

ESTUDO DO IMPACTO DA DEDUPLIÇÃO DE DADOS SOBRE O DESEMPENHO DE SISTEMAS DE ARQUIVOS IMPLEMENTADOS SOBRE RECURSOS EXPLORADOS DE FORMA OPORTUNISTA

Armstrong Mardilson da Silva Goes¹ *, Francisco Vilar Brasileiro² †

28 de julho de 2012

Resumo

A duplicação de dados é um problema comum em sistemas de armazenamento. Este problema causa um aumento nos custos de armazenamento, impactando inclusive aplicações de e-Ciência, visto que muitas dessas aplicações manipulam grandes massas de dados. A deduplicação de dados é uma solução muito utilizada para este problema. No entanto, pouca atenção foi dada à implementação de deduplicação em ambientes oportunistas, que também trazem vantagens econômicas em relação ao melhor aproveitamento dos recursos disponíveis. Visto que as características de uma estratégia podem reduzir os ganhos conseguidos com a outra, é necessário entender como uma estratégia impacta a outra. O objetivo deste trabalho é estudar o impacto da deduplicação de dados sobre o desempenho de um sistema de arquivos distribuído implementado sobre recursos explorados de forma oportunista, em especial sobre o tempo de acesso aos arquivos. Para tal, um modelo que descreve a deduplicação no sistema de arquivos foi desenvolvido e duas análises envolvendo instâncias deste modelo foram realizadas. Tais análises revelaram que popularidade dos arquivos e o padrão de similaridades no sistema de arquivos são características que devem ser levadas em conta ao deduplicar.

Palavras-chave: Deduplicação de dados, Sistemas de arquivos, Ambientes oportunistas.

STUDY OF THE IMPACT OF DATA DEDUPLICATION ON THE PERFORMANCE OF FILE SYSTEMS IMPLEMENTED OVER RESOURCES EXPLOITED IN OPPORTUNISTIC WAYS

Abstract

Data duplication is a common problem in storage systems. This problem causes the increase of storage costs, impacting even e-Science applications, since many of these applications deal with large data sets. Data deduplication is a common solution to resolve this problem. However, little attention has been given to the implementation of deduplication in opportunistic environments, which bring economic advantages concerning the best utilization of the available resources. As the characteristics of one strategy may reduce the gains of the other, it is necessary to understand how each of the strategies impacts the other. The goal of this paper is to study the impact of the data deduplication on the performance of a distributed file system implemented over resources exploited in opportunistic ways, focusing on the file access time. To this end, a model which describes the deduplication in file systems was developed and two analyzes using instances of this model were performed. Such analyses revealed that file popularity and files similiary pattern in the file system are characteristics which must be considered when deduplicating.

Keywords: Data Deduplication, File Systems, Opportunistic Environments.

1 Introdução

Segundo Meyer et al.[2], sistemas de arquivos normalmente apresentam conteúdo redundante. Exemplos de conteúdo redundante são arquivos de configuração de sistema, *downloads* dos mesmos arquivos por membros de uma organização e discos de máquinas virtuais. As aplicações de e-Ciência também são afetadas por este problema, pois manipulam grandes quantidades de dados que muitas vezes são compartilhadas por múltiplos

*Aluno do Curso de Ciência da Computação, Unidade Acadêmica de Ciência da Computação, UFCG, Campina Grande, PB, E-mail: armstrong.goes@ccc.ufcg.edu.br

†Bacharel em Ciência da Computação, Doutor, Unidade Acadêmica de Ciência da Computação, UFCG, Campina Grande, PB, E-mail: fubica@computacao.ufcg.edu.br

usuários. Esta redundância representa um desperdício de espaço para armazenamento e consequente desperdício de recursos nas pesquisas.

Uma das soluções encontradas para resolver este problema é a deduplicação de dados. Deduplicar consiste em reduzir a quantidade de dados armazenados através da eliminação de conteúdo redundante. Ou seja, trechos semelhantes entre arquivos, ou mesmo arquivos inteiros, deixariam de ser armazenados mais de uma vez.

A deduplicação tem sido empregada principalmente em sistemas de arquivos para backup[6] e de arquitetura cliente-servidor, como o NFS[3]. Por sua vez, pouca atenção foi dada aos problemas encontrados quando se utiliza deduplicação em ambientes oportunistas.

Um sistema de arquivos oportunista tem como principal característica a utilização do espaço ocioso do disco das máquinas dos usuários. Em adição à redução de custo devido a utilização deste espaço ocioso, é possível tornar o sistema mais eficiente, uma vez que requisições de acesso aos dados podem ser atendidas localmente. Um exemplo de sistema deste tipo é o BeeFS[4].

O BeeFS, sistema de arquivos distribuído desenvolvido no Laboratório de Sistemas Distribuídos da Universidade Federal de Campina Grande, combina utilização oportunista do disco das máquinas dos usuários com uma política de *cache* que satisfaz grande parte das requisições por metadados para tornar mais eficiente as operações sobre os arquivos.

