Atribuição de Nomes Descentralizada: Estudo de Desempenho e Proposta de Otimização do IPNS

Francisco Vale, Pedro Ákos Costa, Yiannis Psaras, João Leitão

NOVA LINCS & DI/FCT/NOVA University of Lisbon & Protocol Labs
fx.vale@campus.fct.unl.pt pah.costa@campus.fct.unl.pt psaras@protocol.ai
jc.leitao@fct.unl.pt

Resumo A relevância dos sistemas descentralizados tem vindo a aumentar face às suas inerentes vantagens em relação às alternativas centralizadas. Novos sistemas descentralizados de grande escala têm sido desenvolvidos, como é o caso do *InterPlanetary File System* (IPFS), um sistema de ficheiros descentralizado, entre pares (do inglês *peer-to-peer*), cujo funcionamento se baseia na pesquisa de conteúdo a partir de um identificador único (CID, do inglês *content identifier*), que corresponde à aplicação de uma função segura de síntese (do inglês *hash*) ao conteúdo. Deste modo, caso seja realizada alguma alteração ao conteúdo, como é recorrente no caso de sites web com conteúdo mutável alojados no IPFS, o seu *hash* e, consequentemente, o novo CID serão diferentes, o que implica a partilha do novo CID, dificultando o acesso aos conteúdos. De forma a colmatar esta adversidade foi desenvolvida uma solução que, contrariamente à imutabilidade do endereçamento de conteúdo do IPFS, permite uma pesquisa sobre conteúdo mutável. O *Interplanetary Name System* (IPNS) recorre ao uso de apontadores que referenciam o conteúdo partilhado por um utilizador, de modo a que, face a uma alteração do conteúdo, o mesmo apontador fará referencia ao novo CID do conteúdo. Contudo este é um sistema pouco usado devido, maioritariamente, aos tempos de resposta elevados. Assim, no presente artigo, apresentamos um estudo de desempenho que tem por objetivo compreender onde estão as limitações do IPNS. Para tal aplicámos uma metodologia de estudo do IPNS, a partir da utilização de clientes IPFS modificados e, face aos resultados obtidos, propomos uma otimização do IPNS, mantendo a compatibilidade com versões anteriores.

Keywords: Web 3.0 · IPFS · IPNS · Sistemas entre-pares.

1 Introdução

A forma como armazenamos, recuperamos e partilhamos informação online é, cada vez mais, um aspeto crucial. Deste modo as soluções de partilha de conteúdo descentralizadas têm ganho popularidade devido às propriedades que apresentam, como a escalabilidade, a elevada tolerância a falhas e uma melhor resiliência

a censura, comparativamente a soluções centralizados, uma vez que não existe uma entidade central que armazene os dados. As aplicações destes sistemas têm sido notórias, por exemplo, no setor das criptomoedas, com o uso da *blockchain* [11], um sistema distribuído e descentralizado, onde se transfere o controlo e o poder de decisão de uma entidade central para uma rede distribuída. Este crescente interesse nas áreas descentralizadas levou ao seu rápido desenvolvimento e, consequentemente, à criação de diversos sistemas que incorporam a faceta descentralizada, como é o caso do *InterPlanetary File System*(IPFS)[5].

O IPFS é um sistema de ficheiros entre pares, que permite a partilha de ficheiros e que visa conectar dispositivos através de um sistema de ficheiros partilhado. Atualmente o IPFS serve de suporte a conteúdos, desde *websites* como uma cópia integral da Wikipédia, a imagens e até aplicações descentralizadas (*dApps*) que contam com mais de 50 milhões de utilizadores mensais. Para além disto, navegadores web como o *Brave* e o *Chrome* suportam o uso do IPFS nativamente. Este sistema de partilha de ficheiros descentralizado tem diversas vantagens, tais como o potencial para uma melhoria no tempo de acesso ao conteúdo, uma vez que este pode ser obtido através de utilizadores geograficamente mais próximos, uma melhoria na resiliência do sistema pois, uma vez que existem vários utilizadores a partilhar ficheiros, não existe um ponto de falha único, e também uma forte resistência contra a censura de conteúdo, dado que é difícil bloquear todos os utilizadores que partilhem um certo conteúdo. Como suporte às operações do IPFS é utilizada uma tabela de dispersão distribuída (DHT, do inglês *distributed hash table*) - baseado no protocolo *Kademlia* [10] -, para pesquisar e guardar apontadores de conteúdo. Deste modo, o conteúdo no IPFS é distinguido a partir de um identificador único (CID, do inglês *content identifier*), imutável, que representa o *hash* do próprio conteúdo. Assim, face a uma alteração no conteúdo o seu *hash* será distinto do anterior, sendo assim gerado um novo CID. A criação de um novo CID implica voltar a fazer um pesquisa na rede do IPFS, o que é problemático no caso de sites web dinâmicos, onde o conteúdo sofre alterações frequentemente ou, por exemplo, numa aplicação que permita a edição colaborativa de documentos visto que, após qualquer edição, seria gerado um novo CID que seria desconhecido pelos restantes colaboradores, fazendo com que os mesmos continuassem a aceder a uma versão obsoleta do documento.

