



# Otimizando aplicações com Zend Framework

elton luís minetto  
@eminetto

# Quem sou Eu?

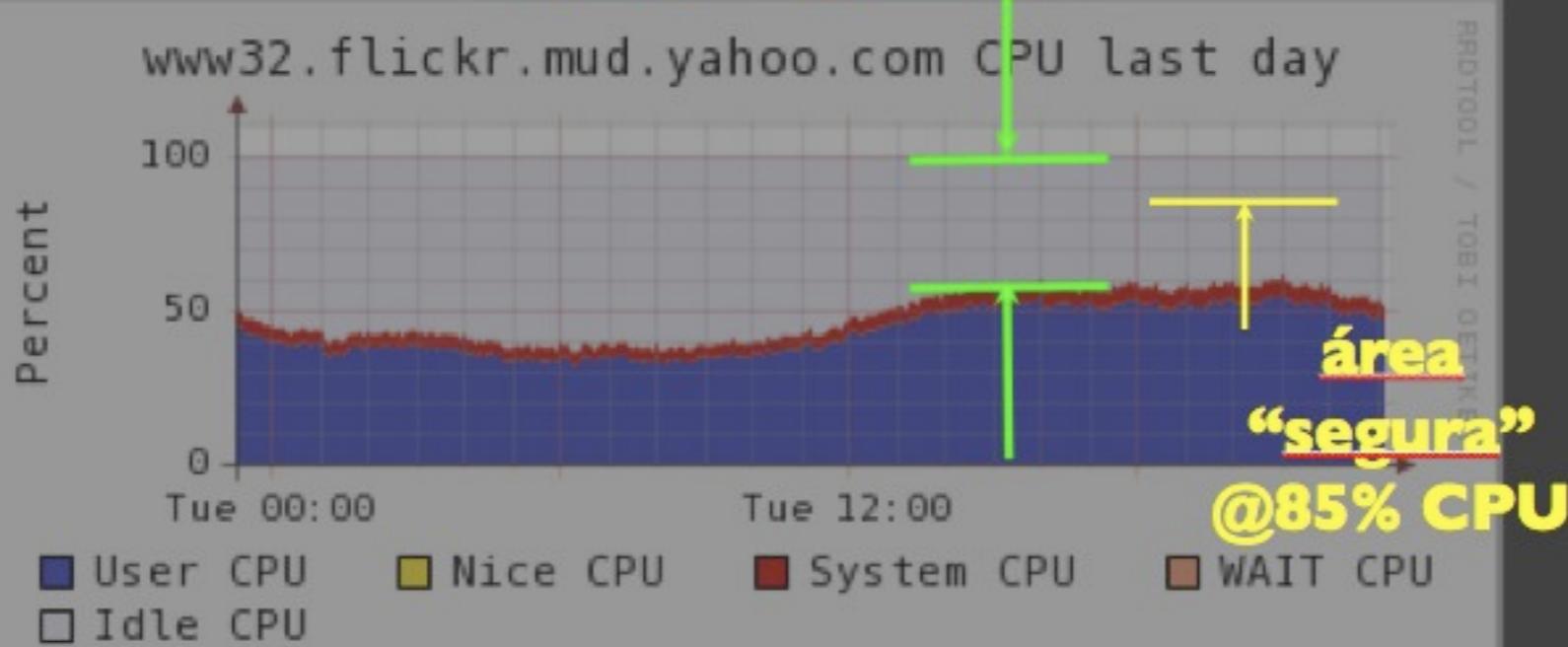
- Graduado e pós-graduado em Ciência da Computação
- Trabalha com PHP/MySQL desde 2000. Trabalha com Linux desde 1997. Professor desde 2004.
- Autor do livro Frameworks para Desenvolvimento em PHP - Editora Novatec, co-autor do livro Grid Computing in Research and Education - IBM Redbooks e autor do e-book Zend Framework na prática
- Membro do PHPSC e sócio da Coderockr



