OpenLDAP | Pacemaker | Watson | Puppet Is

FREE TO SUBSCRIBERS

EPUB, Kindle, Android, iPhone & iPad editions

LINUX JOURNAL™

Since 1994: The Original Magazine of the Linux Community

FOR W

THE SYSADM
BEHIND
THE WATSON
SUPERCOMPUTER

APRIL 2012 | ISSUE 216 | www.linuxjournal.com

SYSTEM ADMINISTRATION

PUPPET and NAGIOS
Advanced Configuration

sar
The Classic Troubleshooting Tool

HOW-TO
Engineer an OpenLDAP Directory

EXPLORING the Pacemaker High-Availability Stack

MANAGING an EFI Installation

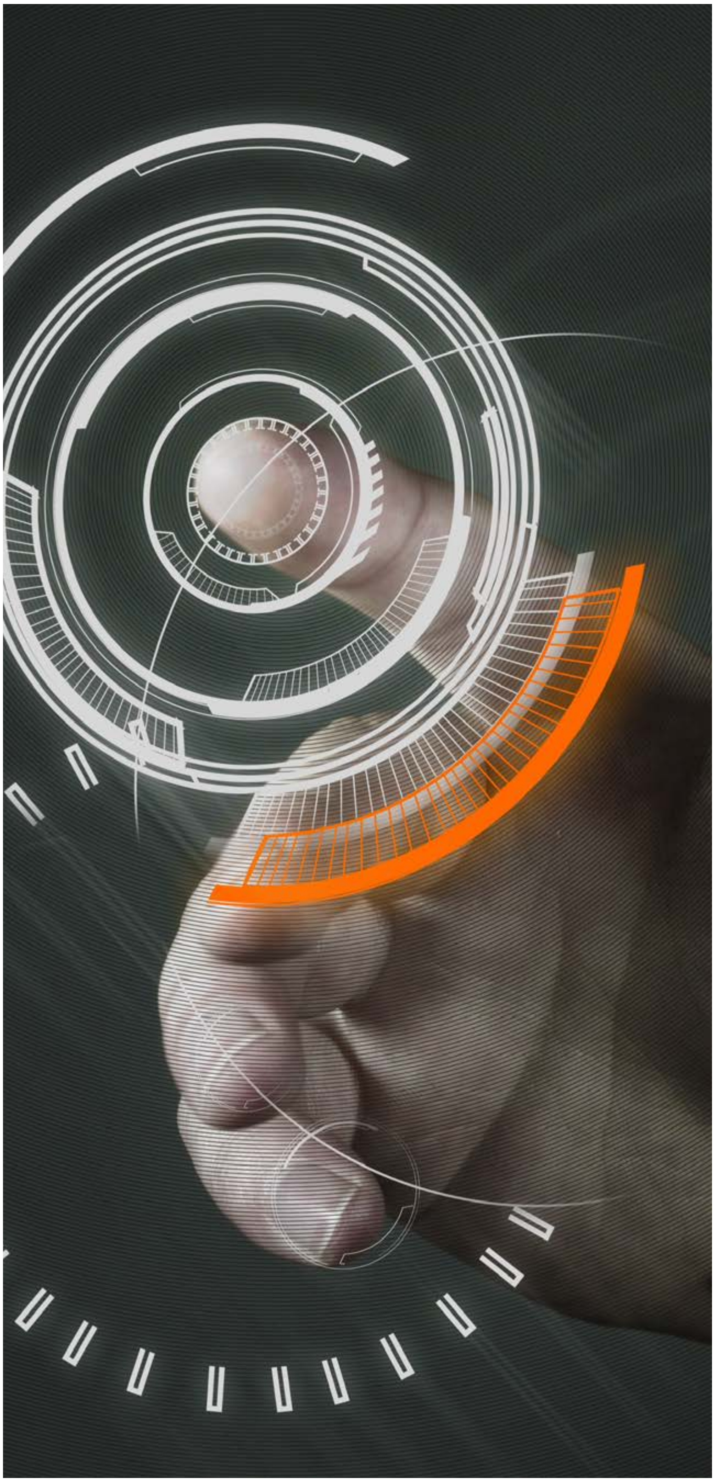
REVIEWED: the ASUS Transformer Prime

CONTENTS

APRIL 2012
ISSUE 216

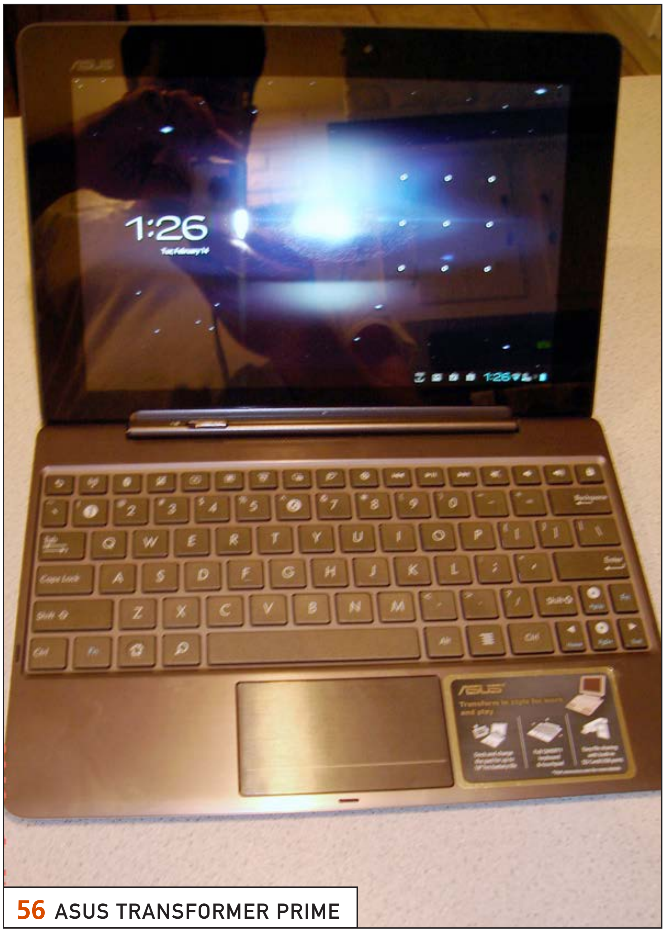
SYSTEM ADMINISTRATION

FEATURES



COVER IMAGE: © Can Stock Photo Inc. / solarseven

COLUMNS

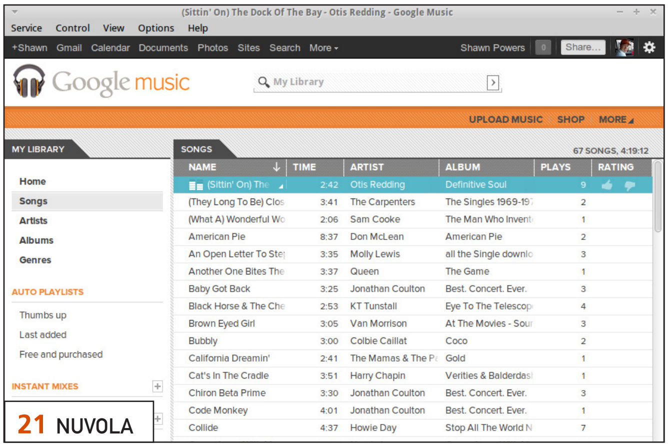


56 ASUS TRANSFORMER PRIME

INDEPTH

REVIEW

IN EVERY ISSUE



NAME	TIME	ARTIST	ALBUM	PLAYS	RATING
(Sittin' On) The	2:42	Otis Redding	Definitive Soul	9	
(They Long To Be) Clos	3:41	The Carpenters	The Singles 1969-19	2	
(What A) Wonderful Wo	2:06	Sam Cooke	The Man Who Invent	1	
American Pie	8:37	Don McLean	American Pie	2	
An Open Letter To Ste	3:35	Molly Lewis	all the Single downlo	3	
Another One Bites The	3:37	Queen	The Game	1	
Baby Got Back	3:25	Jonathan Coulton	Best. Concert. Ever.	3	
Black Horse & The Che	2:53	KT Tunstall	Eye To The Telescop	4	
Brown Eyed Girl	3:05	Van Morrison	At The Movies - Sou	3	
Bubbly	3:00	Colbie Caillat	Coco	2	
California Dreamin'	2:41	The Mamas & The P	Gold	1	
Cat's In The Cradle	3:51	Harry Chapin	Verities & Balderdas	1	
Chiron Beta Prime	3:30	Jonathan Coulton	Best. Concert. Ever.	3	
Code Monkey	4:01	Jonathan Coulton	Best. Concert. Ever.	1	
Collide	4:37	Howie Day	Stop All The World N	7	

21 NUVOLA

ON THE COVER

- Using APIs for Web Apps, p. 30
- The Sysadmin behind the Watson Supercomputer, p. 66
- Puppet and Nagios—Advanced Configuration, p. 86
- sar: the Classic Troubleshooting Tool, p. 42
- How-To: Engineer an OpenLDAP Directory, p. 74
- Exploring the Pacemaker High-Availability Stack, p. 98
- Managing an EFI Installation, p. 108
- Reviewed: the ASUS Transformer Prime, p. 56

LINUX JOURNAL (ISSN 1075-3583) is published monthly by Belltown Media, Inc., 2121 Sage Road, Ste. 310, Houston, TX 77056 USA. Subscription rate is $29.50/year. Subscriptions start with the next issue.

LINUX JOURNAL

Executive Editor	Jill Franklin jill@linuxjournal.com
Senior Editor	Doc Searls doc@linuxjournal.com
Associate Editor	Shawn Powers shawn@linuxjournal.com
Art Director	Garrick Antikajian garrick@linuxjournal.com
Products Editor	James Gray newproducts@linuxjournal.com
Editor Emeritus	Don Marti dmarti@linuxjournal.com
Technical Editor	Michael Baxter mab@cruzio.com
Senior Columnist	Reuven Lerner reuven@lerner.co.il
Security Editor	Mick Bauer mick@visi.com
Hack Editor	Kyle Rankin lj@greenfly.net
Virtual Editor	Bill Childers bill.childers@linuxjournal.com

Contributing Editors
Ibrahim Haddad • Robert Love • Zack Brown • Dave Phillips • Marco Fioretti • Ludovic Marcotte
Paul Barry • Paul McKenney • Dave Taylor • Dirk Elmendorf • Justin Ryan

Proofreader	Geri Gale

Publisher	Carlie Fairchild publisher@linuxjournal.com
Advertising Sales Manager	Rebecca Cassity rebecca@linuxjournal.com
Associate Publisher	Mark Irgang mark@linuxjournal.com
Webmistress	Katherine Druckman webmistress@linuxjournal.com
Accountant	Candy Beauchamp acct@linuxjournal.com

Linux Journal **is published by, and is a registered trade name of,**
Belltown Media, Inc.
PO Box 980985, Houston, TX 77098 USA

Advertising
E-MAIL: ads@linuxjournal.com
URL: www.linuxjournal.com/advertising
PHONE: +1 713-344-1956 ext. 2

Subscriptions
E-MAIL: subs@linuxjournal.com
URL: www.linuxjournal.com/subscribe
MAIL: PO Box 16476, North Hollywood, CA 91615-9911 USA

TrueNAS™ 2U Appliance: You Are the Cloud

Expansion Shelves Available

FreeNAS 8 POWERED

Storage. Speed. Stability.

With a rock-solid FreeBSD® base, Zettabyte File System (ZFS) support, and a powerful Web GUI, TrueNAS™ pairs easy-to-manage FreeNAS™ software with world-class hardware and support for an unbeatable storage solution. In order to achieve maximum performance, the TrueNAS™ 2U System, equipped with the Intel® Xeon® Processor 5600 Series, supports Fusion-io's Flash Memory Cards and 10 GbE Network Cards. Titan TrueNAS™ 2U Appliances are an excellent storage solution for video streaming, file hosting, virtualization, and more. Paired with optional JBOD expansion units, the TrueNAS™ System offers excellent capacity at an affordable price.

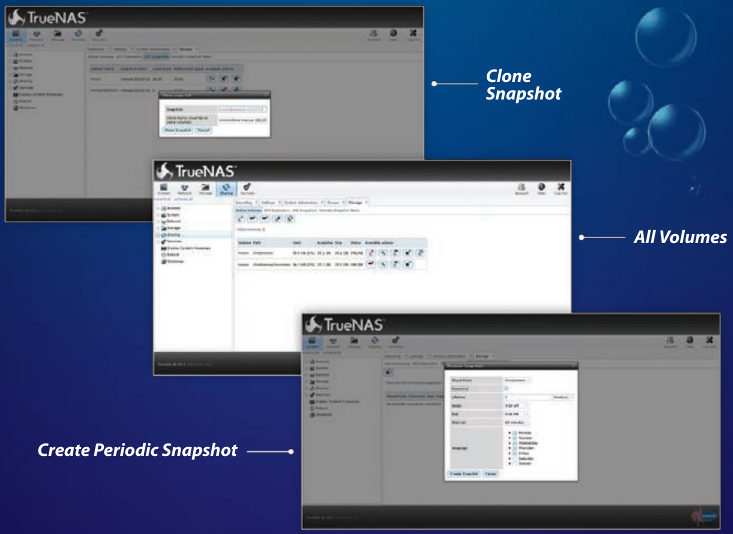
For more information on the **TrueNAS™ 2U System**, or to request a quote, visit: **http://www.iXsystems.com/TrueNAS**.

KEY FEATURES:

- Supports One or Two Quad-Core or Six-Core, Intel® Xeon® Processor 5600 Series
- 12 Hot-Swap Drive Bays - Up to 36TB of Data Storage Capacity*
- Periodic Snapshots Feature Allows You to Restore Data from a Previously Generated Snapshot
- Remote Replication Allows You to Copy a Snapshot to an Offsite Server, for Maximum Data Security
- Software RAID-Z with up to Triple Parity
- 2 x 1GbE Network interface (Onboard) + Up to 4 Additional 1GbE Ports or Single/Dual Port 10 GbE Network Cards

JBOD expansion is available on the 2U System

* 2.5" drive options available; please consult with your Account Manager



iXsystems®

intel® Xeon® inside™

Powerful. Intelligent.

Call iXsystems toll free or visit our website today!

1-855-GREP-4-IX | www.iXsystems.com

Intel, the Intel logo, Xeon, and Xeon Inside are trademarks or registered trademarks of Intel Corporation in the U.S. and/or other countries.

SHAWN POWERS

Sysadmins Ain’t No Fools

This year, April 1st lands on a Sunday. I always enjoy it when April Fools’ Day lands on a weekend, because otherwise I get about a dozen phone calls that go something like this *[our stage is set with Shawn casually sipping his coffee, when suddenly the phone rings]*:

> Me: Hello, technology department, Shawn speaking.
>
> Frantic User: Shawn! My computer was acting slow, then the Internet quit, and now I think I smell smoke!
>
> Me: I see. Have you tried turning it off and back on?
>
> Frantic User: HA HA HA HA HA! April Fools! I so got you, oh you should have heard yourself, classic Shawn. You were so worried, oh man, that was great. I can’t believe you fell for it!

After the 3rd or 4th burning computer, smoking printer or melted projector, I start to wish April 1st was a national holiday so my users could all just go home. This year, we can all sit back and enjoy the day off, thankful that the April issue of *Linux Journal* is focused on us, the sysadmins.

Reuven M. Lerner starts off with some great information on APIs. If you want to interact with other Web sites, programs or even some devices, the API system is how to do so. Reuven shows what that means when it comes to inclusion in your own programs. If your interests are more along the lines of scripting, Dave Taylor likely will pique your interest as he continues his series on how to be a darn dirty cheater in *Scrabble*. Of course, I’m teasing, but Dave does explain how to use the power of scripting to come up with some pretty amazing moves. I’ll leave it up to you to determine whether it’s cheating or not.

Kyle Rankin and I are most comfortable this month, as system administration is right up our alley. Kyle gives a walk-through on using sar, a tool for logging system load. Sure there are other tools for monitoring system load, but sar does a great job of keeping historical records. I have a few tricks up my own sysadmin sleeve this month as well, and I continue my series on LTSP, describing how to tweak your server and clients to get the most out of them both. LTSP 5 provides great flexibility on local apps vs. server apps, and I explain how to set them up.

If you've ever been interested in the inner workings of IBM's Watson supercomputer, or if you ever wondered whether there's just some really smart person behind the curtain speaking in a computer-like voice, Aleksey Tsalolikhin's article will interest you. He takes you behind the scenes and shows off Watson's "guts", many of which are open source. Aleksey also had the chance to interview Eddie Epstein, who was responsible for getting Watson ready to compete on *Jeopardy!* Watson is quite an advanced system, and although it may not be perfect, it's disturbingly close. You won't want to miss the article.

We have a trio of hard-core sysadmin articles this issue as well, all of which should be interesting whether you're a sysadmin yourself or just use a system administered by someone else. Florian Haas writes about Pacemaker, a high-availability stack for Linux. In this crazy data-dependent world, high availability is an important topic. Adam Kosmin follows that with an article on Puppet and Nagios. High availability is great, but unless you can manage your configurations, you'll have highly available junk! Finally, Stewart Walters starts off his series on configuring OpenLDAP for unified logins. Multiple servers means multiple authentication schemes, and when you add different platforms into the mix, things become complicated quickly. Stewart describes how to use one OpenLDAP to rule them all.

Don't worry if you're not a system administrator. As always, we have included tons of other things to tickle the fancy of any Linux geek. Aaron Peters reviews the ASUS Transformer Prime tablet/notebook device. If you're like me and think a tablet computer would be great if only it had a hinge and a keyboard, the Transformer might be just what you're looking for. We've also got product announcements, software spotlights and even a few cheesy jokes thrown in by yours truly. This April issue of *Linux Journal* has something for everyone, and I'm not fooling. Until next month, remember, if your refrigerator is running, you'd better go catch it! ■

Shawn Powers is the Associate Editor for *Linux Journal*. He's also the Gadget Guy for LinuxJournal.com, and he has an interesting collection of vintage Garfield coffee mugs. Don't let his silly hairdo fool you, he's a pretty ordinary guy and can be reached via e-mail at shawn@linuxjournal.com. Or, swing by the #linuxjournal IRC channel on Freenode.net.

letters

Sinatra | HTML5 | Catalyst | Drupal 7 | Moodle
LINUX JOURNAL
INTRO TO THE SINATRA MICRO-FRAMEWORK
THE BEST WAY TO CLIENT CONCURRENCY
WEB DEVELOPMENT
CREATE Web Apps with Catalyst and Perl
AUDIO SOLUTIONS with HTML5
Use DRUPAL 7 for Basic Web Design
INTERVIEW WITH STEPHEN WOLFRAM of Mathematica and WolframAlpha
NEW COLUMN: THE OPEN-SOURCE CLASSROOM

New Game for Old-School Linux People

I just want to give you a tip about a remake of the classic game series *Warlords*. We have made a remake of it in the browser played for free running on our Ubuntu machine. For any old-school gamers who enjoyed turn-based games of the 1990s, it is worth taking a look: **http://www.warbarons.com**. Hope to see you there.

—Piranha

Piranha, thanks for the link! I looked briefly and didn't see hidden fees or anything. Hopefully, it really is as cool as it seems, and I haven't been bamboozled. The only way to tell for sure is to spend time playing. You know, for science!—Ed.

Dave Taylor and *Scrabble*

Just quickly browsing through the new issue, I noticed that Dave Taylor was discussing how to find words in a dictionary that match the desired letters using all sorts of regexes [see Dave's Work the Shell column in the February 2012 issue through the current issue].

Unless I'm missing something very subtle, it seems to me that this problem can be trivially (and efficiently) solved by simply creating a modified version of the dictionary where the letters of each word are themselves put in alphabetical order and then the entire list is resorted. (Each word would be followed by an index that refers to the line number of the original word.)

A trivial binary search then would find the word given an input sequence of letters that are also alphabetically ordered. Use the index beside the word to refer to the original word.

Multiple words that use the same letters would, of course, be right beside each other in the modified list.

—David Jameson

Dave Taylor replies: *An interesting idea, David. I thought about your approach for a few days, but having just finished the third column in the series where I present my own complete solution, it turns out there are a number of reasonably efficient ways to solve the problem. My approach was more to screen words out of the dictionary based on letters appearing in the original "set of tiles",* *than to analyze letter frequency on a word-by-word basis. In that situation, there's no real benefit to sorting alphabetically.*

Further, and this would be a great topic to explore in more detail here in the magazine, I don't really think we need to be as worried about efficiency and performance on modern computers. Heck, I do a lot of my testing in Ubuntu on my MacBook Pro, and it flies through tests and code, even when I give it a big data set to work with. For some situations, of course, speed is of the essence, but I think it's becoming less important for admins and shell script programmers to worry about shaving a few cycles off their code, especially given that if speed's really important, they'll probably switch to Perl, C, Ruby or some other language that's way faster.

In any case, I really appreciate you writing in with your idea. Perhaps you can code your solution, download mine once my column is published, run some speed tests, and then let us know!

More Slackware?

I've had a subscription since 1996. I signed up for the 100 issues for $100 some years ago, and when my subscription was up last year, I renewed. I'm very happy that you've gone all-digital. I love being able to read the journal on my phone, Kindle, Netbook, laptop, PC, etc. My only complaint is there is too much mention of Ubuntu and so little mention of Slackware.

—Albert Polansky

Albert, you've been a subscriber even longer than I have, well done! Believe it or not, we do try to make articles less distro-specific. The problem is popularity. Because Ubuntu is the most popular and most widely used distribution, most of our authors and readers tend to default there. We will continue to strive for more nonspecific distro content. We run into the same problem if we focus on Slackware, in that we alienate a large portion of our readership. Thanks for bearing with us as we attempt to reach everyone.—Ed.

TrueCrypt—Non-Linux FOSS

TrueCrypt has Linux binaries on its download page, and it functions much the same way as its Windows and OS X counterparts. However, you declare the opposite in the Non-Linux Foss article in the Upfront section of the February 2012 issue! Whyyy I oughhttaa....

—Max Mouse

I apologize for the confusion. I didn't mean to say there weren't Linux binaries, I just specifically mentioned the Windows and OS X binaries, because it was the Non-Linux FOSS column. Hopefully, that clears up any misunderstandings.—Ed.

Calibre, No. Firefox/EPUBReader, Yes.

I consume *Linux Journal* in two ways: either in PDF format at my desk with a keyboard and my monitor in portrait mode or as an .epub file from the recliner using my TV and a wireless mouse. For the latter, I found Calibre usable, but unintuitive, unattractive and lacking autoscroll.

Adding EPUBReader to Firefox makes reading the digital edition of *LJ* as good or better an experience as reading the hard copy. In full-screen mode with the bottom menu hidden, only a mouse is needed to toggle autoscroll, adjust text size and move from one section to another.
—J. Lee Becker

I really like the EPUBReader Firefox extension too! In fact, I wrote about it in a recent issue. (Here's the article on-line: **http://www.linuxjournal.com/content/epubreader**.) *Although Calibre is a wonderful tool for managing e-books, I've never really considered the built-in reader to be something I'd actually use to read. It's nice to make sure a conversion went well, but like you mention, it isn't perfect.—Ed.*

Workaround for the Digital Version(s)

I am trying to work with your digital conversion, and I'm hoping it works, because I am trying to get away from paper publications (45+ years of accumulating them gives one a different perspective I suppose). However, it seems it is still a work in progress (at least I hope so).

The PDF is an ongoing annoyance—bloated and requiring constant zoom and scroll to view on any but the biggest monitor (or high-res notebook turned sideways), which means I'm desk-bound. That is why I was not going to renew my digital subscription last year, about the time you switched to digital-only, since that was the only option initially.

The addition of the .epub option got me back onboard again, but it definitely needs some work. I tried using the Android app to read the February 2012 Moodle article, but when it referred to figures, it did not show them unless I switched to the PDF-like page view, which was an unreadably tiny "fail" on my 5" Dell Streak screen. (I got the Streak because it was such a "generously" sized screen for something I could carry with me easily—hah!)

However, I discovered a reasonable workaround when I found that FBReader on my Linux PC provided a viable view, so I copied the .epub to the Streak and opened it with FBReader for Android. It was almost viewable when I long-tapped on the figure image, and not quite as

cumbersome as the PDF zoom-scroll dance. Please keep working on it, y'all.
—ROC

We do continue to work on improving layout every month. On a smartphone, I can't pick my favorite consumption model yet, but on a tablet, I must admit the PDF is pretty slick. I figure since I don't do layout, I can say that without it looking like I'm tooting my own horn! Anyway, thanks for the feedback, and rest assured, we are working constantly to improve the reader experience.—Ed.

Ham Programs

I have been a Ham radio operator for more than 35 years, much longer than an *LJ* subscriber/reader. It sure would be nice to see some articles every now and then about the abundance of Ham radio programs available. I am not much on the programming side. I'm more of a lover/user of Linux, mostly Ubuntu, than a hacker. I love some of your in-depth articles, and I like to see some on the usage of some of the logging programs, propagation and other uses that Linux has with Ham radio. You might even gain some more subscribers (Hams), as I have talked to several who use Linux. Thank you, and keep the magazine going.
—L.B. WB5YDA

WB5YDA, we did an issue focused on Ham radio in January 2010, and we did receive good feedback on it. We'll make sure to be on the lookout for interesting articles about Ham radio and Linux. The crossover between Ham radio operators and Linux users is surprisingly large!—Ed.

Linux Journal Download with wget?

I've been a subscriber to *Linux Journal* ever since I made the move to Linux back in the mid-1990s. There are some aspects of going digital I miss, such as the ability to archive some of my favorite editions on the bookshelf. For example, I still have one of the first *Byte* magazines I bought in June 1979. What many folks don't realize (until they get older) is that those things you absolutely hate in the magazine right now—the ads—are the most fun to read 30 years later! 16K S-100 memory boards for only $300! There's a full-page color ad for a small California company called Apple, showing a 1970s-style family enjoying a personal computer. It really stands out from the other ads emphasizing technical features. I wonder why they're still around?

But, time marches on. Most of my magazines now come in both print and digital format, and I enjoy both. I particularly like the ability to archive and store the PDF editions on my computer, using full-text search to find back-issue articles instantly. I don't yet have an e-reader, because I'm waiting for a full letter-sized E Ink model to come out that

I can build a Gentoo install on. Until then, I use my workstation for reading *Linux Journal*.

The only complaint I have with your digital edition is that, so far, I have not come across a way to use wget to download the issues. I prefer the PDF editions, as I like the traditional publishing look over the EPUB formats. However, I hate downloading PDFs in the browser, as the PDF plugins are far less stable than standalone viewers like Evince. Any chance you could enable wget download, or show me the command-line trick I need to get past the initial re-direction?
—Bob Johnson

Although I don't have an answer for you yet, I'm hoping we can come up with a slightly more geek-friendly download method. This is similar to the desire for direct e-mail delivery to @kindle.com addresses mentioned in the Letters section last month. We'll keep looking into it, and hopefully, we'll be able to come up with something.—Ed.

On-line vs. Print Paradox

For international readers such as myself, *Linux Journal* historically had been an expensive print magazine. Luckily for me, my work subscribed, and I merely had to fight off other readers to get my hands on the precious paper. I always preferred *LJ* to other electronic-only magazines, because I enjoyed the portability of paper. The paradox is this: as soon as *LJ* became electronic-only, I took out my own subscription. Why? Because 12 months of subscription now costs as much as one issue and a bit did previously. At such prices, it becomes a no-brainer to support something I want to survive, even if the format is not my preferred option.
—Geoff Ericksson

Thanks Geoff! It really was a bummer how much it cost for international subscriptions with the paper edition of Linux Journal. *I like the idea that digital distribution levels the international playing field. I'm glad to hear you're still reading, and heck, now you don't have to fight every month!—Ed.*

Going All-Digital

I just wanted to congratulate you on going all-digital—now I know why I couldn't find *LJ* anywhere on the magazine shelves. I just thought they were sold out! I *was* a longtime subscriber from issue #1 to about 2010, and I decided to drop *LJ* when I saw an issue full of articles that really didn't interest me. I enjoy doing a little programming in Java and saw *very* few articles on Java, for one thing. By the way, about that Vol 1, Issue #1, it's autographed by Linus when I met him at a USENIX convention—think it might be worth something? Anyway, congrats on

the new format and good luck. I've seen too many of my favorite mags (such as *Popular Electronics*) go the way of the dodo bird. I'll be watching the content, so heck, I might even resubscribe! Make me an offer I can't refuse.
—Bruce Juntti

Nah, that first issue signed by Linus isn't worth anything, and I think it was printed on asbestos paper. You should send it to me so I can properly dispose of it. Seriously though, I think we're able to take more liberties with the digital edition, because page count and print runs are moot points. We look forward to your possible return.—Ed.

Digital Subscription Rocks

So I picked up my very first *LJ* in July 2011 while scanning a B&N for Linux/computer zines. I won't lie, my eyes looked elsewhere first, but my heart and wallet settled on *LJ*! I read the magazine in about a weekend (my girlfriend wasn't too happy since it was 4th of July) and anxiously awaited the August 2011 issue. When it came out, I read all of it on a couple bus rides to work. During September, I found out the zine was going all-digital, and at first I had mixed feelings, but since I had a NOOK (the original), it would come in good use. I bought my first digital copy in January 2012, and I liked the format a lot, so I purchased again in February and a few days later, purchased a subscription. I like the zine and the Web site for its content, and I am willing to adapt to reading it on an e-reader or a computer.

One recommendation: make the archive in EPUB format, not PDF (I will buy it if you do this), and put more videos on the site. Happily reading digitally in 2012!
—Sean

We do have the EPUB version for every issue since we went digital, and we recently released an EPUB archive of all the issues from 2011 (available at **http://lj.mybigcommerce.com/products/Linux-Journal-2011-eBook.html***). Doing these archives means laying out every issue again from scratch, so we'll be gauging reader interest before making more archives.—Ed.*

"Codger" Love

I am an avid *Linux Journal* reader and have been for several years. There are some other attributes regarding me that are relevant to this communique as well. I'm an old geezer having had a life of IT and am now retired. In addition, I'm an amateur radio operator, and also, I am the editor for a regional Ham radio magazine (paper) serving the southeastern United States. When I took this job, I decided to do the layout with open-source tools rather than the conventional, established toolsets. Therefore, I use Scribus for my 60–80 page publication.