De forma a ultrapassar tal obstáculo, foi concebida uma solução - o IPNS - que possibilita a partilha de conteúdo mutável, a partir da utilização de apontadores que indicam o CID do conteúdo. Assim, o *InterPlanetary Name System*[1] possibilita a partilha do um apontador, que é atualizado quando o conteúdo referenciado é alterado, fazendo referencia ao novo CID do conteúdo, mas permitindo um acesso contínuo por todos os utilizadores. Sempre que um apontador é atualizado, este é republicado na rede e, apesar da efemeridade da DHT, que elimina os conteúdos publicados após 24 horas[3], caso estes não sejam republicados, no período até cópias antigas do apontador serem removidas da rede, existe a possibilidade de coexistirem versões distintas do mesmo apontador, sendo que apenas uma destas aponta para o conteúdo mais recente. Isto leva à necessidade

de procurar várias versões do apontador, de modo a maximizar a probabilidade de obter a versão mais recente. Porém, a pesquisa por múltiplas versões do apontador introduz atrasos significativos no processo de obtenção de conteúdo, sendo esta uma possível explicação para a reduzida adesão ao IPNS, pelos utilizadores do IPFS. Torna-se assim essencial testar esta hipótese e identificar formas de melhorar a solução oferecida pelo IPNS. Contudo, dada a natureza descentralizada do IPFS, não existe um ponto único de observação, pelo que aferir o desempenho do sistema é uma tarefa não trivial.

De modo a endereçar este problema, apresentamos três contribuições principais no presente artigo: *i*) uma metodologia de avaliação do desempenho do IPNS, que consiste na distribuição de vários clientes pelo globo, de modo a minimizar o enviesamento da localização geográfica, estando cada um deles a atualizar registos IPNS com diferentes períodos, e onde cada cliente pesquisa continuamente pelos apontadores publicados pelos restantes clientes na rede do IPFS; *ii*) uma análise dos resultados obtidos com esta metodologia utilizando 8 clientes, de modo a compreender em detalhe a operação do IPNS; e *iii*) uma proposta de otimização do protocolo do IPNS, que seja compatível com versões anteriores, visto que a atualização dos milhares de clientes de IPFS geridos independentemente, de forma coordenada, é impossível.

O restante artigo está organizado da seguinte forma: a secção 2 fornece uma breve descrição do IPFS e IPNS, discutindo a operação de ambos e a sua integração. Na Secção 3 apresentamos a metodologia desenvolvida para este estudo do IPNS, assim como os objetivos da experiência e as hipóteses propostas. A secção 4 detalha o estudo experimental realizado e apresenta e discute os resultados obtidos. A secção 5 discute trabalho relacionado, e finalmente, a secção 6 conclui este artigo.

2 Preliminares

2.1 IPFS

O *InterPlanetary File System*[5] é um sistema distribuído, entre pares, de grande escala, que visa conectar dispositivos através de um sistema de ficheiros partilhado. O conteúdo neste sistema é imutável, sendo que cada parte individual de conteúdo (imagem, ficheiro, bloco, etc) tem associado um identificador único (CID, do inglês *content identifier*) que corresponde à aplicação de uma função segura de síntese (do inglês *hash*) ao próprio conteúdo, possibilitando assim assegurar a integridade dos dados. Os nós que compõem o IPFS têm também um identificador único denominado *peerId*, sendo este o *hash* da sua chave pública. Quando um utilizador publica algum conteúdo, o CID respetivo é propagado na rede do IPFS, mais especificamente numa DHT. Esta DHT é uma implementação do *Kademlia*[10], disponibilizada pela biblioteca *libp2p*[2], um conjunto de especificações de protocolos distribuídos e cuja materialização sobre a forma de uma biblioteca permite o desenvolvimento de aplicações descentralizadas. A partir desta DHT os nós conseguem organizar-se entre si utilizando o seu *peerId*, e partilham apontadores de conteúdo, a partir dos CIDs.

Os utilizadores do IPFS não guardam o conteúdo em si, mas sim registos de provedor (do inglês, *provider records*), que são apontadores para o nó que providencia o conteúdo (aquele que publicou ou re-publicou o conteúdo). Assim, para um utilizador publicar conteúdo na rede do IPFS, o mesmo tem de anunciar que providencia esse conteúdo, guardando na rede *provider records*. Estes *records* contêm um mapeamento de um CID para os *peerIds* dos nós que providenciam o conteúdo. Ao ser publicado na DHT, o *provider record* vai ser guardado nos k (no IPFS são $k = 20$) nós mais próximos (em distancia XOR) do CID do conteúdo.

