rezaminy@gmail.com

  P465448

performance report

Contents

[Abstract 2](#_Toc44152103)

[Definition: 2](#_Toc44152104)

[Code Optimization Importance 2](#_Toc44152105)

[PHP Performance Optimization: 2](#_Toc44152106)

[Where to apply 4](#_Toc44152107)

[Phases of Optimization 4](#_Toc44152108)

[Performance Monitoring Tools 4](#_Toc44152109)

[Conclusion 6](#_Toc44152110)

[References 6](#_Toc44152111)

# Abstract

Application proper performance is an important part of application development where will affect the user’s satisfaction which can make a business successful or take it down. Application performance can be improved by implementing code optimisers and performance tools.

# Definition:

To improve the [code quality](http://www.viva64.com/en/t/0077/) and efficiency it is necessary to optimize the code as much as possible to ensure the best speed and experience for users.

Code optimization is a vital concept in application performance and can make a big difference in application performance when compiling.

Code optimization can be defined as any method of code modification to improve [code quality](http://www.viva64.com/en/t/0077/) and efficiency. A program may be optimized so that it becomes a smaller size, consumes less memory, executes more rapidly, or performs fewer input/output operations.

The basic requirements optimization methods should comply with, is that an optimized program must have the same output and side effects as its non-optimized version. This requirement, however, may be ignored in the case that the benefit from optimization, is estimated to be more important than probable consequences of a change in the program behaviour. (Viva64, 2020)

# Code Optimization Importance

There are many reasons for performance importance and the main ones can be:

* User experience

Loading and running time of the application are the important issues which can easily affect the number of users in using your app or your competitors.

* Conversions

Conversions are the number of users who actually buy or download your product or generally the users who give money for your services. A faster website means more conversions and more profit

* Scalability

The more requests your application can handle per second, the more traffic you can handle. If your application can handle a single process in 100ms, it means your application can serve 10 requests per second so if you reduce the application processing time to the half, your application capacity will double. (Aladdin, 2020)

* Memory usage

Reduces the space consumed and increases the speed of compilation.

# PHP Performance Optimization:

There are two types of optimization:

* Machine Independent Optimization which attempts to improve the intermediate code to get a better target code as the output. The part of the intermediate code which is transformed here does not involve any CPU registers or absolute memory locations.
* Machine Dependent Optimization which is done after the target code has been generated and when the code is transformed according to the target machine architecture. It involves CPU registers and may have absolute memory references rather than relative references. Machine-dependent optimizers put efforts to take maximum advantage of the memory hierarchy. (Singh, 2020)

There are some good practices in optimizing php performance including but not limited to:

* Choose the Right Version

Well, this one is obvious especially after PHP7 which have the best performance among the old PHP versions. I will not make a comparison here between PHP versions since there are so many articles discussed that, but you still can take a look at the following chart showing the performance of PHP versions run by different CMSs.

* PHP Micro-Optimization

Micro-Optimization is the minor changes in your code that improve your application performance. For example, if you are going to use a for-loop, it is always better to calculate the length in advance. The next image shows the result of for-loop with 1000 keys with 1-byte values are given. You can notice the improvement in the execution time.

* XHProf**:** Profiling PHP Code

After we have seen the limited value of PHP Micro-Optimization, it is time to determine precisely which part of our code is slow without the need to guess using a profiling tool called XHProf. After installing and configuring XHProf on your server, XHProf will append header and footer to all your PHP scripts and generate a report where you can find all the executed functions, the execution time, and the number of calls of each function. (Aladdin, 2020)

Code Optimization is done in the following different ways:

* Compile Time Evaluation
* Variable Propagation
* Dead code elimination:

Variable propagation often leads to making assignment statement into dead code

* Code Motion:  
  • Reduce the evaluation frequency of expression.  
  • Bring loop invariant statements out of the loop.
* Induction Variable and Strength Reduction:  
  • An induction variable is used in loop for the following kind of assignment i = i + constant.  
  • Strength reduction means replacing the high strength operator by the low strength.