Until your conversion to the digital format, I was reluctant to purchase a tablet, but I felt it would have to be my next significant purchase. When surprised with your move to digital, I actually printed the first digital issue on paper! *Never* will I do that again! I was incensed that you would make such a change so abruptly. As a paper magazine editor, I had a sense of the costs involved with your printing and distribution, and although we have not yet succumbed to the temptation of digital, I already had written articles and told my board that day may come. Therefore, mad as I may have been with your move, I could not fault you for it. As ours is a membership-based circulation, I do not have to fight the profit motive nearly as fiercely as you must.

Nonetheless, I really did enjoy carrying my *LJ* to various eating establishments and reading it over a meal. (Geriatric and single, I am!) However, even with the Android app on my cell phone, I found it difficult to enjoy the magazine as I did the print version.

2012

USENIX Federated Conferences Week

June 12–15, 2012 Boston, MA

www.usenix.org/fcw12

Networked Systems for Developing Regions
Applic
Storage Provenance
Systems Research Cloud Computing
Networked Systems for Developing Region
Web Application Development
Provenance Systems Research
Cloud Computing Storage
Netwo Developi

Back for 2012, USENIX is combining established conferences and workshops into a week of research, trends, and community interaction. Events include:

///// USENIX ATC '12

2012 USENIX Annual Technical Conference
Wednesday–Friday, June 13–15 www.usenix.org/atc12

WebApps '12 /////

3rd USENIX Conference on Web Application Development
Wednesday, June 13 www.usenix.org/webapps12

///// HotCloud '12

4th USENIX Workshop on Hot Topics in Cloud Computing
Tuesday–Wednesday, June 12–13 www.usenix.org/hotcloud12

HotStorage '12 /////

4th USENIX Workshop on Hot Topics in Storage and File Systems
Wednesday–Thursday, June 13–14 www.usenix.org/hotstorage12

///// TaPP '12

4th USENIX Workshop on the Theory and Practice of Provenance
Thursday–Friday, June 14–15 www.usenix.org/tapp12

NSDR '12 /////

6th USENIX/ACM Workshop on Networked Systems for Developing Regions
Friday, June 15 www.usenix.org/nsdr12

EARLY BIRD DISCOUNT
REGISTER by MAY 21 & SAVE

///// Why Attend

You'll leave the week with insights into the future of many hot topics. Your access to leading researchers and industry experts will not only provide answers to your most burning questions, but will also help you build lasting connections across multiple disciplines. Take full advantage of this opportunity and learn from the top researchers, practitioners, and authors all in one place, at one time.

usenix ASSOCIATION

www.usenix.org

Stay Connected...

http://www.usenix.org/facebook
http://www.usenix.org/linkedin
http://twitter.com/usenix
http://blogs.usenix.org

So, I decided to break down and spend $79 dollars for a Kindle. What a grand decision that was! I can think of three or four things about the Kindle/.mobi version that I do not like, but the convenience beats them all in the final tally! I am beginning to wonder if I should have sprung for the Kindle Fire or even a tablet, however.

Like other readers have commented, I considered just dropping my *LJ* subscription when it runs out. But that decision is made now. This stubborn, old, rather inflexible, codger has become a part of the digital world. Now I begin a quest to migrate *gradually* my own paper periodical to the Web. Best of luck with your continued orientation to digital publishing. What a vast difference it is from paper publishing! I would love to learn at your feet!
—John C. Ghormley

John, I'm happy to hear the Kindle is working well for you! The .mobi version is a bit frustrating for us too, and we keep working to improve it. This is especially true when it comes to coding examples and such. Thanks for sticking with us.—Ed.

Where Is the Loyalty?

I used to enjoy reading the Letters section, but since the switch to digital, it seems like it's being filled with whining. I understand that you are sympathetic to those who are so resistant to change, but one can take only so much of the baseless arguments of so many grown men crying.

LJ moves to digital stating one of the main reasons as being economic, and then dozens of grown men complain that they can't afford a device on which to read the magazine. Don't even get me started on the fact that the world in which Linux is used is *always changing*! They are upset that they don't have a physical copy to read in the bathroom?! In ten years from now, are they just not going to read new content anymore?

Where is the loyalty? "I've been a customer for 14 years...but now I'm leaving." That doesn't make any sense. All because they now have an excuse to purchase a tablet or a smartphone? Please. The interpretation of that to me is this:

"Dear *LJ*, You went digital with the intent to keep *Linux Journal* alive. I don't care enough about the content to keep reading if that means I can't turn the physical pages. Either keep printing and go out of business, or I'm no longer going to subscribe. Sincerely, Ignorant ones."

It's frustrating. I say good riddance! Not really, but you get the point. I wonder if it was this hard for these people to switch to digital cameras because they liked the feel of driving to the store to get their film developed. Maybe they had a hard time transitioning to CDs because they liked the way the cassette tape would get wrapped around the heads, and they'd have to wind it all back by hand. They probably complained about having to switch out all of their VHS tapes too.

It was wise to make the decision to go digital, and I, for one, am glad that the decision was made so I can continue to enjoy the true value of *LJ*—the content.

By the way, you can't Ctrl-f a piece of paper—at least not yet.
—Ryan

Ryan, I certainly appreciate your candor! In defense of those folks upset by the change, it was a very sudden decision. That's not to say it was a surprise given the state of the publishing world, but nonetheless, it was a shock due to the swift transition. To be honest, I'd rather folks write and tell us they're upset than just stay angry. Hopefully, our responses the past few months have been insightful, or at least sympathetic.

The good news is that, yes, the digital format offers some amazing advantages! I won't deny there are drawbacks, but you're right, the ability to Ctrl-f an issue is awesome. If only I can figure out a way to digitize my office, because I can't seem to find anything in here either!—Ed.

WRITE *LJ* A LETTER We love hearing from our readers. Please send us your comments and feedback via **http://www.linuxjournal.com/contact**.

LINUX JOURNAL

At Your Service

SUBSCRIPTIONS: *Linux Journal* is available in a variety of digital formats, including PDF, .epub, .mobi and an on-line digital edition, as well as apps for iOS and Android devices. Renewing your subscription, changing your e-mail address for issue delivery, paying your invoice, viewing your account details or other subscription inquiries can be done instantly on-line: **http://www.linuxjournal.com/subs**. E-mail us at subs@linuxjournal.com or reach us via postal mail at *Linux Journal*, PO Box 16476, North Hollywood, CA 91615-9911 USA. Please remember to include your complete name and address when contacting us.

ACCESSING THE DIGITAL ARCHIVE: Your monthly download notifications will have links to the various formats and to the digital archive. To access the digital archive at any time, log in at **http://www.linuxjournal.com/digital**.

LETTERS TO THE EDITOR: We welcome your letters and encourage you to submit them at **http://www.linuxjournal.com/contact** or mail them to *Linux Journal, PO Box 980985,* Houston, TX 77098 USA. Letters may be edited for space and clarity.

WRITING FOR US: We always are looking for contributed articles, tutorials and real-world stories for the magazine. An author's guide, a list of topics and due dates can be found on-line: **http://www.linuxjournal.com/author**.

FREE e-NEWSLETTERS: *Linux Journal* editors publish newsletters on both a weekly and monthly basis. Receive late-breaking news, technical tips and tricks, an inside look at upcoming issues and links to in-depth stories featured on **http://www.linuxjournal.com**. Subscribe for free today: **http://www.linuxjournal.com/enewsletters**.

ADVERTISING: *Linux Journal* is a great resource for readers and advertisers alike. Request a media kit, view our current editorial calendar and advertising due dates, or learn more about other advertising and marketing opportunities by visiting us on-line: **http://ww.linuxjournal.com/advertising**. Contact us directly for further information: ads@linuxjournal.com or +1 713-344-1956 ext. 2.

diff -u
WHAT'S NEW IN KERNEL DEVELOPMENT

Arjan van de Ven recently found a way to reduce **boot times** on most systems by a staggering amount. Instead of the usual approach, starting each CPU in turn, Arjan's patch starts them all at once. On his system, he achieved a 40% faster bootup with this patch. This also seems to be just the beginning. Arjan has plans to get userspace started up after just the first CPU starts up, while the other CPUs are still pending. This could give the user a usable system extremely quickly. However, as he and **Ingo Molnar** both agreed, this would involve some highly invasive changes to the kernel that would certainly result in much breakage throughout the entire system. Their plan is to go slowly and carefully, fixing problems as they're revealed, and ultimately achieve massive goodness.

Several kernel developers, including **Pavel Emelyanov**, **Oleg Nesterov** and **Tejun Heo** are working on a way to control the **process ID** (PID) of a given process. Most of the time the PID is not relevant to ordinary users—what typically matters most is that the process runs correctly. But, sometimes the ability to change the PID can enable really cool features, such as migrating processes cleanly from one computer to another or allowing a debugger to switch cleanly between several saved states of a running process.

Unfortunately, finding the best place in the kernel source tree to provide control over PIDs is not easy. And, **Linus Torvalds** has said that it's important to keep this kind of feature from running away with itself, adding strange new functionality in places where such things aren't needed and only muddy the interfaces.

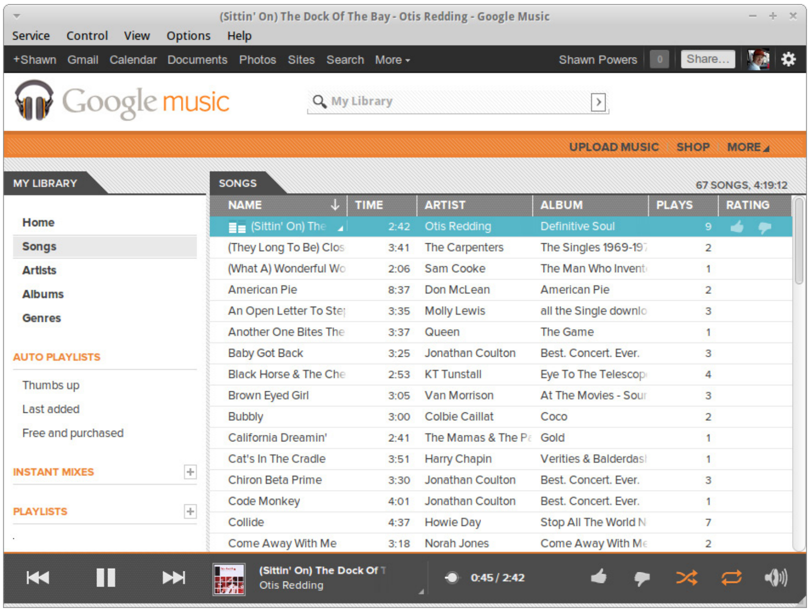
Believe it or not, it's official kernel policy to restrict patches to **80 columns** of text or less. This ancient requirement dates back to the era of text-mode consoles that were themselves restricted to 80 columns. But there are some places in the kernel source that violate that rule, and a very large number of kernel developers would love to ditch the restriction altogether or at least make it something a bit more palatable, like 100 columns or (for the sake of multiple-of-8-ness) 96 columns.

Folks like Ingo Molnar and **Andrew Morton** endorsed changing the policy, and a lot people were very pleased. But, apparently Linus Torvalds is dead set against changing it. In his post to the mailing list, he said that yes, some parts of the kernel were exceptions and used more than 80 columns. He also said that it was important to *keep* those places exceptional, and not get the idea that more and more code on a single line would be a good thing.

So, for now at least, and really probably for a good number of years to come, the 80-column limit on kernel patches remains in effect.

—ZACK BROWN

Nuvola: the Linux Choice for Cloud-y Music



NAME	TIME	ARTIST	ALBUM	PLAYS	RATING
(Sittin' On) The	2:42	Otis Redding	Definitive Soul	9	👍 👎
(They Long To Be) Clos	3:41	The Carpenters	The Singles 1969-197	2	
(What A) Wonderful Wo	2:06	Sam Cooke	The Man Who Invent	1	
American Pie	8:37	Don McLean	American Pie	2	
An Open Letter To Ste	3:35	Molly Lewis	all the Single downlo	3	
Another One Bites The	3:37	Queen	The Game	1	
Baby Got Back	3:25	Jonathan Coulton	Best. Concert. Ever.	3	
Black Horse & The Che	2:53	KT Tunstall	Eye To The Telescop	4	
Brown Eyed Girl	3:05	Van Morrison	At The Movies - Sour	3	
Bubbly	3:00	Colbie Caillat	Coco	2	
California Dreamin'	2:41	The Mamas & The Pa	Gold	1	
Cat's In The Cradle	3:51	Harry Chapin	Verities & Balderdasl	1	
Chiron Beta Prime	3:30	Jonathan Coulton	Best. Concert. Ever.	3	
Code Monkey	4:01	Jonathan Coulton	Best. Concert. Ever.	1	
Collide	4:37	Howie Day	Stop All The World N	7	
Come Away With Me	3:18	Norah Jones	Come Away With Me	2	

Nuvola Player (formerly known as google-music-frame) is a Linux application that integrates cloud-based music services into your Linux desktop. I've tested it only with Google Music, but Nuvola now supports Google Music, Grooveshark, Hype Machine and 8tracks. It also supports Last.FM scrobbling.

What makes Nuvola better than a standard Web browser is that it integrates with the Linux desktop. Nuvola gets its own tray icon and supports multimedia keys on keyboards that have them. I traditionally store my music files in Dropbox, but since Google Music has excellent support on Android phones, I find myself using the cloud-based service almost exclusively. Nuvola makes that experience much nicer. Check it out at **https://launchpad.net/nuvola-player**.

—SHAWN POWERS

Kill A Watt: Now with Less Math!

If you're interested in how much energy your electronics use, it's hard to find a device better than a Kill A Watt—except maybe the Kill A Watt EZ! P3 International now offers model P4600, which provides the same features as its predecessor, but it also automatically calculates device cost per day, week, month or year.

P3's new version does automatic calculation, and it also has a battery-powered memory to retain measurements even when unplugged. We still recommend using a short extension cord, because it's often difficult to read the Kill-A-Watt when it's plugged in to the wall. We recommend checking devices you suspect of high usage, and also those you don't. A laser printer, for example, uses very little power when idle, but things like an electronic pet-containment fence or even a doorbell can use much more than you'd suspect. Visit **http://p3international.com/products/p4460.html** for more information.—SHAWN POWERS

(Photo from **p3international.com**)

They Said It

All of the books in the world contain no more information than is broadcast as video in a single large American city in a single year. Not all bits have equal value.

—Carl Sagan

I am often amazed at how much more capability and enthusiasm for science there is among elementary school youngsters than among college students.

—Carl Sagan

If you wish to make an apple pie from scratch, you must first invent the universe.

—Carl Sagan

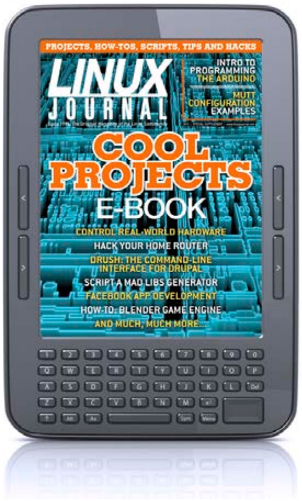
Somewhere, something incredible is waiting to be known.

—Carl Sagan

Absence of evidence is not evidence of absence.

—Carl Sagan

Announcing Our Latest eBook: Cool Projects


PROJECTS, HOW-TOS, SCRIPTS, TIPS AND HACKS
LINUX JOURNAL
INTRO TO PROGRAMMING THE ARDUINO
MUTT CONFIGURATION EXAMPLES
COOL PROJECTS
E-BOOK
CONTROL REAL-WORLD HARDWARE
HACK YOUR HOME ROUTER
DRUSH: THE COMMAND-LINE INTERFACE FOR DRUPAL
SCRIPT A MAD LIBS GENERATOR
FACEBOOK APP DEVELOPMENT
HOW-TO: BLENDER GAME ENGINE
AND MUCH, MUCH MORE

We're big fans of e-reader formats here at *Linux Journal*, and we know a lot of our readers are too, so we're expanding our offerings. We've just introduced a new special-edition compilation of some of the coolest projects, code snippets and how-tos to add to your electronic library. This brand-new special-edition eBook is full of how-tos and projects that will get you coding, hacking and making the most of your tech gear with a collection of articles from many of our most popular authors that is sure to spice up your eBook collection.

The Cool Projects eBook includes topics like programing an Arduino, router hacking, HTML5, Blender game development and shell scripting, and it should provide inspiration for readers to start working on something cool of their own.

Visit **http://www.linuxjournal.com/cp-ebook** to get yours!

—KATHERINE DRUCKMAN

Non-Linux FOSS

IRC is one of those chat protocols that really lends itself well to the command line. Irssi works very well, and because it runs on a server, all a person needs to use it is a terminal window. For many users, however, a friendly GUI makes the chatting experience a lot nicer. In Linux and Windows, there are many options from which to choose, but in OS X, the options are a little more limited. Thankfully, one of the nicest IRC clients for OS X also happens to be open source!

Colloquy has the familiar look of an OS X program, but it doesn't sacrifice any of the text-based power of IRC. It supports file transfers, notification sounds, and if you read Bill Childer's article "Seamlessly Extending IRC to Mobile Devices" in the March 2012 issue, you'll know it can work with your Irssi install for some really awesome multi-endpoint chatting.

Colloquy is available for OS X, and also for mobile devices. Check it out at **http://www.colloquy.info**.—SHAWN POWERS

Chemistry the Gromacs Way

In this article, I'm diving into chemistry again. Many packages, both commercial and open source, are available to make chemistry calculations at the quantum level. The one I cover here is gromacs (**http://www.gromacs.org**). It should be available for your distribution via its package manager.

The gromacs package is actually a full complement of small applications to take care of all of the steps from creating the initialization files, to doing the computational run, to doing analysis and visualization of the results. You also may notice more than one version available. For example, Ubuntu has both a single-processor version and an MPI version to do the calculations on a parallel machine. If you have a multicore machine or a cluster of machines at your disposal, this can help speed up your calculations quite a bit.

Before I start, here's a little background as to what types of calculations are possible and what methods gromacs uses. Ideally in computational chemistry, you should be able to take the Schrodinger's equations for the system in question and solve them completely. But, outside of very simple systems of only a few atoms, doing so becomes impossible rather quickly. This means some sort of approximation is required to get a result. In gromacs, the approximation method is molecular dynamics (MD). MD simulations solve Newton's equation for a set of interacting particles to get their trajectories. The total force on a particle from all of the other particles is calculated and then divided by the mass of the particle to get its acceleration. This is calculated for all of the particles, and every one moves according to its acceleration. Then, time is stepped one unit, and the whole thing is calculated again. You can increase the spatial and temporal resolution, at the cost of more computer time, until you get results that are accurate enough for your purposes.

MD simulations have several limitations. The first is that it is a classical simulation. In most cases, this is fine, but there are situations when you need to know the quantum effects of the atoms in your system. The second limitation is that electrons are in their ground state and are treated as if they move instantly to remain in orbit around their atoms. This means you can't model chemical reactions or any other electronic interactions. The next limitation is that long-range interactions are cut off. This becomes a serious issue when dealing with charged particles. The last limitation that I look at here is the fact that periodic boundary conditions are being used to try to simulate bulk systems. This, combined with the cutoff mentioned above, means you can end up with some unphysical results, especially during long runs.

Now that you have some background, how do you actually use gromacs? You need to start with the inputs: the initial position of all the atoms, the initial velocities and the interaction potential describing the forces between all of the atoms. The center of mass of the entire system is usually defined as having zero velocity, meaning there are no external forces on the system. Once this initial data is entered, gromacs needs to calculate the forces, apply these forces to each particle and calculate their new positions. Once they have moved, gromacs needs to recalculate the forces. This is done using the leapfrog algorithm. To get a better feel for what this looks like, let's consider a concrete example: a protein in water.

The first step is to come up with the initialization files needed to do your calculation. This can be done completely from scratch, but in many cases, it isn't necessary. In the case of proteins, there is the Protein Data Bank (PDB), located at **http://www.pdb.org**. Make sure the protein structure has the required detail you are looking for in your simulation. You also can load it up in PyMOL and see what it looks like. When you are happy with your selection, you can use it to generate the required initialization files for gromacs with `pdb2gmx`:

```
pdb2gmx -f my_protein.pdb -water tip3p
```

where `tip3p` is one of the water models that you can select. This command generates several output files, the most important of which are conf.gro, topol.top and posre.itp. At this point, you still haven't added the water to the initialization files. To do so, you first need to define a box to hold the water and the protein. To do that, you would edit the configuration files with the following:

```
editconf -f conf.gro -bt dodecahedron -d 0.5 -o box.gro
```

This defines a box with the shape of a dodecahedron and a diameter of 0.5nm. You also can use boxes with cubic or octahedron shapes. Now that you have a box, you can add the water with the command:

```
genbox -cp box.gro -cs spc216.gro -p topol.top -o solvated.gro
```

This command takes the box (box.gro) and fills it with water molecules as defined in the file spc216.gro. The `-p topol.top` option adds this water to the topology of the system. Finally, all of this is written out to the file solvated.gro.

If you tried to run it now, you probably would run into issues because of large forces caused by the introduced water molecules. To deal with that, you can minimize the energy of the system before actually doing any calculations. You can do this by creating a parameter file to define how to do the minimization. For example:

```
------em.mdp------
integrator    = steep
nsteps        = 200
nstlist       = 10
rlist         = 1.0
coulombtype   = pme
rcoulomb      = 1.0
vdw-type      = cut-off
rvdw          = 1.0
nstenergy     = 10
------------------
```

In this example, the minimization is being done by steepest-descent, over 200 steps. You can look up the details of all the other options in the gromacs documentation. With this finished, you can do all of the necessary pre-processing with the `grompp` command:

```
grompp -f em.mdp -p topol.top -c solvated.gro -o em.tpr
```

The actual minimization is handled through:

```
mdrun -v -deffnm em
```

The prefix `em` is used for all of the relevant filenames, with different extensions. This makes it easier to do all of the pre-processing and get the initialization steps completed on your desktop, then doing the actual run on a supercomputer.

When you are ready to do the final run, you need to set up a parameter file describing the details. In this example, you could use something like this:

```
------run.mdp------
integrator    = md
nsteps        = 5000
dt            = 0.002
nstlist       = 10
rlist         = 1.0
coulombtype   = pme
rcoulomb      = 1.0
vdw-type      = cut-off
rvdw          = 1.0
tcoupl        = Berendsen
tc-grps       = protein non-protein
tau-t         = 0.1 0.1
ref-t         = 298 298
nstxout       = 1000
nstvout       = 1000
nstxtcout     = 100
nstenergy     = 100
------------------
```

With this parameter file, you can do the pre-processing with:

```
grompp -f run.mdp -p topol.top -c pr.gro -o run.tpr
```

The actual MD calculation is done with the command:

```
mdrun -v -deffnm run
```

Now you can go off and drink a few days' worth of coffee. The actual runtime will depend on how many processors you have to throw at the problem.

Once this run is completed, you still need to analyze it to see what you can learn from it. You can compare the results to experimental results from X-ray structure measurements. You can measure the displacement of the heavy atoms from the X-ray structure with the g_rms program. You can analyze distance and hydrogen bonds with the g_dist and g_hbond programs. You can even make a movie with the trjconv program:

```
trjconv -s run.gro -f run.xtc -e 2500.0 -o movie.pdb
```

This will export 2.5ns to a movie of the trajectories. You then can view it using PyMOL.

This short article has provided only the faintest taste of what gromacs can do. I hope it sparks your interest in reading up on it and doing some new and interesting work.—**JOEY BERNARD**

Make TV Awesome with Bluecop

A few weeks back, I was whining that although *Doctor Who* was available on Amazon Prime streaming, I didn't have any way to watch it on my television. Thankfully, my friend Richard Servello pointed me to the bluecop repo for XBMC. Not only does bluecop support Amazon Prime streaming, but it also has add-ons for Hulu and countless other network-video-streaming collections.

Now, not only can I take advantage of my Amazon Prime membership on our 55" TV, but also my family can watch clips of *The Target Lady* on Hulu. I don't think the add-ons in the bluecop repo are endorsed by any of the streaming-media providers, but it seems they pull video only from the Web. If you want to extend your XBMC setup to include a huge selection of streaming media, check out the bluecop repository (**http://code.google.com/p/bluecop-xbmc-repo**). Richard, thanks for the tip!

—**SHAWN POWERS**

2-Plan Desktop—the Free Project Management App

I have been searching for an eternity for a suitable project management software package that doesn't conflict with my limited budget. My needs seem like they should be relatively easy to fulfill: something simple, flexible, stable and robust that has the ability to grow with my needs, even though I may not be aware of what those needs actually are yet. My options have been extremely limited, as most of my peers have grown accustomed to the mainstream software (Microsoft Project) and the standard output it has produced for them. But, my situation has, without a doubt, changed for the better!

I recently found the answer to my endless on-line searches—2-Plan Desktop. 2-Plan is available as an open-source solution that runs on multiple operating systems, including my bread-and-butter Linux. It also comes with a feature set that is very comparable to Microsoft Project, but without the hefty price tag. Other features further distinguish 2-Plan from many of the other well-known solutions.

One of these features is the simple but effective graphical work breakdown structure (WBS). Users have the ability to create and reuse similar work packages across multiple projects with ease, saving time by eliminating duplication of work. By utilizing the drag-and-drop abilities within the WBS, users can define work packages with little-to-no requirements quickly and create a generalized top-down overview during the initiation phase. Then at a later time, users can revisit these work packages and further define as the requirements become available.

A second great feature is the ability to define various duration types for work packages. This feature is not common among all applications with the exception of Microsoft Project. Whether it be a fixed duration, manual or effort-driven, I can specify the duration type and work toward achieving an overall better-executed project plan.

Another great feature (or downloadable plugin that is available from the 2-Plan site) is the Extended Project Explorer. This feature is similar to mind-mapping software applications where the software displays your work packages in an animated graphical snapshot. Microsoft Project and Gantt Chart don't even offer this solution.

Despite the fact that 2-Plan is open source, it definitely does not come with a limited set of features or options that are available only in the so-called "premium packages", as often is the case. There is no set limit to the amount of projects users are allowed to create within the software. An adequate forum is also available from the software creators to assist with any problems or suggestions for improvements you may have. The reporting features are more than adequate and similar to most other project management software packages.

Project managers tend to become very familiar with the ability to assign project milestones directly to work packages. Project milestones typically play an integral part within a project plan. 2-Plan is similar to MS Project as this feature is built in.

Template creation is a snap within 2-Plan. These templates include the basics, such as work packages, teams and roles that will cut down overall project planning time tremendously.

2-Plan incorporates the best of both the standard project management and Agile philosophies. I have found other software packages to lean more toward one side of the spectrum or the other, not allowing the flexibility or straightforwardness that is needed in real-world projects. I truly believe the software I use should allow me to work in a way that fits my standards best instead of making me alter everything I do to fit the software I'm using.

2-Plan allows me to do exactly that and work more efficiently in a manner to which I have now grown accustomed. Having tried many products during the years that claim to be similar to MS Project or Gantt Chart in their overall feature sets, most seem to fall short in their promises. Very few have delivered something close, and still fewer have delivered a software package with similar features and a price tag I could afford. 2-Plan Desktop has the features I need, and I have to say only one word when it comes to the pricing of the package—*free*. If you find yourself looking to stay on a more flexible OS and also needing a project management solution that can grow with you and your responsibilities, consider 2-Plan Desktop. It will fit into your budget—big or small. You will not be disappointed.—**SCOTT ANDERSON**

REUVEN M. LERNER

APIs

If you're creating Web apps, you're designing APIs. Here are some things to keep in mind before you begin.

The Web was designed for people. When Tim Berners-Lee created the trio of standards that make up the Web—HTTP, HTML and URLs—the intention was for people to browse Web sites, submit information to them and be at the heart of the experience. But for some time now, the notion of the Web as a set of sites that people browse has been somewhat untrue. True, hundreds of millions of people visit an equally large number of sites each day; however, more and more of the visitors to sites aren't people, but programs.

Some of those programs exist to collect information on behalf of other systems. For example, when you search the Web using a site such as Google, you're obviously not searching through all of these sites in real time. Rather, you're asking Google to search through its massive index—an index that it has created and updated via its "bots", programs that go to a Web site, pretend to browse as a person, and then track whatever they find.

But more and more, the programs that are visiting sites aren't doing it on behalf of search indexes. Rather, they're doing it on behalf of...well, on behalf of themselves. Computers exchange information via the Web, using a variety of protocols and data formats. Native apps on mobile devices are using the Web behind the scenes to query Web applications. And, even those Web applications using Ajax are interacting with a Web site without directly being asked to do so.

This represents a massive shift in what Web applications are doing. No longer are we just producing HTML for users (and search bots). Now, we're producing output that's meant for programmatic consumption—and in many cases, the same people are writing the client and server sides. Sure, we could use "scraping" techniques to retrieve the HTML and search through it, but why do so? If we already know we'll be sending data to a program, there's no reason to send HTML. Rather, we can send it in a more program-friendly data format, without all the bells and whistles that people require.

When such use began, people made a big deal out of it. This trend was known as "Web services", and a lot of companies—most prominently Amazon—jumped on them, describing all sorts of standards, from XML-RPC to SOAP to WSDL. These protocols are still used, especially by large-scale enterprise applications, to communicate with one another.

But during the last few years, a more informal sort of API has emerged. Sometimes it's based on XML, but more often it's based on JSON, the "JavaScript Object Notation" format that works not only with JavaScript, but with a wide variety of other languages as well.

(By "more informal", I don't mean that it's not useful or that more formality is needed. I'm merely referring to the fact that it requires coordination between the client and server software authors, rather than adherence to a specification document or established protocol.)

This month, I'm looking at these sorts of APIs—why you would want them, the different styles you can use in designing them, and then how to access and use them.

Why an API?

If you're running a Web application, you probably will want to provide an API at some point. Why? Well, there are a number of reasons:

- To allow your users access to their data via third-party applications. Consider how many third-party Twitter clients exist, all of which use Twitter's API, rather than the Web site. The same is true for Amazon and eBay, among others, which allow users to access their catalog data and even execute sales, all via the API.

- To allow mobile app developers to access your site. Mobile apps—that is, those running on such operating systems as Android and iOS—often send and retrieve data using HTTP, providing their own front end, in what we might consider a "domain-specific browser", albeit with a non-Web interface.