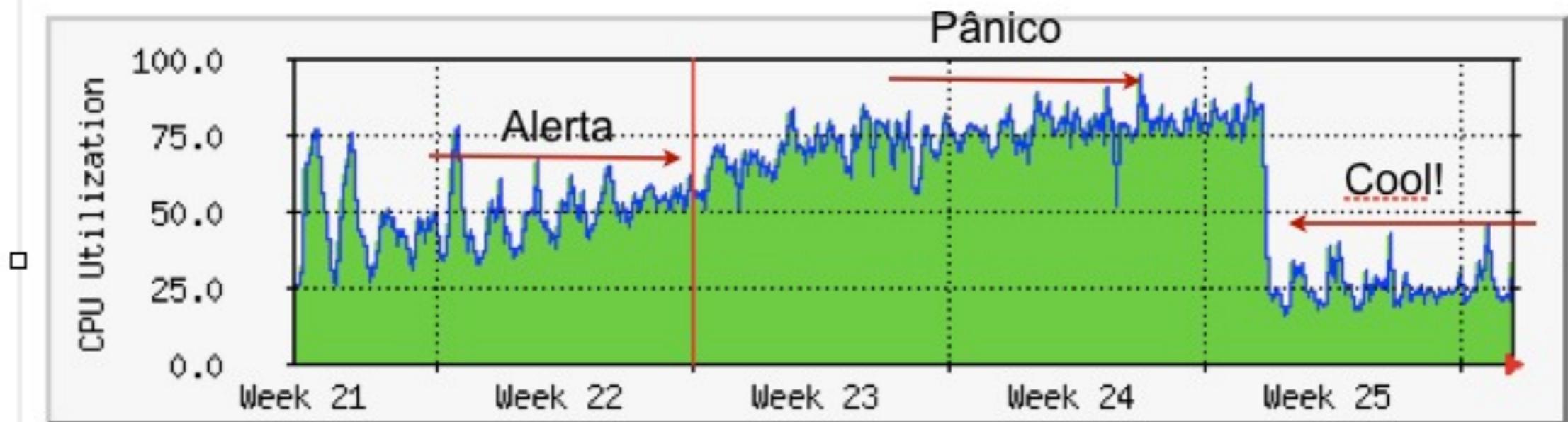
# Métricas



## O que você ainda tem



## 'Monthly' Graph (2 Hour Average)



Javascript/CSS

Imagenes

PHP

MySQL

Apache



# Conceitos

- Avaliação de desempenho
  - “Quão bem isto executa?”
  - Determinar a capacidade de um sistema
- Análise de desempenho
  - “Por que isso executa desta maneira?”
  - Determinar onde a aplicação gasta mais tempo e recurso

# Conceitos

- Avaliar a aplicação inteira (full-stack)
  - ab; siege; http\_load; jMeter
- ou somente um componente(single-component)
  - MySQL Benchmark Suite/innotop; (MySQL) ; Xdebug (PHP); ySlow(html+css+js)

# Conceitos

- O que medir
- **Transações por unidade de tempo**
- Tempo de resposta ou latência
- Escalonamento
- Concorrência

# A máquina

- Intel(R) Core(TM)2 Duo CPU E6550 @ 2.33GHz, 2GB RAM
- Ubuntu Server; Apache 2.2.16; MySQL 5.1.49; PHP 5.3.3; Zend Framework 1.11.9

# Ferramentas

- siege
  - siege -c 100 -r 10 -d 1 http://URL
- Observações:
  - Executado 3 vezes e gerada a média
  - A cada execução são reiniciados os serviços (apache, mysql, memcached, nginx, gearman)

# Criando o projeto

```
zf create project blog
```

- Criado um virtual host no apache

```
<VirtualHost *:80>
    ServerName blog.local
    DocumentRoot /var/www/blog/public

    SetEnv APPLICATION_ENV "development"

    <Directory /var/www/blog/public>
        DirectoryIndex index.php
        AllowOverride All
        Order allow,deny
        Allow from all
    </Directory>
</VirtualHost>
```

- Adicionar no /etc/hosts.

