Estudo de caso: Análise comparativa entre dois sistemas distribuídos ponto a ponto (IPFS e Bittorrent)

Gabriel Cavalcante

Curso de Ciência da Computação – Universidade do Sul de Santa Catarina (UNISUL)
Caixa Postal 15.064 – 91.501-970 – Tubarão – SC – Brasil

gabrielcavalcante1021@gmail.com

Abstract. *This paper describes the steps taken for a comparative analysis between two peer-to-peer distributed systems. The chosen systems utilized in the comparison are IPFS and Bittorrent, two peer-to-peer distributed network systems that do not need dedicated server to operate, on a small scale. Some variables that are taken into account, namely usability, speed and transparency.*

Resumo. *Este artigo descreve passos tomados para uma análise comparativa entre dois sistemas distribuídos ponto a ponto. Os sistemas escolhidos e utilizados na comparação foram IPFS e o Bittorrent, dois sistemas de redes distribuídas ponto a ponto do qual não necessitam de servidor dedicado para operarem, em uma escala pequena. Algumas variáveis são tomadas em conta, nomeadamente usabilidade, velocidade e transparência.*

1. Introdução

A medida que esse gigantesco agrupamento de tecnologias que compõem a internet foram evoluindo, para tornar as experiências dos usuários mais conveniente e agradável, diversos protocolos, que hoje são amplamente estudados e utilizados no compartilhamento de arquivos, foram desenvolvidos. Entre os mais conhecidos e utilizados estão o Bittorrent; um gigantesco sistema de distribuição de arquivos, que, como dizem L. Wang e J. Kangasharju (2013), "até hoje, o Bittorrent mantém uma implementação massiva, com centenas de milhões de nodos funcionando diariamente", o Git; o sistema de controle de versão de código fonte distribuído, e o HTTP; Protocolo de Transferência de Hipertexto. De acordo com Benet (2014), "De longe, o HTTP é o mais bem-sucedido sistema de arquivos distribuído já implementado.". O protocolo alterou profundamente a forma como se navega e se vê a internet, além de ser o mais prevalente quando se diz respeito a transferência de arquivos na rede mundial. Ele continua a ser o mais utilizado por satisfazer a necessidade da maior parte dos usuários.

Com o passar do tempo, os locais físicos e lógicos em que as informações são armazenadas começaram a se centralizar em grandes conjuntos de servidores, quase todos esses conjuntos tendo como donos seletos grupos ou organizações. Um dos primeiros problemas que se pode identificar nesse processo de centralização de informação diz respeito ao controle: um número comparativo cada vez menor de organizações tendo controle sobre um número cada vez maior de informações. Outro problema inerente na maneira como se organizam e se acessam essas informações é: ela é fundamentalmente dependente de um sistema de armazenamento de dados centralizado, que está sujeito a vários riscos, que envolvem desde a privacidade de

informações, censura, até a perda temporária ou permanente de dados pelas mais diversas situações e condições. Por essas razões, se justifica a tentativa de buscar alternativas para tais sistemas e arquiteturas vigentes, de forma a minimizar ou eliminar tais problemas e preocupações.

Para tentar resolver essas questões, Juan Benet iniciou, em 2014, um projeto chamado Inter-planetary File System (em português, Sistema de Arquivos InterPlanetário, ou IPFS, em inglês). Em seu white paper, 'IPFS – Content Addressed, Versioned, P2P File System', Juan Benet visa substituir, pelo menos parcialmente, protocolos estabelecidos, como o HTTP, na maneira como os arquivos são armazenados, tratados e compartilhados na internet. IPFS toma emprestado vários conceitos de design das tecnologias mencionadas para proporcionar um sistema em que os dados são tratados de forma a incentivar a permanência na nas redes, locais ou globais, através de mecanismos como Distributed Hash Tables (DHT, tabelas hash distribuídas, em português), Merkle DAG, entre outros. Este texto, que tem foco no IPFS, analisa e compara dois sistemas distribuídos ponto a ponto, nomeadamente o IPFS e o Bittorrent, para verificar se o primeiro é fiel ao seu propósito de melhorar ou substituir a forma como se usa a internet atualmente.

Os objetivos deste estudo começam por elaborar e construir uma rede IPFS, com o fim de averiguar suas características para análise. Seguindo este passo, fazer um comparativo com outra rede ponto a ponto, o Bittorrent. Finalmente, analisar e comparar os dois sistemas, documentando os resultados.

2. Criptografia

Desde o crescimento estrondoso da internet a partir dos anos 90, aumentou-se a demanda para se desenvolver sistemas capazes de proteger não só a integridade dos dados que trafegam na rede, mas também a privacidade da conexão em si. Como afirma Terada (2008):

> "Sistemas de comunicação como a Internet são controlados por computadores em rede, pelos quais a informação trafega desde a origem onde ela está armazenada ou foi criada até o destino onde o solicitante da informação se encontra. Muitas vezes, há mais de um computador que repassa a informação entre a origem e o destino, principalmente quando estão geograficamente distantes."

Tendo isso em mente, é possível ter uma pequena ideia das possíveis ameaças e problemas que podem ser encontrados no mundo da comunicação digital atual. Do sigilo ou anonimato nas trocas de informação até possibilidade dessas informações serem receptadas, capturadas ou bloqueadas de alguma forma, as questões a serem contempladas no que diz respeito a segurança de redes criou um grande incentivo para estudar, pesquisar e desenvolver soluções voltadas a proteção de informações. Isso inevitavelmente levou ao uso da criptografia como forma de proteger dados.

3. Redes de computadores

Um dos grandes aspectos da vida moderna é, além do tecnocentrismo, onde as rotinas das vidas humanas são direcionadas a tecnologia, a constante conectividade. De acordo com Comer(2016):

> "As redes de computadores têm crescido explosivamente. A partir dos anos 1970, a comunicação via computador transformou-se em uma parte essencial de nossa infraestrutura. A ligação de computadores em rede é usada em cada aspecto dos negócios, incluindo propaganda, produção, transporte, planejamento, faturamento e contabilidade."

Cada vez mais, transferimos sistemas inteiros, alguns dos quais se estabeleceram nas sociedades humanas há séculos, para a rede. Comer(2016) continua:

> "Consequentemente, a maioria das corporações tem múltiplas redes. As instituições de ensino, em todos os níveis, do ensino fundamental à pós-graduação, estão utilizando redes de computadores para fornecer a estudantes e professores o acesso instantâneo a informações em bibliotecas online em todo o mundo. Órgãos governamentais em níveis federal, estadual e municipal utilizam redes, assim como as organizações militares."

A grande rede mundial, a Internet, se tornou parte essencial da economia em todos os setores. Como lembra Torres (2001), "Basta lembrar que grande parte das pessoas compra computadores hoje para ter acesso à maior das redes existentes – a Internet". Tudo vem, fundamentalmente, de uma necessidade intrínseca de compartilhamento de recursos. Essa é a principal razão pela qual as redes de computadores foram criadas.