Quando consideramos o emprego de deduplicação em um sistema oportunista, o ganho de desempenho devido as requisições serem atendidas localmente pode ser afetado, visto que o processo de deduplicação remove alguns dos arquivos do sistema e os usuários destes arquivos removidos passarão a operar sobre os dados remotamente.

Considerando estes aspectos, surge o problema de analisar o impacto que a deduplicação de arquivos tem sobre o tempo de acesso aos arquivos em um sistema de arquivos distribuído implementado sobre recursos explorados de forma oportunista.

2 Materiais e métodos

O método escolhido para resolver o problema descrito acima envolve duas etapas. A primeira etapa consiste em desenvolver um modelo matemático da deduplicação em um sistema de arquivos distribuído oportunista. Concluída a primeira etapa, a primeira fase da análise é realizada sem instanciar o modelo, levando em conta somente os relacionamentos entre as variáveis descritas. A segunda etapa consiste em instanciar o modelo, atribuindo valores para a realização da segunda fase da análise.

O modelo

Consideramos um sistema de arquivos distribuído composto por um conjunto de máquinas M . Este sistema armazena um conjunto S de arquivos. Consideramos que $|S| \neq 0$ e $|M| \neq 0$. Um arquivo f , $f \in S$, é armazenado em m , $m \in M$, se e somente se, $\mu(f) = m$. Portanto, μ é uma função de alocação dos arquivos. Um arquivo f , $f \in S$, tem o mesmo conteúdo de um arquivo g , $g \in S$, se e somente se $\lambda(f, g) = 1$. Caso contrário $\lambda(f, g) = 0$. λ é, então, uma função binária que caracteriza a similaridade entre os arquivos no sistema de arquivos.

O conjunto dos arquivos S pode ser expresso como a união dos subconjuntos de arquivos duplicados, ou seja,

$$S = R_1 \cup R_2 \cup \dots \cup R_j, j \leq |S|,$$

onde todos os subconjuntos R_i são disjuntos

$$\forall(i, j)(i \neq j \rightarrow R_i \cap R_j = \emptyset);$$

os subconjuntos R_i agrupam arquivos de mesmo conteúdo

$$\forall(f, g)((f \in R_i \wedge g \in R_i) \rightarrow \lambda(f, g) = 1); \text{ e}$$

dois arquivos que pertencem a subconjuntos diferentes têm conteúdos diferentes

$$\forall(f, g)((\lambda(f, g) = 0 \wedge f \in R_i \rightarrow g \notin R_i)).$$

A quantidade de acessos a um arquivo f , $f \in S$, a partir de uma máquina m , $m \in M$, é dado por $\gamma(f, m)$. Se f é acessado a partir de m , então $\gamma(f, m) \geq 1$. Caso contrário $\gamma(f, m) = 0$. A partir da última definição deriva-se a definição de popularidade de um arquivo. A popularidade de um arquivo f , $P(f)$, é a razão entre a quantidade de acessos a f e a quantidade de acessos a todos os arquivos no sistema de arquivos em que f está armazenado:

$$P(f) = \frac{\sum_{m \in M} \gamma(f, m)}{\sum_{m \in M} \sum_{f' \in S} \gamma(f', m)}$$

Tipicamente, sistemas de arquivos oportunistas armazenam dados e metadados em componentes distintos. Uma vez que a deduplicação não afeta os metadados, operações sobre estes, tais como criação e deleção de arquivos não são consideradas. Portanto, consideramos que um acesso é definido como uma operação de escrita ou leitura.

Nesta análise trataremos o tempo que os acessos levarão para serem concluídos como constantes. Este tempo depende do tipo do acesso: se este é local, então o tempo do acesso é o tempo de acesso local (T_l); se é remoto, então seu tempo é o tempo de acesso remoto (T_r).

Formalmente, se $\gamma(f, m_1) \neq 0$, $\mu(f) = m_2$ e $m_1 \neq m_2$, então o tempo de acesso a f a partir de m_1 é igual a T_r . Se $m_1 = m_2$, o tempo de acesso ao arquivo a partir de m_1 é igual a T_l .

Como descrito anteriormente, um arquivo pode ser acessado tanto localmente, no caso de acessos feitos a partir da máquina em que ele está alocado, quanto remotamente, no caso de acessos feitos a partir de outras máquinas. A probabilidade de um arquivo f ser acessado localmente é $\Pi(f)$. Considerando que existem acessos remotos e locais, e estes têm tempo diferentes, o tempo médio de acesso a um arquivo f , $\tau(f)$, é dado por:

$$\tau(f) = T_l \times \Pi(f) + T_r \times (1 - \Pi(f))$$

A partir destas considerações, o tempo médio para todos os arquivos do sistema, $\tau(S)$, é dado por:

$$\tau(S) = \frac{\sum_{f \in S} \tau(f) \times P(f)}{|S|}$$

O restante desta seção descreve o conjunto de arquivos do mesmo sistema de arquivos após a deduplicação ser realizada. O desenvolvimento desta descrição visa à comparação dos sistemas de arquivos não-deduplicado e deduplicado para a análise do impacto da deduplicação.

