Entendendo os Efeitos da Localidade de Referência em Hierarquias de Caches na Web

Fabrício Benevenuto, Fernando Duarte, Virgílio Almeida

Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Av. Antônio Carlos, 6627 - 31270-901, Belo Horizonte, MG

{fabricio, fernando, virgilio}@dcc.ufmg.br

Abstract. *This work presents an extensive evaluation of the filtering effect that occur in proxy servers organized as an hierarchy of caches. Using a recently proposed model called ADF (Aggregation, Disaggregation and Filtering) and entropy as a metric for Web traffic characterization, we present an evaluation of the effects that the locality of reference of request streams suffers when passing through a hierarchy of caches. Moreover, we consider the use of average entropy for the comparison of reference locality among streams and we present the necessary information so that entropy can be dynamically calculated by a proxy server.*

Resumo. *Este trabalho apresenta uma extensiva avaliação dos efeitos de filtragem que ocorrem em servidores proxy organizados com uma hierarquia de caches. Utilizando um modelo recentemente proposto, chamado ADF (Agregação, Desagregação e Filtragem) e entropia como métrica para a caracterização do tráfego Web, nós apresentamos uma avaliação dos efeitos que a localidade de referência de seqüências de requisições sofre ao passar por uma hierarquia de caches. Além disso, nós propomos o uso da entropia média para a comparação da localidade de referência entre seqüências de requisições e fornecemos o arcabouço necessário para que a entropia possa ser calculada dinamicamente por um servidor proxy.*

1. Introdução

O rápido crescimento do tráfego e o crescente aumento do número de usuários são características marcantes do fenômeno da Web. Neste contexto, o uso de proxies surge como uma solução bastante eficaz para aumentar o desempenho da Web, melhorando a escalabilidade de servidores, reduzindo o tráfego na rede e o tempo de resposta no atendimento às requisições dos usuários.

Um servidor proxy pode ser entendido como um intermediador de todo o tráfego entre clientes e servidores HTTP. Servidores Proxy são usualmente utilizados para delimitar uma porção da rede, na qual os clientes estão topologicamente (e muitas vezes geograficamente) próximos uns dos outros [4]. Desta forma, cada requisição dos clientes e cada resposta dos servidores passa pelos servidores proxy. Quando estes servidores

repassam um documento para os clientes, uma cópia é armazenada em sua cache local, para que futuras requisições para este documento possam ser atendidas diretamente pelo servidor proxy, sem a necessidade de contatar o servidor HTTP novamente.

Os servidores proxy operam agregando, desagregando e filtrando as seqüências de requisições que passam por eles. Podemos dizer que o servidor proxy *agrega* as requisições que chegam a ele em uma única seqüência, que é processada utilizando sua cache local. Além disso, o servidor proxy atua como um *desagregador* de tráfego, distribuindo as requisições dos seus clientes para diferentes servidores da Internet. Quando uma seqüência de requisições passa por um proxy, apenas as requisições que não puderam ser atendidas por sua cache são desagregadas em direção ao servidor Web destino. Enquanto um servidor proxy agrega e desagrega tráfego, ele realiza sua mais importante função: a *filtragem*. As requisições feitas para documentos que estão na cache do servidor proxy são respondidas de maneira transparente ao cliente.

O uso de caching na Web está muitas vezes associado a uma organização hierárquica. A cache do navegador, localizada na máquina do cliente, é o nível mais baixo da hierarquia. Um nível acima estão as caches das intranets, que consistem de proxies de universidades e organizações. Quando subimos na hierarquia, temos os proxies regionais e assim por diante. Uma requisição que não pode ser satisfeita por um proxy cache pode ser enviada para o proxy imediatamente acima na hierarquia até que ela possa ser atendida, tendo como última opção o servidor destino.

A organização de uma hierarquia de caches eficiente envolve o estudo das propriedades das seqüências de requisições. Diversas questões relacionadas a *caching* tais como a localização dos servidores proxies, o tamanho de cada cache e a configuração dos níveis dentro da hierarquia requerem um estudo mais aprofundado de como as propriedades das seqüências de requisições mudam à medida que elas passam pelos servidores proxy. Neste contexto, uma visão do tráfego da Web segundo os efeitos de agregação, filtragem e desagregação associada a métricas adequadas para esta visão, trazem um melhor entendimento dos efeitos que a localidade de referência sofre à medida que uma seqüência de requisições percorre uma hierarquia de caches.

O estudo de localidade de referência sob esta nova perspectiva da Web foi inicialmente proposto em [11]. Além disso, este trabalho propôs o uso de entropia como métrica para medir a localidade de referência de seqüências de requisições. Neste trabalho, nós aplicamos esta métrica em um contexto de hierarquia de caches, avaliando o impacto que políticas de cache exercem sobre a localidade de referência das requisições. Além disso, nós propomos o uso de novas formas de medir entropia, fornecendo todo o arcabouço necessário para que esta métrica possa ser calculada dinamicamente por um componente qualquer da Web.

Como principais contribuições deste trabalho, podemos destacar:

- Uma extensiva avaliação de desempenho de diversas políticas de caches em diferentes níveis de uma hierarquia de caches. Foram avaliadas tanto métricas tradicionais, como taxa de acerto, quanto métricas recentemente propostas, como entropia.
- Uma avaliação da localidade de referência com base na entropia que pode ser utilizada como guia para o projeto de sistemas de caching em hierarquia, colaborando

para que estes sistemas não sejam construídos com base na intuição.
- Propomos o uso da entropia média para comparação da localidade de referência de diferentes seqüências de requisições. Além disso, fornecemos o arcabouço necessário para que esta a entropia possa ser calculada dinamicamente.