- To allow your own application to access its own data via Ajax calls. When you make a JavaScript call in the background using Ajax, you most likely will want to make use of an API call, receiving XML or JSON, rather than HTML that requires further parsing.

I'm sure there are additional reasons for providing an API, but even one of these might be a compelling reason in your case—and two or three of them might apply as well. There's also an element of customer openness and trust that comes with an API. I'm much more likely to use a Web application, particularly if it's one for which I'm paying, that provides an API to some or all of its functionality. Even if I never end up making use of it, I know I can do so potentially, which gives me a feeling of being a potential partner of the application's authors, rather than a simple user.

The above also demonstrates that even if you never plan to open up your API to the outside world, it might still be in your interest to design one. Indeed, I've recently heard several experienced Web developers argue that a modern Web site should not be designed as a set of pages, with the API tacked on as an afterthought, but rather as a set of APIs, which can be accessed via a mobile app, a remote client application, Ajax calls or a client-side framework, such as Backbone. In other words, first you establish your APIs, and then you get to work writing applications that use those APIs.

In many ways, this is an attractive thought, one that has the potential to make applications cleaner and easier to write and test. After all, the idea behind the MVC (model-view-controller) paradigm is to separate the different components, such that the business logic has no interaction with the interface presented to the user. MVC-style Web frameworks, such as Rails and Django, encourage this separation, but creating an API makes the distinctions even sharper.

API Styles

If you have decided to create an API, there are several questions to ask. One of them is what style of API you will use. The Ruby on Rails community has rallied around the idea of REST—"representational state transfer", a term coined by Roy Fielding—which basically assumes that each URL uniquely identifies a resource on the Internet. Different actions performed on that resource do not make use of different URLs, but rather of different HTTP request methods.

For example, if you have an address book containing information on many people, the first entry might have a URL of:

```
/people/1
```

In such a case, you could retrieve information with:

```
GET /people/1
```

and a new entry with:

```
POST /people
```

Remember that POST requests send their name-value pairs separately from the URL. Because the parameters aren't part of the URL, it sometimes can be a bit tricky to know precisely what is being sent to the server. I generally check the parameters that are being sent using a combination of tools, including the server's logfile, the Firebug plugin for Firefox, the Web developer plugin for Firefox or the ngrep command-line tool for finding and displaying selected network packets.

In the Ruby on Rails universe, you can update an existing entry using the little-known (and little-supported, at least for now) PUT request method:

```
PUT /people/1
```

As with POST, the parameters in a PUT request are sent separately from the URL. Trickier yet is the fact that many browsers cannot handle PUT requests directly. The solution that Rails employs, for the time being, is to use POST, but to add a "_method" parameter as part of the request. When the server sees this, it uses the action that should be associated with PUT, rather than with POST. The system works well, although it obviously would be better if browsers were able to support all of the standard HTTP requests.

One of the key things to remember when working with REST is that URLs should refer to nouns, not verbs. Thus, it's perfectly acceptable, within the REST world, to have URLs that refer to any object on your system, from users to books to airplanes to credit-card statements. However, it's not acceptable to name the action you wish to take in the URL. So:

```
/airplanes/523
```

would be perfectly acceptable, but:

```
/airplanes/get_passenger_list/523
```

would not be. The difference being, of course, that get_passenger_list is presumably the name of an action that you wish to take on the airplane resource. This means you're no longer using the URL to refer to a specific resource, but instead to an action.

RESTless Development

When it was announced that Rails was moving toward REST, I must admit I was somewhat resistant. I preferred to use URLs that had been traditional in the Rails world until then, naming the controller and action, as well as an object ID, in the URL. Thus, if I wanted to retrieve one person's address, I would use a URL such as:

```
/people/get_address/2341
```

where 2341 would be the unique ID for that person. And, for the most part, this paradigm worked just fine.

But, then I discovered what many Rails developers find out: that Rails is, as it claims to be, "opinionated" software, meaning that things are extremely easy if you do them the way it was designed, and very difficult if you try it in any other way. And as time went on, it became clear that my non-REST URLs were giving me problems. Many elements of Rails were no longer working for me, because they depended on REST. When I started to work with Backbone.js in 2011, I found that Backbone works with a variety of back ends, including Rails, but that using a non-REST interface was clumsy to work with, if it was even possible to use at all.

Thus, I began to get REST religion and tried to make every application I wrote conform to this sort of API, and it worked in many ways. The APIs were consistent, and my code was consistent. Because I was using the scaffolding that Rails provided—if only at the start of a project—the code was smaller and more consistent than otherwise would have been the case. In the case of Rails, which dynamically creates identifiers that represent specific URLs (for example, addresses), sticking with the RESTful routes really simplified things.

That is, until it didn't do so. REST, at least in the Rails implementation, gives you a single HTTP request method for each action you want to perform on your resource. This works well, in my experience, until you want to retrieve resources based on parameters, at which point things can get a bit complicated. Sure, you can pass parameters in the HTTP request, but at a certain point, I personally would rather have several small methods than one method with a huge set of if-then statements reflecting different combinations of parameters.

Thus, I've pulled back a bit from my REST absolutism, and I feel like I've reached something of a balance. For creation, updating and deletion, I'm totally fine with using the RESTful paradigm. But when it comes to retrieving resources from a Web application, I've relaxed my requirements, trying to make my API be as expressive and flexible as possible, without creating a huge number of routes or actions in my controller.

For example, I'm totally fine with retrieving information about a single user using the /users URL, tacking on an ID number to get information about a specific one. But, I often want to implement a search system to look for people in the system whose names match a particular pattern. Back in my pre-REST days, I would have used a search controller and had one or more methods that performed a search. In my neo-REST world, I simply add a "search" method to my "users" resource, such that the URL /users/search represents a search. Is that breaking REST? Absolutely. But, I've found that it makes my API more understandable and maintainable for my users, as well as for myself.

Rails itself, as opinionated and RESTful as it might be, offers you an out in the routes file (config/routes.rb). A recent project on which I worked had the following route:

```
resources :articles
```

This translates into the usual combination of RESTful routes and HTTP request methods. But when I decided to add three additional API calls to grab particular types of articles, I was able to do this by adding a block to the resources declaration:

```
resources :articles do
    get :latest, :on => :collection
    get :article_links, :on => :member
    get :stories, :on => :collection
end
```

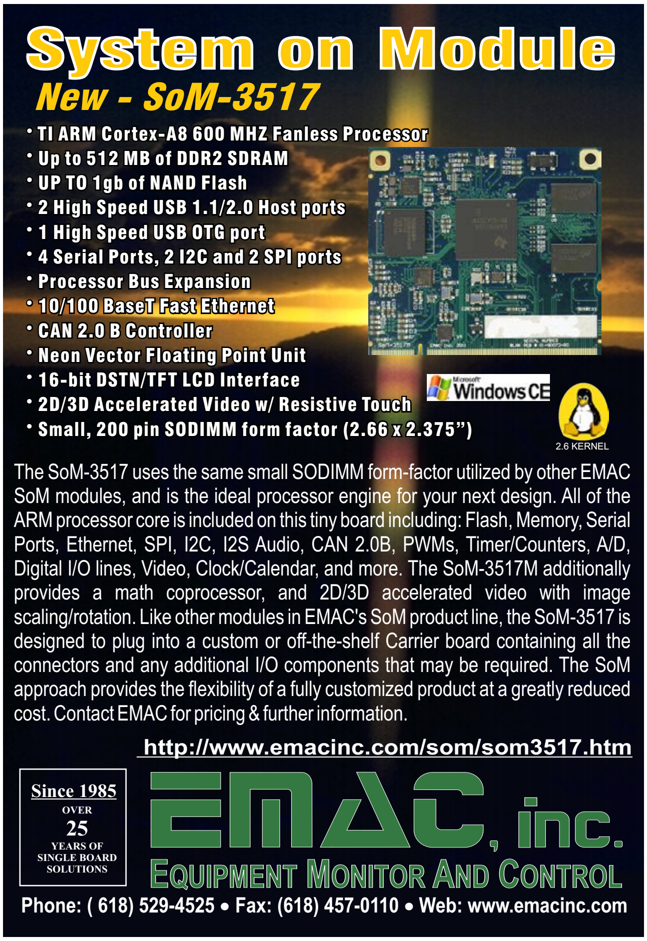
Not only does Rails offer this possibility, but it differentiates between "member" routes (which require a resource ID) and "collection" routes (which don't require a resource ID, because they operate on the entire collection).

General Practices

Once you've set up your API-naming convention, you need to consider what data you'll be passing. This means deciding on a data format, the parameters you'll receive in that format and the response you'll send back in that format.

When it comes to formats, there are really two main players nowadays: XML and JSON. As I mentioned previously, XML is very popular among enterprise users, but JSON has become very popular because of how easily you can transform objects from such languages as Python and Ruby into JSON (and back). In addition, JSON is nearly as self-documenting as XML, without the huge textual overhead or the complexity of an XML parser. Like many other people, I've switched to JSON for all of my API needs, and I haven't regretted it at all.

That said, Rails offers the option of responding in any of several different formats with the respond_to block. It basically lets you say, "if users want JSON,



do A, but if they want XML, then do B."

As for the request and response parameters, I try to keep it pretty simple. However, there's one parameter you absolutely should include in any API you create, and that's a version number. Your API presumably will evolve and improve over time, adding and removing method names, parameters and parameter types. By passing a version number along with your parameters, you can ensure that you're getting the parameters you require, and that both the client's and server's expectations are in sync.

Moreover, I've found that you can use that version number as a key in a hash of parameter lists. That is, rather than having separate variables in your server-side program that track which parameters are expected for each version, you can have a single hash whose keys are version numbers and whose values are arrays of expected parameters. You even can go one step further than this, having a multilevel hash that tracks different parameters for various API calls. For example, consider the following:

```
EXPECTED_PARAMS = { 'location' => {
    1 => ['longitude', 'latitude', 'altitude'],
    2 => [longitude', 'latitude', 'altitude', 'speed', 'timestamp'],
  },

  'reading' => {
    1 => ['time', 'area', 'mean', 'good', 'number'],
    2 => ['time', 'area', 'mean', 'good', 'number', 'seen', 'span',
    ➥'stdDev']
  }
}
```

Then, you can do something like the following:

```
version_number = params['version_number'].to_i
method_name = params['action']
required_fields = EXPECTED_PARAMS[method_name][version_number]
```

This is far easier than a lot of if-then statements, and it allows me to centralize the definition of what I expect to receive from my API clients. If I wanted to validate the data further, I could make each array element a hash rather than a string, listing one or more criteria that the data would need to pass.

Finally, the response that you give to your API users is your public face to the world. If you provide useless error messages, such as "Something went wrong", you'll presumably discover that developers are less than happy to use your system or develop on top of it. But if you provide detailed responses when things go well and poorly, not only will developers enjoy your system, but their end users will as well.■

Reuven M. Lerner is a longtime Web developer, architect and trainer. He is a PhD candidate in learning sciences at Northwestern University, researching the design and analysis of collaborative on-line communities. Reuven lives with his wife and three children in Modi'in, Israel.

DAVE TAYLOR

A Word Finder for *Words With Friends*—Continued

Three months after starting his project, Dave completes his *Scrabble*-helper script with a rather ingenious algorithm and some observations about where to go from here.

For my last few articles, we've been looking at *Scrabble* and *Scrabble*-like games, exploring different ways to work with a dictionary and pattern matching. The first installment focused on a crossword-puzzle word finder, something that proved surprisingly easy once we identified the regex pattern of <letter> <questionmark>, as in "??AR??C?S,", to match words like "Starbucks".

My last article, however, opened up the far more complicated world of *Scrabble*, wherein it became obvious that there's not going to be a simple solution to "find all the words that I can make out of the letters S R C R A E M" or similar.

The real challenge was letter frequency: we can make the word "RACE" out of the letters above, but can we make the word "ERASE"? We can't, because there's only one occurrence of the letter e, but that's a tricky thing to ascertain in a shell script.

As a result, we came up with a shell function that calculates how many times a letter occurs in a word as a simple way to test for out-of-bounds results:

```
occurrences()
{
  # how many times does 'letter' occur in 'word'?
  local count=$( echo $2 | grep -o $1 | wc -l )
  echo $count
}
```

We were using that in conjunction with a script called `findword` that extracts words from our previously downloaded word dictionary that match the set of seven letters, constrained to just those that contain five or more letters.

The basic idea is we're going to test each possible word (identified earlier and saved in a temp file) by stepping through each letter, calculating both the occurrences of the letter in that word and in the original set of letters.

With this set of letters, we'd have:

```
access is a possibility -- length = 6
accesses is a possibility -- length = 8
acers is a possibility -- length = 5
acmes is a possibility -- length = 5
acres is a possibility -- length = 5
```

It's immediately clear that these aren't in fact all valid possibilities because of that darn letter-frequency problem. We have one c and one s, how can "accesses" be possible? Let's fix it.

Screening Against Letter Frequency

Here's a straightforward way to calculate the frequency of each letter in our pattern:

```
while [ $idx -lt 8 ] ; do
  letter=$(echo $1 | cut -c$idx) ; occurrences $letter $1
  echo letter $letter occurs $freq times in $1
  idx=$(( $idx + 1 ))
done
```

Note that we've had to tweak the `occurrences` script to set the global variable `$freq` to be the frequency of the given letter in the pattern. It's sloppy, but as with most shell script programming, this is intended to be more of a quick prototype than a grand elegant solution.

Running this on our pattern produces:

```
letter s occurs 1 times in srcraem
letter r occurs 2 times in srcraem
letter c occurs 1 times in srcraem
letter r occurs 2 times in srcraem
letter a occurs 1 times in srcraem
letter e occurs 1 times in srcraem
letter m occurs 1 times in srcraem
```

We can add some code to make the script more efficient by removing duplicate tests (for example, r should be tested only once), but we can skip that concern because of how our final approach folds that into the solution. Plus, the next step is the interesting code portion, where we'll use this data to test possible words against letter frequency in the original pattern.

The basic idea is we're going to test each possible word (identified earlier and saved in a temp file) by stepping through each letter, calculating both the occurrences of the letter in that word and in the original set of letters. If the possible word has more, it's a fail. If it has less or the same, it's a continued possibility.

Here's how that looks in code:

```
for word in $(cat $possibilities)
do
  length=$(echo $word | wc -c)
  idx=1
  while [ $idx -lt $length ] ; do
    letter=$(echo $word | cut -c$idx)
    occurrences $letter $word
    wordfreq=$freq  # number of times letter occurs #1
    occurrences $letter $1  # and letter occurrences #2
    if [ $wordfreq -gt $freq ] ; then
      echo discarding $word because $letter occurs too \
         many times vs pattern
      break  # get out of the "nearest" loop
    else
      echo ... continuing to test $word, $letter ok
    fi
    idx=$(( $idx + 1 ))   # increment loop counter
  done
done
```

It's a rather complicated piece of code, but let's run it and see what happens, then I'll explain a bit more about what's going on:

```
testing word access from possibilities file
... continuing to test access, a freq is acceptable
discarding access because c occurs too many times vs pattern
testing word accesses from possibilities file
... continuing to test accesses, a freq is acceptable
discarding accesses because c occurs too many times vs pattern
```

To start out, it has correctly identified that neither ACCESS nor ACCESSES are actually possible matches to our original set of letters because the letter c occurs too many times in both cases. What about a word that is valid?

```
testing word acers from possibilities file
... continuing to test acers, a freq is acceptable
... continuing to test acers, c freq is acceptable
... continuing to test acers, e freq is acceptable
... continuing to test acers, r freq is acceptable
... continuing to test acers, s freq is acceptable
```

By not failing out after the last letter, we can conclude that ACERS is indeed a valid word that can be created from the original set of letters.

Great. So we're close to a solution. Let's add a bit of code logic to have it know when it's succeeded at testing each and every letter of a word without a fail, and get rid of these interim status messages. Here's the result:

```
-- word access was skipped (too many c)
-- word accesses was skipped (too many c)
++ word acers can be constructed from the letters srcraem
++ word acmes can be constructed from the letters srcraem
++ word acres can be constructed from the letters srcraem
```

We're actually done with the algorithm and the problem is solved. We just need to clean things up.

Awesome. We're actually done with the algorithm and the problem is solved. We just need to clean things up. Here's what I did to the code for the output to look pretty:

```
for word in $(cat $possibilities)
do
  length=$(echo $word | wc -c); length="$(( $length - 1 ))"
  idx=1
  while [ $idx -lt $length ] ; do
    letter=$(echo $word | cut -c$idx)
    occurrences $letter $word
    wordfreq=$freq  # number of times letter occurs #1
    occurrences $letter $1 # and letter occurrences #2
    if [ $wordfreq -gt $freq ] ; then
      echo "-- word $word was skipped (too many $letter)"
      break # get out of the "nearest" loop
    else
      if [ $idx -eq $length ] ; then
        echo "++ word $word can be constructed from \
          the letters $1"
      fi
    fi
    idx=$(( $idx + 1 )) # increment loop counter
  done
done
```

I haven't changed a lot actually, other than the conditional test when the letter occurrence is acceptable to see if our index = the length of the word.

Want to see only the valid possibilities and not the words that were discarded because of letter frequency? That's easy enough, just add a #? before the appropriate echo statement to comment it out.

And, finally, here's a list of all the five or more letter words you could produce from the original letter list SRCRAEM: acers, acmes, acres, cames, carer, carers, cares, carrs, carse, crams, crare, crares, cream, creams, macer, macers, maces, marcs, mares, maser, racer, racers, races, reams, rearm, rearms, rears, scare, scarer, scrae, scram, scream, serac, serra, smear.

Now you know.

Me? I'd play "racers". It's not as offbeat as the other words that the program produced.

In fact, it'd be interesting to sort the results by length or, better, by score, since different letters have different point values in *Scrabble*. Hmmm...■

Dave Taylor has been hacking shell scripts for more than 30 years. Really. He's the author of the popular *Wicked Cool Shell Scripts* and can be found on Twitter as @DaveTaylor and more generally at http://www.DaveTaylorOnline.com.

KYLE RANKIN

The Sysadmin's Toolbox: sar

If your server has high load when no sysadmin is logged in, use sar to find out what happened.

As someone who's been working as a system administrator for a number of years, it's easy to take tools for granted that I've used for a long time and assume everyone has heard of them. Of course, new sysadmins get into the field every day, and even seasoned sysadmins don't all use the same tools. With that in mind, I decided to write a few columns where I highlight some common-but-easy-to-overlook tools that make life as a sysadmin (and really, any Linux user) easier. I start the series with a classic troubleshooting tool: sar.

There's an old saying: "When the cat's away the mice will play." The same is true for servers. It's as if servers wait until you aren't logged in (and usually in the middle of REM sleep) before they have problems. Logs can go a long way to help you isolate problems that happened in the past on a machine, but if the problem is due to high load, logs often don't tell the full story. In my March 2010 column "Linux Troubleshooting, Part I: High Load" (**http://www.linuxjournal.com/article/10688**), I discussed how to troubleshoot a system with high load using tools such as uptime and top. Those tools are great as long as the system still has high load when you are logged in, but if the system had high load while you were at lunch or asleep, you need some way to pull the same statistics top gives you, only from the past. That is where sar comes in.

Enable sar Logging

sar is a classic Linux tool that is part of the sysstat package and should be available in just about any major distribution with your regular package manager. Once installed, it will be enabled on a Red Hat-based system, but on a Debian-based system (like Ubuntu), you might have to edit /etc/default/sysstat, and make sure that `ENABLED` is set to true. On a Red Hat-based system, sar will log seven days

There's an old saying: "When the cat's away the mice will play." The same is true for servers.

of statistics by default. If you want to log more than that, you can edit /etc/sysconfig/sysstat and change the `HISTORY` option.

Once sysstat is configured and enabled, it will collect statistics about your system every ten minutes and store them in a logfile under either /var/log/sysstat or /var/log/sa via a cron job in /etc/cron.d/sysstat. There is also a daily cron job that will run right before midnight and rotate out the day's statistics. By default, the logfiles will be date-stamped with the current day of the month, so the logs will rotate automatically and overwrite the log from a month ago.

CPU Statistics

After your system has had some time to collect statistics, you can use the sar tool to retrieve them. When run with no other arguments, sar displays the current day's CPU statistics:

```
$ sar
. . .
07:05:01 PM  CPU  %user  %nice  %system  %iowait %steal  %idle
. . .
08:45:01 PM  all   4.62   0.00    1.82     0.44   0.00   93.12
08:55:01 PM  all   3.80   0.00    1.74     0.47   0.00   93.99
09:05:01 PM  all   5.85   0.00    2.01     0.66   0.00   91.48
09:15:01 PM  all   3.64   0.00    1.75     0.35   0.00   94.26
Average:     all   7.82   0.00    1.82     1.14   0.00   89.21
```

If you are familiar with the command-line tool top, the above CPU statistics should look familiar, as they are the same as you would get in real time from top. You can use these statistics just like you would with top, only in this case, you are able to see the state of the system back in time, along with an overall average at the bottom of the statistics, so you can get a sense of what is normal. Because I devoted an entire previous column to using these statistics to troubleshoot high load, I won't rehash all of that here, but essentially, sar provides you with all of the same statistics, just at ten-minute intervals in the past.

RAM Statistics

sar also supports a large number of different options you can use to pull out other statistics. For instance, with the `-r` option, you can see RAM statistics:

```
$ sar -r
. . .
07:05:01 PM kbmemfree kbmemused %memused kbbuffers  kbcached
```

```
kbcommit  %commit
. . .
08:45:01 PM    881280   2652840     75.06    355284   1028636
8336664    183.87
08:55:01 PM    881412   2652708     75.06    355872   1029024
8337908    183.89
09:05:01 PM    879164   2654956     75.12    356480   1029428
8337040    183.87
09:15:01 PM    886724   2647396     74.91    356960   1029592
8332344    183.77
Average:       851787   2682333     75.90    338612   1081838
8341742    183.98
```

Just like with the CPU statistics, here I can see RAM statistics from the past similar to what I could find in top.

Disk Statistics

Back in my load troubleshooting column, I referenced sysstat as the source for a great disk I/O troubleshooting tool called iostat. Although that provides real-time disk I/O statistics, you also can pass sar the -b option to get disk I/O data from the past:

```
$ sar -b
. . .
07:05:01 PM     tps     rtps     wtps   bread/s    bwrtn/s
. . .
08:45:01 PM    2.03     0.33     1.70      9.90      31.30
08:55:01 PM    1.93     0.03     1.90      1.04      31.95
09:05:01 PM    2.71     0.02     2.69      0.69      48.67
09:15:01 PM    1.52     0.02     1.50      0.20      27.08
Average:       5.92     3.42     2.50     77.41      49.97
```

I figure these columns need a little explanation:

- tps: transactions per second.
- rtps: read transactions per second.
- wtps: write transactions per second.
- bread/s: blocks read per second.
- bwrtn/s: blocks written per second.

sar can return a lot of other statistics beyond what I've mentioned, but if you want to see everything it has to offer, simply pass the -A option, which will return a complete dump of all the statistics it has for the day (or just browse its man page).

Turn Back Time

So by default, sar returns statistics for the current day, but often you'll want to get information a few days in the past. This is especially useful if you want to see whether today's numbers are normal by comparing them to days in the past, or if you are troubleshooting a server that misbehaved over the weekend. For instance, say you noticed a problem on a server today between 5PM and 5:30PM. First, use the -s and -e options to tell sar to display data only between the start (-s) and end (-e) times you specify:

So by default, sar returns statistics for the current day, but often you'll want to get information a few days in the past.

```
$ sar -s 17:00:00 -e 17:30:00
Linux 2.6.32-29-server (www.example.net)  02/06/2012   _x86_64_
(2 CPU)

05:05:01 PM  CPU  %user  %nice %system %iowait  %steal  %idle
05:15:01 PM  all   4.39   0.00    1.83    0.39    0.00   93.39
05:25:01 PM  all   5.76   0.00    2.23    0.41    0.00   91.60
Average:     all   5.08   0.00    2.03    0.40    0.00   92.50
```

To compare that data with the same time period from a different day, just use the -f option and point sar to one of the logfiles under /var/log/sysstat or /var/log/sa that correspond to that day. For instance, to pull statistics from the first of the month:

```
$ sar -s 17:00:00 -e 17:30:00 -f /var/log/sysstat/sa01
Linux 2.6.32-29-server (www.example.net)  02/01/2012   _x86_64_
(2 CPU)

05:05:01 PM  CPU  %user  %nice  %system  %iowait %steal  %idle
05:15:01 PM  all   9.85   0.00     3.95     0.56   0.00   85.64
05:25:01 PM  all   5.32   0.00     1.81     0.44   0.00   92.43
Average:     all   7.59   0.00     2.88     0.50   0.00   89.04
```

You also can add all of the normal sar options when pulling from past logfiles, so you could run the same command and add the -r argument to get RAM statistics:

```
$ sar -s 17:00:00 -e 17:30:00 -f /var/log/sysstat/sa01 -r
Linux 2.6.32-29-server (www.example.net)  02/01/2012   _x86_64_
(2 CPU)

05:05:01 PM kbmemfree kbmemused  %memused kbbuffers  kbcached
kbcommit  %commit
05:15:01 PM    766452   2767668     78.31    361964   1117696
8343936    184.03
05:25:01 PM    813744   2720376     76.97    362524   1118808
8329568    183.71
Average:       790098   2744022     77.64    362244   1118252
8336752    183.87
```

As you can see, sar is a relatively simple but very useful troubleshooting tool. Although plenty of other programs exist that can pull trending data from your servers and graph them (and I use them myself), sar is great in that it doesn't require a network connection, so if your server gets so heavily loaded it doesn't respond over the network anymore, there's still a chance you could get valuable troubleshooting data with sar.■

Kyle Rankin is a Sr. Systems Administrator in the San Francisco Bay Area and the author of a number of books, including *The Official Ubuntu Server Book*, *Knoppix Hacks* and *Ubuntu Hacks*. He is currently the president of the North Bay Linux Users' Group.

SHAWN POWERS

LTSP, Part II: Tweaking the Beast

LTSP5 allows thin clients to be customized unlike ever before. This article shows you how.

In my last column, I walked through the process of setting up an LTSP server and explained the boot process for thin clients. If you need only a simple thin-client environment consisting of a single server with a few clients, that article is really all you need. If you want more power from your LTSP setup, Part II of the series is where things start to get really cool. I'm going to assume you have a basic LTSP server set up and that your thin clients are booting successfully.

The Oddly Placed Config File

LTSP thin clients don't need a configuration file in order to work. To take advantage of the really cool things it can do, however, you need to create one. The file is called lts.conf, and it lives in your tftp root directory. For Ubuntu running 32-bit clients, this means /var/lib/tftpboot/ltsp/i386/. Although it may seem like a strange place for a config file to live, keep in mind that the entire LTSP chroot is converted into an NBD block image. Changing the configuration within that image takes a lot of work, so instead of keeping the config inside the LTSP chroot, it's next to the kernel image and fetched during bootup from tftp. That means any changes made to lts.conf are picked up by the thin client on reboot without the need to re-create the chroot NBD image.

Oodles of configuration options can be set within lts.conf. To get a description of them, be sure you have the ltsp-docs package installed, and type `man lts.conf` at the command line. The format is fairly simple. Everything under the `[default]` section will be applied to all thin clients. For options pertaining only to certain devices, a new section is created in square brackets with either the device's IP address or the MAC address—either works. Here is my lts.conf file:

```
# This is a sample lts.conf file
# It's stored in /var/lib/tftpboot/ltsp/i386/
```

```
[default]
VOLUME = 50
LDM_DIRECTX = True
LDM_XSESSION = "gnome-session session=classic-gnome"
CUPS_SERVER = 192.168.1.100
DNS_SERVER = 192.168.1.1
SEARCH_DOMAIN = example.com

# The following is a powerful thin client
[00:a4:5a:44:42:ff]
LOCAL_APPS = True
LOCAL_APPS_MENU = True
LOCAL_APPS_MENU_ITEMS = firefox,vlc,tuxtype
```

Let's start with the [default] section. The man page for lts.conf will give you an explanation for all the options I've set here, but a couple aren't obvious even after reading the description. The VOLUME setting is fairly easy to understand. For some reason, the thin clients I use are maxed out by default, and the Ubuntu login sound rattles windows and can knock out loose dental fillings on startup. It's loud. By setting the VOLUME level in lts.conf to 50, the volume is happily at half its normal level. Users can change the level, but on every reboot, it is set back to 50%.

The next line is LDM_DIRECTX. This line tells the client to forgo the normal encrypted connection to the server and connect directly via X. This is less secure, but it saves on CPU load. You'll have to determine which is more important in your environment. Keep in mind some older thin clients will really slow down unless you set this option.

LDM_XSESSION tells the thin client, or more precisely the display manager LDM, what type of session to load. In Ubuntu 11.04, I force this to the classic GNOME desktop. That's not an option with more recent versions of Ubuntu, but I leave the option here so you can see it if you want to tweak it for other less-common desktop environments.

The printers can be installed on each server, or if you have a central CUPS server, you can specify it here with the CUPS_SERVER directive. This makes it very simple to set up printing on multiple LTSP servers. Because there is nothing to install, just point to the existing CUPS server, and you're ready to go. (This also applies for local apps, which I'll cover in a minute.)

The DNS_SERVER and SEARCH_DOMAIN settings usually aren't needed. There was a strange bug for years that required the settings in some circumstances. If you have problems resolving DNS on your thin client for whatever reason, setting these options is a good thing to try. I leave them in out of habit more than anything; however, having them here certainly doesn't hurt.

The next few lines are specified for a specific thin client. I actually have these settings in my [default] section and turn them off for particularly old thin clients, but this way I can demonstrate how to apply settings to a specific device as well. The device specified by the MAC address will take all the `[default]` settings, then apply the settings in its section as well. It's important to note that you can override settings in this manner. If you wanted the volume for this specified device to be 100%, you could add `VOLUME = 100` to this section. I'll refer back to the `LOCAL_APPS` declarations in the next section.

Local Apps, the LTSP Secret Sauce

One of the more recent issues facing thin-client proponents are inexpensive workstations. That might seem backward, but although management of thin clients is much nicer on a sysadmin, it's hard to convince users a thin client is better than a $299 computer from Walmart. In fact, if you purchase a new Atom-based thin client, it's running the same CPU and RAM a Netbook uses, most of which run Microsoft Windows! Even low-power thin-client devices are quite powerful, so LTSP takes advantage of that power by offloading some of the work to the thin client itself.

