

Mysql para aplicações Web escaláveis



Elton Luís Minetto



segunda-feira, 19 de abril de 2010

Quem sou eu?

- ✓ Graduado e pós-graduado em Ciência da Computação. Cursando MBA em Gerenciamento de Projetos
- ✓ Trabalha com PHP/MySQL desde 2000
- ✓ Autor do livro Frameworks para Desenvolvimento em PHP - Editora Novatec e co-autor do livro Grid Computing in Research and Education - IBM Redbooks
- ✓ Membro do PHPSC
- ✓ Diretor de Desenvolvimento do Drimio e professor na Unochapecó(Chapecó/SC)

E vocês?

- ✓ Quem são vocês?
 - ✓ Desenvolvedores
 - ✓ DBAs
 - ✓ Administrador de sistemas
- ✓ A quanto tempo usa MySQL? E outros bancos de dados?

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...



segunda-feira, 19 de abril de 2010

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



- ✓ Segundo a Wikipédia:
 - ✓ O MySQL é um sistema de gerenciamento de banco de dados (SGBD), que utiliza a linguagem SQL (Linguagem de Consulta Estruturada, do inglês Structured Query Language) como interface. É atualmente um dos bancos de dados mais populares, com mais de 10 milhões de instalações pelo mundo.
 - ✓ No dia 16 de Janeiro de 2008, a MySQL AB, desenvolvedora do MySQL foi adquirida pela Sun Microsystems, por US\$ 1 bilhão, um preço jamais visto no setor de licenças livres. No dia 20 de Abril de 2009 a Oracle compra a Sun Microsystems e todos o seu produtos, incluindo o MySQL

Porque usar MySQL?

- ✓ Portabilidade
- ✓ Compatibilidade com diversas linguagens de programação
- ✓ Excelente desempenho e estabilidade;
- ✓ Pouco exigente quanto a recursos de hardware;
- ✓ É um Software Livre com base na GPL;
- ✓ Contempla a utilização de vários Storage Engines como MyISAM, InnoDB, Falcon, BDB, Archive, Federated, CSV, Solid...
- ✓ Suporta controle transacional, Triggers, Stored Procedures e Functions;
- ✓ Replicação facilmente configurável;

Quem usa MySQL?

- ✓ Facebook
- ✓ Twitter
- ✓ Feedburner
- ✓ LinkedIn
- ✓ Digg
- ✓ Friendster
- ✓ Flickr
- ✓ 37Signals apps (Basecamp, Campfire)

