Desafios na Gerência de Cache Web Multi-locatários

Anna Lira[1], **João Ramalho**[1], **Ruan Alves**[1]
Thiago Emmanuel Pereira[1], **Francisco Vilar Brasileiro**[1]
Mariana Mendes[2]

[1]Universidade Federal de Campina Grande (UFCG)
Caixa Postal 10.106 – 58.109-970 – Campina Grande – PB – Brazil

[2]VTEX
Rio de Janeiro, RJ - Brazil

{anna.lira,ruan.alves,joao.ramalho}@lsd.ufcg.edu.br

{temmanuel,fubica}@computacao.ufcg.edu.br, mariana.mendes@vtex.com

Abstract. *Although caching is far from being a newly technique for improving system performance, there are still challenges in its operation. For example, multi-tenant web platforms have little support, both in methodology and tooling, to manage their multiple workloads. In particular, it is not clear the trade-offs on choosing between multiple exclusive caches and a single shared cache. This work is the result of a cooperation with an industrial partner that operates a large e-commerce platform. The purpose of the cooperation is to better understand the challenges in multi-tenant cache management. To do this, we characterize this partner's workload and discuss some strategies that can support the management of multi-tenant caches.*

Resumo. *Caching é uma técnica antiga e consolidada para melhorar o desempenho de sistemas, porém ainda existem desafios em sua operação. Em plataformas web com múltiplos locatários, não é fácil decidir, por exemplo, se a carga de trabalho gerada pelos vários locatários deve ser atendida de forma exclusiva por vários caches ou de forma compartilhada por um único cache. Neste trabalho, descrevemos a cooperação com um parceiro industrial que opera uma grande plataforma de comércio eletrônico. Tivemos como objetivo entender melhor os desafios na gerência de cache compartilhada entre vários locatários. Para isso, caracterizamos a carga de trabalho desse parceiro e discutimos algumas estratégias que podem apoiar a gerência desse tipo de cache.*

1. Introdução

Para obter melhor desempenho e eficácia, sistemas de cache web precisam ser configurados de acordo com as características da carga de trabalho às quais estão submetidos. Uma vez entendidas características como nível da carga, localidade temporal, *footprint*, entre outras, são definidos parâmetros importantes, incluindo: a capacidade do cache, algoritmos de reposição (e admissão) de itens e quantidade de servidores que serão usados no serviço de cache.

Esses procedimentos de observação e configuração foram bem discutidos na literatura e são conhecidos na indústria, principalmente para o caso típico no qual caches

web são usados por somente uma aplicação. Entretanto, esse não é o caso de quando, em uma mesma organização, múltiplas aplicações web usam cache. Por exemplo, não é fácil decidir se cada aplicação deve ter acesso a um cache próprio, privado, ou se múltiplas aplicações devem compartilhar o serviço de cache. Por um lado, o uso de cache privado evita a interfência de uma aplicação no desempenho de outras. Por outro lado, manter vários serviços de cache aumenta a complexidade de operação na organização. Ainda, um cache compartilhado pode ser configurado com menor capacidade do que a soma de caches privados correspondentes, tirando proveito do fato de que o pico da demanda das aplicações pode não acontecer ao mesmo tempo.

Este trabalho descreve os resultados obtidos em um projeto de cooperação com a empresa VTEX[1], cujo objetivo principal foi mapear desafios para a gerência de caches em ambientes com múltiplos locatários (*multi-tenants* em inglês). Para esse estudo, consideramos o caso da plataforma de comério eletrônico construída pela empresa parceira. Essa plataforma hospeda diversas lojas, de diferentes empresas. Cada loja tem bases de clientes e produtos próprios. Os inventários de produtos das lojas têm tamanho distintos. Há lojas com diferentes padrões de venda: sazonais e esporádicos. O acesso aos diferentes inventários é feito através de um serviço de cache compartilhado, gerenciado pela plataforma.