3.1 Tipos de rede

Existem duas maneiras predominantes de se organizar logicamente uma rede de computadores: ponto a ponto e cliente-servidor. Torres(2001) explica que: "O primeiro tipo é usado em redes pequenas, enquanto o segundo tipo é largamente usado tanto em redes pequenas quanto em redes grandes." Essa é uma das principais razões pela qual o modelo cliente-servidor ser o predominante. Existem, até hoje, algumas limitações com relação a implementação e utilização do de tecnologias ponto a ponto quando se diz respeito a coisas como escalabilidade.

3.1.1 Redes ponto a ponto

A mais simples forma de conectar dois dispositivos. De acordo com Torres (2001), "na rede ponto a ponto, os micros compartilham dados e periféricos sem muita 'burocracia'." Isso denota a simplicidade deste design de rede.

Torres (2001) também lista as vantagens e desvantagens de uma rede ponto a ponto:

> "A grande vantagem é a facilidade de instalação e configuração, onde os próprios usuários podem configurar manualmente a que recursos os demais usuários podem ter acesso em seu micro. Essa vantagem, entretanto, traz algumas desvantagens, a principal delas é em relação à segurança de rede".

Uma simples rede ponto a ponto (p2p) não tem a estrutura necessária para oferecer segurança aos dispositivos conectados. Isso acontece tanto em termos de usabilidade entre dispositivos, quanto em termos de confiabilidade dos arquivos acessados. Este último problema se torna particularmente pernicioso em sistemas p2p com grandes quantidade de dispositivos conectados, pois se torna mais difícil verificar a autenticidade de todos os dados. Esses são alguns dos principais problemas que sistemas como o IPFS e o Bittorrent, apresentados neste papel, tentam endereçar, enquanto mantém a base de um sistema p2p.

4. Centralização, descentralização, distribuição

Um grande paradigma no mundo da tecnologia, no século 21, é o paradigma de centralização e descentralização de inúmeros aspectos da informática. Da forma como as instituições tecnológicas se relacionam e estruturam, até os fundamentos de como os dados são partilhados em uma rede, o debate diz respeito ao controle de informações.

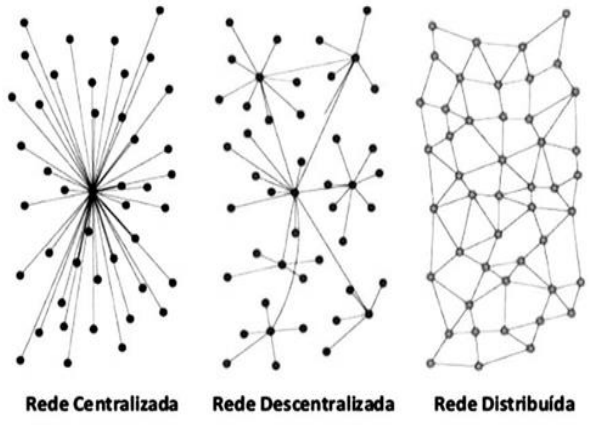


Figura 1. O paradigma das estruturas de redes

4.1. Redes e sistemas distribuídos

Antes mesmo do início da popularização da internet, grupos de pessoas envolvidas no que eventualmente se tornaria a internet já articulavam sobre os princípios que deveriam

reger as redes e as informações que nelas se encontram. Um desses princípios, segundo Pretto (2010), dita que “se deve sempre desconfiar da autoridade e, assim, estimulam-se procedimentos pouco burocráticos, com liberdade de circulação de informações e acesso a elas por qualquer um”. Os problemas inerentes das redes e sistemas centralizados, como, por exemplo, quem tem poder para controlar (seja no sentido de parar ou até alterar) o fluxo de informações, e as fragilidades de tais sistemas, em termos de segurança, levam engenheiros, programadores e hackers de tais sistemas a buscarem formas mais seguras, menos frágeis e mais descentralizadas de compartilhar informações. Tendo esses problemas em mente, Pretto (2010) define uma rede distribuída como “não-hierárquicas e descentralizadas, nas quais cada usuário, ou seja, cada computador na rede pode, ao mesmo tempo, exercer a função de cliente e servidor e, o mais importante, sem um gerenciamento central, faz com que a informação trafegue velozmente, possibilitando múltiplas conexões simultâneas.”.

4.2 Redes Cliente-servidor

O termo cliente-servidor é, segundo Wittgenstein (2011), um termo genérico para definir “qualquer arquitetura de aplicação que divide o processamento entre dois ou mais processos, constantemente entre duas ou mais máquinas”. Esta arquitetura é a forma mais comum de se organizar uma rede e está diretamente relacionada com redes centralizadas, onde o conteúdo é fornecido de um ponto central da rede. As tecnologias vistas neste texto tem como principal objetivo oferecer uma alternativa para tal modelo de arquitetura de rede.

5. IPFS

Na medida que as tecnologias que conectam a humanidade evoluem, novos paradigmas, questões e problemas aparecem pelo caminho. A internet, como é conhecida atualmente, funciona de forma satisfatória para a maioria das pessoas. Parker(2018) explica:

> Geralmente, quando se acessa um web site pelo seu navegador, o seu navegador rastreia a origem do servidor (ou servidores) que são repositórios centralizados e últimos para o conteúdo do web site. Ele manda um pedido do seu computador para o servidor de origem, onde quer que ele esteja no mundo, e esse servidor manda o conteúdo de volta para o seu computador. Esse sistema serviu a Internet por décadas, mas existe uma desvantagem: a centralização torna impossível manter o conteúdo online por mais tempo que os servidores de origem que o hospedam.

Isso demonstra uma falha fundamental nesse modelo predominante utilizado para o armazenamento e compartilhamento de informações, tanto na internet, quanto nas redes locais. Uma das propostas de Juan Benet, criador do InterPlanetary File System (IPFS), é um sistema que permita, pela própria natureza, que os dados permaneçam na rede.

Em 2014, Benet lançou o livro branco (documento proposta para o IPFS) descrevendo as bases do sistema. Benet(2014) descreve o IPFS como “um sistema de arquivos distribuído que busca conectar todos os dispositivos de computação sobre o mesmo sistema de arquivo.”

5.1 Problemas em escala mundial

O atual modelo (cliente-servidor) mais utilizado para compartilhar informações na internet ou em redes locais têm uma série de complicações que necessitam ser observadas para se ter um escopo mais completo dos problemas que o IPFS tenta endereçar. O que acontece com o processo de troca de informações quando componentes centrais numa rede centralizada param de funcionar? O acesso a essas informações se torna mais difícil ou até impossível. Mesmo que dispositivos finais tenham uma informação requisitada por outros dispositivos na rede, o acesso acaba sendo negado por falhas no próprio design da rede.

Além de possíveis falhas em redes, o controle central em redes centralizadas abre a possibilidade de censura por diversos agentes. Em 2017 o governo turco chegou a bloquear a enciclopédia online Wikipédia, de acordo com o site Reuters (2017), “citando a lei que o permite banir acesso à sites considerados obscenos ou uma ameaça a segurança nacional”. Uma infraestrutura onde as informações são armazenadas de forma centralizada são fundamentais para que este tipo de censura seja possível.