Métricas



segunda-feira, 19 de abril de 2010

Métricas



segunda-feira, 19 de abril de 2010

Métricas

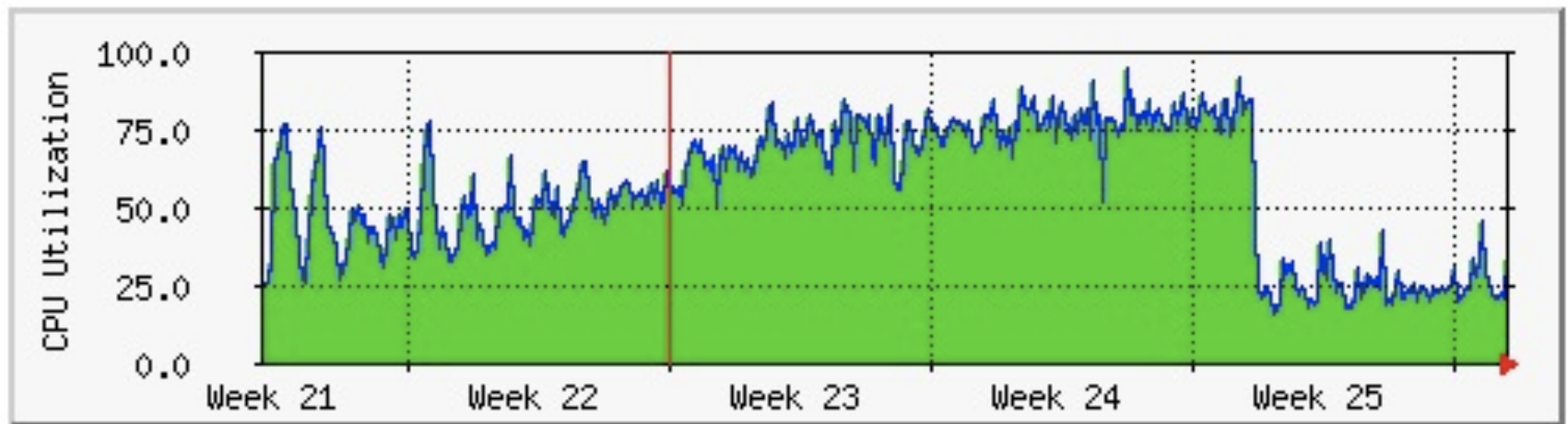


segunda-feira, 19 de abril de 2010

Métricas - Drimio

✓ Gráfico de CPU de um dos servidores

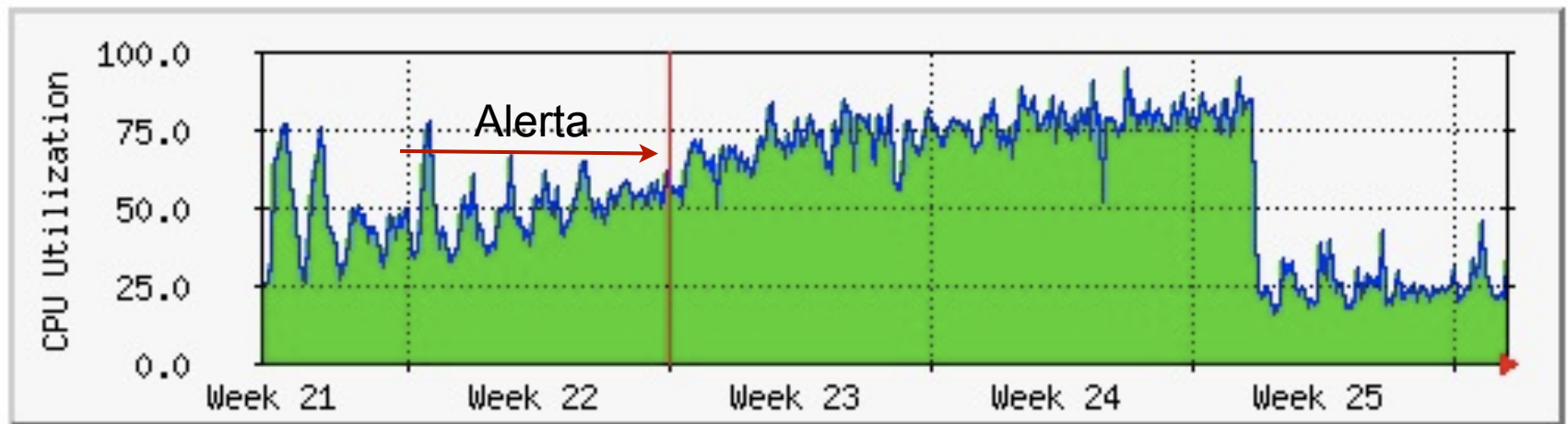
`Monthly' Graph (2 Hour Average)



Métricas - Drimio

✓ Gráfico de CPU de um dos servidores

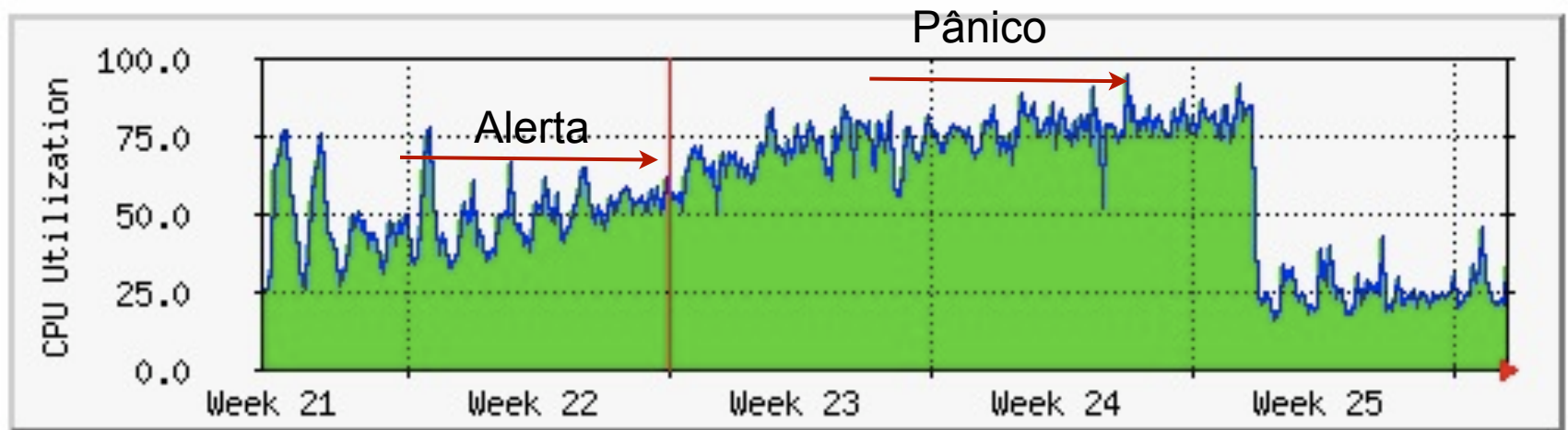
'Monthly' Graph (2 Hour Average)



Métricas - Drimio

✓ Gráfico de CPU de um dos servidores

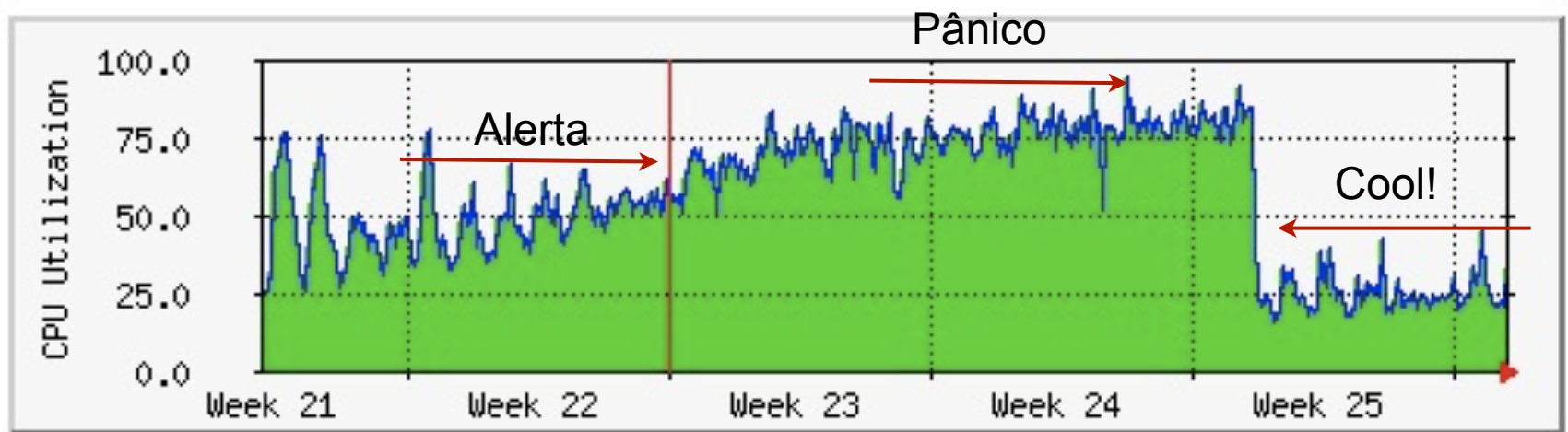
'Monthly' Graph (2 Hour Average)