Este artigo resume alguns resultados técnicos obtidos com essa cooperação. A Seção 2 descreve o padrão de uso do serviço de cache da empresa parceira, durante um período de observação de 10 horas, no qual milhares de lojas locatárias da plataforma são acessadas. Descrevemos também, de modo breve, alguns aspectos técnicos da coleta de dados e do método de análise de dados, que foram discutidos em mais detalhes em um outro artigo [Lira et al. 2024]. Em seguida, descrevemos algumas implicações das análises realizadas (Seção 3). Por fim, na Seção 4, apresentamos os próximos passos e possíveis impactos dessa parceria.

2. Observações do ambiente de produção

Esta seção apresenta uma visão geral das características da carga de trabalho que coletamos do ambiente de produção da VTEX. A empresa atua como um provedor global de comércio eletrônico *business-to-consumer* (B2C) e *business-to-business* (B2B). A plataforma desenvolvida pela VTEX oferece suporte para empresas de comércio eletrônico na criação e operação de suas lojas *online*. Neste trabalho, nosso foco está no sistema de catálogo dessa plataforma, que gerencia dados não-efêmeros dos produtos disponíveis na plataforma (p.ex., títulos, descrições, identificadores de produtos, identificadores de locatários). O sistema de catálogo expõe, através de uma interface HTTP, informações dos produtos armazenados em um banco de dados distribuído. Uma camada de cache, implementada com servidores NGINX[2], atua como um proxy para a API HTTP do sistema de catálogo.

Para coletar a carga de trabalho, instrumentamos os servidores NGINX por um período contínuo de 10 horas, entre 21h e 7h GMT-3, dos dias 13 e 14 de dezembro de 2022. Foram coletadas as seguintes informações para cada requisição: o URI da requisição, que contém as identificações do produto e do locatário; o *status* da resposta

[1]https://vtex.com

[2]Documentação Nginx. https://nginx.org/en/docs/

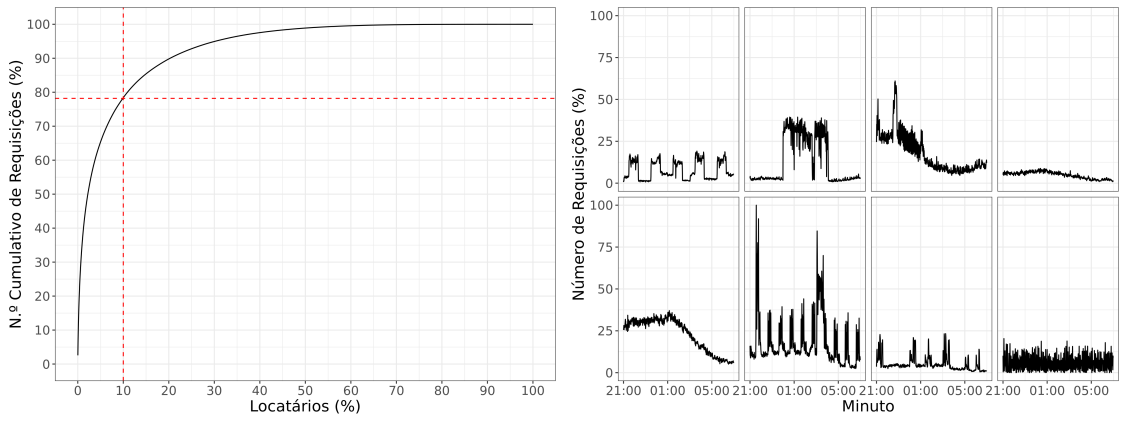
do cache, indicando se a resposta foi um desacerto (*miss*) ou um acerto (*hit*); a hora (*timestamp*) em que a requisição chegou à camada de cache; e, o tempo de resposta para atendimento da requisição. Durante o período de observação foram coletadas 56 milhões de requisições, para 25 milhões de itens de cerca de 5 mil lojas. Os dados foram anonimizados usando a normalização min-max, onde o máximo número de requisições em um minuto é representado por 1 e o mínimo por 0, considerando todos os tempos e locatários observados nos dados. Todos os identificadores foram anonimizados usando uma função *hash*.

