

MEMCACHED

Uma alternativa de cache para arquiteturas de alta escalabilidade

Felipe Bergamaschi

O que é *CACHE*?

Em ciência da computação, cache é uma coleção de dados previamente computados, armazenados em algum local de acesso mais rápido que o original, utilizado em casos onde o dado original exige muito esforço computacional para ser obtido ou gerado.

<https://pt.wikipedia.org/wiki/Cache>

Quem é Memcache?

É um serviço de **alta performance** de cache em memória **distribuído** e **genérico** por natureza, mas destinado a ser usado na aceleração de aplicativos dinâmicos da web, aliviando por exemplo a carga do banco de dados.

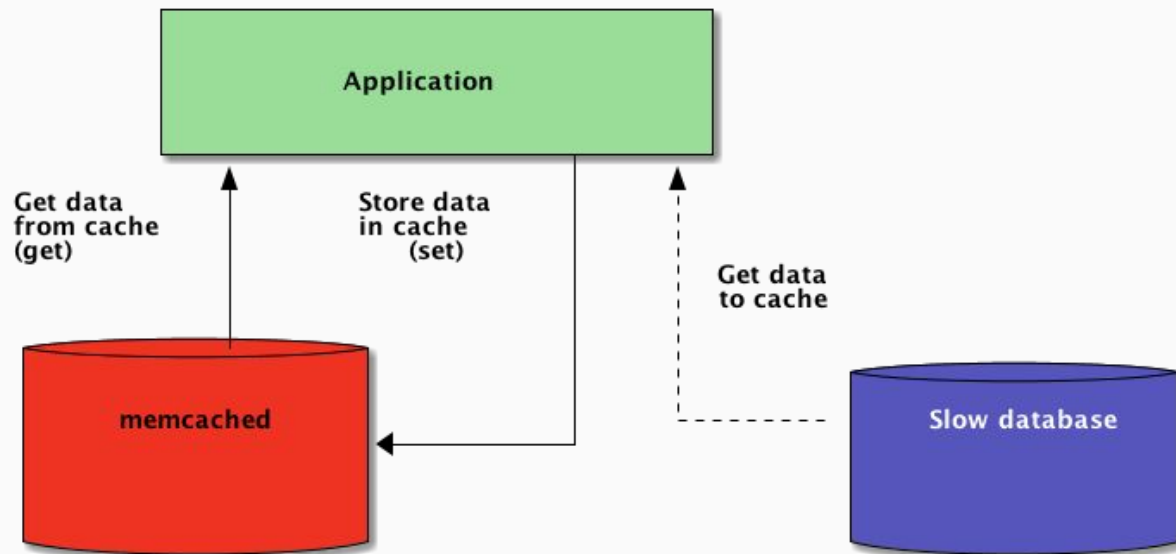
Memcached é **simples**, mas **poderoso**. Seu design simples promove implantação rápida, facilidade de desenvolvimento e resolve muitos problemas enfrentados por grandes caches de dados.

Memcached vs. Memcache

Memcache é o módulo que fornece a relação processual e orientada a objeto para o memcached, cache daemon altamente eficaz, que foi especialmente projetado para reduzir a carga de banco de dados em aplicações dinâmicas.

Memcached é o software de alta performance. Memória distribuída baseado em sistema de cache de objeto, de natureza genérica, mas destinado a ser utilizado para acelerar aplicações dinâmicas, aliviando a carga do banco de dados.