Métricas - Drimio

✓ Gráfico de CPU de um dos servidores

'Monthly' Graph (2 Hour Average)



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
- ✓ Avaliar a aplicação inteira (full-stack) ou somente o MySQL (single-component)
- ✓ O que medir
 - ✓ Transações por unidade de tempo
 - ✓ Tempo de resposta ou latência
 - ✓ Escalonamento
 - ✓ Concorrência

Ferramentas

- ✓ Full-stack
 - ✓ ab
 - ✓ siege
 - ✓ http_load
 - ✓ jMeter
- ✓ Single-component
 - ✓ mysqlslap
 - ✓ sysbench
 - ✓ Database Test Suite
 - ✓ MySQL Benchmark Suite (sql-bench)
 - ✓ innotop, mytop

segunda-feira, 19 de abril de 2010

siege -c50 "<http://www.drimio.com/usuario/eminetto>" -b -r10
No ab verificar "Requests per second". No siege "Transaction rate"

./run-all-tests --user=root -password=senha --log

Nativas do MySQL - show status

- ✓ **Open_tables, Opened_tables:** Número de tabelas abertas atualmente e o número de tabelas abertas desde que foi iniciado. Analisa se está certo o cache de tabelas. Valores altos significam que o cache deveria ser maior
- ✓ **Slow_queries:** Número de queries demorando mais do que um valor pré-determinado de tempo (my.cnf).
- ✓ **Select_scan:** Número de queries que estão usando um full scan para encontrar os dados. Valores altos significam que as queries devem ser otimizadas.
- ✓ **Select_full_join:** Número de joins que estão sendo executadas sem o uso de índices. Valor alto significa que índices devem ser criados ou consultas otimizadas.
- ✓ **Qcache_hits:** Número de acessos ao cache de queries do MySQL. Um valor alto significa que o MySQL está usando o cache de maneira eficiente, sem precisar reconstruir a query em toda execução

Ferramentas -mysqladmin

✓ Conexões

- ✓ `mysqladmin extended -i10 | grep Threads_running`

✓ Total de queries

- ✓ `mysqladmin extended -i10 | grep Questions`



segunda-feira, 19 de abril de 2010

Melhorando a performance

- ✓ É possível melhorar a performance do MySQL de três maneiras:
 - ✓ Tuning do arquivo de configuração (my.cnf)
 - ✓ Otimizando as consultas SQL
 - ✓ Melhorando hardware/arquitetura

- ✓ Você também pode dar uma folga pro coitado!

- ✓ O arquivo original da instalação do MySQL possui uma configuração com recursos reduzidos, para ser rodado na grande maioria dos servidores
- ✓ Usar o arquivo apropriado:
 - ✓ my-huge.cnf (enorme capacidade) - Servidores com mais de 1 Gb de memória RAM dedicada ao MySQL
 - ✓ my-large.cnf (grande capacidade) - Servidores com 512 megas a 1 Gb de memória RAM dedicadas ao MySQL
 - ✓ my-medium.cnf (média capacidade) - Servidores com 128 a 256 megas de memória RAM dedicadas ao MySQL
 - ✓ my-small.cnf (pequena capacidade) - Servidores com 64 a 128 megas de memória RAM dedicadas ao MySQL

MySQL Performance Tuning Primer Script

- ✓ Script que auxilia a identificar mudanças no arquivo de configuração do MySQL
- ✓ Fazer o download do script
 - ✓ `wget http://day32.com/MySQL/tuning-primer.sh`
- ✓ Tornar o script executável
 - ✓ `chmod +x ./tuning-primer.sh`
- ✓ Executar o script
 - ✓ `./tuning-primer.sh`
- ✓ É apresentado um relatório dividido em várias seções com sugestões de modificações nos parâmetros
 - ✓ <http://dev.mysql.com/doc/refman/5.0/en/server-system-variables.html>

segunda-feira, 19 de abril de 2010

tunning_resultado_real.txt
tunning_resultado_mac.txt

Otimização de consultas/dados

- ✓ “Fazer o tuning do MySQL durante a escolha da engine de armazenamento das tabelas. Usar InnoDB quando precisa de transações e MyISAM quando não precisa” (Digg)
- ✓ “Desnormalização ou cacheamento são as únicas formas de gerar uma tag cloud em milissegundos para milhões de tags” (Flickr)
- ✓ Usar o tipo correto de dados na tabela:
 - ✓ INT x SMALLINT x TINYINT
 - ✓ CHAR x VARCHAR

Otimização de consultas/dados

✓ Explain

- ✓ table - mostra o nome da tabela à qual o resultado diz respeito (para quando são efectuados JOINS entre tabelas);
- ✓ type - tipo de join usado. Do melhor para o pior tipo temos: system, const, eq_ref, ref, range, index, all;
- ✓ possible_keys - indica quais os índices que o MySQL pode usar para encontrar resultados nesta tabela;
- ✓ key - índice usado na consulta, ou NULL caso não tenham sido usados índices;
- ✓ key_len - tamanho do índice usado, caso exista;
- ✓ ref - coluna(s) usada(s) com a key para devolver resultados;
- ✓ rows - número de registos que o MySQL tem que examinar para executar a consulta;
- ✓ extra - informação adicional acerca de como o MySQL vai executar a consulta. A evitar o aparecimento de “using filesort” e “using temporary”.

segunda-feira, 19 de abril de 2010

explain.tx
t

Hardware

- ✓ CPU
 - ✓ CPUs mais rápidas são melhores do que mais CPUs
 - ✓ 64bits com SOs de 64bits
- ✓ I/O: discos mais rápidos
- ✓ Mais memória