5.2 Design IPFS

Para endereçar as questões relevantes do modelo predominante utilizado nas redes mundiais atuais, foi preciso primeiro conceber o design por trás do IPFS. De acordo com o site IPFS.io, existem sete conceitos que são fundamentais para a existência do IPFS. São esses: Content Identifiers (CIDs), DNSLink, Hashes, InterPlanetary Name System (IPNS), Mutable File System (MFS), Pinning, UnixFS.

5.2.1 Content Identifiers

Um dos conceitos que mais diferencia o IPFS da maioria dos protocolos p2p é a ideia de um identificador único para o conteúdo que será compartilhado. Descrevem-se os CIDs da seguinte forma na documentação do IPFS no site ipfs.io (2017):

> “Um content identifier, ou CID, é um rótulo usado para apontar para um material no IPFS. Ele não indica onde o conteúdo está guardado, mas forma um tipo de endereço baseado no próprio conteúdo.”

Os CIDs são gerados utilizando-se de um método de hash criptográfico que usa o próprio conteúdo como base para o endereço. O resultado disso é que qualquer diferença no conteúdo dentro da rede IPFS gerará um CID diferente. Por causa disso, de forma independente do nodo/dispositivo em que o conteúdo é colocado na rede, o endereço será o mesmo.

O CID toma formas diferentes em diferentes versões. As duas versões, como é especificado no site IPFS.io, são CIDv0 e CIDv1, e “se uma CID tem 46 caracteres começando com ‘Qm’, é CIDv0.”

A partir do CIDv1, afirma o site IPFS.io, o CID “contém alguns identificadores que clarificam exatamente quais representações são usadas, junto ao hash do conteúdo em si”. A lista inclui:

- Um prefixo multibase. Que especifica a codificação usada para parte do CID.
- Um identificador de versão de CID
- Um identificador multicodec

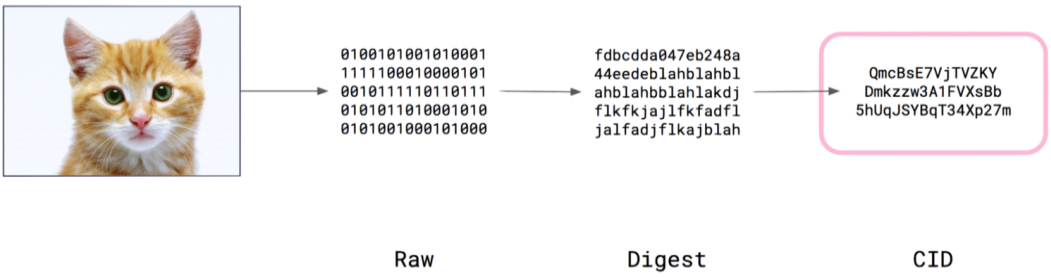


Figura 2. Criando a identificação de um conteúdo.

5.2.2 DNSLink

Uma das formas encontradas para fazer com que os nós das redes IPFS se comuniquem é, de acordo com site ipfs.io (2017), fazer um “mapa” de um domínio HTTPS dentro de um endereço IPFS. A conveniência em seu uso está no fato de um endereço DNSLink se parecer com um endereço IPNS, um dos conceitos fundamentais que serão vistos a seguir.

5.2.3 Hashes

Uma das condições críticas para o funcionamento do IPFS é a capacidade de esconder informações dos nós e usuários por questões de segurança e privacidade. Hashes, de acordo com o site ipfs.io (2017), “são funções que pegam um input arbitrário e retornam um valor de tamanho fixo”. Existem incontáveis formas de se fazer esse processo, sendo as mais comuns SHA-1 (usado no Git) e o SHA-256, que é usado pelo IPFS.

O IPFS utiliza hashes para suas chaves públicas (que mostram a identidade do objeto na rede) e privadas.

5.2.4 IPNS

O Inter-Planetary Name System (IPNS) é descrito no site ipfs.io (2017) como “um sistema para criação e atualização de links mutáveis para conteúdo IPFS. Já que os objetos no IPFS são endereçados por conteúdo, seus endereços mudam toda vez que o conteúdo do objeto muda”. A hash da chave pública é o nome IPNS de um objeto na rede.

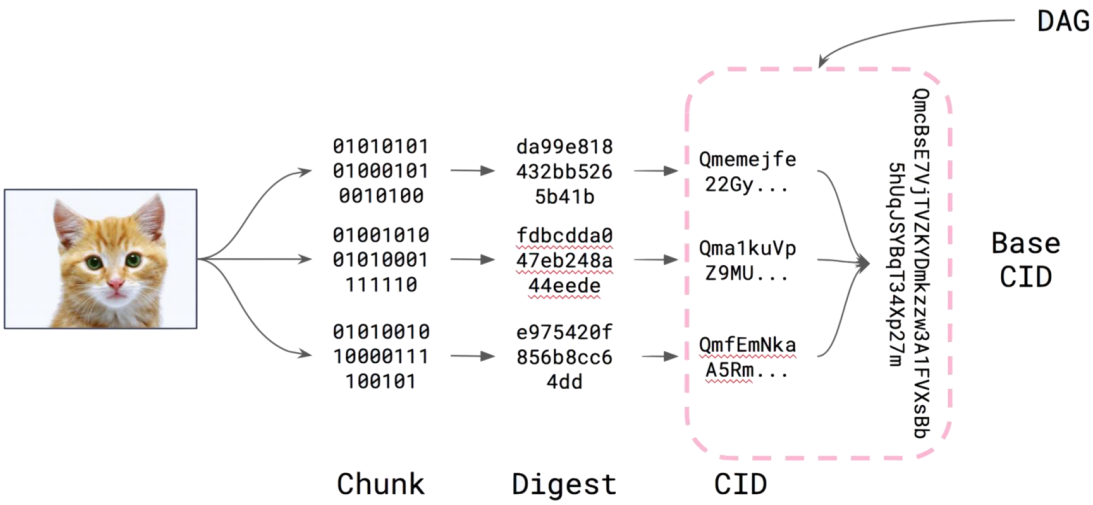
5.2.5 Merkle-DAGs

Lidar com os dados de forma distribuída e eficiente como é a proposta do IPFS, requer, para Singh (2019), que as seguintes características sejam observáveis:

- Endereçamento por conteúdo: onde todo o conteúdo disponível na rede possa ser identificado por si mesmo.
- A prova de adulterações: qualquer mudança no conteúdo altera sua identificação e pode ser identificada.
- Deduplicação: qualquer arquivo que já exista na rede só pode ser colocado uma única vez na rede.

Uma árvore Merkle ou árvore de dispersão é uma estrutura "para construir esquemas de autenticação e assinatura seguros derivados de funções hash", afirma Szydlo (2004). Essa estrutura permite gerar identidades únicas para objetos dentro de um sistema a partir dos próprios objetos. Isso é feito utilizando funções que geram um hash a partir dos dados em questão. Uma Merkle DAG (DAG significa "grafo acíclico direcionado") é uma árvore Merkle na qual, de acordo com Singh (2019), "não precisa ser balanceada e seus nós que não são folhas podem conter dados". Um DAG é uma forma de modelar uma sequência de informações que não tem ciclos. Uma analogia que pode ser feita para comparação é uma árvore de uma família.

