LINUX JOURNAL™

Since 1994: The Original Magazine of the Linux Community

IMPROVE PERFORMANCE AND RELIABILITY WITH MYSQL REPLICATION

AN OVERVIEW OF THE TAILS DESKTOP FOR SECURITY

APRIL 2014 | ISSUE 240 | www.linuxjournal.com

HIGH-PERFORMANCE COMPUTING

A PROCESS FOR MANAGING AND CUSTOMIZING HPC OPERATIONS

LIGHTWEIGHT VIRTUALIZATION WITH LINUX-BASED CONTAINERS

JOB SCHEDULING IN HADOOP WITH YARN

WATCH: ISSUE OVERVIEW

LINUX™ JOURNAL

Since 1994: The Original Magazine of the Linux Community

IMPROVE PERFORMANCE AND RELIABILITY WITH MYSQL REPLICATION

AN OVERVIEW OF THE TAILS DESKTOP FOR SECURITY

APRIL 2014 | ISSUE 240 | www.linuxjournal.com

HIGH-PERFORMANCE COMPUTING

A PROCESS FOR MANAGING AND CUSTOMIZING HPC OPERATIONS

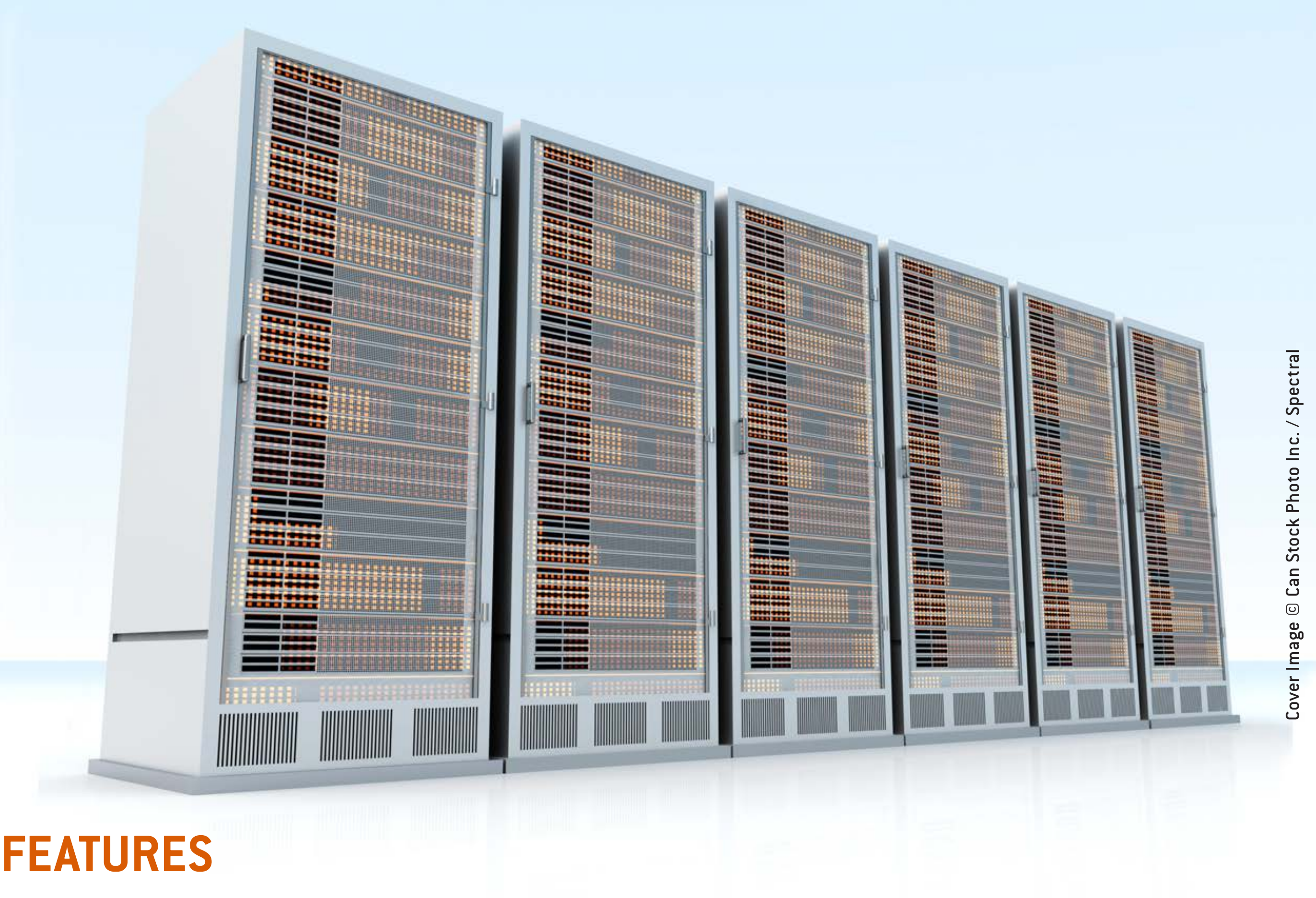
LIGHTWEIGHT VIRTUALIZATION WITH LINUX-BASED CONTAINERS