Quando um utilizador procura um conteúdo no IPFS, é feita uma pesquisa inicial, otimista, em que se tenta encontrar o CID do conteúdo, perguntando a todos os vizinhos diretos do utilizador. Esta operação é efetuada pelo protocolo *Bitswap*[13]. Se o protocolo obtiver uma resposta positiva então prossegue-se para a transferência do ficheiro. Caso contrário recorre-se à DHT para encontrar o *provider record* do CID pesquisado.

2.2 IPNS

O conteúdo presente na rede do IPFS é imutável, uma vez que o conteúdo é identificado por um CID gerado a partir do *hash* do próprio conteúdo, pelo que se o conteúdo for alterado o *hash* resultante será diferente e consequentemente o CID também será diferente. Assim, o suporte a conteúdo atualizado frequentemente, como sites web ou documentos gerados com ferramentas de edição colaborativa, tornar-se-ia extremamente árduo, uma vez que cada alteração gera um novo CID que deve ser divulgado pelos utilizadores que acedem ao conteúdo, caso contrário estes observaram sempre uma versão antiga desse conteúdo. Deste modo foi necessário desenvolver uma solução para endereçar este aspeto do IPFS.

O *InterPlanetary Name System*[1] é um sistema que permite criar apontadores mutáveis, associados a um CID, permitindo assim a partilha de conteúdo mutável na rede do IPFS. Estes apontadores são denominados por *names* ou IPNS *names* e são representados pelo hash da chave pública do utilizador que publica o conteúdo. Cada IPNS *name* está associado a um IPNS *record* que contém campos cruciais para o correto funcionamento do protocolo: o campo *value* contém o CID associado ao respetivo IPNS *record*; *validity* contém a data de expiração do *record*; *sequence* mantém a versão do *record*; *TTL* (do inglês *time to live*) especifica o tempo máximo que o *record* pode ficar na *cache* de um utilizador; e, finalmente, *signature* que contém a assinatura criptográfica do publicador desse registo, fazendo com que apenas o utilizador que possui a chave privada possa publicar o *record*, o que torna o IPNS *name* auto-certificado.

Existem duas principais operações no IPNS, *i) Publish* e *ii) Resolve* que têm como objetivo publicar e procurar um *record*, na rede do IPFS, respetivamente.

i) A operação *Publish* tem duas fases. Inicialmente é preciso criar o IPNS *record* ou atualizar um existente. Para o criar é necessário definir todos os campos, onde o campo *value* terá o CID da última versão do conteúdo que vai ser associado ao IPNS *name*. Já o campo *sequence*, no caso criação do *record* tem

o valor de zero, caso esteja a ser atualizado o valor anterior é incrementado em uma unidade.

Após a definição dos campos e da assinatura do IPNS *record*, este será publicado na rede do IPFS de forma semelhante a qualquer outro conteúdo, sendo colocado nos 20 nós mais próximos na DHT do IPFS relativamente ao CID do IPNS *record*. No entanto, os IPNS *records* têm uma versão associada. Assim, se a cada republicação os *records* forem publicados em diferentes nós (devido a mudanças na topologia da DHT) existe a possibilidade de coexistirem versões distintas de um mesmo *record*.

ii) A operação *Resolve* permite obter um IPNS *record* da rede do IPFS. Contrariamente à pesquisa por conteúdo, não é utilizado o *Bitswap* por omissão, sendo a pesquisa realizada diretamente na DHT, a partir do CID correspondente ao IPNS *name*. Uma vez que não se sabe qual a versão mais recente do IPNS *record*, para aumentar a probabilidade de obter a versão mais recente, esta pesquisa termina apenas quando 16 nós retornam o *record* procurado.

3 Metodologia

3.1 Hipóteses / Objetivo

Apesar da utilidade do IPNS, existe uma baixa aderência a esta solução por parte dos utilizadores do IPFS. A demora na execução de operações, tanto na publicação como na obtenção dos *records*, pode ser uma explicação para esta reduzida adesão.

Por um lado, a publicação dos *records* é um processo demorado, uma vez que sempre que há alguma alteração no conteúdo apontado pelo *record*, tem que haver uma republicação, o que implica uma nova pesquisa pelos 20 nós mais próximos do IPNS *name*. Contudo, como o IPNS *name* permanece inalterado, uma vez que este representa o *hash* da chave publica do nó que o publica, então existe uma alta probabilidade de nos novos 20 nós mais próximos estarem contidos nós que haviam sido encontrados previamente. Por outro lado a espera pelas 16 respostas para um mesmo IPNS *record* na pesquisa, pode atrasar significativamente o processo de obtenção de conteúdo, não existindo uma justificação formal ou empírica para a utilização do valor 16.

Deste modo, torna-se relevante testar estas hipóteses, contudo, dada a natureza descentralizada do IPFS aferir aspetos de desempenho do sistema é uma tarefa não trivial, uma vez que não existe um ponto único de observação. Apresentamos de seguida, uma metodologia para o estudo deste sistema, com o objetivo de identificar e quantificar limitações do IPNS para guiar a sua futura optimização.

