

Installing memcached And The PHP5 memcache Module On Debian Etch (Apache2)

By Falko Timme

Published: 2008-09-09 18:11

Installing memcached And The PHP5 memcache Module On Debian Etch (Apache2)

Version 1.0

Author: Falko Timme <ft [at] falkotimme [dot] com>

Last edited 08/27/2008

This guide explains how to install [memcached](#) and the PHP5 memcache module on a Debian Etch system with Apache2. memcached is a daemon that can store objects in the system's memory (e.g. results of database queries) which can speed up your web site tremendously. You can use memcached over a network (i.e., install your web application on one server and memcached on another server), but usually you install both on one server to avoid the networking overhead.

It should be noted the memcached is no out-of-the-box solution for speeding up your web applications. Typically you have to adjust your scripts (PHP, Perl, etc.) to work with memcached, so this requires a little bit of work.

This document comes without warranty of any kind! I do not issue any guarantee that this will work for you!

1 Preliminary Note

I have tested this on a Debian Etch server with the IP address `192.168.0.100` where Apache2 and PHP5 are already installed and working. I'll use Apache's default document root `/var/www` in this tutorial for demonstration purposes. Of course, you can use any other vhost as well, but you might have to adjust the path to the `info.php` file that I'm using in this tutorial.

2 Checking PHP5's Current State

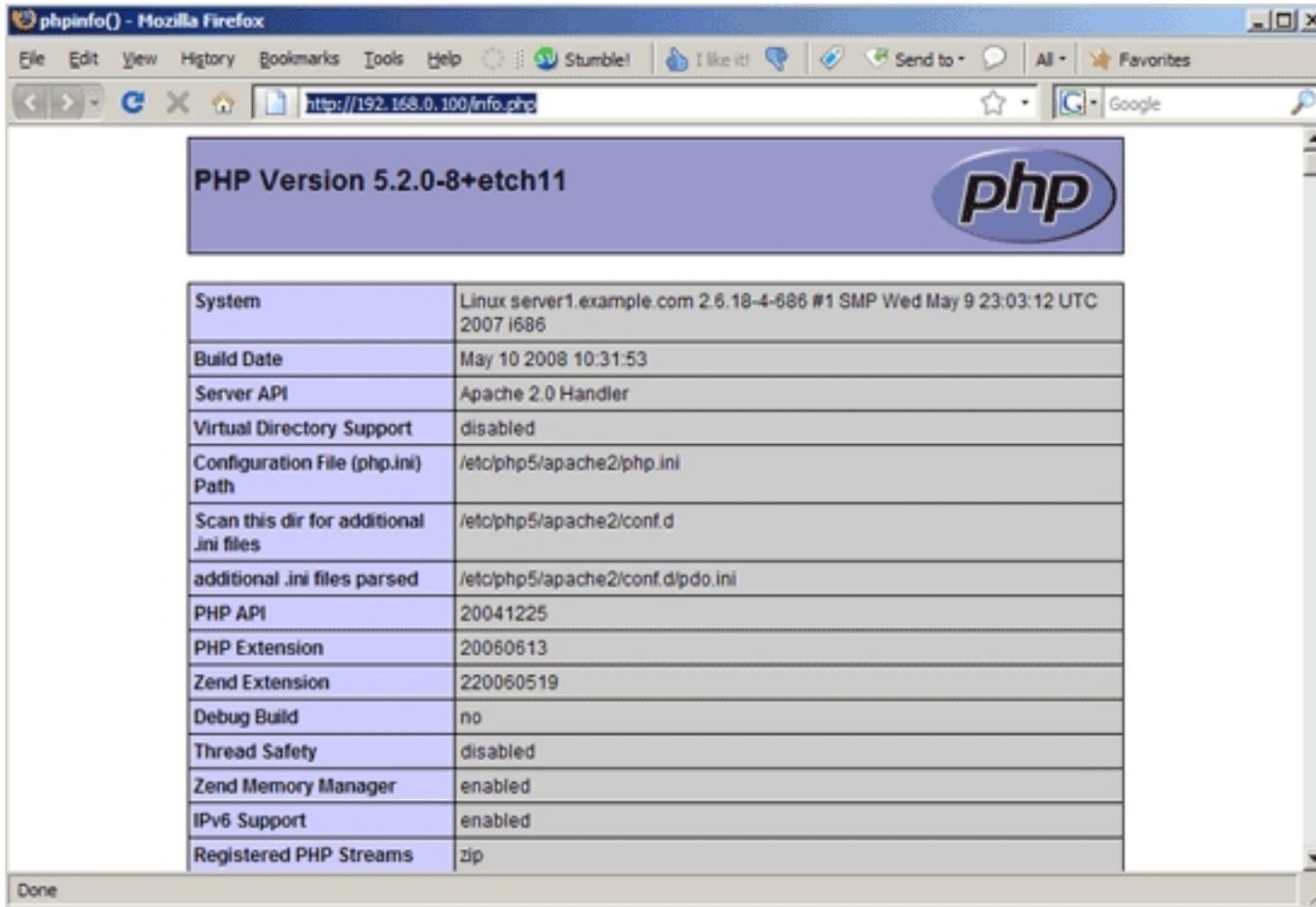
First, before we install memcached, let's find out about our PHP5 installation. To do this, we create the file `info.php` in our document root `/var/www`:

```
vi /var/www/info.php
```

```
<?php  
phpinfo();  
?>
```

