INTRODUCTION TO PHP

BY: BASSAM BOKHOWA

OPEN UNIVERSITY MALAYSIA

INTRODUCTION TO PHP

BY

BASSAM BOKHOWA

Open University Malaysia

2008

November, 2008

ABSTRACT

"Introduction to PHP"

By: Bassam Bokhowa

Supervised by: Mr. Nazar Marouf

Introducing PHP as a web application development technology, in terms of its uses, advantages, the competing technologies and how it compares in contrast with them.

DEDICATION

To the architects of change in spite of those who fear it.

ACKNOWLEDGEMENTS

I would like to heartily acknowledge:

- My supervisor; Mr. Nazar Marouf for his generous guidance

- Ms. Jennifer J. Harbster, Digital Reference Specialist at the Library of Congress for help with Internet libraries

- Khalid Al-Baloushi and Nader Al-Shurooqi from the Bahrain Linux User Group (http://www.reblug.com) for their friendship.

TABLE OF CONTENTS

LIST OF FIGURES

GLOSSARY

GNU: a project aimed at creating a free, open source operating system, similar in functionality to the commercial UNIX operating system. It has achieved its aim in 1992 (Free Software Foundation, Inc., 2008).

HTML: HyperText Markup Language is an open standard for writing richly formatted documents which are able to link text phrases to other similar documents located on a network such as the Internet.

Open Source: a licensing scheme under which free software programs make their source code available for anyone to use, alter, and redistribute without charge (Welling, et al., 2005).

Source code: the original programming language code that a programmer writes and later compiles in order to create an executable software program.

CHAPTERS

1. BACKGROUND

1.1 What is PHP?

PHP is the most widely-used, open source, general-purpose, server-side scripting language (TIOBE Software, 2008), PHP is especially suited for web development and can be embedded into HTML (The PHP Group, 2008). PHP primarily acts as a filter taking input from a file or stream containing text and/or PHP instructions and outputs another stream of data; most commonly, the output will be HTML. The main goal of the language is to allow web developers to write dynamically generated web pages quickly (The PHP Group, 2008). PHP is a server-side scripting language. Meaning, PHP code will be executed and interpreted on the web server each time a user visits the page containing the code. HTML code results from the execution and is viewed by the visitor (Welling, et al., 2005).

1.2 Why is it called 'PHP'?

PHP originally was named PHP/FI which stood for 'Personal Home Page/ Forms Interpreter'. As its scope widened, it was later changed in line with the GNU project's recursive naming convention (GNU=Gnu's Not Unix) (Welling, et al., 2005) and now stands for 'PHP: Hypertext Preprocessor' (The PHP Group, 2008).

1.3 What is a scripting language?

Unlike traditional programming languages such as C and C++, scripting languages such as PHP do not have to be compiled. Instead, the source code is interpreted at runtime using a 'runtime interpreter' on the server (e.g. the Zend Engine) that compiles the code in real time. Interpreted code usually has slower performance in comparison with compiled code. PHP may at times be referred to as a 'dynamic programming language' (Zend Technologies Inc., 2007).

1.4 Beginnings of PHP

PHP was conceived in 1995 and was originally the work of Danish programmer Rasmus Lerdorf who had originally called it 'PHP/FI' and wrote it as Perl scripts.

He later added more functions and implemented it in C and released it as Open Source software. It was officially released in 1997 as 'PHP/FI 2.0'. It was reportedly installed on 50,000 domains and many programmers started contributing to its code.

1.5 PHP version 3

Officially released as in 1998 by two Israelis; Andi Gutmans and Zeev Suraski, as a complete re-write of PHP/FI and renamed 'PHP' but stayed as Open Source software. It was meant for developing an eCommerce

application they were working on for a University project. It was very successful for its extensibility features which attracted dozens of developers to join in and submit new extension modules.

1.6 PHP version 4

Officially released by Andi Gutmans and Zeev Suraski's company; 'Zend technologies' in 2000 as open source software. PHP 4 introduced the more capable Zend Engine for handling complex applications efficiently (The PHP Group, 2008). Zend Technologies is a commercial entity, which funds its open source development by offering support and related software on a commercial basis (Welling, et al., 2005).