3.2 Arquitetura

De modo a conseguir estudar o desempenho do IPNS, na rede descentralizada do IPFS, desenvolvemos uma metodologia que, face à ausência de um ponto central de observação, se baseia na utilização de nós cliente modificados.

Para que os resultados não sejam enviesados pela localização geográfica de um só nó, propomos o uso de múltiplos nós em localizações distribuídas globalmente.

De forma a compreender os impactos das operações intrínsecas ao IPNS esta metodologia baseia-se na utilização das operações *Publish* e *Resolve*. Cada nó deve ser capaz de publicar IPNS *records*, enquanto os restantes procuram por esses mesmos *records* na rede do IPFS. De modo a possibilitar esta pesquisa, é necessário gerar um conjunto de chaves pública e privada para atribuir a cada um dos nós, para que seja possível conhecer previamente o IPNS *name* (*hash* da chave pública do nó) de cada um dos outros nós contidos na experiência. Para garantir o distanciamento entre os nós as chaves geradas não devem ser próximas, considerando a distância em XOR sobre os *peerId*.

3.3 Parâmetros

Para conseguir responder às hipóteses apresentadas de forma completa consideramos vários parâmetros relevantes.

Perfil do utilizador: um utilizador pode operar em modo cliente ou em modo servidor. Por omissão, os utilizadores são inicialmente anunciados em modo servidor, contudo, de modo a não responder a pedidos externos da DHT, o cliente deve ser inicializado em modo cliente.

Cache de *records*: quando um novo utilizador se junta à rede do IPFS a opção de manter os IPNS *records* em cache, durante o tempo presente no campo TTL do *record*, está ativa. Contudo se quisermos testar os tempos de obtenção do *record* na DHT é necessário desativar esta opção.

Publish: a frequência com que a operação de *Publish* é realizada pode ter impactos no dinamismo do sistema, uma vez que nas republicações mais distantes temporalmente há um maior *churn* [7] (variação nos nós presentes na rede), pelo que a probabilidade de o *record* ser guardado em nós diferentes dos anteriores é superior.

Resolve: na operação de *Resolve* é possível alterar o número de *records* pelos quais o sistema deve esperar até retornar o resultado ao cliente. Por omissão este valor é 16, contudo para testar o tempo de resposta de uma pesquisa por um IPNS *name* é útil variar este valor.

4 Trabalho Experimental

4.1 Configuração

A experiência foi realizada, durante aproximadamente duas semanas, utilizando a metodologia descrita, recorrendo a 8 clientes. Para diminuir o enviesamento geográfico as localizações selecionadas foram: Brasil, Austrália, Canada, Japão, Alemanha, Europa do Norte, África do Sul e Estados Unidos. De modo a conseguir executar os nós de cliente IPFS modificados em cada uma das localizações, foram usadas máquinas em centros de dados da *Azure*. Cada um dos nós correu

uma versão modificada e instrumentada da implementação do IPFS na linguagem *GO*, *Kubo*. As chaves pública e privada para cada nó foram geradas com uma distância, entre elas, superior a $1 * 10^{75}$, sendo a maior distancia possível, aproximadamente, $1 * 10^{77}$. Para além disto todos os nós foram iniciados em modo cliente, com o objetivo de diminuir a carga nos mesmos. Foi também removida a opção de *cache* de IPNS *records* nos clientes.

As alterações feitas ao *Kubo* prendem-se maioritariamente com a modificação e adição de *logs* para identificar os eventos de baixo nível da DHT, como o tempo de obtenção de cada um dos 16 *records*, a latência da pesquisa de cada um dos nós onde será colocado um *record* após uma operação de *Publish* ou ainda, qual foi o nó que retornou cada versão do *record* numa operação de *Resolve*.

Para além destas alterações foi desenvolvida uma aplicação que atua sobre o *Kubo* e que executa as operações de *Publish* e *Resolve*. Esta aplicação correu em cada um dos nós separadamente e teve como principal função repetir as operações relativas ao IPNS, de forma recorrente e paralela (entre elas).

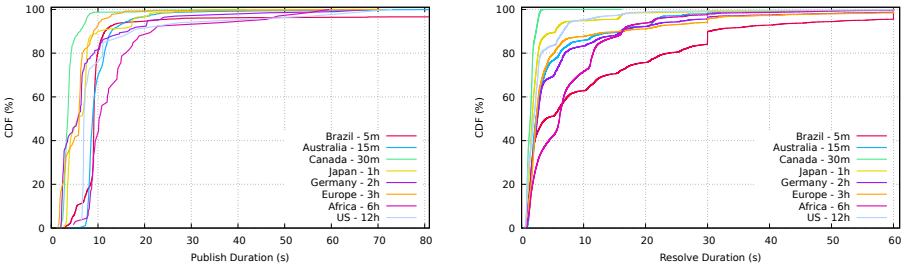
A operação de *Publish* foi executada com diferentes períodos em cada nó, de modo a compreender as implicações na distribuição dos *records* na DHT, face a períodos de republicação distintos. A distribuição destes períodos foi a seguinte: Brasil 5 minutos; Austrália 15 minutos; Canada 30 minutos; Japão 1 hora; Alemanha 2 horas; Europa do Norte 3 horas; África do Sul 6 horas; e Estados Unidos 12 horas.

