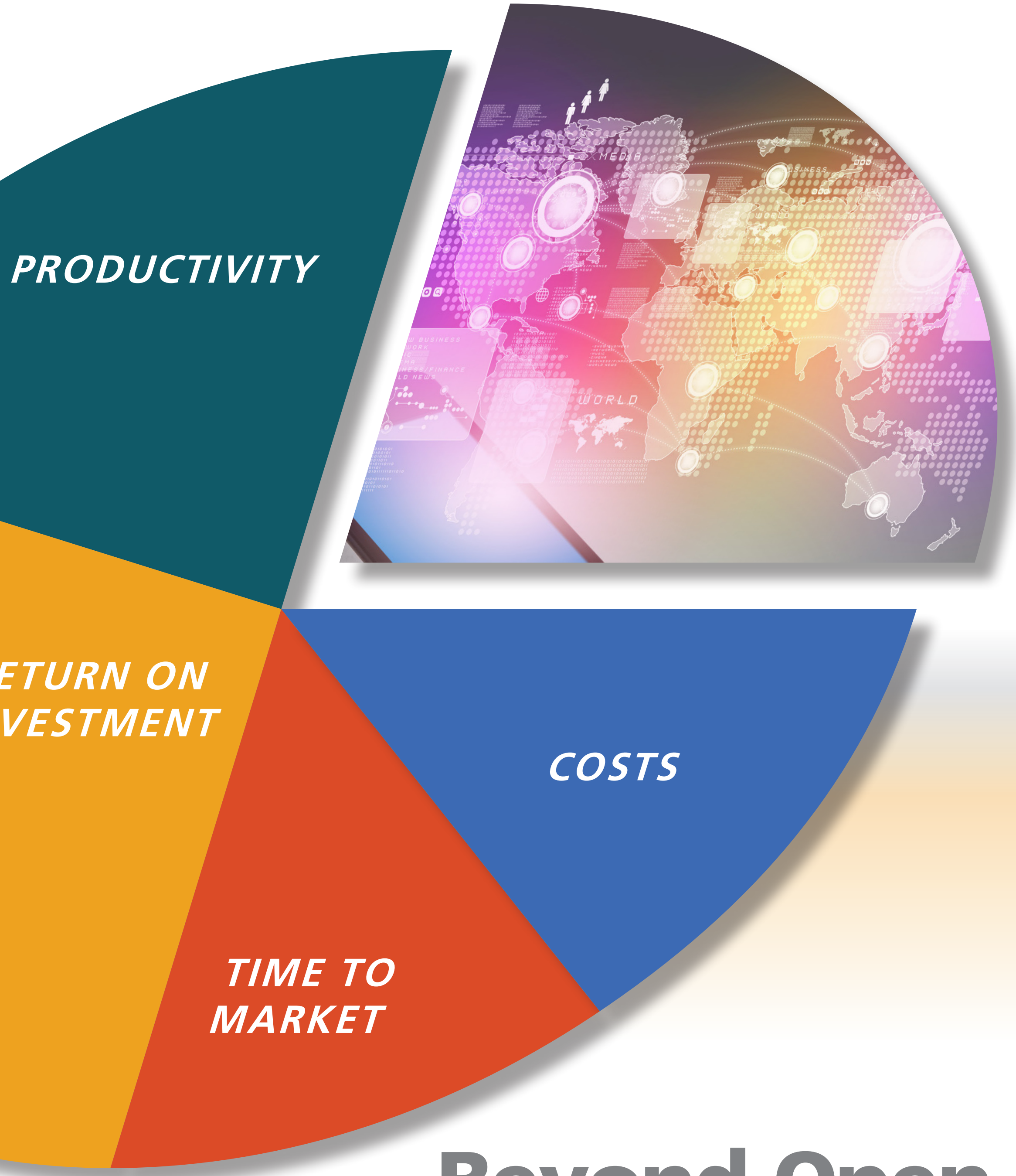
UBM
Tech

A UBM TECH **Business Case Builder**

JULY 2013



# Beyond Open Source

## 4 Key Considerations When Investing in a PHP Platform

By Eric J. Bruno

Sponsored by zend The PHP Company

Unanticipated spikes in application usage at inopportune times can have a large impact on the success of an application, and ultimately on the company that depends on it. Companies need to take stock of whether their applications and the platforms they run on can scale to meet demand, and they're asking themselves whether it's time to invest in a commercial PHP application platform.