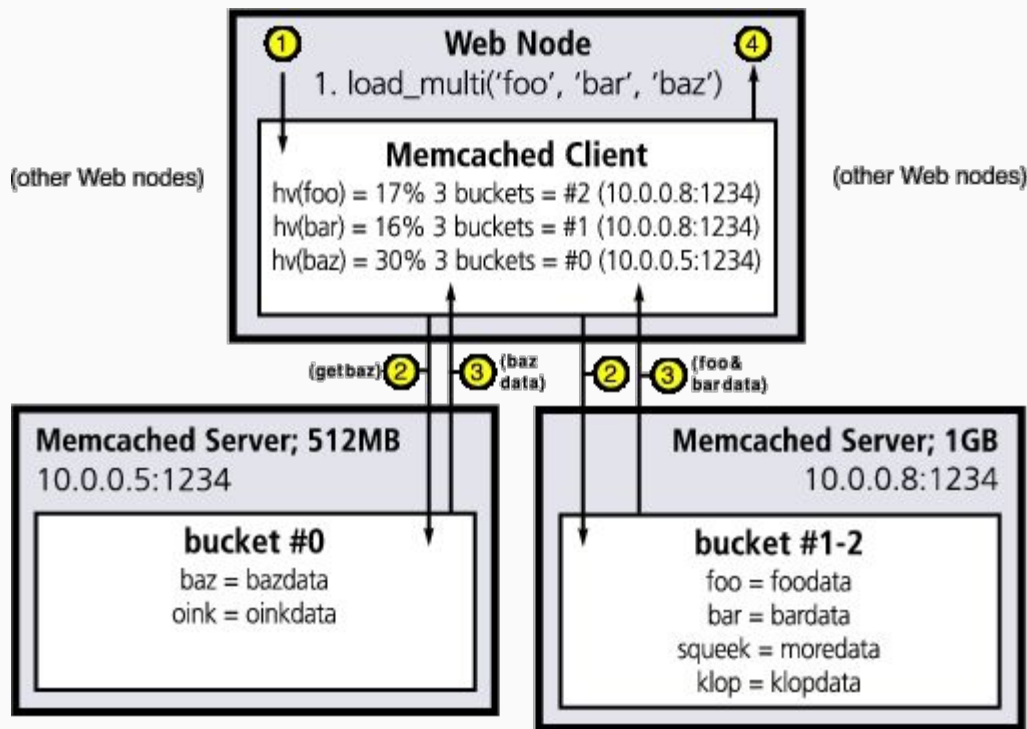
Diagrama



Exemplo

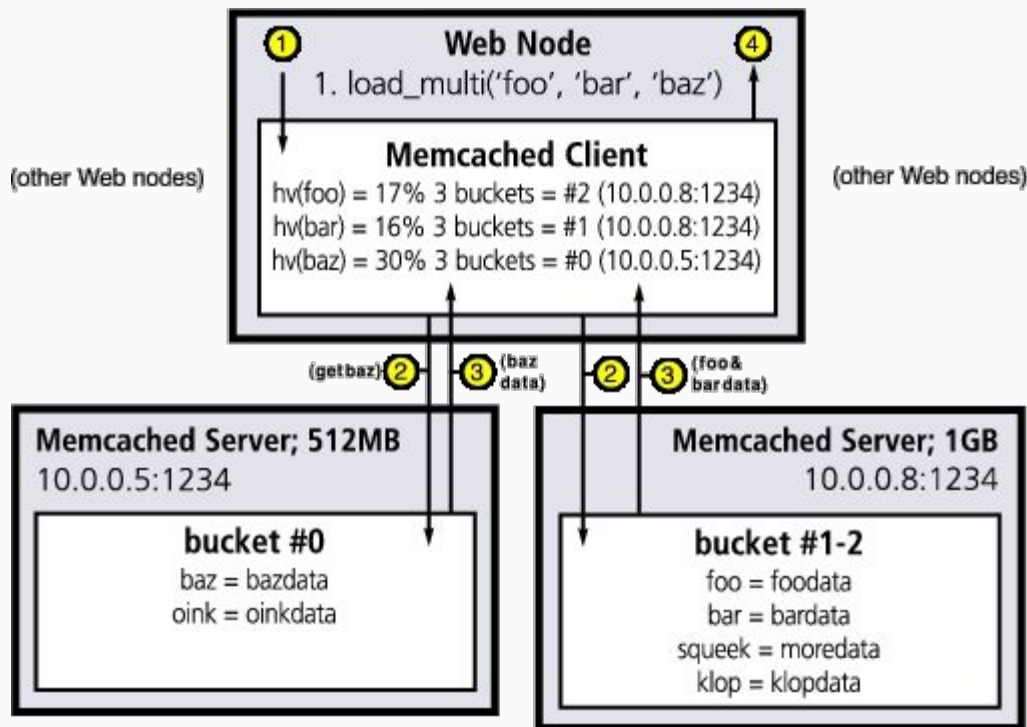
A aplicação solicita as chaves FOO, BAR e BAZ pelo client Memcached.

O client calcula o hash das caches e determina qual servidor atenderá a requisição.



