5.3(b).
- **Native LAN WiFi:** Da mesma forma que o anterior e com o mesmo mapeamento, mas o ambiente de rede foi substituído pelo WiFi Figura 5.3(c) (legenda *LAN WiFi*).
- **Native USB:** Escrita e leitura realizadas em um dispositivo móvel de armazenamento conectado por uma interface USB 2.0 com sistema de arquivos FAT32. Este cenário é cotidiano a usuários, tanto domésticos quanto organizacionais para o transporte de informações entre diferentes computadores sem a utilização de rede (legenda *USB*).
- **Dokan C++<sup>7</sup>:** sistema de arquivo FUSE em C++ que dispõe de um volume remoto como um diretório do sistema de arquivos local. Este exemplo de implementação é tido como o de menor impacto no tempo de resposta do sistema de arquivos na camada de usuário (FUSE para Windows) serve para balizar a qualidade do desenvolvimento do CHSAN (legenda *Dokan C++*).
- **Dokan .Net:** possui funcionalidades iguais ao cenário anterior. Entretanto, foi implementado com a linguagem C# .NET, mesma utilizada pelo protótipo CHSAN . O objetivo é servir como parâmetro de comparação com o sistema de arquivo CHSAN (legenda *Dokan .Net*).

---

<sup>7</sup> Dokan, disponível em <<https://github.com/dokan-dev/>>.

Com exceção dos cenários, nativo LAN cabeada e LAN Wifi os demais testes foram executados em ambiente de teste isolado Figura 5.3(a), ou seja somente na máquina virtual, que estava hospedada em um dos notebooks do ambiente de testes. Os gráficos a seguir exprimem os valores médios, o desvio padrão, valores máximos e mínimos (intervalo de confiança de 95%).

A Figura 5.4 e a Tabela 5.2 contém os valores de tempo de execução e vazão obtidos no teste. Os cenários *Nativo local*, *Nativo USB*, *Nativo LAN cabeada* e *Nativo LAN Wifi* não alteram seu modo de funcionamento, sendo que o teste de referência *benchmark postmark*, visa apenas obter os dados desta condição de armazenamento para comparação com cenários em que existe a introdução de componentes Dokan/FUSE.

Os cenários *Dokan C++* e *Dokan .Net* apontam como repositório um diretório temporário em disco local e são modelos de códigos disponibilizados pelo projeto Dokany, que redirecionam as operações de sistema de arquivos a um diretório local, passando pelo FUSE. A comparação desses modelos com o CHSAN permitem averiguar a qualidade da implementação do protótipo, servindo como balizadores para determinar a eficiência do CHSAN. Em CHSAN, as operações realizadas são atendidas pelas etapas (i), (ii) e (iii) (Figura 4.3).

Cenário	1 MB		50 MB		100 MB		200 MB	
	$\bar{\sigma} \pm \sigma$	min; max	$\bar{\sigma} \pm \sigma$	min; max	$\bar{\sigma} \pm \sigma$	min; max	$\bar{\sigma} \pm \sigma$	min; max
Nativo local	10 $\pm$ 0	10; 10	71,43 $\pm$ 9,09	62,50; 83,33	49,75 $\pm$ 10,09	37,04; 67,02	31,49 $\pm$ 3,85	23,53; 35,71
Nativo LAN cabeada	8,33 $\pm$ 2,72	3,33; 10	20,26 $\pm$ 5,77	7,35; 25	23,54 $\pm$ 3,38	15,63; 27,03	23,29 $\pm$ 4,33	12,20; 26,67
Nativo LAN Wifi	1,02 $\pm$ 0,37	0,5; 1,25	6,44 $\pm$ 15,31	1,39; 50	1,59 $\pm$ 0,49	1,19; 2,88	1,68 $\pm$ 0,40	1,37; 2,66
Nativo USB	1,25 $\pm$ 0,21	0,83; 1,67	3,01 $\pm$ 0,09	2,89; 3,16	3,14 $\pm$ 0,03	3,09; 3,17	3,18 $\pm$ 0,04	3,15; 3,27
Dokan C++	7,75 $\pm$ 2,99	2,50; 10	7,18 $\pm$ 1,29	5,10; 8,77	7,90 $\pm$ 2,99	4,20; 15,38	7,29 $\pm$ 1,46	4,21; 8,93
Dokan .Net	5,83 $\pm$ 2,26	3,33; 10	6,75 $\pm$ 0,73	5,05; 7,69	6,53 $\pm$ 0,69	5,18; 7,35	6,14 $\pm$ 0,81	4,85; 7,19
CHSAN	3,39 $\pm$ 0,87	3,30; 5	6,07 $\pm$ 0,64	4,95; 6,76	6,19 $\pm$ 0,54	5,43; 6,99	5,92 $\pm$ 1,21	2,81; 6,87

Tabela 5.2: Vazão alcançada pelos sistemas analisados (em MB/s).