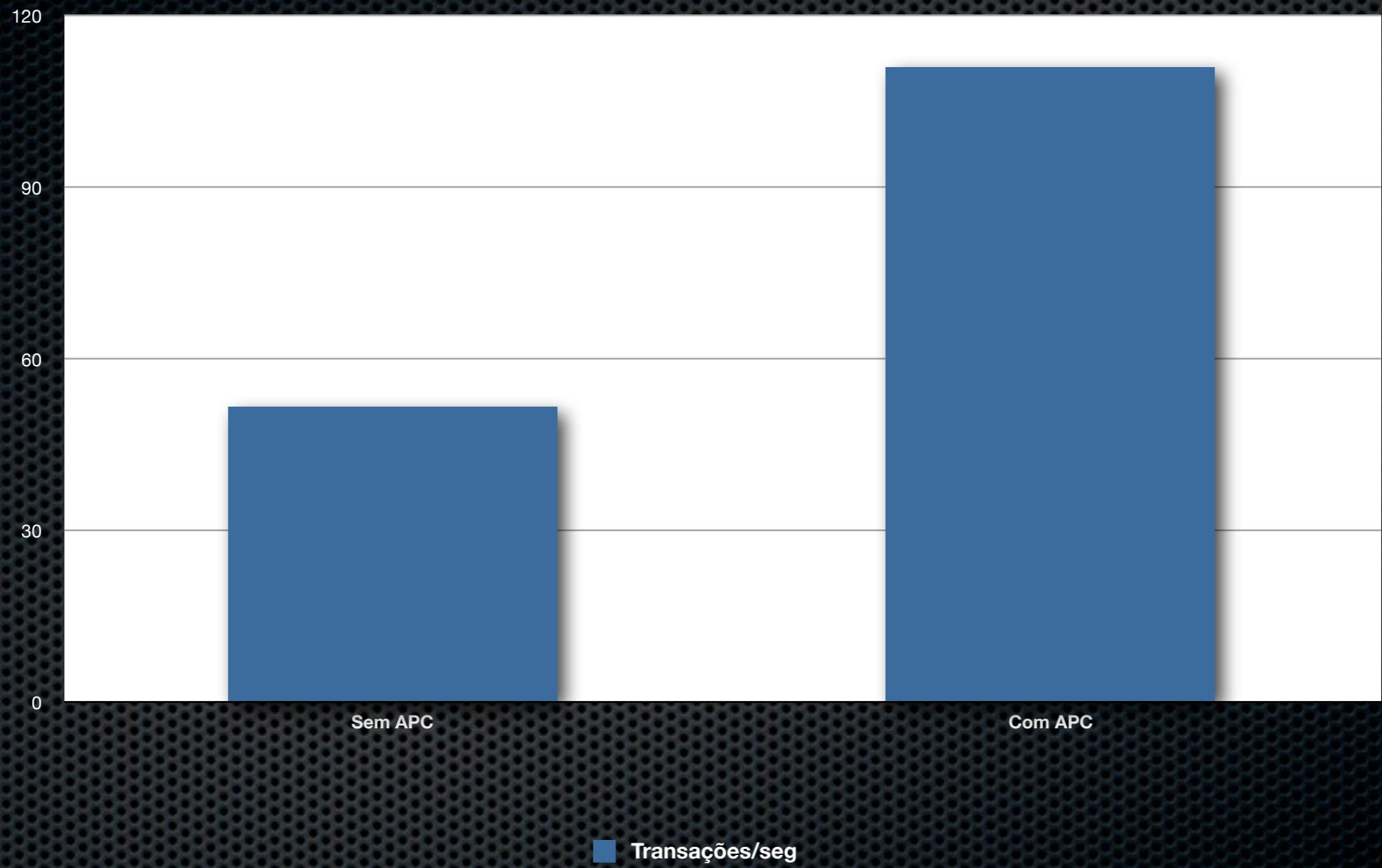
```
127.0.0.1 blog.local
```

# O primeiro teste (sem APC)

- Transações por segundo: 51,55

# APC

- Transações por segundo: 110,94



# Banco de dados

```
mysql -uroot -proot
create database performance;
use performance;
create table post (id int primary key auto_increment, title varchar(100), text text);
exit;
for ((i=1;i<100;i++)) ; do
    mysql -uroot -proot performance -e "insert into post values (null, 'Titulo $i', 'Texto $i')"
done
```