O restante do artigo está organizado da seguinte forma. A próxima seção apresenta trabalhos anteriores sobre o estudo de localidade de referência e hierarquia de caches. A seção 3 apresenta os conceitos básicos sobre entropia e as novas formas propostas para medição desta métrica. A seção 4 apresenta a metodologia experimental adotada, além das principais características da carga utilizada nos experimentos. Nossos principais resultados são detalhados na seção 5, enquanto que a seção 6 conclui o artigo e apresenta possíveis direções.

2. Trabalhos Relacionados

Basicamente, podemos diferenciar dois tipos de localidade no tráfego Web: a localidade espacial e a localidade temporal. A localidade espacial [3] consiste no estudo da correlação estrutural existente entre a referências de uma seqüência de requisições, enquanto que a localidade temporal [3, 13] estabelece que objetos recentemente acessados possuem uma maior probabilidade de serem acessados em um futuro próximo.

O estudo de localidade temporal foi, em grande parte, motivado pelo impacto desta propriedade no desempenho de sistemas de caches. Existem vários exemplos de aplicação destes estudos que, de forma geral, serviram de base para o desenvolvimento de políticas de reposição de cache [17, 7], protocolos de coordenação entre caches [9] e algoritmos de prefetching [5].

As primeiras idéias sobre como caracterizar os efeitos de proxies sobre seqüências de requisições foram introduzidas em [18]. Este trabalho introduz uma visão de caches como filtros, e compara propriedades de chegada e saída de seqüências de referências, no contexto de referências de programas em memória. No contexto de caches na Web, Mahanti, Williamson e Eager [14] estudaram como a localidade temporal muda em diferentes níveis de uma hierarquia de caches. Eles mostraram que a concentração de referências tende a diminuir e a calda da distribuição de Zipf tende a aumentar quando se sobe na hierarquia. Este efeito também foi percebido e caracterizado em [19, 8].

Williamson [19] avalia, através de simulações, diferentes políticas de reposição de cache em diferentes níveis de uma hierarquia de caches na Web. O impacto do uso de cada política em cada nível na localidade de referência das requisições foi observado através de métricas indiretas como taxa de acerto. Enquanto [19] somente considerou a filtragem, [11, 10] introduziu o estudo de duas outras transformações nas quais seqüências de requisições estão submetidas: agregação e desagregação. Eles ainda organizaram estas transformações em um modelo, propuseram e validaram métricas para a análise de localidade temporal quando as seqüências de requisições se movem através deste modelo.

Em nosso trabalho, nós apresentamos uma avaliação semelhante aos experimentos realizados em [19], porém nós avaliamos o impacto na localidade de referência através das ferramentas propostas em [11]. Além disso, nós propomos novas formas de se utilizar a métrica proposta em [11] para analisar quantitativamente a localidade de referência,

fornecendo um arcabouço para que esta métrica possa ser calculada dinamicamente e aplicada a ambientes reais.

3. Métricas para Localidade de Referência

Esta seção apresenta as métricas utilizadas neste artigo para medir quantitativamente a localidade de referência de seqüências de requisições. Na seção 3.1 nós apresentamos o conceito de entropia e as principais razões para a utilização desta métrica. Na seção 3.2 mostramos uma forma eficiente de se calcular a entropia dinamicamente e as vantagens desse método para o estudo da localidade de referência. Na seção 3.3 propomos a entropia média como uma métrica adequada para comparar a popularidade das requisições que chegam a um servidor na Web.

3.1. Entropia

Muitos dos estudos sobre tráfego na Web têm se focalizado nas propriedades de localidade temporal apresentadas pelas seqüências de requisições [3]. A idéia por trás da localidade temporal é que um objeto recentemente referenciado possui uma *alta* probabilidade de ser referenciado em um futuro próximo[15]. Este conceito de localidade temporal pode ser dividido em dois efeitos [13, 14]: *popularidade* e *correlação*. O foco deste trabalho está apenas no efeito de popularidade.

A distribuição de popularidade de um conjunto de requisições é usualmente caracterizada através da *Lei de Zipf* [12, 3, 6]. A Lei de Zipf determina que a popularidade do n-ésimo objeto mais popular é proporcional a $1/n$. Em geral, distribuições similares a distribuições que obedecem a Lei de Zipf têm sido utilizadas para aproximar muitas seqüências de requisições na Web. Neste tipo de distribuição:

$$P[O_n] \propto n^{-\alpha}$$

no qual $P[O_n]$ é a probabilidade de referenciar o n-ésimo objeto mais popular (tipicamente $\alpha \leq 1$). O o coeficiente α é muitas vezes utilizado como um indicador da concentração de popularidade das requisições.