Uma deduplicação de um conjunto de arquivos é um subconjunto deste sistema de arquivos com parte de sua redundância eliminada. Seja S_d uma deduplicação de S , que elimina toda a redundância em S . S_d é um conjunto de arquivos no qual todos os arquivos têm conteúdos diferentes, ou seja:

$$\forall f, g ((f \in S_d \wedge g \in S_d \wedge f \neq g) \rightarrow \lambda(f, g) = 0)$$

Além disso, S_d contém um, e somente um, arquivo de cada subconjunto R_i

$$\forall R_i, 1 \leq i \leq |S_d|, \exists f | f \in R_i \wedge f \in S_d$$

Note que, como S_d não contém arquivos semelhantes entre si, não poderia conter mais do que um arquivo de cada R_i , pois todos os arquivos de cada R_i são semelhantes, o que levaria S_d a também ter arquivos semelhantes entre si.

A cópia não-removida no conjunto R_i é:

$$f_i | f_i \in R_i \wedge f_i \in S_d$$

O arquivo de conteúdo igual a f que não é removido pela deduplicação é $\Delta(f)$,

$$\Delta(f) = g | g \in S_d \wedge \lambda(f, g) = 1$$

O conjunto S_r de arquivos removidos de S é definido como:

$$S_r = S - S_d$$

E o sistema deduplicado, S' , é:

$$S' = S_r \cup S_d$$

A função de probabilidade $\Pi(f)$ é expressa como:

$$\Pi(f) = \begin{cases} \frac{\gamma(f, \mu(f))}{\sum_{m \in M} \gamma(f, m)} & \text{se } f \in S \vee f \in S_d \\ \frac{\gamma(f, \mu(\Delta(f)))}{\sum_{m \in M} \gamma(f, m)} & \text{se } f \in S_r \end{cases}$$

Esta função descreve o seguinte: se o arquivo f não foi removido, pertencendo ele a S_d ou a S , então a

probabilidade de ele ser acessado localmente é a probabilidade de ele ser acessado a partir da máquina em que ele está alocado ($\mu(f)$). Se o arquivo f foi deletado, então a única versão de seu conteúdo que permanece no sistema de arquivos é $\Delta(f)$. Consequentemente os acessos ao conteúdo de f só serão locais se oriundos da máquina que armazena $\Delta(f)$, ($\mu(\Delta(f))$).

Definimos inicialmente o impacto sobre o tempo médio de acesso aos arquivos, I , como a diferença entre os tempos médios de acesso no sistema depois e antes da deduplicação. Utilizamos esta definição para simplificar a primeira fase da análise. Usando esta definição:

$$I = \tau(S') - \tau(S)$$

chega-se a:

$$I = \frac{(T_r - T_l) \times \sum_{f \in S} (\gamma(f, \mu(f)) - \gamma(f, \mu(\Delta(f))))}{|S| \times \sum_{m \in M} \sum_{f \in S} \gamma(f, \mu(f))}$$

3 Resultados e discussão

3.1 Análise Algébrica

Analizamos algebricamente os seguintes casos:

1. Sistema sem redundância.
2. Impacto sobre os acessos a um arquivo removido que, no sistema não-deduplicado, estava alocado na mesma máquina do arquivo preservado pela deduplicação.
3. Impacto sobre os acessos a um arquivo removido que, no sistema não-deduplicado, não estava alocado na mesma máquina do arquivo preservado pela deduplicação em um sistema sem compartilhamento de arquivos.
4. Mesmas características do caso 3, mas com compartilhamento de arquivos

3.1.1 Sistema sem redundância

Caso f não tenha duplicatas, $f = \Delta(f)$ e $\gamma(f, \mu(f)) - \gamma(f, \mu(\Delta(f))) = 0$. Portanto arquivos que não têm duplicatas não adicionam valor algum ao impacto. Consequentemente, se não houver duplicatas no sistema o impacto é igual a 0.

3.1.2 Impacto sobre os acessos a um arquivo removido que no sistema não-deduplicado estava alocado na mesma máquina do arquivo preservado pela deduplicação

Caso $f \neq \Delta(f)$ e $\mu(f) = \mu(\Delta(f))$, $\gamma(f, \mu(f)) - \gamma(f, \mu(\Delta(f))) = 0$. Neste caso, os acessos a f a partir de uma máquina m , $m \neq \mu(f)$, continuarão remotos e os acessos a f a partir de $\mu(f)$ continuarão locais pois acessarão $\Delta(f)$.

Chega-se à conclusão de que somente os acessos aos arquivos removidos pela deduplicação e que não estavam alocados na mesma máquina do arquivo preservado pela deduplicação são impactados pela deduplicação.