### *Open Source: An Excellent Start*

More than 240 million websites, including those of ADP, Facebook, NYSE Euronext and Toyota, are written largely in PHP. This raises the question: Why do so many companies choose PHP when building Web and mobile applications? First, PHP has cost benefits as an open source development language. Secondly, when combined with a development framework or platform, PHP helps developers quickly and efficiently build rich Internet applications, addressing a wide variety of application needs. In fact, when comparing the job trend data of PHP to other scripting languages, PHP is one of the top three most popular Web languages in use today, well ahead of Python and Ruby.[1]

**240 million**

**Number of websites that are written largely in PHP.**

When evaluating environments, PHP developers have two key considerations: what's required to develop an application, and what's required once that application is in production. For development, developers need a preconfigured, enterprise-class PHP stack that supports native installation on the most popular OS platforms. That stack ideally should also include drivers for popular enterprise databases such as Oracle, IBM DB2, MySQL and Microsoft SQL Server, which will help developers get started quickly. Developers will also want to choose a PHP development platform that boosts productivity with a built-in source-level PHP debugger and profiler capabilities, and a Web services debugger to test SOA infrastructure.

In terms of deployment in an agile process, a PHP platform needs to support rapid releases so that quality isn't sacrificed for speed. Cooperation and collaboration when delivering a quality application are the keys to successful DevOps, including shared infrastructure and visibility across development, test and production.

When it comes to moving an application into production, companies should look for a PHP platform that provides the capability to combine all of an application's artifacts for deployment such as scripts, configuration settings and application metadata, and that comes equipped with a Web-based administrator console that reports on application performance, stability, average load, memory usage and other key metrics. The right platform also should include performance and scalability enablers (such as clustering), and enable developers to queue key processes for offline processing.

### *Putting Customers First*

Clearly, PHP has a lot to offer as a Web development language, but is an open source PHP platform enough? When an organization and its users depend on the performance and uptime of PHP applications, it's time to consider a commercial platform with deep application monitoring, massive scalability and enough elasticity to meet demand; holistic deployment support; high availability clustering; and enterprise-level production support. These capabilities will ensure that an organization will know precisely how its applications are performing, allowing developers to troubleshoot immediately if something goes wrong with an application once in production.

The right commercial PHP platform should provide the ability to cluster servers, easily spin up new server instances and dynamically meet spikes in demand while keeping servers in sync. For true elasticity, all of this should be supported automatically, without human intervention. Further, session clustering needs to be supported to prevent information loss in the event a server fails or suddenly becomes unavailable.

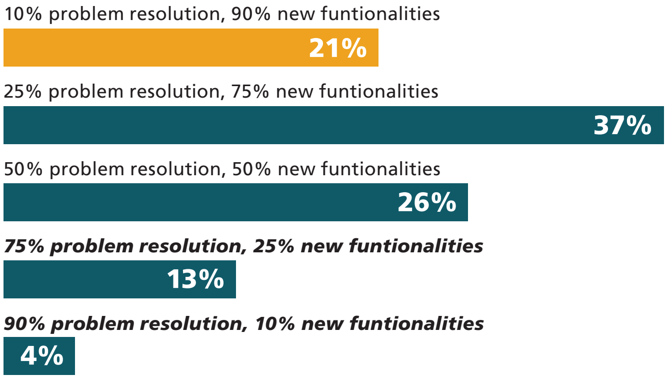
A commercial PHP platform creates value in terms of reliability, security, uptime and support, and it allows developers to focus on more-productive tasks. Here's what to look for in a commercial PHP platform:

### 1. Diagnostics and Problem Resolution

A commercial PHP platform provides all of the benefits of the open source PHP stack, with additional support and diagnostic features. The platform should provide access to a full set of log files, beyond a set time frame. In addition to logs, a production-quality platform will include full diagnostics and deep views into the server's runtime statistics. Event logs with code tracing alert administrators to application problems they may not know about, before it's too late.