# Banco de dados

```
zf configure dbadapter "adapter=Pdo_Mysql&host=localhost&username=root&password=root&dbname=performance"
zf create db-table Post post
zf create controller Post
```

# Banco de dados

```
<?php
class PostController extends Zend_Controller_Action
{
    public function indexAction() {
        $post = new Application_Model_DbTable_Post();
        $this->view->entries = $post->fetchAll();
    }
}
```

# Banco de dados

application/views/scripts/post/index.phtml

```
Posts: <br />
<dl>
    <?php foreach ($this->entries as $entry): ?>
        <dt><?php echo $this->escape($entry->title) ?></dt>
        <dd><?php echo $this->escape($entry->text) ?></dd>
    <?php endforeach ?>
</dl>
```

# Banco de dados

- Transações por segundo: 85,25

# Banco de dados usando cache em disco

```
<?php
class PostController extends Zend_Controller_Action {
    public function indexAction() {
        $frontendOptions = array(
            'lifetime' => 7200,
            'automatic_serialization' => true
        );
        $backendOptions = array('cache_dir' => '/tmp/');
        $cache = Zend_Cache::factory('Core', 'File', $frontendOptions, $backendOptions);
        if( ($result = $cache->load('posts')) === false ) {
            $post = new Application_Model_DbTable_Post();
            $result = $post->fetchAll();
            $cache->save($result, 'posts');
        }
        $this->view->entries = $result;
    }
}
```

# Banco de dados usando cache em disco

- Transações por segundo: 100,70

# Banco de dados usando cache em MEMCACHE

```
<?php
class PostController extends Zend_Controller_Action {
    public function indexAction() {
        $frontendOptions = array('lifetime' => 7200, 'automatic_serialization' => true);
        $backendOptions = array('servers' => array(
            array('host' => 'localhost', 'port' => 11211, 'persistent' => true)
        ));
        $cache = Zend_Cache::factory('Core', 'Memcached', $frontendOptions, $backendOptions);
        if( ($result = $cache->load('posts')) === false ) {
            $post = new Application_Model_DbTable_Post();
            $result = $post->fetchAll();
            $cache->save($result, 'posts');
        }
        $this->view->entries = $result;
    }
}
```

# Banco de dados usando cache em Memcache

- Transações por segundo: 102,85

# Cache do html

copiar views/scripts/post/index.phtml para views/scripts/post/\_index.phtml

Alterar views/scripts/post/index.phtml para:

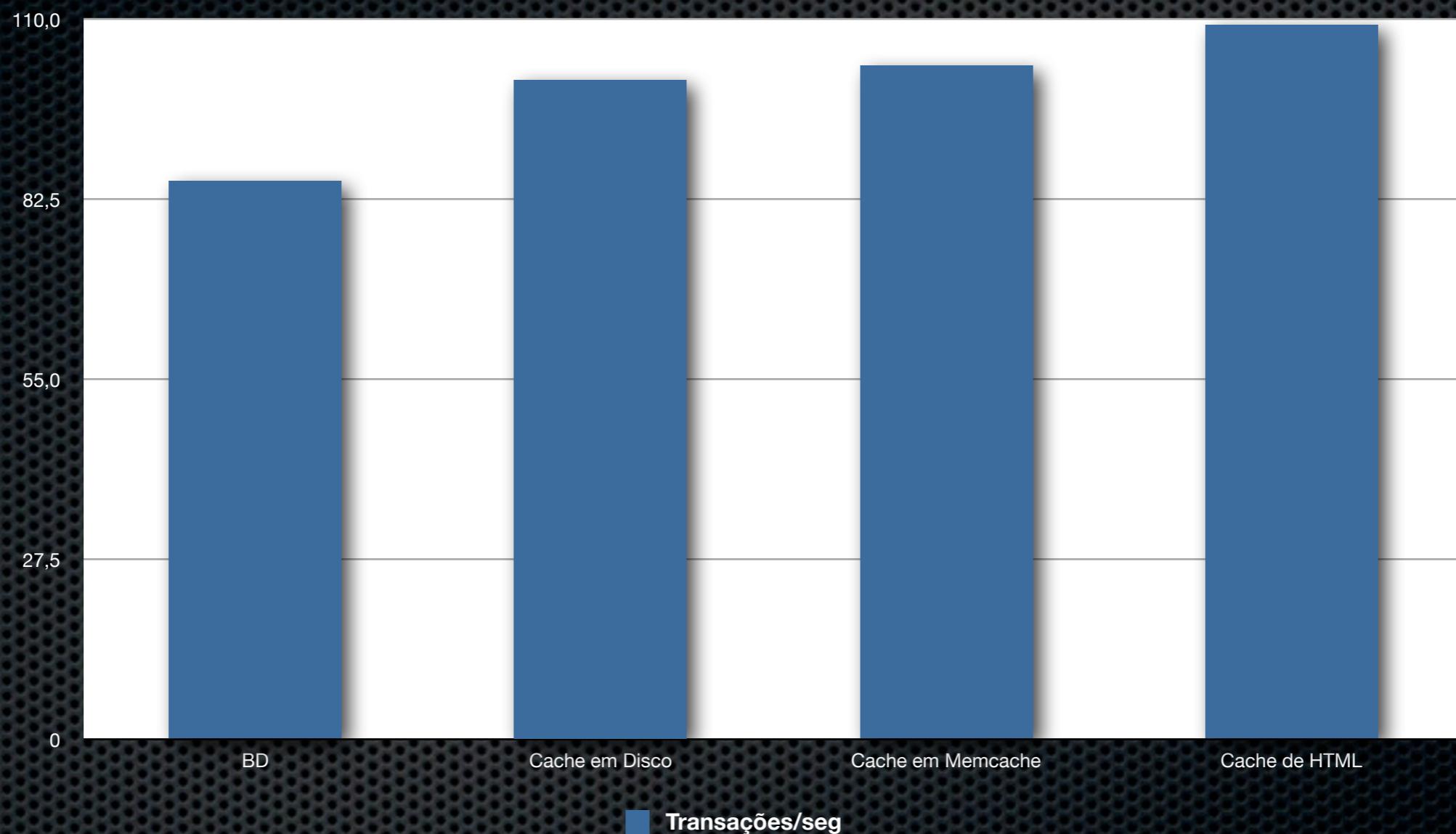
```
<?php echo $this->page; ?>
```

# Cache do html

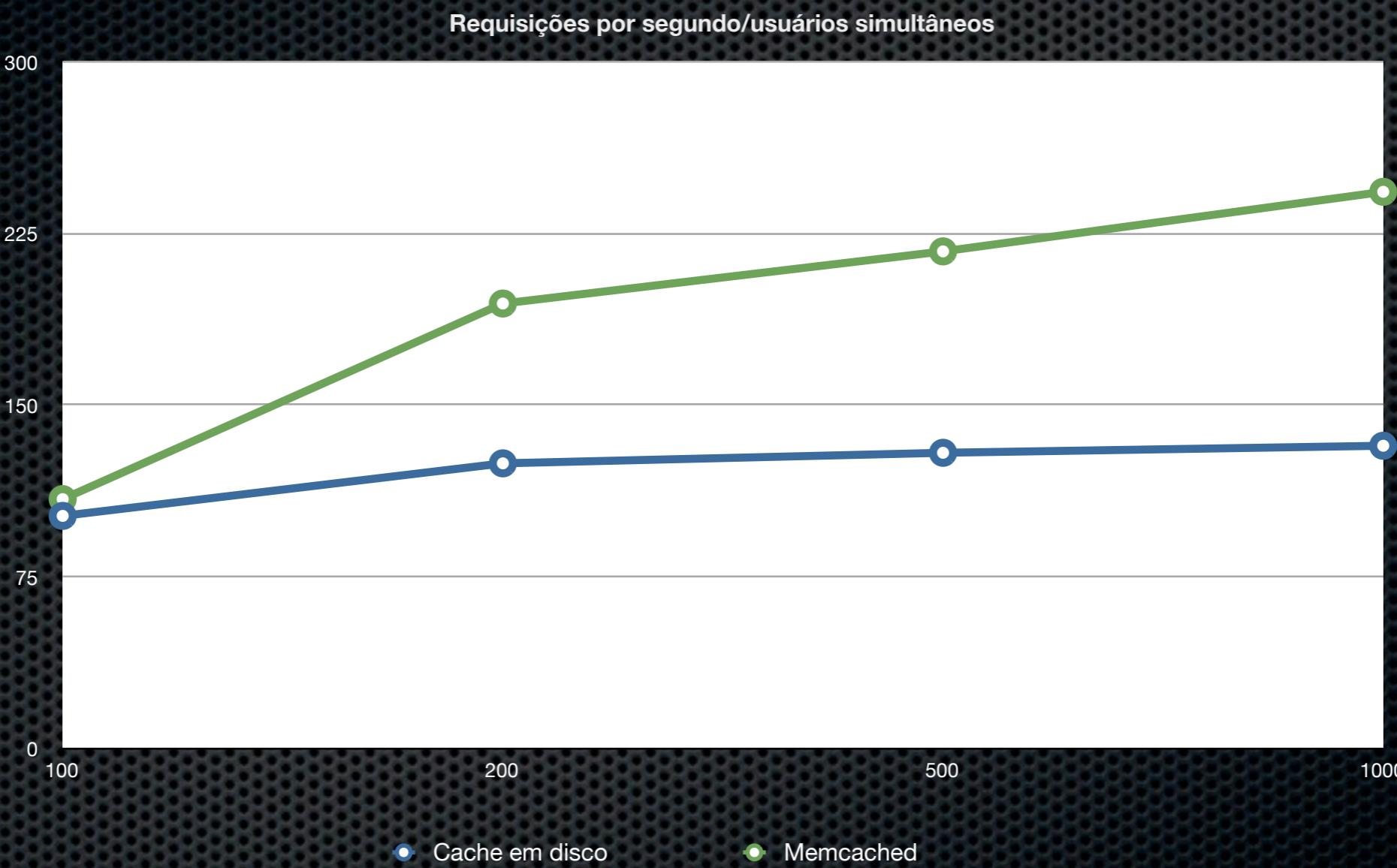
```
<?php
class PostController extends Zend_Controller_Action {
    public function indexAction() {
        $frontendOptions = array('lifetime' => 7200, 'automatic_serialization' => true);
        $backendOptions = array('servers' => array(
            array('host' => 'localhost', 'port' => 11211, 'persistent' => true)));
        $cache = Zend_Cache::factory('Core', 'Memcached', $frontendOptions,
$backendOptions);
        if( ($page = $cache->load('posts_page')) === false ) {
            $post = new Application_Model_DbTable_Post();
            $this->view->entries = $post->fetchAll();
            $page = $this->view->render('post/_index.phtml');
            $cache->save($page, 'posts_page');
        }
        $this->view->page = $page;
    }
}
```

# Cache do HTML

- Transações por segundo: 109,11



# Escalando



# Inclusão em banco de dados

```
mysql -uroot -proot performance
create table access (id int primary key auto_increment, page varchar(100), time timestamp
default current_timestamp);
exit;
zf create db-table Access access
```

# Inclusão em banco de dados

```
//novo método no PostController
public function insertAction() {
    $access = new Application_Model_DbTable_Access();
    $data = array('page'=>'/insert');
    $access->insert($data);
    echo 'Inserido';
    exit;
}
```