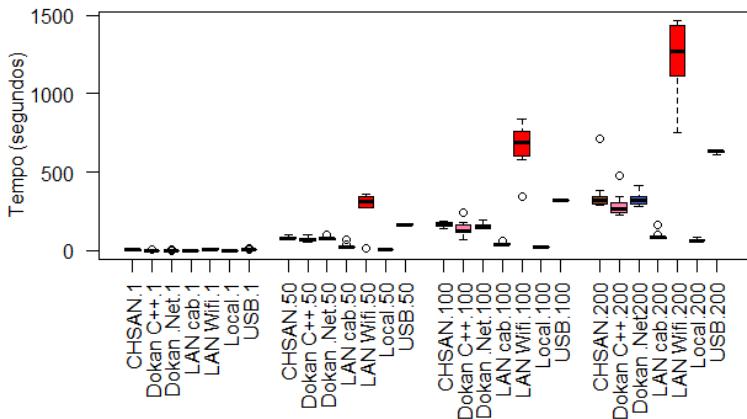


Figura 5.4: Tempo de execução (em segundos) dos sistemas analisados.

Na Tabela 5.2, estão representados os valores para a vazão (megabytes por segundo) alcançada durante a execução dos testes nos cenários propostos.

### Impacto da transferência híbrida

Os testes de transferência visam averiguar se o desempenho do CHSAN de envio e recebimento de arquivos para nuvem é compatível com o cliente *OneDrive*. Além disso, o teste pretende averiguar o tempo necessário para o recebimento de arquivos em um armazenamento NAS, sobre uma rede local nas mesmas condições. O ambiente de rede utilizado é o da UDESC. A Figura 5.3(d) ilustra a disposição do ambiente.

Os testes foram realizados em dois ciclos, CHSAN envia enquanto *OneDrive* recebe no primeiro ciclo e no segundo ciclo *OneDrive* envia e CHSAN recebe. No primeiro ciclo, foram medidos os tempos de envio de cada arquivo do teste do CHSAN para nuvem e do recebimento pelo cliente *OneDrive*. Aproveita-se ainda o primeiro ciclo para medir o tempo que um segundo cliente CHSAN demorou para receber o arquivo do NAS. E no segundo ciclo o tempo que o CHSAN precisa para receber

um arquivo da nuvem que não está armazenado no NAS.

No primeiro ciclo, os arquivos aleatórios são criados em CHSAN 1 contabilizado o tempo necessário para que cada arquivo seja gravado na nuvem, obtendo a data de criação do arquivo nos metadados da nuvem, comparando a data de criação do arquivo local. Durante o processo de envio de cada arquivo, o cliente CHSAN 1 gravou os mesmos arquivos no NAS. O cliente *OneDrive* quando notificado de um novo arquivo na nuvem fará o recebimento dos arquivos. A data de modificação do arquivo no cliente *OneDrive* é comparada com a data de criação do arquivo na nuvem.

O cliente CHSAN 2 recebeu a atualização dos metadados. Após o recebimento dos 10 arquivos pelo cliente *OneDrive* e da atualização dos metadados do CHSAN 2 foi iniciado uma cópia sequencial dos 10 arquivos no cliente CHSAN 2 para o disco local. Tal cópia foi obtida do NAS, e este tempo necessário para copiar cada arquivo foi medido.

O segundo ciclo é iniciado gerando-se dez arquivos aleatórios no cliente *OneDrive*, medindo-se a diferença entre a data de criação do arquivo local e a data de criação do arquivo na nuvem. Quando os 10 arquivos do *OneDrive* foram enviados à nuvem, é iniciado o recebimento dos 10 arquivos no CHSAN 2 para disco local. Este tempo de cópia é medido para cada arquivo, encerrando-se os experimentos de transferência.

A Figura 5.5 avalia o desempenho das operações em arquivos compartilhados localizados no sistema de armazenamento na nuvem e no armazenamento NAS e os gráficos exprimem os valores médios, valores máximos e mínimos (intervalo de confiança de 95%). O tempo de envio para o NAS está contido no tempo de envio para nuvem, os processos de envio para o NAS e nuvem acontecem de forma paralela, sendo o tempo NAS sempre menor que o tempo CHSAN para envio à nuvem.

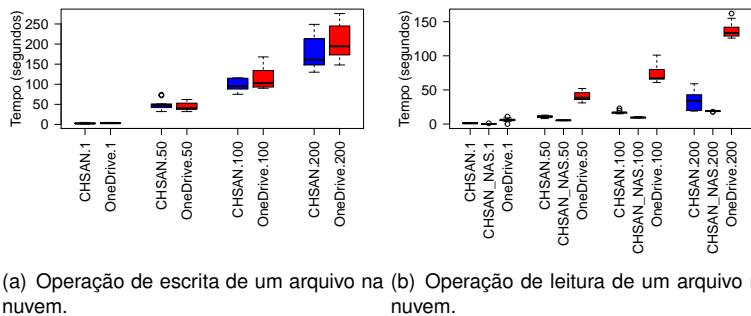


Figura 5.5: Operações em arquivos localizados no armazenamento externo (CHSAN vs. *OneDrive*).