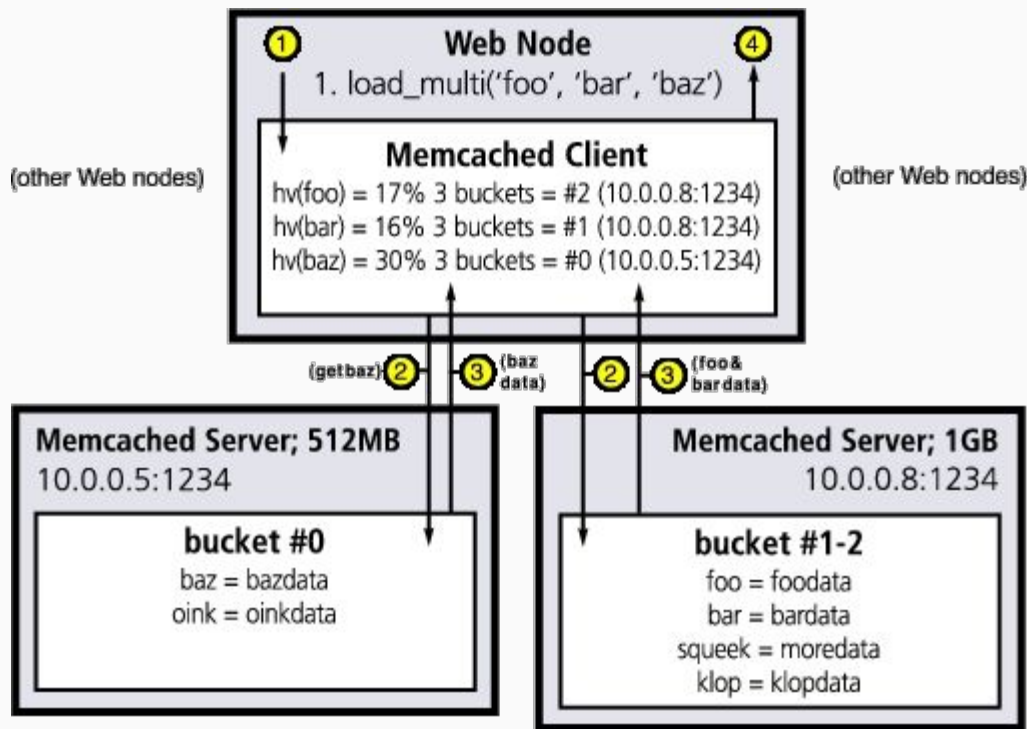
Exemplo

O client do Memcached faz paralelamente às requisições a todos os servidores identificados.



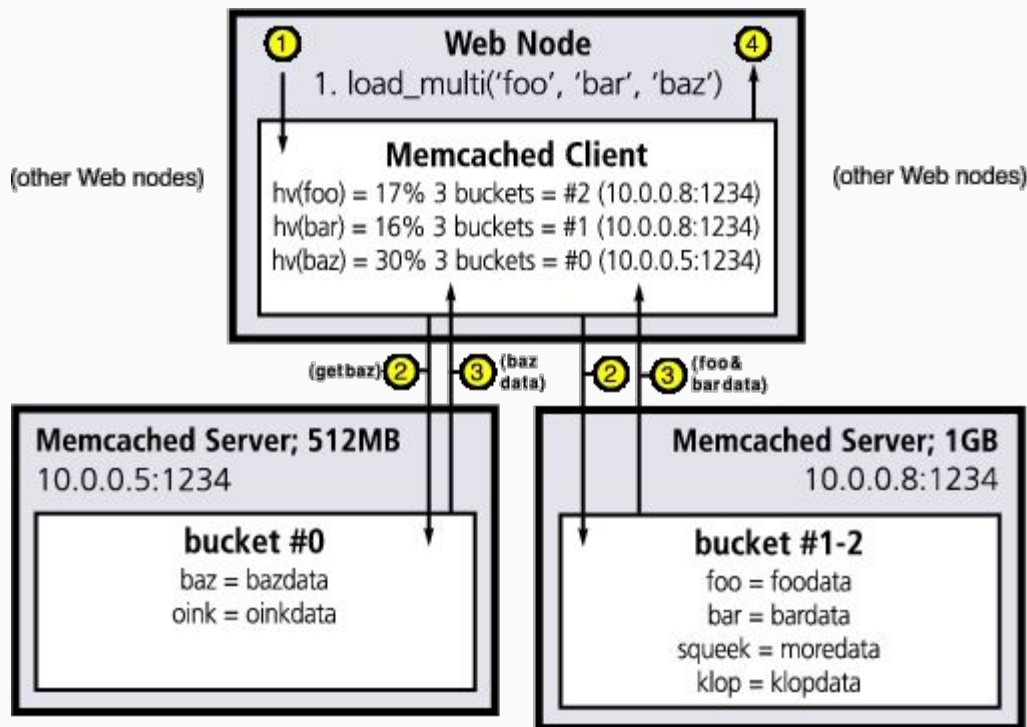
Exemplo

O servidor do Memcached retorna as respostas ao client que fez a requisição.



