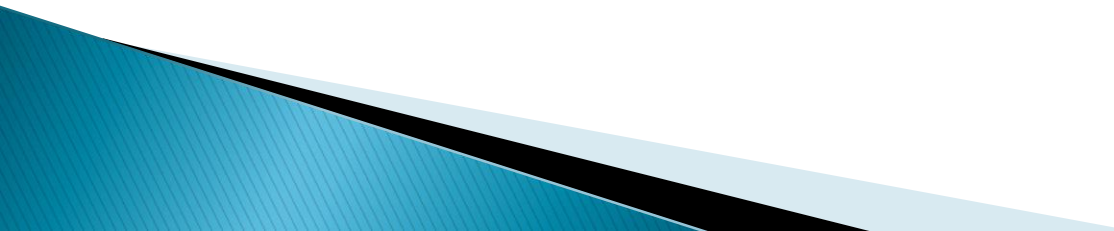


¿Qué es Memcached y cómo usarlo?

Carmen Arcos Aneas y María Orgaz Martínez



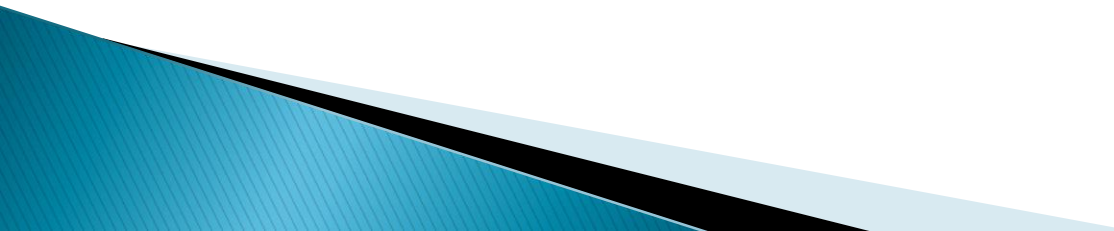
Índice

- ▶ ¿Qué es Memcached?
 - ▶ Funcionamiento
 - ▶ ¿Quiénes lo usan?
 - ▶ Instrucciones
 - ▶ Ventajas
 - ▶ Instalación
 - ▶ Ejemplo de uso
- 

¿Qué es Memcached?

- ▶ **Memcached** es una herramienta desarrollada por la empresa Danga Interactive, que mantiene el proyecto bajo licencia BSD y que fue pensado para incrementar la velocidad de aplicaciones web dinámicas, reduciendo la carga de la base de datos.

Funcionamiento

- ▶ Su funcionamiento se basa en una tabla hash, distribuida a lo largo de varios equipos, donde va almacenando los valores asociados a una clave única para cada elemento. Conforme ésta se va llenando, los datos que más tiempo llevan sin ser utilizados se borran para dar espacio a los nuevos.
 - ▶ Cada dato introducido en Memcached tiene un tiempo de expiración; pasado ese tiempo, el servicio Memcached lo elimina, dejando ese espacio para el almacenamiento de otro objeto
- 

¿Quiénes lo usan?

- ▶ Este sistema es empleado por varios de los sitios más activos y visitados de la red, como YouTube, Reddit, Playdom, Zynga, Facebook y Twitter.



Instrucciones

- ▶ **Set():** Guarda el valor dado por value en un servidor de memcached bajo la clave especificada por key. El parámetro expiration se puede emplear para controlar cuándo se considera que ha expirado el valor.

```
set(string $key , mixed $value [, int $expiration ] )
```

- ▶ **Replace():** Es similar a set(), pero la operación fallará si la clave dada por key no existe en el servidor.

```
replace(string $key , mixed $value [, int $expiration ])
```



- ▶ **AddServer():** Agrega el servidor especificado al grupo de servidores. No se establece la conexión en ese momento. Si vamos a añadir más de un servidor es recomendable usar **addServers()**.

addServer(string \$host , int \$port [, int \$weight = 0])

```
<?php
$m = new Memcached();

/* Añadimos dos servidores, el segundo con más probabilidad de ser seleccionado */
$m->addServer('mem1.domain.com', 11211, 33);
$m->addServer('mem2.domain.com', 11211, 67);
?>
```

- ▶ Flush(): Invalida todos los elementos existentes en caché inmediatamente o después del tiempo especificado.

