

Search...

CHANNELS

HOME
 TOP STORIES
 DATA CENTER NEWS
 COLUMNS
 OPINIONS
 SPECIAL REPORTS
 JOB BOARD
 EVENTS CALENDAR
 RESOURCE CENTER
 WEBINARS
 BLOG
 RSS



ON THE WEB

SITE MAP
 ADVERTISE
 EDITORIAL
 PRIVACY POLICY
 CONTACT US
 REPORT A BUG

Think you
 have enough
 information?
 So did they.



Get the info you
 need with our
 intelligent power
 management
 solution.

Take a Test Drive Now ▶

SPECIAL REPORT

AS OF 7/16/2008 8:54AM EST

Playing Scrabble With the LAMP Stack

By Alex Hardy

July 1, 2008 — LAMP has been shedding light on server architecture for more than 10 years now. Perhaps its greatest strength is the fact that the letters representing Linux, Apache, MySQL and PHP/Perl/Python can all be scrambled, removed, substituted and swapped around. LAMP can be mixed and matched to suit the needs of an enterprise. But with all that potential for variation, how can an IT organization keep track of the ever-shifting components inside their stacks?

The L in LAMP can be replaced by W, S, F, O and sometimes M. After all, Windows administrators have access to Apache, MySQL and PHP in the WAMP stack. FreeBSD users interested in Perl and PostgreSQL might create a FAPP stack. And now, thanks to Sun's purchase of MySQL AB, Solaris users might prefer to use Java, yielding a SAMJ stack.



"M can mean MySQL, or it can mean PostgreSQL," said Ian Murdock, vice president of platforms and operating systems at Sun. "The P can mean Ruby. Why can't the L mean Solaris?"

Amid all these letters and mixes of possible acronyms, there is a single truth of LAMP, LFPP, WAMP, FAPP and all the other possible combinations: With a four-piece stack based on open-source software, the toughest part of the task can be keeping all the pieces working through varying patches and updates.

Companies like SpikeSource offer stack management and support subscriptions to help keep track of the constantly shifting platforms used in most LAMP stacks, but is this a problem that warrants a third-party solution?

Getting AMP'd Up

Will Whitmill, manager of IT at a health care company with more than 50 offices, said that the solution for his company's constantly shifting LAMP stacks is a quick dose of diff. "We use a diff engine and SVN [Subversion] for all our config files, so we can quickly and easily find broken configs," said Whitmill. His company uses Ubuntu-based Linux servers, so updating individual items in the stack is taken care of by the Linux distribution's built-in package management system.

But when one of those packages breaks compatibility with the applications running on top of the stack, Whitmill and his cohorts turn to diff to narrow down the possibilities. Typically, the problem comes from a single update or patch applied to something inside of the stack.

Whitmill said that, typically, the most common issues in a LAMP stack are those specific to each item in the stack. Some software is less likely to throw monkey wrenches into the stack, such as Apache and Perl, while other packages that see more frequent updates, like PHP, can more commonly be the source of errors. In the constructed stack game, a shift in one piece can cause the rest of the tower to sit cock-eyed.

LAMP Shopping

Dominic Sartorio, senior director of product management at SpikeSource, spends much of his time tracking the evolution of LAMP and LAMP-like stacks. SpikeSource monitors and manages LAMP stacks for enterprise customers who don't have the time to keep up with the constant changing of components. He agrees with Whitmill that the biggest challenge facing LAMP users will typically be related to which specific components are used in the stack.

"The different technologies in the LAMP stack each have their own challenges. Linux is pretty mainstream now, so there's a pretty broad set of tools for managing that from companies like Nagios and Hyperic. It becomes more of an issue the higher up the stack you go," said Sartorio.

CLICK HERE to Plug Into...

Colocation with
 10 Mbps of
FREE
 Bandwidth!

866.332.0821
 Hurry...Offer Ends July 31st!



ADVERTISER LINKS

Altova	APC
Avocent	AVTECH Software
Coyote Point	DNSstuff
dtSearch	EventSentry (Netikus)
GroundWork Open Source	Idera
KACE	Lieberman Software
LinMin	Microsoft
PowerGadgets	Raritan
Red Gate Software	Sanblinc
Special Operations Software	SQL Sentry
Sunbelt Software	Symark International
VMware	

SUBSCRIBE TODAY!
Systems Management Week

Email

SUBMIT

CLICK HERE to Plug Into...

Colocation with
 10 Mbps of
FREE
 Bandwidth!

866.332.0821
 Hurry...Offer Ends July 31st!



PDF & PRINT EDITION

"The A is Apache, and there are tools you can use to manage Apache, even virtual instances of Apache," said Sartorio. MySQL, he said, has a console for its database, provided users are subscribed to the company's commercial offerings.

LOADING...

Sartorio said that the best way to be sure your stack will remain solid is to use mature, well-supported pieces.

"Our experience has been wherever you've got a strong commercial entity around an open-source project, that entity will drive the road map and when patches are released. The challenges of, 'Has a certain patch been tested?' don't matter as much there," said Sartorio, because corporate-backed open-source projects tend to have the resources needed to fully test patches before release. "As for Apache, that's a really mature community and they've got their patch system under control."