#### 5.2.4 Análise dos Resultados Experimentais

Os resultados obtidos são promissores, tanto interno ao sistema de arquivos CHSAN, quanto no envio e recebimento da nuvem. A seguir apresenta-se a análise dos resultados obtidos.

##### Análise do impacto do armazenamento híbrido

Na Figura 5.4 e na Tabela 5.2, os sistemas baseados em FUSE (*Dokan C++*, *Dokan .Net* e CHSAN) apresentam impacto no desempenho oriundo das interações entre o *kernel* e a camada de usuário. Os cenários com sistemas nativos fazem acesso direto ao dispositivo via VFS, naturalmente, apresentam tempos de acesso inferiores aos demais, exceto quando o meio de acesso ao dado é o gargalo, o que ocorre em *Native LAN Wifi* e *Native USB*.

O tempo de execução das operações (em segundos) nos diferentes repositórios é resumido na Figura 5.4. Os cenários *Native LAN cabeada* e *Native LAN Wifi* representam a linha de base de acesso a sistemas de armazenamento privados implantados em uma LAN. Embora o

CHSAN possua um mecanismo com maior número de operações (discutido no Capítulo 4), a sobrecarga imposta por CHSAN quando comparado ao cenário *Nativo LAN cabeada*, é somente perceptível para operações com arquivos pequenos (1 MB), sendo suavizada para arquivos maiores (aproximadamente 2.9, 3.7 e 3.9 segundos maior para cenários com (50 MB, 100 MB e 200 MB), respectivamente. O tamanho dos arquivos não influencia no tempo das operações de banco de dados do CHSAN.

Quando comparado aos cenários com operações de sistema de arquivos virtual que possuem FUSE, identificados pelas legendas *Dokan C++* e *Dokan .Net*, CHSAN obteve-se resultados competitivos, mesmo requerendo mais ciclos de CPU para acesso à base de dados. *Dokan .Net* apresentou uma sobrecarga aproximada de 36%, 10%, 5%, 8% e 6%, para operações com arquivos de 1 MB até 200 MB, respectivamente. Esses valores representam o tempo necessário para atualização do repositório de metadados. CHSAN obteve um menor tempo médio de execução das operações, quando comparado aos cenários *Nativo Wifi* e *Nativo USB*. Em suma, a sobrecarga acrescentada por CHSAN, quando comparada às linhas de base (*Nativo local* e *Nativo LAN cabeada*), é esperada, devido às operações necessárias para controle e gerenciamento dos metadados. Entretanto, quando comparado a cenários logicamente próximos (*Dokan C++* e *Dokan .Net*), CHSAN obteve um desempenho equivalente. Quando comparado aos cenários *Nativo USB* e *Nativo LAN Wifi*, CHSAN, obteve um desempenho superior em todos os casos. É importante ressaltar que usuários já estão acostumados a tais tempos de resposta, fornecidos por esses tipos de dispositivos, o que indica que são populares e satisfatórios para uma gama de aplicações.

O comportamento dos resultados de CHSAN perante os cenários *Nativo local* e *Nativo LAN cabeada* é semelhante ao discutido na Figura 5.4. A vazão média alcançada por CHSAN é comparável aos sistemas de escrita nativa sobre sistemas de armazenamento virtuais.

É importante ressaltar que, independente do volume de dados escritos e lidos no sistema, CHSAN manteve uma vazão superior aos sistemas *Nativo USB* e *Nativo LAN WiFi*. Ainda, a vazão obtida por CHSAN é de aproximadamente 6,18 MB/s para arquivos entre 50 e 200 MB, com valores menores para arquivos de 1 MB (3,9 MB/s) e tal comportamento é esperado, pois o número de operações em banco de dados é o mesmo independente do tamanho do arquivo, dependendo da forma como o arquivo está sendo manipulado. Operações com conclusão a cada bloco transferido necessitaram de atualização dos atributos do arquivo (a cada bloco), como por exemplo: última data de atualização do arquivo.

Os resultados do primeiro conjunto de testes comprovam a eficiência do CHSAN nas operações de escrita, leitura e deleção de arquivos sem atualização imediata do repositório na nuvem. Arquiteturas dos clientes de sincronização de arquivos na nuvem baseiam-se em sistemas de arquivos *Nativo local*, otimizando seus resultados nesse contexto, porém, ignoram o conteúdo dinâmico do repositório de metadados e, consequentemente, suas vantagens.

