

# Memcached

---

Optimizando y escalando una “granja web”

Mario Andrés Pérez Batanero  
Jonathan Molinero Castillo  
Álvaro Fernández Palma

# Introducción

“memcached” fué desarrollado en primer lugar para intentar mejorar el rendimiento de un sistema web que realizaba un uso intensivo de las BBDD, realizando gran cantidad de consultas SQL.(uso INTENSIVO del DISCO DURO) (Paginas web DINAMICAS).

Y ha acabado siendo un **sistema de memoria**(en RAM) **caché distribuida**.

Desarrollo de Brad Fitzpatrick para mejorar el backend de Live

Arquitectura cliente-servidor

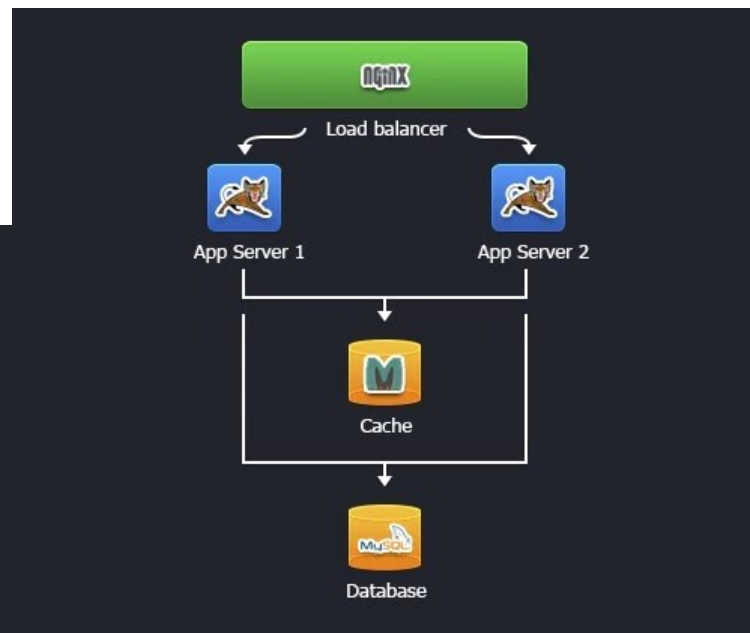
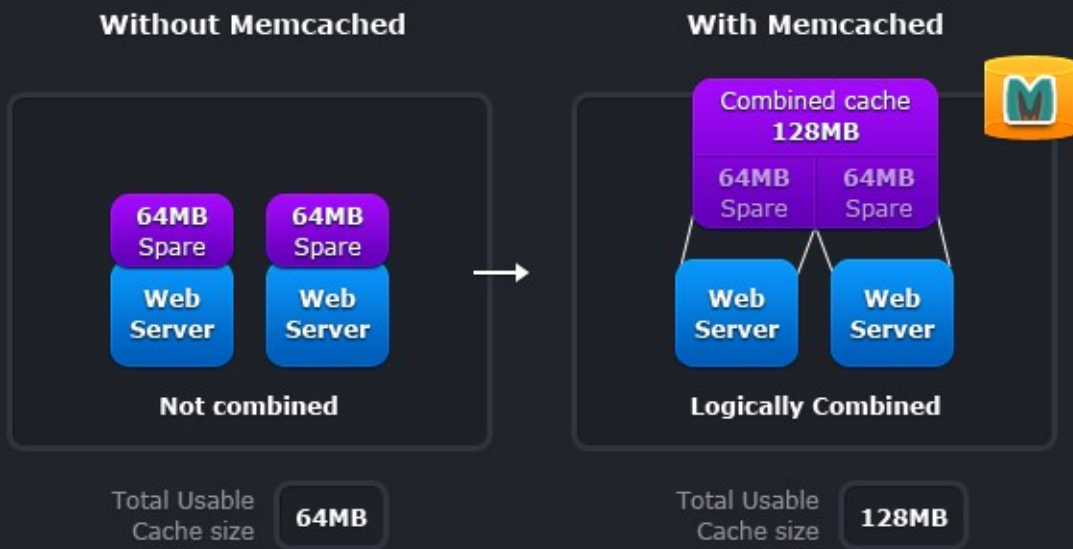
Escrito en C - Licencia BSD (libre con restricciones)