É através deste mecanismo que os dados de um arquivo inserido dentro do IPFS são divididos em pedaços, que através de funções hash, são identificados por seus



conteúdos.

Figura 3. Gerando a identidade dos dados.

5.2.6 Mutable File System

O fato dos arquivos numa rede IPFS serem imutáveis (em cima de serem endereçados pelo conteúdo) pode tornar o processo de edição desses arquivos (e sua eventual inserção dentro do sistema) em algo complicado. De acordo com o site ipfs.io(2017), "Mutable File System (MFS) é uma ferramenta construída dentro do IPFS que lhe permite tratar arquivos como se trataria num sistema de arquivos baseado em nome

normal...". Isso significa permitir que os arquivos sejam manipulados (adicionados, excluídos, movidos e editados) de forma acessiva para usuários comuns.

5.2.7 Pinning

Uma das formas de se manter largas quantidades de dados é tratá-las como memória volátil ou cache. Num sistema de arquivos, seja qual for o tipo, isso significa que os dados não são permanentes e são armazenados apenas por um tempo determinado, geralmente relativamente curto. Pinning é o método pelo qual "é dito para o serviço IPFS que o dado é importante e não deve ser jogado fora", como afirma o site ipfs.io (2017). Pinning torna o arquivo no nodo do usuário "permanente" (até que ele diga o contrário), ficando totalmente isento do processo de "garbage collection" (coletor de lixo). Tal método traduz-se em funcionalidade daquilo que é um dos pilares do IPFS: a capacidade de acessar os dados da rede mesmo não estando conectado ou tendo conectividade intermitente à internet.

5.2.8 UnixFS

Qualquer sistema de rede distribuído que visa lidar com dados precisa levar em conta a possibilidade de grandes arquivos ou conjuntos de arquivos serem inseridos na rede e a forma como esses dados são armazenados e vistos. Para representar os dados, os conjuntos de dados inseridos e os metadados na rede IPFS, UnixFS foi criado e "é vagamente baseado na forma como os arquivos funcionam no Unix" afirmar o site ipfs.io (2017).

5.2.9 IPLD

O IPLD é tanto uma parte integral do IPFS quanto um projeto independente. É definido por Vaibhav (2019) como "um modelo de dados para a web endereçada por conteúdo". Através do IPLD, o IPFS consegue tratar todas as estruturas de dados ligadas por hash como um subconjunto de um espaço de informações unificadas. Isso torna as estruturas de dados utilizadas pelo sistema em informações que podem ser endereçadas universalmente, ou, como afirma Vaibhav (2019), "essas estruturas permitem que nós façamos com os dados o que as URLs fizeram com para as páginas web HTML".

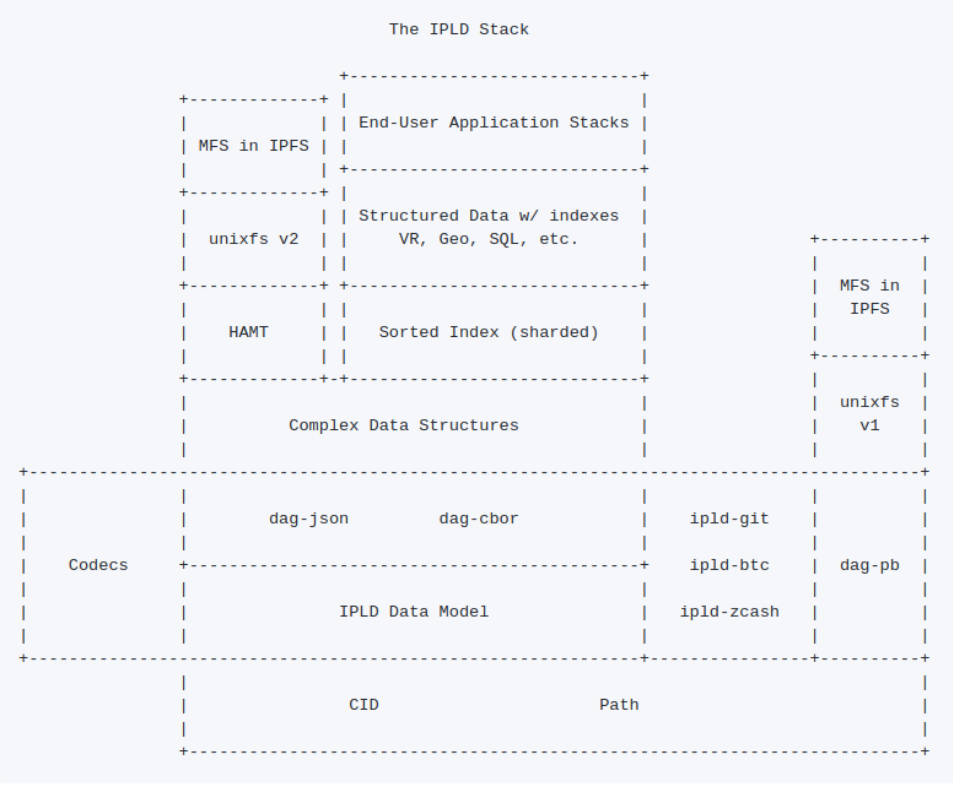


Figura 4. As espe

As especificações

cificações do IPLD.

5.3 Tratamento de dados no IPFS

Sendo um sistema ponto a ponto distribuído, o conteúdo no IPFS é “acessível através de pares que podem retransmitir informação ou guardá-la (ou fazer os dois), e esses pares podem estar localizados em qualquer lugar no mundo”, consta no site ipfs.io (2017). Para conseguir este feito, todo arquivo inserido na rede IPFS passa por um processo de divisão, e cada arquivo é dividido em partes menores de até 256Kbs.

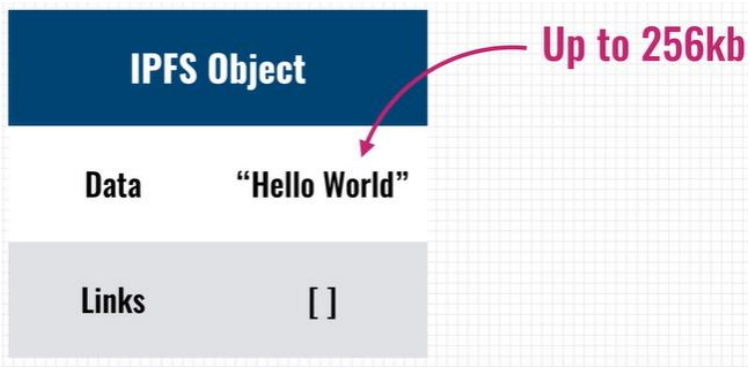


Figura 5. Objeto na rede IPFS.