This concept is a little rough to visualize, but if you understand tunneling X11 applications over SSH, it shouldn't be too bad. Like I explained in Part I of this series, the local thin client loads a very stripped-down version of Linux over the network. It has enough to start X11 and then connect to the server where it runs all its applications. This is sort of like running all your applications over an SSH tunnel from a powerful remote server. With local apps, you're still doing that, but instead of running all apps on the server, a few of the memory/CPU-heavy ones are run locally. For the end user, this is completely transparent. As the sysadmin, however, you'll appreciate Firefox with Flash running on the thin client instead of having 50 copies of Firefox with Flash running on your server.

Preparing the chroot

The first thing you need to do in order to use local apps is install the apps locally! Because the NBD image is created from the chroot environment that lives in /opt/ltsp/i386, the first step is to chroot to that, update the apt repositories and install some apps (note that you also mount the proc system once inside the chroot environment):

```
sudo chroot /opt/ltsp/i386
mount -t proc proc /proc
apt-get update
apt-get install firefox adobe-flashplugin
exit
sudo umount /opt/ltsp/i386/proc
sudo ltsp-update-image
```

A quick walk-through shows that I installed Firefox and Adobe Flash inside the chroot. Then I exited the chroot and ran `ltsp-image-update`, which creates a new NBD image from the chroot environment. Remember the chroot isn't mounted directly, so changes made to the chroot take effect only after a new NBD image is created and the thin clients are restarted. (If you get an error about the Flash plugin not existing, you may have to add the Ubuntu partner repository to the sources.list inside your chroot.)

Now that Firefox is installed locally, peek back at the lts.conf file from earlier. The three lines look similar, but there are subtle differences. `LOCAL_APPS` is the directive that makes local apps possible. Without this set, the thin client will always launch applications from the server, and not its own system. `LOCAL_APPS_MENU` goes one step further and tells the thin clients it can rewrite the menu system so that specified applications are launched from the local computer when selected from the menu. The final directive, `LOCAL_APPS_MENU_ITEMS`, tells the thin client which applications are to be run from the local machine and not the server. With the settings I've listed, any time a user on the specified machine selects Firefox from the menu, it will launch Firefox as a local application. The confusing part is that for the end user, it won't be any different from launching it from a thin client *not* configured for local apps. The difference is easy to tell when a lab of users starts using an Adobe Flash site, however, because instead of 30 instances of Flash running on the server, each thin client runs it locally.

Consequences, Consequences

This new way of launching applications locally seems simple once you wrap your brain around what's actually happening. It does introduce a few frustrating things, however. Because the application (let's take Firefox as an example) is running on the thin client, any Web plugins required will have to be installed in the chroot. This also means printing has to be working inside the chroot, because Firefox is running locally, in order to print. Thankfully, that `CUPS_SERVER` directive takes care of things if you have a separate print server. If you have a locally installed printer on the server, you need to make sure your CUPS settings allow for other machines to connect. In that case, the thin-client local apps connect to the server's own CUPS server to print.

There are other complications, like downloaded files. Where do they go? Thankfully, the thin client automatically mounts the user's home directory over SSH, so downloads are stored in the user's folder on the server, even though they're downloaded with Firefox as a local app. Local apps are a powerful way to utilize your thin client's hardware, but keeping in mind the complications involved might help you determine when local apps are appropriate. Sometimes, they're not.

Just Because You Can, Doesn't Mean You Should

A few apps might seem logical to run as local apps to save strain on the server. One of those apps is OpenOffice.org (or LibreOffice in newer distros). Interestingly enough, some applications run so well in the shared server environment, it's silly to run them as local apps. LibreOffice is one of those. Granted, the office suite uses a lot of resources and can hog memory, but with concurrent users, that memory usage is shared by many, many people. In fact, once one user is running LibreOffice, subsequent users can start the application almost instantly, because the program is already in RAM.

The moral of the local app story is to identify which applications are bogging down your server and then see how they behave as local apps. Every LTSP release gets better and better at handling local apps, so if you were burned several releases ago, I recommend giving it a try now.

Tips & Tricks

SSH Keys One nifty feature of the new LTSP chroot model is that it's possible to tweak your thin client's system any way you like. For me, that includes activating the root user and adding public key authentication for that root user. Creating SSH keys for auto-login is something we've covered in *Linux Journal* many times before, but setting it up inside your chroot environment might not be something you considered.

If you have the ability to log in to your thin client over SSH (again, this requires installing openssh-server inside the chroot), you easily can reboot a thin client remotely. You also can log in and see if a particular local app is bogging down the thin client. And, because all thin clients share a single chroot environment, it means you can log in to *any* thin client automatically if you set up SSH keys! This means you can shut down a lab with a single bash loop or reboot a failing thin client without ever leaving your desk. SSH is awesome, and with LTSP, it's easy to set up.

VNC on Clients If the ability to SSH in to a thin client is useful, the ability to control a thin-client screen remotely is absolutely invaluable. This is slightly more complicated, but well worth the effort (thanks to bootpolish.net for originally posting this method).

Log in to chroot:

```
sudo chroot /opt/ltsp/i386
mount -t proc proc /proc
```

Then, install x11vnc:

```
apt-get update
apt-get install x11vnc
```

Next, create a vnc password file. You'll need to remember this when you want to connect:

```
echo "yourpassword" > /etc/vncpassword
chmod 400 /etc/vncpassword
chown root:root /etc/vncpassword
```

Then, you can exit the chroot:

```
Exit
sudo umount /opt/ltsp/i386/proc
```

And the last step, which is a little tricky, is to create two files. These files tell LDM (the thin client's display manager) to start and stop VNC. The first file is /opt/ltsp/i386/usr/share/ldm/rc.d/I99-x11vnc:

```
# /opt/ltsp/i386/usr/share/ldm/rc.d/I99-x11vnc
# LDM Script to start x11vnc

XAUTH=`find /var/run/ldm-xauth* -type f`
start-stop-daemon --start --oknodo --pidfile /var/run/x11vnc.pid
➥--background --nicelevel 15 --make-pidfile --exec
➥ /usr/bin/x11vnc --
➥-display :7 -loop -passwdfile /etc/vncpassword -nossl -logfile
➥/var/log/x11vnc -auth $XAUTH
```

Note that the `start-stop-daemon` line is one continuous line ending with $XAUTH.

The second file is /opt/ltsp/i386/usr/share/ldm/rc.d/X99-x11vnc:

```
# /opt/ltsp/i386/usr/share/ldm/rc.d/X99-x11vnc
# LDM Script to stop x11vnc
start-stop-daemon --stop --oknodo --pidfile /var/run/x11vnc.pid
```

Once created, run `sudo ltsp-update-image`, and any rebooted thin clients should be accessible over VNC using the password you created in the second step. It's a complex setup, but definitely worth it.

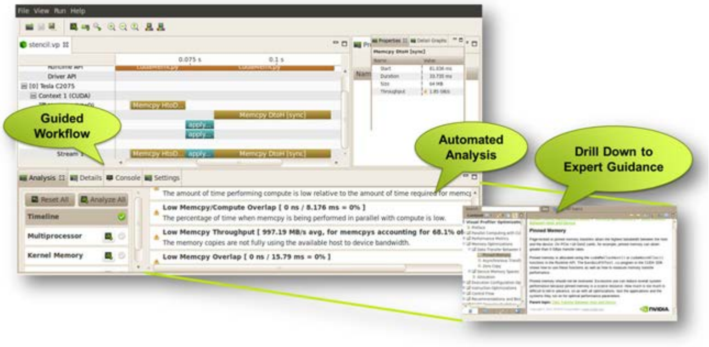
Until Next Time...

I threw a lot of LTSP information at you in this article. If you got lost, or felt overwhelmed, my apologies. Hopefully with a few reviews, it will all make sense. If nothing else, I urge you to try getting local apps working. Harnessing the power of modern thin clients is a great way to stretch your server budget, because with the thin clients running programs like Firefox, your server can handle many more actual thin-client connections.

My next column will finish up the LTSP series as I discuss scaling. LTSP can be scaled in multiple ways, all of which have advantages and disadvantages. The great thing about Linux is that regardless of the number of thin clients you add, you'll never have to pay license fees!■

Shawn Powers is the Associate Editor for *Linux Journal*. He's also the Gadget Guy for LinuxJournal.com, and he has an interesting collection of vintage Garfield coffee mugs. Don't let his silly hairdo fool you, he's a pretty ordinary guy and can be reached via e-mail at shawn@linuxjournal.com. Or, swing by the #linuxjournal IRC channel on Freenode.net.

NVIDIA CUDA Toolkit



The amount of time performing compute is low relative to the amount of time required for memcp

Low Memcpy/Compute Overlap [0 ns / 8.176 ms = 0%]
The percentage of time when memcpy is being performed in parallel with compute is low.

Low Memcpy Throughput [997.19 MB/s avg, for memcpys accounting for 68.1% of
The memory copies are not fully using the available host to device bandwidth.

Low Memcpy Overlap [0 ns / 15.79 ms = 0%]

Pinned Memory

NVIDIA, a leading force in graphics processing units (GPUs), recently released a major new update to the CUDA parallel computing platform. This new version features three key enhancements that make parallel programing with GPUs easier, more accessible and faster, improving options for scientists and engineers seeking to advance their simulations and research. These enhancements include a redesigned Visual Profiler with automated performance analysis, a new and nimbler compiler based on the LLVM open-source compiler infrastructure and hundreds of new imaging and signal-processing functions, with the latter doubling the size of the NVIDIA Performance Primitives library. The latest CUDA release is available for free download for Linux, Mac OS and Windows from NVIDIA's Web site.
http://www.nvidia.com/cuda

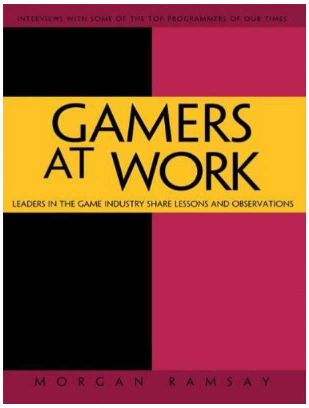
CloudPassage's Halo NetSec

A survey by cloud-computing company CloudPassage found the top concern of 45% of IT managers to be a lack of perimeter defenses in their public cloud. This lack of solutions is exacerbated by the fluctuating nature of on-demand computing. To the rescue comes CloudPassage's new Halo NetSec, an automated solution that provides advanced network access control for servers running in public clouds. "Securing cloud servers doesn't have to be a full-time job", says CloudPassage. Halo NetSec incorporates a cloud-ready firewall, two-factor authentication and an open API for integration with other systems. It is purpose-built for the cloud so it can scale up instantly to accommodate large development projects and back down again. The result is "a solution providing network security functions expected in private data centers, enabling cloud developers and security engineers to embrace IaaS computing with confidence".
http://www.CloudPassage.com

Caringo Object Storage Platform

Got (massive unstructured) data stores? Then Caringo has something just for you. Caringo Object Storage Platform 5.5 is the new, updated version of the company's three-application solution that "makes Big Data manageable via powerful reporting, increased serviceability and massive object size support." The Object Storage Platform is composed of the CAStor object storage engine that can now store files of unknown size via chunked encoding, as well as objects as large as 4 terabytes; the Content Router rules-based work-flow engine for metadata-driven global replication; and the Cluster Services Node Web-based management application. In version 5.5, each component of the platform was upgraded, including increased service and support functions, reporting and management capabilities, and the aforementioned support for objects up to 4 terabytes.
http://www.caringo.com

Morgan Ramsay's *Gamers at Work* (Apress)


INTERVIEWS WITH SOME OF THE TOP PROGRAMMERS OF OUR TIMES
GAMERS AT WORK
LEADERS IN THE GAME INDUSTRY SHARE LESSONS AND OBSERVATIONS
MORGAN RAMSAY

Tech startups fascinate us for many reasons. Among them are quirky tech-savvy personalities, impeccable (or lucky) timing and a penchant for garages. Morgan Ramsay's new book *Gamers at Work: Stories Behind the Games People Play* uncovers the founding stories of the most successful and fascinating game-company startups, including Linden Lab, Obsidian Entertainment, Atari, Persuasive Games, Appy Entertainment and Oddworld Inhabitants. Targeted at those considering setting up their own independent studios, as well as those interested in the history of the video-game industry, Ramsay's book explores the formation of entertainment software companies from the perspectives of successful founders who risked it all. It offers insights into why experienced professionals sacrifice the comfort of gainful employment for the uncertainty and risk of the startup and shares the experiences and lessons that shape the lives, decisions and struggles of entrepreneurs in this volatile business. Will you be the next gaming sensation, or will you crash and burn?
http://www.informit.com

Digi's ConnectPort X2e for Smart Energy

The greener Smart Grid is coming, but scaling it will take open architecture, lower cost and improved management of the dispersed device network. Digi International engages these issues with the help of its new Linux-powered ConnectPort X2e for Smart Energy gateway. The gateway connects ZigBee Smart Energy devices from a home area network to an energy service provider via broadband. A key advancement is the device's additional processing power and memory, which not only permits more efficient establishment and management of large deployments, but also enables support for more complex local energy-efficiency applications and the impending ZigBee Smart Energy 2.0 standard. As part of the open Digi X-Grid Solutions, the gateway can connect to numerous types of devices to enable a wide range of services. All Digi Smart Energy gateways feature the iDigi Device Cloud, allowing Smart Energy devices to integrate energy consumption data into new and existing Smart Energy applications easily and securely.

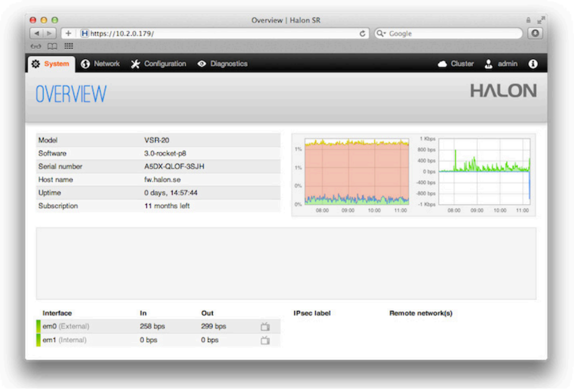
http://www.digi.com/x2e

Opengear ACM5500

According to device-maker Opengear, existing tools for remote monitoring and management tend to be complex and costly. Problem solved, says Opengear, thanks to the firm's new Opengear ACM5500 family of customizable remote monitoring and management gateways. The product line makes it convenient for managed service providers to monitor and manage their customers' network infrastructure remotely, whether on the road, in an office or at another customer site. Customized alert thresholds proactively track device status, and problems often can be fixed earlier than ever, allowing for satisfied customers and avoided crises. A key product feature is Opengear's dedicated secure hardware agent, which gives MSPs visibility into and the ability to monitor and control customers' network infrastructure devices even behind a firewall. Based on open standards, the Opengear ACM5500 product family complements the tools currently used to manage customers' IT environments (such as Cisco, Juniper, Avaya, F5 and so on), so there is no "rip and replace".

http://www.opengear.com

Halon Security Router Firewall



Model	VSR-20
Software	3.0-rocket-p8
Serial number	A5DX-QLOF-3SJH
Host name	fw.halon.se
Uptime	0 days, 14:57:44
Subscription	11 months left

Interface	In	Out	IPsec label	Remote network(s)
em0 (External)	258 bps	299 bps		
em1 (Internal)	0 bps	0 bps		

The aim of the new Halon Security Router Firewall is to bring the Internet firewall into the IPv6 age. Halon Security claims that despite the move to IPv6 looming on everyone's doorstep, "few firewall manufacturers seem to focus on the greatest change in how the world communicates since the Internet was born". The company's innovation is to build a custom, OpenBSD-based OS from the ground up with a focus on IPv6, security and performance. Core features include high-performance routing, wire-speed throughput and hardware-accelerated AES IPSec VPN, load balancing and failover. Customers can choose a hardware appliance, a virtualization environment or a standard server hardware solution, all of which offer identical security, performance and management capabilities. A free evaluation copy is available for download from Halon Security's Web site.

http://www.halonsecurity.com

Strategic Test's i.MX283 Computer-on-Module

Expanding on its line of Freescale-based i.MX Computer-on-Modules, Strategic Test recently announced the addition of the low-cost TX-28S COM for Linux and Windows CE 6.0. The little guy measures in at 68 x 25 mm (2.6" x 1") and contains a 454MHz i.MX283 processor coupled with 64MB DDR SDRAM, 128MB NAND Flash memory and a 200-pin SODIMM connector. The TX-28S, says its producer, is targeted at embedded fanless applications where low price, small size and low power consumption are critical factors. As a member of the well-received TX family of COMs, the TX-28S allows developers to design one baseboard that accepts different modules. This permits protecting design against obsolescence and scalability. Strategic Test also notes the advantage of Freescale Semiconductor's seven-year Longevity Program, which further prevents rapid obsolescence.

http://www.strategic-embedded.com

Please send information about releases of Linux-related products to newproducts@linuxjournal.com or New Products c/o *Linux Journal*, PO Box 980985, Houston, TX 77098. Submissions are edited for length and content.

HARDWARE

ASUS Transformer Prime

The Transformer Prime is a killer piece of hardware, no doubt... but can it replace a full-fledged Linux Netbook? AARON PETERS

The original Transformer was a unique concept put forth by ASUS in an effort to gain ground against the then-ubiquitous iPad. With its attachable keyboard, the tablet gained a fair amount of attention and performed well sales-wise to boot. But, although it had a "wow" factor with its keyboard accessory, some felt it lacked the build quality and style of its competition. So, ASUS went back to the drawing board, and the sequel device has all the advantages of the original with its detachable keyboard. It's wrapped in a thinner, lighter, better-looking case that's every bit as stylish as anything else on the market, and it adds kick-butt performance to boot.

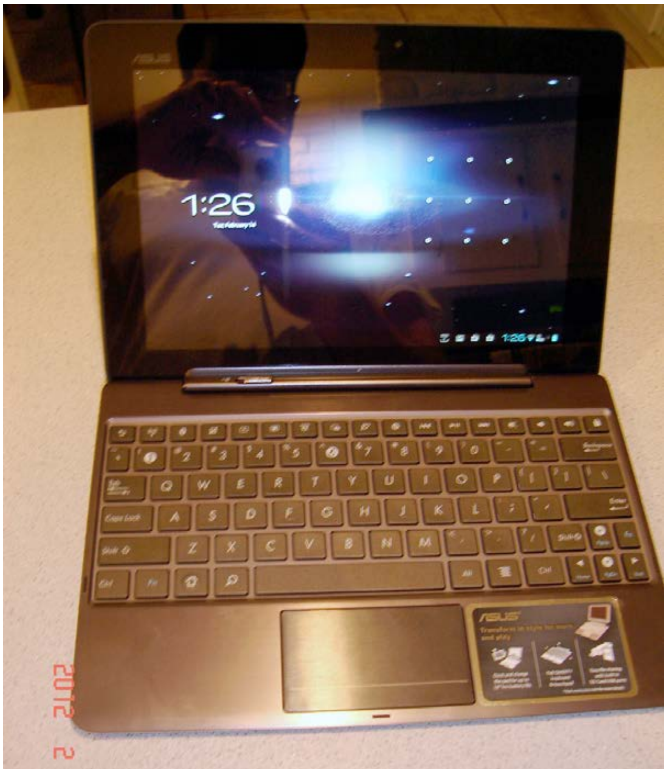


Figure 1. The Prime, in All Its Glory

Device Overview

The specs of the tablet portion of the Prime are, in most ways, common to a number of other devices on the market, including the following:

- Size/weight: 263 x 180.8 x 8.3mm; 586g.
- RAM: 1GB.

- Storage: 32GB Flash storage.
- Screen: 10", 1200px W x 800px H Gorilla Glass display (178° viewing angle).
- Power: 25Wh Li-polymer battery (est. 12-hour life).
- Controls: power switch and volume rocker control.
- I/O: 40-pin proprietary connector (charging via cable or keyboard dock); MicroSD card slot; 8MP, F2.4 rear-facing camera with flash; 1.2MP front-facing camera; Mini-HDMI port and 3.5mm headphone/microphone jack.

Connecting to the keyboard adds the additional features:

- Size/weight: 263 x 180.8 x 810.4mm; 537g.
- Power: 22Wh Li-polymer battery (est. additional 6-hour life).
- 73-key, 254mm island-style (that is, chiclet-style) keyboard.
- Multitouch touchpad/button (one-button).
- Full-size USB port.
- SD card slot.
- 40-pin male proprietary connector (for connection to/charging of tablet).
- 40-pin female proprietary connector (for charging the tablet and keyboard).

The main thing setting the Prime apart is its processor: the 1.2GHz Tegra 3—a Quad-core processor that impressed the tech media when NVIDIA first demonstrated it. The Prime has been the only mainstream tablet to feature this chip, and it provides the Prime with a nice boost under the hood.

Device Introduction and First Impressions

Before even cutting the plastic wrap on the Prime's box, one thing you notice is how svelte even its packaging is. The box also is blissfully uncluttered within, as the only things it contains are the tablet itself, a quick-start guide, the warranty form, a screen-cleaner cloth, and the power cable and plug block. You notice at once when you lift the tablet out of the box how solid it feels, to the point where it almost feels heavier than it is. The casing, which features the same circular brushed-aluminum design with more-recent ASUS ultrabooks and other machines, feels smooth. There is a little flex to the tablet's casing, but only if you squeeze it harder

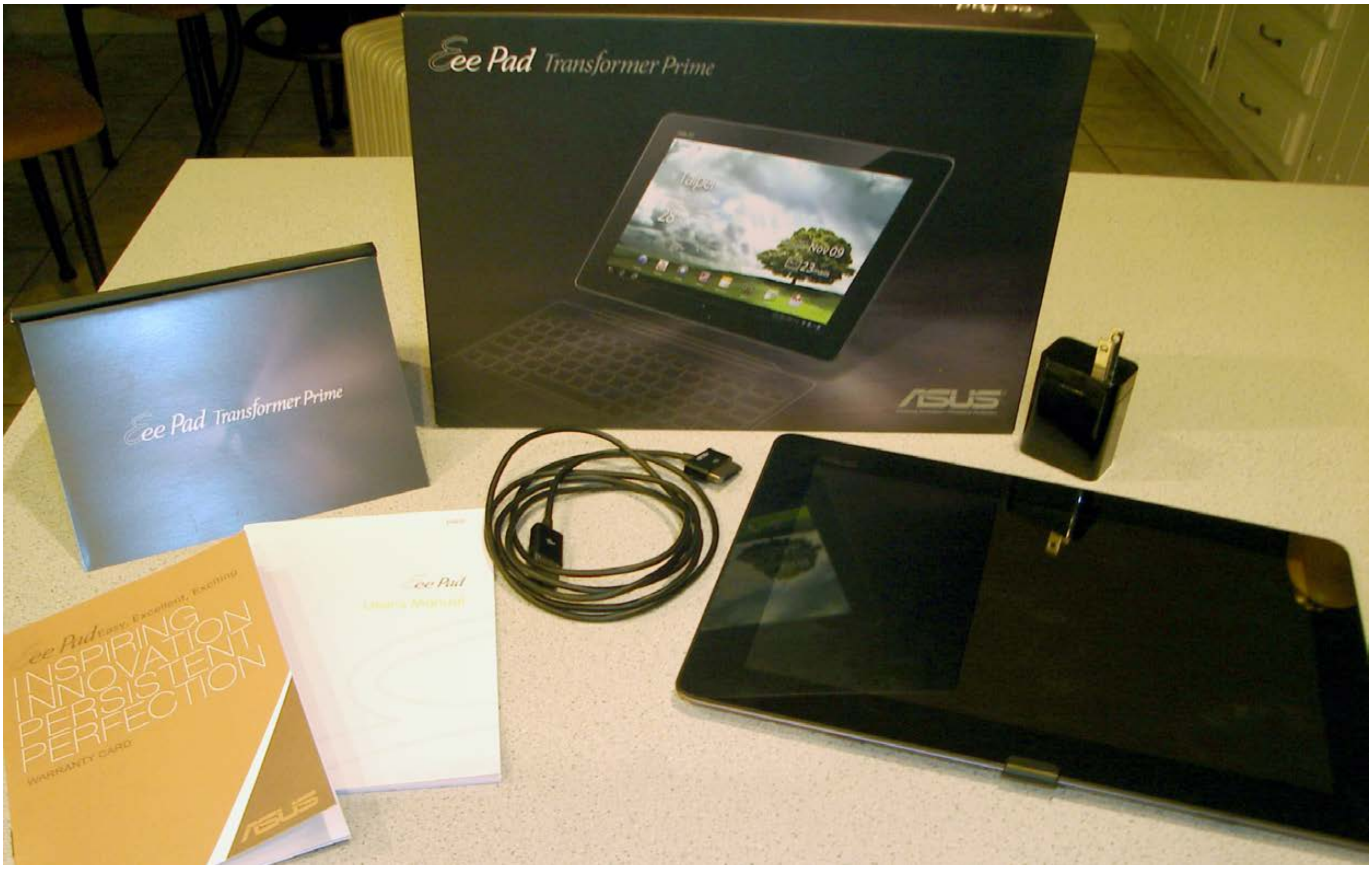

Eee Pad Transformer Prime
ASUS
Eee Pad Transformer Prime
Eee Pad Easy, Excellent, Exciting
INSPIRING
INNOVATION
PERSISTENT
PERFECTION
WARRANTY CARD
ASUS
Eee Pad

Figure 2. When you look this good, you don't need a lot of packaging.

than most people are likely to do.

Although the initial boot of the Prime puts you into a fairly stock version of Android 3.2 (Honeycomb), ASUS thankfully has not gone the route of heavy customizations to the interface. But, due to the arrival date of my device, I spent so little time with it, it was difficult to give the software platform a thorough walk-through. I received the tablet on a Thursday, and it was upgraded to Android 4.0 (Ice Cream Sandwich) the following Wednesday. This review focuses on that version of the operating system, as anyone purchasing a new Prime will be upgraded in short order.

As you start to use the Prime, you'll notice its responsiveness. Swiping through screens is pleasantly smooth, and apps pop open with little hesitation. If you own another Android device, you also may be surprised to see any apps you've installed begin showing up on the Prime as well. This is a nice touch, as I have more than 60 apps on my Motorola Droid, and after a few moments, I had them on my Prime too. ASUS preloaded a few of its own apps too, including ones to handle file management, media and cloud storage, although with my old, familiar apps installed automatically, I haven't used them much.

Transformer Prime vs. a Netbook

For the bulk of this review, I tried to remain objective in assessing the Prime's capability as a tablet—that is, how it stacks up to other tablets and people's expectation of tablets. Although the Prime certainly excels in these areas, one goal I had in mind when I purchased this was how it would serve as a replacement to my faithful yet aging MSI Wind Netbook. In some cases, the Prime trounces the Wind quite frankly, but there are some areas in which the Netbook is still king.

Daily busy work: I refer here to the little things that people do every day—move some files around, install programs and so on. And as a stalwart KDE user, in my opinion, doing these things in a full desktop OS still is miles beyond what you need to go through sometimes on Android just to cut and paste a file from one folder to another.

Simple Internet tasks: in this area, it's Prime all the way. My Kubuntu Netbook takes so long to get to a desktop at this point (it's almost always powered off, because it's so old the battery is shot, and it often will run out of power when in suspend) that if I'm just looking to do a quick Web search of some kind, I'm done on the Prime before I'd even be able to start on the Wind. Because I use Gmail as well, it's easy-peasy on Android.

Simple productivity tasks: once again, Prime all day long. If I need to jot down some notes at a meeting, Evernote, Supernote (an app ASUS includes with the device), or a simple text editor does the trick. And I'd consider Polaris capable of doing spreadsheets as well as drafting text documents, although I'd want to open them in a real word processor before I'd send them to anyone.

Multimedia: this is a tough call, as both are perfectly capable of playing most types of file-based media. But I have to give the Prime the edge here, because 1) Android supports more streaming and other Internet-based media services, and 2) the Prime is so light, it's with me all the time. Pulling out a tablet to listen to some music while enjoying a latte is a little more convenient in many cases than booting up a Netbook.

Complex productivity tasks: as mentioned earlier, the lack of hard-core business apps for Android is the main problem here, and so the Wind has to take this category. There's nothing that comes close to opening up a document in LibreOffice on Android at the moment. And although I also haven't had the chance to tinker with some code on the Prime, I have seen the text editors available, and as a beginner programmer, I can see them not holding my hand through Java inner classes as well as Eclipse does.

So for my purposes (which may be different from yours), I'm mostly satisfied with the Prime as a replacement to the MSI Wind, although I certainly feel the lack of LibreOffice at times. But I haven't carried the Wind out of the house since I got the Prime, so it must be doing a pretty good job so far.