JOB SCHEDULING IN HADOOP WITH YARN

WATCH:
ISSUE OVERVIEW

CONTENTS APRIL 2014 ISSUE 240

HIGH-PERFORMANCE COMPUTING



Cover Image © Can Stock Photo Inc. / Spectral

FEATURES

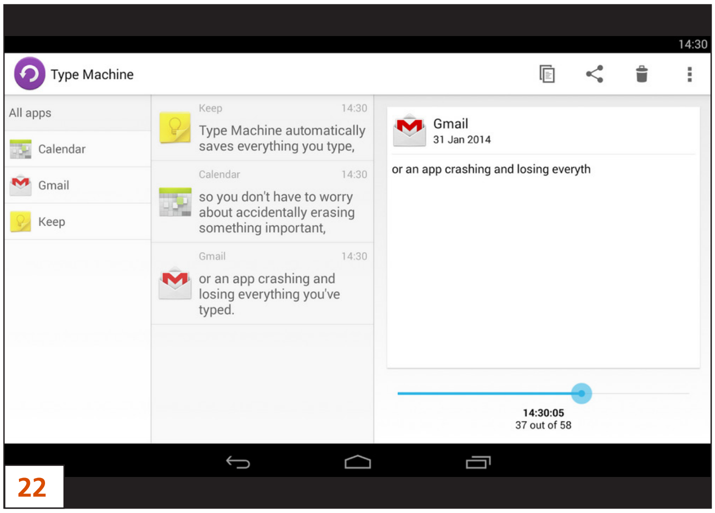
INDEPTH

COLUMNS

IN EVERY ISSUE

ON THE COVER

- Improve Performance and Reliability with MySQL Replication, p. 98
- An Overview of the Tails Desktop for Security, p. 44
- A Process for Managing and Customizing HPC Operations, p. 64
- Lightweight Virtualization with Linux-Based Containers, p. 86
- Job Scheduling in Hadoop with YARN, p. 74



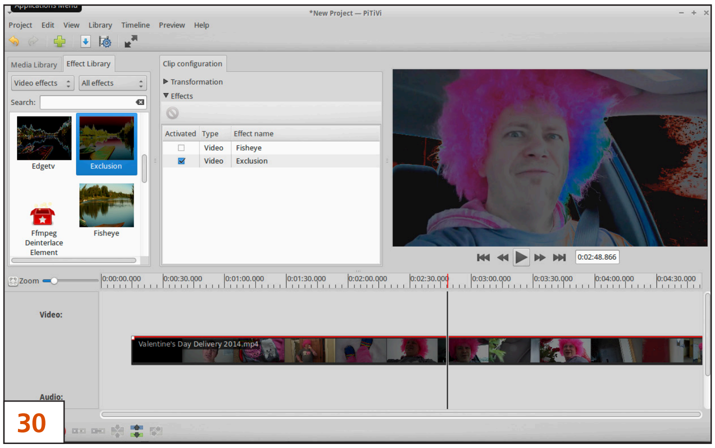
Type Machine automatically saves everything you type,

so you don't have to worry about accidentally erasing something important,

or an app crashing and losing everything you've typed.