Afterwards, we call that file in a browser: `http://192.168.0.100/info.php`

As you see, we have PHP 5.2.0 installed...



... but the PHP5 memcache module isn't mentioned anywhere on the page:

under the GNU Lesser General Public License version 2.1.

Directive	Local Value	Master Value
mbstring.detect_order	no value	no value
mbstring.encoding_translation	Off	Off
mbstring.func_overload	0	0
mbstring.http_input	pass	pass
mbstring.http_output	pass	pass
mbstring.internal_encoding	ISO-8859-1	no value
mbstring.language	neutral	neutral
mbstring.strict_detection	Off	Off
mbstring.substitute_character	no value	no value

mime_magic

mime_magic support	enabled
--------------------	---------

Directive	Local Value	Master Value
mime_magic.debug	Off	Off
mime_magic.magicfile	/usr/share/file/magic.mime	/usr/share/file/magic.mime

Done

3 Installing memcached And The PHP5 memcache Module

memcached and the PHP5 memcache module are available as packages for Debian Etch, so we can install them as follows:

```
apt-get install memcached php5-memcache
```

After the installation, memcached should already be running. You can check that by typing

```
netstat -tap | grep memcached
```

```
server1:~# netstat -tap | grep memcached
tcp        0      0  *:11211                *:*                    LISTEN     3053/memcached
server1:~#
```

As you see, memcached is running on port `11211` (the default memcached port), and it's listening on all interfaces on the system. As memcached has no built-in authentication mechanisms (in order to not give up on speed), this means that anyone can connect to it from outside and use it. To avoid this, you can either close port `11211` in your firewall, or you configure memcached to listen on localhost only. I will use the latter method here.

To do this, open the memcached configuration which is stored in `/etc/memcached.conf`:

```
vi /etc/memcached.conf
```

Add `-l 127.0.0.1` to the configuration (you can also adjust the other settings if you like - the file contains explanations for each setting):

```
[...]
# Specify which IP address to listen on. The default is to listen on all IP addresses
# This parameter is one of the only security measures that memcached has, so make sure
# it's listening on a firewalled interface.
# -l 12.34.56.78
-l 127.0.0.1
[...]
```

Restart memcached...

```
/etc/init.d/memcached restart
```

... and run

```
netstat -tap | grep memcached
```

again. As you see, memcached is now listening on localhost only:

```
server1:~# netstat -tap | grep memcached
tcp          0          0 localhost.localdo:11211 *.*          LISTEN      3092/memcached
server1:~#
```

Afterwards, we restart Apache so that our new PHP configuration takes effect:

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

Afterwards, open *info.php* again in a browser: <http://192.168.0.100/info.php>

You should now see memcache mentioned on the page which means it has successfully been integrated and is working as expected:

The screenshot shows the output of a `phpinfo()` script in a Mozilla Firefox browser window. The browser title is "phpinfo() - Mozilla Firefox" and the address bar shows "http://192.168.0.100/info.php".

The output includes a table of mbstring configuration:

mbstring.func_overload	0	0
mbstring.http_input	pass	pass
mbstring.http_output	pass	pass
mbstring.internal_encoding	ISO-8859-1	no value
mbstring.language	neutral	neutral
mbstring.strict_detection	Off	Off
mbstring.substitute_character	no value	no value

Below this is a section for **memcache** with the following configuration:

memcache support	enabled
Active persistent connections	0
Revision	\$Revision: 1.39 \$

Next is a section for **mime_magic** with the following configuration:

mime_magic support	enabled	
Directive	Local Value	Master Value
mime_magic.debug	Off	Off

The browser status bar at the bottom shows "Done".

To use the PHP memcache module with your PHP applications, you should check out the [memcache examples](#) and the [memcache function](#) reference.