A operação de *Resolve* foi executada, procurando em paralelo, cada um dos 7 IPNS *names* dos restantes clientes, recomeçando cada pesquisa a cada 30 segundos, aproximadamente. Para além disto a operação esperou sempre pela descoberta dos 16 *records* até retornar o resultado de forma a aferir quantos *records*, dos 16, são necessários para receber o mais recente. Foi também possível concluir as diferenças no tempo de resposta caso a pesquisa fosse finalizada após um número de respostas diferente de 16, uma vez que registamos o tempo de recepção de cada resposta individual.

4.2 Resultados

Tempo de resposta das operações: Começamos por analisar os tempos de resposta das operações de *Publish* e *Resolve*. A figura 1 representa a percentagem (eixo y) dos pedidos que finalizaram após um certo período temporal, em segundos (eixo x), observado pelos clientes nas diferentes localizações. A figura 1a apresenta os tempos de resposta do *Publish* e a figura 1b os de *Resolve*, sendo que apenas são contabilizadas operações com sucesso (98.78%).

A figura 1a mostra que cerca de 90% dos clientes conseguiram publicar *records* num tempo máximo de 25 segundos, o que representa um tempo de espera significativo. Para além disso observa-se que muitas operações de *Publish* nos nós localizados no Brasil e Austrália tiveram tempos de resposta mais elevados (300 e 125 segundos respetivamente). Estes tempos de resposta na generalidade são condicionados principalmente pela procura na DHT dos 20 nós mais próximos do CID do IPNS *record*.



(a) Publish (b) Resolve

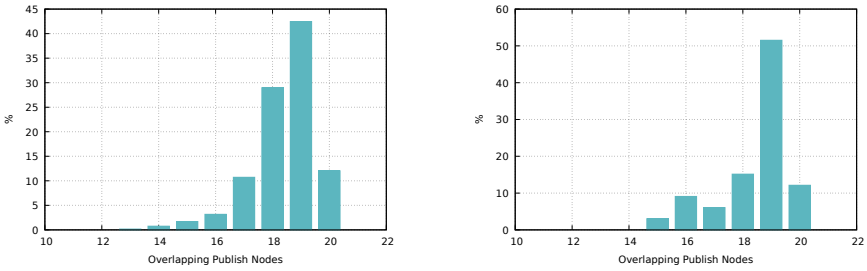
Figura 1: CDF com Tempos de Resposta

A figura 1b mostra que nenhuma operação de *Resolve* demora mais de 60 segundos, contudo isto deve-se a um *timeout* na DHT. É possivel também observar que as operações de *Resolve* feitas aos *records* do Brasil são as mais lentas, o que não parece relacionado com o período de publicação (este nó republica os *records* a cada 5 minutos).

Interseção dos nós entre republicações: Estudamos agora a interseção (ou sobreposição), dos nós onde os *records* são registados, entre republicações de cada IPNS *record*. Estudamos este fenómeno considerando cada cliente individualmente, uma vez que os períodos de republicação naturalmente afetam estes resultados.

A figura 2 apresenta a percentagem de republicações (eixo y) com número de interseções de nós (eixo x) face à publicação do *record* anterior, para diferentes períodos de republicação. A figura 2a contem os resultados obtidos para o nó com um período de republicação de 5 minutos (Brasil) e a figura 2b foca-se no nó com período de 12 horas (Estados Unidos).

Com um período de republicação de 5 minutos (figura 2a) observa-se que em mais de 40% das republicações, existe uma interseção de 19 nós e em cerca de 80% existe uma interseção que varia de 18 a 20 nós. Estes são valores bastante elevados mas face ao baixo período de republicação seria expectável pois o *churn*,



(a) 5 minutos (b) 12 horas

Figura 2: Interseção de nós, após republicação.

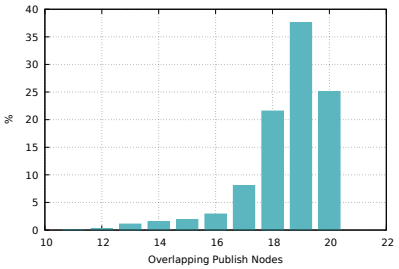


Figura 3: Média da interseção de nós, após republicação, para os diferentes períodos de republicação.

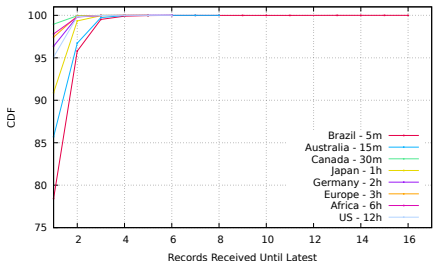


Figura 4: Número de respostas até receber o *record* mais atualizado.