☐ Wikipedia, Facebook, Youtube, Twitter y en general **sitios web con tráfico elevados** necesitan a “**memcached**” o algún otro

API disponible para la mayoría de lenguajes



# Memcached $\approx$ Memoria web

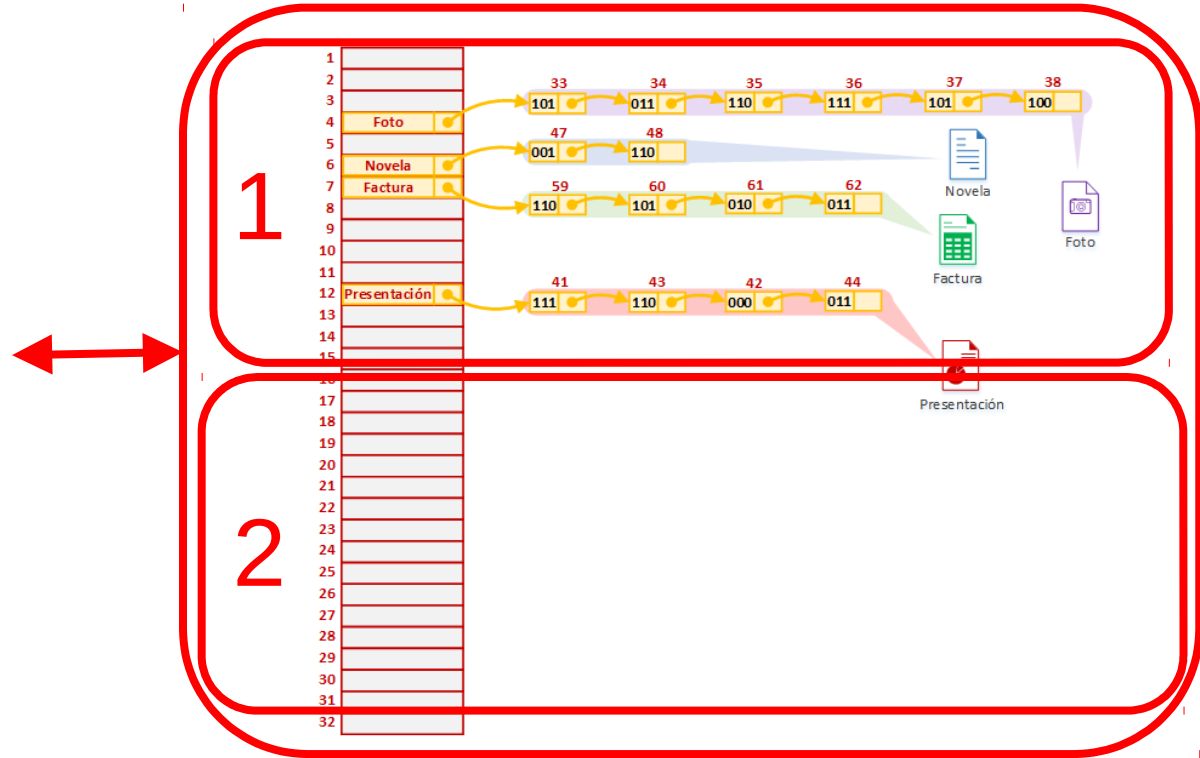