Recentemente, uma medida mais direta do que o coeficiente α para avaliar a concentração de popularidade foi proposta para a Web, a entropia [11]. A entropia $H(X)$ de uma variável aleatória X, tomando n possíveis valores com probabilidade p_i, pode ser calculada da seguinte maneira:

$$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{1}$$

Note que $H(X)$ depende apenas da probabilidade de ocorrência das requisições e do número n de diferentes requisições do conjunto. O valor máximo ($H(X) = log_2 n$) é atingido quando as requisições são equiprováveis ($\forall i(p_i = 1/n)$), e o valor mínimo ($H(X) = 0$) ocorre quando um único objeto concentra todas as referências($\exists i(p_i = 1)$). Desse modo, para conjuntos com o mesmo número de requisições, quanto maior o valor da entropia, menor a concentração de popularidade em poucos objetos e, conseqüentemente, quanto menor a entropia, maior a concentração em poucos objetos.

3.2. Entropia Dinâmica

Nesta seção, propomos uma forma de encontrar o valor da entropia dinamicamente, mostrando que esta métrica pode ser ampliada para ser utilizada em ambientes reais. Isto permite uma análise mais detalhada da localidade de referência, que pode ser utilizada para tomada decisões. Um servidor proxy pode, por exemplo, variar algum parâmetro de seu funcionamento baseado nas variações que ocorrem na localidade de referência da seqüência de requisições que está recebendo.

Para que a definição de entropia, proposta e validada em [11], possa ser utilizada em ambientes reais, seu valor deve ser medido de maneira incremental, sendo recalculado a cada nova requisição adicionada na seqüência. Entretanto, em trabalhos anteriores [11, 2], a entropia foi calculada somente a partir do momento em que se teve conhecimento de todo o conjunto de requisições do log analisado.

Expandindo a equação 1, encontramos uma forma prática e dinâmica para calcular a entropia. Considerando como n_t o total de requisições do conjunto e que n_i como o número de requisições para o objeto i, então $p_i = n_i/n_t$ e podemos encontrar a entropia da seguinte forma:

$$H(X) = \log_2 n_t - \frac{1}{n_t} \sum_{i=1}^{n} n_i log_2 n_i \tag{2}$$

Utilizando a equação 2, o valor da entropia pode ser encontrado atualizando o valor de n_t e o valor do somatório $S = \sum_{i=1}^{n} n_i \log_2 n_i$ a cada requisição.

3.3. Entropia Média

Nesta seção propomos o uso da entropia média como forma de comparar seqüências com diferentes números de requisições. Como a entropia depende do número de requisições distintas do conjunto é necessário fazer algum tipo de ajuste nesta métrica (ex. normalização) para comparar seqüências com diferentes números de requisições.

A entropia normalizada é uma forma proposta para se comparar a entropia de conjuntos com número diferente de requisições [11]. Essa normalização se baseia no maior valor possível para a entropia do conjunto de requisições. Sendo n o número de requisições distintas, a entropia normalizada $H^n(X)$ é definida como:

$$H^n(X) = \frac{H(X)}{log_2 n} \tag{3}$$

Entretanto, quando tratamos de conjuntos de requisições de tamanhos iguais, as entropias destes conjuntos podem ser comparadas diretamente, sendo desnecessário uma normalização conforme a apresentada na equação 3. Com base nisso, propomos a entropia média. A idéia da entropia média é comparar diferentes conjuntos baseando em um mesmo número de requisições. Utilizamos o conceito de janela para encontrar o valor da entropia. Fixamos um tamanho para a janela, representando o número de requisições utilizadas para o cálculo, e percorremos o conjunto com a janela, requisição a requisição, calculando o valor da entropia para cada janela, e encontrando o valor médio ao final.

Considerando uma janela de tamanho m, uma seqüência com um total de n_t requisições e $H(X_{[i,j]})$ como sendo o valor da entropia da janela que contém o intervalo da i-ésima até a j-ésima requisição, definimos a entropia média $H^m(X)$ como:

$$H^m(X) = \frac{\sum_{i=0}^{n_t-m} H(X_{[i+1,m+i]})}{n_t - m + 1} \quad (4)$$

Se quisermos utilizar a noção de localidade de referência em um ambiente real como, por exemplo, em um servidor Web, precisamos avaliar a localidade de uma certa amostra das requisições que chegam ao servidor. Neste contexto a entropia média, associada ao cálculo da entropia de forma dinâmica, surge como uma forma adequada para capturar a variação de popularidade do fluxo de requisições que chega a cache. Isso pode ser feito, por exemplo, comparando a entropia da última janela com o valor da entropia média.

A escolha do tamanho da janela para o cálculo da entropia pode influenciar na análise da concentração de popularidade. Por exemplo, se a janela for pequena, a entropia obtida passa a noção da localidade de uma amostra pequena, que pode não representar corretamente a popularidade de todo o fluxo. Por outro lado, se o tamanho da janela for relativamente grande, a noção de variação de popularidade no cálculo de cada janela de requisições será menos perceptiva. Sugerimos que, para comparar diferentes seqüências de requisições é necessário que o tamanho da janela esteja na mesma ordem de grandeza do total de requisições destas seqüências.