# Inclusão em banco de dados

- Transações por segundo: 108,45

# Inclusão em banco de dados

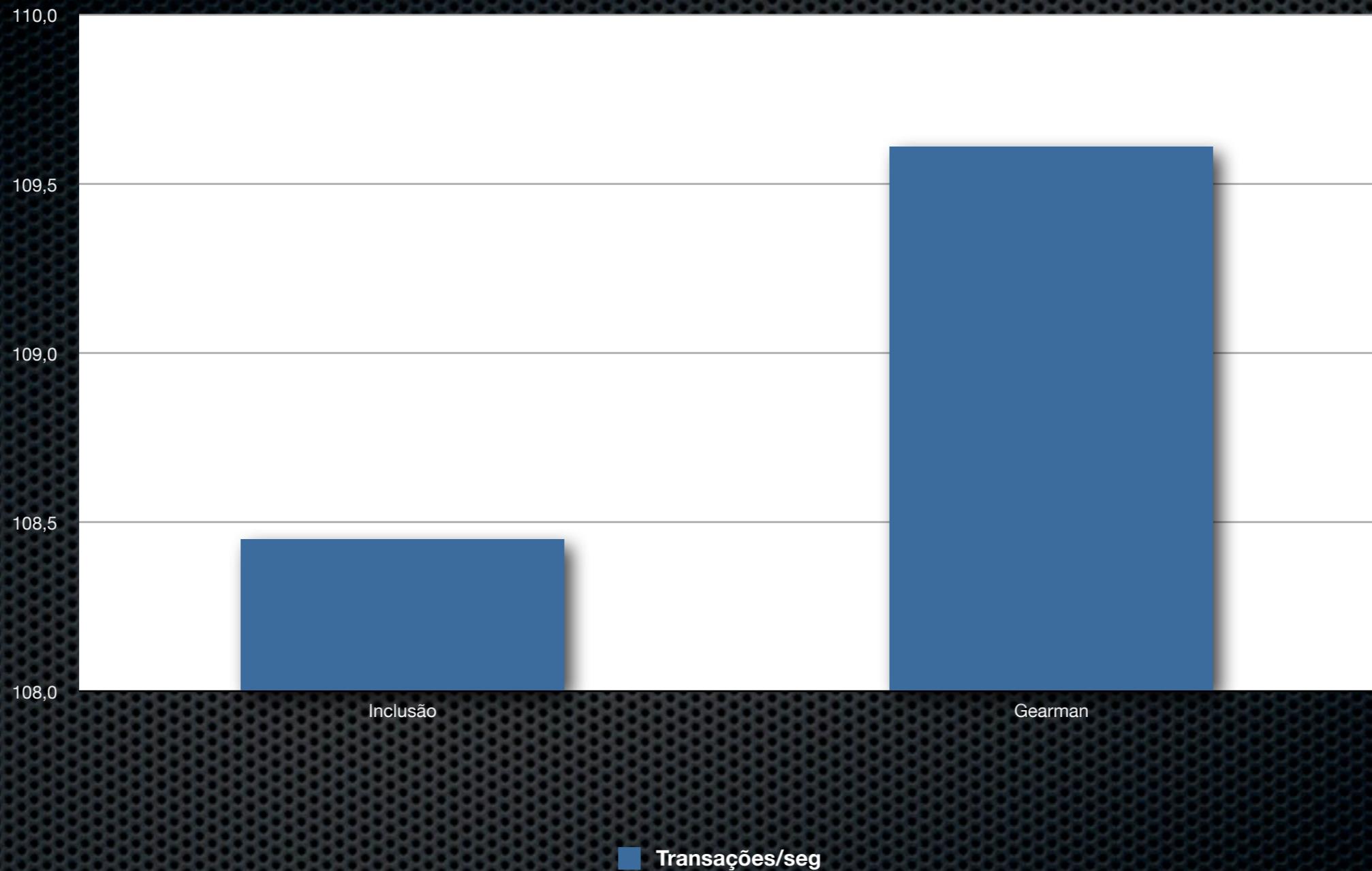
## - Gearman

```
//novo método no PostController
public function insertAction() {
    $client= new GearmanClient();
    $client->addServer();
    $data = array('page'=>'/insert');
    $client->doBackground("access_queue", serialize($data));
    echo 'Inserido';
    exit;
}
```