Estas partes são colocadas individualmente dentro de um objeto IPFS, sendo cada objeto capaz de conter até 256kbs de dados.

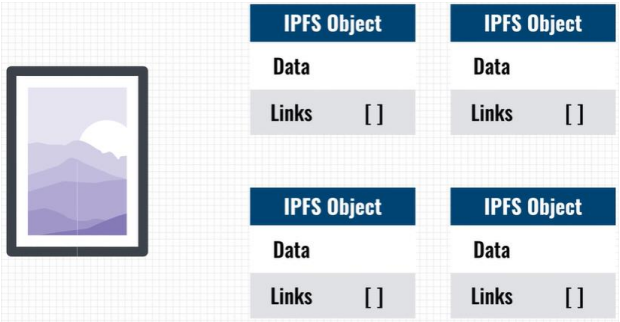


Figura 6. Objeto a ser dividido.

Objetos no IPFS podem conter links para outros objetos na rede. Após a divisão, o sistema cria outro objeto vazio que aponta para todos os objetos do arquivo dividido. Este objeto é usado como referência para a identificação na rede. Isso possibilita que o IPFS se assemelhe e possa ser usado como um sistema de arquivos também.

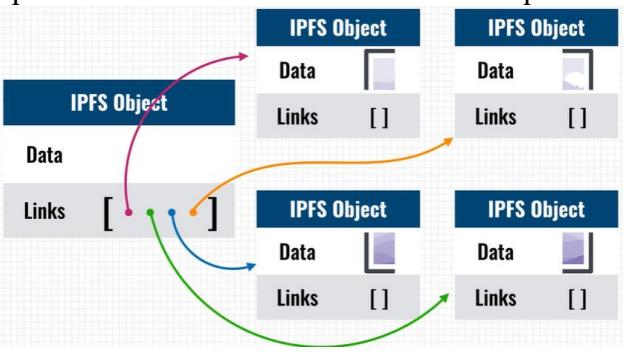


Figura 7. Objetos na rede IPFS.

O fato dos arquivos serem divididos e colocados em objetos faz com o IPFS tenha, por design, a característica de imutabilidade, isto é, arquivos inseridos dentro do IPFS não podem ser alterados. Para permitir que os dados no IPFS possam receber novas versões, o sistema é capaz de trabalhar com versionamento. Novas versões de um arquivo, que são reconhecidas pelo hash, são ligadas as versões anteriores através de um objeto "commit", que aponta para a nova versão e a anterior. Esta solução, de acordo com Benet (2014), foi inspirado pelo sistema Git.

5.4 Protocolo Bittorrent

A popularização da internet transformou a forma como as pessoas compartilham informações. Dentre as diversas inovações, o modelo peer-to-peer (p2p, ponto a ponto), que, segundo Johnsen, Karlsen e Birkeland (2005) já existia antes da criação do Bittorrent, passa a ganhar uma nova dimensão com este último. Bittorrent, um protocolo de compartilhamento de dados ponto a ponto é "de longe o mais popular programa ponto a ponto na história", afirmam Johnsen, Karlsen e Birkeland (2005).

5.4.1 Arquitetura Bittorrent

A arquitetura da rede do Bittorrent funciona, de acordo com Johnsen, Karlsen e Birkeland (2005), de um conjunto de quatro entidades, sendo estas: um arquivo estático de metainformações (o arquivo torrent), um tracker, um nó da rede que fornece o conteúdo (download original) e o nó destino (que faz o download). Este conjunto torna o processo de distribuição de arquivos mais fácil para fornecedores, pois divide esta tarefa com os outros usuários da rede.

Tradicionalmente, um dispositivo entra num swarm (conjunto de dispositivos trocando informações na rede Bittorrent) carregando um arquivo torrent em um cliente Bittorrent. O cliente contacta o tracker (cliente especial que mantém uma lista de computadores conectados) especificado no arquivo para permitir a troca entre os pares (dispositivos) na rede.

6. Metodologia de análise

Para que seja possível avaliar os sistemas em questão neste estudo de caso, é necessário estabelecer parâmetros nos quais tanto o IPFS quanto o Bittorrent possam ser observados. Para L. Wang e J. Kangasharju (2013), "métodos de medida podem ser divididos em diferentes categorias baseadas no sistema ou na metodologia utilizada". Isso significa que, a metodologia utilizada para o estudo de caso apresentado é dependente dos objetos de estudo, no caso, das tecnologias estudadas.

Para fins de averiguar e comparar o IPFS e Bittorrent, foram definidas métricas em um ambiente de pequena escala. Sendo assim, a análise apresentada neste texto tem caráter qualitativo, isto é, são medidas as qualidades das características analisadas dos dois sistemas.

6.1 Métricas

As métricas necessárias para analisar e comparar os dois sistemas são definidas pelas necessidades e demandas dos usuários, sejam estas de uma rede local ou da rede mundial de computadores, uma vez que os sistemas são destinados a eles.

As métricas definidas para análise são: usabilidade, velocidade e transparência.

6.2 Ambiente de análise

Sendo o ambiente de estudo em uma escala pequena, foi decidido que é necessário apenas dois dispositivos para efetuar os testes. A máquina utilizada o sistema operacional Windows 10 64bits para efetuar os testes e tem as seguintes configurações:

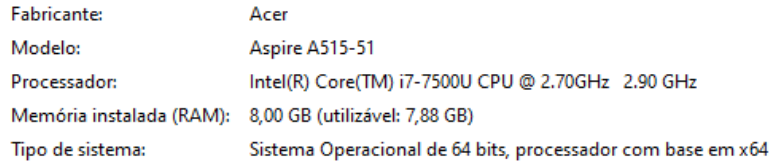


Figura 8. Especificações do ambiente de teste.

Dentro do dispositivo, será utilizada uma máquina virtual com o sistema operacional Linux Manjaro 18.1.2 64bits. O sistema operacional foi testado com as seguintes especificações de rede:

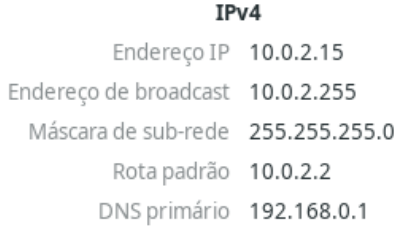


Figura 9. Especificações da conexão do Manjaro.

As ferramentas utilizadas para a utilização dos dois sistemas foram escolhidas ferramentas apropriadas para o ambiente. A implementação do protocolo IPFS escolhida foi a da linguagem de programação Go na versão 0.4.22. Os software o IPFS WebUI versão 0.9.7 para IPFS e o qBittorrent versão 4.1.6 para o Bittorrent.

6.3 Usabilidade

Uma das características que definem o sucesso (em termos de uso) de um software é a facilidade de seu uso.

