

PETER BAER GALVIN

Pete's all things Sun: AMPing up your Web environment



Peter Baer Galvin is the chief technologist for Corporate Technologies, a premier systems integrator and VAR (www.cptech.com). Before that, Peter was the systems manager for Brown University's Computer Science Department. He has written articles and columns for many publications and is co-author of the *Operating Systems Concepts* and *Applied Operating Systems Concepts* textbooks. As a consultant and trainer, Peter teaches tutorials and gives talks on security and system administration worldwide. Peter blogs at <http://www.galvin.info> and twitters as "PeterGalvin."

pbg@cptech.com

ONE OF THE FRUSTRATIONS FACED BY system administrators (and believe me, there are many frustrations) is the frequent repetition of tasks already performed by other admins. Consider how many times Emacs has been installed in the history of computing, for example. It's frightening how many man-hours are wasted in performing the same computing tasks. And that's ignoring the whole Web-surfing thing.

Over time, many of these repetitive tasks have gotten optimized. In the Emacs example, I used to have to download the source code, compile it, and install the results. I would have to repeat this task (except the download) for any given operating system or computer architecture, and for any given Emacs release that I wanted to use or provide to my users. Now, depending on the tools provided by a given operating system, one `apt-get` or `pkg` command can check if Emacs is currently installed, determine its version number, check Internet repositories for later versions, and, if found, download and install the binaries of the newer version.

While that functionality is astoundingly useful for a given application, we still face other repetitive administration tasks involving application installation. Configuration of applications has no uniformity beyond the fact that there are usually configuration files and command-line options to inspect, modify, test, and implement. Now consider the more daunting and time-consuming task of installing and configuring a set of tools that must work together. Clearly there are "right" and "wrong" configurations, but beyond that there are optimum configurations. Getting the components to work well together, with maximum performance and reliability, is still an exercise in reading manuals, checking forums, using Web search engines, testing, and repeating until relatively happy. Usually these results are actually not optimum, but good enough for a given environment given the amount of time and effort already expended and the diminishing returns of expending more of the same. As DTrace has shown many of us, there are still many performance problems at many, many sites caused by configuration issues (as well as by poor coding).

And let's not forget the frustration of knowing that for a given common set of tools, this task has been performed before and will be performed again by legions of admins.

{L,S}AMP

A common set of tools deployed on Web servers and for Web application development includes Apache, MySQL, and the PHP/Perl/Python programming languages. These open source tools have great features and utility, but can each present installation and production readiness challenges. Implementing AMP, as those three tool sets are collectively known, is done frequently and it frequently requires a lot of effort, regardless of the operating system. Note that even operating systems that ship with some or all of those components provide administration challenges as versions of various components change and as other applications require certain versions of each component.

The AMP stack, as built for Linux (LAMP) and Solaris (SAMP), tries to reduce the admin's work from hours to minutes. It provides a pre-integrated, pre-optimized, and pre-tested set of tools that can turn a standard server into a Web server or a Web development server. Previously, this Sun pre-built, integrated, and tested stack of applications used to be called "coolstack." Sun has renamed it the "Web Stack" because it is no longer coolthreads-specific. (Coolthreads is one of the names given to the lower-power many-thread CPUs Sun ships. They are also known as Niagara or T CPUs.) The current Web Stack also includes GlassFish, Sun's open source application server, and Tomcat. These are optional, and a more limited version of the Web Stack is available without them. The Web Stack home page [1] includes links to videos and download options.

The Web Stack is available in binary form in RPM and SVR4 formats for installation on Solaris 10 SPARC, Solaris 10 x86, OpenSolaris, and Red Hat Enterprise Linux 5 U2. As of this writing, the following components were included in the Solaris x86 version of version 1.4 of the Web Stack from Sun:

- Apache Server 2.2.9
- lighttpd 1.4.19
- Squid 2.6.STABLE17
- MySQL 5.0.67
- PHP 5.2.6
- Ruby 1.8.6
- Python 2.5.2
- memcached 1.2.5
- Apache Tomcat 5.5.27
- GlassFish 9.1.02

AMPed Up

The Web Stack journey starts with a download (after registration) of the appropriate build. For the purposes of this column, I tested the Solaris 10 x86 and OpenSolaris versions. A check of the release notes published in the documentation wiki tells the current version information and any important bug or installation information. Oddly enough, the release notes do not include mention of GlassFish. It's a fairly separate package, in spite of its inclusion. For example, downloading the compressed tarball and untar-ing it reveals an install script to install all of the components, except for GlassFish. GlassFish is installed via its own sjsas script. Clearly this is sub-optimal and, hopefully, it will be better integrated in the future.