### Análise do impacto da transferência híbrida

Nas Figuras 5.5 o CHSAN superou o cliente *OneDrive* na média de todas as medições, com exceção ao tempo de envio para arquivos de 50 MB. Na Figura 5.5(a), as médias de CHSAN são: 3 s, 50 s, 97 s e 178s enquanto para *OneDrive* são: 4 s, 44 s, 115 s e 205 s para os tamanhos de arquivos de 1 MB, 50 MB, 100 MB e 200 MB respectivamente nas operações de escrita para o armazenamento externo. Como ambos (CHSAN e *OneDrive*) se utilizam da mesma API, da mesma infra-estrutura e foram medidos em igualdade de condições a diferença se dá pelo paralelismo aplicado, pela estratégia do *OneDrive* e pela forma de medir. O tempo apresentado nas Figuras 5.5 é contabilizado em cada arquivo, entretanto 10 arquivos são escritos sequencialmente no início da medição. O cliente *OneDrive* paralelizou o envio e recebimento de 4 arquivos por vez, en-

quanto o CHSAN faz o envio sequencial de cada arquivo. Além disso, a natureza da CHSAN para escrita é síncrona, enquanto a do *OneDrive* é assíncrona. Se considerar o tempos entre execução do observador do *OneDrive*, os resultados do CHSAN nesta condição seriam ainda melhores.

Na Figura 5.5(b), os resultados médios do CHSAN para operação de escrita são: 1 s, 11 s, 17 s e 36 s do CHSAN\_NAS (menor que  $> 1$  s), 6 s, 10 s e 19 s enquanto do *OneDrive* 6 s, 40 s, 75 s e 138 s para os tamanhos de arquivos 1 MB, 50 MB, 100 MB e 200 MB respectivamente. Nota-se que o tempo de recebimento de um arquivo do armazenamento CHSAN\_NAS é até 7 vezes menor que *OneDrive* enquanto CHSAN é até 4 vezes menor. O recebimento do *OneDrive* é executado de forma paralela.

Os resultados do CHSAN quanto ao envio e recebimento compatíveis com *OneDrive* foram satisfatórios. Uma análise de técnicas de paralelização de transferência de arquivos deve ser objeto de trabalhos futuros. A superioridade do armazenamento NAS já era esperado devido ao tempo de latência reduzido. Além disso, cada arquivo recebido do NAS reflete diretamente em economia do enlace de Internet. A quantidade de vezes que o armazenamento NAS pode ser superior ao armazenamento na nuvem depende exclusivamente da capacidade do enlace de Internet versus a capacidade do ambiente de rede LAN e da capacidade do servidor NAS. Tais evidências direcionam a prospectar um cenário de utilização real do CHSAN em um ambiente corporativo.

### 5.3 Análise da Escalabilidade da Arquitetura CHSAN

O cenário descrito na Seção 5.2.3 demonstra que a arquitetura do CHSAN é capaz de operar em harmonia e coexistir com os clientes e serviços de acesso a arquivos em nuvem de forma transparente para

os usuários e para o serviço. O CHSAN ocupa no disco local uma área de armazenamento de *cache* configurável e este espaço independe do tamanho dos dados armazenados no repositório da nuvem.

No estudo de caracterização de armazenamento em nuvem realizado por (GONÇALVES, 2014) foi identificado que 333 usuários possuíam 3 milhões de arquivos totalizando aproximadamente 1,38 TB de dados. No protótipo de CHSAN , no banco de dados utilizado, esse volume seria indexado e disponibilizado, consumindo apenas 2,2 GB de área de armazenamento e poderia estar disponível a todos os usuários.

O fluxo total de entrada e saída para o armazenamento externo é diminuído, pois somente são trafegados arquivos solicitados, abordagem distinta da realizada pelos clientes de sincronização convencionais que mantêm a sincronização independente do uso. Além disso, a implementação do CHSAN com um armazenamento NAS reduz o tráfego de Internet para a leitura de arquivos por múltiplos usuários colaborativos, reduzindo o tempo de acesso (latência) em 7 vezes ou mais como pode ser identificado na Figura 5.5(b). Os clientes de um mesmo compartilhamento NAS deixam de buscar os dados da nuvem e passam a atualizar a *cache* com base nas versões locais. Destaca-se que o contato com a nuvem é necessário a cada escrita para disponibilizar aos usuários internos e externos as informações sobre o procedimento realizado.

A equação  $v = b_e * \alpha(1 + n)$  pode ser utilizada para estimar o volume de dados trafegados com os repositórios de armazenamento na nuvem, sendo  $v$  o volume total de dados trafegados,  $b_e$  o volume de bytes escritos no provedor de nuvem,  $n$  o número de clientes a sincronizar, e  $\alpha \leq 1$  o coeficiente de compressão. Tal fórmula é válida para diretórios compartilhados pelos mesmos dispositivos. Na arquitetura híbrida de CHSAN,  $v$  passa a ser guiado por  $v = \alpha * b_e$ , sendo independente do número de clientes colaborativos e do grau de compartilhamento.