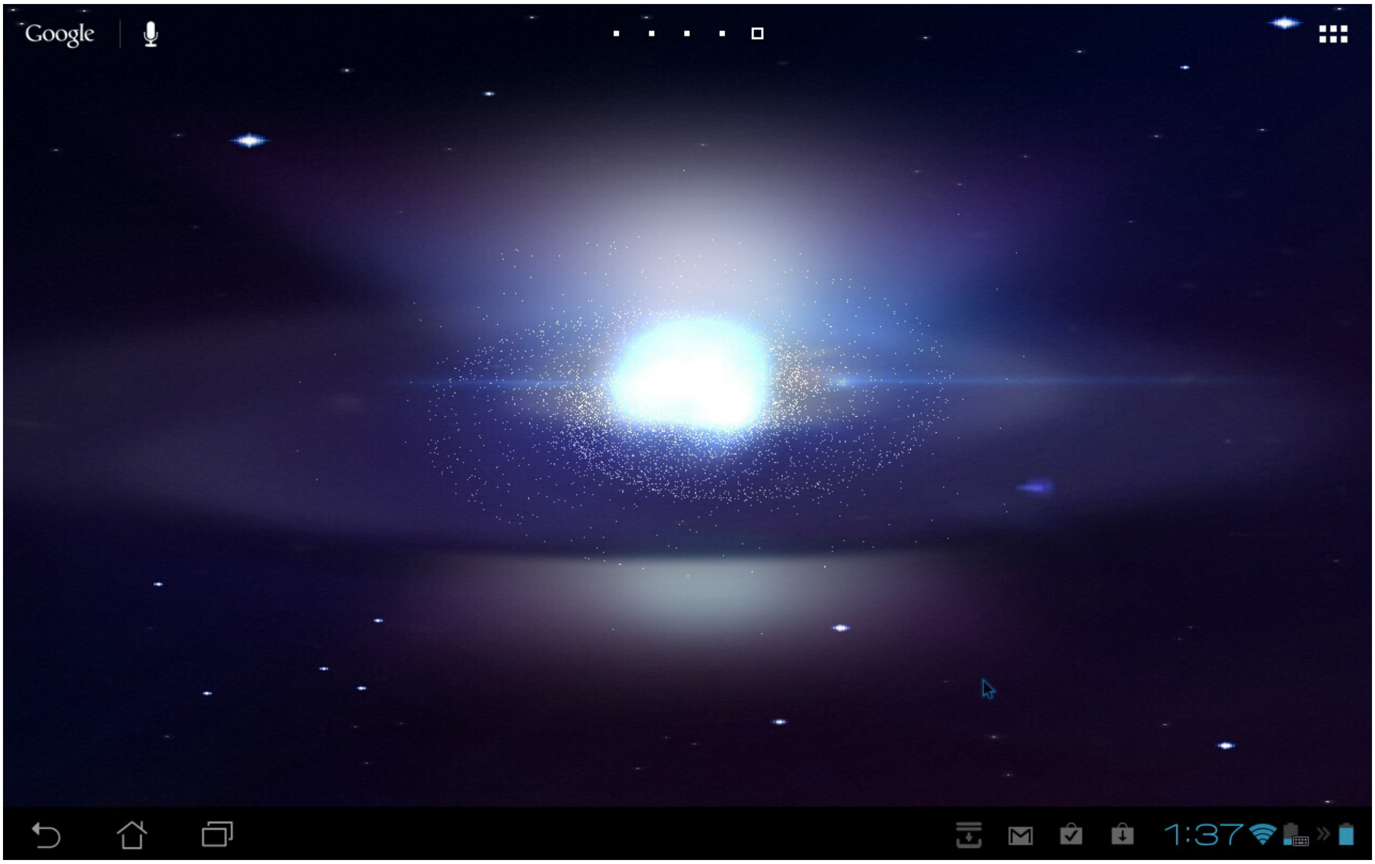
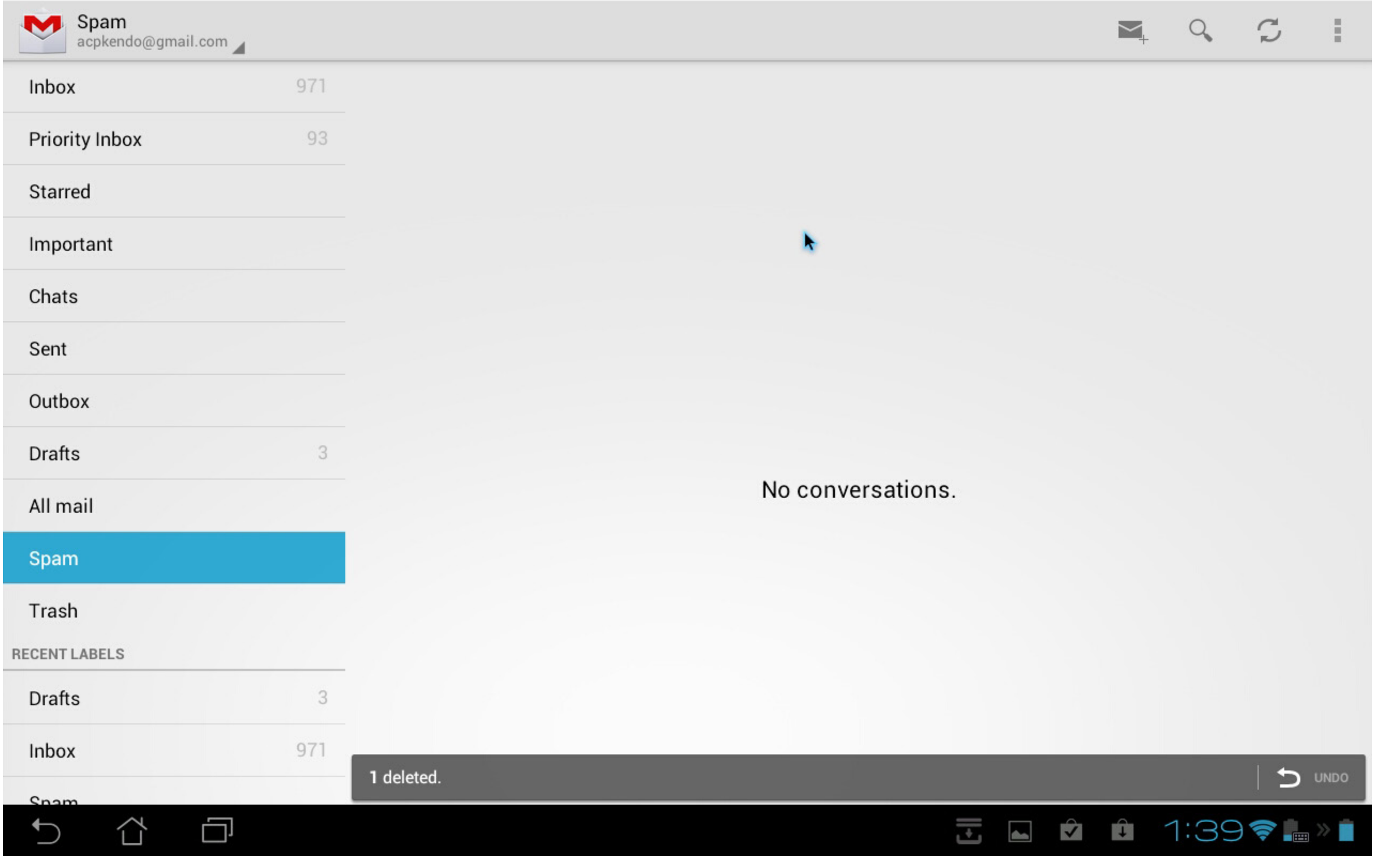


Figure 3. One of the Prime's live wallpapers—it spins so smoothly it'll make you dizzy.

I spent a fun week or so with the tablet on its own, during which time I got all the justification I needed for my purchase. Having used an iPad I received at work for a brief period, I did gain an appreciation for the form factor, which was perfect for media consumption. Google Reader isn't quite so easy or pleasant to use on either of my previous two main devices (the aforementioned Droid and an MSI Wind Netbook). And forget about video—although I had watched YouTube videos or shows on Netflix on the Droid, but not on my Kubuntu-powered Netbook (more on this in the Transformer Prime vs. a Netbook sidebar), it paled in comparison to the nice, bright, crisp screen of the Prime. The first few weeks with the Prime highlighted a number of this device's other strengths.

The Good

When considering the tablet half of the Prime, reading and watching video are really the activities that make you appreciate it. The ability to have two panes on-screen in something like Google Reader or Gmail felt like a huge upgrade from the tap-item-list-view-back-to-previous that's so common on smartphones. Video playback that didn't skip and that also didn't have to sit three inches from my



No conversations.

1 deleted.

Figure 4. In the Gmail Spam folder view—no spam here, hooray!

face in order to see everything was just gravy. Within these activities, a few specific examples stand out:

- Reading mail in Gmail: Gmail is a good, you might even say great, mail client (for Gmail, anyway) on a smartphone, but handling mail on the Prime is just better. The aforementioned panes make buzzing through a long list of messages and dealing with the important ones a breeze, and having the omnipresent list of labels to your left lets you jump back and forth between them as needed. The experience in other mail clients (including the stock Mail app that ships with Android as well as K9 Mail, which I use for my IMAP account at work) isn't as productive, in large part because neither of these apps seem to have been "upgraded" for tablets.

- Reading documents, such as PDFs and e-books: I've always considered reading on a phone to be a convenience thing—I have a few minutes, so I open something in a PDF or e-book reader on the phone while I can. Not so with the Prime. Although reading for long stretches requires resting the eyes every so often, I've

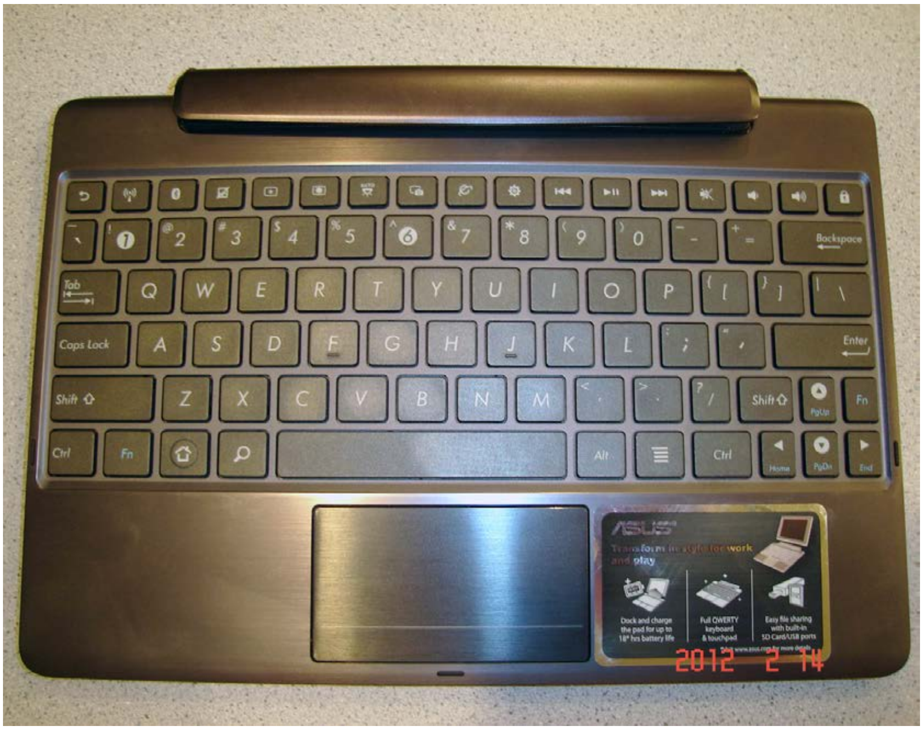


Figure 5. The missing link—the Prime's keyboard dock.

already gone through more than one full-length book on the Prime, as well as other PDFs and lots of Web-based documentation. I'm not saying that a secondary device with a nice, soft E Ink screen wouldn't be nice, but I'm not rushing out to buy one either.

- Watching video on Netflix: I do have Netflix installed on my phone, but as Joe Piscopo would have said in *Johnny Dangerously*, "I watched Netflix on my phone once. Once!" Grainy. Jumpy. Over Wi-Fi, it played all right, but given an extra few minutes, I'd probably spend them reading a book rather than trying to watch any video that wasn't a 30-second YouTube clip. But, Netflix on the Prime is smooth and sharp.

The experience with the tablet portion of the Prime on its own is quite good, and the polish Ice Cream Sandwich brings to the tablet form factor is readily apparent. Swiping through multiple apps; clicking on common links, such as e-mail addresses (which presents an option to open with Gmail or Mail, if the latter is configured), URLs (which open the Browser by default) and addresses (which open Google Maps); and opening common file types (which open the appropriate app or apps) all work as expected. I was able to multitask easily, and with the help of some must-have apps like Google Reader, GTasks, Evernote and Epistle (which were installed following my first boot, as mentioned above), I was putting together draft blog posts in no time. In this regard, actually creating content was a productive enough exercise, which is to say it was no easier or more difficult than on any other tablet.

That is, until my next delivery a week or so later. This is the best part of the Transformer experience: the keyboard dock. After removing the keyboard from its minimal packaging, I popped the Prime into the slot at the top, and following a firmware update, I was happily tippity-tapping away in about five seconds. It was that easy. I never tried hooking up the original Transformer to its keyboard, but I've read some complaints about the mechanism. I can report that the tablet clicks smoothly into place, and there is no hint of flimsiness or difficulty getting it to slot together with the keyboard.

You can't imagine how much more useful the two pieces become until you have them both. I tend to spend time flipping through Web pages or RSS feeds with the tablet alone. But if I have an idea for, well, anything, I can have it captured before it evaporates (which happens faster and faster these days). The keyboard is every bit as usable for longer typing sessions as the Netbook this tablet-combo is meant to replace. Although I haven't had the opportunity to try any other tablets with any other types of keyboards (such as Bluetooth models to pair with the iPad or Galaxy Tab, both of which I've spent some time using), I can't imagine any of them are as convenient and easy to use as the Prime and its keyboard dock. Highlights of the tablet/keyboard combination include:

- The additional battery life, which really does make for all-day usage.

- The trackpad and full-size USB port (if you have one of those mini-mice with the retractable cords, just plug it in, and it works).

- A row of dedicated, incredibly useful shortcut keys in place of the usual "Function" keys. Until recently, I'd actually taken the steps of going to Settings, switching to the Wireless & Networks screen, then turning the radio off. This is, of course, also done with a quick tap of the F1 key.

The keyboard does add a lot to the tablet experience, but it also highlights some shortcomings as well.

The Bad

The keyboard is a huge benefit, and the experience using it with OS functions and many of the built-in apps is really tight. But, support of the keyboard by some other apps isn't quite so good. For example, the stock Android Browser app responds to many of the keyboard shortcuts you'd expect (Ctrl-T for a new tab, Ctrl-W to close current tab, Ctrl-Left to go back and so on). But some other apps don't support them at all, requiring significant trackpad usage. This doesn't sound like a lot, but after one evening of replacing images on a Joomla site using only the built-in Browser (which, as correctly pointed out by many early reviews of the Prime, is no speed demon) and the Astro file manager, I can tell you that performing the same task on a desktop would have taken one-tenth the time (no exaggeration, I did the math when I finished it at work the next day). Worse are some of the apps

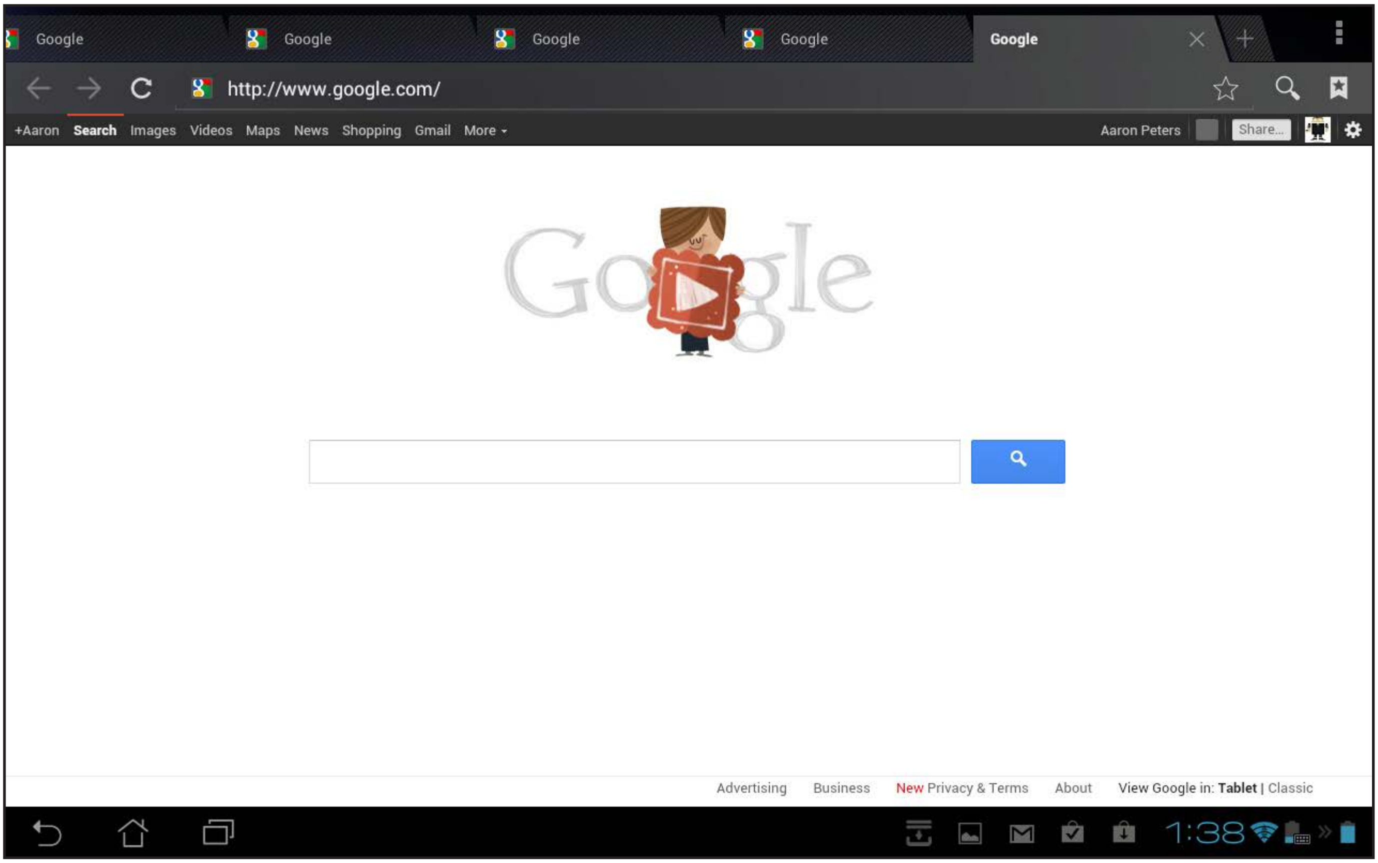


Figure 6. In Browser, opening these tabs via the keyboard was nearly instantaneous. Not so with other apps.

that support keyboard shortcuts, but only inconsistently. For example, in the Gmail app, pressing the C key will open a new mail (Compose, I imagine, an uncommon choice but I've seen it), but no combination of other keys with the letter R will allow you to Reply.

And just as the keyboard support is at the moment somewhat half-baked, the availability of apps in the Market optimized for tablets is still behind the curve. Although an icon in the notification bar will tell you if you're using a smartphone app that's "scaled-up" for use on bigger screens, you don't really need it—it's painfully obvious. Take the Facebook app for instance, in which nearly the entire right side of the app is empty space when in landscape orientation, or one of my favorites, ConnectBot, which refuses to sit in landscape orientation unless the keyboard is attached. Some of these apps just need a bit of thought given to typical tablet use cases, but for the time being, there are enough idiosyncrasies that it warrants more than just a shrug and a "meh".

Last, and least, for the time being, is that as someone who's doing the business/document thing during the day, the pre-installed Polaris office just doesn't hold up to most documents. It's fine for jotting down meeting notes or doing a quick spreadsheet to sum up some numbers, and the Google Docs integration is pretty neat. But for anything that involves heavy formatting, what you open is not going to look like what was sent to you, and what you send won't look like what the person you sent it to opens. Hopefully, the upcoming LibreOffice port (**http://people.gnome.org/~michael/blog/2012-02-04.html**) will give Android tablets, and especially a business-savvy model like the Prime, something to combat the likes of iWork on the iPad (which is a great example of what a slimmed-down office package can be).

Other minor nags include:

- The camera is okay—not great, not bad, but just okay.
- The little rubber doors that are all over the ports, with the exception of where the keyboard links up, which comes with a neat plastic dummy-card-looking thing I can't believe I haven't lost yet.
- I will mention for those who dislike this sort of thing that it is something of a fingerprint magnet. I'll also immediately qualify this by saying I hate when reviews point this out. It's a tablet. You use it with your hands, and your hands have fingers on them. It's going to get fingerprints on it. Luckily, they include a cleaning cloth with the tablet that's great at removing them.
- I find the speaker, despite some other reviews to the contrary, to be a little weak, especially when hooked up to the keyboard. But I so rarely rely on this speaker that it's not a problem for me.

The Conclusion

As someone who entered the mobile development field last year, I still held out for more than six months before spending some cheddar on this device. Now that I've done so, I have a hard time picturing what it would be like to go back. If I could get a full-blown, reasonably performing Linux build on this machine, it would make me think closely about whether I actually need a real notebook. But as it is, the ASUS Transformer Prime is certainly best in class among Android tablets currently on the market: a cutting-edge processor, the latest-and-greatest Android OS and the flexibility to go from a content-devouring media device to a quick-switching productivity machine (even if the lack of strong apps means you're working in plain text), all in a solid, stylish package. At approximately $650 for the whole kit, it's not the cheapest machine on the block, but I certainly consider it worth the price.■

Aaron Peters is a project manager and business analyst at a Web/mobile development firm, and he splits his free time between learning tech, writing and attacking other people with bamboo sticks. When he and his wife are not trying to corral the five animals living with them in Allentown, Pennsylvania, he sometimes answers e-mails sent to acpkendo@gmail.com.

SYSTEM ADMINISTRATION OF THE IBM WATSON SUPERCOMPUTER

Find out how the brains at IBM handle system administration of the Watson supercomputer.

ALEKSEY TSALOLIKHIN

System administrators at the USENIX LISA 2011 conference (LISA is a great system administration conference, by the way) in Boston in December got to hear Michael Perrone's presentation "What Is Watson?"

Michael Perrone is the Manager of Multicore Computing from the IBM T.J. Watson Research Center. The entire presentation (slides, video and MP3) is available on the USENIX Web site at **http://www.usenix.org/events/lisa11/tech**, and if you really want to understand how Watson works under the hood, take an hour to listen to Michael's talk (and the sysadmin Q&A at the end).

I approached Michael after his talk and asked if there was a sysadmin on his team who would be willing to answer some questions about handling Watson's system administration, and after a brief introduction to Watson, I include our conversation below.

What Is Watson?

In a nutshell, Watson is an impressive demonstration of the current state of the art in artificial intelligence: a computer's ability to answer questions posed in natural language (text or speech) correctly.

Watson came out of the IBM DeepQA Project and is an application of DeepQA tuned specifically to *Jeopardy* (a US TV trivia game show). The "QA" in DeepQA stands for Question Answering, which means the computer can answer your questions, spoken in a human language (starting with English). The "Deep" in DeepQA means the computer is able to analyze deeply enough to handle natural language text and speech

WATSON'S VITAL STATISTICS

- 90 IBM Power 750 servers (plus additional I/O, network and cluster controller nodes).
- 80 trillion operations per second (teraflops).
- Watson's corpus size was 400 terabytes of data—encyclopedias, databases and so on. Watson was disconnected from the Internet. Everything it knows about the world came from the corpus.
- Average time to handle a question: three seconds.
- 2880 POWER7 cores (3.555GHz chip), four threads per core.
- 500GB per sec on-chip bandwidth (between the cores on a chip).
- 10Gb Ethernet network.
- 15TB of RAM.
- 20TB of disk, clustered. (Watson built its semantic Web from the 400TB corpus. It keeps the semantic Web, but not the corpus.)
- Runs IBM DeepQA software, which has open-source components: Apache Hadoop distributed filesystem and Apache UIMA for natural language processing.
- SUSE Linux.
- One full-time sysadmin on staff.
- Ten compute racks, 80kW of power, 20 tons of cooling (for comparison, a human has one brain, which fits in a shoebox, can run on a tuna-fish sandwich and can be cooled with a handheld paper fan).

successfully. Because natural language is unstructured, deep analysis is required to interpret it correctly.

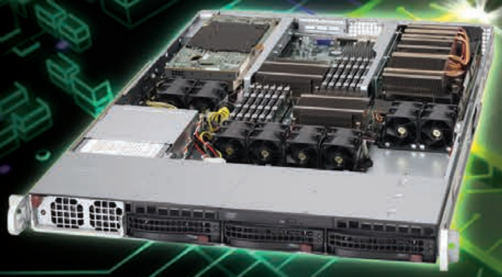
It demonstrates (in a popular format) a computer's capability to interface with us using natural language, to "understand" and answer questions correctly by quickly searching a vast sea of data and correctly picking out the vital facts that answer the question.

Watson is thousands of algorithms running on thousands of cores using terabytes of memory, driving teraflops of CPU operations to deliver an answer to a natural language question in less than five seconds. It is an exciting feat of technology, and it's just a taste of what's to come.

IBM's goal for the DeepQA Project is to drive automatic Question Answering technology to a point where it clearly and consistently rivals the best human performance.

How Does Watson Work?

First, Watson develops a semantic net. Watson takes a large volume of text (the corpus) and parses that with natural language processing to create "syntatic frames" (subject→verb→object). It then uses syntactic frames to create "semantic frames", which have a degree



of probability. Here's an example of semantic frames:

- Inventors patent inventions (.8).
- Fluid is a liquid (.6).
- Liquid is a fluid (.5).

Why isn't the probability 1 in any of these examples? Because of phrases like "I speak English fluently". They tend to skew the numbers.

WATSON IS BUILT ON OPEN SOURCE

Watson is built on the Apache UIMA framework, uses Apache Hadoop, runs on Linux, and uses xCAT and Ganglia for configuration management and monitoring—all open-source tools.

To answer questions, Watson uses Massively Parallel Probabilistic Evidence-Based Architecture. It uses the evidence from its semantic net to analyze the hypotheses it builds up to answer the question. You should watch the video of Michael's presentation and look at the slides, as there is really too much under the

XCAT HAS VERY POWERFUL PUSH FEATURES, INCLUDING A MULTITHREADED PUSH THAT INTERACTS WITH DIFFERENT MACHINES IN PARALLEL.

hood to present in a short article, but in a nutshell, Watson develops huge amounts of hypotheses (potential answers) and uses evidence from its semantic Web to assign probabilities to the answers to pick the most likely answer.

There are many algorithms at play in Watson. Watson even can learn from its mistakes and change its *Jeopardy* strategy.

Interview with Eddie Epstein on System Administration of the Watson Supercomputer

Eddie Epstein is the IBM researcher responsible for scaling out Watson's computation over thousands of compute cores in order to achieve the speed needed to be competitive in a live *Jeopardy* game. For the past seven years, Eddie managed the IBM team doing ongoing development of Apache UIMA. Eddie was kind enough to answer my questions about system administration of the Watson cluster.

AT: Why did you decide to use Linux?

EE: The project started with x86-based blades, and the researchers responsible for admin were very familiar with Linux.

AT: What configuration management tools did you use? How did you handle updating the Watson software on thousands of Linux servers?

EE: We had only hundreds of servers. The servers ranged from 4- to 32-core machines. We started with CSM to manage OS installs, then switched to xCat.

AT: xCat sounds like an installation system rather than a change management system. Did you use an SSH-based "push" model to push out changes to your systems?

EE: xCat has very powerful push features, including a multithreaded push that interacts with different machines in parallel. It handles OS patches, upgrades and more.

AT: What monitoring tool did you use and why? Did you have any cool visual models of Watson's physical or logical activity?

EE: The project used a home-grown cluster management system for development activities, which had its own monitor. It also incorporated ganglia. This tool was the basis for managing about 1,500 cores.

The Watson game-playing system used UIMA-AS with a simple SSH-based process launcher. The emphasis there was on measuring every aspect of runtime performance in order to reduce the overall latency. Visualization of performance data was then done after the fact. UIMA-AS managed the work on thousands of cores.

CONFIGURATION MANAGEMENT OF THE WATSON CLUSTER

CSM is IBM's proprietary Cluster Systems Management software (**www-03.ibm.com/systems/software/csm**). It is intended to simplify administration of a cluster and includes parallel execution capability for high-volume pushes:

> [CSM is] designed for simple, low-cost management of distributed and clustered IBM Power Systems in technical and commercial computing environments. CSM, included with the IBM Power Systems high-performance computer solutions, dramatically simplifies administration of a cluster by providing management from a single point-of-control.... In addition to providing all the key functions for administration and maintenance of typical distributed systems, CSM is designed to deliver the parallel execution required to manage clustered computing environments effectively.

xCAT also originated at IBM. It was open-sourced in 2007. The xCAT Project slogan is "Extreme Cloud Administration Toolkit", and its logo is a cat skull and crossbones. It now lives at **xcat.sourceforge.net**, which describes it as follows:

- Provision operating systems on physical or virtual machines: SLES10 SP2 and higher, SLES 11 (incl. SP1), RHEL5.x, RHEL 6, CentOS4.x, CentOS5.x, SL 5.5, Fedora 8-14, AIX 6.1, 7.1 (all available technology levels), Windows 2008, Windows 7, VMware, KVM, PowerVM and zVM.

- Scripted install, stateless, satellite, iSCSI or cloning.

- Remotely manage systems: integrated lights-out management, remote console, and distributed shell support.

- Quickly set up and control management node services: DNS, HTTP, DHCP and TFTP.

xCAT offers complete and ideal management for HPC clusters, render farms, grids, WebFarms, on-line gaming infrastructure, clouds, data centers, and whatever tomorrow's buzzwords may be. It is agile, extendible and based on years of system administration best practices and experience.

xCAT grew out of a need to rapidly provision IBM x86-based machines and has been actively developed since 1999. xCAT is now ten years old and continues to evolve.

WHAT IS UIMA-AS?

UIMA (Unstructured Information Management Architecture) is an open-source technology framework enabling Watson. It is a framework for analyzing a sea of data to discover vital facts. It is computers taking unstructured data as input and turning it into structured data and then analyzing and working with the structured data to produce useful results.

The analysis is "multi-modal", which means many algorithms are employed, and many kinds of algorithms. For example, Watson had a group of algorithms for generating hypotheses, such as using geo-spatial reasoning, temporal reasoning (drawing on its historical database), pun engine and so on, and another group of algorithms for scoring and pruning them to find the most likely answer.

In a nutshell, this is Massively Parallel Probabilistic Evidence-Based Architecture. (The evidence comes from Watson's 400TB corpus of data.)

The "AS" stands for Asynchronous Scaleout, and it's a scaling framework for UIMA—a way to run UIMA on modern, highly parallel cores, to benefit from the continuing advance in technology. UIMA brings "thinking computers" a giant step closer.

To understand unstructured information, first let's look at structured information. Computers speak with each other using structured information. Sticking to structured information makes it easier to extract meaning from data. HTML and XML are examples of structured information. So is a CSV file. Structured information standards are maintained by OASIS at **http://www.sgmlopen.org**.

Unstructured information is much more fluid and free-form. Human communication uses unstructured information. Until UIMA, computers have been unable to make sense out of unstructured information. Examples of unstructured information include audio (music), e-mails, medical records, technical reports, blogs, books and speech.