1.7 PHP version 5

The newest version of PHP - PHP 5 - integrates Zend Engine 2, which is a complete rewrite of the previous version. It was released in 2004 and has introduced major changes such as:

1. Greatly improved PHP performance and capabilities
2. Better object-oriented support built around a new object model
3. Scalable, maintainable error-handling through 'exceptions'.
4. SimpleXML to handle connecting to web services through XML data.
5. Added the SQLite database (Welling, et al., 2005)

Unfortunately, PHP 5 could also break some PHP 4-specific code (The PHP Group, 2008) which has caused a slower upgrade momentum (Meer, 2006).

1.8 PHP related projects

As well as the major progress seen through the release of PHP version 4, the year 2000 also witnessed many other important projects, which stemmed in order to assist or extend the PHP technology:

1. **PEAR (PHP Extension and Application Repository)** originally, PHP Extension and Add-on Repository, PEAR is PHP's version of foundation classes, and may grow in the future to be one of the key ways to distribute PHP extensions among developers. Since early 2000, PEAR has grown to be a big, significant project with a large number of developers working on implementing common, reusable functionality for the benefit of the entire PHP community. PEAR today includes a wide variety of infrastructure foundation classes for database access, content caching, mathematical calculations, eCommerce and much more.
2. **The PHP Quality Assurance Initiative** was setup in the summer of 2000 in response to criticism that PHP releases were not being tested well enough for production environments. The team now consists of a core group of developers with a good understanding of the PHP code base. These developers spend a lot of their time

3. **PHP-GTK** is the PHP solution for writing client side GUI applications using the Gtk+ toolkit. It uses a PHP code generator program that reads a set of .defs files containing the Gtk+ classes, constants, and methods information and generates C code that interfaces PHP with them. What cannot be generated automatically can be written by hand in .overrides files (The PHP Group, 2008).

2. PHP SYNTAX

PHP syntax draws upon C, Java, and Perl, and is easy to learn (The PHP Group, 2008). What distinguishes PHP from client-side scripting like JavaScript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was.

A PHP file normally contains HTML tags, just like an HTML file, and some PHP scripting code. The PHP file is saved with the extension '.php' on a web server that has a PHP parser configured.

Below is an example of a simple PHP script which assigns the string "Hello World" to a variable called $txt, then sends the variable's content to the browser:

```
//This is a single-line comment
<html>
<body>
<?php $txt="Hello World";
   echo $txt;
?>
</body>
 </html>
/* A comment block here */
```

A PHP scripting block always starts with **'<?php**' and ends with '**?>**'. A PHP scripting block can be placed anywhere in the document and allows to jump into and out of "PHP mode".

All variables in PHP start with a $ sign symbol. Variables may contain strings, numbers, or arrays.

Each code line in PHP must end with a semicolon. The semicolon is a separator and is used to distinguish one set of instructions from another (The PHP Group, 2008).

3. STRENGTHS OF PHP

- Free-of-cost solution with wide support and documentation options.
- Huge number of full-blown solutions and ready-made code snippets for a variety of applications that are available under open source license.
- PHP can be used on all major operating systems and most web servers.
- Choice of using procedural programming or object oriented programming, or a mixture of both.
- PHP is extremely simple for a newcomer to start writing simple scripts in a few hours, even less if accustomed to C++ or Java. Yet PHP offers many advanced features to professional programmers.
- Efficient use of server hardware to serve millions of user requests per day.
- Built-in libraries of functions allow few lines of code to generate many useful web-related tasks.
- PHP supports a wide range of relational databases, currently 20+ of the most famous, plus the ODBC (Open DataBase Connection) standard.
- PHP's outputting abilities are not limited to HTML. They include images, PDF files and even Flash movies generated on the fly. PHP can auto-generate any text files, such as XHTML and any other XML

file to save them in the file system, instead of printing them out, forming a server-side cache for dynamic content.

- PHP talks to other services using their native protocols, and supports data exchange between virtually all Web programming languages including Java objects.
- Text processing features, including XML parsing.
- Many other interesting functions through extensions.
- Open source gives flexibility for future modifications by user communities (Welling, et al., 2005) (The PHP Group, 2008).

4. PHP DRAWBACKS

4.1 Performance

Unlike traditional programming languages such as C and C++, scripting languages such as PHP do not have to be compiled. Instead, the source code is interpreted at runtime using a 'runtime interpreter' on the server (e.g. the Zend Engine) that compiles the code in real time. Interpreted code usually has slower performance in comparison with compiled code (Zend Technologies Inc., 2007). Benchmarks do not differ on this fact (Lai, 2008) (Nene, 2008)

4.2 Security

The proportion of insecure software written in PHP, out of the total of all common software vulnerabilities is quite large. It amounted to:

12% in 2003, 20% in 2004, 28% in 2005, 43% in 2006, 36% in 2007, and 35.6% for the first half of 2008.