The P's are the sticky pudding. Sartorio said that PHP can cause a great deal of headaches for users due to its fragmented community and relatively young commercial backers.

"Zend is certainly the strong commercial entity there, but they came along pretty long after the PHP community was established. You can have a lot of things coming out of the PHP community that Zend isn't aware of," said Sartorio. "You have a lot of open-source software, like Drupal, and it's all over the map what patch level of PHP they build on."

Sartorio added that some versions of PHP required in open-source projects aren't supported by either Zend or SpikeSource.

SpikeSource is, at its heart, in the business of LAMP support. The key to the company's ability to keep up with all the updates, patches and changes to each component is found in its information gathering tools, said Sartorio.

"We've got hundreds of components under management. We've thrown automation at the problem. We've got crawlers that go out and see bug databases and patches. A lot of our customers I can envision having a hard time doing that. I don't see an easy way of doing this without throwing some automation at the problem," said Sartorio.

So which pieces of the LAMP puzzle are right for your situation? It all comes down to the needs of the customer. "This is product management 101," said Sartorio. "Interact with your customers and the target users. What's their preferred environment? How many are Linux versus Windows? How many have standardized on a given Web server and database? Make sure you've got your market research done. Develop in the technology environment that will satisfy most of your customers."

When it comes to substitutions within the stack, Sartorio suggests that there are certain components that may become a piece of the stack for purely technical reasons.

"The ones that get swapped out are the ones that are easy to do from a runtime perspective and matter less during development and design time. The Web server is pretty relevant. Sometimes our customers build something, and it'll be tested in Apache, but it'll typically run in IIS unchanged. With the database, as long as you're an application developer and you're using best practices or standards-compliant SQL, then you can pretty easily swap out MySQL, for PostgreSQL or for MS SQL Server," said Sartorio. "P's are hard to swap out," said Sartorio. "If you've developed just for Linux, rarely is it going to run as is in Windows. Then the developers have to get involved."

The Problem With P's

Javier Soltero, CEO of Hyperic, said that those P languages can cause more problems than simply tying applications to a specific operating system. It can also be very difficult to monitor applications running in Perl and PHP.



Javier Soltero

"The challenge there is actually quite daunting. The things that make PHP, Perl and Python very popular are the same things that make it extraordinarily difficult to get the visibility into the runtime," said Soltero. Both Perl and PHP gain a great deal of their usefulness in the LAMP stack from being integrated with Apache. Mod_Perl and Mod_PHP are both built into the Apache Web server, meaning that the runtime for these dynamic languages is as close to the bone as possible.

Unfortunately, this also means that gaining direct metrics from Perl and PHP running in Apache can be tough.

"The fun starts when you get to the P," said Sartorio. "There's PHP, Perl and Python. The runtime environment is usually Apache, so if you have an Apache management tool you can manage PHP and Perl through that. Python, though, has its own runtime. All the best and brightest Python developers out there have been long ago hired away by Google with

golden handcuffs, so that's kind of a challenge area."

The problem, said Soltero, is "visibility into Apache itself. The main technical obstacle for getting visibility into the P languages is that when that's running inside Apache, there's no clear way to distinguish between the regular runtime and Apache itself," said Soltero. "There's similar problems with other languages. But I think Ruby, at least as young a language, lends itself more to internal visibility. [It could eventually have] things you'd be able to get out of a Java or a .NET application. It's somewhat lazy and somewhat irresponsible for the community," not to have that level of visibility into Perl and PHP, said Soltero.

"Out of Apache, you can get plenty of info that's very request-oriented. But because of the Apache process model, you may have 50 or 100 child processes that are consumed. Request statistics aside, what's happening inside something like Perl or PHP, you can't actually get anything that is reasonably useful," said Soltero, who hinted that software that can monitor the actual processes running in Apache-hosted runtimes might be in Hyperic's future.

"Especially in Apache 1.3, you wouldn't be able to distinguish between 100 forked child processes. In order to solve this problem most completely, it would require both the Apache Web server itself, as well as the folks building the runtimes, to No. 1 agree there is a problem, and No. 2 build types of things into the language itself that are difficult to provide this late in the game. In Java you have a series of standard constructs for monitoring. There is no analogy to that in any of the P languages."

Related Search Term(s): [Linux](#), [server management](#), [Windows](#)

[EMAIL THIS ARTICLE](#)

[SEND FEEDBACK](#)

[MORE SPECIAL REPORTS](#)

[SHARETHIS](#)

Digg

Search...

SYSTEMS MANAGEMENT
For IT Sys Admin & Data Center Managers NEWS



Subscribe **FREE** today

LOADING...



HOME	SITE MAP	RSS	CURRENT ISSUE	BACK ISSUES	SUBSCRIBER SERVICES	ABOUT US	CONTACT US	BZ MEDIA
----------------------	--------------------------	---------------------	-------------------------------	-----------------------------	-------------------------------------	--------------------------	----------------------------	--------------------------

Copyright © 2007-2008 BZ Media LLC, all rights reserved. [Privacy & Legal](#)

Phone: +1 (631) 421-4158 • E-mail: info@bzmedia.com