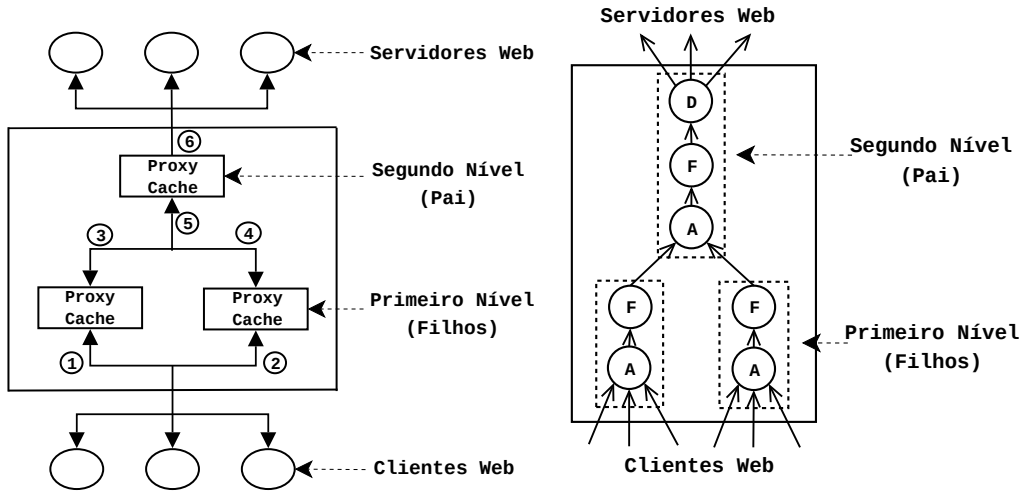
4. Metodologia Experimental

Esta seção discute a metodologia utilizada para as simulações desenvolvidas. Foi construído um simulador de um sistema de hierarquia de caches organizado como mostra a figura 1(a).

Com o intuito de entender melhor os efeitos da localidade de referência em um sistema de hierarquia de caches, utilizamos o modelo ADF [11], uma abstração da topologia da Web que representa as principais transformações que as seqüências de requisições sofrem ao passar pelos diferentes pontos da Web. O modelo ADF representa a Web através de um grafo onde os vértices representam pontos onde as seqüências de requisições podem ser alteradas, e as arestas são os caminhos de conexão entre estes pontos. Os vértices no grafo são de três diferentes tipos, dependendo de qual efeito eles causam no tráfego da Web: Agregação (A), Desagregação (D) e Filtragem (F). Desta forma, diferentes componentes da topologia da Web podem ser representados por combinações destes três tipos de vértices.

Para explicar a idéia por trás desta abordagem, vamos considerar a configuração da figura 1(a). Esta figura apresenta um sistema de caches de dois níveis com duas caches (filhos) no primeiro nível e uma cache (pai) no segundo. As requisições dos usuários são recebidas diretamente pelas caches no primeiro nível e, as requisições que não podem ser satisfeitas neste nível são agregadas formando uma seqüência de requisições que é encaminhada para a cache do segundo nível. Não há nenhuma interação entre as cache do primeiro nível.

A figura 1(b) mostra a representação no modelo ADF do sistema hierárquico de caches que estamos utilizando em nossos experimentos. Observe que as caches do primeiro nível funcionam como pontos de agregação das requisições que chegam dos vários clientes. Estas caches aplicam outra transformação nestas requisições, a filtragem, pois, algumas destas requisições são atendidas diretamente pela cache e não chegam no segundo nível. As seqüências de requisições vindas das caches do primeiro nível são agregadas e novamente, são filtradas no segundo nível, de onde são desagregadas para os servidores.



(a) Pontos de medição de entropia (b) Modelo ADF

Figura 1: Sistema de hierarquia de caches

Nas simulações, nós avaliamos o comportamento da entropia média quando uma seqüência de requisições passa por uma hierarquia de caches, variando o tamanho destas caches. O tamanho das caches foi incrementado seguindo uma potência de 2, indo de $1MB$ até $16GB$, sendo que o tamanho máximo foi escolhido com base no tamanho total da carga (ponto no qual a cache se torna infinita). Para escolher o tamanho da janela utilizado no cálculo da entropia média, foram realizados diversos experimentos variando o tamanho deste intervalo. A diferença dos resultados obtidos para tamanhos de intervalo com ordem de grandeza 100.000 varia muito pouco. Desta forma, este valor foi escolhido para os nossos experimentos.

4.1. Políticas de Reposição de Cache

Políticas de reposição de cache determinam quais documentos serão removidos quando for necessário liberar espaço na cache para armazenar um novo documento que chega. Quatro políticas de reposição de cache foram consideradas: LRU, LFU-Aging, GD-Size e LRU-Threshold.

- **LRU**: A política LRU (*Least Recently Used*) tende a manter em cache os documentos mais recentes, retirando os documentos menos recentemente acessados.
- **LFU-Aging**: Consiste em manter em cache os documentos mais freqüentemente acessados, além de evitar que documentos antigos permaneçam em armazenados [16].

- **GD-Size**: A política *Greed Dual Size* [7], comumente chamada de GD-Size, consiste em manter em cache documentos menores além de levar em consideração o aspecto de recenticidade dos documentos. Desta forma, a cache pode armazenar mais documentos e obter uma taxa de acerto maior. Entretanto a quantidade de dados servida por documento é relativamente pequena quando comparada com as outras políticas.
- **LRU-Threshold**: Esta política é semelhante à política LRU. Entretanto, documentos maiores que um determinado tamanho pré-estabelecido não são armazenados nas caches do primeiro nível da hierarquia. A idéia principal é fazer com que as caches do primeiro nível atendam os arquivos pequenos e que as caches do segundo nível possam agregar os documentos maiores, atendendo a um maior número de objetos no primeiro nível e tirando um maior proveito da localidade temporal existente entre os documentos que passam pelas caches do segundo nível.