Usabilidade, num contexto de software, diz respeito ao quão fácil é a utilização de um aplicativo ou sistema, como afirma Linda (2018). Para medir tal conceito, é preciso definir quais passos são necessárias para o uso de um software, foram escolhidas três características de uso comuns para usuário. A escolha passou por uma das questões fundamentais definidas por Nielsen (1990): a eficiência de uso, ou quantos passos são necessários para utilizar o software? Tendo em mente esta questão, as características são: instalação, acesso e compartilhamento.

A utilização do IPFS, até o momento dos testes e comparação, na versão testada, requer, antes da utilização de um cliente, a instalação do protocolo tanto no Windows quanto no Linux. No Windows é necessário ir ao site oficial do IPFS (ipfs.io) e baixar o instalador. É possível repetir o passo no Manjaro Linux, além da possibilidade de instalação pela linha de comando ou gerenciador de repositório. Isso é um passo extra para o seu uso, comparado ao uso do Bittorrent, que requer apenas um cliente para ser utilizado nos dois sistemas operacionais.

A principal implementação de interface de usuário do IPFS para Windows e Linux é o IPFS WebUI, que pode ser acessado num navegador web através do endereço 'localhost:5001/webui'. Esta interface permite que todas as funcionalidades do IPFS possam ser utilizadas de forma acessível. Sua alternativa é a linha de comando, que não foi escolhida para o teste pois o Bittorent também utiliza uma interface gráfica. O que age em benefício do caso de usabilidade para os dois sistemas.

Figura 10. Interface IPFS WebUI.

O protocolo Bittorrent tem inúmeros clientes que permitem sua utilização. Para este estudo de caso, foi escolhido o aplicativo qBittorrent, um cliente P2P multiplataforma, gratuito e de código aberto para o teste.

A interface dos dois permite a adição dos arquivos necessários para a ingressão destes na rede. No caso do IPFS, basta selecionar os arquivos para que estes sejam adicionados a rede.

	Instalação	Acesso	Compartilhamento
IPFS	2 passos	Requer conhecimento do hash pelas parte envolvidas	Seleção de arquivos, compartilhar Hash.
Bittorrent	1 passo	Requer link magnet ou arquivo torrent pelas partes envolvidas	Requer configuração do torrent, seleção de arquivos, criação do arquivo torrent, tracker definidos, compartilhar link magnet ou arquivo torrent

Tabela 1. Comparativo de usabilidade.

6.4 Velocidade

Para averiguar o quão rápido a troca de informações é feita nos dois sistemas, A velocidade é medida em KB/s (kilobytes por segundo), pois este é o padrão utilizado para mostrar a velocidade de transferência de dados nas duas ferramentas testadas. A velocidade de upload foi limitada em 250 KB/s (com pequenas flutuações acima deste número), pois esta é a capacidade máxima de upload permitida pela provedora de internet utilizada nos testes. Este também foi um fator definidor do tamanho do ambiente de testes.

É possível que exista diferença nas taxas de transferência entre os nodos, nos dois sistemas, devido à diferença no tratamento de dados entre os sistemas de arquivos tanto no Windows 10, que usa o NTFS, quanto no Manjaro Linux, que usa o Ext4. Nenhuma diferença de velocidade foi superficial e imediatamente notada durante o teste de velocidade. Por esta razão, averiguar a possível diferença entre os dois, tanto no que diz respeito a download, quanto a upload, não está incluída no escopo desta métrica.

Os testes foram feitos utilizando um arquivo MP4 de 806.4MB (megabytes) de tamanho. Antes de qualquer transferência, o arquivo foi copiado uma vez, sendo assim, uma cópia foi usada para o IPFS, e outra foi usada no Bittorrent, para análise e comparação. Os dois softwares escolhidos para os testes foram instalados tanto na máquina host (Windows 10) quanto na máquina virtual (Linux Manjaro).

6.4.1 Velocidade no Bittorrent

O primeiro teste foi feito com o Bittorrent, utilizando o qBittorrent. O tempo do experimento é contado no mesmo tempo em que a duração da transferência do arquivo começa e termina. Para avaliar a velocidade do download, foi anotada velocidade do download durante determinado tempo. O download, de um par da rede (host) até o outro (máquina virtual) demorou 51 minutos. Este tempo foi dividido em quatro pontos, onde em cada ponto é mostrado a velocidade do download. A velocidade do download variou entre 215KB/s e 254KB/s. A linha, na figura 11, representa a velocidade do download em um determinado tempo.

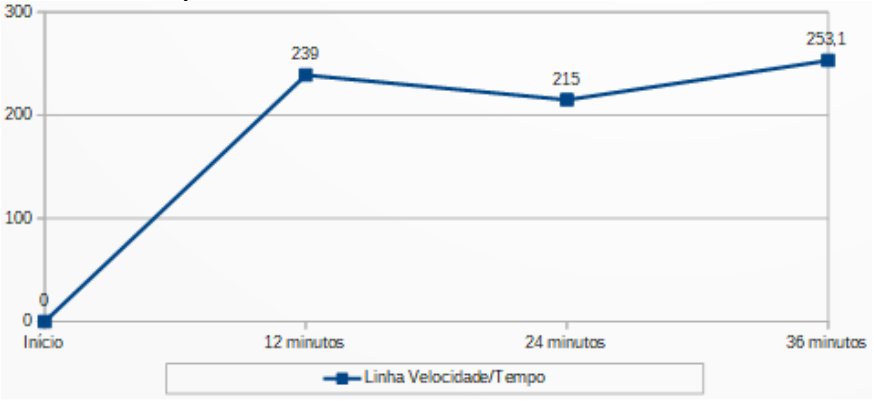


Figura 11. Gráfico de velocidade do qBittorrent.

6.4.2 Velocidade no IPFS

O segundo teste foi realizado com o IPFS, utilizando o IPFS WebUI tanto na máquina host quanto na máquina virtual para o teste. O tempo do experimento é contado no mesmo tempo em que a duração da transferência do arquivo começa e termina. Para

avaliar a velocidade do download, foi anotada velocidade do download durante determinado tempo. O download, de um par da rede (host) até o outro (máquina virtual) demorou 1 hora e 8 minutos. Este tempo foi dividido em quatro pontos, onde em cada ponto é mostrado a velocidade do download. A velocidade do download variou entre 201KB/s e 226KB/s A linha vermelha representa a velocidade do download em um determinado tempo.

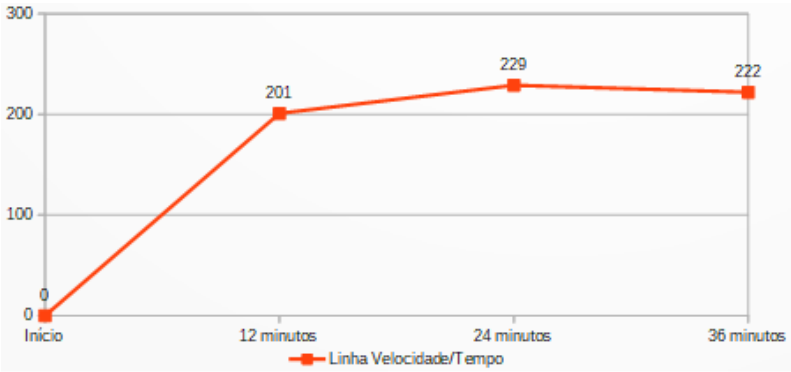


Figura 12. Gráfico da velocidade no WebUI.