3.1.3 Impacto sobre os acessos a um arquivo removido que no sistema não-deduplicado não estava alocado na mesma máquina do arquivo preservado pela deduplicação, em um sistema sem compartilhamento de arquivos

Caso $f \neq \Delta(f)$, $\mu(f) \neq \mu(\Delta(f))$ e não havendo compartilhamento de arquivos, $\gamma(f, \mu(\Delta(f))) = 0$ e $\gamma(f, \mu(f)) - \gamma(f, \mu(\Delta(f))) = \gamma(f, \mu(f))$, ou seja, o impacto para f é proporcional à quantidade de acessos a f , e, consequentemente, à popularidade de f . O impacto será mínimo quando o somatório $\sum_{f \in S} (\gamma(f, \mu(f)))$

for mínimo. Isto acontece quando os arquivos que contribuem para o somatório ($f \neq \Delta(f)$, $\mu(f) \neq \mu(\Delta(f))$) tiverem popularidade mínima. O impacto aumenta a medida que a popularidade dos arquivos deletados pela deduplicação aumenta em relação à popularidade dos arquivos preservados pela deduplicação. Deste modo, a popularidade dos arquivos é um fator que deve ser levado em conta na deduplicação.

3.1.4 Impacto sobre os acessos a um arquivo removido que no sistema não-deduplicado não estava alocado na mesma máquina do arquivo preservado pela deduplicação, em um sistema com compartilhamento de arquivos

Caso $f \neq \Delta(f)$, $\mu(f) \neq \mu(\Delta(f))$ e havendo compartilhamento de arquivos, os acessos a f serão divididos entre as máquinas do sistema de arquivos. Parte dos acessos a f será de acessos locais (aqueles feitos a partir de $\mu(f)$), e parte será de acessos remotos. Após deduplicar, os acessos feitos a partir de $\mu(\Delta(f))$ se tornarão locais,

enquanto que os acessos feitos a partir de $\mu(f)$ se tornarão remotos. Os acessos feitos a partir das outras máquinas permanecerão remotos e não contribuirão para o impacto. Deste modo, se $\gamma(f, \mu(f)) > \gamma(f, \mu(\Delta(f)))$ haverá mais acessos transformados em remotos do que em locais, e o impacto é positivo. Se $\gamma(f, \mu(f)) < \gamma(f, \mu(\Delta(f)))$, a situação se inverte e o impacto é negativo. Esta última é uma situação improvável, pois espera-se que em um sistema de arquivos o acesso proprietário seja mais comum do que o acesso compartilhado. Em outras palavras, havendo compartilhamento de arquivos, o impacto será proporcional à diferença entre a quantidade de acessos proprietários e a quantidade de acessos originados do compartilhamento de arquivos.

3.2 Análise de instâncias do modelo

Na primeira fase da análise, por simplificação, tratamos o impacto como a diferença entre os tempos médios de acesso no sistema depois e antes da deduplicação. No entanto, como para este caso o impacto é uma variável relacionada a tempo, concluímos que esta definição não é informativa o bastante. O mesmo valor de I pode representar tanto um impacto significativo como um impacto desprezível, dependendo da proporção entre I e o tempo de acesso no sistema antes da deduplicação.

Desejávamos expressar o quanto o tempo médio de acesso no sistema deduplicado é maior do que o tempo médio de acesso no sistema não-deduplicado. Deste modo, definimos na segunda fase da análise o impacto como a razão entre o tempo médio de acesso no sistema deduplicado e o tempo médio de acesso no sistema não-deduplicado. Formalmente,

$$I = \tau(S')/\tau(S).$$

Fazendo as substituições e as simplificações necessárias chega-se a:

$$I = \frac{\sum_{f \in S_d} [T_l \times \gamma(f, \mu(f)) + T_r \times \sum_{m \in M - \{\mu(f)\}} \gamma(f, m)] + \sum_{f \in S_r} [T_l \times \gamma(f, \mu(\Delta(f))) + T_r \times \sum_{m \in M - \{\mu(\Delta(f))\}} \gamma(f, m)]}{\sum_{f \in S} [T_l \times \gamma(f, \mu(f)) + T_r \times \sum_{m \in M - \{\mu(f)\}} \gamma(f, m)]}$$

A realização da análise necessária à solução do problema proposto exige uma solução do modelo apresentado, atribuindo valores às constantes e às distribuições utilizadas. Nesta análise buscamos considerar uma carga de trabalho próxima da carga de trabalho típica de ambientes corporativos. Para as constantes do modelo utilizamos valores apresentados em artigos sobre cargas de trabalho e sistemas de arquivos.

O valor para o tempo de acesso local é igual à razão entre a média do tamanho em bytes das operações e a vazão do acesso ao disco local. O valor para o tempo de acesso remoto é igual à soma do tempo de acesso local com razão da média do tamanho em bytes das operações e a vazão da rede. Esta razão é o tempo consumido na transmissão dos dados na rede. A vazão do acesso ao disco local é a quantidade de bytes que podem ser transferidos para o disco ou lidos do disco local por unidade de tempo. Para esta vazão adotamos o valor 20 MB/s, utilizado por da Cunha Silva[1]. A vazão da rede é a quantidade de bytes que podem ser transmitidos ou lidos da rede por unidade de tempo. Para esta vazão adotamos o valor 12,5 MB/s, utilizado por da Cunha Silva. O valor apresentado para a vazão da rede supõe uma rede local.

