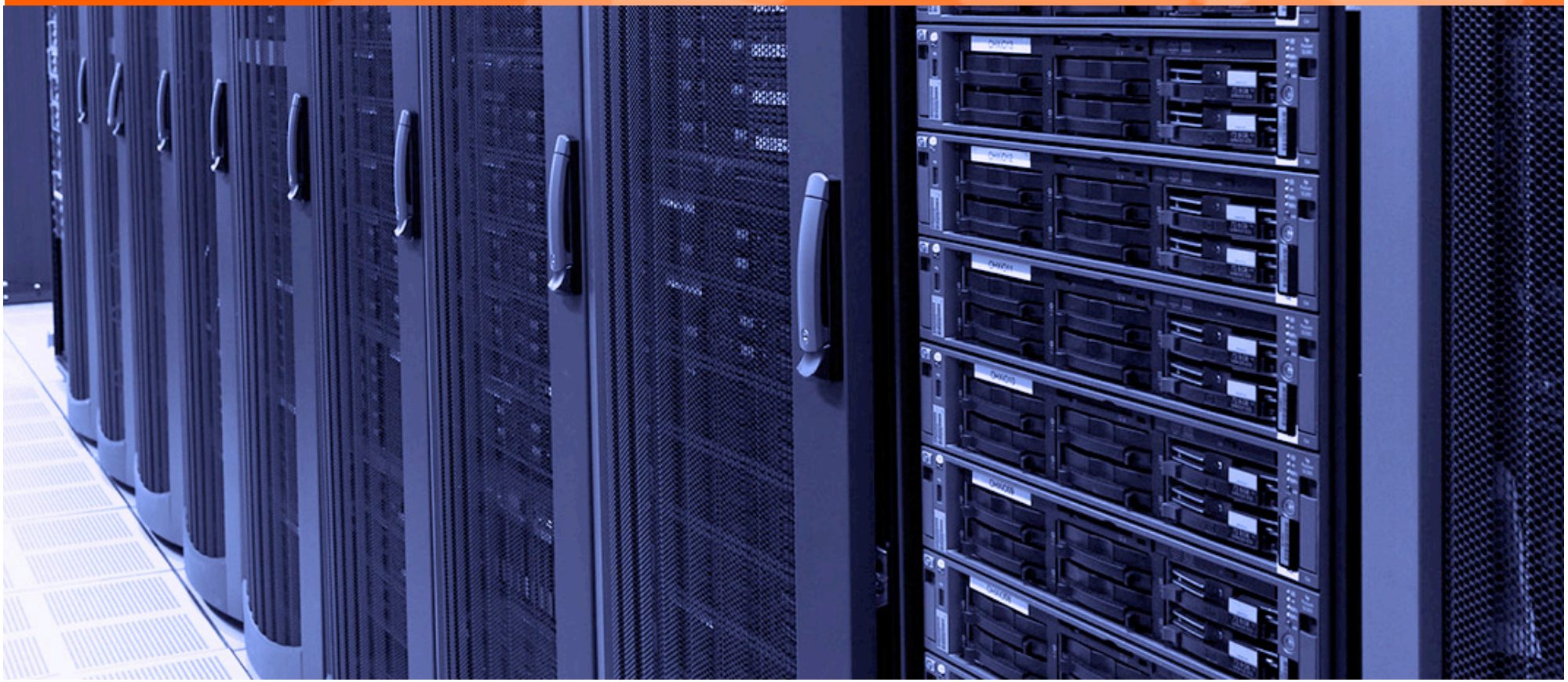
Melhorando a performance de aplicações com o uso do MemCache



Osdeni José Sadzinski

Elton Luís Minetto

drimio

drimio

Osdeni José Sadzinski

- ✓ Graduado e pós-graduado em Sistemas de Informação. Cursando MBA em Gerenciamento de Projetos
- ✓ Trabalha com PHP desde 2004
- ✓ Membro do PHPSC
- ✓ Desenvolvedor Web da Drimio