According to a recent survey of almost 5,000 developers, a full 43 percent of developers spend more than half their time on problem resolution and maintenance for their apps after they move into production, when they could be developing new features.[2] When developers spend

## How much of your time is spent on production problem resolution and maintenance versus developing new functionalities?



Base:4,809 enterprise, SMB and independent developers worldwide
Source: Zend Technologies, Zend Developer PulseTM: Developers and App Economy 2013

this much time recreating issues instead of resolving them, it's clearly a waste of valuable resources. As a result, choosing a rapid development open source platform for production doesn't necessarily save money if that value is lost because of a lack of rapid troubleshooting tools. In this case, it can put a business and its customers at risk.

### 2. Enterprise PHP Application Support and Security

A commercial platform vendor should provide technical support, security patches, and multiserver support and management. To help resolve issues more efficiently, commercial PHP platforms often provide complete application analytics data when issues arise.

In addition to the latest security patches, a commercial PHP platform should provide user access controls, details of user access and changes. For example, the platform should provide a compliance trail that tracks and reports every change made to an application's configuration, along with who made those changes and when they were made.

User access controls protect access to configurations and application code in production, enabling specified developers to access the diagnostic capability of the platform while also preventing them from being able to directly change application settings on production servers. Many commercial platforms offer integration with Active Directory or LDAP servers for seamless identity management.

### 3. Enhanced Application Deployment and Administration

Companies should consider a platform that supports automated deployment to ensure that applications are deployed correctly each time. This may be offered either through an administrative user interface or an API for further automation and integration with existing continuous integration or application monitoring systems. A production-quality platform should allow developers to create workflows using all application artifacts in addition to PHP code. Companies should also make sure that their PHP platform of choice supports safe rollback, which allows them to immediately revert to the last known correct application version, should an issue arise.

### 4. Meeting Service Level Agreements (SLAs)

A commercial PHP platform that offers high availability with seamless failover ensures that customer SLAs are met with the maximum uptime. Beyond free and open source platforms, a commercial platform helps developers make informed decisions with performance metrics and real-time performance alerts before problems occur.

### *Calculating the ROI of a PHP Application Platform*

To prove the value of a commercial PHP platform, Zend offers customers a return-on-investment (ROI) evaluation/assessment option. This uses a tool with an enhanced spreadsheet that accepts parameters specific to a PHP application, and applies vectors of value for Zend Server (Zend's PHP application platform) in the following areas:

- **Developer productivity**
- **Troubleshooting time**
- **Integration costs**
- **Cost of downtime**
- **Application performance and scalability**

**25%**

**Reduction in performance-related issues realized when using Zend Server.**

For instance, customer experience shows that 30 to 50 percent of a developer's time is spent resolving quality assurance and production issues. Experience also shows that Zend Server can reduce that time by nearly half. Further, results show that Zend Server reduces performance-related issues by at least 25 percent. Zend's ROI tool provides a detailed report of savings in terms of time and dollars for an application. In fact, the company has published case studies[3] in actual customer savings that include the following highlights:

- **"50% reduction in development and testing time by using Zend" — Netretail Holding**
- **"A six-digit dollar cost savings within the first year, while introducing a whole new level of e-commerce functionality" — Villeroy & Boch**
- **"Within three days of our first site launch, Zend helped us optimize performance by more than 200%." — NYSE/Euronext**
- **"The most impactful and visible benefit for end users is that our environment is now highly available." — Marilan**

### *DIY Puts Businesses at Risk*

The value of purchasing a commercial PHP platform is clear. Moving forward, organizations should also keep in mind the potential risks involved with choosing the wrong platform, or choosing to create their own:

- The knowledge, maintenance and enhancement of the PHP stack reside with the developers who implement it. When developers leave or shift roles, they take that knowledge with them. Organizations that opt for open source can sometimes lack the support of an outside entity whose business it is to ensure that their PHP applications continue to perform and scale reliably.
- Organizations may struggle with scaling their PHP applications beyond the initial usage parameters. For example, core PHP doesn't include clustering capabilities. Building and supporting this capability in-house requires developers to provide production integration and support, which can be a distraction.
- The lack of a real-time analysis and troubleshooting capability means organizations may not see an issue coming until it's too late.
- A custom-built platform of core PHP with add-ons can turn into a business liability, since companies can't turn to outside help when problems arise.