Admitimos nesta análise que a quantidade de bytes manipulada no acesso é uma constante. Para obter a média do tamanho em bytes das operações baseamos-nos no apresentado por Vogels[5]. O trabalho de Vogels apresenta resultados quanto ao tamanho de cada operação. Segundo Vogels, 80% dos acessos manipulam até cerca de 5 KB. Também é apresentado que 60% dos acessos manipulam em torno de 1 KB. Deste modo, utilizamos o valor de 1 KB como aproximação para o tamanho médio das operações.

Levando em conta as considerações acima, o valor para o tempo de acesso local é 0,048 milissegundo e o valor para o tempo de acesso remoto é 0,1269 milissegundo.

3.2.1 Simplificações

Na análise consideramos que não há compartilhamento de arquivos no sistema de arquivos descrito e que não há arquivos iguais na mesma máquina. Deste modo, os termos $T_r \times \sum_{m \in M - \{\mu(f)\}} \gamma(f, m)$ e $T_l \times \gamma(f, \mu(\Delta(f)))$ são iguais a zero. O termo $\sum_{m \in M - \{\mu(\Delta(f))\}} \gamma(f, m)$ é igual a $\gamma(f, \mu(f))$. Portanto, o impacto pode ser expresso como:

$$I = \frac{T_l \times \sum_{f \in S_d} \gamma(f, \mu(f)) + T_r \times \sum_{f \in S_r} \gamma(f, \mu(f))}{T_l \times \sum_{f \in S} \gamma(f, \mu(f))}$$

Nesta última equação $\gamma(f, \mu(f)) = P(f)$.

Quanto à popularidade consideramos nesta análise dois padrões: no primeiro todos os arquivos têm a mesma popularidade; e no segundo os arquivos são subdivididos em arquivos muito populares e arquivos pouco populares. A intenção em utilizar este último padrão é analisar o impacto de ignorar a distribuição de popularidade no sistema de arquivos ao deduplicar, supondo uma deduplicação em que os arquivos mais populares são removidos.

Levando em conta uma popularidade p , igual para todos os arquivos, o impacto é:

$$I = \frac{T_l \times \sum_{f \in S_d} p + T_r \times \sum_{f \in S_r} p}{T_l \times \sum_{f \in S} p}$$

Para obter uma equação mais simples, devemos conhecer $|S_d|$ e $|S_r|$. Supondo $|S_r| = n$, a equação para o impacto é:

$$I = \frac{T_l \times (|S| - n) \times p + T_r \times n \times p}{T_l \times |S| \times p}$$

Esta equação, simplificada, é:

$$I = 1 + \frac{n}{|S|} \times \left(\frac{T_r}{T_l} - 1 \right)$$

Para encontrar uma expressão para n , supomos que existem g conjuntos de arquivos iguais R_i e cada um destes arquivos tem o mesmo número de arquivos t , $t \geq 1$. Formalmente:

$$|R_i| = t, 1 \leq i \leq g$$

Concluimos então que existem $t \times g$ arquivos com pelo menos uma duplicata no sistema. Seja d a proporção de arquivos no sistema que têm duplicatas:

$$|S| \times d = t \times g$$

Como de cada conjunto R_i , $|R_i| - 1$ arquivos pertencem a S_r , n é:

$$n = g \times (t - 1)$$

Retirando a dependência de g da equação:

$$n = \frac{|S| \times d \times (t - 1)}{t}$$

Usando esta equação na equação para o impacto:

$$I = 1 + d \times \left(\frac{t-1}{t} \right) \left(\frac{T_r}{T_l} - 1 \right) \quad (1)$$

Levando em conta duas popularidades no sistema, p e $j \times p$, $1 \leq j$, podemos estudar dois tipos de deduplicação: uma que remove os arquivos mais populares e outra que remove os arquivos menos populares. O impacto para o primeiro tipo de deduplicação é:

$$I = \frac{T_l \times (|S| - n) \times p + T_r \times n \times j \times p}{T_l \times ((|S| - n) \times p + n \times j \times p)}$$

Expressando o impacto independente de p :

$$I = \frac{T_l \times (|S| - n) + T_r \times n \times j}{T_l \times ((|S| - n) + n \times j)}$$

O impacto independente de n é:

$$I = \frac{T_l(1 - d \times \frac{t-1}{t}) + T_r \times d \times j \times \frac{t-1}{t}}{T_l \times (1 + d \times \frac{t-1}{t} \times (j - 1))}$$

Supondo que $t = 2$:

$$I = \frac{T_l(1 - \frac{d}{2}) + T_r \frac{d \times j}{2}}{T_l \times (1 + d \times \frac{j-1}{2})} \quad (2)$$

O impacto para a popularidade que remove os arquivos menos populares, considerando que os arquivos que