Elton Luís Minetto

- ✓ Graduado e pós-graduado em Ciência da Computação. Cursando MBA em Gerenciamento de Projetos
- ✓ Trabalha com PHP desde 2000
- ✓ Autor dos livros Frameworks para Desenvolvimento em PHP - Editora Novatec e Grid Computing in Research and Education - IBM RedBooks
- ✓ Membro do PHPSC
- ✓ Gerente de Desenvolvimento da Drimio e professor na Sociesc (Joinville)/Unochapecó(Chapecó)

Terminologia

- ✓ Performance: a habilidade que uma aplicação tem de atingir um objetivo, como por exemplo responder no menor tempo possível
- ✓ Capacidade: a carga total que uma aplicação pode suportar
- ✓ Escalabilidade: a habilidade de uma aplicação manter a performance quando a carga de trabalho aumenta. É a junção da capacidade e da performance

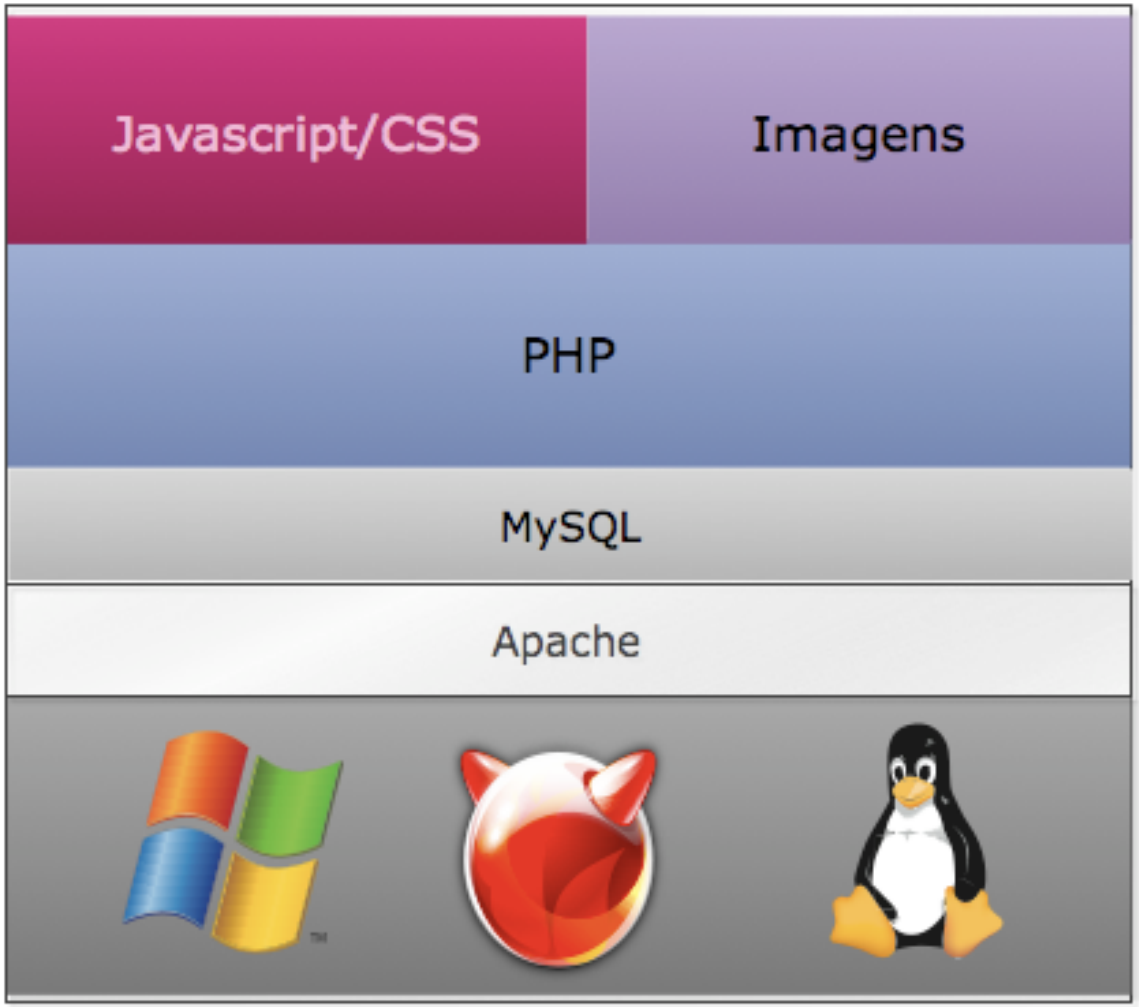
Ou…

drimio

Ou…

- ✓ **Performance: a velocidade do carro**
- ✓ **Capacidade: o limite de velocidade e o número de pistas da estrada**
- ✓ **Escalabilidade: quantos carros e pistas eu posso adicionar sem diminuir a velocidade do tráfego**
  - ✓ **“Performance is a problem. Scaling your performance is a bigger problem”**

