How to enhance SugarCRM Performance? (v.9)

Ingo Jaeckel (ingo.jaeckel@student.hpi.uni-potsdam.de)

March 2, 2009

1 Motivation

With the steps in this tutorial i can execute my JMeter test script **3.4 times faster than before** . I went from 5 requests per second to 17 requests per second. I think this is already quite impressive although this is just the beginning of performance tuning.

I want to find out what you can do to enhance the performance of SugarCRM running on a single server with a limited amount of cpu power and memory - like 4 GiB RAM, a 2.5 GHz dual core cpu and a "desktop harddisk" (250 GiB, 7200 rpm, 16 MiB cache).

I hope you give me **a lot** feedback and hints on this documentation. I would like to know your experiences about SugarCRM performance tuning. You can discuss about performance tuning issues on the openqc forum `sugarforge.org/projects/openqc/`.

2 The Setup

1. Install
   - apxs (which is included in the apache2-threaded-dev package in ubuntu),
   - the php extensions apc and memcache,
   - memcached [1]

   (see listing 1).

```
1 $ aptitude install apache2-threaded-dev # for building apc
2 $ aptitude install memcached php5-memcache
3 $ pecl install apc
4 $ pecl install memcache
```

Listing 1: Installation of software neccessary for performance tuning.

2. Let your users work with SugarCRM for a while, then check the statistics of apc using the `/usr/share/php/apc.php` script (see figure 1) and adjust your apc configuration if neccessary.

3. Start `memcached`.

   ```
   $ memcached -d -m 16 -l 127.0.0.1 -u <Webserver Username>
   ```

   You need to tell `memcached` the address it should listen on and the amount of memory you want to give it. In this example it should listen on 127.0.0.1 and is allowed to use 16 MiB of RAM.

   It seems like SugarCRM only stores very little amounts of data in that cache (less than one MiB?). You should probably just start memcached, let your users work with SugarCRM for a while and then have a look at your memcache stats and adjust your memcache configuration if neccessary. You can do that with the `memcache.php` PHP script [2] (see figure 2).

3 PHP Configuration

1. Enable apc and memcache in your php configuration file [3] (see listing 2).

```
1  extension=apc.so
2  apc.shm_size = 128
3  apc.shm_segments = 2
4
5  extension=memcache.so
6  [memcache]
7  memcache.dbpath="/var/lib/memcache"
8  memcache.maxreclevel=0
9  memcache.maxfiles=0
10 memcache.archivememlim=0
11 memcache.maxfilesize=0
12 memcache.maxratio=0
```

Listing 2: Append these lines to your php.ini file to enable and configure apc and memcache.

4 Apache configuration

1. Enable the Apache modules
   - `mod_expires`
   - `mod_cache`
   - `mod_deflate`[1]
   - `mem_cache`
2. Configure `mod_mem_cache` by editing the `mem_cache.conf` file (see listing 3).

```
1 <IfModule mod_mem_cache.c>
2   CacheEnable mem /
3   MCacheSize 102400
4   MCacheMaxObjectCount 10000
5   MCacheMinObjectSize 1
6   MCacheMaxObjectSize 2048
7 </IfModule>
```

Listing 3: Enlarge MCacheSize to 102400 (100 MiB) and MCacheMaxObjectCount to 10000.

3. Configure `mod_deflate` by editing `deflate.conf` file (see listing 4).

```
1 <IfModule mod_deflate.c>
2           AddOutputFilterByType DEFLATE text/html text/plain
               text/xml text/javascript application/x-
               javascript
3 </IfModule>
```

Listing 4: You should configure the compression of JavaScript files.

4. Configure `mod_expires` by editing `expires.conf` file [4] (see listing 5). This **greatly**[2] reduces the number of requests the browser sends to the server.

```
1 <IfModule mod_expires.c>
```

[1] In my tests the use of `mod_deflate` had almost no affect. I could not measure any serious increase/decrease in speed. However the use of `mod_deflate` is highly recommended in general when tuning a LAMP environment.

[2] From 55 requests per pageview to one request per pageview for example.

```
2     ExpiresActive On
3     ExpiresByType image/png "access plus 10 years"
4     ExpiresByType image/gif "access plus 10 years"
5     ExpiresByType text/css "access plus 10 years"
6     ExpiresByType text/javascript "access plus 10 years"
7     ExpiresByType application/x-javascript "access plus 10 years
        "
8   </IfModule>
```

Listing 5: Images, JavaScript and CSS files should expire in 10 years after accessing them the first time. **Check if this makes sense in your setup!**

5 MySQL configuration

1. Increase the MySQL query cache size. Add this line to the mysqld section of your `my.cnf` file.

   ```
   query_cache_size = 2M
   ```

   Play around with this value and check how much space in the cache is still left using the MySQL statement `SHOW STATUS LIKE 'Qcache_free_memory'` for example.