or an app crashing and losing everyth

22



30

50

LINUX JOURNAL (ISSN 1075-3583) is published monthly by Belltown Media, Inc., 2121 Sage Road, Ste. 395, 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 • Adam Monsen

Publisher	Carlie Fairchild publisher@linuxjournal.com
Director of Sales	John Grogan john@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

Editorial Advisory Panel
Brad Abram Baillio • Nick Baronian • Hari Boukis • Steve Case
Kalyana Krishna Chadalavada • Brian Conner • Caleb S. Cullen • Keir Davis
Michael Eager • Nick Faltys • Dennis Franklin Frey • Alicia Gibb
Victor Gregorio • Philip Jacob • Jay Kruizenga • David A. Lane
Steve Marquez • Dave McAllister • Carson McDonald • Craig Oda
Jeffrey D. Parent • Charnell Pugsley • Thomas Quinlan • Mike Roberts
Kristin Shoemaker • Chris D. Stark • Patrick Swartz • James Walker

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 980985, Houston, TX 77098 USA

SILICON MECHANICS

SERVERS • STORAGE • HPC

Are you considering software-defined storage?

zStax

ZFS Unified Storage

zStax StorCore ZFS Unified Storage from Silicon Mechanics is truly software defined storage.

From modest data storage needs to a multi-tiered production storage environment, the **zStax StorCore** ZFS unified storage appliances have the right mix of performance, capacity, and reliability to fit your needs.

April Case Study Features

INgrooves
MEDIA DELIVERED. MEDIA MARKETED.
New Storage Solution is Music to the Ears of Fast-Growing Digital Music Company

Click here to download all of our case studies.

zStax StorCore 64

zStax StorCore 104

Talk with an expert today: 866-352-1173
www.siliconmechanics.com/zstax

Current_Issue.tar.gz

SHAWN POWERS

Big Block Linux with a Four-Barrel Hemi

We often brag about how few resources Linux needs to operate: a Raspberry Pi or Beagle Board, in a Web browser (**http://bellard.org/jslinux**), or any number of other tiny places. Just because our beloved OS can run on a tiny scrap of hardware, however, certainly doesn't mean that's all it will do! In this issue, we talk about high-performance computing. Whether you're calculating trajectory corrections for a spacecraft millions of miles away or hashing transactions for the Bitcoin network, Linux is the perfect vehicle for all that computing power.

Reuven M. Lerner starts us out this month, this time with information on how to leverage geolocation information in your Web application. Whether you want to give your Web visitors a local weather forecast or just want to present them with location-appropriate options from your Web applications, geolocation is a powerful tool. Since the Internet is global, it's important to know where users are located. Reuven shows how to integrate geolocation awareness into your Web applications. Dave Taylor follows with the next in his series on *Zombie Dice*. It may feel like you're just making a cool game, but it's really just a ruse to help you learn something. (Well, it's a cool game too, but you really are learning!)

Kyle Rankin continues his series on Tails. When it comes to browsing the Web securely, you either can wear a tinfoil hat or look into something like Tails. The former won't help with security, but people at the coffee shop probably will leave you alone. Last month, Kyle explained how to install Tails, and this month, he describes using

VIDEO:
Shawn Powers runs through the latest issue.

it. I took a completely different look at Linux this month. Instead of discussing security, I talk about entertainment-ivity. I recently set up my XBMC devices to share a centralized MySQL database, and I found it a little more difficult than I expected. In my column, I walk you through the process, and also discuss my mobile entertainment system: Plex. I get lots of e-mail regarding my home media setup, so this month I tell all.