Camadas



Memcached

- ✓ Sistema distribuído e de alta performance para fazer cache de objetos em memória RAM. É genério por natureza mas muito usado para acelerar aplicações web dinâmicas, reduzindo a carga de bases de dados, sessões de usuários, arquivos CSS.
- ✓ Foi desenvolvido pela Danga Interactive para aumentar a performance do site LiveJournal.com, que possui mais de 20 milhões de page views por dia e atende 1 milhão de usuários. memcached reduziu a carga dos servidores de banco de dados forncecendo páginas mais rápidas e melhor utilização de recursos.

Porque usar?

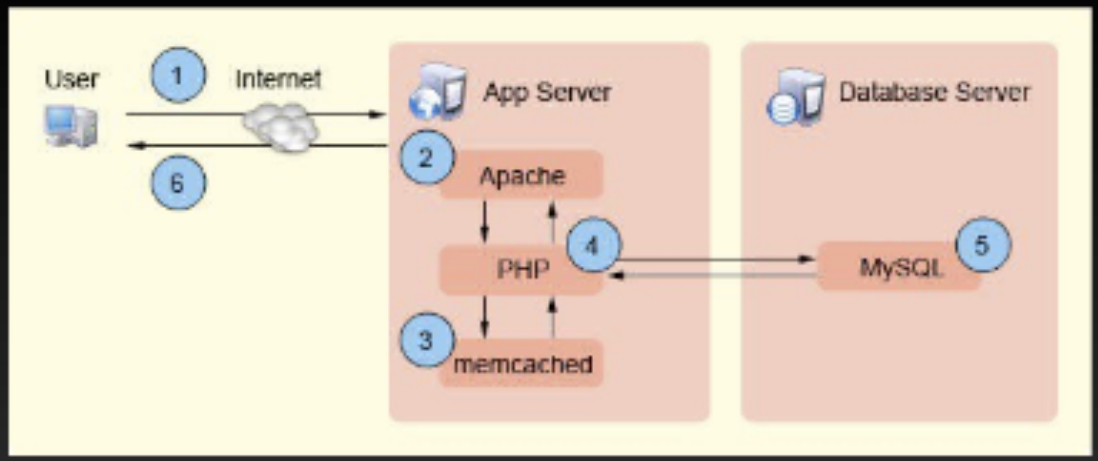
- ✓ Escalável: fácil adicionar máquinas e instâncias
- ✓ RAM é muito mais rápido que I/O em disco
- ✓ Alivia a carga do banco de dados
- ✓ Muito flexível: pode armazenar qualquer dado, desde que não ultrapasse 1 MB
- ✓ Bibliotecas client disponíveis em diversas linguagens (PHP, C, Java, Ruby, Python)

Como funciona?

- ✓ Armazena qualquer coisa que pode ser serializado
- ✓ Armazena com uma chave única (255 caracteres) e tempo de validade
- ✓ O programador usa da seguinte forma :
  - ✓ Encontrou no cache? Retorna os dados
  - ✓ Não encontrou no cache? Processa, armazena no cache e retorna o dado

Como funciona?

How memcached Works



1. User requests web page
2. Web server passes request to PHP
3. PHP checks if object is in memcached
4. If yes, build page with object
5. If no, PHP gets object from DB & stores in memcached
6. Send page to user

O que armazenar?