Support on the Web Stack is available, if desired, as part of Sun's GlassFish Portfolio offering [2]. The Sun documentation claims that Web Stack will follow a regular release schedule where it is updated, integrated, tested, and released as a bundle for convenient deployment. Certainly the current Web Stack is a good start on this, and a continuation of this effort by Sun would be a boon for all administrators running Web servers and wanting to use this collection of tools. Short of paying for support (or as well as), there is an active Web Stack community with discussion forums [3]. Also, the on-line documentation wiki [4] looks to be quite extensive. It includes "getting started" guides as well as an FAQ and links to the communities associated with each of the packages. Sun of course also sends interested folks to the individual open source project pages of the various packages. There is no magic here in terms of Sun creating a new branch of any of these projects. Rather, Sun has just built, integrated, and tested existing tools.

To install the full set of tools, I first executed `# ./install amp`, and then `# ./sjsas-9_1_02-solaris-i586.bin`. Note that the install tool can also install any one of the packages, if a subset of the tools is desired rather than the full AMP stack.

OpenSolaris is a different kettle of fish altogether. Web Stack releases are being timed to coincide with the distributions of OpenSolaris. For example, a Web Stack version was released at the time of OpenSolaris 2008.5, and another was released concurrently with OpenSolaris 2008.11. This practice is expected to continue.

Rather than the download-unpack-install cycle that was needed for Solaris 10, OpenSolaris installation is done via the new `pkg` mechanism. One command is all it takes: `$ pfexec pkg install amp-dev`. This same command will refresh the Web Stack when a newer version comes out. The `pfexec` command enables privileges associated with the current user. If you are logged in as a user who has the root role, `pfexec` will enable that role and execute `pkg` with root privileges. Though frowned upon, an alternative is to simply become root and execute `# pkg install amp-dev`. There is also a package management GUI that could likewise be used to install the `amp-dev` package. To just install the runtime components of all the AMP packages (say, to run a Web server but not to develop Web software) `$ pfexec pkg install amp` would do the trick.

Actually, a bit more than the `pkg` command is needed for a full, ready-to-use environment. For example, in the OpenSolaris GUI, each user would need to execute Applications > Developer Tools > Web Stack Initialize to initialize the development environment. There are other extra steps needed to start Apache and MySQL. These tasks are all well documented in the Getting Started Guide.

Once again, GlassFish is not tightly integrated into the Web Stack. In fact, a separate sequence is needed to install and configure it on OpenSolaris. The full sequence is documented in a blog [5]. In essence it involves pointing the system at a `pkg` repository that contains GlassFish, doing a `pkg install glassfishv2`, creating an application server domain, and starting GlassFish in that domain.

One added bonus of using the Web Stack on Solaris is the ability to use DTrace to debug problems and optimize performance of your code. The Web Stack tools have DTrace support enabled where possible and appropriate. Once again the Getting Started Guide is handy, showing how to DTrace various aspects of the Web Stack. The Guide also includes the steps needed to transfer a configuration and its contents between systems—for example, when moving a development configuration into production.

In the following example, my version of OpenSolaris was not within the allowed range of the current Web Stack package, so I upgraded my OpenSolaris from release 109 (snv_109 in the /etc/release file) to 110, then installed the amp-dev package, bringing in the entire suite, minus GlassFish. Finally, I installed the GlassFish package.