I. **Caché en RAM** (Evita cuello botella BBDD, etc)

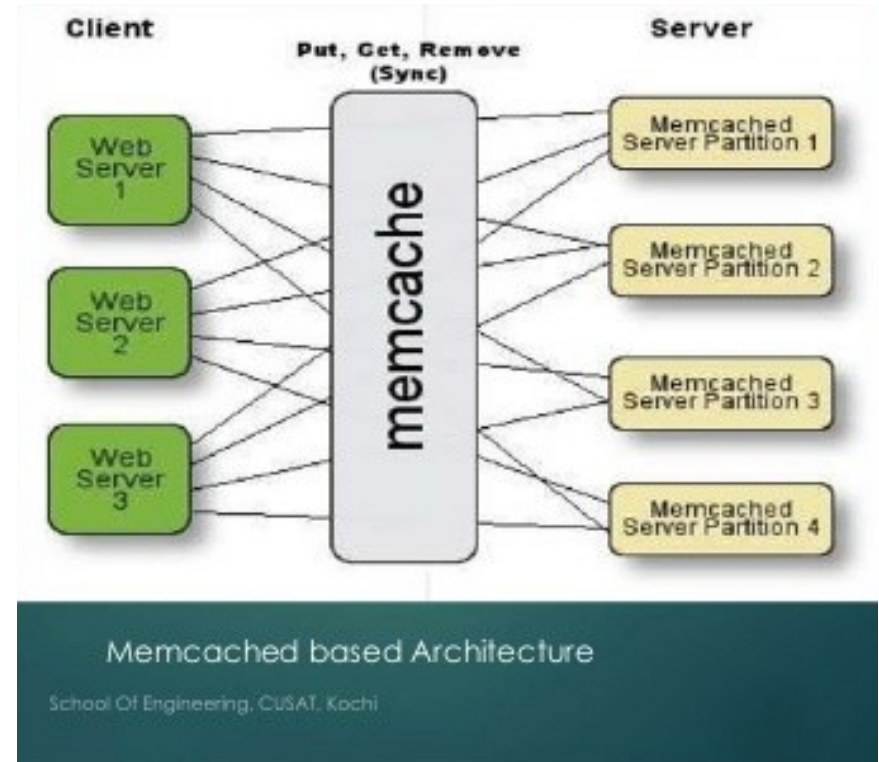
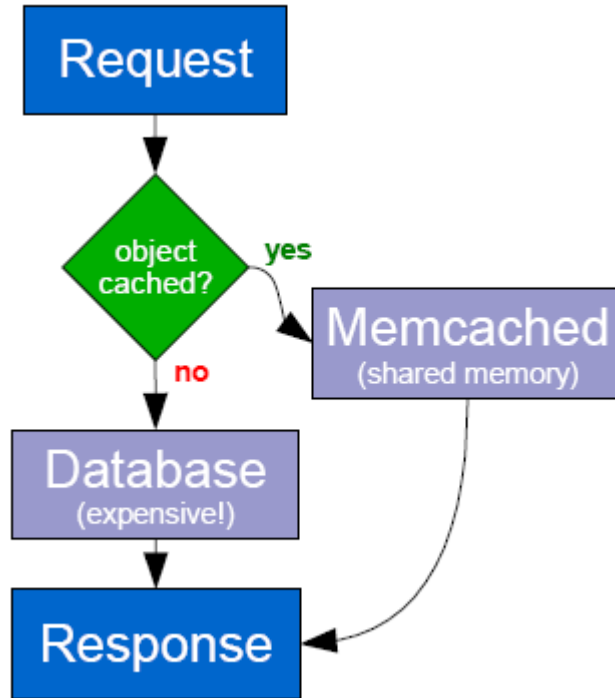
II. **Memoria compartida** entre servidores (sesión [php] activa, etc)