The financial risk associated with launching an open source or do-it-yourself (DIY) environment in production can have an impact on customers as well. In fact, the biggest risk of DIY is that it will affect customers if failures occur that require a long time to fix or if application-performance issues go unnoticed. If customers are forced to deal with slow response times or recurring issues, they may abandon an application for a competitor's. The goodwill and confidence lost translates into financial losses beyond just a single application.

Getting applications out the door quickly with a free PHP stack and no supporting application platform may seem like an affordable alternative, but enterprises quickly learn that they need to sustain the right quality of experience for their users for the lifetime of their applications. This is the value that commercial PHP platforms such as Zend Server offer companies, developers and ultimately, customers.

## Zend Server for Your Production Needs

While the free edition of Zend Server provides an ideal starting point for developers, the commercial edition is designed for production deployments.

**Development tools and runtime:** A consistent stack with popular extensions and libraries built in, as well as built-in connectivity to enterprise database technologies

**Support and security patches:** Support for all components of Zend Server, including PHP runtime and Zend Framework, as well as periodic security hot fixes. Zend also provides long-term support for end-of-life versions of PHP and custom fixes to PHP or Zend Framework open source code.

**Diagnostics, monitoring and problem resolution:** All the benefits of the free edition, with additional support and diagnostic features. These include access to full performance and troubleshooting data with longer data retention, safe viewing of server runtime statistics, and application event logs. Additionally, Zend Server provides a detailed view of the application ecosystem with monitoring dashboards. Monitoring rules can be configured to suit the requirements of each application, providing notification of PHP errors, high memory usage and slow request execution.

When something does go wrong, code tracing provides a snapshot of the issue, giving developers function-level insight into the problematic PHP request. Via separate login credentials and read-only access controlled by the application, developers can safely access debugging data in production to resolve problems faster.

**Enterprise integration:** Integration with third-party monitoring tools such as Nagios, as well as with deployment solutions such Hudson or Jenkins. Businesses can use the audit trail to track all user activity indicating what happened, when and by whom. Zend Server's user management also integrates with LDAP directories. Further, applications running on Zend Server have access to Java legacy code via a Java connector.

**Enhanced application deployment and administration:** Streamlined application deployment with a workflow for creating and deploying app packages containing code, scripts, metadata and installation parameters. Applications can be deployed either from the Zend Server UI or via the API, enabling iterative and continuous deployment with a safe rollback feature that reliably reverts all servers to the last-known good version if needed.

Zend Server also ensures configuration consistency across a cluster and monitors all servers within a cluster to trigger alerts on any configuration changes. This centralized configuration management also allows backup and restore of PHP settings, maintaining consistent environments through the entire application life cycle.

**High performance levels:** Deep application performance acceleration to ensure applications respond to users' requests through unanticipated spikes or unusually high sustained loads. Zend Server's Job Queue feature controls how time-consuming tasks are executed to reduce bottlenecks and enhance application throughput.

Page-level caching enhances page load time based on site usage, while bytecode caching ensures that PHP applications will perform consistently well. The caching of user sessions is shared across all Zend Servers, resulting in greater uptime. If a server fails, user sessions are transitioned to other servers transparently. To manage workload, Zend Server allows easy scaling of applications up or down, both on-premises and in the cloud. Zend Server also aggregates monitoring data across all servers in a cluster, providing a simple way to identify and diagnose problems.

zend®server

[1] *Redmonk.com, The Redmonk Programming Language Rankings, http://redmonk.com/sogrady/2013/02/28/language-rankings-1-13/*

[2] *Zend Developer Pulse: Developers and App Economy 2013 survey of more than 4,800 developers worldwide*

[3] *Zend Customer Success Stories, http://www.zend.com/en/resources/case-studies/*

## ABOUT ZEND

zend® The PHP Company

Zend partners with businesses to rapidly deliver modern apps across mobile and cloud. Zend helped establish the PHP language, which today powers over 240 million applications and web sites. Zend's flagship offering, Zend Server, is the leading Application Platform for developing, deploying and managing business-critical applications in PHP. Zend solutions are deployed at more than 40,000 companies, including NYSE Euronext, BNP Paribas, Bell Helicopter, France Telecom and other leading brands worldwide. Learn more at www.zend.com.