A relação entre o grau de compartilhamento e o volume de dados trafegados é explorada nos trabalhos (GONÇALVES, 2016) e (GONÇALVES, 2015) e prospecta conclusões semelhantes às aqui apresentadas. O trabalho de (GONÇALVES, 2016) coletou dados sobre o comportamento do *DropBox* durante um ano com um público total de mais de 27 K dispositivos divididos em dois tipos de ambientes, ambiente domésticos (provedores de Internet residencial) 13 K dispositivos e dois campus com 14 K dispositivos. O trabalho indica que usuários de universidades (organizações) compartilham mais arquivos que usuários domésticos.

As equações  $v = b_e * \alpha(1 + n)$  e  $v = \alpha * b_e$  permitem prospectar uma utilização do CHSAN em ambiente empresarial de forma estimada. Em um cenário composto de um escritório de vendas com 10 usuários (pequeno), uma unidade com 1000 usuários (médio) e uma localidade com 10 K usuários (grande). Se cada usuário gerar uma média de 7 MB por dia (*upload*), média geral encontrada em (GONÇALVES, 2016), e seu grupo de trabalho tiver ao menos 10 pessoas compartilhando os mesmos arquivos. Considerando que o cliente *OneDrive*, conforme apresentado na Tabela 3.2 não possui compressão o coeficiente  $\alpha$  será = 1. O efeito exponencial decorrente da arquitetura do cliente *OneDrive* resulta em um volume de dados diário com a nuvem equivalente a 700 MB para localidade pequena, 7 GB para localidade média e 700 GB para uma localidade grande. Com a arquitetura CHSAN a mesma atividade poderia ser realizada pelas localidades consumindo 7 MB, 690 MB e 68 GB respectivamente.

## 5.4 Considerações Parciais

O CHSAN é uma arquitetura que pode ser adotada com qualquer provedor de serviço de acesso a arquivos em nuvem, e que se utiliza dos dispositivos NAS existentes, apresentando resultados compatíveis com

outras implementações em FUSE for Windows, indicando que sua sobre-carga está compatível com as de outras soluções que adotam tal tecnologia. Além disso, o CHSAN apresenta menor latência para acesso a arquivos, principalmente quando estes estão disponíveis em um repositório NAS local. No cenário apresentado, é possível identificar que a arquitetura do *OneDrive* é exponencial em relação ao compartilhamento, enquanto o CHSAN se mantém estável independente do grau de compartilhamento exigido.



## 6 CONCLUSÃO E TRABALHOS FUTUROS

A adoção de serviços de armazenamento em nuvem por usuários domésticos é uma realidade e o interesse das organizações no tema é um fato. As características essenciais de NIST estão alinhadas com as necessidades das organizações pois promovem otimizações de capital, equipamentos e pessoas.

Por sua vez, o armazenamento distribuído que se faz presente na maioria das organizações é disponibilizado através de soluções legadas, como NAS por exemplo. Este modelo de armazenamento distribuído é base para sistemas implantados e corresponde as necessidades internas das organizações. Todavia, a concepção de organizações com múltiplos usuários, geograficamente distribuídos e concentrados em redes privadas tem se expandido. Estas organizações continuam apoiando-se nas características de soluções legadas e desejam incrementar seus negócios com a incorporação dos serviços de acesso a arquivos na nuvem.

Os modelos de serviços da nuvem (IaaS, PaaS e SaaS) descrevem a forma de fornecimento de recursos. No contexto de armazenamento, o enquadramento dos serviços de acesso a arquivos depende da forma de acesso aos dados. Neste trabalho, definiu-se que o modelo de serviço IaaS entrega os dados através do acesso a blocos, que PaaS disponibiliza os dados através de objetos e que, finalmente, o SaaS dispõe os dados em forma de arquivos. Dentre as formas de acesso a arquivos (SaaS), os fornecedores mais populares são *DropBox*, *Google Drive* e *OneDrive*.

Estes fornecedores oferecem clientes que não foram desenvolvidos para atender os requisitos específicos de organizações com múltiplos usuários colaborativos, geograficamente distribuídos e concentrados em redes privadas. As organizações se diferem nos objetivos o que afeta a densidade versus localização e tamanho dos grupos de colabo-

ração por cliente. A identificação de uma arquitetura comum desse serviço bem como um análise de funcionalidades dos clientes de sincronização é mais uma contribuição deste trabalho. Embora todos os clientes de sincronização estudados não se afastem da arquitetura comum, eles diferem em seus objetivos. *DropBox* é o mais completo em termos de funcionalidades, atacando, por exemplo, problemas como sincronização direta entre dois pontos em uma mesma sub rede, compressão, sincronização incremental e sincronização por *streaming*. Em contrapartida, *OneDrive* atende o mercado corporativo associando a sua iniciativa a outras ferramentas como *Office 365*, porém é o cliente mais enxuto em termos de funcionalidades. O *Google Drive* preocupa-se com a compressão dos dados, mas destaca-se mesmo na integração com as aplicações do Google, seu desenvolvedor. Entretanto, a latência e o compartilhamento, pontos críticos para a sincronização nas organizações, não são preocupações de nenhum destes clientes.