**ESCALABILIDAD**

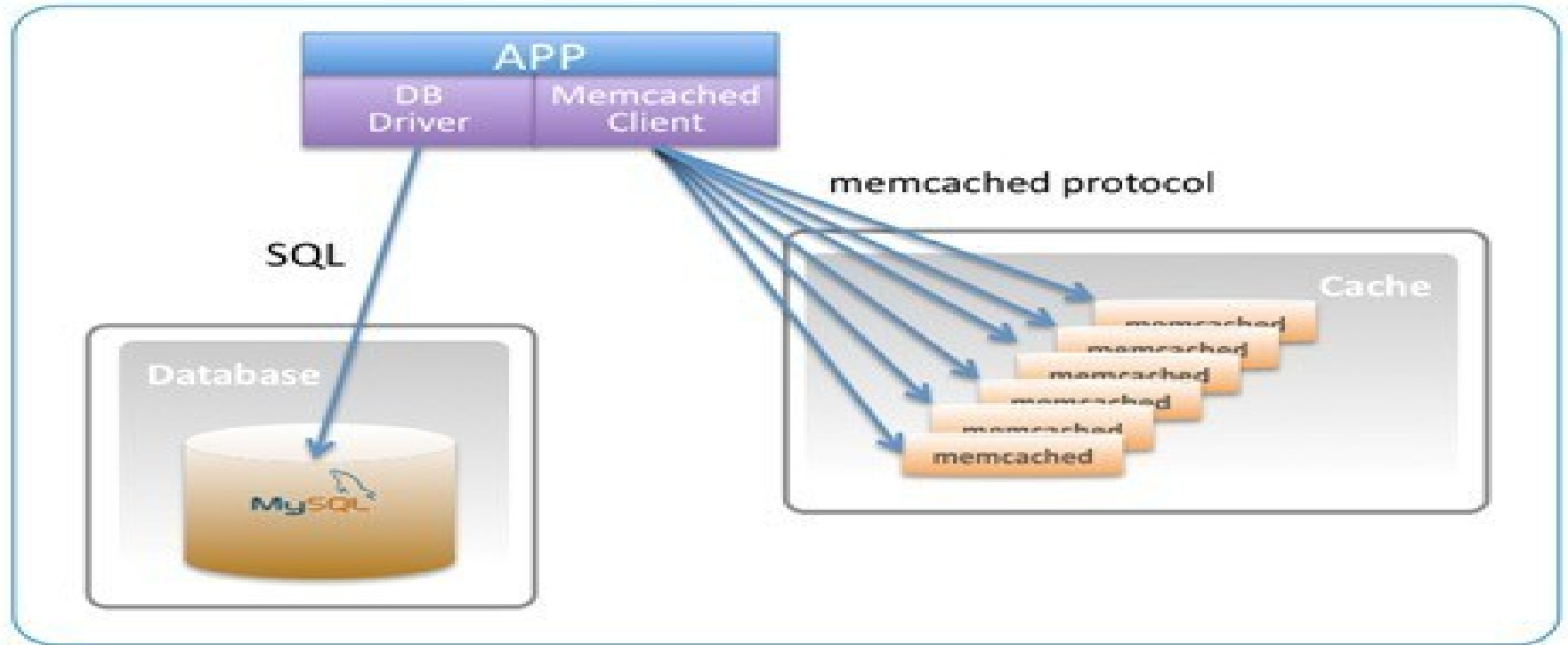
# Memcached $\approx$ Tabla hash distribuida



# Esquema de funcionamiento

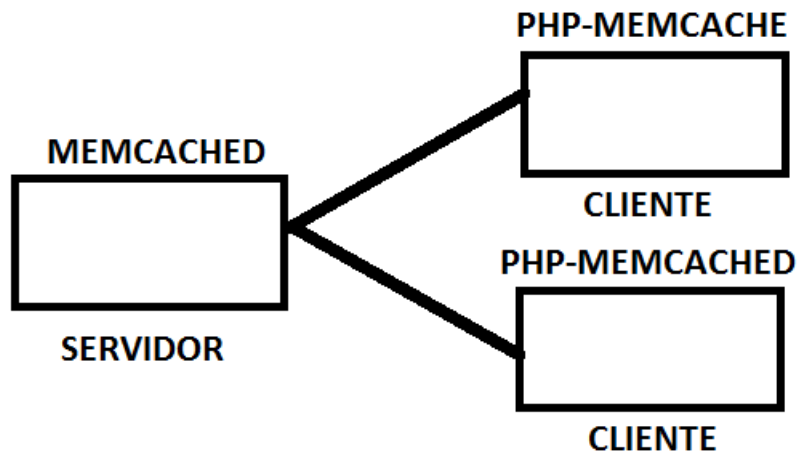


# Cliente - Servidor



# Cliente MEMCACHED

(PHP-MEMCACHE  
Vs  
PHP-MEMCACHED)



	<a href="#">pecl/memcache</a>	<a href="#">pecl/memcached</a>
First Release Date	2004-06-08	2009-01-29 (beta)
Actively Developed?	Yes	Yes
External Dependency	None	<a href="#">libmemcached</a>
Automatic Key Fixup <sup>1</sup>	Yes	No
Append/Prepend	No	Yes
Automatic Serialization <sup>2</sup>	Yes	Yes
Binary Protocol	No	Optional
CAS	No	Yes
Compression	Yes	Yes
Communication Timeout	Connect Only	Various Options
Consistent Hashing	Yes	Yes
Delayed Get	No	Yes
Multi-Get	Yes	Yes
Session Support	Yes	Yes
Set/Get to a specific server	No	Yes
Stores Numerics	Converted to Strings	Yes

# Configuración básica (server)

/etc/memcached.conf

```
-l  
[IP_EN_RED_SERVIDORES_Mem  
cached] // IP en la RED
```