- ✓ Resultados de consultas SQL
- ✓ Páginas inteiras
- ✓ Sessões de usuários
- ✓ Fragmentos de HTML gerados
- ✓ Imagens (thumbnails?)
- ✓ Arquivos css
- ✓ Arquivos js
- ✓ etc

Memcached – Como instalar (Linux)

- ✓ Baixar código em http://www.danga.com/memcached/
- ✓ Seguir o procedimento:
  - ✓ tar -xvzf memcached-1.2.8.tar.gz
  - ✓ cd memcached-1.2.8
  - ✓ ./configure
  - ✓ make
  - ✓ make install (executar como root)
- ✓ No exemplo abaixo iniciamos o daemon com 128MB RAM, ouvindo na porta 11211 no ip 127.0.0.1
  - ✓ /usr/local/bin/memcached -d -m 128 -l 127.0.0.1 -p 11211
- ✓ Pode-se criar quantas instâncias forem necessárias, mudando a porta. Assim aumenta-se o espaço total do cache

PHP e Memcached (em Linux)

- ✓ Download do módulo em http://pecl.php.net/package/memcache.
- ✓ Executar:
  - ✓ tar -xvzf memcache-2.2.5.tgz
  - ✓ cd memcache-2.2.5
  - ✓ phpize
  - ✓ ./configure
  - ✓ make
  - ✓ make install (executar como root)
- ✓ Adicionar a linha abaixo no php.ini e reiniciar o Apache
  - ✓ extension=/usr/local/lib/php/extensions/memcache.so (o caminho é indicado no make install)

API

- ✓ Comandos de armazenamento:
  - ✓ set, add, replace, append
- ✓ Comandos de recuperação:
  - ✓ get
- ✓ Comandos de exclusão:
  - ✓ delete
- ✓ Outros comandos:
  - ✓ stats, flush_all, version, verbosity, quit

Como usar com PHP

```
<?php //exemplo usando cliente PECL
$memcache = new Memcache;
$memcache->connect('localhost', 11211) or die ("Could not connect");
$version = $memcache->getVersion();
echo "Server's version: ".$version."<br/>\n";
$tmp_object = new stdClass;
$tmp_object->str_attr = 'test';
$tmp_object->int_attr = 123;
//params: chave, objeto, comprimimir ou não, tempo em segundos
$memcache->set('key', $tmp_object, false, 10) ;
$get_result = $memcache->get('key');
echo "Data from the cache:<br/>\n";
var_dump($get_result);
?>
```

Como usar com PHP

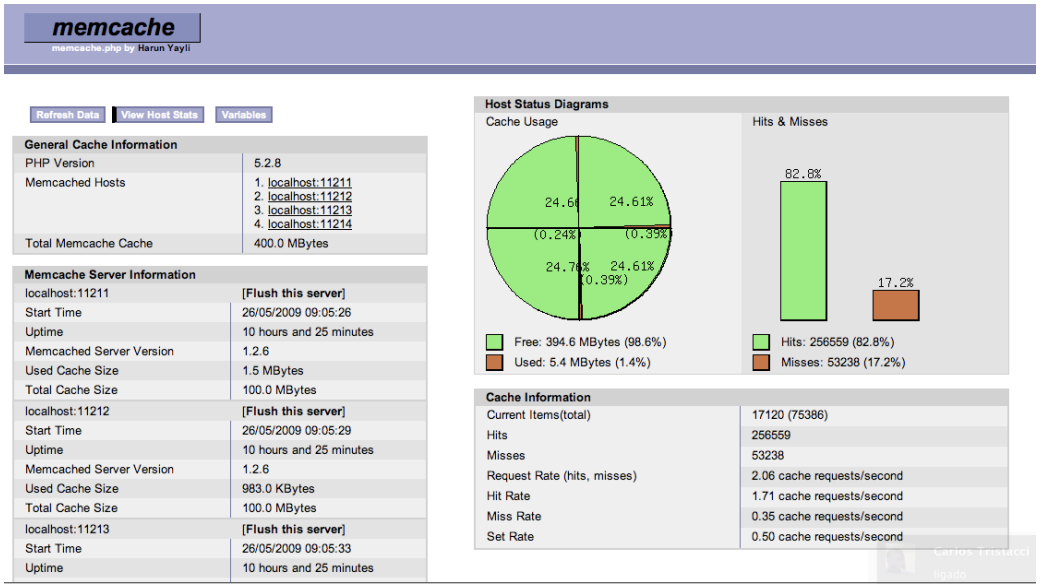
```
<?php
$memcache = new Memcache;
$memcache->connect('localhost', 11211);
$memcache->connect('localhost', 11212);
$memcache->connect('192.168.0.10', 11212);
$conteudo = $memcache->get('estados');
if($conteudo === false) {
   $uf = array('SC','RS','SP');
   $memcache->set('estados', $uf, false, 100) ;
   $conteudo = $uf;
}
var_dump($conteudo);
?>
```