98.5% of these software vulnerabilities can be exploited remotely.

The most common vulnerabilities (81.9%) are caused by not following best practice programming rules and vulnerabilities related to software written in older PHP versions (Coelho, 2008) (Suraski, 2006).

5. PHP USAGE

PHP is used by significantly high hit-rate web sites such as Wordpress.com (Wordpres.org), the user-interface for Facebook (Shire, 2007), Wikipedia (MediaWiki, 2008), and Yahoo! (Naraine, 2002).

Packages combining and easing the deployment of standard web technologies (e.g. EasyPHP, LAMP, WAMP, or XAMPP) have become popular methods in the web industry for easy and fast deployment of web applications. PHP is most commonly used as the '**P**' in such bundles alongside the **L**inux (or Windows) operating system, **A**pache web server, and **M**ySQL relational DBMS.

PHP usage was tracked by Netcraft and published on 'php.net/usage.php' at 20,917,850 domain installations as of April 2007.

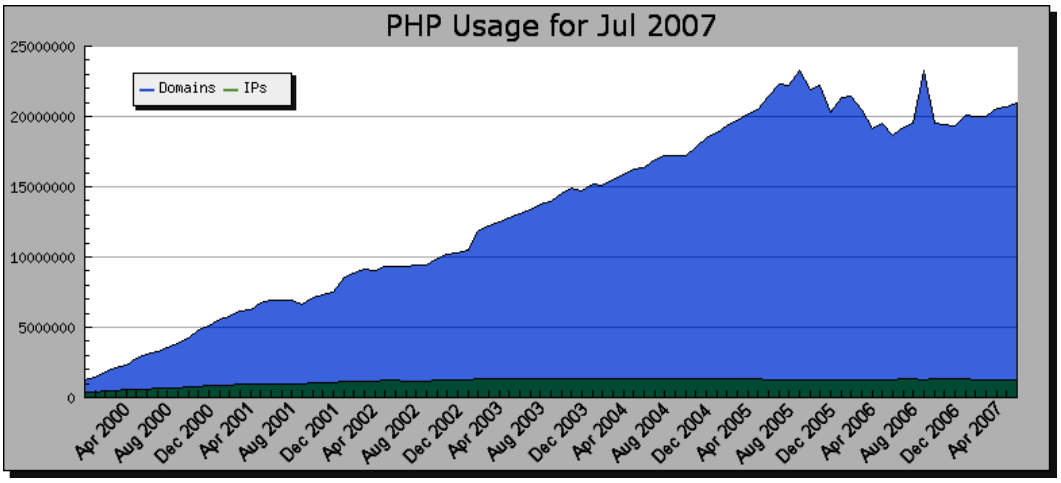

Figure 1: PHP usage (in domains)

PHP is found on 33,15 % of web sites (Nexen.net, 2008). Version 4 is still dominant, but PHP 5 reached 45.12%. All other major versions are negligible (Nexen.net, 2008).

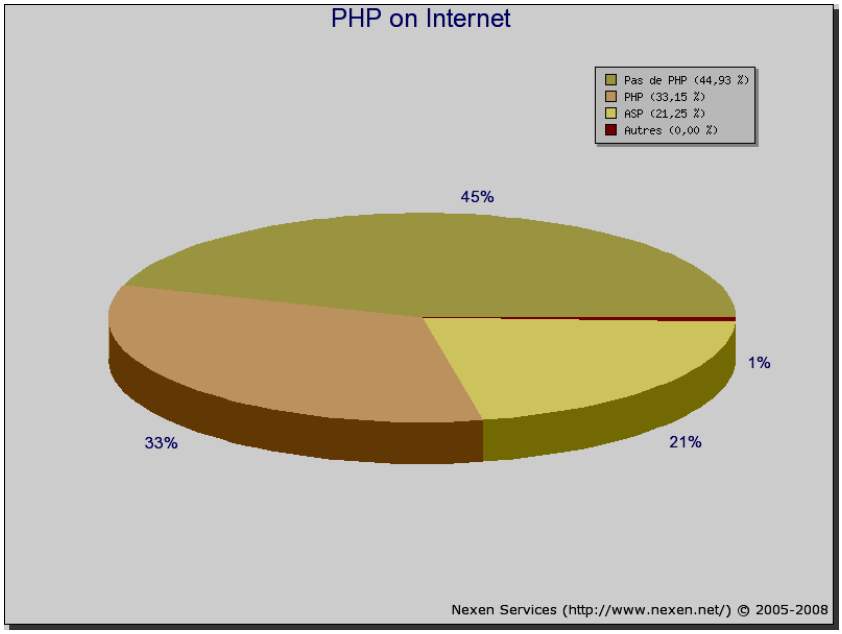

Figure 2: PHP market share, on worldwide internet web sites.