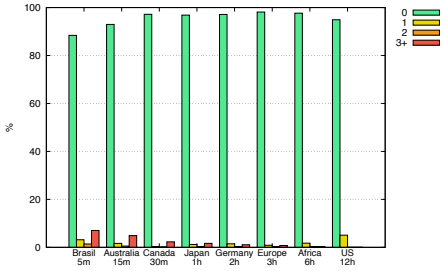
para um período de 5 minutos, é reduzido, logo a probabilidade de os nós mais próximos não variarem em 5 minutos é bastante alta.

Com um período de republicação de 12 horas (figura 2b) nota-se que mesmo com um elevado *churn*, existe uma interseção de 19 nós, em mais de 50% das republicações. E, mais uma vez, em cerca de 80% existe uma interseção que varia entre 18 e 20 nós. Dada a improbabilidade deste resultado, testámos esta propriedade para todos os outros períodos de republicação e verificámos que os resultados eram semelhantes.

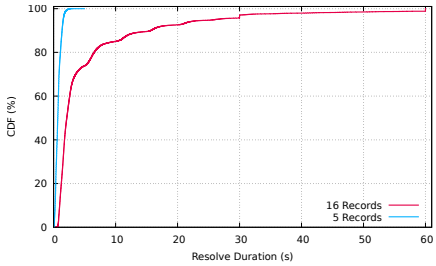
A figura 3 apresenta uma média, para os diferentes períodos de republicação testados, da percentagem de republicações (eixo y) com uma interseção dos nós (eixo x) onde é guardado o IPNS *record*. É possível voltar a observar os resultados obtidos na figura 2, onde cerca de 80% das republicações têm uma interseção de nós compreendida entre 18 e 20 relativamente à ultima publicação desse *record*. Estes resultados indicam que existe uma oportunidade de optimização do processo, permitindo aos nós manterem em cache os identificadores do nós onde publicaram cada registo da última vez, e iniciar um processo otimista de atualização desses registos durante a republicação, acompanhada de uma confirmação utilizando a DHT.

Versões dos records: Analisamos agora as versões dos *records* retornadas pelo sistema em resposta a uma operação de *Resolve*. Note-se que para conduzir esta parte do estudo recorremos ao facto de que os nossos clientes estão sincronizados e como tal podemos saber com alguma precisão qual a última versão publicada pelo nó responsável por cada IPNS *name*.

A figura 4 apresenta a percentagem de operações (eixo y) de *Resolve* que já dispunham de informação da versão do *record* mais atualizado após os diferentes números de respostas colacionadas (eixo x). A figura 5a apresenta uma média das percentagens (eixo y) dos *records* recebidos considerando a distância desse *record* à versão mais atual do mesmo. Em verde representa-se um *record* a uma distância de zero da versão mais atual (i.e., o *record* era o atual); a amarelo representamos *records* desatualizados numa única versão; a laranja *records* desatualizados em duas versões; e finalmente a vermelho *records* que se encontravam a uma distância



(a) Fração de respostas contendo a última verssão



(b) Considerando 5 respostas.

Figura 5: Conteúdo obtido nas respostas para a operação de *Resolve*.

de três ou mais da versão mais atual. Esta figura considera resultados para os diferentes períodos de republicação (eixo x).

Analisando a figura 4 podemos concluir que cerca de 100% das operações de *Resolve* receberam a versão mais recente do *record* a ser procurado, ao fim de de 4 respostas. É também possível observar que as respostas com *records* publicados pelos nós no Brasil e na Austrália obtém a versão mais recente do *record* ao fim de um maior número de respostas, o que é expectável visto que estes nós usam o período de republicação mais baixo, pelo que, a probabilidade de coexistirem mais versões distintas na rede é maior (ao longo da experiência o nó do Brasil publicou 4455 versões, o da Austrália 1559 versões e o dos Estados Unidos apenas 32 versões).

A figura 5a mostra no entanto que para qualquer período de republicação, cerca de 90% das respostas recebidas contém o *record* mais atualizado.

Finalmente, a figura 5b apresenta a comparação feita entre as latências de todas as operações de *Resolve* da figura 1b, se estas fossem configuradas para retornar uma resposta à aplicação após colecionarem 5 respostas, ao invés das 16 atualmente utilizadas. É possível verificar que, como esperado, caso o cliente tenha que esperar apenas por 5 *records*, o tempo de resposta da operação *Resolve* é muito inferior, comparativamente ao caso em que o cliente espera pelas 16 respostas, sendo que 100% das operações retornariam ao fim de apenas 5 segundos.

É de notar que a percentagem de operações feitas com sucesso, isto é, as operações de *Resolve* que devolveram ao cliente modificado a versão mais recente do *record*, são equiparáveis, sendo $98,78\%$ no caso de o cliente esperar pelas 16 respostas e $98,72\%$ no caso de esperar apenas por 5.