2. Change default MySQL storage engine to InnoDB. Add this line to the mysqld section in your `my.cnf` file.

   ```
   default-storage-engine = innodb
   ```

3. If you want to change the type of all tables of an existing database to InnoDB use the `ALTER TABLE <table> TYPE=INNODB` statement. Iterate over the table names of a database and execute the alter table statement on each table[3].

   ```
   $ for t in ‘echo ’show table status’ \
       | mysql -u <MySQL username> <DB name> \
       | cut -f 1‘; do \
     echo ’alter table $t type=InnoDB;’
       | mysql -u <MySQL username> <DB name>; \
     done
   ```

[3]This listing assumes that your MySQL password is empty.

6 SugarCRM configuration

1. You should add five lines to your `config.php` file to speedup SugarCRM [5] (see listing 1).

```
<?php
$sugar_config = array (
  ...
  'hide_subpanels' => true,
  'hide_subpanels_on_login' => true,
  'verify_client_ip' => false,
  'disable_vcr' => true,
  'disable_count_query' => true,
);
```

7 Measuring performance with JMeter

To be written.

1. If you are running Windows you can use Badboy to record some tests you want to execute later [6]. You can export the recording to a JMeter script. After exporting the recording to the JMeter script, you can use JMeter to configure and execute your tests.

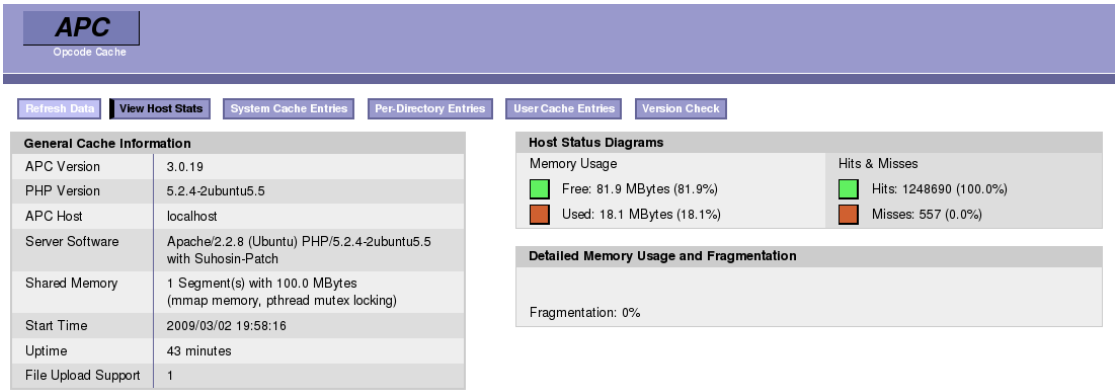

Figure 1: Status of apc generated by apc.php

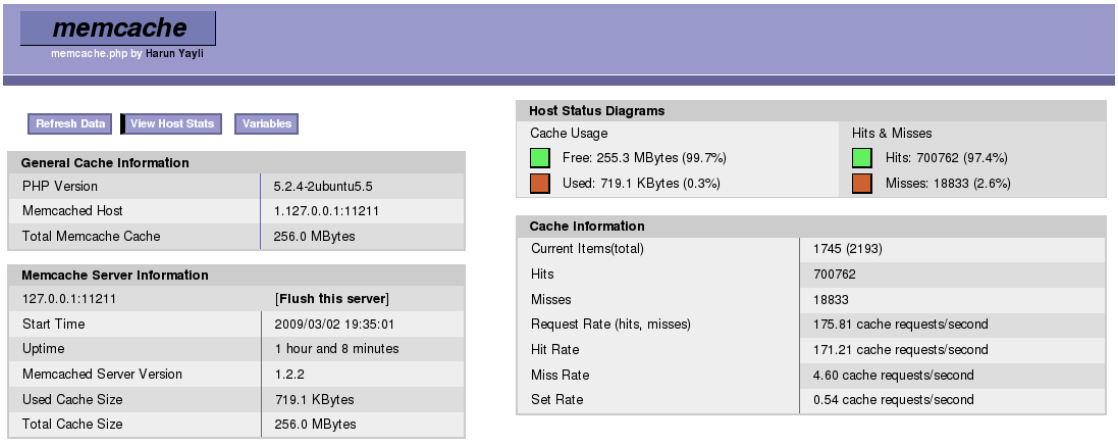

Figure 2: Status of memcached generated by memcache.php

References

[1] Website of memcached. `http://www.danga.com/memcached/`.

[2] Download memcache.php. `http://livebookmark.net/journal/2008/05/21/memcachephp-stats-like-apcphp/`.

[3] Php caches and accelerators. `http://www.sugarcrm.com/wiki/index.php?title=PHP_Caches_and_Accelerators`.

[4] Apache Documentation on mod_expires. http://httpd.apache.org/docs/2.0/mod/mod_expires.html.

[5] Performance tweaks for large systems. http://www.sugarcrm.com/wiki/index.php?title=Performance_Tweaks_for_Large_Systems.

[6] Download Badboy Software. http://badboy.com.au/download/add.