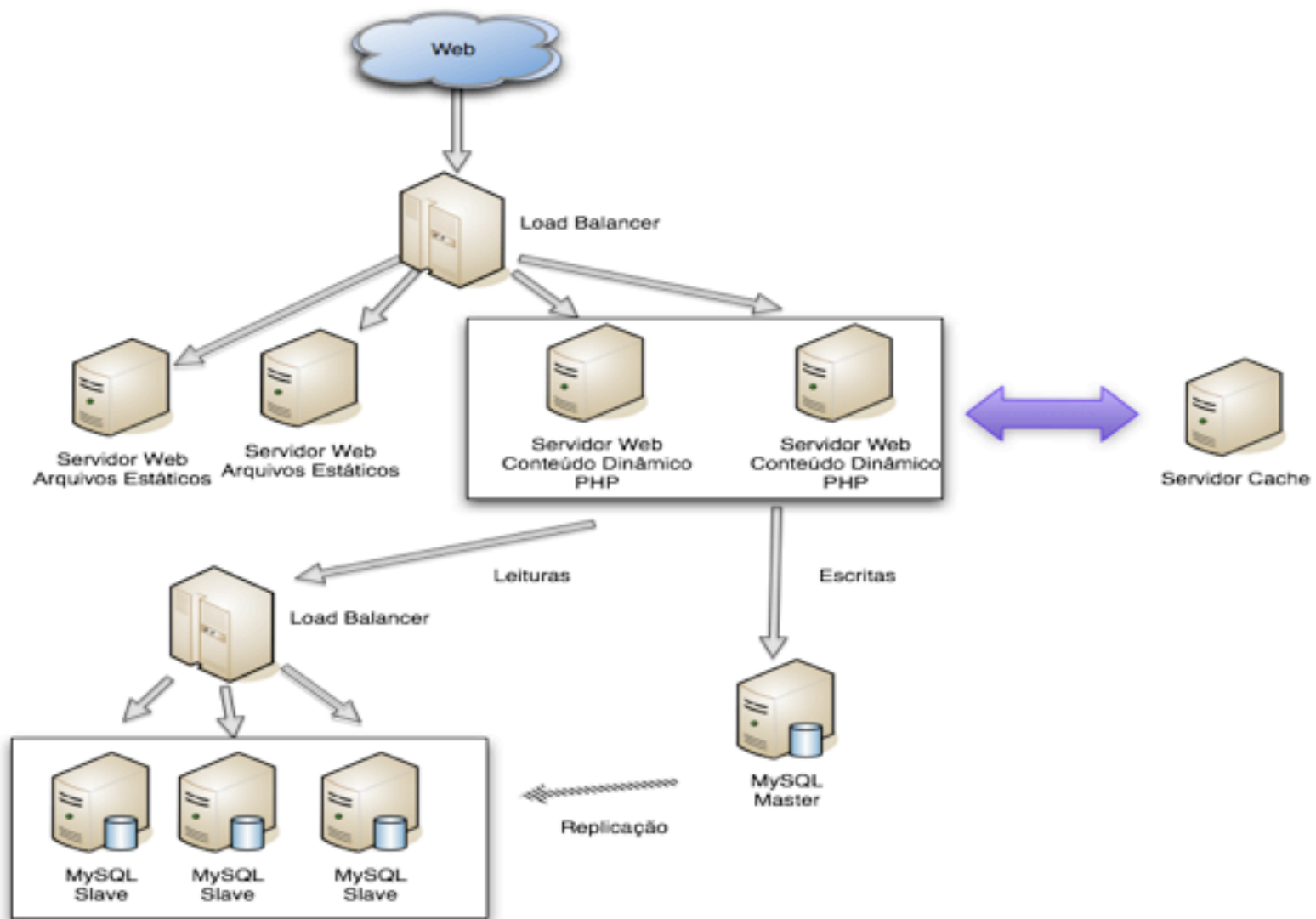
Arquitetura

- ✓ Load Balancers
- ✓ Cache servers
- ✓ Bancos de dados Master/Slave, Sharding
- ✓ Scale-Out Wins Over Scale-Up
(escalar horizontalmente adicionando mais máquinas é melhor do que verticalmente adicionando mais memória/CPU)

Arquitetura

- ✓ Sobre o Youtube:
 - ✓ “Eles seguiram uma evolução comum: servidor único, único master e múltiplos slaves para leitura e depois particionaram a base de dados.”
- ✓ Dividir a carga entre servidores. As requisições de modificação (INSERT, UPDATE,DELETE) podem ser enviadas para o Master. Os dados são replicados para os Slaves. As requisições de leitura (SELECT) são enviadas direto para os Slave