I will use the example script from <http://dk.php.net/manual/en/memcache.examples.php> and save it in the file `/var/www/memcachetest.php`:

```
vi /var/www/memcachetest.php
```

```
<?php

$memcache = new Memcache;
$memcache->connect('localhost', 11211) or die ("Could not connect");

$version = $memcache->getVersion();
echo "Server's version: ".$version."<br/>\n";

$tmp_object = new stdClass;
$tmp_object->str_attr = 'test';
$tmp_object->int_attr = 123;

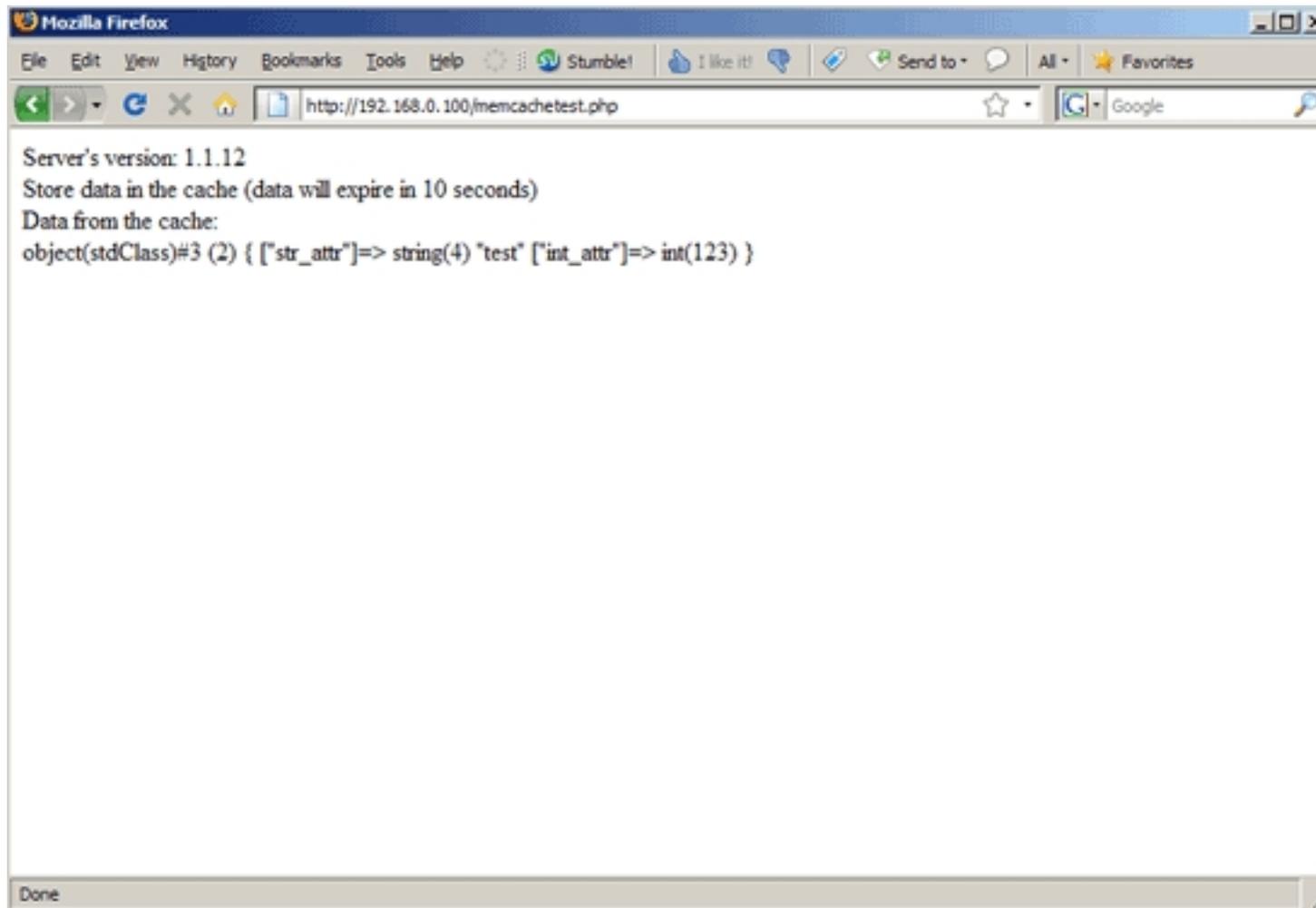
$memcache->set('key', $tmp_object, false, 10) or die ("Failed to save data at the server");
echo "Store data in the cache (data will expire in 10 seconds)<br/>\n";

$get_result = $memcache->get('key');
echo "Data from the cache:<br/>\n";

var_dump($get_result);

?>
```

Then I call that file in a browser (<http://192.168.0.100/memcachetest.php>). If all goes well, the output should look as follows:



4 Links

- memcached: <http://www.danga.com/memcached>
- PHP: <http://www.php.net>
- Apache: <http://httpd.apache.org>

- Debian: <http://www.debian.org>