



#MEMCACHED

Alexandre, Gustavo, Marco Aurélio, Rodrigo, Yago

#MEMCACHED

1. Visão geral: O que é? O que não é?
2. Aplicabilidade
3. Memcached users
4. Demonstração
5. Perguntas ?

1.1 O QUE É?

1 Visão geral



1.1) Visão geral - O que é?

- **Memcached** é um cache para armazenar informações frequentemente usadas para evitar o carregamento e processamento de informações de origens mais lentas, como discos ou um banco de dados.
- O Memcached é um projeto de software livre projetado para fazer uso da RAM sobressalente em muitos servidores para agir como um cache de memória para informações acessadas com frequência.
- **OBJETIVO:** acelerar processos de aplicativos.



1.2 O QUE NÃO É?

1 Visão geral

1.2) Visão geral - O que não é?

- ❑ O Memcached **NÃO** é um banco de dados.
 - ❑ O Memcached nunca deverá ser usado como a única origem de dados de que necessita para executar seu aplicativo.
 - ❑ Os dados sempre deverão ser deriváveis de alguma outra origem.
 - ❑ O Memcached é somente um armazenamento de chave/valor. Não é possível realizar uma consulta sobre os dados ou realizar uma iteração no conteúdo para extrair informações.



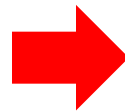
2 APLICABILIDADE

2) Aplicabilidade

- Grande parte das aplicações, especialmente aplicações web, precisam otimizar o acesso as informações e retorná-las de maneira rápida ao cliente. Porém, freqüentemente, a mesma informação é buscada e devolvida. O carregamento de dados da sua fonte (banco de dados ou sistema de arquivos) pode ser ineficiente, especialmente se você requisitar periodicamente as mesmas informações.

Problema

**Sistema ineficiente,
repetitiva busca dos
mesmos dados**



Otimização: Memcached

“Este é o lugar onde memcached pode ajudar. Memcached fornece um armazenamento de memória generalizada que pode armazenar qualquer coisa.”

2) Aplicabilidade – Exemplos

- ▣ Sessões de usuário: uso do memcached
 - guardar os registros de cada sessão de usuário em disco pode ser ineficiente caso haja muitos usuários logados ao mesmo tempo;
 - guardar os registros em banco de dados pode sofrer um pouco caso seja necessário atualizar a tabela de sessões a cada página visitada pelo usuário
 - memcached seria a melhor opção, pois basta armazenar as informações de cada sessão numa entrada cuja chave seja id ou token do cookie do user.

2) Aplicabilidade – Exemplos

▣ Quem está online:

- Parecido com o anterior, armazena-se no memcached uma entrada cuja chave vai ser o id ou token do cookie do usuário e armazena nela o momento em que o usuário fez a ultima ação no site, e vai atualizando a cada página que ele abrir.

▣ Contador de visitas:

- Uma forma de fazer isso seria, num primeiro acesso à uma página, consultar no banco de dados o número de visitas da pagina e em seguida armazenar esse valor já incrementado numa entrada do memcached
- Nos acessos subsequentes a leitura e o incremento serão feitos direto no memcached.

2) Aplicabilidade - Outros exemplos

Site Type	Repeatable Use
Social networking	Profile caching
Content aggregation	HTML/page caching
Ad targeting	Cookie/profile tracking
Gaming and entertainment	Session caching
Location-based services	DB query scaling
Relationship	Session caching
E-commerce	Session and HTML caching

2) Aplicabilidade

□ Concluindo:

- ▣ Memcached é aplicável com objetivos de otimização;
- ▣ Perfeito para websites com grande carregamento de dados;
- ▣ Pode ser usado com o objetivo de reduzir a quantidade de I/Os a um disco rígido;
- ▣ **Não** deve ser usado para cache de objetos grandes.

#3 MEMCACHED USERS



[My LJ](#)[Journal](#)[ScrapBook](#)[Manage](#)[Search](#)[Gift S](#)[Update](#)[Recent](#)[Archive](#)[Friends ...](#)[Info ...](#)[Memor](#)

[Search](#) : **By Interest**

Interests

Interested users

The following users are interested in **43 things**. If
More fun stuff can be found on the [interests page](#)

43 matches:

LIVEJOURNAL



FACEBOOK

WIKIPEDIA



FLICKR



bebo

BEBO

you blew it.



YOUTUBE

WORDPRESS



WORDPRESS

TWITTER



Twitter



• • •

Typepad, Yellowbot, Digg, Craigslist, Mixi ...

4 DEMONSTRAÇÃO



5 PERGUNTAS ?