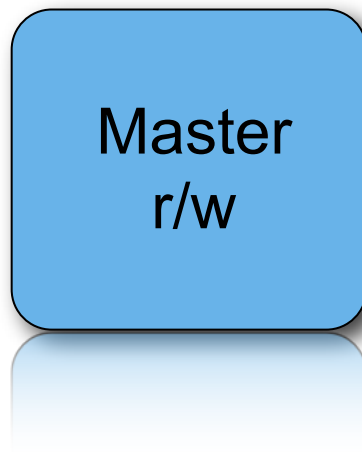
Arquitetura



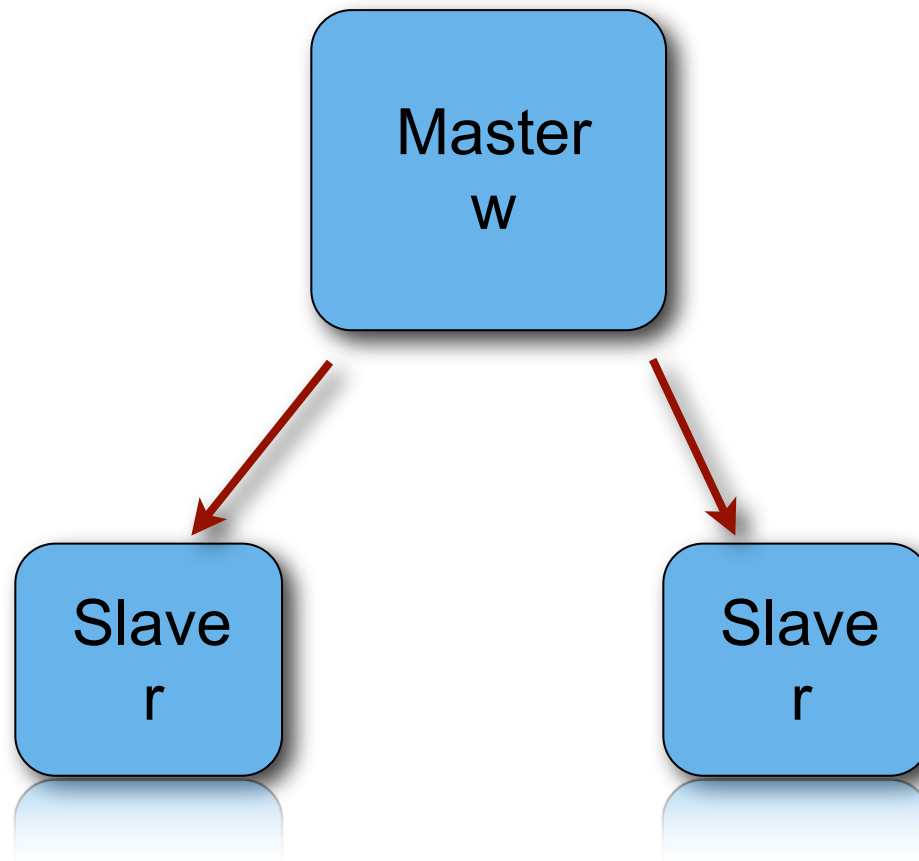
Arquitetura - Observações

- ✓ Como agora um cliente pode ser atendido por diversos servidores Web durante o uso, as sessões dos usuários devem ser salvas no banco de dados ou nos servidores de cache (Memcached)
- ✓ O MySQL Master é um SPOF(Single Point of Failure) - adicionar mais Masters em um esquema de replicação Master/Master.
- ✓ Diversos servidores de cache podem ser adicionados