Em nossos experimentos, nós avaliamos o impacto de diferentes políticas de reposição de cache combinadas em níveis diferentes da hierarquia de caches.

4.2. Características da Carga

Nesta seção, nós apresentamos as principais características dos logs utilizados em nossos experimentos. Estes logs foram obtidos do POP-MG [1] que atende, além de clientes incorporados, universidades e usuários que utilizam Internet via transmissão a rádio.

O POP-MG possui um sistema de caches em hierarquia semelhante ao apresentado na figura 1(a). Para os nossos experimentos, nós utilizamos logs de duas máquinas do primeiro nível do POP-MG, as quais chamamos de *Pop-1* e *Pop-2*. Estes logs seguem o formato do *Squid* e as principais características desta carga são apresentadas na Tabela 1. Para aquecer as caches foram utilizados os logs dos dias 16-17/10/01 enquanto que as medições de entropia e taxa de acerto foram realizadas com os logs dos dias 18-19/10/01.

Analisando as características dos logs, notamos que o número de objetos diferentes representa cerca de 26% do total das requisições. Desta porcentagem, cerca de 69% são documentos com apenas uma referência (*1-timers*). Em geral, estes logs são constituídos de objetos pequenos pois, como podemos observar, o 3° quartil não chegou a atingir os objetos de $3KB$. Entretanto, alguns objetos são relativamente grandes para documentos Web, o que explica o coeficiente de variabilidade das distribuições de tamanhos serem relativamente altos. As cargas de trabalho possuem um tamanho que gira em torno de 4 GB, tamanho máximo que as caches do primeiro nível necessitam para armazenar todos os objetos presentes nas seqüências de requisições (no pior caso, no qual todos os objetos são diferentes).

5. Resultados Experimentais

Nesta seção apresentamos os resultados da simulação da hierarquia de caches mostrada na figura 1(a). A entropia média foi medida nos pontos numerados nesta figura, que são os pontos onde percebemos os efeitos das operações de filtragem, agregação e desagregação. Foram avaliadas diferentes políticas de cache no segundo nível, quando

fixamos as políticas LRU, LFU-Aging e GD-Size respectivamente no primeiro nível. A taxa de acerto foi calculada para cada uma das caches, sendo que *Filho D*, *Filho E* e *Pai* foram os nomes dados às caches do primeiro e segundo níveis, respectivamente. Nas figuras 2, 3 e 4, os gráficos da esquerda mostram a taxa de acerto e os gráficos da direita mostram a entropia média quando variamos o tamanho das caches.

Quando comparamos a eficácia das caches do primeiro nível com as caches do segundo nível, podemos ver que o primeiro nível de caches, para todas as políticas avaliadas, obtém maior taxa de acerto do que o segundo nível. A razão para isto é a que filtragem absorve parte da localidade temporal, gerando uma seqüência de requisições que não foram atendidas com pouca concentração de popularidade. Isto significa que o primeiro nível de cache filtra a propriedade de localidade temporal da seqüência enviada ao segundo nível. Este efeito pode ser observado quando comparamos a entropia das seqüências de requisições que chegam na hierarquia de cache, pontos 1 e 2, com a entropia após o primeiro nível de caches, pontos 3 e 4.

Em alguns gráficos, a taxa de acerto no segundo nível cai com o aumento do tamanho da cache. Isto acontece porque as caches do primeiro nível, também aumentando de tamanho, filtram mais requisições e reduzem a popularidade das requisições que chegam no segundo nível (podemos ver um exemplo deste efeito nas curvas 2(a), 3(a)).

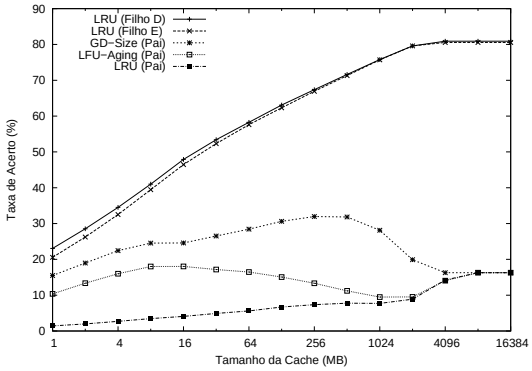
À medida que o tamanho das caches aumenta, o valor da entropia nos pontos 3, 4, 5 e 6 cresce até estabilizar. Para tamanhos pequenos de cache, à medida que a cache aumenta, mais objetos são mantidos em cache, o que diminui a localidade de referência da seqüência de requisições que sai das caches do primeiro nível, aumentando a entropia média, o que explica a elevação inicial das curvas.