Exemplo

O client do Memcached retorna os dados para a aplicação, encapsulando toda a complexidade da infraestrutura.



Filosofia de design

Simple Key/Value Store

O servidor não se importa com a aparência dos seus dados.

Os itens são compostos de uma chave, um tempo de expiração, sinalizadores opcionais e dados brutos.

Não compreende estruturas de dados; você deve carregar dados pré-seriados.

Alguns comandos (incr / decr) podem operar nos dados subjacentes, mas de uma maneira simples.

Filosofia de design

Logic Half in Client, Half in Server

Uma "implementação do memcached" é parcialmente em um cliente e parcialmente em um servidor.

Os clientes sabem como escolher qual servidor ler ou gravar um item e o que fazer quando não pode contatar um servidor.

Os servidores entendem como armazenar e buscar itens. Eles também gerenciam quando despejar ou reutilizar memória.

Filosofia de design

$O(1)$ - *Big O*

Todos os comandos são implementados para serem o mais rápidos e fáceis de bloquear possíveis.

Isso permite velocidades de consulta quase determinísticas para todos os casos de uso.

Consultas em máquinas lentas devem rodar bem abaixo de 1ms. Servidores de última geração podem servir milhões de chaves por segundo em taxa de transferência.

<https://www.adayinthelifeof.nl/2009/12/21/big-o-notation/>

Filosofia de design

Forgetting is a Feature

O Memcached é, por padrão, um **Least Recently Used** cache. (*LRU*) - *Descarta os itens menos usados recentemente primeiro.*

Itens expiram após um período de tempo especificado.

- Expirar itens após um minuto para limitar o retorno de dados obsoletos
- Liberar dados não utilizados em uma operação para reter informações solicitadas com frequência.

Nenhum "pause" à espera de um *garbage collector* garante baixa latência e o espaço livre é recuperado sob demanda.

[https://en.wikipedia.org/wiki/Cache_replacement_policies#Least_recently_used_\(LRU\)](https://en.wikipedia.org/wiki/Cache_replacement_policies#Least_recently_used_(LRU))

Filosofia de design

Cache Invalidation

Em vez de transmitir as alterações para todos os hosts disponíveis, os clientes abordam diretamente o servidor que contém os dados a serem invalidados.

Hardware

Em suma, o uso da CPU é geralmente baixo, a memória será conforme disponibilizado e o uso de rede irá variar de leve a moderado, dependendo do tamanho médio de seus itens.

Memcached é tipicamente leve no uso da CPU, devido ao seu objetivo de responder muito rápido. Memcached é multithreaded, padronizando em 4 threads.

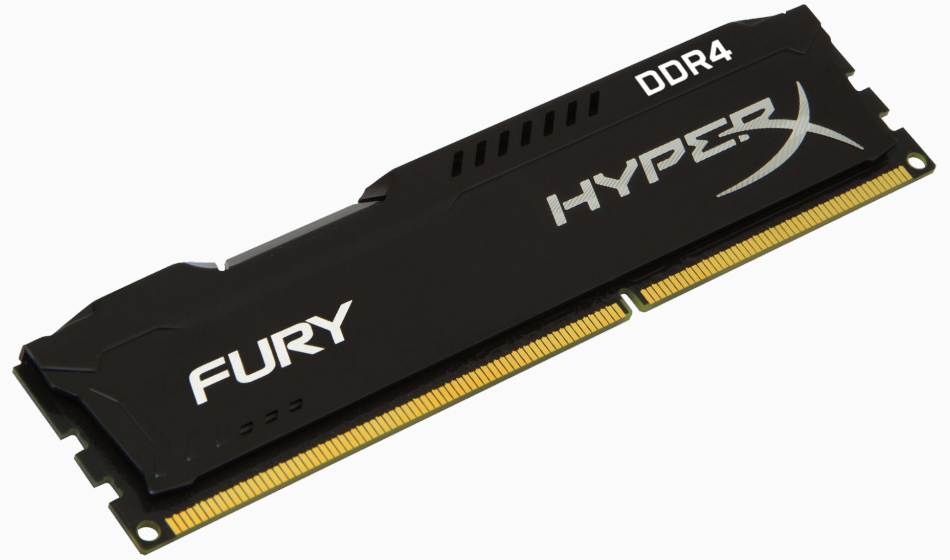
A maioria das instalações só precisa de uma thread.