Then we get into the nitty gritty of this month's issue. David Brown gives some great processes for managing HPC clusters. High-performance computing usually means lots of computers, and managing those clusters can be overwhelming! David walks through a phased process to deal with rolling out an HPC environment. Adam Diaz follows with a great article on YARN. No, he's not teaching how to knit, but rather a new method to deal with resource management in Hadoop. Rather than continuing to stretch MapReduce beyond its abilities, YARN (Yet Another Resource Negotiator) attempts to take over the resource management features of MapReduce. If you're a Hadoop user, you need to read about YARN.

Linux containers continue to be the rage as the more efficient method for virtualization allows for far more dense computing. High-performance computing gets far more performance out of containers than out of traditional virtualized hardware. Rami Rosen discusses Linux containers and how the combination of lightweight virtualization and HPC will shape the future of cloud computing. And finally, Brian Trapp shows how to use MySQL's built in replication to help secure data integrity in your database implementations. My mantra is always "backup, backup, backup", but with the fast pace data changes in a database system, frequent backups aren't enough. With data replication, backups are still important, but the database administrator can sleep a little better between backup cycles.

This issue of *Linux Journal* has tons of HPC information, but if that's not your cup of penguin juice (eiw!), there's plenty of tech tips, product announcements and general Linux information to keep you entertained and educated. So whether you run Linux on your pocketwatch or need a bicycle to get from one end of your data center to the other, this issue is perfect for you. We hope you enjoy it as much as we enjoyed putting it together.■

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

LINUX JOURNAL
SPONSORED BY
New Relic.
Since 1994: The Original Magazine of the Linux Community
FEBRUARY 2014 | ISSUE 238 | www.linuxjournal.com
WEB DEVELOPMENT
Develop and Deploy Quickly with PaaS Cloud Services
Build a Blog with Django and MongoDB
Keep Your DNS Search History Private
+
GUEST EOF: GIRLS AND SOFTWARE
WATCH: ISSUE OVERVIEW
ADD SPLIT TESTING TO YOUR WEB APP
INTRO TO RACK HOSTING AND DEPLOYMENT
SECURITY GUIDELINES FOR WEB DEVELOPMENT

More Secure SSH Sessions

I read Federico Kereki's January 2014 article "More secure SSH connections", and the article provides some really good pointers. I would also like to add that small changes to the sshd_config file to ensure that potentially weak ciphers and message digests are excluded can harden the server even more. Additionally, the time out to keep connections alive can be decreased so inactive sessions are closed faster. An example config would be:

```
Ciphers aes256-ctr
ClientAliveInterval 300
ClientAliveCountMax 0
MACs hmac-sha2-512-96,hmac-sha2-256-96,
➥hmac-sha2-512,hmac-sha2-256
```

The above may not be suitable for older machines. To check the ciphers that are supported by your sshd implementation, you can check the man page for sshd_config and look for the "ciphers" and "MACs" sections:

> Ciphers
>
> Specifies the ciphers allowed for protocol version 2. Multiple ciphers must be comma-separated. The supported ciphers are "3des-cbc", "aes128-cbc", "aes192-cbc", "aes256-cbc", "aes128-ctr", "aes192-ctr", "aes256-ctr", "arcfour128", "arcfour256", "arcfour", "blowfish-cbc", and "cast128-cbc". The default is: aes128-ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128, aes128-cbc,3des-cbc,blowfish-cbc,cast128-cbc,aes192-cbc, aes256-cbc,arcfour.

These sections list the available ciphers as well as the default order in which they will be used. This is important, as by default your sshd might negotiate a weaker cipher because of the default order.

Explicitly specifying the order will eliminate that.

Finally, the above config example I have tested with Debian 7 Stable for SSH server and Windows PuTTY as a client. I have also set up a 4096-bit RSA private key, and the connection works like a charm. The PuTTY bit is important as it seems that many Windows SSH/SFTP programs rely on it for their connections.

Apologies if the above has already been covered in previous articles, and I am just repeating it.