UIMA was originally an internal IBM Research project. It is a framework for creating applications that do deep analysis of natural human language text and speech.

In Watson, UIMA managed the work on nearly 3,000 cores. Incidentally, Watson could run on a single core—it would take it six hours to answer a question. With 3,000 cores, that time is cut to 2–6 seconds. Watson really takes advantage of massively parallel architecture to speed up its processing.

AT: What were the most useful system administration tools for you in handling Watson and why?

EE: clusterSSH (**http://sourceforge.net/apps/mediawiki/clusterssh**) was quite useful. That and simple shell scripts with SSH did most of the work.

AT: How did you handle upgrading Watson software? SSH in, shut down the service, update the package, start the service? Or?

EE: Right, the Watson application is just restarted to pick up changes.

AT: How did you handle packaging of Watson software?

EE: The Watson game player was never packaged up to be delivered elsewhere.

AT: How many sysadmins do you have handling how many servers? You mentioned there were hundreds of operating system instances—could you be more specific? (How many humans and how many servers?) Is there actually a dedicated system administration staff, or do some of the researchers wear the system administrator hat along with their researcher duties?

EE: We have in the order of 800 OS instances. After four years we finally hired a sysadmin; before that, it was a part-time job for each of three researchers with root access.

AT: Regarding your monitoring system, how did you output the system status?

EE: We are not a production shop. If the cluster has a problem, only our colleagues complain.

What's Next?

IBM wants to make DeepQA useful, not just entertaining. Possible fields of application include healthcare, life sciences, tech support, enterprise knowledge management and business intelligence, government, improved information sharing and security.■

Aleksey Tsalolikhin has been a UNIX/Linux system administrator for 14 years. Wrangling EarthLink's server farms by hand during its growth from 1,000 to 5,000,000 users, he developed an abiding interest in improving the lot of system administrators through training in configuration management, documentation and personal efficiency (including time management for system administrators and improving communication). Aleksey also provides private and public training services; send e-mail to aleksey@verticalsysadmin.com for more information.

Resources

IBM's Watson Site—"What is Watson?", "Building Watson" and "Watson for a Smarter Planet": **http://ibmwatson.com**

IBM's DeepQA Project: **http://www.research.ibm.com/deepqa/deepqa.shtml**

Eddie Epstein's IBM Researcher Profile: **http://researcher.ibm.com/view.php?person=us-eae**

Wikipedia Article on Watson: **http://en.wikipedia.org/wiki/Watson_%28computer%29**

Apache UIMA: **http://uima.apache.org**

OpenLDAP Everywhere Reloaded, Part I

How to engineer an OpenLDAP directory service to create a unified login for heterogeneous environments.

STEWART WALTERS

Directory services is one of the most interesting and crucial parts of computing today. They provide our account management, basic authentication, address books and a back-end repository for the configuration of many other important applications.

It's been nine long years since Craig Swanson and Matt Lung originally wrote their article "OpenLDAP Everywhere" (*LJ*, December 2002), and almost six years since their follow-up article "OpenLDAP Everywhere Revisited" (*LJ*, July 2005).

In this multipart series, I cover how to engineer an OpenLDAP directory service to create a unified login for heterogeneous environments. With current software and a modern approach to server design, the aim is to reduce the number of single points of failure for the directory.

In this article, I describe how to configure two Linux servers to host core network services required for clients to query the directory service. I configure these core services to be highly available through the use of failover pools and/or replication.

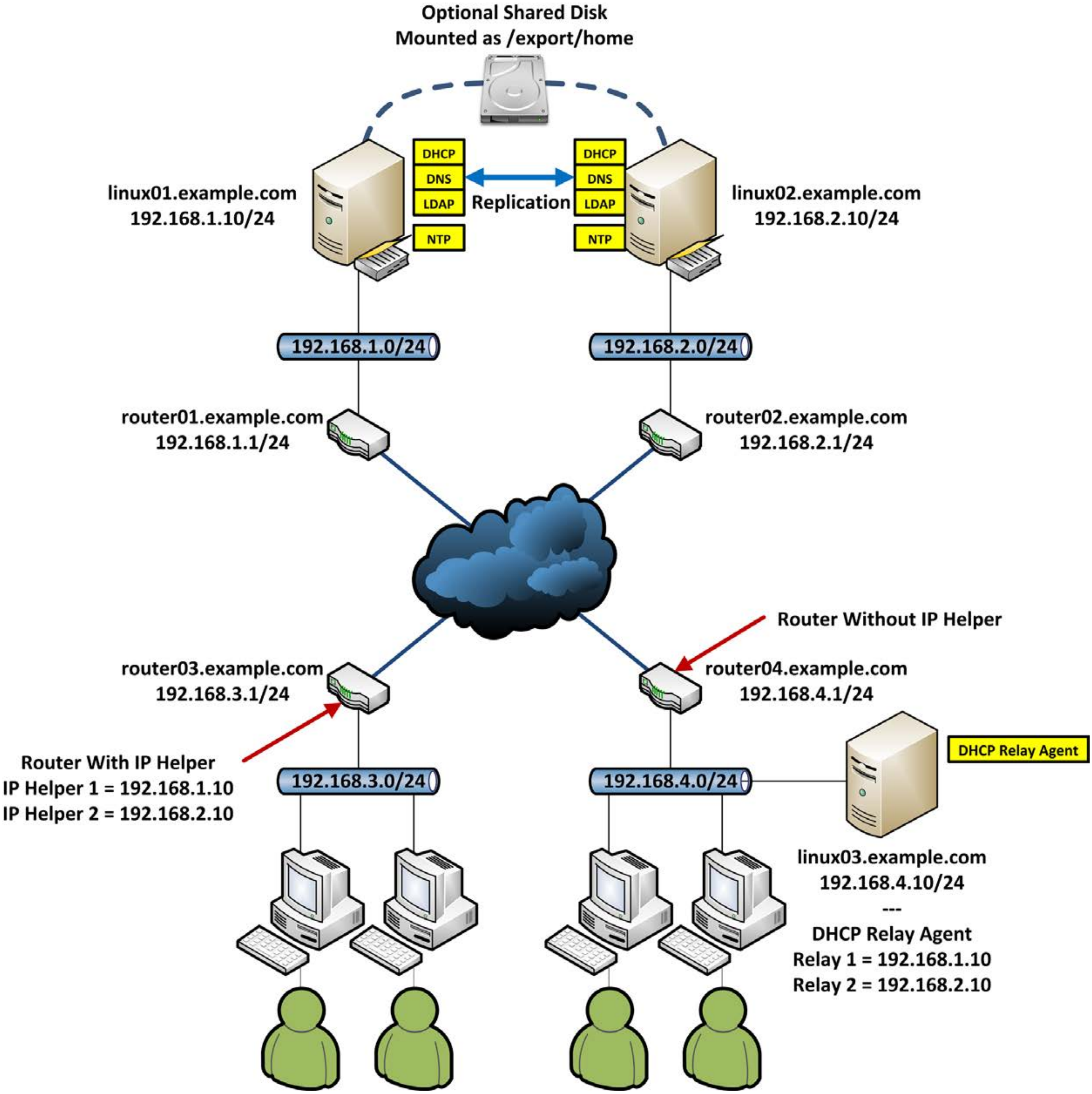


Figure 1. An Overall View of Core Network Services, Including LDAP (Note: the image of the hard disk icon in this figure was taken from the Open Icon Library Project: http://openiconlibrary.sourceforge.net.)

Assumptions and Prerequisites

Certain approaches were taken in this design with small-to-medium enterprises (SMEs) in mind. You may wish to custom-tailor the design if you are a small-to-medium business (SMB) or large-scale enterprise.

The servers discussed in this article were installed with the latest stable version of the Debian GNU/Linux. At the time of this writing, this was Debian 6.0.2.1 (Squeeze). Although it has not been tested for Ubuntu, Ubuntu users should be able to log in as root (run `sudo su -`) and have few problems.

As per Figure 1, the fictional local domain name is example.com. Four fictitious subnetworks exist: 192.168.1.0/24, 192.168.2.0/24, 192.168.3.0/24 and 192.168.4.0/24. Routing between the four subnets is assumed to be working correctly. Where appropriate, please substitute applicable values for your domain name, IP addresses, netmask addresses and so on.

LDAP users are assumed to have home directories in /export/home rather than /home. This allows LDAP credentials to be compatible for operating systems other than Linux. Many proprietary UNIXes, for example, use /export/home as the default home directory. /home on Solaris is also reserved for other purposes (automount directories).

The design assumes that /export/home is actually a shared disk.

This is typically implemented as a mountpoint to an NFS server on a host or NAS; however, the design makes no determination about how to achieve the shared disk, which is beyond the scope of the article, so I'm leaving it to the reader to decide how to implement this.

You can opt not to implement the shared disk, but there are some serious drawbacks if you don't. All LDAP users will need their $HOME directory to be created manually by the administrator for every server to which they wish to log in (prior to them logging in). Also, the files a user creates on one server will not be available to other servers unless the user copies them to the other server manually. This is a major inconvenience for users and creates a waste of server disk space (and backup tape space) because of the duplication of data.

All example passwords are set to "linuxjournal", and it's assumed you'll replace these with your own sensible values.

Install Packages

On both linux01.example.com and linux02.example.com, use your preferred package manager to install the ntp, bind9, bind9utils, dnsutils, isc-dhcp-server, slapd and ldap-utils packages.

Start with Accurate Timekeeping (NTP)

Accurate timekeeping between the two Linux servers is a requirement for DHCP failover. There are additional benefits in having accurate time, namely:

- It's required if you intend to implement (or already have implemented) secure authentication with Kerberos.

- It's required if you intend to have some form of Linux integration with Microsoft Active Directory.

- It’s required if you intend to use N-Way Multi-Master replication in OpenLDAP.

- It greatly assists in troubleshooting, eliminating the guesswork when comparing logfile timestamps between servers, networking equipment and client devices.

Once ntp is installed on both linux01.example.com and linux02.example.com, you are practically finished. The Debian NTP team creates very sensible defaults for ntp.conf(5). Time sources, such as 0.debian.pool.ntp.org and 1.debian.pool.ntp.org, will work adequately for most scenarios.

If you prefer to use your own time sources, you can modify the lines beginning with `server` in /etc/ntp.conf. Replace the address with that of your preferred time source(s).

You can check on both servers to see if your ntp configuration is correct with the ntpq(1) command:

```
root@linux01:~# ntpq -p
     remote           refid      st t when poll reac
h   delay   offset  jitter
=====================================================
==========================
+warrane.connect 130.95.179.80    2 u  728 1024  377
   74.013  -19.461 111.056
+a.pool.ntp.uq.e 130.102.152.7    2 u  179 1024  377
   79.178  -14.069 100.659
*ntp4.riverwillo 223.252.32.9     2 u  749 1024  377
   76.930  -13.306  89.628
+c122-108-78-111 121.0.0.42       3 u  206 1024  377
   78.818    6.485  72.161
root@linux01:~#
```

Don’t be concerned if your ntpq output shows a different set of servers. The *.pool.ntp.org addresses are DNS round-robin records that balance DNS queries among hundreds of different NTP servers. The important thing is to check that ntp can contact upstream NTP servers.

Name Resolution (DNS)

If the LDAP client can’t resolve the hostname of the Linux servers that run OpenLDAP, they can’t connect to the directory services they provide. This can include the inability to retrieve basic UNIX account information for authentication, which will prevent user logins.

As such, configure ISC bind to provide DNS zones in a master/slave combination between the two Linux servers. The example workstations will be configured (through DHCP) to query DNS on linux01.example.com, then linux02.example.com if the first query fails.

Note: /etc/bind/named.conf normally is replaced by the package manager when the bind9 package is upgraded. Debian’s default named.conf(5) has an `include /etc/bind/named.conf.local` statement so that site local zone configurations added there are not lost when the bind9 package is upgraded.

On linux01.example.com, modify

/etc/bind/named.conf.local to include the following:

```
//// excerpt of named.conf.local on linux01

// --- Above output suppressed

zone "168.192.in-addr.arpa" {
   type master;
   file "/etc/bind/db.168.192.in-addr.arpa";
   notify yes;
   allow-transfer { 192.168.2.10; }; // linux02
};

zone "example.com" {
   type master;
   file "/etc/bind/db.example.com";
   notify yes;
   allow-transfer { 192.168.2.10; }; // linux02
};

// --- Below output suppressed
```

On linux01.example.com, create the zone files /etc/bind/db.168.192.in-addr.arpa and /etc/bind/db.example.com, and populate them with appropriate zone information. For very basic examples of the zone files, see the example configuration files available on the *LJ* FTP site (see Resources for the link).

Before committing changes to a production DNS server, always check that no mistakes are present. Failure to do this causes the named(8) dæmon to abort when restarted. You don't want to cause a major outage for production users if there is a trivial error. On linux01.example.com:

```
# named-checkconf /etc/bind/named.conf
# named-checkconf /etc/bind/named.conf.local
# named-checkzone 168.192.in-addr.arpa /etc/bind/db.
168.192.in-addr.arpa
zone 168.192.in-addr.arpa/IN: loaded serial 20111003
01
OK
# named-checkzone example.com /etc/bind/db.example.com
zone example.com/IN: loaded serial 2011100301
OK
#
```

On linux01.example.com, instruct the named(8) dæmon to reload its configuration files, then check that it didn't abort:

```
root@linux01:~# /etc/init.d/bind9 reload
Reloading domain name service...: bind9.
root@linux01:~# ps -ef|grep named|grep -v grep
bind      1283     1  0 16:05 ?        00:00:00 /usr
/sbin/named -u bind
root@linux01:~#
```

It is possible during normal operations that the named(8) dæmon on linux01.example.com could abort and the rest of the server would otherwise continue to function as normal (that is, single service failure, not entire server failure). As linux02.example.com will have a backup copy of the zones anyway, linux01.example.com should use linux02.example.com as its secondary DNS server.

On linux01.example.com, create and/or modify /etc/resolv.conf. Populate it with the following:

```
search example.com
nameserver 127.0.0.1
nameserver 192.168.2.10
```

On linux01.example.com, check, and if necessary, modify /etc/nsswitch.conf to include the following "hosts" definition. This line already was in place for me, but it strictly does need to be present for you if it isn't:

```
## /etc/nsswitch.conf on linux01 & linux02

# --- Above output suppressed

hosts:          files dns

# --- Below output suppressed
```

Finally, test that linux01.example.com can resolve records from the DNS server:

```
root@linux01:~# dig linux02.example.com +short
192.168.2.10
root@linux01:~# dig -x 192.168.2.10 +short
linux02.example.com.
root@linux01:~# nslookup linux02.example.com
Server:         127.0.0.1
Address:        127.0.0.1#53

Name:   linux02.example.com
Address: 192.168.2.10

root@linux01:~# nslookup 192.168.2.10
Server:         127.0.0.1
Address:        127.0.0.1#53

10.2.168.192.in-addr.arpa      name = linux01.example.com.

root@linux01:~#
```

Now, configure linux02.example.com as the slave server. First, modify /etc/bind/named.conf.local to include the following:

```
//// excerpt of named.conf.local on linux02

// --- Above output suppressed

zone "168.192.in-addr.arpa" {
   type slave;
   file "/var/lib/bind/db.168.192.in-addr.arpa";
   masters { 192.168.1.10; }; // the linux01 server
};

zone "example.com" {
   type slave;
   file "/var/lib/bind/db.example.com";
   masters { 192.168.1.10; }; // the linux01 server
};

// --- Below output suppressed
```

Take careful note of the placement of the slave zone files in /var/lib/bind, not in /etc/bind!

This change is for two reasons. First,

/etc/bind is locked down with restrictive permissions so named(8) is not able to write any files there. named(8) on linux02.example.com cannot and should not write a transferred zone file there.

Second, the /var partition is intentionally designated for files that will grow over time. /var/lib/bind is the Debian chosen directory for named(8) to store such files.

Please resist the urge to change permissions to "fix" /etc/bind! I cannot stress this enough. It not only compromises the security on your RNDC key file, but also the dpkg package manager is likely to revert any change you made on /etc/bind the next time the bind9 package is upgraded.

If you require a single location for both servers to store their zone files, it would be better to move the local zone files on linux01.example.com to /var/lib/bind, rather than force a change to /etc/bind on linux02.example.com. Don't forget to update the paths for the zone files in linux01.example.com's /etc/bind/named.conf.local accordingly.

On linux02.example.com, run named-checkconf(1) to check the new configuration, as you did before for linux01.example.com. If the new configuration checks out, tell named(8) to reload by running the `/etc/init.d/bind9 reload` command. Also check that the dæmon didn't abort by running `ps -ef|grep named|grep -v grep` as was done before.

If the zone transfer from linux01.example.com was successful, you should have something like the following appear in /var/log/syslog on linux02.example.com:

```
# --- above output suppressed ---

Oct 3 20:37:11 linux02 named[1253]: transfer of '168
.192.in-addr.arpa/IN' from 192.168.1.10#53: connected
 using 192.168.2.10#35988
  --- output suppressed ---
Oct 3 20:37:11 linux02 named[1253]: transfer of '168
.192.in-addr.arpa/IN' from 192.168.1.10#53: Transfer
completed: 1 messages, 12 records, 373 bytes, 0.001
secs (373000 bytes/sec)
  --- output suppressed ---
Oct 3 20:37:12 linux02 named[1253]: transfer of 'exa
mple.com/IN' from 192.168.1.10#53: connected using 1
92.168.2.10#41155
  --- output suppressed ---
Oct 3 20:37:12 linux02 named[1253]: transfer of 'exa
mple.com/IN' from 192.168.1.10#53: Transfer complete
d: 1 messages, 12 records, 336 bytes, 0.001 secs (33
6000 bytes/sec)

# --- below output suppressed ---
```

On linux02.example.com, create and/or modify /etc/resolv.conf. Populate it with the following:

```
search example.com
nameserver 127.0.0.1
nameserver 192.168.1.10
```

This is the only device on the network

that will ever have linux02.example.com as its primary DNS server. It's done for performance reasons, on the assumption that linux01.example.com will fail first. Of course, you never can predict which server will fail first. However, if linux02.example.com happens to fail first, the workstations, in theory, won't notice it—DHCP tells them to query linux01.example.com before linux02.example.com.

Now, on linux02.example.com, check, and if necessary, modify /etc/nsswitch.conf to include the `hosts: files dns` in the same way performed previously. Check that dig(1) and nslookup(1) can resolve linux01.example.com in a similar manner as done before.

IP Address Assignment (DHCP)

If your LDAP clients can't receive an IP address to communicate with the network, they can't communicate with the OpenLDAP servers.

As such, configure ISC dhcpd to use failover pools between the two Linux servers. This ensures that IP addresses always are being handed out to clients. It also ensures that the two Linux servers are not allocating the duplicate addresses to the workstations.

The failover protocol for dhcpd is still in development by ISC at the time of this writing, but it is subject to change in the future. It works fairly well most of the time in its current state, and it's an important part of mitigating the risk of server failure for the directory service.

Create a new file on both linux01.example.com and linux02.example.com by running the command `touch /etc/dhcp/dhcpd.conf.failover`.

Separate the failover-specific configuration from the main /etc/dhcp/dhcpd.conf file. That way, /etc/dhcp/dhcpd.conf can be synchronized between both servers without destroying the unique configuration in the "failover peer" stanza. You never should synchronize /etc/dhcp/dhcpd.conf.failover between the two Linux servers.

On linux01.example.com, populate /etc/dhcp/dhcpd.conf.failover as follows:

```
failover peer "dhcp-failover" {
   primary;
   address linux01.example.com;
   port 519;
   peer address linux02.example.com;
   peer port 520;
   max-response-delay 60;
   max-unacked-updates 10;
   load balance max seconds 3;
   mclt 3600;
   split 128;
}
```

Notice that the parameters `split` and `mclt` are specified only on the primary server linux01.example.com.

`max-response-delay` controls how many seconds one server will wait for communication from the other before it assumes a failure.

`split` controls how many IP addresses available in the pool are pre-allocated to each DHCP server. The only valid values are from 0 to 255. As per the example, a `split 128;` value governs that each server takes 50% of the leases of the entire pool.

On linux02.example.com, populate /etc/dhcp/dhcpd.conf.failover as follows:

```
failover peer "dhcp-failover" {
   secondary;
   address linux02.example.com;
   port 520;
   peer address linux01.example.com;
   peer port 519;
   max-response-delay 60;
   max-unacked-updates 10;
   load balance max seconds 3;
}
```

Note: IANA has not allocated a reserved port number for ISC dhcpd failover at the time of this writing. This means the port/peer port numbers of 519 and 520 are subject to change.

On both linux01.example.com and linux02.example.com, you now should populate /etc/dhcp/dhcpd.conf with appropriate subnet information. For a very basic example of dhcpd.conf, see the example configuration files provided in the Resources section.

The crucial parameters to have in /etc/dhcp/dhcpd.conf are:

```
# excerpt of dhcpd.conf on linux01 and linux02

#-----------------
# Global DHCP parameters
#-----------------

# --- outputs heavily suppressed ----

#-----------------
# Failover parameters
#-----------------

include "/etc/dhcp/dhcpd.conf.failover";

# --- outputs heavily suppressed ---

subnet 192.168.3.0 netmask 255.255.255.0 {
   option routers 192.168.3.1;
   option subnet-mask 255.255.255.0;
   option broadcast-address 255.255.255.255;
   pool {
      failover peer "dhcp-failover";
      range 192.168.3.20 192.168.3.250;
   }
}

subnet 192.168.4.0 netmask 255.255.255.0 {
   option routers 192.168.4.1;
   option subnet-mask 255.255.255.0;
   option broadcast-address 255.255.255.255;
   pool {
      failover peer "dhcp-failover";
      range 192.168.4.20 192.168.4.250;
   }
}
```

These parameters alone are not enough to get a new DHCP server up and running. But, once a working dhcpd.conf is built for your network, add the above parameters for DHCP failover.

Restart dhcpd(8) on both linux01.example.com and linux02.example.com. Do this by running the command `/etc/init.d/isc-dhcp-server restart`. Check that the dhcpd(8) process did not abort by running `ps -ef|grep dhcpd|grep -v grep`.

If dhcpd(8) is no longer running, the problem is usually a typo. Re-check in dhcpd.conf and dhcpd.conf.failover that every opening parenthesis (the { character) has a closing parenthesis (the } character). Also check that lines not ending with open/closed parentheses are terminated by a semicolon (;).

Check /var/log/syslog on both servers for messages from dhcpd. When DHCP failover works, both servers should say the pool is "balanced", and that "I move from communications-interrupted to normal" or "I move from startup to normal".

Synchronize /etc/dhcp/dhcpd.conf from linux01.example.com to linux02.example.com every time you modify it. This can be done manually, via an rsync(1) script, via puppetd(8) or via the Network Information Service (though I don't recommend NIS—it's insecure and obsoleted by DNS/LDAP and rsync/puppet).

The drawback to the failover protocol is that it's a long way off from being considered mature. There are plenty of weird situations where the protocol fails to do its job. However, it will not be young forever, and when it does work, it works well. I recommend you monitor your logs and sign up to ISC's dhcp-users mailing list for assistance when things go wrong (see Resources for a link).

One last note on the DHCP failover protocol: if you have more devices on one subnet than 50% of the overall amount available in the pool range, seriously consider re-engineering your network before implementing DHCP failover.

The protocol inherently relies on having an excess of free addresses to allocate, even after the pool range is cut in half by `split 128;`.

The maximum amount of available IP addresses for DHCP in a C Class subnet most of the time is 253 (255 addresses, minus 1 address for broadcast, minus 1 address for the router).

If you have more than 126 devices within one failover subnet, either split it into more subnets (for example, one subnet for each floor of the building), or use a larger subnet than C Class. Configure the subnet declaration in /etc/dhcpd.conf to increase its pool range accordingly. It will save you problems later on.

Now that the DHCP servers are configured with failover pools, the final thing to do is configure the 192.168.3.0/24 and 192.168.4.0/24 to forward DHCP client broadcasts through the LAN/WAN to 192.168.1.10

and 192.168.2.10.

This is done on router03.example.com with IP Helper addresses. On linux03.example.com, it's done with ISC's DHCP Relay Agent.

Assume router03.example.com is a Cisco Catalyst Multi-layer Switch. Configure IP Helper addresses by entering privileged mode (run the `enable` command). Using the `ip helper-address` command, apply the two DHCP server IP addresses to the router interface that has the 192.168.3.1/24 address. On the Catalyst 3750G in my lab, this is interface "vlan20". The commands are applied like so:

```
router03#show running-config
Building configuration...

  --- output suppressed ---

interface Vlan20
 description linuxjournal_vlan
 ip address 192.168.3.1 255.255.255.0
end

  --- output suppressed ---

router03#configure terminal
router03(config)#interface vlan 20
router03(config-if)#ip helper-address 192.168.1.10
router03(config-if)#ip helper-address 192.168.2.10
router03(config-if)#end
router03#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
0 bytes copied in 8.715 secs (0 bytes/sec)
router03#show running-config interface vlan 20
Building configuration...

Current configuration : 154 bytes
!
interface Vlan20
 description linuxjournal_vlan
 ip address 192.168.3.1 255.255.255.0
 ip helper-address 192.168.1.10
 ip helper-address 192.168.2.10
end

router03#
```

On linux03.example.com, you need to install the isc-debian-relay package. Once it's installed, it will ask for the "multiple server names be provided as a space-separated list". Enter "linux01.example.com linux02.example.com", or if this host isn't configured to resolve from our DNS server pair, "192.168.1.10 192.168.2.10". It will ask on which interface to listen. If you have no preference, leave it blank and press Enter. It will ask you to specify additional options, but you simply can press Enter.

If you make a mistake, you can reconfigure by running the command `dpkg-reconfigure isc-dhcp-relay` or modify the `SERVERS` variable in /etc/default/isc-dhcp-relay.

Your DHCP clients now should be able to contact either DHCP server.

In Part II of this series, I'll explain how to configure OpenLDAP on the two Linux servers and start to populate the directory with data.■

Stewart Walters is a Solutions Architect with more than 15 years' experience in the Information Technology industry. Amongst other industry certifications, he is a Senior Level Linux Professional (LPIC-3). Where possible, he tries to raise awareness of the "Parkinson-Plus" syndromes, such as crippling neurodegenerative diseases like Progressive Supranuclear Palsy (PSP) and Multiple System Atrophy (MSA). He can be reached for comments at stewart.walters@googlemail.com.

Resources

Example configuration files for this article: **ftp://ftp.linuxjournal.com/pub/lj/listings/issue216/11148.tgz**

Debian GNU/Linux: **http://www.debian.org/distrib**

Download Debian 6.0.2.1: **http://cdimage.debian.org/debian-cd/6.0.2.1**

Manual Page for ntp.conf(5): **http://linux.die.net/man/5/ntp.conf**

Manual Page for named.conf(5): **http://linux.die.net/man/5/named.conf**

Manual Page for dhcpd.conf(5): **http://linux.die.net/man/5/dhcpd.conf**

Manual Page for dhcp-options(5): **http://linux.die.net/man/5/dhcp-options**

ISC dhcp-users Mailing List: **https://lists.isc.org/mailman/listinfo/dhcp-users**

Cisco IOS 12.3 T Command Reference for Idle through IP local-proxy-arp (includes `ip helper-address`): **http://www.cisco.com/en/US/docs/ios/12_3t/ip_addr/command/reference/ip1_i1gt.html**

PUPPET and NAGIOS: *a Roadmap to Advanced Configuration*

Going above and beyond the widely known approaches to managing Nagios with Puppet.

ADAM KOSMIN

Puppet has provided baked-in Nagios support for a long time now. When combined with Exported Resources, Puppet is well suited to manage an intelligent Nagios configuration where nodes are automatically inventoried and monitored. The excellent *Pro Puppet*, written by James Turnbull, provides a fairly complete rundown of the installation and configuration steps needed in order to progress in this direction, so I won't repeat the information here. Instead, this article highlights some less-than-optimal default behavior of the Nagios types and details my solution that results in a cleaner filesystem and improved performance.

Had it not been for Pieter Barrezeele's blog (**http://pieter.barrezeele.be/2009/05/11/puppet-and-nagios**), I may have ended up settling for Puppet's fairly inefficient approach to storing resources managed via its Nagios types. By default, these bits are maintained in hard-coded file paths according to type used. For example, all resources based on the nagios_service type are collected and stored in /etc/nagios/nagios_service.cfg and so on. For performance reasons, I want each collected resource to be stored in its own file path based on the following naming convention:

```
<base_directory>/<type>_<title>_<hostname>.cfg
```

Furthermore, I want my filenames

Not All Resources Should Be Exported!

This took me an embarrassingly long time to figure out. Just like resources that are defined in a manifest, Exported Resources must be unique. For example, suppose we have nodes foo and bar, which we'd like to categorize into a Nagios hostgroup named "PMF". At first glance, adding the following code to foo's manifest might seem like the way to go:

```
@@nagios_hostgroup { "PMF":
  ensure => present,
  hostgroup_members +> [ $::hostname ]
}
```

In theory, the resource will be exported to the database when the first node compiles its manifest, but the next node's compilation will complain with a duplicate resource error. For this reason, we will avoid exporting resources created by this particular type. Instead, we will manage our hostgroup memberships via the hostgroup parameter of the nagios_host type.

Listing 1. modules/nagios/manifests/init.pp

```
# This class will be used by the nagios server
class nagios {

  service { nagios:
    ensure => running,
    enable => true,
  }

  # Be sure to include this directory in your nagios.cfg
  # with the cfg_dir directive

  file { resource-d:
    path => '/etc/nagios/resource.d',
    ensure => directory,
    owner => 'nagios',
  }

  # Collect the nagios_host resources
  Nagios_host <<||>> {
    require => File[resource-d],
    notify => Service[nagios],
  }
}
```

Listing 2. /modules/nagios/manifests/export.pp

```
# All agents (including the nagios server) will use this
class nagios::export {

  @@nagios_host { $::hostname:
    address => $::ipaddress,
    check_command => 'check_host_alive!3000.0,80%!5000.0,100%!10',
    target => "/etc/nagios/resource.d/host_${::hostname}.cfg",
  }
}
```

to be composed of all lowercase letters and spaces replaced with underscores. For starters, let's add the bare minimum snippets of code into our manifests in order to export and collect resources using the nagios_host type (Listings 1 and 2).