Nesse contexto, o presente trabalho apresentou a arquitetura de um Cliente Híbrido para Sincronização de Arquivos em Nuvem CHSAN que combina o uso do disco local, de sistemas de armazenamento privados e do armazenamento na nuvem. CHSAN estendeu as características básicas dos clientes de sincronização de arquivos na nuvem, fornecendo soluções para as três principais limitações de uso desses clientes nas organizações: *(i) ausência de integração com sistemas de armazenamento privado; (ii) necessidade de cópia local de todos os arquivos compartilhados; e (iii) latência de sincronização entre arquivos da nuvem e repositórios locais*. A arquitetura do CHSAN apoia-se diretamente na camada de aplicação do serviço via API, interagindo com os módulos de armazenamento distribuído e gerenciamento dos provedores de forma transparente. A integração com NAS reduz o uso do enlace de Internet e a latência percebida pelo cliente, pois utiliza os dados disponíveis em redes privadas nas organizações e ao mesmo tempo fornece o acesso a estes dados na nuvem.

---

Os resultados experimentais apontam que CHSAN é uma alternativa viável, pois não exige adequações nos serviços disponibilizados pelos fornecedores de armazenamento na nuvem e tem tempos de acessos compatíveis com sistemas de armazenamento semelhantes. O protótipo do CHSAN foi comparado com o cliente padrão do *OneDrive*, na *análise do impacto de transferência* e com outros sistemas de armazenamento locais e remotos, na *análise do impacto do armazenamento híbrido*. No primeiro cenário, CHSAN obteve tempos inferiores para operações de escrita para todos os arquivos testados (1MB, 50MB, 100MB e 200MB). No segundo cenário, CHSAN foi comparado com os sistemas de arquivos virtuais *Dokan C++* e *Dokan.Net*, obtendo resultados competitivos, com sobrecarga aproximada de 36%, 10%, 5%, 8% e 6%, para operações com arquivos de 1MB até 200MB, respectivamente.

Os valores de sobrecarga representam o tempo necessário para atualização do repositório de metadados, operação inexistente nos sistemas de arquivos virtuais comparados, porém essencial para sistemas de arquivos na nuvem. Portanto, os resultados obtidos com o protótipo do CHSAN evidenciam as potencialidades exploradas pela arquitetura para o armazenamento na nuvem no contexto das organizações.

Finalmente, em trabalhos futuros, pode-se definir duas categorias: conceitual e operacional. No âmbito conceitual, a arquitetura de CHSAN não explorou técnicas mais complexas de *cache* que tratem localidade, por exemplo. A arquitetura trata o disco local e o espaço de armazenamento NAS como *caches* de igual importância, porém os resultados obtidos com o protótipo indicam que o uso da *cache* NAS tem um impacto positivo na redução do consumo do enlace de internet importante. No operacional, é evidente a necessidade de desenvolvimento do cliente CHSAN compatível com outros fornecedores de serviço de armazenamento na nuvem. Embora o *OneDrive* tenha um apelo comercial devido a seu forte vínculo com as organizações, foco principal deste trabalho, ele é o cliente com o menor número funcionalidades disponíveis.



## REFERÊNCIAS BIBLIOGRÁFICAS

- ANDRIANI, G.; KOSLOVSKI, G.; PILLON, M. A. Sincronização de arquivos entre nuvens de armazenamento e repositórios geograficamente distribuídos. 2016.
- ANNAPUREDDY, S.; FREEDMAN, M. J.; MAZIERES, D. Shark: Scaling file servers via cooperative caching. In: **Proceedings of the 2Nd Conference on Symposium on Networked Systems Design & Implementation - Volume 2**. Berkeley, CA, USA: USENIX Association, 2005. (NSDI'05), p. 129–142. Disponível em: <<http://dl.acm.org/citation.cfm?id=1251203.1251213>>.
- A. BESEN, H. CHEONG, H. MUELLER, F. PAPE e D. WURTZ. **Sharing and synchronizing electronically stored files**. 2015. US Patent 8,949,179. Disponível em: <<https://www.google.com/patents/US8949179>>.
- BESSANI, A. et al. Scfs: a shared cloud-backed file system. In: **2014 USENIX Annual Technical Conference (USENIX ATC 14)**. Philadelphia, PA: USENIX Association, 2014. p. 169–180. ISBN 978-1-931971-10-2. Disponível em: <<https://www.usenix.org/conference/atc14/technical-sessions/presentation/bessani>>.
- BHAGWAT, D. et al. Extreme binning: Scalable, parallel deduplication for chunk-based file backup. In: IEEE. **International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems, 2009. MASCOTS'09**. [S.I.], 2009. p. 1–9.
- BHAT, S. et al. **Active cache offline access and management of project files**. Google Patents, 2007. US Patent App. 11/390,813. Disponível em: <<https://www.google.com/patents/US20070239725>>.
- CHIAVENATO, I. **Administração nos novos tempos**. [S.I.]: Elsevier Brasil, 2005.
- DOUCEUR, J. R.; BOLOSKY, W. J. A large-scale study of file-system contents. In: **Proceedings of the 1999 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems**. New York, NY, USA: ACM, 1999. (SIGMETRICS '99), p. 59–70. ISBN 1-58113-083-X. Disponível em: <<http://doi.acm.org/10.1145/301453.301480>>.