/etc/php5/conf.d/memcache.ini

```
; uncomment the next line to  
enable the module  
extension=memcache.so  
//ACTIVAR módulo memcache o  
memcached de PHP
```

## Configuración básica (client)

Instalar php5-memcached

La configuración nos bastaría (comprobamos con



# Programación php (Uso de API)

Funcion	DE VU EL VE	Descripción
<b>addServer</b> ( string \$host , int \$port, int \$weight )	boolean	Servidores memcached
<b>get</b> (string \$key [, callable \$cache_cb [, float &\$cas_token ]] )	mixed (boolean o item)	Devuelve el "item" correspon diente a la "key" sino devuelve "false"
<b>set</b> ( string \$key	boolean	Almacena

# Ejemplo de uso - Caché php

```
<?php
$mem = new Memcache();
$mem->addServer("10.0.0.11",
11211);
$mem->addServer("10.0.0.12",
11211);

$result = $mem->get("prueba");

if ($result) {
    echo $result;
} else {
    echo "Ninguna clave que
coincida. La almacenara a
continuación";
    $mem->set("prueba", "Yo soy
los datos en memcached!") or
die("No pude almacenar nada en
memcached!");
}
```

# Configuración para soporte sesiones PHP

/etc/php5/apache2/php.ini

```
session.save_handler = memcache
//Manejador de
sesión->memcache
session.save_path =
'tcp://10.0.0.11:11211,tcp://10.0.0.1
2:11211' //Lista servidores
```

/etc/php5/mods-available/memcache.ini (\*si falla algún servidor  
memcached)

```
memcache.allow_failover=1
//Habilitar tolerancia a fallos
memcache.session_redundancy=3
//Debe ser 1 más por un bug de
PHP
```

# Ejemplo de uso - Información de sesión comp.

```
<?php // prueba_sesion.php
header('Content-Type:
text/plain');
session_start();
if(!isset($_SESSION['visit']))
{
    echo "This is the first time
you're visiting this server\n";
    $_SESSION['visit'] = 0;
}
else
    echo "Your number of
visits: ".$_SESSION['visit'] . "\n";

$_SESSION['visit']++;

echo "Server IP: ".
$_SERVER['SERVER_ADDR'] .
"\n";
echo "Client IP: ";
```

# Ejemplo de uso - Información de sesión comp.

```
$ curl -v -s  
http://10.0.0.11/prueba_sesion.p  
hp 2>&1 | grep 'Set-Cookie:'
```

```
< Set-Cookie:  
PHPSESSID=8lehte2dnqegtp1  
q3v9pau08k4; path=/  

```

Your number  
of visits: 1

Server IP:

10.0.0.11

Client IP:

10.0.0.10

Array

(

[PHPSESSI  
D] =>

8lehte2dnqegt  
p1q3v9pau08k4

Your number of  
visits: 2

Server IP:

10.0.0.12

Client IP:

10.0.0.10

Array

(

[PHPSESSID]  
=>

8lehte2dnqegtp1  
q3v9pau08k4

# Ejemplo de uso - Tolerancia a fallos

- En el servidor

1(10.0.0.11):

```
$ service memcached stop
```

- Accedemos al servidor

1 (ESTA CAIDO)

```
$ curl --cookie
```

```
"PHPSESSID=8lebte2dnqegtp1q3v  
9pau08k4"
```

```
http://10.0.0.11/prueba_sesion.php
```

- Salida (**De forma transparente accede a la sesión del otro servidor**)

```
Your number of visits: 3
```

```
Server IP: 10.0.0.11
```

```
Client IP: 10.0.0.10
```

# Algunas de las ventajas o conclusiones

Cambiamos **accesos pesados a BBDD** por **acessos ligeros a caché(memoria RAM)**

Reducción brutal de carga en BBDD

Una página típica tiene 0..5 queries, y 100+ accesos a Memcached

\*\*\*Un PERO.. Cuello de botella: Escrituras en la base de datos

Nodo de **memoria virtual compartida** entre servidores

Aumento en los recursos de memoria si se comparte(**ESCALABILIDAD**)

Redundancia de sesiones PHP y tolerancia a fallos

# Bibliografía/Referencias

[Memcached - Sitio web oficial](#)

[Memcached - Wiki oficial](#)

[Memcached - Wikipedia](#)

[PHP.net: Memcached - Manual](#)

[Memcached - Blog Riaolanetworks](#)