\$ pfexec pkg install amp-dev

Creating Plan \

pkg: pkg: the following package(s) violated constraints:

Package pkg:/SUNWj6rt@0.5.11,5.11-0.110 conflicts with constraint in installed pkg:/entire:

Pkg SUNWj6rt: Optional min_version: 0.5.11,5.11-0.109 max version: 0.5.11,5.11-0.109 defined by: pkg:/entire

\$ more /etc/release

OpenSolaris 2009.06 snv_109 X86
Copyright 2009 Sun Microsystems, Inc. All Rights Reserved.
Use is subject to license terms.
Assembled 05 March 2009

\$ pfexec pkg image-update

DOWNLOAD	PKGS	FILES	XFER (MB)
Completed			631/631
196.45/196.45			13962/13962

PHASE	ACTIONS
Removal Phase	4957/4957
Install Phase	6370/6370
Update Phase	18110/18110

PHASE	ITEMS
Reading Existing Index	9/9
Indexing Packages	631/631

A clone of opensolaris-4 exists and has been updated and activated.
On the next boot the Boot Environment opensolaris-5 will be mounted on '/'.
Reboot when ready to switch to this updated BE.

NOTE: Please review release notes posted at:
<http://opensolaris.org/os/project/indiana/resources/relnotes/200811/x86/>

\$ pfexec init 6

\$ more /etc/release

OpenSolaris 2009.06 snv_110 X86
Copyright 2009 Sun Microsystems, Inc. All Rights Reserved.
Use is subject to license terms.
Assembled 19 March 2009

\$ pfexec pkg install amp-dev

PHASE	ITEMS
Indexing Packages	631/631

DOWNLOAD	PKGS	FILES	XFER (MB)
Completed			52/52
335.22/335.22			19332/19332

PHASE	ACTIONS
Install Phase	22102/22102
Reading Existing Index	7/7

```

Indexing Packages 52/52
Optimizing Index...
PHASE ITEMS
Indexing Packages 683/683
$ ls /usr/apache2/2.2
bin build include lib libexec man manual
$ pkg install glassfishv2
DOWNLOAD PKGS FILES XFER (MB)
Completed 11/11 6197/6197
110.19/110.19

PHASE ACTIONS
Install Phase 7193/7193
PHASE ITEMS
Reading Existing Index 7/7
Indexing Packages 11/11
$ pfexec pkg contents -m amp-dev
set name=fmri value=pkg:/amp-dev@0.5.11,5.11-0.101:20081210T004605Z
set name=description value="AMP Development cluster"
set name=info.classification
value="org.opensolaris.category.2008:Development/Integrated Development
Environments"
set name=publisher value=dev
depend fmri=SUNWsvn@1.4.3-0.101 type=require
depend fmri=SUNWmysql5@5.0.67-0.101 type=require
depend fmri=netbeans type=require
...

```

For more information on all things SAMP, LAMP, and Web Stack, a good starting place is the BigAdmin page [6].

Conclusion

Installing a full set of the latest Web server and Web service development components can require hours of work, depending on the operating system and its starting status. Getting that set of tools configured and running can increase that time immensely. Getting all of those tools running optimally and keeping them updated with the latest versions can be a never-ending system administration task. Sun's Web Stack is a great help, easing many tasks of such a project. In spite of a few rough edges it is a time and sanity saver.

Random Tidbits

While on a development theme, for those interested in using DTrace for debugging applications under development, I recommend reading the new DLight tutorial [7]. DLight is part of Sun Studio and is a visual DTrace, in essence.

Sun has announced flash storage (SSD) disks available for some of its servers. They have a central Web page [8] for information on the offerings, including an analyzer tool to run on a server to determine if flash would improve performance. I expect flash to be a major driving factor in improving server I/O performance in the next few years.

REFERENCES

- [1] <http://www.sun.com/software/webstack/>.
- [2] http://www.sun.com/software/products/glassfish_portfolio/support.jsp.
- [3] <http://opensolaris.org/os/project/Webstack/>.
- [4] <http://wikis.sun.com/display/WebStack/Sun+Web+Stack+Documentation>.
- [5] http://blogs.sun.com/hyau/entry/opensolaris_is_now_ready_onamazon.
- [6] <http://wikis.sun.com/display/BigAdmin/LAMP-SAMP>.
- [7] <http://sun.systemnews.com/articles/133/4/Solaris/21537>.
- [8] <http://www.sun.com/storage/flash/resources.jsp>.