Como usar com Zend Framework

```
$frontendOptions = array('lifetime' => 72000,'automatic_serial
   ization' => false );
$backendOptions = array('servers' => array(
array('host' => 'localhost', 'port' => 11211, 'persistent' =>
   true )));
$cache = Zend_Cache::factory('Core', 'Memcached' ,
   $frontendOptions , $backendOptions);
if(!$result = $cache->load('myresult')) {
   $db = Zend_Db::factory( [...] );
   $result = $db->fetchAll('SELECT * FROM huge_table');
   $cache->save($result, 'myresult');
} else {
   // cache hit! shout so that we know
   echo "This one is from cache!\n\n";
}
print_r($result);
```

Como monitorar

- ✓ Plugin para o Nagios
  - ✓ http://search.cpan.org/~zigorou/Nagios-Plugins-Memcached-0.02/lib/Nagios/Plugins/Memcached.pm
- ✓ memcache.php
  - ✓ http://livebookmark.net/journal/2008/05/21/memcachephp-stats-like-apcphp/

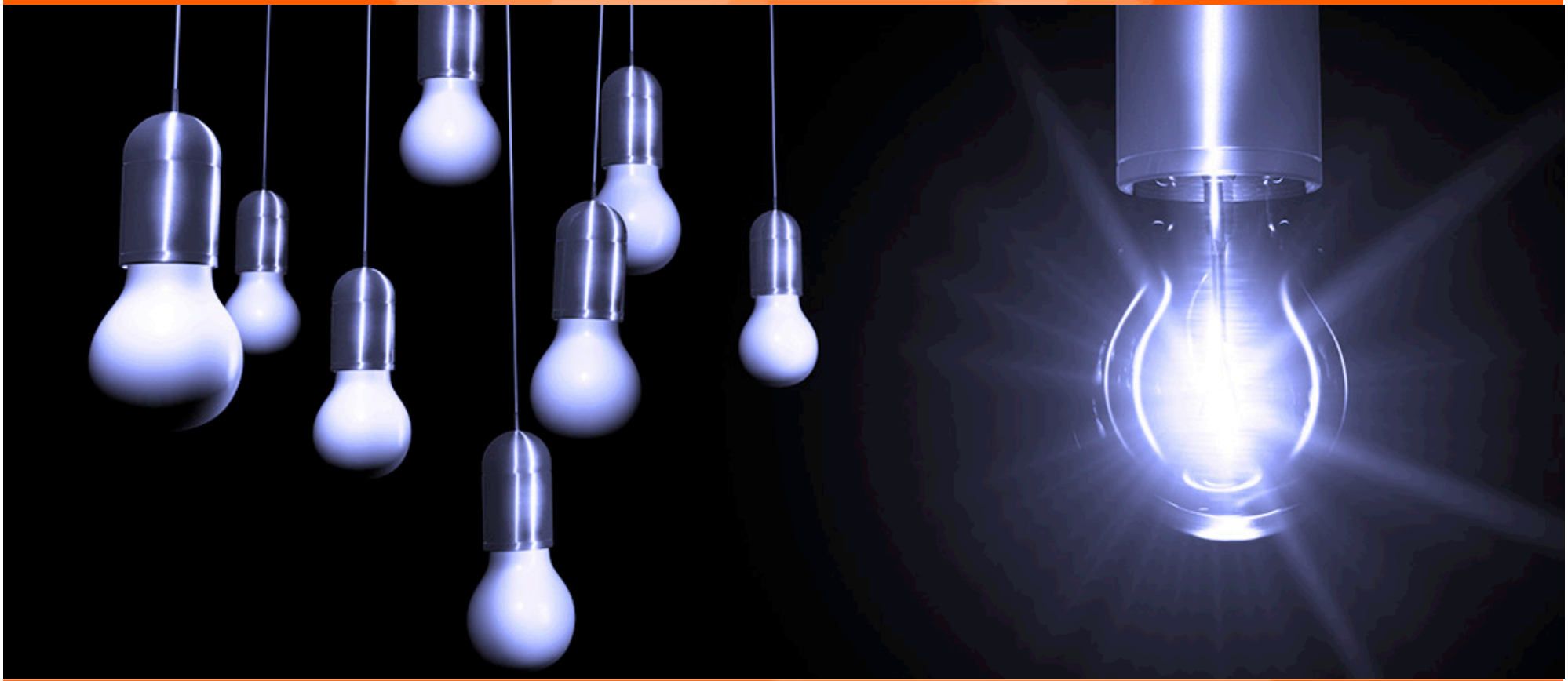


Quem usa?

- ✓ Facebook
  - ✓ 805 servidores 8-core. 28TB de RAM. 300K req/seg (Set. 2008)
  - ✓ Desenvolveu fork com suporte a UDP