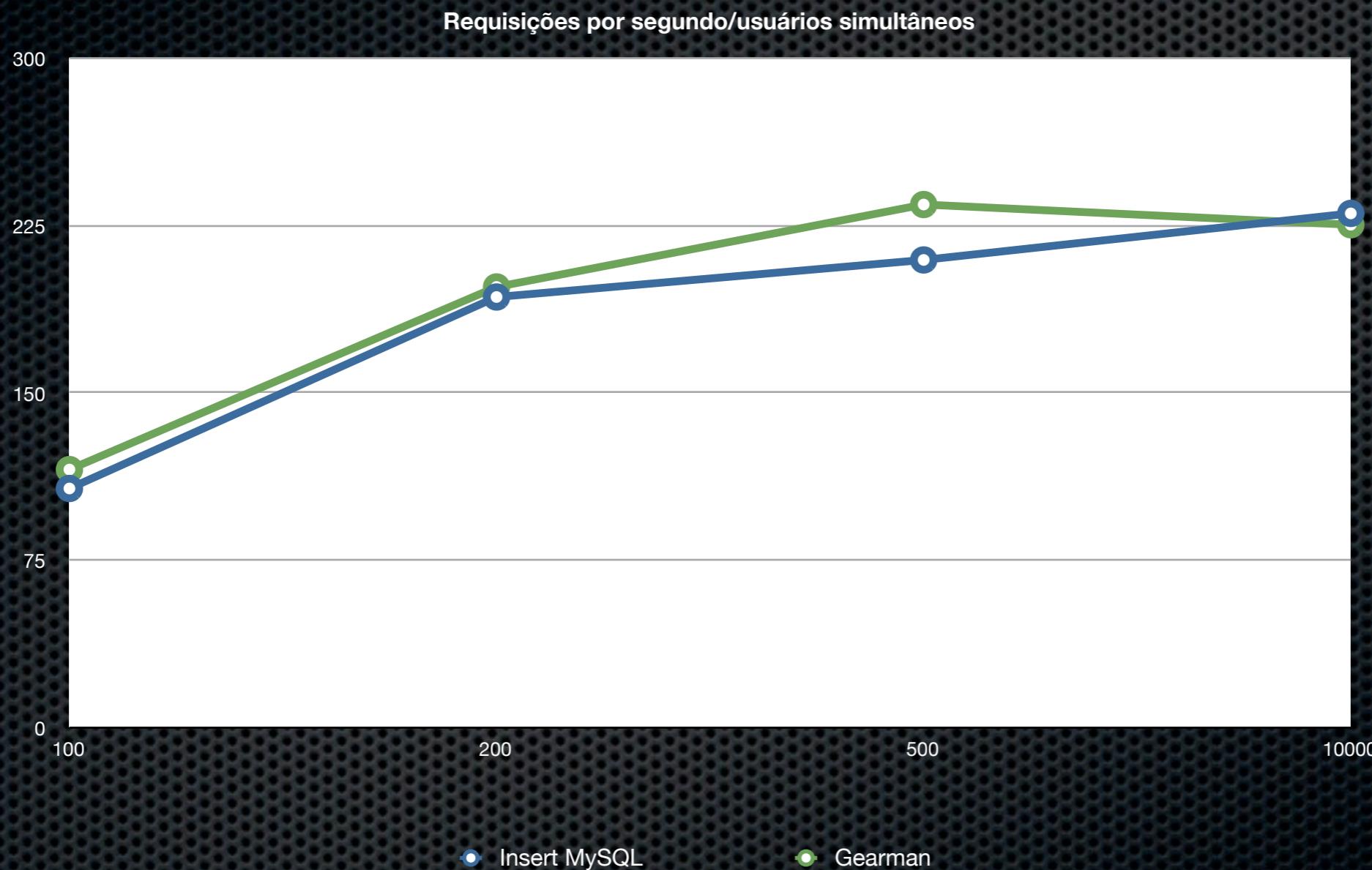
# Inclusão em banco de dados

## - gearman

- Transações por segundo: 109,61



# Escalando



# Referências

- <http://framework.zend.com>
- <http://talks.php.net/show/oscon06>
- <http://talks.php.net/show/digg/>
- <http://vimeo.com/20387525>
- <http://vimeo.com/20565041>
- <http://www.slideshare.net/eminetto/ao-infinito-e-alm-com-php-memcached-e-gearman>
- <http://www.slideshare.net/eminetto/desenvolvendo-aplicaes-web-escalveis-com-php>

# Contato

```
<?php
$card = array(
    'nome' => 'Elton Luís Minetto',
    'site' => 'http://www.eltonminetto.net',
    'e-mail' => 'eminetto@coderockr.com',
    'twitter' => '@eminetto',
    'all' => 'http://eminetto.me'
);
var_dump($card);
```