- DRAGO, I. et al. Benchmarking personal cloud storage. In: **Proceedings of the 2013 Conference on Internet Measurement Conference**. New York, NY, USA: ACM, 2013. (IMC '13), p. 205–212. ISBN 978-1-4503-1953-9. Disponível em: <<http://doi.acm.org/10.1145/2504730.2504762>>.
- DRAGO, I. et al. Inside Dropbox: Understanding personal cloud storage services. In: **Proceedings of the 2012 ACM Conference on Internet Measurement Conference**. New York, NY, USA: ACM, 2012. (IMC '12), p. 481–494. ISBN 978-1-4503-1705-4. Disponível em: <<http://doi.acm.org/10.1145/2398776.2398827>>.
- DUAN, H. et al. CSTORE: A desktop-oriented distributed public cloud storage system. **Computers & Electrical Engineering**, Elsevier, v. 42, p. 60–73, 2015.
- FRENCH, S. M.; TEAM, S. A new network file system is born: Comparison of smb2, cifs and nfs. In: **Linux Symposium**. [S.I.: s.n.], 2007. p. 131.
- GHEMAWAT, S.; GOBIOFF, H.; LEUNG, S.-T. The Google file system. In: ACM. **ACM SIGOPS Operating Systems Review**. [S.I.], 2003. v. 37, n. 5, p. 29–43.
- GIBSON, G. A.; METER, R. V. Network attached storage architecture. **Communications of the ACM**, ACM, v. 43, n. 11, p. 37–45, 2000.
- GONÇALVES, G. et al. Caracterização e modelagem da carga de trabalho do Dropbox. **Anais do SBRC**, 2014.
- GONÇALVES, G. D. et al. Analisando o impacto de compartilhamentos no Dropbox em uma rede acadêmica. **Anais do SBRC**, 2015.
- GONÇALVES, G. D. et al. Workload models and performance evaluation of cloud storage services. **Anais do SBRC**, 2016.
- GONÇALVES, G. D. et al. Trabalho colaborativo em serviços de armazenamento na nuvem: Uma análise do dropbox.
- GRUENWALD III, C. et al. Hare: A file system for non-cache-coherent multicores. In: **Proceedings of the Tenth European Conference on Computer Systems**. New York, NY, USA: ACM, 2015. (EuroSys '15), p. 30:1–30:16. ISBN 978-1-4503-3238-5. Disponível em: <<http://doi.acm.org/10.1145/2741948.2741959>>.
- HÖFER, C.; KARAGIANNIS, G. Cloud computing services: taxonomy and comparison. **Journal of Internet Services and Applications**, Springer,

- v. 2, n. 2, p. 81–94, 2011. Disponível em: <<http://link.springer.com/article/10.1007/s13174-011-0027-x>>.
- DREW HOUSTON e ARASH FERDOWSI. **Network folder synchronization**. 2014. US Patent 8,825,597.
- KANAUJIA, V.; GIRIDHAR, C. Fuseing Python for development of storage efficient filesystem. **The Python Papers**, v. 7, p. 4, 2012.
- KATCHER, J. **Postmark: A new file system benchmark**. Technical Report TR3022, Network Appliance, 1997. Disponível em: <<http://www.filesystems.org/docs/auto-pilot/Postmark.html>>.
- KATZER, M.; CRAWFORD, D. Office 365: Moving to the cloud. In: **Office 365**. [S.I.]: Springer, 2013. p. 1–23.
- KHAJEH-HOSSEINI, A.; SOMMERVILLE, I.; SRIRAM, I. Research challenges for enterprise cloud computing. **arXiv preprint arXiv:1001.3257**, 2010.
- LI, Z. et al. Efficient batched synchronization in Dropbox-Like cloud storage services. In: EYERS, D.; SCHWAN, K. (Ed.). **Middleware 2013**. [S.I.]: Springer Berlin Heidelberg, 2013, (Lecture Notes in Computer Science, v. 8275). p. 307–327. ISBN 978-3-642-45064-8.
- MELL, P.; GRANCE, T. The NIST definition of cloud computing. **National Institute of Standards and Technology**, IGI Publication, v. 53, n. 6, p. 50, 2009.
- Microsoft Corporation. **Common Internet File System (CIFS) Protocol**. 2014. Disponível em: <<http://msdn.microsoft.com/en-us/library/ee442092.aspx>>.
- MUNISWAMY-REDDY, K.-K.; MACKO, P.; SELTZER, M. I. Provenance for the cloud. In: **USENIX Conference on File and Storage Technologies (FAST)**. [S.I.: s.n.], 2010. v. 10, p. 15–14.
- MUTHITACHAROEN, A.; CHEN, B.; MAZIÈRES, D. A low-bandwidth network file system. **SIGOPS Operating Systems Review**, ACM, New York, NY, USA, v. 35, n. 5, p. 174–187, out. 2001. ISSN 0163-5980. Disponível em: <<http://doi.acm.org/10.1145/502059.502052>>.
- NALDI, M.; MASTROENI, L. Cloud storage pricing: A comparison of current practices. In: **Proceedings of the 2013 International Workshop on**

**Hot Topics in Cloud Services.** New York, NY, USA: ACM, 2013. (Hot-TopiCS '13), p. 27–34. ISBN 978-1-4503-2051-1. Disponível em: <<http://doi.acm.org/10.1145/2462307.2462315>>.