2.1. Características da Carga de Trabalho

Relatamos algumas característica da carga de produção usando análises já consolidadas na literatura. Na seção 2.2, discutimos a demanda e sua concentração entre as lojas que fazem parte da plataforma. Na seção 2.3, analisamos o *footprint* da carga de produção, que é uma medida da quantidade de dados acessados em um intervalo de tempo, considerando o número de itens distintos nesse intervalo [Xiang et al. 2013]. A seção 2.4 descreve a popularidade dos itens acessados durante o período de observação. A popularidade dos itens foi medida através da taxa de repetição de acesso a itens (IRR, *Item Repetition Rate* em inglês). A IRR indica um limite superior da taxa de acerto [Gu et al. 2023]. Ou seja, essa característica da carga indica o máximo potencial de desempenho que o serviço de cache pode cumprir. Por melhor que um cache esteja projetado e dimensionado, não é possível melhorar o desempenho de uma carga não-amigável ao cache (com baixa repetição de acesso a itens).

2.2. Demanda e Popularidade

A demanda submetida ao cache varia bastante ao longo do dia. A quantidade média de requisições do minuto com maior carga é $210,43\%$ maior do que no minuto com menor carga. A disparidade entre os locatários é evidente, com apenas 10% deles respondendo por uma parcela de aproximadamente 80% do total de requisições, como ilustra a Figura 1a, destacando a distribuição desigual. Além disso, a quantidade de requisições por locatário varia ao longo do tempo (Figura 1b), o que torna a alocação de recursos mais complexa.



(a) Porcentagem cumulativa de número de requisições feitas por locatário.

(b) Carga normalizada de uma amostra dos maiores locatários.

Figura 1. Demanda.

Há também grande disparidade na popularidade de acesso aos itens. A Figura 2 mostra que cerca de 50% das requisições concentra-se em apenas 10% dos itens. Itens com baixa frequência de solicitação, muitas vezes não justificam os recursos alocados para seu armazenamento no cache, causando má utilização. As observações dessa seção sugerem que nem todos os itens são adequados para armazenamento em cache. Portanto, uma política de controle de admissão pode ser eficaz para melhorar o uso de recursos do sistema.

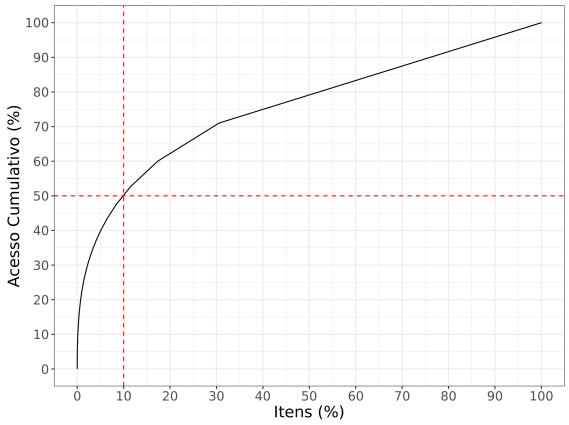


Figura 2. Percentual de requisições por conjunto de itens.

2.3. Footprint

A carga analisada apresenta *footprint* total de cerca de 25 milhões de itens. O footprint de um dado locatário é dado pelo número de itens únicos daquele locatário constantes na carga de trabalho coletada. Há uma forte correlação entre o número de requisições e o *footprint*, com um coeficiente de correlação de Pearson de $0{,}88$, o que pode indicar que o número de itens distintos tende a aumentar conforme o número de requisições. O *footprint* dos locatários muda ao longo do tempo, influenciado pelo padrão da carga de trabalho e mudanças no número de requisições. A Figura 3 ilustra essa dinamicidade, mostrando a variação, a cada hora, no *footprint* dos seis maiores locatários, considerando o número de requisições. Isso evidencia que o *footprint* dos locatários não é estático.