Quando observamos a entropia nos vários pontos da hierarquia podemos notar como cada política atua na localidade de referência existente nas requisições. Ao passar pelo primeiro nível a entropia das seqüências de requisições aumenta. Isto pode ser per-

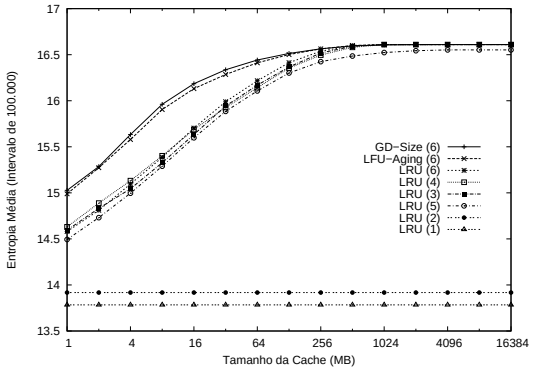
Ítem	**Pop-1**	**Pop-2**	**Pop-1**	**Pop-2**
Período	16-17/10/01	16-17/10/01	18-19/10/01	18-19/10/01
# requisições	882.639	908.317	902.998	919.541
Documentos diferentes	234.663	246.560	238.880	237.290
% Documentos diferentes	26%	27%	26%	25%
1-timers	161.646	173.796	164.011	164.878
% 1-timers	69%	70%	69%	69%
Tamanho total (MB)	3.865	4.220	3.974	4.213
Menor arquivo	0	0	0	0
Maior arquivo (MB)	33,13	41,75	29,61	49,70
Tamanho médio (KB)	4,48	4,76	4,51	4,69
1° **Quartil (Bytes)**	365	372	364	371
Mediana (Bytes)	757	746	1.392	778
3° **Quartil (Bytes)**	2.690	2.698	2.576	2.571
CV	16,52	29,29	14,22	19,62
Entropia Média	14,64	14,32	13,78	13,92

Tabela 1: Características dos logs

cebido quando comparamos as entropias nos pontos 1 e 2 com as entropias nos pontos 3 e 4 respectivamente. No ponto 5 (ponto no qual as seqüências foram agregadas) a entropia diminui, indicando um aumento na localidade de referência. Ao passar pela cache do segundo nível, a entropia média aumenta novamente, sendo que, à medida que o tamanho da cache aumenta e a cache passa a ter espaço suficiente para armazenar todos os objetos distintos, se comportando como se tivesse tamanho infinito, a entropia tende a um limite superior que representa a seqüência sem popularidade.

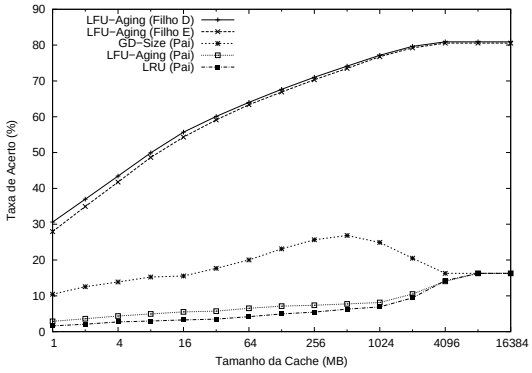


(a) Taxa de Acerto

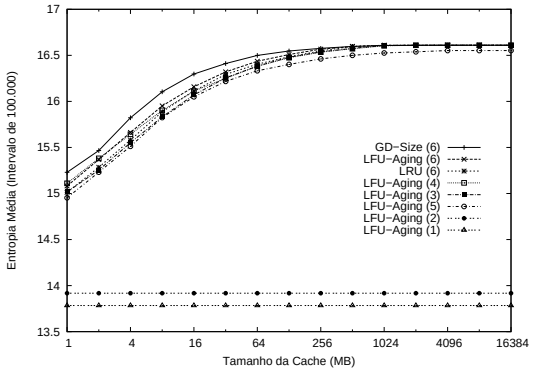


(b) Entropia Média

Figura 2: Política LRU no primeiro nível



(a) Taxa de Acerto



(b) Entropia Média

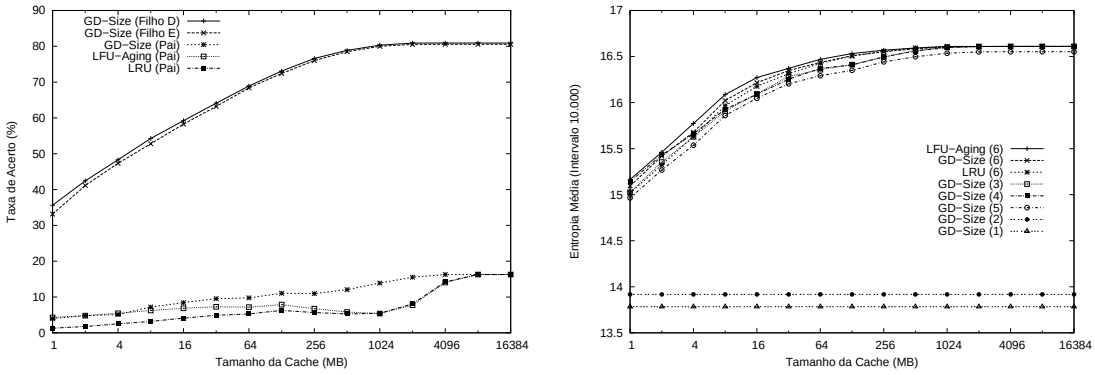
Figura 3: Política LFU-Aging no primeiro nível

Uma outra estratégia para a configuração de um sistema de hierarquia de caches avaliada foi o uso de LRU-Threshold. Nesta configuração, é permitido para as caches do primeiro nível manter em cache apenas documentos menores do que um determinado tamanho pré-estabelecido.

A figura 5 mostra a taxa de acerto e a entropia média quando utilizamos LRU-Threshold no primeiro nível de caches armazenando somente arquivos menores que $4KB$. A tabela 1 mostra que este valor é maior que o terceiro quartil de todos os logs, o que indica que a maior parte dos objetos dos logs podem ser mantidos nas caches do primeiro nível, fazendo com que o segundo nível fique responsável pelos objetos maiores.