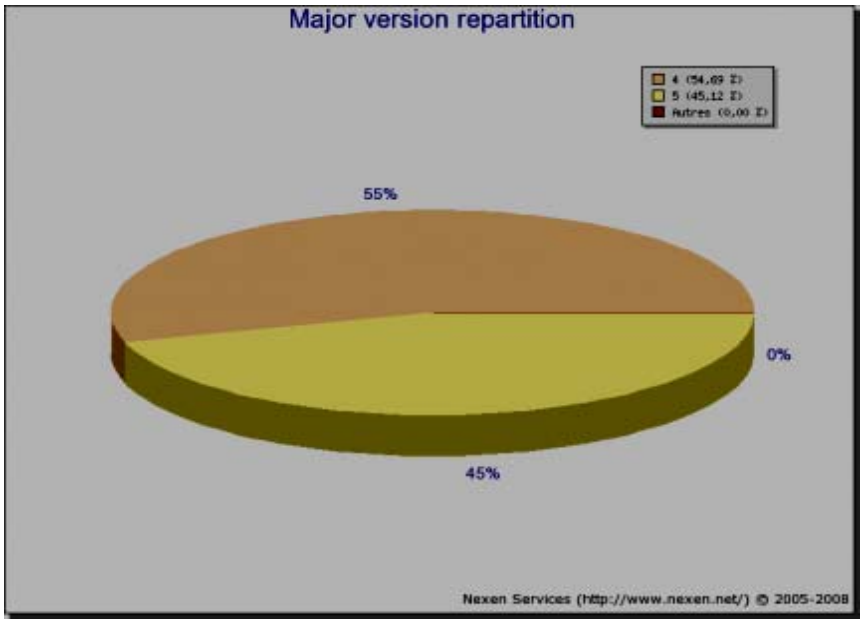

Figure 3: PHP major version distribution

6. COMPETING TECHNOLOGIES

Comparisons with a few other server-side scripting languages follow:

6.1 ASP.NET

ASP.NET is not really a language in itself, it's an acronym for Active Server Pages, the actual language used to program ASP.NET with can be C#, J#, VB.NET, and other.

The biggest drawbacks of ASP.NET are:

1. It's a proprietary system that is natively used only on Microsoft Internet Information Server (IIS). This limits it's availability to purchased Windows Server software.
2. ASP.NET code is more cumbersome to use and complicated to read than PHP code which is smaller and simpler.
3. Limited database connectivity support in comparison to PHP.
4. Resulting runtime code is often slower than PHP.

Some of the strengths of ASP.NET are:

1. It's relatively easy to pick up the languages as they are very common.
2. The languages are more robust than PHP as they are full-fledged programming languages.
3. Object-oriented support in ASP.NET is very capable.

4. The available development tools are more capable than PHP tools to produce large applications rapidly (BizFive.com, 2007).

6.2 Cold Fusion

Drawbacks include:

1. PHP is commonly said to be faster and more efficient for complex programming tasks and trying out new ideas than is Cold Fusion.
2. PHP is generally referred to as more stable and less resource intensive as well.
3. PHP runs on almost every platform there is; while Cold Fusion is only available on Windows, Solaris, Linux and HP/UX.

As for strengths:

1. Cold Fusion has better error handling, database abstraction, and date parsing, although database abstraction is addressed in PHP4.
2. Another of Cold Fusion's strengths is its excellent search engine, though it has been noted that a search engine is not something that should be included in a web scripting language.
3. Cold Fusion has a good IDE and is generally easier to get started with, whereas PHP initially requires more programming knowledge.
4. Cold Fusion is designed with non-programmers in mind, while PHP is focused on programmers.