4.3 Discussão

Análise: Os resultados obtidos durante a nossa campanha de medições foram, de forma geral, positivos. Apesar dos tempos de resposta apresentados, na primeira secção dos resultados, serem altos, as secções seguintes revelaram observações relevantes para guiar a otimização do IPNS.

A partir das figuras 2 e 3 conseguimos compreender o dinamismo da operação de *Publish*. Estas figuras comprovam que, como o IPNS *name* associado ao *record* que está a ser publicado não se altera, uma vez que este representa o *hash* da chave pública do nó que executa o *Publish*, a probabilidade de os nós resultantes da pesquisa na DHT para armazenar o IPNS *record* apresentam uma elevadíssima sobreposição. Estes resultados são ainda corroborados pelas figuras 4 e 5a, uma vez que, como os *records* são publicados quase sempre nos mesmos nós, a quantidade de versões distintas de um *record* a coexistir na rede não é significativamente alta, pois os nós que tinham as versões obsoletas vão ser atualizados com a versão mais recente. Isto implica que quando é feita uma operação de *Resolve*, a quantidade de respostas com versões atualizadas será dominante, como demonstra a figura 5a.

Combinando este resultado com as observações nas figuras 4 e 5b podemos concluir que a espera pela receção de 16 *records* numa operação de *Resolve* é claramente desnecessária, uma vez que existe uma alta probabilidade de o nó receber uma versão do *record* atualizada, nas primeiras 4 respostas do *Resolve*.

Assim, analisando a figura 5b e, face às percentagens de sucesso em cada um dos casos, conseguimos concluir que é possível modificar o protocolo de forma a garantir que 100% destas operações terminam no máximo em 5 segundos.

Otimizações: Assim propomos duas principais otimizações ao IPNS: *i*) após a primeira operação de *Publish* realizada por um utilizador, os 20 nós onde foram guardados os *records* serão mantidos em *cache*, por um período de tempo alargado. Isto trará dois benefícios: *a*) vai acelerar o processo de republicação, uma vez que não será necessário procurar os nós mais próximos do IPNS *name* na DHT (sendo esta pesquisa feita assincronamente para as seguintes operações de *Publish*); e *b*) vai minimizar o número de *records* divergentes na rede, uma vez que o cliente tem informação sobre os nós que guardaram a versão anterior do *record*; e *ii*) diminuir o número de respostas, por omissão, antes de retornar o resultado ao cliente de 16 para um valor inferior, na ordem de 5, o que tornará o processo de resolução significativamente mais rápido.

5 Trabalho Relacionado

Medir e compreender o comportamento de sistemas descentralizados de grande escala, tem sido um tema importante que atraiu a comunidade científica, com vários trabalhos a realizar medições de sistemas reais entre-pares. O que torna desafiante a compreensão do funcionamento destes sistemas é a sua natureza descentralizada, tornando árdua a obtenção de pontos de observação onde seja possível recolher informações abrangentes sobre as interações entre componentes destes sistemas. Os estudos realizados pela comunidade cientifica cobrem vários sistemas, nomeadamente *BitTorrent* [12], *Gnutella* [14], *Tor* [9], *Napster* [14] e o IPFS [4,15,7,14,6].

Uma parte significativa deste trabalhos tem-se focado na medição de propriedades de redes descentralizadas. Por exemplo, o ecossistema *BitTorrent*, um

dos sistemas de partilha de ficheiros entre-pares mais populares, tem sido estudado extensivamente de modo a tonar mais transparente a sua estrutura e dinâmica [12]. Da mesma forma, a rede *Gnutella*, outra rede popular entre-pares mais antiga, foi analisada para capturar *snapshots* precisos de seu estado [14].

O estudo da estrutura e desempenho de redes entre-pares também foi estendido para outros tipos de redes. Por exemplo, a estrutura e o desempenho da rede *Tor*, uma rede popular de comunicação anónima, onde foram feitos estudos para compreender o seu nível de resiliência contra vários tipos de ataques [9].

No contexto do IPFS, Henningsen et al. desenvolveram um rastreador (do inglês, *crawler*), para gerar *snapshots* da rede do IPFS [8]. O rastreador é otimizado para pequenos tempos de rastreamento e é capaz de rastrear 50.000 nós em cerca de 4 minutos, em média. É uma ferramenta eficaz para avaliar o estado e a eficácia da rede, fornecendo dados importantes para pesquisas adicionais sobre o seu desempenho e resiliência. Já o trabalho apresentado em [6] foca-se em analisar *logs* de uma gateway pública do IPFS para identificar características da workload.

Estudos recentes [4,15,7] têm-se dedicado a analisar o IPFS de maneira complementar ao nosso trabalho. Esses estudos examinaram a distribuição de nós, em diferentes regiões geográficas, o tráfego do Bitswap dentro do IPFS e o desempenho geral do sistema. No entanto, o nosso estudo destaca-se por analisar mais profundamente o funcionamento do IPNS, tendo em consideração a relação geográfica entre os clientes do IPFS e os provedores de conteúdo, uma abordagem que os estudos anteriores não seguiram.