Let's examine the good and the not-so-good aspects of what we've defined up to this point. On the positive side, all agents will export a nagios_host resource. The Nagios server, upon compiling its manifest, will collect each resource, store it in a unique file, and refresh the Nagios service. At first glance, it may seem like our work is done. Unfortunately, our solution is littered with the following issues and shortcomings:

1. Nagios will not be able to read the newly created .cfg files since the Puppet Agent will create them while running as the root user.

NOTE: Due to the inherent space limitations of published articles, all code will be kept as minimal as possible while conforming to the structure of Puppet Modules. However, no attempt will be made to reproduce a complete module capable of managing a Nagios instance. Instead, I focus on the concepts that have been defined in this article's introduction. Please see **http://docs.puppetlabs.com** if you need an introduction to Puppet modules.

2. There is too much “coordination” needed with the target parameter of the nagios_host type. We should not have to work so hard in order to ensure our target points to the correct file and is void of unpleasant things like spaces and/or mixed case.

3. The address parameter is hard-coded with the value of the ipaddress fact. Although this may be acceptable in some environments, we really should allow for greater flexibility.

4. No ability exists to leverage Nagios hostgroups.

5. Puppet will be unable to purge our exported resources, because we are not using the default behavior of the target parameter.

Refactoring the Code

In order to solve these issues, let’s write a new definition to act as a wrapper for all of the Nagios types we plan to use. Before we begin, let’s make sure we understand the most important problem—the issue of file ownership and permissions for the newly generated .cfg files. Because these files are created via the target parameter of each associated Nagios type, they’ll be written to disk by the user Puppet runs as. This means they will be owned by the root user/group, and Nagios will not have permission to read them (because I know you are not running Nagios as root, correct?). Although some people have chosen to work around this problem by chowning the files via Puppet’s exec type, we’re going to do something far cleaner in order to maintain Puppet’s greatest attribute, abstraction.

After plenty of failed “bright ideas” and confusion on my end, it became clear that it would be quite simple to control the ownership and permissions of each newly created .cfg file if each was managed as a file resource. We can fold the creation of these file resources into our wrapper definition and export them just as we do with the Nagios types. Each file resource then can be defined easily with appropriate

Listing 3. modules/nagios/manifests/params.pp

```
class nagios::params {

  $resource_dir = '/etc/nagios/resource.d'
  $user = 'nagios'

  case $::operatingsystem {

    debian: {
      $service = 'nagios3'
    }
    solaris: {
      $service = 'cswnagios'
    }
    default: {
      fail("This module is not supported on $::operatingsystem")
    }
  }
}
```

Listing 4. modules/nagios/manifests/resource.pp

```
define nagios::resource(
  $export,
  $type,
  $host_use = 'generic-host',
  $ensure = 'present',
  $owner = 'nagios',
  $address = '',
  $hostgroups = '',
  $check_command = ''
) {

  include nagios::params

  # figure out where to write the file
  # replace spaces with an underscore and convert
  # everything to lowercase
  $target = inline_template("${nagios::params::resource_dir}
➥/${type}_<%=name.gsub(/\\s+/, '_').downcase %>.cfg")

  case $export {
    true, false: {}
    default: { fail("The export parameter must be
➥set to true or false.") }
  }

  case $type {
    host: {
      nagios::resource::host { $name:
        ensure => $ensure,
        use => $host_use,
        check_command => $check_command,
        address => $address,
        hostgroups => $hostgroups,
        target => $target,
        export => $export,
      }
    }
    hostgroup: {
      nagios::resource::hostgroup { $name:
        ensure => $ensure,
        target => $target,
        export => $export,
      }
    }
    default: {
      fail("Unknown type passed to this define: $type")
    }
  }

  # create or export the file resource needed to support
  # the nagios type above
  nagios::resource::file { $target:
    ensure => $ensure,
    export => $export,
    resource_tag => "nagios_${type}",
    requires => "Nagios_${type}[${name}]",
  }
}
```

properties as well as requiring their corresponding Nagios type. When our Nagios server collects these resources, it first will create the file from the collected Nagios type before managing the file's attributes. Let's examine the new and refactored code.

The nagios::params class: First, let's define a few variables in a central location. Doing so will aid us in our

Listing 5. modules/nagios/manifests/resource/file.pp

```
define nagios::resource::file(
  $resource_tag,
  $requires,
  $export = true,
  $ensure = 'present',
) {

  include nagios::params

  if $export {

    @@file { $name:
      ensure => $ensure,
      tag => $resource_tag,
      owner => $nagios::params::user,
      require => $requires,
    }
  } else {

    file { $name:
      ensure => $ensure,
      tag => $resource_tag,
      owner => $nagios::params::user,
      require => $requires,
    }
  }
}
```

Listing 6. modules/nagios/manifests/resource/host.pp

```
define nagios::resource::host(
  $address,
  $hostgroups,
  $export,
  $target,
  $check_command,
  $use,
  $ensure = 'present'
) {

  include nagios::params

  if $export {

    @@nagios_host { $name:
      ensure => $ensure,
      address => $address,
      check_command => $check_command,
      use => $use,
      target => $target,
      hostgroups => $hostgroups ? {
        '' => undef,
        default => $hostgroups,
      },
    }
  } else {

    nagios_host { $name:
      ensure => $ensure,
      address => $address,
      check_command => $check_command,
      use => $use,
      target => $target,
      require => File[$nagios::params::resource_dir],
      hostgroups => $hostgroups ? {
        '' => undef,
        default => $hostgroups,
      },
    }
  }
}
```

quest to be "lazy" and not have to match values in various areas of our manifests (Listing 3).

The nagios::resource definition and friends: Our custom resource definition will serve as a wrapper for all Nagios types. Due to space considerations, the included code covers only the nagios_host and nagios_hostgroup types. Of course,

Listing 7. modules/nagios/manifests/resource/hostgroup.pp

```
define nagios::resource::hostgroup(
  $target,
  $ensure = 'present',
  $hostgroup_alias = '',
  $export = false
) {

  include nagios::params

  if $export {
    fail("It is not appropriate to export the Nagios_hostgroup
➥type since it will result in duplicate resources.")
  } else {
    nagios_hostgroup { $name:
      ensure => $ensure,
      target => $target,
      require => File[$nagios::params::resource_dir],
    }
  }
}
```

this definition can and should be extended to support every Nagios type we intend to use. Each supported type is represented in its own appropriately named definition 1 level under the nagios::resource namespace. Also included is a nagios::resource::file definition that is responsible for creating the previously mentioned .cfg file (Listings 4–7).

Listing 8. modules/nagios/manifests/export.pp

```
# All agents (including the nagios server) will use this
class nagios::export {

  nagios::resource { $::hostname:
    type => 'host',
    address => inline_template("<%= has_variable?('my_nagios_interface') ?
➥eval('ipaddress_' + my_nagios_interface) : ipaddress %>"),
    hostgroups => inline_template("<%= has_variable?('my_nagios_hostgroups') ?
➥$my_nagios_hostgroups : 'Other' %>"),
    check_command => 'check_host_alive!3000.0,80%!5000.0,100%!10',
    export => true,
  }
}
```

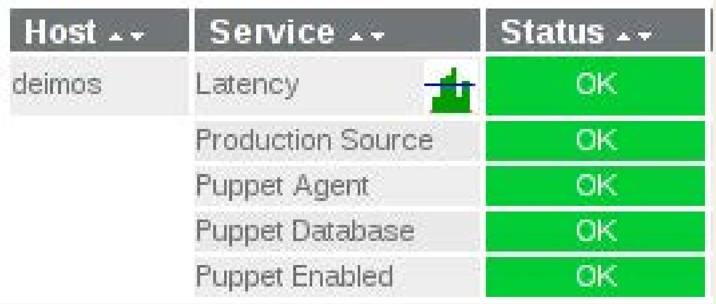
Listing 8 shows our refactored nagios::export class that is meant to be used by all nodes. Notice how we no longer leverage the nagios_host type directly. Instead, we call upon our newly created nagios::resource definition. Both the address and hostgroups parameters will use sane defaults unless they are overridden with node scoped variables. Also, notice how the target parameter is no longer required, as our nagios::resource definition performs the heavy lifting for us.

As you can see, the nagios::export class is ready to be extended with any kind of resource supported by our nagios::resource definition. Whenever we want all clients to export a particular resource, we just add it here so long as the following requirements are met:

1. The resource name must be unique.

2. The type parameter must be set.

TIP: Short-and-Sweet Nagios Service Descriptions



Host	Service	Status
deimos	Latency	OK
	Production Source	OK
	Puppet Agent	OK
	Puppet Database	OK
	Puppet Enabled	OK

Efficient Service Names in Nagios

When you get around to extending nagios::resource with support for the nagios_service type, you may want to consider using an inline ERB template to handle the service_description parameter. The following code removes the last word (which should be the hostname) from the description displayed in Nagios:

```
service_description => inline_template("<%= name.gsub(/\\w+$/,
➥'').chomp(' ') %>"),
```

Now, a resource defined with a unique title, such as “Puppet Agent $::hostname”, is displayed as “Puppet Agent” in Nagios.

3. The export parameter must be set to a value of true.

Now that all of our agents are exporting a nagios_host resource, we can focus on the collection side of things.

Expire, Collect and Purge Exported Resources

Up until this point, the job of our Nagios server simply has been to collect exported resources. In the real world, the nodes it monitors are retired for one reason or another quite routinely. When a node is retired, I want to be sure the relevant Nagios objects are removed and the corresponding database records are deleted. According to Puppet’s documentation, these resources can be purged from the collector only when default target locations are leveraged (**http://docs.puppetlabs.com/references/stable/type.html#nagioshost**). Even so, I wasn’t happy to see orphaned database records left behind and decided to address this issue with a few Puppet functions and some basic class ordering. Before we dive in, some work flow and terminology must be understood:

- Expire: a Nagios resource is “expired” by setting the value of its “ensure” parameter to “absent”.

- Collect: the resource is removed from the collector due to the value of its “ensure” parameter.

Listing 9. nagios/lib/puppet/parser/functions/expire_exported.rb

```
Puppet::Parser::Functions::newfunction(
  :expire_exported,
  :doc => "Sets a host's resources to ensure =>
➥absent as part of a purge work-flow.") do |args|

  require 'rubygems'
  require 'pg'
  require 'puppet'

  raise Puppet::ParseError, "Missing hostname." if args.empty?
  hosts = args.flatten

  begin
    conn = PGconn.open(:dbname => 'puppet', :user => 'postgres')

    hosts.each do |host|
      Puppet.notice("Expiring resources for host: #{host}")
      conn.exec("SELECT id FROM hosts WHERE name =
➥\'#{host}\'") do |host_id|
        raise "Too many hosts" if host_id.ntuples > 1
        conn.exec("SELECT id FROM param_names WHERE name =
➥'ensure'") do |param_id|
          conn.exec("SELECT id FROM resources WHERE host_id =
➥#{host_id.values.flatten[0].to_i}") do |results|

            resource_ids = []
            results.each do |row|
              resource_ids << Hash[*row.to_a.flatten]
            end

            resource_ids.each do |resource|
              conn.exec("UPDATE param_values SET VALUE =
➥'absent' WHERE resource_id = #{resource['id']} AND
➥param_name_id = #{param_id.values}")
            end
          end
        end
      end
    end
  rescue => e
    Puppet.notice(e.message)
  ensure
    conn.close
  end
end
```

- Purge: all database records associated with the expired host are deleted.

Ordering is obviously a big deal here. In order to ensure proper execution of each task, we will break out each unit of work into its own class and use a mix of "include" and "require" functions. Using Puppet terminology, we now can express this "expire, collect, then purge" work flow as follows:

- The nagios class requires the nagios::expire_resources class.
- The nagios class includes the nagios::purge_resources class.
- The nagios::purge_resources class requires the nagios::collect_resources class.

Now, let's look at a few custom functions, expire_exported and purge_exported. These functions (written for PostgreSQL) perform the database operations that are required in

Listing 10. nagios/lib/puppet/parser/functions/purge_exported.rb

```
# This function will be used by the exported
# resources collector (the nagios box)
Puppet::Parser::Functions::newfunction(:purge_exported,
➥:doc => "delete expired resources.") do |args|

  require 'rubygems'
  require 'pg'
  require 'puppet'

  raise Puppet::ParseError, "Missing hostname." if args.empty?
  hosts = args.flatten

  begin
    conn = PGconn.open(:dbname => 'puppet', :user => 'postgres')

    hosts.each do |host|

      Puppet.notice("Purging expired resources for host: #{host}")
      conn.exec("SELECT id FROM hosts WHERE name =
➥\'#{host}\'") do |host_id|

        raise "Too many hosts" if host_id.ntuples > 1
        conn.exec("SELECT id FROM resources WHERE host_id =
➥#{host_id.values.flatten[0].to_i}") do |results|

          resource_ids = []
          results.each do |row|
            resource_ids << Hash[*row.to_a.flatten]
          end

          resource_ids.each do |resource|
            conn.exec("DELETE FROM param_values WHERE
➥resource_id = #{resource['id']}")
            conn.exec("DELETE FROM resources WHERE id =
➥#{resource['id']}")
          end
        end

        conn.exec("DELETE FROM hosts WHERE id =
➥#{host_id.values}")
      end
    end
  rescue => e
    Puppet.notice(e.message)
  ensure
    conn.close
  end
end
```

order to expire hosts and their resources. They both operate on a node-scoped variable named $my_nagios_purge_hosts, which should contain an array of hostnames. If used, this variable should be placed somewhere in your Nagios server's node definition. For example:

```
node corona {
  $my_nagios_purge_hosts = [ 'foo', 'bar', 'baz' ]
  include nagios
}
```

With this node-scoped variable defined, your (affectionately named) Nagios server will reconfigure itself after dropping all resources for the three hosts mentioned above (Listings 9 and 10).

And, now for the refactored nagios class and related code (Listings 11–14).

The basic building blocks are now in place. Extend nagios::resources, plug the classes in to your nagios module and kick back. If a node goes MIA and

Listing 11. modules/nagios/manifests/init.pp

```
# This class will be used by the nagios server
class nagios {

  include nagios::params
  require nagios::expire_resources
  include nagios::purge_resources

  service { $nagios::params::service:
    ensure => running,
    enable => true,
  }

  # nagios.cfg needs this specified via the cfg_dir directive
  file { $nagios::params::resource_dir:
    ensure => directory,
    owner => $nagios::params::user,
  }

  # Local Nagios resources
  nagios::resource { [ 'Nagios Servers', 'Puppet Servers', 'Other' ]:
    type => hostgroup,
    export => false;
  }
}
```

needs to be purged, toss it into your $my_nagios_purge_hosts array and be done with it. Until next time, may your Nagios dashboards be green and your alerts be few.■

Adam Kosmin is a longtime Free Software advocate with 15+ years of professional systems engineering experience. He is currently employed by Reliant Security, Inc., where he leverages Puppet extensively. Adam was one of the featured speakers at PuppetConf 2011 where he spoke about his approach to managing thousands of nodes across isolated customer environments via a central source tree.

Listing 12. modules/nagios/manifests/expire_resources.pp

```
class nagios::expire_resources {

  if $my_nagios_purge_hosts {
    expire_exported($my_nagios_purge_hosts)
  }
}
```

Listing 13. modules/nagios/manifests/purge_resources.pp

```
class nagios::purge_resources {

  require nagios::collect_resources

  if $my_nagios_purge_hosts {
    purge_exported($my_nagios_purge_hosts)
  }
}
```

Listing 14. modules/nagios/manifests/collect_resources.pp

```
class nagios::collect_resources {

  include nagios::params

  Nagios_host <<||>> {
    require => $nagios::params::resource_dir,
    notify => Service[$nagios::params::service],
  }

  File <<| tag == nagios_host |>> {
    notify => Service[$nagios::params::service],
  }
}
```

LinuxFest Northwest

Bellingham, WA
April 28th & 29th

Grassroots Linux gathering
Exhibits of all flavors
Presentations of all levels
Prizes and after party
FREE admission & parking
FREE open source software
Bring the whole family!

Hosted By
Bellingham
TECHNICAL
COLLEGE

linuxfestnorthwest.org

AHEAD OF THE PACK: the Pacemaker High-Availability Stack

A high-availability stack serves one purpose: through a redundant setup of two or more nodes, ensure service availability and recover services automatically in case of a problem. Florian Haas explores Pacemaker, the state-of-the-art high-availability stack on Linux.

FLORIAN HAAS

Hardware and software are error-prone. Eventually, a hardware issue or software bug will affect any application. And yet, we're increasingly expecting services—the applications that run on top of our infrastructure—to be up 24/7 by default. And if we're not expecting that, our bosses and our customers are. What makes this possible is a high-availability stack: it automatically recovers applications and services in the face of software and hardware issues, and it ensures service availability and uptime. The definitive open-source high-availability stack for the Linux platform builds upon the Pacemaker cluster resource manager. And to ensure maximum service availability, that stack consists of four layers: storage, cluster communications, resource management and applications.

Cluster Storage

The storage layer is where we keep our data. Individual cluster nodes access this data in a joint and coordinated fashion. There are two fundamental types of cluster storage:

1. Single-instance storage is perhaps the more conventional form of cluster storage. The cluster stores all its data in one centralized instance, typically a volume on a SAN. Access to this data is either from one node at a time (active/passive) or from multiple nodes simultaneously (active/active). The latter option normally requires the use of a shared-cluster filesystem, such as GFS2 or OCFS2. To prevent uncoordinated access to data—a surefire way of shredding it—all single-instance storage cluster architectures require the use of fencing. Single-instance storage is very easy to set up, specifically if you already have a SAN at your disposal, but it has a very significant downside: if, for any reason, data becomes inaccessible or is even destroyed, all server redundancy in your high-availability architecture comes to naught. With no data to serve, a server becomes just a piece of iron with little use.

2. Replicated storage solves this problem. In this architecture, there are two or more replicas of the cluster data set, with each cluster node having access to its own copy of the data. An underlying replication facility then guarantees that the copies are exactly identical at the block layer. This effectively makes replicated storage a drop-in replacement for single-instance storage, albeit with added redundancy at the data level. Now you can lose entire nodes—with their data—and still have more nodes to fail over to. Several proprietary (hardware-based) solutions exist for this purpose, but the canonical way of achieving replicated block storage on Linux is the Distributed Replicated Block Device (DRBD). Storage replication also may happen at the filesystem level, with GlusterFS and Ceph being the most prominent implementations at this time.

Cluster Communications

The cluster communications layer serves three primary purposes: it provides reliable message passing between cluster nodes, establishes the cluster membership and determines quorum. The default cluster communications layer in the Linux HA stack is Corosync, which evolved out of the earlier, now all but defunct, OpenAIS Project.

Corosync implements the Totem single-ring ordering and membership protocol, a well-studied message-passing algorithm with almost 20 years of research among its credentials. It provides a secure, reliable means of message passing

Since these humble beginnings, however, Pacemaker has grown into a tremendously useful, hierarchical, self-documenting text-based shell, somewhat akin to the "virsh" subshell that many readers will be familiar with from libvirt.

that guarantees in-order delivery of messages to cluster nodes. Corosync normally transmits cluster messages over Ethernet links by UDP multicast, but it also can use unicast or broadcast messaging, and even direct RDMA over InfiniBand links. It also supports redundant rings, meaning clusters can use two physically independent paths to communicate and transparently fail over from one ring to another.

Corosync also establishes the cluster membership by mutually authenticating nodes, optionally using a simple pre-shared key authentication and encryption scheme. Finally, Corosync establishes quorum—it detects whether sufficiently many nodes have joined the cluster to be operational.

Cluster Resource Management

In high availability, a resource can be something as simple as an IP address that "floats" between cluster nodes, or something as complex as a database instance with a very intricate configuration. Put simply, a resource is anything that the cluster starts, stops, monitors, recovers or moves around. Cluster resource management is what performs these tasks for us—in an automated, transparent, highly configurable way. The canonical cluster resource manager in high-availability Linux is Pacemaker.

Pacemaker is a spin-off of Heartbeat, the high-availability stack formerly driven primarily by Novell (which then owned SUSE) and IBM. It re-invented itself as an independent and much more community-driven project in 2008, with developers from Red Hat, SUSE and NTT now being the most active contributors.

Pacemaker provides a distributed Cluster Information Base (CIB) in which it records the configuration and status of all cluster resources. The CIB replicates automatically to all cluster nodes from the Designated Coordinator (DC)—one node that Pacemaker automatically elects from all available cluster nodes.

The CIB uses an XML-based configuration format, which in releases prior to Pacemaker 1.0 was the only way to configure the cluster—something that rightfully made potential users run away screaming. Since these humble beginnings, however, Pacemaker has grown into a tremendously useful, hierarchical, self-documenting text-based shell, somewhat akin to the "virsh" subshell that many readers will be familiar with from libvirt. This shell—unimaginatively called "crm" by its developers—hides all that nasty XML from users and makes the cluster much simpler and easier to configure.

In Pacemaker, the shell allows us to configure cluster resources—no surprise there—and operations (things the cluster does with resources). Besides, we can set per-node and cluster-wide attributes, send nodes into a standby mode where they are temporarily ineligible for running resources, manipulate resource placement in the cluster, and do a plethora of other things to manage our cluster.

Finally, Pacemaker's Policy Engine (PE) recurrently checks the cluster configuration against the cluster status and initiates actions as required. The PE would, for example, kick off a recurring monitor operation on a resource (such as, "check whether this database is still alive"); evaluate its status ("hey, it's not!"); take into account other items in the cluster configuration ("don't attempt to recover this specific resource in place if it fails more than three times in 24 hours"); and initiate a follow-up action ("move this database to a different node"). All these steps are entirely automatic and require no human intervention, ensuring quick resource recovery and maximum uptime.

At the cluster resource management level, Pacemaker uses an abstract model where resources all support predefined, generic operations (such as start, stop or check the status) and produce standardized return codes. To translate these abstract operations into something that is actually meaningful to an application, we need resource agents.

Resource Agents

Resource agents are small pieces of code that allow Pacemaker to interact with an application and manage it as a cluster resource. Resource agents can be written in any language, with the vast majority being simple shell scripts. At the time of this writing, more than 70 individual resource agents ship with the high-availability stack proper. Users can, however, easily drop in custom resource agents—a key design principle in the Pacemaker stack is to make resource management easily accessible to third parties.

Resource agents translate Pacemaker's generic actions into operations meaningful for a specific resource type. For something as simple as a virtual "floating" IP address, starting up the resource amounts to assigning that address to a network interface. More complex resource types, such as those managing database instances, come with much more intricate startup operations. The same applies to varying implementations of resource shutdown, monitoring and migration: all these operations can range from simple to complex, depending on resource type.

Highly Available KVM: a Simple Pacemaker Cluster

This reference configuration consists of a three-node cluster with single-instance iSCSI storage. Such a configuration is easily capable of supporting more than 20 highly available virtual machine instances, although for the sake of simplicity, the configuration shown here includes only three. You can complete this configuration on any recent Linux distribution—the Corosync/Pacemaker stack is universally available on CentOS 6,[1] Fedora, OpenSUSE and SLES, Debian, Ubuntu and other platforms. It is also available in RHEL 6, albeit as a currently unsupported Technology Preview. Installing the `pacemaker`, `corosync`, `libvirt`, `qemu-kvm` and `open-iscsi` packages should be enough on all target platforms—your preferred package manager will happily pull in all package dependencies.

This example assumes that all three cluster nodes—alice, bob and charlie—have iSCSI access to a portal at 192.168.122.100:3260, and are allowed to connect to the iSCSI target whose IQN is `iqn.2011-09.com.hastexo:virtcluster`. Further, three libvirt/KVM virtual machines—xray, yankee and zulu—have been pre-installed, and each uses one of the volumes (LUNs) on the iSCSI target as its virtual block device. Identical copies of the domain configuration files exist in the default configuration directory, /etc/libvirt/qemu, on all three physical nodes.

Corosync

Corosync's configuration files live in /etc/corosync, and the central configuration is in /etc/corosync/corosync.conf. Here's an example of the contents of this file:

```
totem {
  # Enable node authentication & encryption
  secauth: on

  # Redundant ring protocol: none, active, passive.
  rrp_mode: active

  # Redundant communications interfaces
  interface {
    ringnumber: 0
    bindnetaddr: 192.168.0.0
    mcastaddr: 239.255.29.144
    mcastport: 5405
  }
  interface {
    ringnumber: 1
    bindnetaddr: 192.168.42.0
    mcastaddr: 239.255.42.0
    mcastport: 5405
  }
}

amf {
  mode: disabled
}

service {
  # Load Pacemaker
  ver: 1
  name: pacemaker
}

logging {
  fileline: off
  to_stderr: yes
  to_logfile: no
  to_syslog: yes
  syslog_facility: daemon
  debug: off
  timestamp: on
}
```

The important bits here are the two interface declarations enabling redundant cluster communications and the corresponding `rrp_mode` declaration. Mutual node authentication and encryption (`secauth on`) is good security practice. And finally, the `service` stanza loads the Pacemaker cluster manager as a Corosync plugin.

With secauth enabled, Corosync also requires a shared secret for mutual node authentication. Corosync uses a simple 128-byte secret that it stores as /etc/corosync/authkey, and which you easily can generate with the `corosync-keygen` utility.

Once corosync.conf and authkey are in shape, copy them over to all nodes in your prospective cluster. Then, fire up Corosync cluster communications—a simple `service corosync start` will do.

Once the service is running on all nodes, the command `corosync-cfgtool -s` will display both rings as healthy, and the cluster is ready to communicate:

```
Printing ring status.
Local node ID 303938909
RING ID 0
        id      = 192.168.0.1
        status  = ring 0 active with no faults
RING ID 1
        id      = 192.168.42.1
        status  = ring 1 active with no faults
```

Pacemaker

Once Corosync runs, we can start Pacemaker with the `service pacemaker start` command. After a few seconds, Pacemaker elects a Designated Coordinator (DC) node among the three available nodes and commences full cluster operations. The `crm_mon` utility, executable on any cluster node, then produces output similar to this:

```
============
Last updated: Fri Feb  3 18:40:15 2012
Stack: openais
Current DC: bob - partition with quorum
Version: 1.1.6-4.el6-89678d4947c5bd466e2f31acd58ea4e1edb854d5
3 Nodes configured, 3 expected votes
0 Resources configured.
============
```

Much less intimidating is the standard configuration facility for Pacemaker, the crm shell.

The output produced by `crm_mon` is a more user-friendly representation of the internal cluster configuration and status stored in a distributed XML database, the Cluster Information Base (CIB). Those interested and brave enough to care about the internal representation are welcome to make use of the `cibadmin -Q` command. But be warned, before you do so, you may want to instruct the junior sysadmin next to you to get some coffee—the avalanche of XML gibberish that `cibadmin` produces can be intimidating to the uninitiated novice.

Much less intimidating is the standard configuration facility for Pacemaker, the crm shell. This self-documenting, hierarchical, scriptable subshell is the simplest and most universal way of manipulating Pacemaker clusters. In its configure submenu, the shell allows us to load and import configuration snippets—or even complete configurations, as below:

```
primitive p_iscsi ocf:heartbeat:iscsi \
  params portal="192.168.122.100:3260" \
    target="iqn.2011-09.com.hastexo:virtcluster" \
  op monitor interval="10"
primitive p_xray ocf:heartbeat:VirtualDomain \
  params config="/etc/libvirt/qemu/xray.xml" \
  op monitor interval="30" timeout="30" \
  op start interval="0" timeout="120" \
  op stop interval="0" timeout="120"
primitive p_yankee ocf:heartbeat:VirtualDomain \
  params config="/etc/libvirt/qemu/yankee.xml" \
  op monitor interval="30" timeout="30" \
  op start interval="0" timeout="120" \
  op stop interval="0" timeout="120"
primitive p_zulu ocf:heartbeat:VirtualDomain \
  params config="/etc/libvirt/qemu/zulu.xml" \
  op monitor interval="30" timeout="30" \
  op start interval="0" timeout="120" \
  op stop interval="0" timeout="120"
clone cl_iscsi p_iscsi
colocation c_xray_on_iscsi inf: p_xray cl_iscsi
colocation c_yankee_on_iscsi inf: p_yankee cl_iscsi
colocation c_zulu_on_iscsi inf: p_zulu cl_iscsi
order o_iscsi_before_xray inf: cl_iscsi p_xray
order o_iscsi_before_yankee inf: cl_iscsi p_yankee
order o_iscsi_before_zulu inf: cl_iscsi p_zulu
```

Besides defining our three virtual domains as resources under full cluster management and monitoring (`p_xray`, `p_yankee` and `p_zulu`), this configuration also ensures that all domains can find their storage (the `cl_iscsi` clone), and that they wait until iSCSI storage becomes available (the `order` and `colocation` constraints).

This being a single-instance storage

cluster, it's imperative that we also employ safeguards against shredding our data. This is commonly known as node fencing, but Pacemaker uses the more endearing term STONITH (Shoot The Other Node In The Head) for the same concept. A ubiquitous means of node fencing is controlling nodes via their IPMI Baseboard Management Controllers, and Pacemaker supports this natively:

```
primitive p_ipmi_alice stonith:external/ipmi \
  params hostname="alice" ipaddr="192.168.15.1" \
    userid="admin" passwd="foobar" \
  op start interval="0" timeout="60" \
  op monitor interval="120" timeout="60"
primitive p_ipmi_bob stonith:external/ipmi \
  params hostname="bob" ipaddr="192.168.15.2" \
    userid="admin" passwd="foobar" \
  op start interval="0" timeout="60" \
  op monitor interval="120" timeout="60"
primitive p_ipmi_charlie stonith:external/ipmi \
  params hostname="charlie" ipaddr="192.168.15.3" \
    userid="admin" passwd="foobar" \
  op start interval="0" timeout="60" \
  op monitor interval="120" timeout="60"
location l_ipmi_alice p_ipmi_alice -inf: alice
location l_ipmi_bob p_ipmi_bob -inf: bob
location l_ipmi_charlie p_ipmi_charlie -inf: charlie
property stonith-enabled="true"
```

The three location constraints here ensure that no node has to shoot itself.