NORONHA, R.; PANDA, D. K. IMCa: A high performance caching front-end for GlusterFS on InfiniBand. In: IEEE. **ICPP'08. 37th International Conference on Parallel Processing.** [S.I.], 2008. p. 462–469.

PANZURA: Hybrid Cloud NAS. 2008. <<http://panzura.com/solutions/hybrid-cloud-nas/>>. Accessed: 2016-10-30.

PATIL, S.; GIBSON, G. A. Scale and concurrency of GIGA+: File system directories with millions of files. In: **USENIX Conference on File and Storage Technologies (FAST).** [S.I.: s.n.], 2011. v. 11, p. 13–13.

SHEPLER, S. et al. Network file system (nfs) version 4 protocol. **Network**, 2003. Disponível em: <<http://tools.ietf.org/html/rfc3530.html>>.

SHVACHKO, K. et al. The Hadoop distributed file system. In: IEEE. **IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST).** [S.I.], 2010. p. 1–10.

SMITH, D. M. et al. **Cloud Computing Affects All Aspects of IT.** Gartner Group, 2013. Disponível em: <<https://www.gartner.com/doc/2631851/predicts--cloud-computing-affects>>.

SRINIVASAN, V.; MOGUL, J. Spritely NFS: Experiments with cache-consistency protocols. **SIGOPS Operating Systems Review**, ACM, New York, NY, USA, v. 23, n. 5, p. 44–57, nov. 1989. ISSN 0163-5980. Disponível em: <<http://doi.acm.org/10.1145/74851.74856>>.

TANENBAUM, A. S.; WOODHULL, A. S. **Sistemas Operacionais: Projetos e Implementação.** [S.I.]: Bookman Editora, 2009.

TATE, J. et al. **Introduction to storage area networks.** [S.I.]: IBM Redbooks, 2005.

THENG, D.; HANDE, K. VM management for cross-cloud computing environment. In: IEEE. **International Conference on Communication Systems and Network Technologies (CSNT).** [S.I.], 2012. p. 731–735.

TRIDGELL, A.; MACKERRAS, P. et al. **The rsync algorithm.** [S.I.]: The Australian National University, 1996.

UNIFS - A True Global File System. 2015. <<http://www.nasuni.com/>>. Accessed: 2016-10-30.

- VRABLE, M.; SAVAGE, S.; VOELKER, G. M. Bluesky: A cloud-backed file system for the enterprise. In: USENIX ASSOCIATION. **Proceedings of the 10th USENIX conference on File and Storage Technologies (FAST)**. [S.I.], 2012. p. 19–19.
- WANG, H. et al. On the impact of virtualization on dropbox-like cloud file storage/synchronization services. In: **Proceedings of the 2012 IEEE 20th International Workshop on Quality of Service**. Piscataway, NJ, USA: IEEE Press, 2012. (IWQoS '12), p. 11:1–11:9. ISBN 978-1-4673-1298-1. Disponível em: <<http://dl.acm.org/citation.cfm?id=2330748.2330759>>.
- WAUTIER, M. et al. **Sync framework extensibility**. Google Patents, 2015. US Patent 9,189,533. Disponível em: <<https://www.google.com/patents/US9189533>>.
- WEIL, S. A. et al. Ceph: A scalable, high-performance distributed file system. In: **Proceedings of the 7th Symposium on Operating Systems Design and Implementation**. Berkeley, CA, USA: USENIX Association, 2006. (OSDI '06), p. 307–320. ISBN 1-931971-47-1. Disponível em: <<http://dl.acm.org/citation.cfm?id=1298455.1298485>>.
- YANG, C. et al. **User interface for saving documents using external storage services**. Google Patents, 2013. US Patent App. 13/287,937. Disponível em: <<https://www.google.com/patents/US20130111404>>.
- YU, J.; WU, W.; LI, H. DMooseFS: Design and implementation of distributed files system with distributed metadata server. In: IEEE. **IEEE Asia Pacific on Cloud Computing Congress (APCloudCC)**. [S.I.], 2012. p. 42–47.
- ZHANG, Q.; CHENG, L.; BOUTABA, R. Cloud computing: state-of-the-art and research challenges. **Journal of Internet Services and Applications**, Springer, v. 1, n. 1, p. 7–18, 2010.