# Speed Matters:

## hints to speed things up

**Andrej Arn**
**Sam Blume**

blueshoes

HANDLE MORE REQUESTS, SHORTEN TIME-TO-CLIENT, WASTE LESS BANDWIDTH.

1. **Install a PHP-accelerator (also referred to as PHP-cache)**

   Whenever a request to a PHP page is made, PHP needs to fetch all source code, parse and compile it, and finally run it. For a hello-world page that is no big deal. But when you use some ***BS classes***, which again use some classes (etc) there are some thousand lines to work on.
   A PHP-accelerator saves the parsed and compiled code to disk (and/or memory). When your source code changes, which does not happen too often anyway, the cache detects that and rereads, reparses and recompiles that file.
   It's a great thing, and gives you a speed increase of **200 to 1000 percent**. You absolutely want that.
   There are different ones on the market, for an up-to-date list go to
   http://www.blueshoes.org/en/developer/links/#PHPAcceler

2. **Upgrade hardware**

   Upgrading the hardware is usually a cheap solution, and comes somewhat after installing a PHP-accelerator. It may solve all your problems. Not only the RAM matters, CPU is even more important because PHP needs a lot of processing power.

3. **Cache your content**

   There's nothing faster than static HTML. If you can and your content does not change too often, persist your generated html in files. Check out our ***Bs_FileCache*** class, it is able to store generated data and auto-detect changes. For some heavy-traffic sites or CPU-intensive tasks that's a killer feature.
   The BlueShoes CMS has this feature built-in: http://www.blueshoes.org/en/applications/cms/

4. **Don’t serve the same content twice**

   By default for .php files PHP sends the client a header that tells him the content he gets should not be cached in the browser - expire immediately. This default behavior is because it’s assumed that the php page may change any time (as it generates dynamic content).
   But this is not true all the time! Let’s say you have a database. You generate pages using the content of the db. But the db does not change every few seconds or minutes. Why don’t you tell the client to keep the pages for 5 minutes or 2 hours or whatever is appropriate for your page? That really matters, especially because the users always request the same pages again and again (by clicking forward and back, and clicking on the same links).

   Have a look at the if-modified-since header of the HTTP protocol.

   Use our ***Bs_UrlCache.class.php***, set some values, and all the rest is done for you.
   This not only saves CPU but also tons of network traffic. That can be a bottleneck too.

   The BlueShoes CMS has this feature built-in: http://www.blueshoes.org/en/applications/cms/

5. **Update your software**

   You should do that anyway because of security issues. Newer software versions often have speed improvements.

6. **Don’t lose time where you don’t lose time**

   Code optimization may be an option. But before you start tweaking your code, use our ***Bs_StopWatch*** class to see where your program uses all that time. Remember: Only 10% of your code uses 90% of your CPU. The goal is to find those 10%!
   Have a look at the PHP Benchmarks: http://www.blueshoes.org/en/developer/php_bench/
   Loops can be written in many ways, and there are quite some time differences.

7. **There are lots of other tricks.**
   Depending on your site the following may help as well (but does not affect the speed of the ***BlueShoes Framework*** as much as the tips above)
   - put your database on a different machine than your web server
   - serve static content (images, html files) from a different machine than dynamic content.

   Search google for more if you need it.

**Additional reading:**

Compare servers: http://www.serverwatch.com/stypes/compare/