Finally, in future articles, I would like to see if it is possible to secure SSH with one-time passwords (OTP), use dual-factor authentication or even google-authenticator.
—Martin

Federico Kereki replies: *Martin's comments are appropriate, and can add even more security, as he suggests, by disallowing methods considered less safe. The only point that must be carefully considered is whether all used clients will be able to connect after some methods are restricted. Although for up-to-date clients, the answer should be positive. As to including other ways of authentication, the idea is a good one, and I'll certainly look into it.*

Finally, for even more complex rules, which allow for finer-grained controls, I'd suggest looking into the "Match" conditional block rules, which allow specifying different setups depending on the user *who's connecting, his or her group, address and more.*

Choice of Ads Served

So here I was, trying to learn how to do floating-point arithmetic in bash. Google seemed to think I wanted to visit **http://www.linuxjournal.com/content/floating-point-math-bash**. Imagine my surprise when I discovered that an integral part of Linux education are ads that are GIFs of scantily-clad female video-game characters writhing about. While that isn't something that particularly interests me in my free time, it's certainly not something I need to deal with while I'm in my office, surrounded by co-workers and trying to get work done. Thank you so much for not only derailing me from getting coding done but also contributing to the culture that keeps the number of women entering software engineering steadily declining since it peaked in the 1980s.

Guess I'll be following the second and third search-engine hits instead.
—K. McNeelyshaw

From time to time, we run Google ad network ads to generate extra much-needed revenue from LinuxJournal.com, and since these are network ads, we do not contract with the individual advertisers. We don't know whose ads will display until they display. Further complicating matters is that different ads will display for different people because of their browsing history. Many advertisers use retargeting to increase the effectiveness of their ads. For example, ever shopped for say, cute dog beds on Amazon and then gotten ads for those same dog beds elsewhere while browsing? Since we have limited control over Google ads, we can't prevent certain ads from displaying, but we can try to eliminate them after the fact. So, if you are ever bothered by a specific ad campaign that is displayed via Google ads on LinuxJournal.com, please feel free to drop us an e-mail to let us know the name of the advertiser. A screenshot of the offending ad would be helpful as well.
—Katherine Druckman, LJ Webmistress

A Bundle of Tor

I was especially interested in Kyle Rankin's article "A Bundle of Tor" in the January 2014 issue. In the end, I was able to get the Tor browser up and running in a Linux environment (Lubuntu 13.10), but I must say that the images and explanation in the article were many times "out of date" and not according to the latest state of the Tor setup and browser views.

The text sometimes did not correspond to what I really saw, and the browser and app images in the article were also not really as they currently are in the latest version of the browser.

I would assume that, since it's a brand-new *LJ* issue, the content of the article would be closer to the "reality". In my humble opinion, it wasn't. And that's a pity, because for someone new to the Tor browser setup (like me), it can give some confusing impressions and might lead to a non-working (or not-working-as-it-should-work) Tor browser setup.
—Geert Vancompernolle

Kyle Rankin replies: *I'm sorry to hear that the images and commands weren't up to date enough when you tried everything out. Unfortunately, magazine articles require a bit of lead time compared to Web publishing, and the screenshots and commands were all based on the latest Tor Browser Bundle you could download at the time I wrote the piece in November 2013. That said, apart from filenames being somewhat different (so you may have to adapt the exact command I typed if you downloaded a slightly newer version), the general steps should still be the same. I am surprised that the browser itself looked that different, although perhaps that's just a desktop-theming issue (it could be worse, I normally have a green-on-black theme that I change to something more normal for screenshots). In any case, I do hope you were able to get Tor up and running.*

More Secure SSH Sessions, II

I loved Federico Kereki's article on SSH security in the January 2014 issue. I have to admit I was surprised not to see the OAUTH 2 Factor Verification PAM Module mentioned. I use this paired with Google Authenticator to access my machine via SSH. It asks for the verification code on my phone before it even offers a chance for my password. Since it changes every 30 seconds, it makes it even stronger than a password:

```
#Pam Module dependency
$ sudo apt-get install libpam-devel

#Authenticator PAM Module
$ sudo apt-get install libpam-google-authenticator

# Run as a user to create your hash key, DON'T LOSE IT!
$ google-authenticator
```

Now add the hash to your Google Authenticator app. Very *important*: you have to edit your sshd config in /etc/ssh/sshd_config and enable "ChallengeResponseAuthentication" and "UsePAM" by setting them to "yes". And voilà. Now restart your sshd server and give it a try.
—Brian