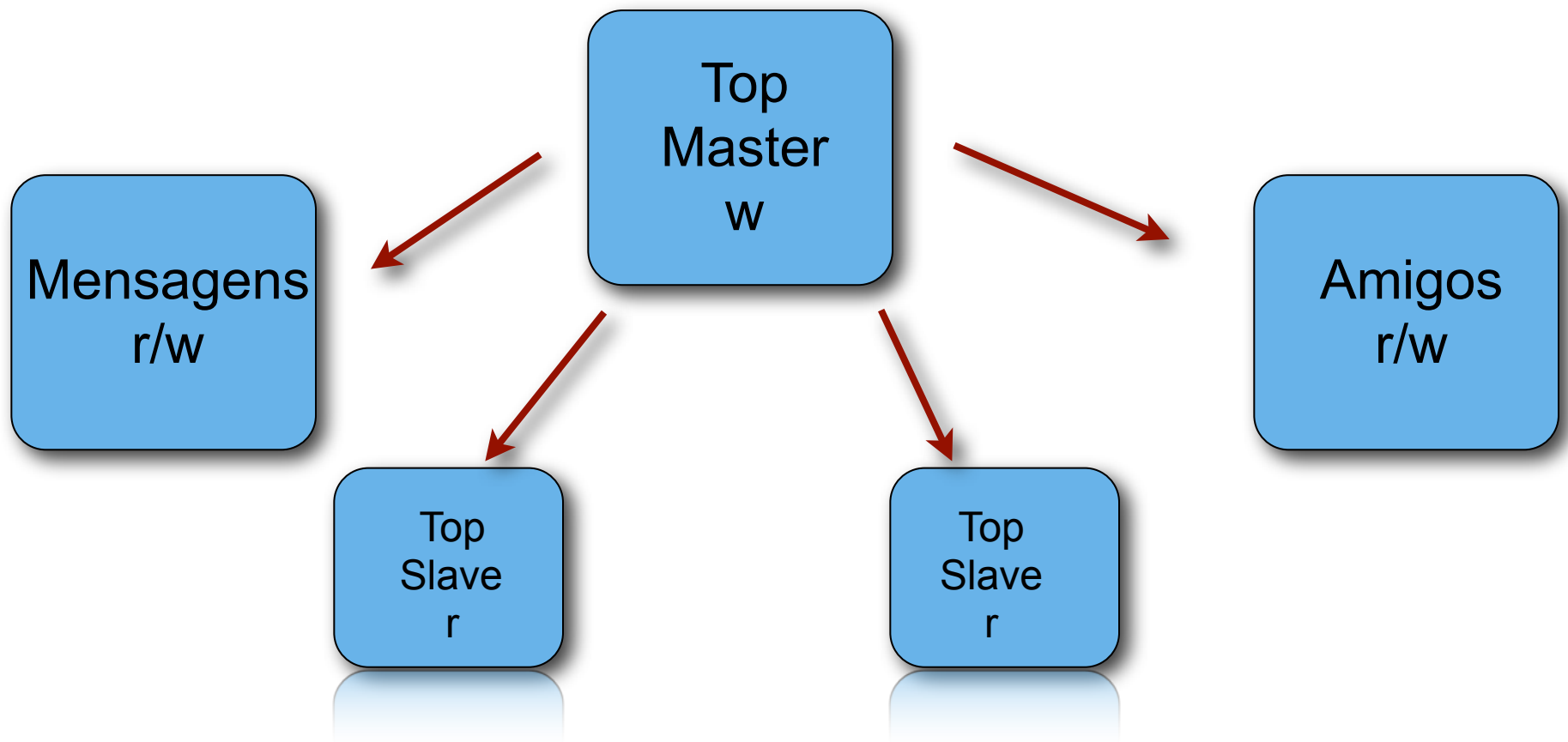
Arquitetura - Replicação



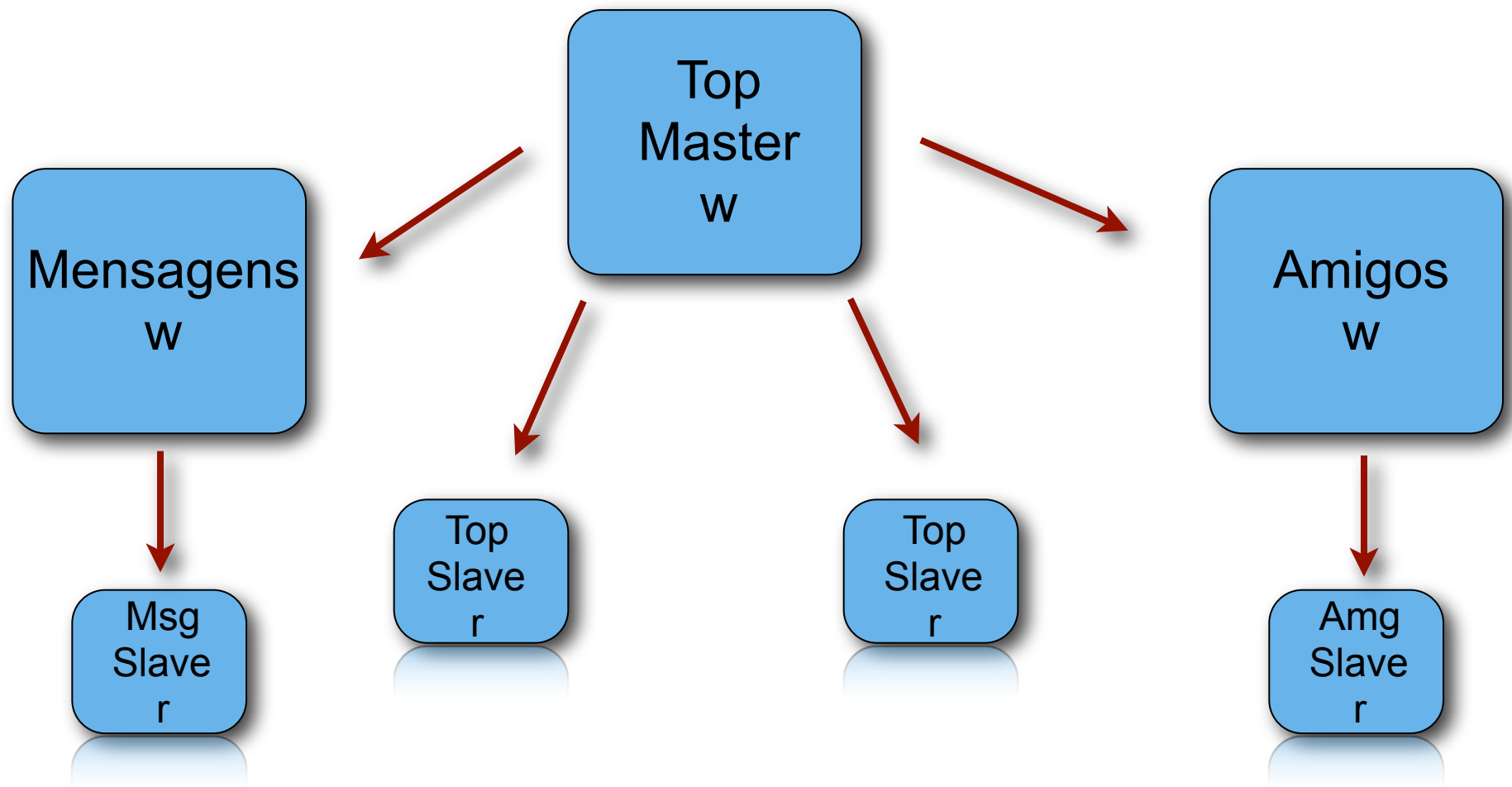
Arquitetura - Replicação



Arquitetura - Replicação



Arquitetura - Replicação



Arquitetura - Sharding

- ✓ Master-Slave tem o problema do tempo de sincronização. Resposta: sharding.
 - ✓ “Uma base de dados pode ser sharded por tabelas, dados ou faixas (ranges). É similar ao particionamento, mas possui algumas diferenças. Sharding envolve separar os dados em máquinas fisicamente distintas, enquanto que particionamento geralmente ocorre em mesmo hardware. MySQL não suporta nativamente sharding, mas sim tabelas particionadas, tabelas federadas (federated) e clusters.”
 - ✓ Ou você pode fazer uma solução “caseira”, usando uma linguagem de programação.

Arquitetura - Sharding

✓ Exemplo: um blog

- ✓ Você tem uma tabela com os posts do blog, com as colunas: id, titulo, texto, data, autor_id
- ✓ autor_id é uma chave estrangeira (foreign key) com a tabela usuarios
- ✓ Nós vamos dividir os posts do blog em duas bases de dados
- ✓ Posts cujo autor_id é par vão estar armazenados na base de dados 1
- ✓ Posts cujo autor_id é ímpar vão estar armazenados na base de dados 2
- ✓ A consulta: "mostre-me todos os posts do autor de id 26"

Arquitetura - Sharding

```
<?php
$id = 26;//$_GET['id'];
if($id % 2 == 0) {
    $con = mysql_connect('servidor1','root','root');
    mysql_select_db("sh_db1");
}
else {
    $con = mysql_connect('servidor2','root','root');
    mysql_select_db("sh_db2");
}
$res = mysql_query("select * from post where autor_id=
$id",$con);
while($conteudo = mysql_fetch_array($res, MYSQL_ASSOC))
    var_dump($conteudo);
```

Arquitetura - Sharding: Como particionar?