  http://github.com/fbmarc/facebook-memcached/tree/master
- ✓ Flickr
  - ✓ 38K req/seg. 12 milhões de objetos (Set. 2008)
- ✓ Digg
- ✓ Inphonex - 6 servidores armazenando sessões de usuários
- ✓ Doctrine (sim! coloquei isso durante a palestra do Guilherme Blanco)

Drimio: Uma Rede Social focada em MARCAS



drimio

Drimio: Uma Rede Social focada em MARCAS

- ✓ **O Drimio** é uma Rede Social que catalisa e amplia o relacionamento entre consumidor e marca,permitindo que cada vez mais o

  usuário e consumidor SEJA e FAÇA a MARCA!!

Conteúdo: Co-criação e COLABORAÇÃO

- ✓ Manutenção participativa de um **hub de CONTEÚDO**, uma coleção de:
  - ✓ Vídeos, fotos, artigos, blogs, outras redes sociais
  - ✓ Fóruns, entrevistas com especialistas e representantes da MARCA, “hotsites”
  - ✓ Lançamentos, propagandas e campanhas, exclusividades
  - ✓ Eventos culturais e esportivos patrocinados pela MARCA
  - ✓ Notícias genéricas, notícias oficiais de interesse público geradas pela própria MARCA (news release, media release, press release)
  - ✓ **Tudo sobre a Marca**