Once that configuration is active, Pacemaker fires up resources as determined by the cluster configuration. Again, we can query the cluster state with the `crm_mon` command, which now produces much more interesting output than before:

```
============
Last updated: Fri Feb  3 19:46:29 2012
Stack: openais
Current DC: bob - partition with quorum
Version: 1.1.6-4.el6-89678d4947c5bd466e2f31acd58ea4e1edb854d5
3 Nodes configured, 3 expected votes
9 Resources configured.
============

Online: [ alice bob charlie ]

 Clone Set: cl_iscsi [p_iscsi]
     Started: [ alice bob charlie ]
 p_ipmi_alice   (stonith:external/ipmi):        Started bob
 p_ipmi_bob     (stonith:external/ipmi):        Started charlie
 p_ipmi_charlie    (stonith:external/ipmi):        Started alice
 p_xray   (ocf::heartbeat:VirtualDomain):       Started alice
 p_yankee       (ocf::heartbeat:VirtualDomain): Started bob
 p_zulu   (ocf::heartbeat:VirtualDomain):       Started charlie
```

Note that by default, Pacemaker clusters are symmetric. The resource manager balances resources in a round-robin fashion among cluster nodes.

This configuration protects against both resource and node failure. If one of the virtual domains crashes, Pacemaker recovers the KVM instance in place. If a whole node goes down, Pacemaker reshuffles the resources so the remaining nodes take over the services that the failed node hosted.

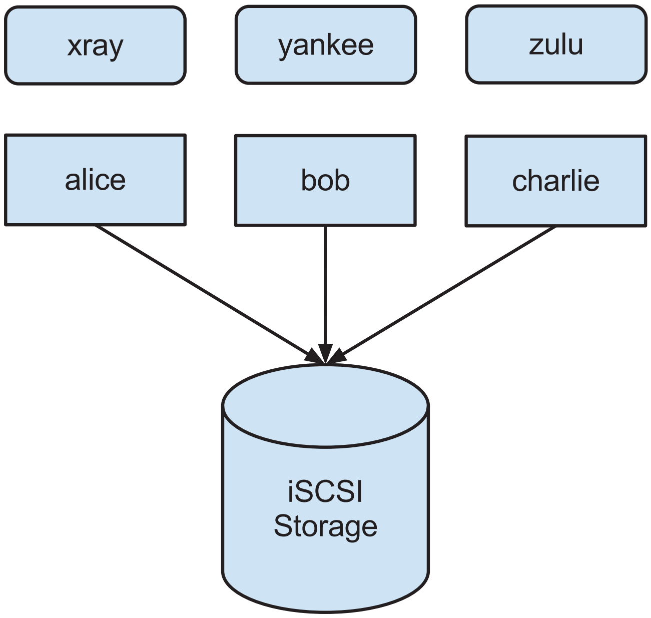


Figure 1. Normal operation: virtual domains spread across all three cluster nodes.

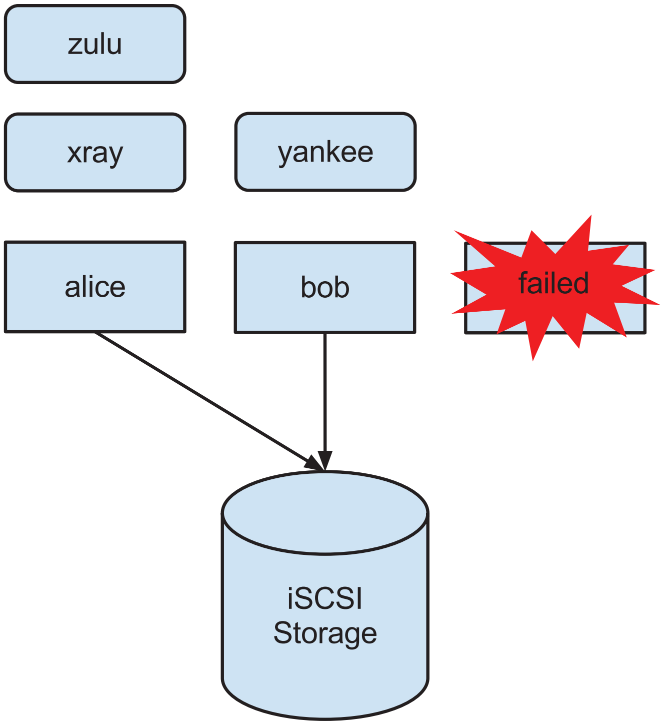


Figure 2. Node charlie has failed; alice has automatically taken over virtual domain zulu.

GLOSSARY

Node: in cluster terminology, any system (typically a server) that participates in cluster communications and can potentially host cluster resources.

Fencing: a means of coordinating access to shared resources in the face of communications failure. Once a node stops responding to cluster messages unexpectedly (as opposed to gracefully signing off), other nodes shut it down to ensure it no longer has access to any shared resources. Usually enabled by making an out-of-band connection to the offending node and flipping the virtual power switch, IPMI-over-LAN being the most widely used implementation.

Resource: anything that a cluster typically manages. Resources can be very diverse, from simple IP addresses to complex database instances or dæmons.

Ring: in Totem protocol terminology, one of the (typically redundant) links over which cluster messaging communicates.

In the screen dump below, charlie has failed and bob has duly taken over the virtual machine that charlie had hosted:

```
============
Last updated: Sat Feb  4 16:18:00 2012
Stack: openais
Current DC: bob - partition with quorum
Version: 1.1.6-4.el6-89678d4947c5bd466e2f31acd58ea4e1edb854d5
3 Nodes configured, 2 expected votes
9 Resources configured.
============

Online: [ alice bob ]
OFFLINE: [ charlie ]

Full list of resources:

 Clone Set: cl_iscsi [p_iscsi]
     Started: [ alice bob ]
     Stopped: [ p_iscsi:2 ]
 p_ipmi_alice   (stonith:external/ipmi):       Started bob
 p_ipmi_bob     (stonith:external/ipmi):       Started alice
 p_ipmi_charlie (stonith:external/ipmi):       Started alice
 p_xray   (ocf::heartbeat:VirtualDomain):      Started bob
 p_yankee      (ocf::heartbeat:VirtualDomain): Started bob
 p_zulu   (ocf::heartbeat:VirtualDomain):      Started alice
```

Once the host charlie recovers, resources can optionally shift back to the recovered host automatically, or they can stay in place until an administrator reassigns them at the time of her choosing.

In this article, I barely scratched the surface of the Linux high-availability stack's capabilities. Pacemaker supports a diverse set of recovery policies, resource placement strategies and cluster constraints, making the stack enormously powerful.■

Florian Haas is a Principal Consultant and co-founder at hastexo, an independent professional services organization specializing in Linux high availability, storage replication and the open-source cloud. He has been active in this community for half a decade, and has been called one of "the jedi grandmasters" of Linux high availability. He frequently speaks on high availability and cloud topics at technical conferences.

Resources

"The Totem Single-Ring Ordering and Membership Protocol" (research paper explaining the Totem protocol): **http://www.cs.jhu.edu/~yairamir/tocs.ps**

"Clusters From Scratch" (hands-on documentation for Pacemaker novices): **http://www.clusterlabs.org/doc/en-US/Pacemaker/1.1/html/Clusters_from_Scratch**

"Pacemaker Configuration Explained" (reference documentation of Pacemaker configuration internals, not for the faint at heart): **http://www.clusterlabs.org/doc/en-US/Pacemaker/1.1/html/Pacemaker_Explained**

Using Linux with EFI, Part IV: Managing an EFI Installation

Troubleshooting and other EFI maintenance. RODERICK W. SMITH

In my last article, I described the process of installing Linux on an EFI-based computer. With any luck, that should have gotten your system up and running; however, a number of problems can occur during or after installation. In this article, I look at these issues, as well as general EFI maintenance. Specifically, I describe how to use an emergency disc on an EFI system, how to make your firmware recognize your bootloader, how to configure GRUB and rEFIt to chainload to Windows (or to other OSes), how to update your kernel and how to deal with miscellaneous kernel problems.

Booting an Emergency Disc

If you're like me, you keep a number of Linux emergency discs on hand. These include both distribution installation discs, many of which contain recovery systems, and specialized tools, such as Parted Magic, SystemRescueCD and RIPLinuX. Such tools can be invaluable aids in recovering from various system problems.

Unfortunately, most emergency discs don't yet include EFI boot support, the rescue mode of distribution installers being an exception to this rule. Thus, if you want to use a rescue disc, such as Parted Magic, SystemRescueCD or RIPLinuX, you may need to do so in BIOS mode. This is usually not difficult. Most EFI implementations automatically switch to BIOS mode when they detect BIOS-bootable optical discs; however, some might need some coaxing to boot the disc. Check your firmware boot options if you have problems booting a rescue disc in BIOS mode.

For the most part, using a BIOS-mode emergency disc is fine. You can mount or repair your filesystems, edit configuration files, modify your partition layout and even edit your EFI bootloader configuration. There is, however, one important exception to this rule: the efibootmgr program, which you use to register a bootloader with the EFI, is useless when the computer is booted in BIOS mode. This program also is missing from the usual emergency disc programs.

If you're having problems that necessitate using efibootmgr, therefore, your best bet may be to use a distribution's installer in EFI mode. For this purpose, you even can use a distribution other than the one you normally use. Ubuntu includes a good live CD mode, and you can install software to it using apt-get, as in `sudo apt-get install efibootmgr`. Thereafter, you can use efibootmgr to manage bootloaders, even if your computer doesn't normally run Ubuntu.

Making EFI Recognize Your Bootloader

One common post-installation problem on EFI computers is that the computer doesn't recognize your bootloader. The computer might inform you that it couldn't find a bootable OS, or it might boot directly into Windows, OS X or whatever other OS you had installed previously. Several solutions to this problem exist, such as renaming your bootloader to the default name or using efibootmgr to manage your EFI's list of bootloaders.

If no other OS exists on the computer, the simplest solution to the problem is to rename your bootloader to use the default bootloader name, EFI/BOOT/bootx64.efi on the EFI System Partition (ESP). As described in Part II of this series, this is the filename the EFI uses when you haven't registered any other bootloader with the firmware. Thus, you can rename your bootloader's default location (such as EFI/SuSE or EFI/ubuntu) to EFI/BOOT and then rename the bootloader file itself (such as grub.efi or elilo.efi) to bootx64.efi. In the case of GRUB Legacy, you should rename the configuration file from grub.conf to bootx64.conf too.

If the computer boots straight into another OS, you may be able to rename an existing EFI/BOOT directory and then rename your Linux bootloader as just described. Alternatively, you can boot Linux in EFI mode using an EFI-enabled emergency system and use efibootmgr to add your bootloader to

Table 1. The efibootmgr utility manages the EFI's list of bootloaders.

Long Option	Short Option	Description
--active	-a	Activates a bootloader
--inactive	-A	Deactivates a bootloader
--bootnum *X*	-b *X*	Modifies bootloader*X* (hexadecimal)
--delete-bootnum	-B	Deletes a bootloader
--create	-c	Creates a new bootloader entry
--disk *dev*	-d *dev*	Sets the disk to modify (default is /dev/sda)
--loader *name*	-l *name*	Specifies a bootloader's filename
--label *label*	-L *label*	Specifies a label for the bootloader
--bootnext *X*	-n *X*	Sets the bootloader to use on the next boot
--bootorder *order*	-o *order*	Sets the order in which the EFI tries bootloaders
--part *num*	-p *num*	Specifies the ESP's partition number (defaults to 1)
--timeout *sec*	-t *sec*	Sets the boot managers timeout, in seconds
--verbose	-v	Produces extra output

the EFI's boot list:

```
efibootmgr -c -l \\EFI\\elilo\\elilo.efi -L ELILO
```

The details vary depending on the bootloader, of course. You may need to change the path to the bootloader, which is specified using double backslashes (\\) rather than the more conventional forward slash (/). You can use efibootmgr to manipulate the boot list in other ways too. Table 1 summarizes some of efibootmgr's options, but you can consult its man page for more obscure options.

If you use efibootmgr without any options, it displays a summary of the system's current bootloaders. You can add the -v option to produce additional information. The preceding example used the -c option to create a new boot manager entry, thus adding ELILO to the list of boot options that the firmware provides. This command, though, might leave another bootloader earlier in the EFI's boot order. If you want to change that order, you would use the -o option, as in:

```
efibootmgr -o 5,A,3
```

This example tells the EFI to attempt to boot using bootloader #5; if that fails, to try #A (hexadecimal); and if that fails, to try #3. The -n option sets the bootloader to use on the next boot, overriding the default boot order for that boot.

Some options require you to use the -b option to identify a specific bootloader. For instance, if you want to give a bootloader a new name in a menu, you might type the following:

```
efibootmgr -b 3 -L "Windows 7"
```

This example gives entry #3 the label "Windows 7".

Unfortunately, efibootmgr's effects are not 100% reliable. For instance, on one of my computers, the -o option has no real effect. I must use the firmware's own interface to adjust the boot order. More distressingly, some reports indicate that efibootmgr damages some Macs' firmware, requiring re-flashing the firmware. (The OS X "bless" utility serves a role similar to efibootmgr on Macs.) Nonetheless, it's worth trying efibootmgr if you want to add bootloaders or adjust the boot order on a UEFI-based PC.

Configuring GRUB to Chainload to Windows

If you're using GRUB Legacy or GRUB 2 and you want to dual-boot with Windows, you may need to add an entry to the bootloader to make it work correctly. On a BIOS-based computer, this entry would reference the Windows boot partition, where part of the Windows bootloader resides. On an EFI-based computer though, you must reference the Windows bootloader on the ESP. A suitable entry in GRUB Legacy's grub.conf file looks like this:

```
title Windows 7
        root (hd0,0)
        chainloader /EFI/microsoft/bootmgfw.efi
```

This entry might need to be changed if your ESP is not the first partition on the first disk, or if the bootloader isn't in its usual location. An equivalent entry in GRUB 2's grub.cfg file looks like this:

```
menuentry "Windows 7" {
        set root='(hd0,gpt1)'
        chainloader /EFI/microsoft/bootmgfw.efi
}
```

In GRUB 2, you probably would add such an entry to /etc/grub.d/40_custom and then rebuild grub.cfg by using update-grub or some other configuration script.

You can chainload other bootloaders in a similar way. One twist on Macs is that the OS X bootloader resides on the OS X boot partition, as /System/Library/CoreServices/boot.efi. GRUB Legacy can't mount Mac OS X's HFS+ though, so you can't chainload OS X using GRUB Legacy. GRUB 2 can mount HFS+ and so can chainload OS X in this way. GRUB 2 also can load the OS X kernel directly, and your GRUB 2 configuration scripts might create a rather wordy entry that does just that. If you want to dual-boot Linux and OS X though, rEFIt is generally the best way to do the job. It gives you a graphical menu showing OS X and your Linux bootloader.

Configuring rEFIt to Chainload to Windows

If you use ELILO to boot Linux, you'll need to use another boot manager to boot non-Linux OSes. Although EFI implementations

Secure Boot and UEFI

In September 2011, a small firestorm erupted concerning an obscure UEFI feature known as Secure Boot. This feature, if enabled, causes the firmware to boot only bootloaders that have been cryptographically signed using a key stored in the firmware's Flash memory. The intent of Secure Boot is to block one avenue that malware has used for decades: to load before the OS loads, hang around and wreak havoc.

Microsoft is requiring that computers bearing its Windows 8 logo ship with Secure Boot enabled. The problem is that the Linux community is very diverse and has no centralized key, and manufacturers might not include keys for individual Linux distributions. Thus, Secure Boot could become an extra obstacle for installing Linux.

If you have a computer with Secure Boot enabled, and if this feature is preventing you from booting Linux, you can look for an option to disable this feature or to import or generate a key for your Linux distribution. Unfortunately, I can't be more precise about how to work around this problem, since computers with Secure Boot are still extremely rare.

In a worst-case scenario, a computer that ships with Windows 8 might lack the ability to disable Secure Boot or to add your own keys. In such a case, you won't be able to install Linux on the computer—at least, not without replacing the firmware or hacking it in some way. If you've got such a computer, your best bet is to return it to the store for a refund.

provide boot managers for this purpose, they're generally quite awkward, so you may want to use rEFIt to present a friendlier menu for dual booting.

As described in Part I of this series, rEFIt can chainload another bootloader, but it can't load a Linux kernel itself. You don't need to configure rEFIt explicitly to boot any OS; it scans all the partitions it can read for EFI bootloader files and generates a menu automatically. Thus, rEFIt should detect Microsoft's bootloader file (called bootmgfw.efi) and label it appropriately.

The trick is in rEFIt installation. The version available from the rEFIt Web site is a "fat binary" (32-/64-bit) that works only on Macs. To use rEFIt on a UEFI-based PC, you need a 64-bit-only version. Debian and Ubuntu both ship with such versions; check the Resources section of this article for another source.

Updating Your Kernel

Assuming you've successfully installed Linux on an EFI-based computer, you can use the system just as you'd use a BIOS-based installation. Most operations, including most software updates, will work just as they do on a BIOS-based computer. One possible exception to this rule is in kernel updates.

Most distributions include scripts that automatically update your bootloader configuration to add a new kernel when you install one via the distribution's package management system. This process usually works fine if you use the distribution's own bootloader; however, as detailed in previous articles in this series, you're more likely to have to deviate from your distribution's default bootloader if you use EFI than if you use BIOS. If this happens, you may need to manage the bootloader configuration yourself manually.

Details of bootloader configuration vary with the bootloader. Part II of this series described ELILO configuration, so you can use it as a reference if you use ELILO. If you use GRUB Legacy or GRUB 2, any source of documentation on those bootloaders under BIOS applies almost as well to the EFI version. The main exception is in handling chainloading, as noted earlier.

If you manage your bootloader yourself, the biggest problem is to note carefully when you upgrade your kernel so that you can make suitable changes to your bootloader configuration. If you miss an update, you'll end up booting an old kernel. If your distribution uninstalls your old kernel, your system might end up being unbootable. If you're comfortable compiling your own kernel, you might consider doing so. That should give you a kernel you manage yourself outside the package system, which should give you a secure backup kernel.

Dealing with Kernel and Video Problems

For the most part, Linux works equally well on BIOS-based and EFI-based computers. The kernel interacts directly with hardware; once it's booted, it has little need to use the firmware. There are exceptions to this rule though. Sometimes, kernel options (passed via your bootloader configuration) can be helpful. Specifically:

- `noefi` — this option disables the kernel's support for EFI runtime services. Ordinarily, you wouldn't want to do this, but sometimes bugs cause Linux's EFI runtime services to misbehave, causing a failure to boot or other problems. Using the noefi option can work around such problems, at the cost of an inability to use the efibootmgr utility.

- `reboot_type=k` — this option disables the EFI's reboot runtime service. This option can work around problems with failed reboot operations.

- `reboot=e` — this option is essentially the opposite of the preceding one; it tells the kernel to attempt to reboot using the EFI's reboot runtime service. It's normally the default on EFI boots, but you can try it if Linux is rebooting improperly.

- `reboot=a,w` — this option tells Linux to perform warm reboots using ACPI. This can work around bugs in some EFI implementations that cause a system hang when rebooting.

- `add_efi_memmap` — this option alters how Linux allocates memory, which can avoid memory conflicts with some EFI implementations.

- `nomodeset` — this option sometimes works around problems that cause a blank video display.

EFI implementations vary among themselves, and the Linux kernel has yet to incorporate surefire workarounds for every quirk and bug. Thus, you may need to experiment with these and perhaps with other options.

If you build your own kernel, you should be sure to include a few options:

- `CONFIG_EFI` — this option, near the end of the Processor Type and Features kernel configuration area, enables Linux to use the EFI runtime services.

- `CONFIG_FB_EFI` — EFI provides its own framebuffer device, and this option (available in the Graphics Support→Support for Frame Buffer Devices kernel configuration area) enables the kernel to use the EFI's framebuffer. You often can use a more device-specific framebuffer driver, but this one can be an important fallback on some computers.

ATLANTA ASTERISK USERS GROUP | ATLAUG.COM

VoIP CONFERENCE

Saturday, April 14 | 9AM - 5PM ET

>> Register today to attend! http://atlaug.com <<

Lecture Hall A - Presentation Track - Times and Order are Subject to Change!

09:00 AM - Steven Henke, Xelatec.com, Mary Clare, gtisc.gatech.edu - Welcome & Announcements
09:15 AM - Joe Roper, Star2Billing.com
09:45 AM - Bill Soto, Xorcom.com - Increasing Your Sales of Asterisk Based PBX Solutions
10:15 AM - Mike Storella, Snom.com - Subscriber Sets for Asterisk Systems
10:45 AM - Bryan Johns, Digium.com
11:00 AM - Break
11:15 AM - Marcus Graham, GMvoices.com
11:30 AM - Tom Ray, Sangoma.com
12:00 PM - Lunch - Courtesy of Sangoma
01:00 PM - Adam Wayment, Patrick Dexter, Cepstral.com
01:30 PM - Scott Navratil, Vitelity.com
01:45 PM - Mark Carson, Aastra.com - An Android SIP Phone
02:15 PM - Alex Balashov - Evaristesys.com - Special Presentation
02:45 PM - Ben Klang, MojoLingo.com - Ruby in the Dialplan
03:15 PM - Break
03:30 PM - Surprise Guests - Don't miss this!
04:00 PM - Presenters Round Table Question and Answer Session, Closing Remarks
05:00 PM - Food, Beverage and Fellowship at the 5 Seasons Restaurant & Brewery, 5seasons.info

AAUG

Lecture Hall B - Digium Asterisk 123 Training
9:00 AM - 5:00 PM

The Asterisk 123 course is a gentle introduction to the Asterisk Open Source PBX. It introduces the student to the many roles that Asterisk can play, and walks them through setting up Asterisk for the first time.

- `CONFIG_EFI_VARS` — this option, available in the Firmware Drivers kernel area, provides access to EFI variables via Linux's sysfs. You need this support to use efibootmgr.

- `CONFIG_EFI_PARTITION` — you'll find this option in the File Systems→Partition Types kernel area. It enables support for GPT, so of course you need it to boot from a GPT disk.

Some EFI-based computers present video difficulties. Such problems are particularly common with Macs booted in EFI mode. Typically, the video display goes blank in text mode, in X or in both. Unfortunately, there's no easy solution to such problems. Sometimes though, switching bootloaders can help. You also can try experimenting with framebuffer options and your X configuration. Typing `Xorg -configure` as root and with X *not* running generates an X configuration file, /root/xorg.conf.new. Copying and renaming this file to /etc/X11/xorg.conf sometimes gets X up and running.

Conclusion

EFI provides a number of improvements over the older BIOS it replaces. These include quicker boot times, an integrated boot manager and the ability to manage more complex bootloaders as files on a filesystem. In an ideal world, the transition from BIOS to EFI should be painless, but in the real world, developers require time to write the software to use EFI. At the moment, all of the pieces for handling EFI exist—you have a choice of EFI bootloaders, partitioning tools exist, the Linux kernel supports EFI, and you even can use efibootmgr to configure the EFI's boot manager. Distribution maintainers, however, have not yet worked out all the kinks in integrating all of these tools. A few programs also have bugs or are awkward to use. Knowing your way around these problems will help you make the transition from a BIOS-based computer to one built around EFI. I hope this series of articles helps you with that task.■

Roderick W. Smith is a Linux consultant, writer and open-source programmer living in Woonsocket, Rhode Island. He is the author of more than 20 books on Linux and other open-source technologies, as well as of the GPT fdisk (gdisk, cgdisk, and sgdisk) family of partitioning software.

Resources

Parted Magic: **http://partedmagic.com**

RIPLinuX: **http://www.tux.org/pub/people/kent-robotti/looplinux/rip**

SystemRescueCD: **http://www.sysresccd.org**

rEFIt: **http://refit.sourceforge.net**

Pure 32- and 64-Bit Builds of rEFIt That Work on UEFI-Based PCs: **http://www.rodsbooks.com/efi-bootloaders/refit.html**

DOC SEARLS

Looking Past Search

Can we make search organic again? Or should we look past search completely?

Searching has become shopping, whether we like it or not. That's the assumption behind the results, and behind recent changes, at least to Google's search features and algorithms. I'm sure this isn't what Google thinks, even though it is a commercial enterprise and makes most of its money from advertising—especially on searches. Credit where due: from the moment it started adding advertising to search results, Google has been careful to distinguish between paid results and what have come to be called "organic".

But alas, the Web is now a vast urban scape of storefronts and pitch-mills. The ratio of commercial to noncommercial sites has become so lopsided that Tim Berners-Lee's library-like World Wide Web of linked documents has been reduced to a collection of city parks, the largest of which is Wikipedia (which, like Google, runs on Linux, according to Netcraft). Without Wikipedia, organic results for many subjects would be pushed off the front page entirely.

Let's say you want to know more about the common geometric figure called a *square*. (Or that you'd like to explore one of the other organic meanings of *square*, such as *correct*, *level*, *honest* or *retro*.) You can find your answers using Google or Bing, but none will be the top "I'm feeling lucky" result if you just search for "square" alone. Instead, you'll get the name of a company.

For example, when I look up *square* on my Chrome browser while also logged in to my Google account, the engine produces a crufty URL more than 200 characters long. The top result for this search is the company Square.com. With that result, Google also tells me Robert Scoble gives the company a "+1", and that I have "110 personal results and 296,000,000 other results". Google's engine is now tweaked to think I might want those results because I've shared a bunch of personal data in the course of whatever I do in Google's presence. But, while I think Square.com is a fine and interesting company, that's not the result I'm looking for.

When I carve the cruft out of that URL (so it just says https://www.google.com/search?en&q=square), the result is the same, without the bonus item from Scoble or the other personalized "social" stuff. When I search on Bing (with which I have no logged-in relationship), I also get long a crufty URL, again with Square.com as the top result. Likewise, if I reduce Bing's URL to just http://www.bing.com/search?q=square, I'm still at square zero (pun intended).

In both search engines, the #2 or #3 result is Wikipedia's one for the geometric square. So, you might say, the systems work in any case: they just happen to make Square.com the top result, because page ranking is based mostly on inbound links, and Square.com gets more of those than any page that describes the geometric shape. And hey, if you're looking for "taxi" on the streets of New York, you're probably looking for a business with four wheels, not for a definition of one.

Search engines operate today in the Web's urban environment, which is at least as commercial as any brick-and-mortar one. Search

Advertiser Index

Thank you as always for supporting our advertisers by buying their products!

ADVERTISER	URL	PAGE #
AceNet	http://www.ace-host.net	29
Atlanta Asterisk Users Group VOIP CONFERENCE 2012	http://atlaug.com	115
Emac, Inc.	http://www.emacinc.com	35
iXsystems, Inc.	http://www.ixsystems.com	7
LinuxFest NorthWest	http://linuxfestnorthwest.org	97
Lullabot	http://www.lullabot.com	2
Microway, Inc.	http://www.microway.com	68, 69
O'Reilly Where	http://whereconf.com/where2012	117
RackMountPro	http://www.rackmountpro.com	15
Silicon Mechanics	http://www.siliconmechanics.com	3
SouthEast LinuxFest	http://www.southeastlinuxfest.org	37
USENIX	http://www.usenix.org	17

engines are also not libraries, even if libraries share Google's mission of "organizing the world's information and making it accessible and useful" (**http://www.google.com/about/company/**). Fact is, the Web isn't organized by search engines at all. It's only indexed, which isn't the same thing.

See, the Web actually is organized, to some degree. That is, it has a directory, with paths to locations. Those paths can branch to infinity, but at least each one points somewhere. Search engines do crawl the Web's directory, and they do index its locations, but the index they search is a meta-version of the actual path-based Web. That meta-Web is organized by search engines into a giant haystack made entirely of needles. The job of the search engine is to make good guesses about the needles you might be looking for. At that they mostly succeed, which is why we have come to depend on them. But we're still dealing with a haystack of meta, rather than the real thing. This is important to bear in mind if we ever want to find a better way.

Lately Google has become convinced that the better way is to make searches more "social". Its latest move in that direction is Search, plus Your World—a non-memorable name even when abbreviated to the unpronounceable SpYW (**http://googleblog.blogspot.com/2012/01/search-plus-your-world.html**). The "Your World" of which Google speaks is what you've organized mostly through Google + and Picasa, though it also indexes the "public-facing" stuff on sites such as Quora. But, if you're one of the 3/4 billion persons who live in the Matrix-like Facebook, SpYW won't help you, because Facebook won't let Google index your Facebook-based social stuff. Nor will it provide you with the option to expose it to SpYW or anything like it outside Facebook itself. Whatever the reasons for that, the effect is to partition the social Web into commercial zones with exclusivities that resemble AOL, CompuServe, Prodigy and other silo'd "on-line services" that the Web obsoleted back in the last millennium.

Why should we think that searches need to be social at all? There are lots of reasons, but the main one is advertising. Google, along with everybody else in the advertising business, wants to get personal with you. It wants to make the ads you see more relevant to you, personally. If it didn't want to do that, SpYW wouldn't exist.

Two problems here. One is that much of what we might search for (such as the case above) does not benefit from "social" help. In those cases, the social stuff is just more noise. The other is that our worlds, even on-line, are not just about stores, shopping and sites that live off advertising. They are much bigger than that.

In his new book, *Too Big to Know:*

Rethinking Knowledge Now That the Facts Aren't the Facts, Experts Are Everywhere, and the Smartest Person in the Room Is the Room, David Weinberger writes, "Knowledge is now a property of the network, and the network embraces businesses, governments, media, museums, curated collections, and minds in communications." Google still embraces all those things, but in ways that enlarge business at the expense of everything else. Google might not want to do that, but its advertising business can't help rewarding countless sites that live off on-line advertising, as well as businesses that advertise—while doing little for all the other stuff that comprises the Web.

I don't think the way to solve this is with a better search engine. I think we need something completely different—something that isn't so meta that we lose track of what the Web was in the first place, and why it is still an ideal way to keep and to find knowledge, and not just a place to shop.■

Doc Searls is Senior Editor of *Linux Journal*. He is also a fellow with the Berkman Center for Internet and Society at Harvard University and the Center for Information Technology and Society at UC Santa Barbara.

OLPC | Arduino | Trim-Slice | vDSO | Elmer | MPD
LINUX JOURNAL
ONE LAPTOP PER CHILD STATUS REPORT
USE YOUR PHONE TO LOCK YOUR SCREEN AUTOMATICALLY
HACK THIS
HOW TO: ADD A CUSTOM vDSO TO A LINUX KERNEL
Get Started Programming the Arduino
Create a Unified Boot Environment on a USB Drive
NETWORKING