Federico Kereki replies: *This is a valid suggestion (thanks, Brian, for the tip!), and indeed more could be written on one-time password (OTP) systems or two-factor authentication (TFA) systems, such as Google Authenticator and similar ones. I'll keep it in mind for the future. I should point out that NTP configuration is important. Your server and your smartphone should be (reasonably) in sync as to time. Also, you probably should think about configuring sshd to enable other authentication methods, or else be certain never to forget or lose your smartphone if you need to connect to a server! Outside from that, Google Authenticator also can be used for extra security in Web sites to better protect your accounts.*

Dave Taylor's Spitfire Photo

Love the Spitfire picture in Dave Taylor's "Framing Images with ImageMagick" article in the February 2014 issue, but I am sure I won't be alone in pointing out they were made by Supermarine, not Submarine. *Per ardua ad astra*, as they say in the RAF.
—Roger Greenwood

Dave Taylor replies: *Bah, I blame autocorrect! Thanks for the clarification, mate!*

Susan Sons' EOF, I

If I were a better writer, this could have been written by me—well put, Susan. [See Susan Sons' Guest EOF "Girls and Software" in the February 2014 issue.] I'm probably of the same generation as Susan and have had similar experiences, and I'm sick and tired of the "female quota" thing. I'm hoping that initiatives like the Raspberry Pi will allow young girls to discover that there are far more exciting things to do than painting their faces. But it probably requires a major shift in the mindset of ordinary people (parents, grandparents, teachers) to encourage girls to experiment with non-girly pastimes. I keep hoping.
—Uschi

Susan Sons replies: *Thanks for reading (and for writing).*

There's so much we should be doing

to expose all kids *to more things so they can pick a "just right fit" set of interests for themselves. People spend too much energy on "girls must need this..." or "high IQ kids must need that..." and so on—less demographics, more experimentation.*

<Waits for the world to take a deep breath and let kids be kids.>

Susan Sons' EOF, II

Susan's editorial was so honest, so powerful and so transcendent, that she managed to ruthlessly expose one of the most disturbing trends in our modern world—an ever-increasing self-polarization of "wronged groups" that change their lives' narratives into ever more strident "calling out" of their presumed guilty offenders and histrionic elucidation of gross offenses against their group. Paraphrasing Susan, in reality, some people are jerks, some are not, and if the human race is to survive, hopefully the majority will strive to avoid earning the first categorization.

It has always been this way, it always will be, and ultimately the responsibility for which category one falls into lies with the individual, not the group into which they've been categorized by others. Thank you for publishing Susan's take on one instance of this polarizing movement, that of gender inequality in technology fields, and the movement's apparent need to demonize all non-victims as intentional or witless perpetrators of the injustice.

What an amazing person Susan comes across as in her article. I, as one member of the infinitely guilty "white male club" that I had the misfortune of being born into, regret not having had the pleasure of working and interacting directly with such a clear thinker and merit-focused technologist. I'm not in her league by any stretch of the imagination but applaud the mindset that urges humanity to re-focus on accomplishment rather than self-categorization into aggrieved groups and counterattacks against all perceived tormentors.

There are, have been, and will continue to be grievous injustices in the world. At times, it seems we've made no progress at all in addressing the ultimate source of such injustice, our weaknesses as individual human beings. By creating new polarizations in the 21st century that strive to rise to the same level of histrionics as those that wreaked such horrors in the 20th century (Fascism, Communism, etc.), we are only setting the stage for new manifestations of counter-productive revenge rather than focusing on improving ourselves and becoming better human beings, one individual at a time.

Kudos again for publishing one of the most forthright, heartfelt, and powerful articles I've read in any publication in quite a while.
—Chris Munger

Susan Sons replies: *Thanks, Chris, for your kind words.*

I can't fix it all, but I've found through the letters of readers like yourself that there are far more who just want to get things done rather than argue about demographics. That's heartening, because if we can just get more willing to say so in public, things should start getting back to sanity.