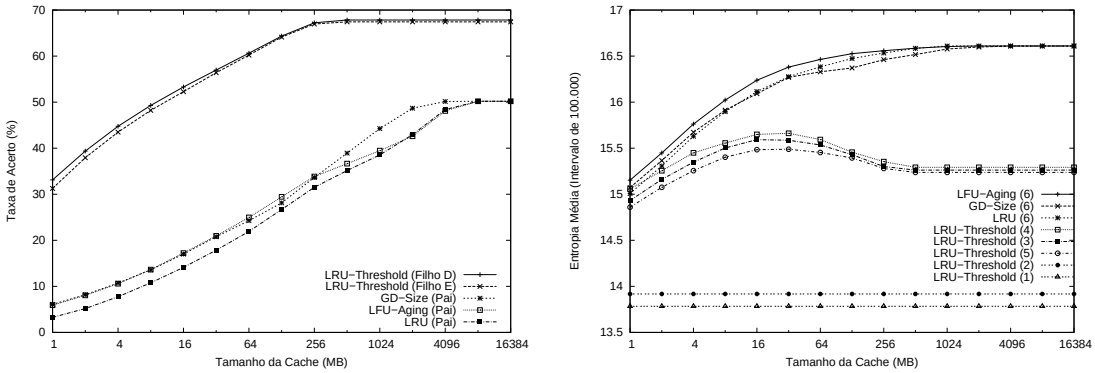
Uma análise da relação entre entropia e taxa de acerto pode ser vista na discussão dos resultados dos gráficos da figura 5. Com a utilização do LRU-Threshold, os objetos maiores que $4KB$ saem do primeiro nível com sua popularidade inalterada (eles não estão sendo filtrados). Desta forma, quando as caches do primeiro nível se tornam grande o suficiente para armazenar todos os objetos menores do que $4KB$, a seqüência de requisições que sai destas caches possuem apenas *1-timers* de objetos menores que $4KB$ e documentos maiores que este tamanho, fazendo com que estes objetos maiores se tornem relativamente populares e diminuindo a entropia. Esta diminuição da entropia do primeiro nível (por volta dos tamanhos de cache 64 MB e 256 MB) teve implicação direta na taxa de acerto da cache do segundo nível. Até este ponto a política LFU-Aging obteve uma melhor taxa de acerto e deste ponto em diante a política GD-Size conseguiu o melhor resultado. Este efeito sugere que variações na entropia (provocadas por mudanças repentinas nas seqüências de requisições) podem ser utilizadas para configurar dinamicamente as políticas de cache em hierarquia

Finalmente, quando avaliamos o sistema de cache como um todo, esperamos que a seqüência de requisições que sai do sistema, possua o menor número de documen-



(a) Taxa de Acerto (b) Entropia Média

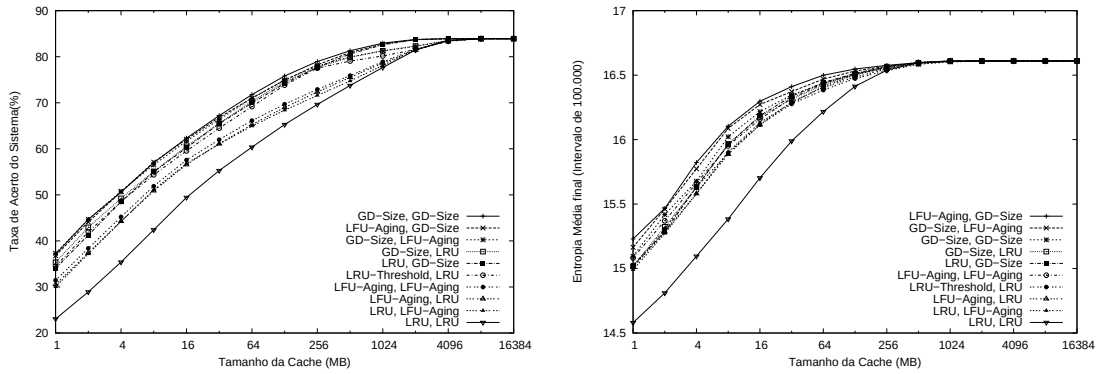
Figura 4: Política GD-Size no primeiro nível



(a) Taxa de Acerto (b) Entropia Média

Figura 5: Política LRU-Threshold(4KB) no primeiro nível

tos possível, de forma que um mesmo documento não apareça replicado muitas vezes nesta seqüência, indicando que aquela configuração do sistema conseguiu filtrar um maior número de requisições. Uma das formas de avaliar a eficácia do sistema de caches é medindo a entropia média da seqüência de requisições que sai da hierarquia de caches. A figura 6(b) mostra esta medida para várias configurações de políticas de cache e a figura 6(a) mostra a taxa de acerto do sistema de caches. Como era de se esperar, a combinação de LRU nos dois níveis resultou no valor mais baixo para a taxa de acerto e a entropia média do sistema. Além disso, podemos notar que a combinação das políticas LFU-Aging e GD-Size nos dois níveis produziram os melhores resultados. É importante observar que apesar de configuração com GD-Size nos dois níveis produzir a melhor taxa de acerto, esta política não obtém a melhor entropia final. Isso acontece porque esta política favorece objetos menores, o que contribui bastante para a taxa de acerto. Entretanto, objetos grandes e com alguma popularidade são descartados, fazendo com que a seqüência de requisições final ainda possua alguma localidade temporal. Este mesmo efeito foi observado na política LRU-Threshold, que também não prioriza apenas a localidade temporal.