Where to apply  
Optimization can be applied in the following stages of coding:

* Source program  
  Optimizing the source program involves making changes to the algorithm or changing the loop structures. User is the actor here.
* IntermediateCodeOptimizing the intermediate code involves changing the address calculations and transforming the procedure calls involved. Here compiler is the actor.
* TargetCodeOptimizing the target code is done by the compiler. Usage of registers, select and move instructions is part of optimization involved in the target code. (Singh, 2020)

Phases of Optimization  
There are generally two phases of optimization:

* GlobalOptimization**:**Transformations are applied to large program segments that includes functions, procedures and loops.
* LocalOptimization**:**  
  Transformations are applied to small blocks of statements. The local optimization is done prior to global optimization. (wikipedia, 2020)

# Performance Monitoring Tools

Here are some performance monitoring tools you may want to consider in building your PHP application.

* PHP code profilers:

There are two types of code profilers: standard and tracing profilers which are widely used by the web developers to improve their performance or to determine the weak points.

Standard profilers periodically record stack traces of your application, while tracing profilers are lighter-weight than standard profiles and can be used while you code.

Standard code profilers give you a snapshot of important metrics like CPU, memory usage, time spent per line of code, and frequency of method calls.

Tracing profilers have quite an advantage from standard as you can use it every day, and it won’t slow you down. It is designed to catch errors in real time while you’re developing, instead of catching errors after the fact. Tracing profilers are considered as your first line of defence in dealing with bugs.

* PHP application performance management (APM) tools

Tracking web application behaviours in certain scenarios is so important so that developers can optimize parts of the application that might be underperforming or which part of a web application slows down the entire application. In most cases, performance bottlenecks happen during database queries and API calls because the application has to wait for these processes to complete before moving on to the next task. Retrace is one of the tools that can be used in performance monitoring. Here are some screen captures on how Retrace works.

Based on the data gathered by Retrace, this web application has multiple inserts and select queries that put a lot of load on the server. By analysing these insert and select queries, the developers were able to pinpoint what’s causing the multiple inserts and select database queries. A possible solution like applying left joins in select queries also improved the performance of the web application.

* Real user monitoring (RUM) tools

Real user monitoring (RUM) such as [Google Analytics](http://www.google.com/analytics/), relies heavily on services that constantly measure the application in the background. This is done by a small JavaScript in a webpage that will collect some data as the user browses the web application. One of the main uses of RUM is to record which pages are being viewed by the users and how long they stay on each one. This kind of data is useful to decide where to put ads or where to put information for maximum viewability. The data gathered from RUM can also be used to improve the user experience of a website or application.

* Web server access logs

For Retrace to be able to monitor a web application, the time, request path, status, and log event date must be present in the web server’s access logs. These parameters, if not present in the logs, can be enabled in the configuration of the web server. Using the data collected from the access logs, Retrace is able to formulate the requests per minute, average load time, and HTTP error request data of the web application.

* Exception tracking

PHP, like other programming languages, handles exceptions pretty well. Exceptions are very important to track because they usually stop the execution of a system. But if the system is throwing exceptions in the back-end, e.g. web services, then it’s hard to keep track of these exceptions. (MENDEZ, 2018)

# Conclusion

In conclusion, application performance is an important aspect of coding and need to be optimized to ensure customer satisfaction and a wider market. Users need a faster and more reliable site which can provide them with the best services. Meanwhile we should consider optimizing the code using specialized utilities whenever possible.

# References

Aladdin, M. (2020). *PHP Performance Optimization*. Retrieved from https://codeburst.io: https://codeburst.io/php-performance-optimization-992acaa78817

MENDEZ, J. (2018). *PHP Performance: Five Types of Tools You Need To Know*. Retrieved from https://stackify.com: https://stackify.com/php-performance-tool-types/#:~:text=%20PHP%20Performance%3A%20Five%20Types%20of%20Tools%20You,web%20application%2C%20the%20time%2C%20request%20path%2C...%20More%20

Singh, P. (2020). *Code Optimization in Compiler Design*. Retrieved from https://www.geeksforgeeks.org: https://www.geeksforgeeks.org/code-optimization-in-compiler-design/

Viva64. (2020). *Code Optimization*. Retrieved from https://www.viva64.com: https://www.viva64.com/en/t/0084/

wikipedia. (2020). *Optimizing\_compiler* . Retrieved from https://en.wikipedia.org: https://en.wikipedia.org/wiki/Optimizing\_compiler