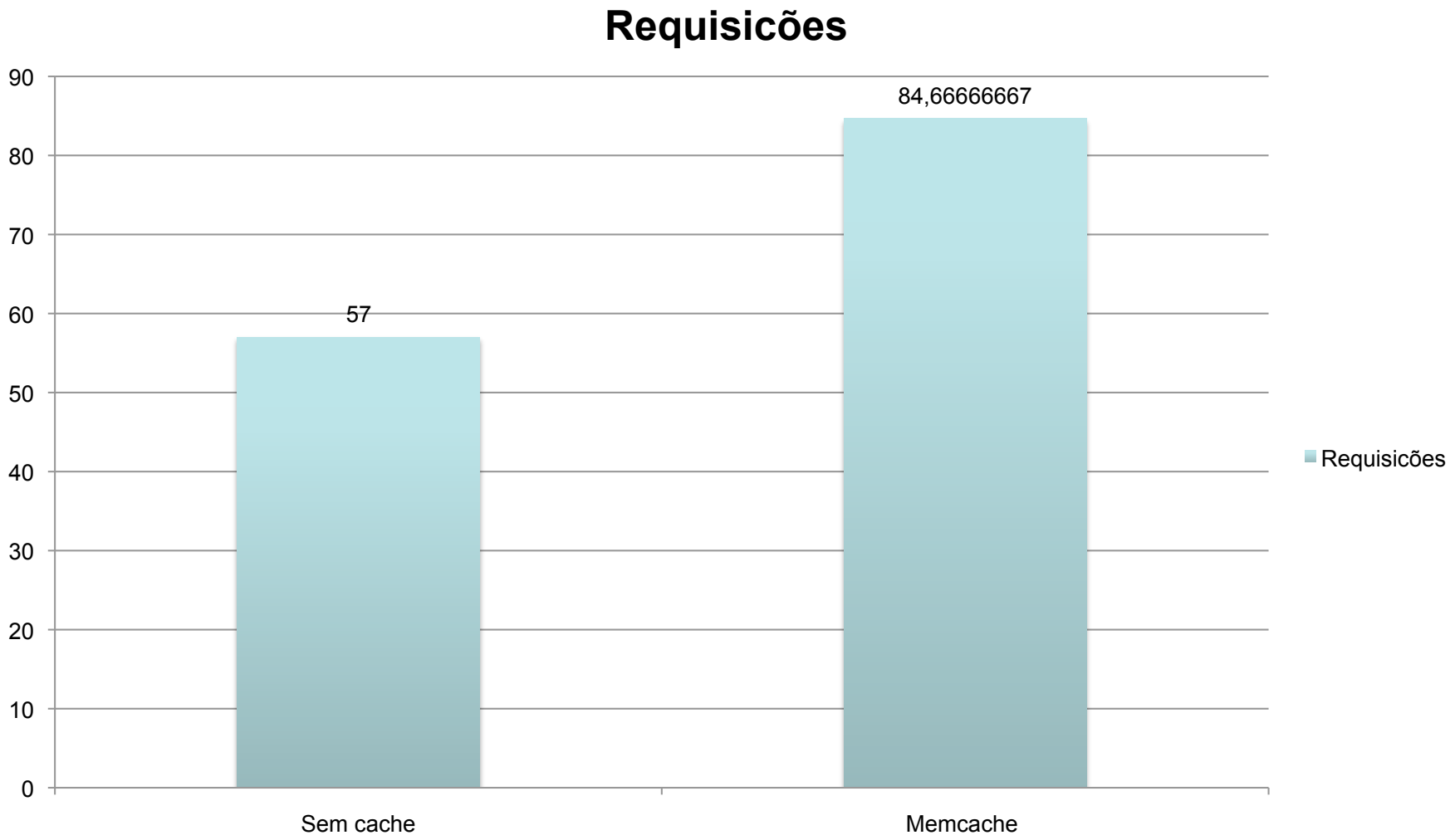
Quem são MARCAS na Drimio ?

- ✓ Extrapolamos o conceito limitante de MARCAS de produtos e anunciantes tradicionais
- ✓ Na Drimio as 5.564 cidades brasileiras são MARCAS
  - ✓ Procuram ampliar sua visibilidade no mercado para tornarem-se destino do turismo de lazer e de negócios
- ✓ **Universidades**, Associações, Festas regionais, Times de Futebol, Escolas de Samba, Praias, Concursos Nacionais, Casas noturnas, Feiras e Eventos, e inclusive Celebridades são MARCAS
  - ✓ **Todas estas entidades** estão sujeitas aos **mesmos princípios que governam a existência e desenvolvimento de qualquer MARCA no mercado**

Case - Drimio

- ✓ Usuários podem compartilhar vídeos, links, imagens, notícias, enviar mensagens e participar de fórum sobre suas marcas de maior afinidade.
- ✓ Integração com Youtube, Twitter, Google Search, Yahoo Search, Flickr, leitura e geração de RSS/Atom.
- ✓ Versão mobile com suporte a iPhone, Nokia N95, Blackberry e outros dispositivos
- ✓ Crescimento de 300% no número de usuários nas primeiras 24 horas depois do lançamento
- ✓ Triplicou o número de pageviews nas primeiras 24 horas depois do lançamento