6.5 Transparência

Transparência define-se, para Coulouris et al. (2017), como "a capacidade de ocultação do usuário e do programador da aplicação da separação dos componentes num sistema distribuído, para que o sistema seja percebido como um todo…". Isso tem profunda significância no desenvolvimento de um software. Razão pela qual esta métrica foi escolhida para esta análise.

6.5.1 Aspectos de transparência

Como todo tipo de sistema em TI, na medida em que seu uso cresce, sistemas distribuídos necessitam de uma definição de características comuns acuradas e bem elaboradas. Dessa necessidade, Coulouris et al. (2017) definiu características (que também definem desafios de design aos sistemas distribuídos). São eles: transparência de acesso, transparência de localização, transparência de concorrência, transparência de replicação, transparência de falha, transparência de mobilidade, transparência de performance, transparência de escalabilidade. A utilidade da transparência em sistemas distribuídos é, segundo Coulouris et al. (2017), a "separação de componentes num sistema distribuído, para que o sistema seja percebido como um todo ao invés de como uma coleção de componentes independentes". Os dois sistemas são avaliados a partir desses aspectos desta métrica.

6.5.1.1 Transferência de acesso

É definido como o aspecto responsável por permitir que recursos, sejam estes locais ou remotos, possam ser utilizados e acessados utilizando operações idênticas.

Apesar das diferenças em usabilidade, os dois sistemas passam de forma aceitável neste quesito. A natureza de sistemas ponto a ponto, combinada ao design de interface de usuário amigável, são características compartilhadas tanto pelo IPFS

WebUI quanto pelo qBittorrent, fazendo com que as operações de acesso aos recursos sejam de fácil compreensão, utilização e idênticas tanto para os pares que baixam dados, quanto para os que disponibilizam dados na rede.

6.5.1.2 Transparência de localização

Aspecto que permite o acesso e uso de recursos sem o conhecimento de sua localização física ou lógica. Este aspecto está diretamente ligado a privacidade, e esta última ligada a segurança.

A importância da privacidade no contexto de redes diz respeito a aspectos de segurança de dados. Um agente mal intencionado numa rede necessita de uma quebra de privacidade e segurança para ser bem-sucedido. Como afirma o Escritório de Avaliação de Tecnologia dos Estados Unidos (1994), “o uso impróprio de informações em rede pode afetar a privacidade, bem-estar e a vida de milhões de pessoas”.

Os dois sistemas têm uma falha fundamental no aspecto localização, pois os dois permitem a visualização da localização dos pares (nós ou usuários), mesmo que no caso do IPFS WebUI essa localização seja relativa, mostrando apenas o país em que os pares se encontram. No protocolo qBittorrent o problema é maior pois é possível ver o IP público de todos os pares no swarm (conjunto de computadores/pares) de um torrent, além do país em que se encontram. Além disso, é possível verificar também o tipo de conexão que o par está fazendo e a quantidade de informações que ele trocou.

País	IP	Porta	Conexão	Bandeiras	Cliente	Progresso	ade de download	cidade de upload	Baixado	Subido	Relevância
	172.92.132.40	10000	BT			0,0%			0 B	0 B	0,0%
	181.118.155.252	28173	BT			0,0%			0 B	0 B	0,0%
	113.190.213.165	3800	BT			0,0%			0 B	0 B	0,0%
	88.11.130.135	33233	BT			0,0%			0 B	0 B	0,0%
	193.186.9.83	10350	BT			0,0%			0 B	0 B	0,0%

Figura 13. Endereços IP no Bittorrent.

Adicionalmente, o IPFS se sobressai no quesito privacidade, pois tem suporte a criptografia dos arquivos que são adicionados no repositório, seja local (do próprio par) ou compartilhado. Contudo, esta funcionalidade não está implementada na versão testada do IPFS WebUI, estando disponível apenas na linha de comando.

6.5.1.3 Transparência de concorrência

Aspecto que permite diversos processos operarem concorrentemente de forma que não haja interferência entre eles. Sendo os sistemas testados designados para troca de informações em uma rede, nenhum dos dois demonstra qualquer problema com relação a capacidade de efetuar downloads e uploads ao mesmo tempo.

6.5.1.4 Transparência de replicação

Múltiplas instâncias de recursos podem ser usadas para aumentar confiabilidade e performance sem o conhecimento de possíveis réplicas pelos usuários através deste aspecto.

O IPFS identifica o conteúdo da rede pelo próprio conteúdo, isso significa que o conteúdo compartilhado por múltiplos pares só aparece uma única vez na rede. Isso é chamado de “deduplicação” e resolve por completo o problema de réplicas dentro da rede.

O Bittorrent sofre com problemas de replicação pois todo o conteúdo na rede é definido pelo arquivo torrent, que também ajuda a definir os participantes de um grupo de dispositivos que compartilham conteúdo (swarm). É possível então que múltiplos arquivos torrent contenham o mesmo conteúdo (réplicas). Isso, porém, não afeta a performance na rede.

6.5.1.5 Transparência de falha

Este aspecto administra a ocultação de faltas ou erros, permitindo o uso da aplicação distribuída independentemente de erros de componentes de software ou hardware.

Nenhum dos dois softwares testados apresentou qualquer tipo de falha de funcionalidade durante a fase de testes. Tendo isso em mente, foi feito um teste de conectividade para averiguar se alguma falha ocorre quando os download são interrompidos. O teste consiste em desconectar o dispositivo enquanto o download está em progresso, e então averiguar o que acontece e então reconectar os dispositivos.

Ao ser desconectado, o software que baixava um arquivo, tanto na rede IPFS, quanto na rede Bittorrent, não aponta qualquer sinal de falha, e não mostra erro de conectividade, apenas continua buscando por pares na rede. Isso significa que, em todos os quesitos deste aspecto, nenhum dos dois sistemas apontam qualquer falha de download, que foi resumido assim que a conectividade foi restaurada.

6.5.1.6 Transparência de mobilidade

Com este aspecto, a movimentação de recursos e clientes dentro do sistema pode ocorrer sem que as operações de usuários e programas sejam interrompidas.

Os dois sistemas testados permitem que todo o conteúdo de um par possa ser movido.

O IPFS tem uma falha neste aspecto, pois não permite que nada na rede seja modificado sem que a identificação do conteúdo na rede seja alterada como resultado desta ação. Isso significa que não é possível um usuário movimentar um arquivo localmente que estiver na rede, de um lugar para o outro (dentro de seu dispositivo), sem que sua disponibilidade na rede seja afetada. No caso dos clientes, o IPFS cria chaves públicas e privadas quando se cria um nó na rede. Essa chave pode ser removida e gerada novamente, permitindo uma nova identificação, mas isso só pode ser feito se a conexão com a rede IPFS for interrompida.