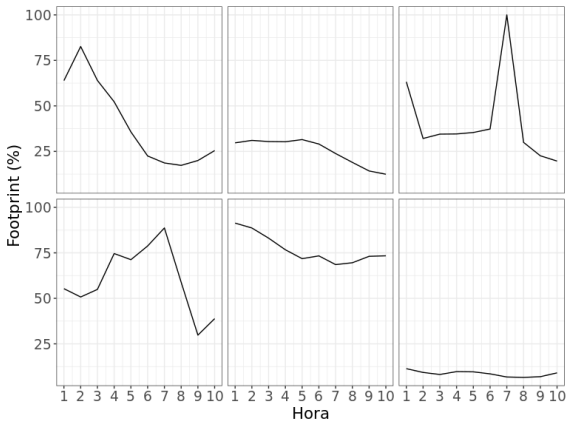
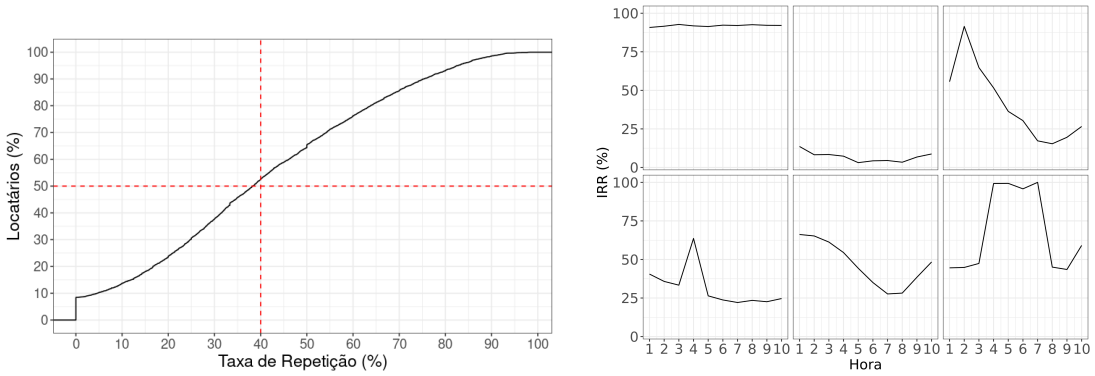


Figura 3. Footprint ao longo do tempo para uma amostra de locatários.

2.4. Taxa de Repetição de Acesso a Itens

A distribuição de IRR dos locatários sugere que alguns se beneficiam mais do cache do que outros. Cerca de 50% dos locatários têm IRR igual ou inferior a 40%, como observado na Figura 4a, indicando baixa repetição de acesso a itens e menos eficácia do cache para esses locatários. Isso sugere a necessidade de estratégias adaptativas para gerenciar o cache, identificando esses padrões e otimizando o desempenho do sistema.



(a) Distribuição de locatários conforme o IRR.

(b) IRR ao longo do tempo para uma amostra de locatários.

Figura 4. Análise da taxa de repetição de acesso aos ítens

Ao reduzir o período de análise do IRR para uma hora (em vez de usar todo o período observado), é possível perceber mudanças no comportamento do IRR ao longo do tempo, como mostra a Figura 4b; alguns locatários apresentam variação significativa em seus IRR. Isso destaca a importância de avaliar regularmente a adequação do cache.

3. Discussão

Os níveis de carga e padrões de acesso observados influenciam o gerenciamento de cache e a alocação de recursos em um ambiente com múltiplos locatários. Esta seção discute algumas possíveis implicações das observações realizadas, destacando possíveis estratégias para otimizar o desempenho do cache.

Um primeira implicação é que itens com baixa frequência de acesso podem não justificar sua inclusão no cache. Quando incluídos podem ocupar espaço que poderia ser usado para armazenar itens que tirariam melhor proveito do cache. Por exemplo, no sistema de cache estudado, 50% dos itens armazenados correspondem a apenas 20% das requisições, indicando baixa frequência de acesso, apesar do uso significativo dos recursos. Estratégias como Lazy Adaptive Cache Replacement [Huang et al. 2016] e CacheSack [Yang et al. 2023], que impedem a entrada de itens pouco populares, poderiam ser utilizadas. Essas abordagens podem ser potencialmente melhoradas com o conhecimento do padrão de carga dos locatários.

