Analysis of Approaches Used to Raise Productivity of PHP Programs

Mykola Alieksieiev, Volodymyr Bondarenko

Abstract – **In this work, the authors discuss methods for raising productivity levels of web-resources on PHP, for decreasing their load conditions on the application servers, and determine their efficiency.**

Keywords – **PHP, productivity.**

I. INTRODUCTION

It is common knowledge that PHP is one of the most popular technologies for the creation of Internet-projects. At the same time, it is quite demanding towards the hardware of application servers. Hence, as the popularity of such web-resource increases with time, there might happen a "critical moment," when the server would stop responding to requests.

II. MAIN PART

There is a popular skeptical opinion on how reasonable and necessary the optimization of PHP code actually is. It is thought that this language can only be used for sites that are not loaded much. Nevertheless, the correct approach to optimization can significantly speed up the work of the Internet resource.

As the load on the server increases, every PHP page will be processed slower. Thus, one can consider the time, needed for PHP to respond to the web server, to be the objective criteria of the processing speed of the system.

To test the processing speed of running the code, it is suggested to choose software product Vulcan Logic Disassembler, which shows the number of actions, carried out by the processor as it processes the program code of web-application. This will allow comparing the studied approaches independently of the processing characteristics of hardware.

During the work, the following main principles of raising PHP productivity were identified:

1) Choice of more productive elements of the program code;
2) Usage of cashing the compiled code;
3) Cashing code that is not compiled;
4) Introduction of multithreaded process;
5) Usage of Zend Optimizer.

The first approach is about the choice of operations, which are processed faster and carry out the same tasks. For example, usage of combinations if-else multiple times, instead of the switch() combination, usage of quotes in the work with the associative arrays ($row['id'] instead of $row[id]), usage of local instead of global variables, cleaning memory upon the completion of work with variables (unset()) etc.

Man steps in the work of PHP code are:

1. User request;
2. Fork (slow operation);
3. Assemblage and compilation (very slow);
4. Running the code;
5. Return of result to the web server.

Second and third approaches of increasing productivity are based on the technology of cashing. Usage of special programs, called accelerators, will allow avoiding second and third steps in the work of PHP code. Accelerators cash byte-code that is already compiled, that will be taken from cash and run right away. One more way of using technology of cashing is using program Memcached. It can be used during the second step (Fork), for example to speed up requests to the database. The program can also be used during the compilation stage to cash the code that is not compiled. The drawback of this approach to optimization is the necessity to add additional lines of code into PHP pages, while accelerometers do not need that. Introduction of threads into PHP code is a very effective method, however it is only reasonable with tasks that allow parallel algorithms and the application range of this method is very small [2]. A one more approach is usage of server module to launch coded with Zend Encoder and Zend SafeGuard Suite scipts - Zend Optimizer. Although the developer claimed productivity increase additional 40%, such results were not received in the research during this work.

III. CONCLUSION

Summary of results of using each approach are in Table 1.:

№	Approach	Ave. productivity growth
1	Using the more productive constructions of PHP code	Up to 10%
2	Using cashing of compiled code (accelerators)	Up to 300%
3	Cashing of not compiled code (Memcached)	Up to 300%
4	Multithreading	Up to 50%
5	Zend Optimizer	Up to 20%

We can think that the most preferred way to raise PHP productivity, which would not require recoding, is the usage of accelerators. However, for best results, it is important to combine all the discussed methods.

REFERENCES

[1] M. Ternovoy, O. Shtogrina, "Approach to data processing in information-telecommunication environment of complex administrative systems" in 19th International Crimean Conference "Microwave & Telecommunication Technology. CriMiCo 2009", 2009, pp. 358 – 359

[2] L.S. Globa, R.L. Novogrudska, "Informational and computational resourses of Internet-portal «strength of materials» integration" in 21th International Crimean Conference "Microwave & Telecommunication Technology. CriMiCo 2011", 2011, pp. 485-486