- ✓ Vertical Partitioning: dividir os dados em um nível de tabela/feature. Exemplo: tabela users, no DB1, tabela posts no DB2
- ✓ Range-based Partitioning: dividir os dados de acordo com faixas. Exemplo: posts de id 1 a 500.000 estão na tabela posts do DB1 e acima deste valor no DB2
- ✓ Key or Hash based Partitioning: usar uma função matemática para determinar o DB a partir do id. Exemplo: $\text{id} \% 10 = 2$, o dado está no DB2, $\text{id} \% 10 = 3$, o dado está no DB3.
- ✓ Directory based Partitioning: usar uma tabela ou serviço de diretório para armazenar onde encontra-se o dado procurado. A desvantagem é que esta tabela/serviço de diretórios torna-se um SPOF (Single Point Of Failure)

Arquitetura - Sharding: Problemas

- ✓ Dependência da programação
- ✓ Consultas SQL complexas para serem cross-shard
- ✓ Consistência dos dados
- ✓ Integridade referencial
- ✓ Complexidade do código
 - ✓ Adicionar novos servidores
- ✓ Escolha do modo de particionamento é importante
- ✓ Os servidores web conectam-se a mais servidores MySQL.
 - ✓ Problemas de rede



segunda-feira, 19 de abril de 2010

Dando uma folga pro MySQL

- ✓ “Não é só porque você tem um martelo que tudo é um prego”
- ✓ Para algumas aplicações pode-se usar outras tecnologias
 - ✓ Memcached
 - ✓ Sphinx
 - ✓ NoSQL (Cassandra/MongoDB/CouchDB)

Memcached - O que é?

- ✓ Sistema distribuído e de alta performance para fazer cache de objetos em memória RAM. É genérico 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 fornecendo páginas mais rápidas e melhor utilização de recursos.

Memcached - Por que 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)

Memcached

- ✓ 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

Memcached - Exemplo (PHP)

```
<?php
$memcache = new Memcache;
$memcache->connect('localhost', 11211);
$conteudo = $memcache->get('estados');
if($conteudo === false) {
    echo "buscando do banco de dados\n";
    $con = mysql_connect('localhost','user','senha');
    mysql_select_db("banco");
    $res = mysql_query("select * from state",$con);
    $conteudo = mysql_fetch_array($res, MYSQL_ASSOC);
    $memcache->set('estados',$conteudo,false,100);
    mysql_close($con);
}
var_dump($conteudo);
```

Memcached e MySQL

- ✓ O memcache_engine permite que o Memcached trabalhe como uma "storage engine" do MySQL. Isto permite fazer SELECT/UPDATE/INSERT/DELETE em uma tabela armazenada no Memcached
 - ✓ http://tangent.org/index.pl?node_id=506
- ✓ Existe um conjunto de MySQL UDFs (user defined functions) que permite acessar dados do Memcached de dentro de uma consulta SQL ou uma Trigger/Procedure
 - ✓ http://tangent.org/586/Memcached_Functions_for_MySQL.html

Sphinx

- ✓ Sphinxsearch é um motor de pesquisa de texto completo, distribuído sob a versão GPLv2 .
- ✓ Sphinx foi especialmente desenvolvido para integrar um banco de dados SQL com uma linguagem de 'scripts' (oficialmente suportada - PHP). Atualmente o Sphinx conecta-se diretamente com os bancos de dados MySQL e Postgres
- ✓ O nome "Sphinx" é um acrônimo oficial da frase 'SQL Phrase Index'.

Sphinx - Arquitetura

- ✓ O Sphinxsearch inclui os seguintes programas:
 - ✓ indexer - um utilitário para criar índices fulltext
 - ✓ search - uma simples (teste) utilidade para consulta índices fulltext de linha de comando
 - ✓ searchd - um servidor('daemon') para pesquisar através de índices fulltext. (web utilizando scripts Sphinx API; ou MySQL com SphinxSE, ou o seu servidor de aplicações)
 - ✓ sphinxapi - um conjunto de APIs de bibliotecas populares da web (API - PHP, Python, Java, Perl e Ruby).

Sphinx - Características

- ✓ Alta velocidade de indexação (10 MB/s em uma CPU)
- ✓ Alta velocidade de pesquisa (menor que 0.1 segundo em 2-4 GB de texto)
- ✓ Alta escalabilidade (mais de 100 GB de texto, mais de 100 M de documentos em uma CPU)
- ✓ Suporte a pesquisa distribuída
- ✓ Suporte nativo ao banco de dados MySQL
- ✓ Suporte a pesquisa por frases
- ✓ Suporte a 'ranking' de proximidade das frases, com boa relevância

Sphinx - Características

- ✓ Suporte a 'stemming' de Inglês e Russo
- ✓ Suporte a vários números de campos de documentos (os pesos podem ser mudados na hora)
- ✓ Suporte a grupos de documentos
- ✓ Suporte a 'stopwords'
- ✓ Suporte a diferentes modos de pesquisas ("match all", "match phrase" e "match any" - v.0.9.5)
- ✓ PHP API

Sphinx X MySQL

```
mysql> select id from entity where name = 'Apple';
```

```
2 rows in set (0.01 sec)
```

```
mysql> select id from entity where name like '%Apple%';
```

```
6610 rows in set (36.49 sec)
```

```
[root@linux ~]# search -i entity apple
```

```
Sphinx 0.9.9-release (r2117)
```

```
Copyright (c) 2001-2009, Andrew Aksyonoff
```

```
using config file '/usr/local/etc/sphinx.conf'...
```

```
index 'entity': query 'apple ': returned 1000 matches of
```

```
5930 total in 0.001 sec
```

NoSQL

- ✓ Algumas aplicações podem se beneficiar de um ambiente “sem esquema”, ou “não normalizado”
- ✓ Os bancos NoSQL (ou não relacionais) possuem a capacidade de escalar de maneira mais rápida
- ✓ Mas não são todas as aplicações que podem “abrir mão” da integridade ACID

NoSQL - Exemplos

- ✓ Orientados a Documento
 - ✓ CouchDB
 - ✓ MongoDB
- ✓ Orientados a Chave/Valor (Key/Value)
 - ✓ Memcachedb
 - ✓ Project Voldemort
 - ✓ Redis
 - ✓ SimpleDB
 - ✓ Hbase
- ✓ Tabular
 - ✓ Cassandra
 - ✓ Hypertable