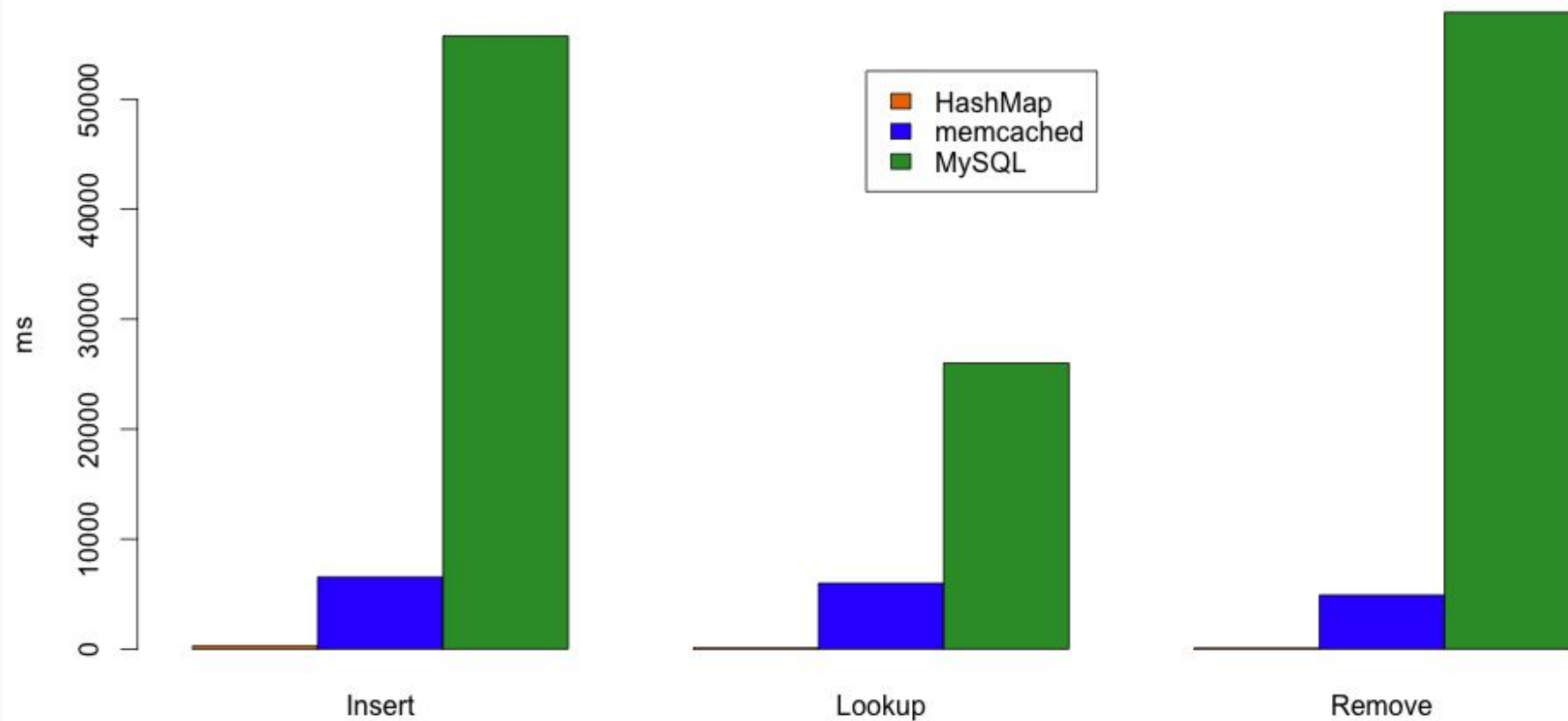
Hardware

O ponto principal do memcached é unificar seções de memória de vários hosts e fazer com que seu aplicativo o veja como uma grande e única memória.

Quanto mais memória melhor.



Times for execution



Hora do código.

TCC

Implantando em um
ambiente real



Obrigado!

Felipe Bergamaschi

Email

felipebergamaschi0@gmail.com

Linkedin

/felipebergamaschi

Github

/felipebergamaschi

Este projeto

github.com/felipebergamaschi/meetup-me-mcached-php-docker