`flush([int $delay = 0])`

```
<?php
/* Invalida todos los elementos en 10 segundos */
$m->flush(10);
?>
```

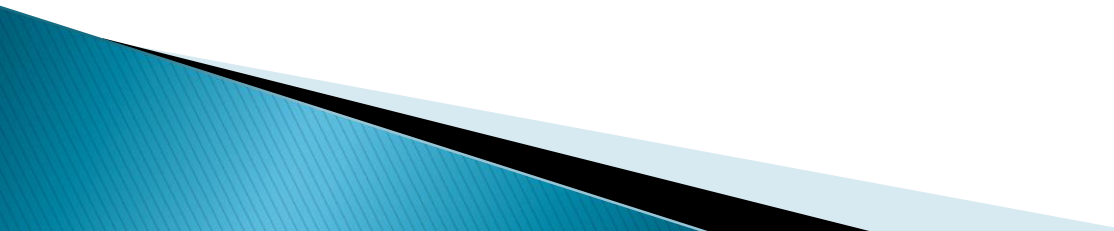
- ▶ Delete(): Esta función elimina la clave de memcached.

`delete(string $key [, int $time = 0])`

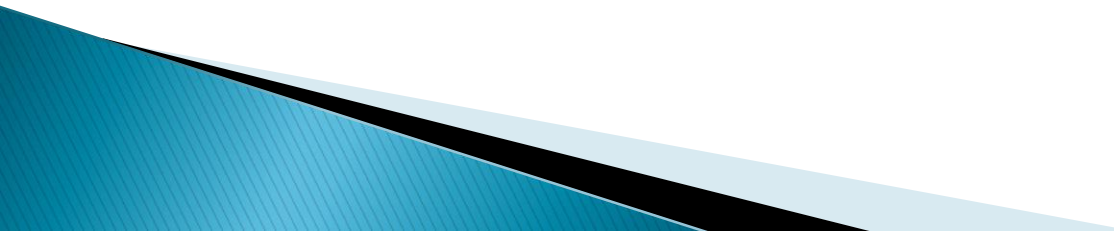
- ▶ Get(): Devuelve el ítem que fue previamente guardado bajo la clave dada por key.

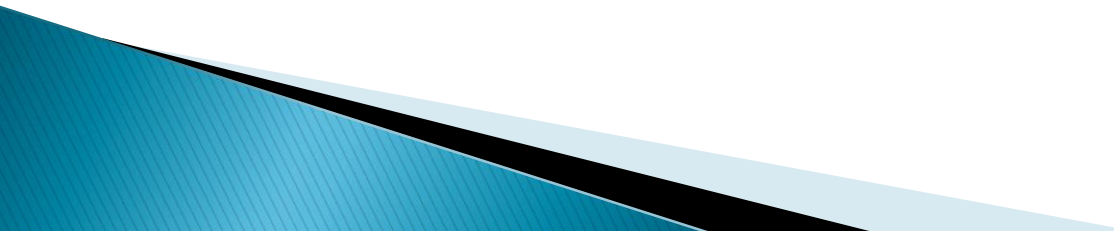
`get(string $key)`

Otras...

- ▶ **getStats()**: Devuelve un array que contiene el estado de todos los servidores de memcached disponibles.
 - ▶ **fetch()**: Obtiene el siguiente resultado de la última petición.
 - ▶ **fetchAll()**: Obtiene todos los resultados restantes de la última petición.
- 

Ventajas

- ▶ Reducir los niveles de carga de los servidores. Ésta es la primera gran ventaja del uso de este sistema, ya que gracias a él se reducen el número de consultas a la base de datos.
 - ▶ Memcached permite ajustar el espacio de memoria dedicado, dado que se ejecuta como servidores montados sobre uno o varios equipos, permitiendo indicar la cantidad de memoria destinada para el almacenamiento de información.
- 