Também pode fazer sentido avaliar controle de admissão na granularidade do locatário. Ou seja, um certo locatário, em algum momento, não deveria usar o cache. Como vimos, parte considerável dos locatários tem baixo IRR, de modo que terão baixa taxa de acerto no cache, independentemente de como este funcione. Para a carga de trabalho que estudamos, é possível reduzir cerca de 10% da capacidade do cache (mantendo o mesmo desempenho) removendo os locatários com IRR menor ou igual a 20% [Lira et al. 2024].

De modo geral, podemos consolidar os resultados das análises feitas em duas observações: 1) de fato, mesmo em uma plataforma com um mesmo propósito (comércio eletrônico), há grande variedade nas característica da carga de trabalho do locatários que usam a plataforma; 2) as características dos locatários não são estáticas. Como implicações, temos que: 1) é preciso levar em consideração que os locatários existem, ao configurar o cache; 2) as configurações do cache devem ser modificadas ao longo do tempo para melhorar a eficiência do sistema.

Um primeiro passo para aprofundar as implicações indicadas é revisitar os estudos de políticas de gerenciamento, para que sejam cientes aos locatários, incluindo: planejamento de capacidade, reposição e controle de admissão de itens. É possível também que, dado que as características da carga são dinâmicas, políticas de definição de capacidade precisem ser realizadas em tempo de execução. Ou seja, é possível que seja útil ter mecanismos de *auto-scaling* para gerência de recursos de cache.

4. Conclusões

Este artigo apresentou alguns desafios e oportunidades para melhorar a gerência de cache usados por vários locatários. Com esse propósito, coletamos e analisamos uma carga de trabalho de um sistema de cache usado em produção, compartilhado entre milhares de locatários. Acreditamos que os dados coletados dessa carga podem ser úteis para outros pesquisadores e consideramos como trabalho futuro descrever em mais detalhes a carga coletada e disponibilizá-la para uso público.

Por fim, consideramos também avaliar novas políticas de gerência de cache cientes de locatários. Em particular, é importante entender o que é possível implementar, com menor ou maior dificuldade, em ferramentas *off-the-self* já disponíveis, incluindo sistemas de caching como *NGINX, memcached* e *Redis*.

5. Agradecimentos

Este trabalho foi financiado pelo MCTIC/CNPq-FAPESQ/PB (EDITAL Nº 010/2021) e pela VTEX BRASIL (EMBRAPII PCEE1911.0140).

Referências

Gu, R., Li, S., Dai, H., Wang, H., Luo, Y., Fan, B., Basat, R. B., Wang, K., Song, Z., Chen, S., Wang, B., Huang, Y., and Chen, G. (2023). Adaptive online cache capacity optimization via lightweight working set size estimation at scale. In *USENIX ATC 2023*, pages 467–484.

Huang, S., Wei, Q., Feng, D., Chen, J., and Chen, C. (2016). Improving flash-based disk cache with lazy adaptive replacement. *ACM Trans. Storage*, 12(2):8:1–8:24.

Lira, A., Alves, R., Pereira, T. E., Morais, F., Ramalho, J., and Mendes, M. (2024). No clash on cache: Observations from a multi-tenant ecommerce platform. In *Proceedings of the 2024 ACM/SPEC International Conference on Performance Engineering*.

Xiang, X., Ding, C., Luo, H., and Bao, B. (2013). HOTL: a higher order theory of locality. In *ASPLOS 2013*, pages 343–356. ACM.

Yang, T., Pollen, S., Uysal, M., Merchant, A., Wolfmeister, H., and Khalid, J. (2023). Cachesack: Theory and experience of google's admission optimization for datacenter flash caches. *ACM Trans. Storage*, 19(2):13:1–13:24.