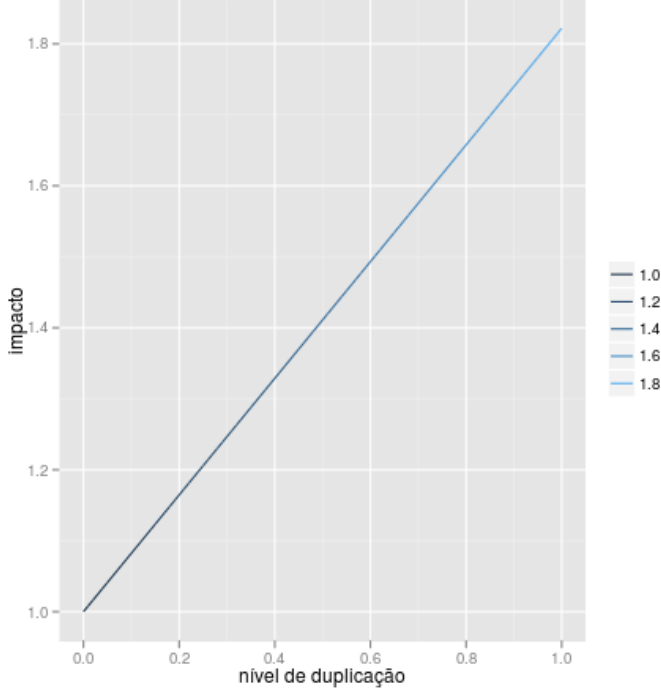


Figura 1: Impacto da deduplicação com popularidade constante variando o nível de duplicação. O tamanho dos grupos de arquivos similares foi fixado em 2.

não têm duplicatas têm popularidade igual a p e que $t = 2$, é:

$$I = \frac{T_l \times p \times (|S| - 2 \times n) + T_l \times j \times p \times n + T_r \times p \times n}{T_l \times ((|S| - 2 \times n)) \times p + n \times j \times p + n \times p}$$

Eliminando da equação a dependência de n , p e $|S|$, tem-se:

$$I = \frac{T_l \times (1 - d) + \frac{d \times (T_l \times j + T_r)}{2}}{T_l \times (1 + \frac{d \times (j-1)}{2})} \quad (3)$$

Foram realizados ao todo 4 experimentos. Em todos os experimentos a variável dependente foi o impacto sobre o tempo de acesso no sistema de arquivos.

Nos experimentos utilizamos a variável nível de duplicação. Ela é a proporção de arquivos no sistema de arquivos que têm pelo menos uma duplicata.

3.2.2 Impacto da deduplicação com popularidade constante variando o nível de duplicação

Neste experimento consideramos que todos os arquivos têm a mesma popularidade e que $t = 2$. Utilizamos a Equação 1. Substituindo na expressão o valor de t obtemos:

$$I = 1 + \left(\frac{d}{2}\right)\left(\frac{T_r}{T_l} - 1\right)$$

Como todas as variáveis do modelo, a exceção do nível de duplicação, foram tratadas como constantes, o resultado do impacto depende somente do nível de duplicação. Neste caso, o impacto apresenta crescimento linear, como observa-se na Figura 1.

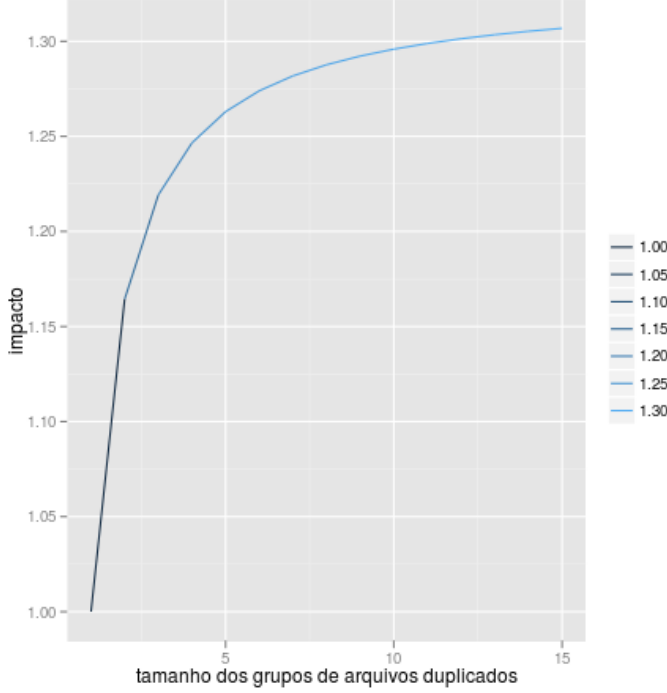


Figura 2: Impacto da deduplicação com popularidade constante variando o tamanho dos grupos de arquivos duplicados. O nível de duplicação no sistema foi fixado em 0,2.