- ▶ Este sistema ofrece la posibilidad de almacenar lo que se quiera, a criterio del desarrollador. Lo que facilita que se puedan cachear cosas tan diversas como resultados de cálculos o consultas a base de datos complejas, información de sesiones de los usuarios, etc.
 - ▶ La comunicación que se produce entre clientes y el servidor es muy sencilla, y basada en comandos.
 - ▶ Ofrece la posibilidad de controlar el tiempo de vida de un objeto, indicando el “tiempo de expiración” asociado al mismo, en el momento de realizar una operación de almacenamiento.
- 

Instalación

- ▶ **Prerequisitos:** Tener instalado Ubuntu 14.0 y Tener instalado LAMP (Apache + MySQL+ PHP)
- ▶ Según el tipo de sistema operativo:
 - Ubuntu/Debian:** `sudo apt-get install memcached`
 - RedHat/Fedora:** `yum install memcached`
 - FreeBSD:** `portmaster databases/memcached`

```
root@ubuntu:/home/maria# sudo apt-get install memcached
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  libcache-memcached-perl libmemcached
The following NEW packages will be installed:
  memcached
0 upgraded, 1 newly installed, 0 to remove and 234 not upgraded.
Need to get 66.7 kB of archives.
After this operation, 229 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu/ trusty/main memcached amd64 1.4.14-0u
buntu9 [66.7 kB]
0% [1 memcached 0 B/66.7 kB 0%]
```

Es muy importante activar el paquete de memcached.

Accedemos al archivo memcached.ini que encontramos en la ruta /etc/php5/conf.d y descomentamos la línea que contiene el siguiente texto:
extension=memcached.so

Reiniciamos Apache para actualizar los cambios:

```
sudo /etc/init.d/apache2 restart
```

► Comprobamos la correcta instalación

Para comprobar que se ha instalado correctamente creamos un archivo llamado `info.php` con el siguiente contenido:

```
<html>
<body>
<?php phpinfo();
?>
</body>
</html>
```

memcached

memcached support	enabled
Version	2.1.0
libmemcached version	1.0.8
Session support	yes
igbinary support	no
json support	no

Directive	Local Value	Master Value
memcached.compression_factor	1.3	1.3
memcached.compression_threshold	2000	2000
memcached.compression_type	fastlz	fastlz
memcached.serializer	php	php
memcached.ssess_binary	0	0
memcached.ssess_lock_wait	150000	150000
memcached.ssess_locking	1	1
memcached.ssess_prefix	memc.sess.key.	memc.sess.key.

Ejemplo

```
<?php
    require_once('config.php');

    //Creamos el objeto de la clase Memcache
    $memcache = new Memcached();
    $memcache->addServer('localhost', 11211) or die ("Could not connect");

    //Conectamos a la base de datos
    try {
        $conexion = new PDO('mysql:host=localhost;dbname=productos', 'root', 'contraseña');
        $conexion->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    } catch ( PDOException $e ) {
        die($e->getMessage());
    }

    //Creamos la consulta y la key
    $sql = "SELECT usuario,producto FROM usuariosProducto WHERE id=1";
    $key = md5($sql);
```

```
//Comprobamos si los datos están en memoria
$resultado = $memcache->get($key);

//En caso contrario buscamos en la base de datos
if (!$resultado) {
    $resultado = $conexion->query($sql);
    $resultado = $resultado->fetchAll();
    $memcache->set($key, $resultado, time() + 3600);
    $mem = $memcache->getStats();
}

//cerramos la conexión con la bd
$conexion = null;

?>
```


Resultados

- Nuestra Tabla:

```
mysql> select * from usuariosProducto
-> ;
+-----+-----+-----+
| id  | usuario | producto |
+-----+-----+-----+
| 1   | Carmen  | Impresora |
| 2   | Maria   | Portatil  |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

- Tras una primera ejecución obtenemos que el dato no está en memcached almacenado:

```
root@ubuntu:/var/www/html# php ejemplo3.php
no esta en memcached
```

- Si volvemos a ejecutarlo observamos que ya sí está en memcached:

```
root@ubuntu:/var/www/html# php ejemplo3.php
ya esta en memcachedarray(1) {
  [0]=>
  array(4) {
    ["usuario"]=>
    string(5) "Maria"
    [0]=>
    string(5) "Maria"
    ["producto"]=>
    string(8) "Portatil"
    [1]=>
    string(8) "Portatil"
  }
}
```