(a) Taxa de Acerto do sistema

(b) Entropia Média Final

Figura 6: Comparação entre diferentes configurações da hierarquia de caches

6. Conclusões e Trabalhos Futuros

Sistemas de caches muitas vezes são configurados em hierarquia na tentativa de melhorar a qualidade do serviço percebida pelos usuários e diminuir o tráfego na rede. Neste trabalho, utilizamos um modelo chamado ADF (Agregação, Desagregação e Filtragem) e entropia como métrica para a caracterização do tráfego Web, para avaliar os efeitos que a localidade de referência de seqüências de requisições sofrem ao passar por uma hierarquia de caches. Nossos resultados mostram como as transformações de agregação, filtragem e desagregação atuam na localidade de referência e qual o impacto dessas operações no desempenho de uma hierarquia de caches. Além disso, mostramos que configurações heterogêneas de políticas de cache tiram um melhor proveito da localidade de referência e produzir melhores taxas de acerto.

Algumas futuras direções incluem explorar o cálculo da entropia dinâmica em servidores proxy e desenvolver um modelo para um sistema de caches em hierarquia no qual as políticas de cache para os diferentes níveis desta hierarquia possam ser alteradas

dinamicamente, com base nas variações da entropia das seqüências de requisições que chegam à cache.

Referências

[1] POP-MG. Ponto de Presença da Rede Nacional de Pesquisa em Minas Gerais. `http://www.pop-mg.rnp.br`.

[2] B. Abrahao and F. Benevenuto. Evaluating Cache-Layering to Improve Web Cache System Performance. In *Simpósio Brasileiro de Sistemas Multimídia e Web (WebMidia)*, Salvador, Brasil, Novembro 2003.

[3] V. Almeida, A. Bestavros, M. Crovella, and A. Oliveira. Characterizing Reference Locality in the WWW. In *Proc. the Fourth International Conference on Parallel and Distributed Information Systems (PDIS96)*, Dezembro 1996.

[4] F. Benevenuto, B. Vitorino, B. Coutinho, D. Guedes, and W. Meira Jr. A Scalable Approach for the Distribution of E-Commerce Services Based on Application Level Active Networks. In *Anais do 22 Simpósio Brasileiro de Redes de Computadores, SBRC2004*, Gramado, Brasil, Maio 2004.

[5] A. Bestavros. Using speculation to reduce server load and service time on the www. In *Proc. CIKM'95*, Baltimore, Maryland, novembro 1995.

[6] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web Caching and Zipf-like Distributions: Evidence and Implications. In *Proc. 18nd Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom)*, 1999.

[7] P. Cao and S. Irani. Cost-Aware WWW Proxy Caching Algorithms. In *Proc. the 1997 Usenix Symposium on Internet Technologies and Systems (USITS-97)*, Monterey, CA, 1997.

[8] R. Doyle, J. Chase, S. Gadde, and A. Vahdat. The Trickle-Down Effect: Web Caching and Server Request Distribution. In *Proc. the 6th Web Caching Workshop*, pages 1–18, Junho 2001.

[9] L. Fan, P. Cao, J. Almeida, and A. Broder. Summary Cache: a Scalable Wide-area Web Cache Sharing Protocol. *IEEE / ACM Transactions on Networking*, 8(3):281–293, 2000.

[10] R. Fonseca, V. Almeida, and M. Crovella. Locality in a Web of Streams. *Communications of ACM*, 48(1):82–88, 2005.

[11] R. Fonseca, V. Almeida, M. Crovella, and B. Abrahao. On the Intrinsic Locality Properties of Web Reference Streams. In *Proc. the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom)*, 2003.

[12] S. Glassman. A caching relay for the World Wide Web. In *Proc. the First International World Wide Web Conference*, pages 69–76, 1994.

[13] S. Jin and A. Bestavros. Sources and Characteristics of Web Temporal Locality. In *Proc. of the 8th Int. Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*. IEEE Computer Society Press, Agosto 2000.

[14] A. Mahanti, D. Eager, and C. Williamson. Temporal Locality and its Impact on Web Proxy Cache Performance. *Performance Evaluation Journal: Special Issue on Internet Performance Modelling*, 42(2/3):187–203, Setembro 2000.

[15] V. Phalke and B. Gopinath. An Interreference Gap Model for Temporal Locality in Program Behavior. In *Proc. the 1995 ACM SIGMETRICS Conference*, pages 291–300, 1995.

[16] J. Robinson and M. Devarakonda. Data Cache Management Using Frequency-Based Replacement. In *Proc. the 1990 ACM SIGMETRICS conference on Measurement and modeling of computer systems*, pages 134–142. ACM Press, 1990.

[17] J. Wang. A Survey of Web Caching Schemes for the Internet. *ACM Computer Communication Review*, 25(9):36–46, 1999.

[18] D. Weikle, S. Mckee, and W. Wulf. Cache as Filters: A New Approach to Cache Analysis. In *6th Intl. Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'98)*, Julho 1998.

[19] C. Williamson. On Filter Effects in Web Caching Hierarchies. *ACM Transactions on Internet Technology*, 2(1):47–77, Fevereiro 2002.