3.2.3 Impacto da deduplicação com popularidade constante variando o tamanho dos grupos de arquivos duplicados

Neste experimento consideramos que todos os arquivos têm a mesma popularidade e que o nível de duplicação no sistema d é igual a 0,2. Utilizamos a Equação 1. Substituindo na expressão o valor de d obtemos:

$$I = 1 + 0,2 \times \left(\frac{t-1}{t}\right) \left(\frac{T_r}{T_l} - 1\right)$$

Levando em conta que todos os grupos de arquivos duplicados têm o mesmo tamanho, o aumento do tamanho dos grupos leva à redução do número de grupos. Como cada grupo, após a deduplicação ser executada, contribui com um arquivo para S_d , a redução do número de grupos acarreta a redução da quantidade de arquivos em S_d e o aumento do número de arquivos removidos no sistema, que representa um aumento na quantidade de acessos remotos, o que pode ser observado na Figura 2. Portanto, não apenas a quantidade de redundância mas também como ela está organizada no sistema influencia o impacto da deduplicação.

3.2.4 Impacto da deduplicação com popularidades variadas variando o nível de duplicação

Neste experimento consideramos dois tipos de popularidade no sistema. Para isso, utilizamos as Equações 2 e 3. Fixamos o valor da razão entre a maior popularidade e a menor popularidade, j , em 2. Para o impacto da deduplicação que elimina arquivos mais populares obtemos:

$$I = \frac{T_l \times (1 - \frac{d}{2}) + T_r \times d}{T_l \times (1 + \frac{d}{2})}$$

Para o impacto da deduplicação que elimina arquivos menos populares obtemos:

$$I = \frac{T_l \times (1 - d) + \frac{d \times (2 \times T_l + T_r)}{2}}{T_l \times (1 + \frac{d}{2})}$$

Observa-se na Figura 3 a diferença entre remover arquivos muito populares e arquivos pouco populares. A deduplicação que remove arquivos pouco populares causa um impacto de cerca de 17% no tempo de acesso aos arquivos no sistema caso o nível de duplicação no sistema seja de 20%, enquanto que a deduplicação que remove arquivos muito populares causa um impacto de cerca de 30%. O valor para o nível de duplicação citado foi

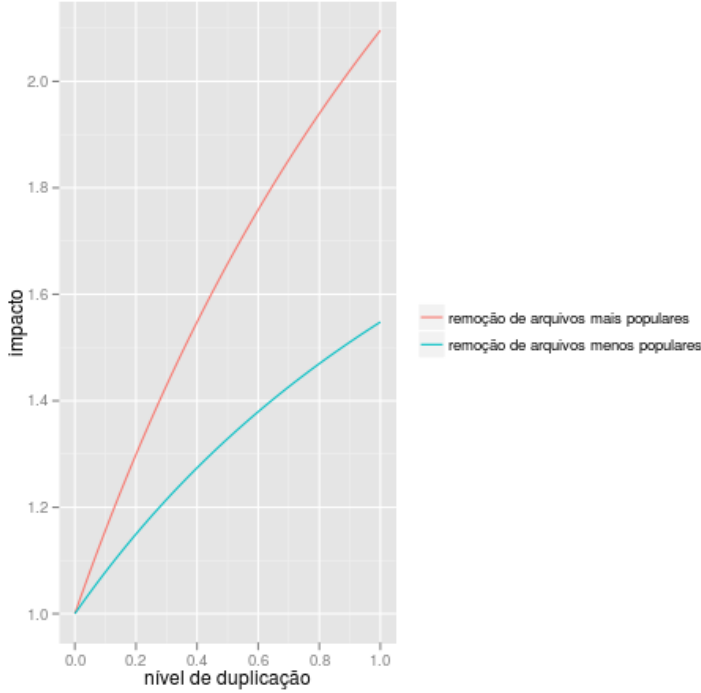


Figura 3: Impacto da deduplicação com popularidades variadas variando o nível de duplicação. Resultados com dois valores de popularidade em todo o sistema, com a razão entre a maior e a menor popularidade fixada em 2.

o valor obtido a partir de um experimento realizado em um sistema de arquivos do Laboratório de Sistemas Distribuídos. A carga de trabalho no ambiente do laboratório é semelhante às cargas de trabalho em ambientes corporativos, que são as cargas de trabalho aqui consideradas.

3.2.5 Impacto da deduplicação variando a razão entre as popularidades

Neste experimento consideramos dois tipos de popularidade no sistema. Para isso, utilizamos as Equações 2 e 3. Fixamos o valor do nível de duplicação d em 0,2. Para o impacto da deduplicação que elimina arquivos mais populares obtemos:

$$I = \frac{0,9 \times T_l + 0,1 \times j \times T_r}{T_l \times (0,1 \times j + 0,9)}$$

Para o impacto da deduplicação que elimina arquivos menos populares obtemos:

$$I = \frac{T_r + T_l \times j + 8 \times T_l}{T_l \times j + 9 \times T_l}$$

Variando a razão entre a maior popularidade e a menor popularidade, o impacto causado pela deduplicação que elimina arquivos muito populares aumenta à medida que a razão aumenta, como pode ser visto na Figura 4. Neste caso, aumenta a quantidade de acessos remotos no sistema. O impacto causado pela deduplicação que elimina arquivos pouco populares diminui com o aumento da razão. Esta última faz com que a proporção de acessos locais se torne maior.