A identificação dos clientes no qBittorrent é feita pelo próprio IP público, o que o faz depender da conectividade para identificação. O conteúdo no software pode ser movido a qualquer momento pelo usuário, independentemente do status do torrent.

6.5.1.7 Transparência de performance

A função deste aspecto é fazer com que sistemas possam ser reconfigurados, na medida do necessário, para que a performance seja otimizada. A versão testada do IPFS não tem a capacidade de regular ou reconfigurar qualquer aspecto relacionado a performance. O qBittorrent permite regular a velocidade de download e de updload livremente, o que torna o protocolo Bittorrent superior ao IPFS em termos de performance.

6.5.1.8 Transparência de escalabilidade

Sistemas e aplicações podem expandir em escala (dentro ou fora da área de desenvolvimento) sem que a estrutura da aplicação ou do sistema sejam alteradas. Devido ao ambiente de pequena escala em que os testes foram realizados, este aspecto está fora do escopo desta análise comparativa qualitativa.

7 Conclusões

Os dois sistemas podem ser considerados estabelecidos em termos de uso. As funções examinadas do ponto de vista da usabilidade são satisfatórias, levando em conta os parâmetros da métrica estabelecida. Através da tabela 1, verifica-se que o Bittorrent se sobressai por ter menos passos de instalação, enquanto o IPFS tem vantagem por facilitar o compartilhamento de dados em si. Outro problema, subjetivo, do IPFS, é que os hashes usados para o compartilhamento são extensos e difíceis de lembrar. Uma possível solução para isso é utilizar o Domain Name System (DNS) para mascarar os links com hash do sistema.

O IPFS, além de oferecer a possibilidade de melhor privacidade e confidencialidade através da possibilidade de criptografia, tem a vantagem de ter um escopo maior do que simplesmente um protocolo para troca de dados ponto a ponto. O escopo do projeto, com todos os seus componentes, se estende até de um arquivo de sistema, até a possível substituição do protocolo HTTP como modelo mais utilizado na web.

Comparando as figuras 11 e 12, é possível averiguar que a velocidade em que os dados trafegam na rede IPFS é menor, porém mais constante (menor variação para mais ou para menos velocidade) do que no Bittorrent. Isso significa que o IPFS seja mais apropriado para compartilhar arquivos menores, pois seu tempo de download tende a ser menor. Tornando o Bittorrent mais apropriado para arquivos maiores, na comparação entre os dois. Apesar disso, o fator de deduplicação do IPFS pode tornar o armazenamento de grandes quantidades de arquivos mais viável para distribuição, dependendo da organização entre os pares.

Apesar de estar além do escopo desta análise, IPFS tem a pretensão de ser também um sistema de arquivo, mas pela velocidade do download efetuado nos experimentos feitos, especula-se que ele não seja rápido o suficiente para ser útil como sistema de arquivos.

Finalmente, para trabalhos futuros, a condução de testes e comparações de escalabilidade em larga escala pode aprofundar significativamente o escopo do estudo de tecnologias distribuídas como o IPFS e o Bittorrent. Testes comparativos qualitativos de segurança de dados dos dois sistemas também podem ser incluídos. Além disso, trabalhos futuros podem incluir testes de streaming (transmissão) de vídeo diretamente pelo IPFS, sem o uso aplicativos terceiros e comparar com funções similares em outros sistemas distribuídos. Adicionalmente, levando em conta que o projeto IPFS e a versão do IPFS usada neste estudo de caso estão em desenvolvimento constante (o projeto se encontra na sua fase beta), futuros estudos envolvendo uma versão de lançamento oficial do IPFS podem ter resultados consideravelmente diferentes dos vistos neste texto utilizando os mesmos parâmetros e métricas.

Referências

Benet, Juan (2014) “IPFS – Content Addressed, Versioned, P2P File System”, https://ipfs.io/ipfs/QmR7GSQM93Cx5eAg6a6yRzNde1FQv7uL6X1o4k7zrJa3LX/ipfs.draft3.pdf

Comer, Douglas E. (2016) “Redes de Computadores e Internet”, Brasil, Editora Bookman.

Coulouris, C., Dolimore, J., Kindberg, T., Blair, G. (2017) “Distributed Systems: Concepts and Design”, Addison Wesley, 5h edition.

Office of Technology Assessment (1994) “Information Security and Privacy in Network Enviroments”, Departamento de Bibliotecas de Documentos do Governo, Estados Unidos, Government Printing Office.

Parker, Andy (2018) “Cloudflare goes Interplanetary”, https://blog.cloudflare.com/distributed-web-gateway/ , Maio.

IPFS.io (2017) “IPFS Documentation” , https://docs.ipfs.io/ , Outubro.

Johnsen, J. A., Karlsen, E. L., Birkeland, S. S.. (2005) “Peer-to-Peer networking with BitTorrent”, Noruega, Universidade de Ciência e Tecnologia Norueguesa.

L. Wang e J. Kangasharju (2013) “Measuring large-scale distributed systems: case of bittorrent mainline dht”, In Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on, Estados Unidos.

Linda (2018) “The Ultimate Guide – Difference Between Usability and User Experience”, https://medium.muz.li/the-ultimate-guide-difference-between-usability-and-user-experience-afb757ded1c4 , Novembro.

Nielsen, J., Molich, R. (1990) “Heuristic Evaluation of User Interfaces”, http://cs.ashoka.edu.in/cs102/papers/heuristic-evaluation-of-user-interfaces-nielsen.pdf , Dezembro.

Pretto, Nelson (2010) “Redes colaborativas, ética hacker e educação”, http://www.scielo.br/pdf/edur/v26n3/v26n3a15 , Junho.

Terada, Routo (2008) “Segurança de Dados: Criptografia em rede de computador”, Brasil, Editora Blucher.

Torres, Gabriel (2001) “Redes de Computadores: Curso Completo”, Brasil, Axcel Books do Brasil Editora.

Vaibhav, Saini (2019) “Understanding IPFS in Depth: A Beginner to Advanced Guide”,

https://hackernoon.com/understanding-ipfs-in-depth-1-5-a-beginner-to-advanced-guide-e937675a8c8a , Outubro.

Reuters (2017) “Turkey blocks access to Wikipedia”, https://www.reuters.com/article/us-turkey-security-internet-wikipedia-idUSKBN17V06Q .

Singh, Niharika (2019) “IPFS and Merkle Forest“, https://hackernoon.com/ipfs-and-merkle-forest-a6b7f15f3537 , Outubro.

Szydlo, M. (2004) “Recent Improvements in the Efficient Use of Merkle Trees: Addicional Options for the Long Term”, http://www.rsasecurity.com/rsalabs/node.asp?id=2003 , Outubro.

Wennergren O., Vidhall M., Sörensen J. (2018) “Transparency analysis of distributed file systems: With a focus on InterPlanetary File System”, Suécia, Escola de Informática da Universidade de Skövde.

Wittgenstein, L. (2011) “Distributed Application Architecture”, https://web.archive.org/web/20110406121920/http://java.sun.com/developer/Books/jdbc/ch07.pdf , Dezembro.