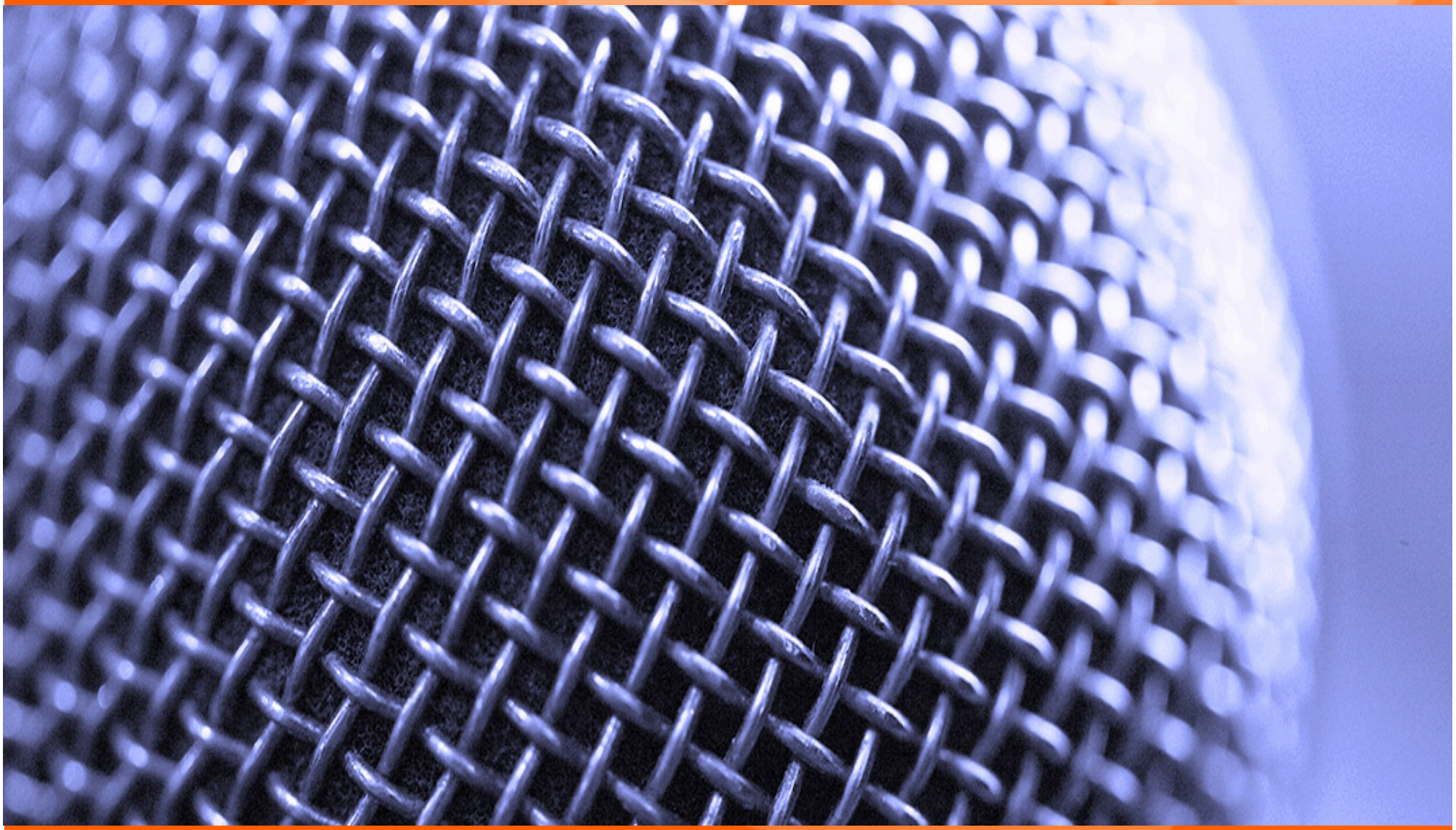
Exemplo


Requisicões
90
80
70
60
50
40
30
20
10
0
57
84,66666667
Sem cache
Memcache
Requisicões

Referências

- ✓ http://www.slideshare.net/FordAntiTrust/php-performance-with-apc-memcached
- ✓ http://www.slideshare.net/TheMarco/memcached-ftw
- ✓ http://www.slideshare.net/benramsey/give-your-site-a-boost-with-memcache
- ✓ http://www.slideshare.net/cb1kenobi/memcached-and-mysql
- ✓ http://www.danga.com/memcached/
- ✓ http://www.danga.com/memcached/users.bml

Perguntas?



drimio

Contato

Osdeni José Sadzinski
http://www.osdeni.net/
osdeni.sadzinski@drimio.com

Elton Luís Minetto
http://www.eltonminetto.net
elton.minetto@drimio.com
http://www.twitter.com/eminetto


drimio
drimio.com