6.3 Perl

Drawbacks include:

1. The biggest advantage of PHP over Perl is that PHP was designed for scripting for the web where Perl was designed to do a lot more and can because of this get very complicated.
2. The flexibility/complexity of Perl makes it easier to write code that another author/coder would have a hard time reading. PHP has a less confusing and stricter format without losing flexibility.
3. PHP is easier to integrate into existing HTML than Perl. PHP has pretty much all the 'good' functionality of Perl: constructs, syntax, and so on, without making it as complicated as Perl can be.

Perl is a very tried and true language, it's been around since the late eighties, but PHP is maturing very quickly (The PHP Group, 2008).

7. FUTURE ROADMAP

PHP 6, which is still under active development at the time of this writing, can be downloaded as a developer version. It will introduce:

- The Zend Engine 3 (ZE3), which brings full Unicode support to the language (The PHP Group, 2008).
- Removal of features and functions considered broken or a potential security risk.
- Web 2.0 functions such as SOAP and XML protocols will be enabled by default
- Namespaces will be made available to avoid name collisions between functions and classes without using prefixes in naming conventions that make the names of methods and classes unreadable (Good, 2008)
- An alternative PHP caching mechanism (higher performance) which won't be enabled by default.

Some analysts have pointed that with the above changes, version 6 will be more of a code cleanup rather than a feature update.

Also, with the slow developer adoption experienced with PHP 5, it's not reasonable to expect more from version 6 (Meer, 2006).

BIBLIOGRAPHY

Free Software Foundation, Inc. 2008. Overview of the GNU System. *The GNU Operating System.* [Online] Free Software Foundation, Inc., October 7, 2008. [Cited: November 4, 2008.] http://www.gnu.org/gnu/gnu-history.html.

Good, Nathan A. 2008. The future of PHP. *IBM Web site.* [Online] IBM, May 6, 2008. [Cited: November 3, 2008.] http://www.ibm.com/developerworks/opensource/library/os-php-future/.

Lai, Hongli. 2008. Ruby vs PHP performance. *Hongli Lai Blog.* [Online] Hongli Lai Blog, January 17, 2008. [Cited: November 14, 2008.] http://izumi.plan99.net/blog/index.php/2008/01/17/ruby-vs-php-performance/.

MediaWiki. 2008. Manual:Parameters to index.php. *MediaWiki website.* [Online] MediaWiki, September 30, 2008. [Cited: November 4, 2008.] http://www.mediawiki.org/wiki/Index.php.

Meer, Jeroen van der. 2006. Taking a look at PHP 6. *Personal homepage of Jeroen van der Meer.* [Online] Meer, Jeroen van der, March 12, 2006. [Cited: November 3, 2008.] http://jero.net/articles/php6.

Naraine, Ryan. 2002. Yahoo Goes PHP in Open Source Embrace. *Internetnews.com.* [Online] Internetnews.com, October 30, 2002. [Cited:

November 4, 2008.] http://www.internetnews.com/dev-news/article.php/1491221.

Nene, Dhananjay. 2008. Performance Comparison - C++ / Java / Python / Ruby/ Jython / JRuby / Groovy. *Dhananjay Nene Blog.* [Online] Dhananjay Nene, July 8, 2008. [Cited: November 14, 2008.] http://blog.dhananjaynene.com/2008/07/performance-comparison-c-java-python-ruby-jython-jruby-groovy/.

Nexen.net. 2008. PHP statistics for August 2008. *Nexen.net.* [Online] Nexen.net, October 9, 2008. [Cited: November 4, 2008.] http://www.nexen.net/chiffres_cles/phpversion/18780-php_statistics_for_august_2008.php.

Shire, Brian. 2007. The Facebook Blog. *Facebook.com.* [Online] Facebook, May 3, 2007. [Cited: November 4, 2008.] http://blog.facebook.com/blog.php?post=2356432130.

The PHP Group. 2008. PHP Manual. *php.net.* [Online] The PHP Group, October 31, 2008. [Cited: November 3, 2008.] http://php.net/manual.

—. 2008. *PHP Manual.* [CHM file] s.l. : The PHP Group, October 31, 2008.

TIOBE Software. 2008. TIOBE Index. *TIOBE Software.* [Online] TIOBE Software, November 2008. [Cited: November 4, 2008.] http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html.

Welling, Luke and Thomson, Laura. 2005. *PHP and MySQL Web Development.* Indianapolis : Sams Publishing, 2005. 0-672-32672-8.

Zend Technologies Inc. 2007. *An overview on PHP.* [PDF file] Cupertino, California, USA : Zend Technologies Inc., February 7, 2007.