4 Conclusões

Foi apresentada uma análise da influência da deduplicação no tempo de acesso aos arquivos em um sistema de arquivos distribuído sob uma carga de trabalho típica de ambientes corporativos. Supondo uma relação entre a deduplicação e o tempo médio de acesso, um modelo foi descrito com o intuito de descrever matematicamente esta relação. Experimentos foram realizados de forma a verificar a viabilidade da implementação da deduplicação de dados no ambiente considerado.

A análise puramente analítica revelou que a popularidade dos arquivos do sistema é uma variável que deve ser levada em consideração pelo método deduplicador, fato confirmado pelas análises experimentais. As análises

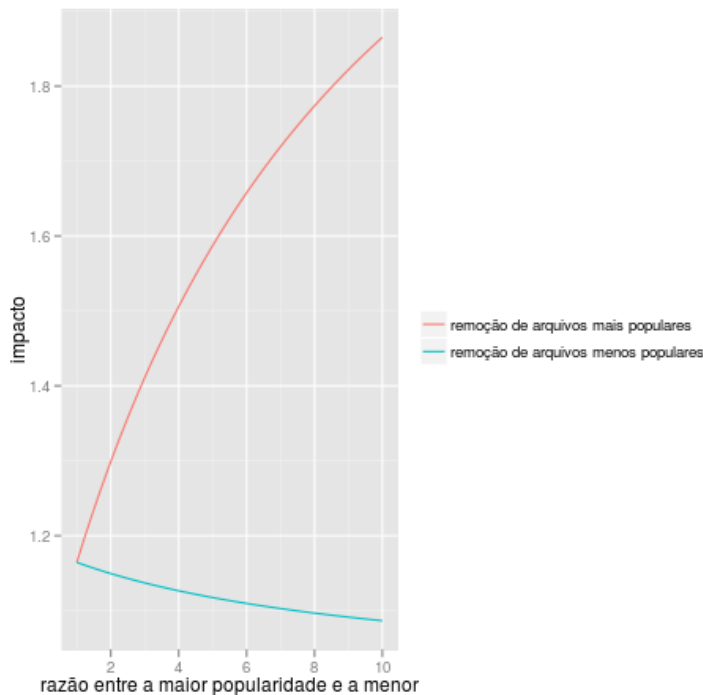


Figura 4: Impacto da deduplicação variando a razão entre as popularidades.

experimentais também mostraram que outra variável importante é o formato do agrupamento de arquivos similares no sistema de arquivos. Detectamos que a taxa de crescimento do tempo de acesso em relação à quantidade de conteúdo redundante é menor se a deduplicação for feita de maneira inteligente, removendo arquivos menos populares.

A simplificação mais importante do modelo é a restrição quanto a quantidade de conteúdo removido pela deduplicação. No modelo a deduplicação sempre remove o máximo possível de conteúdo duplicado. Esta política de remoção poderia ser substituída por uma política mais inteligente, na qual alguns arquivos poderiam ser preservados com o intuito de diminuir o impacto da deduplicação. O desenvolvimento desta política inteligente e avaliação do seu desempenho podem ser temas de trabalhos futuros.

Agradecimentos

Ao CNPq pela bolsa PIBIC concedida, aos colegas e amigos do Laboratório de Sistemas Distribuídos (LSD), em especial Manel, pela ajuda e inesgotável paciência, ao professor Francisco Brasileiro (Fubica), pela sua orientação e dedicação em tornar-me um cientista.

Referências

- [1] DA CUNHA SILVA, T. E. P. Políticas de Alocação e Migração de Arquivos em Sistemas de Arquivos Distribuídos para Redes Locais. Master's thesis, Universidade Federal de Campina Grande, Campina Grande, Paraíba, Brasil, 2010.
- [2] MEYER, D. T., AND BOLOSKY, W. J. A study of practical deduplication. In *9th USENIX Conference on File and Storage Technologies, San Jose, CA, USA, February 15-17, 2011* (2011), G. R. Ganger and J. Wilkes, Eds., USENIX, pp. 1–13.
- [3] PAWLOWSKI, B., NOVECK, D., ROBINSON, D., AND THURLOW, R. The nfs version 4 protocol. In *In Proceedings of the 2nd International System Administration and Networking Conference (SANE 2000)* (2000).
- [4] SOUZA, C., LACERDA, A. C., SILVA, J., PEREIRA, T., SOARES, A., AND BRASILEIRO, F. Beefs: Um sistema de arquivos distribuído posix barato e eficiente para redes locais. In *SBRC 2010 - Salão de Ferramentas* (may 2010).

- [5] VOGELS, W. File system usage in windows nt 4.0. *SIGOPS Oper. Syst. Rev.* 33, 5 (Dec. 1999), 93–109.
- [6] ZHU, B., LI, K., AND PATTERSON, R. H. Avoiding the disk bottleneck in the data domain deduplication file system. In *6th USENIX Conference on File and Storage Technologies, FAST 2008, February 26-29, 2008, San Jose, CA, USA* (2008), M. Baker and E. Riedel, Eds., USENIX, pp. 269–282.