Referências

- ✓ Alto Desempenho em MySQL. Editora Alta Books
- ✓ <http://ha-mc.org/node/24>
- ✓ <http://dev.mysql.com/news-and-events/newsletter/2004-01/a0000000301.html>
- ✓ <http://architects.dzone.com/news/your-mysql-server-loaded>
- ✓ <http://www.dicas-l.com.br/dicas-l/20090511.php>
- ✓ <http://axonflux.com/mysql-sharding-for-5-billion-p>
- ✓ <http://www.jurriaanpersyn.com/archives/2009/02/12/database-sharding-at-netlog-with-mysql-and-php/#databasesetup4>
- ✓ <http://www.hitk.com.br/?q=node/40>
- ✓ <http://www.plugmasters.com.br/sys/materias/888/1/Otimiza%E7%E3o-de-Aplica%E7%E5es-MySQL---Parte-II>
- ✓ <http://www.mysqlperformanceblog.com/mysql-performance-presentations/>

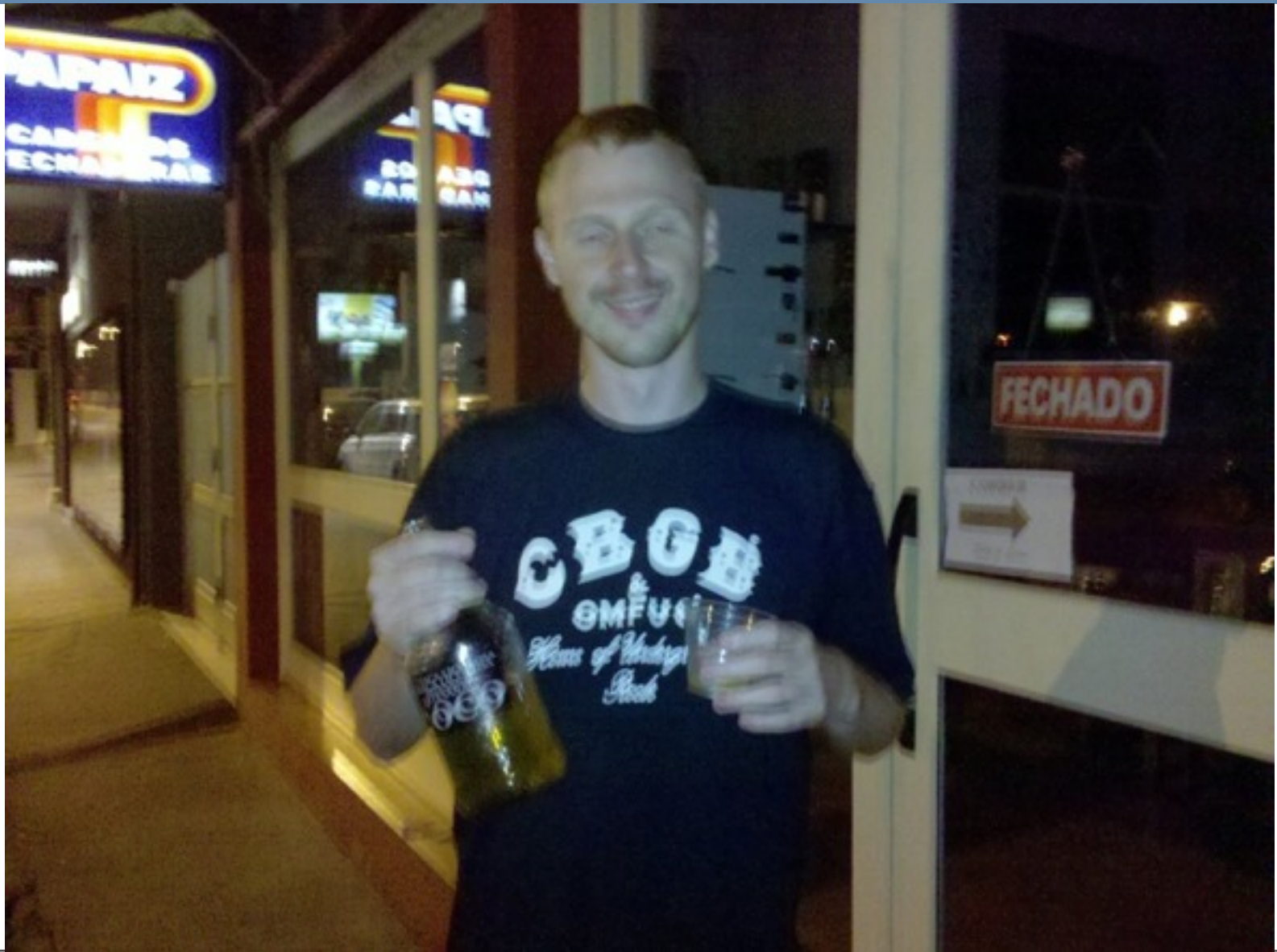
Referências

- ✓ http://articles.techrepublic.com.com/5100-10878_11-5211760.html
- ✓ <http://josefernandes.pt/artigos/optimizar-mysql-linux>
- ✓ <http://www.mysqlperformanceblog.com/>
- ✓ <http://www.linux.com/archive/feature/41348>
- ✓ <http://www.myloadtest.com/>
- ✓ <http://blog.thiagobelem.net/mysql/2009/07/08/otimizando-consultas-mysql-com-o-memcached-563/>
- ✓ <http://ilkinbalkanay.blogspot.com/2010/03/load-testing-relational-databases-with.html>
- ✓ <http://www.sphinxsearch.com.br/Sobre-SphinxSearch>

Contato

```
<?php
$card = array(
    'nome' => 'Elton Luís Minetto',
    'site' => 'http://www.eltonminetto.net',
    'e-mail' => 'elton.minetto@drimio.com',
    'twitter' => '@eminetto',
    'fone' => '(47) 9189 6359'
);
var_dump($card);
?>
```

PHPinga



segunda-feira, 19 de abril de 2010