Em resumo, a medição de redes descentralizadas e entre-pares é uma área crítica de pesquisa, fornecendo insights valiosos sobre sua estrutura, desempenho e resiliência. Estas perceções são particularmente importantes para o desenvolvimento e melhoria de sistemas como o IPFS, que dependem dessas redes para o seu funcionamento.

6 Conclusões

Neste artigo apresentámos uma metodologia para o estudo do desempenho do IPNS que se baseia no uso de clientes do IPFS modificados, assim como os resultados da aplicação da metodologia desenvolvida e, finalmente, com base nos resultados propusemos otimizações ao IPNS, mantendo a compatibilidade com versões anteriores.

Como trabalho futuro pretendemos aplicar as otimizações propostas na versão corrente do IPNS e repetir este estudo de desempenho de forma a validar e quantificar a eficácia das mesmas. Para além disso pretendemos ainda repensar completamente o desenho do IPNS, de uma forma que não seja necessariamente compatível com a sua operação atual, de forma a aplicar as lições aprendidas com este estudo e obter um melhor desempenho com menor custo operacional.

Agradecimentos: Este trabalho foi parcialmente suportado pelo projeto Europeu TaRDIS (Grant Agreement No 101093006) e pela Fundação para a Ciência e Tecnologia através do laboratório NOVA LINCS (UIDB/04516/2020).

Referências

1. IPNS (InterPlanetary Name System) | IPFS docs. https://docs.ipfs.tech/concepts/ipns/, (Accessed on 06/11/2023)
2. libp2p. https://libp2p.io/, (Accessed on 06/11/2023)
3. Persistence | IPFS docs. https://docs.ipfs.tech/concepts/persistence/#garbage-collection, (Accessed on 06/11/2023)
4. Balduf, L., Henningsen, S., Florian, M., Rust, S., Scheuermann, B.: Monitoring data requests in decentralized data storage systems: A case study of IPFS. In: 2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS). pp. 658–668. IEEE (2022)
5. Benet, J.: IPFS-content addressed, versioned, p2p file system. arXiv preprint arXiv:1407.3561 (2014)
6. Costa, P.Á., Leitão, J., Psaras, Y.: Studying the workload of a fully decentralized web3 system: IPFS. In: Patiño-Martínez, M., Paulo, J. (eds.) Distributed Applications and Interoperable Systems. pp. 20–36. Springer Nature Switzerland, Cham (2023)
7. Daniel, E., Tschorsch, F.: Passively measuring IPFS churn and network size. In: 2022 IEEE 42nd International Conference on Distributed Computing Systems Workshops (ICDCSW). pp. 60–65. IEEE (2022)
8. Henningsen, S., Rust, S., Florian, M., Scheuermann, B.: Crawling the IPFS network. In: 2020 IFIP Networking Conference (Networking). pp. 679–680. IEEE (2020)
9. Loesing, K., Murdoch, S.J., Dingledine, R.: A case study on measuring statistical data in the Tor anonymity network. In: Proceedings of the Workshop on Ethics in Computer Security Research (WECSR 2010). LNCS, Springer (January 2010)
10. Maymounkov, P., Mazieres, D.: Kademlia: A peer-to-peer information system based on the xor metric. In: Peer-to-Peer Systems: First InternationalWorkshop, IPTPS 2002 Cambridge, MA, USA, March 7–8, 2002 Revised Papers, pp. 53–65. Springer (2002)
11. Nofer, M., Gomber, P., Hinz, O., Schiereck, D.: Blockchain. Business & Information Systems Engineering **59**, 183–187 (2017)
12. Pouwelse, J., Garbacki, P., Epema, D., Sips, H.: The bittorrent p2p file-sharing system: Measurements and analysis. In: Peer-to-Peer Systems IV: 4th International Workshop, IPTPS 2005, Ithaca, NY, USA, February 24-25, 2005. Revised Selected Papers 4. pp. 205–216. Springer (2005)
13. De la Rocha, A., Dias, D., Psaras, Y.: Accelerating content routing with bitswap: A multi-path file transfer protocol in IPFS and filecoin (2021)
14. Saroiu, S., Gummadi, P.K., Gribble, S.D.: Measurement study of peer-to-peer file sharing systems. In: Multimedia computing and networking 2002. vol. 4673, pp. 156–170. SPIE (2001)
15. Trautwein, D., Raman, A., Tyson, G., Castro, I., Scott, W., Schubotz, M., Gipp, B., Psaras, Y.: Design and evaluation of IPFS: A storage layer for the decentralized web. In: Proceedings of the ACM SIGCOMM 2022 Conference. p. 739–752. SIGCOMM '22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3544216.3544232, https://doi.org/10.1145/3544216.3544232