Susan Son's EOF, III

I am impressed with Susan Sons' EOF article "Girls and Software", and the plainness with which she applies common sense to issues that have been taken to absurd levels of complexity due to political correctness. It's very good and refreshing to see these opinions in writing, and now I have this article as a reference to respond with when having discussions on the matter. Unfortunately, in these types of arguments, it looks like only women are entitled to have certain opinions, which leads me to an important point Susan makes in her article: when certain people see that a woman expresses a different opinion, then they say she doesn't represent the female point of view for whatever reason they make up at the moment. To me, this only means that they do not understand how intellectual exchange of opinion works, basically, by listening to all opinions and counter-arguing with facts and reasoning, and not paying attention to the person.

Of course, I don't agree with everything in the article but I mostly concur. In any case, I find it an interesting topic to be brought up in our beloved journal. Thank you, Susan.
—Juan Olmedilla

Susan Sons replies: *Thanks for the kudos, Juan...hopefully we'll see a return to sanity over time.*

Susan Sons' EOF, IV

Susan Sons' EOF piece (February 2014) "Girls and Software" is one of the most refreshing articles I've read anywhere in I don't know how long; you need to keep this "girl" on board—not to disparage the "boys".

About six years ago, I relocated from Detroit to northeastern Michigan and have gotten to know quite a few folks, including a number of teen and post-teen girls. Something I've noticed is that young girls, 'round here anyway, just aren't like young girls when I was a young boy (which would be 60-some years ago). Lot of farm kids, lot of tough-times kids, willing and hard workers, smart, able, jump right in and make something happen, know how to drive a pickup, know how to drive a snowplow, know how to plow a field, cripes, know how to fix a busted pickup! Perfect candidates for technology, not a few actually doing technology. Overheard a recent conversation comparing the benefits of smartphone operating systems (Android versus Windows versus iPhone) in detail with references, between a 16-year-old, an 18-year-old and a 22-year-old. Serious talk with demonstrations to make a point. A little bit of kid, a lot of savvy (a lot more savvy than I'll ever be about smart or dumb phones). Seems like just the kind of folks Ms Sons is talking about.

I'd like to hear a lot more from her.
—Thomas Ronayne

Susan Sons replies: *Thanks for the compliment, Thomas.*

I've always thought it funny how much—despite the stereotype of the sheltered upper-middle-class suburban gamer becoming a techie—country life or growing up poor in any environment dovetails with hackerdom.

No one talks about the "maker movement" or hacking there, because it's a way of life. If you need something, you build it. If something breaks, you fix it. When resources are scarce, you improvise. For those who have the interest in computers, applying the same principles there is second nature.

I've thought, for a while now, that kids from these sorts of places are the ones that will save us...save us from the costs of sheltering kids, of keeping them away from work, of not letting them experiment or think for themselves. It matters little whether they become hackers, or apply those skills in some other area.

Susan Sons' EOF, V

I just got my February 2014 issue, and the first article I read was the EOF by Susan Sons. I agree with her that people should be judged on their merit, rather than their sex. This goes for anything, not just our little world of tech.

I grew up in a small town in Washington, and was part of a tech club in high school. There were no girls, not because they weren't welcome, but because of the very dichotomy that Susan discusses in her article. It's sad, because I knew a few girls who wanted to join in, but assumed that they were not welcome because of their sex.

We need more people with the hacker mindset, women included, because we have issues that need solutions, and hackers provide us with the mental resources we need to solve them. We should encourage intelligence and creativity in all of our children, regardless of their sex.
—Reed Brousseau

BirdCam

The first time I visited your site I saw a mourning dove in the snow. I absolutely couldn't believe it. I grew up in South Dakota and they are definitely "gone South" for the winter there. [See Shawn Powers' BirdCam articles in the October 2013 and February 2014 issues.]

Anyway, I love visiting the BirdCam page a couple times a day just to see what's going on in your backyard. Great ideas and good implementation.
—Bill

Thanks Bill! My family teases me about BirdTopia, but then just moments ago, my wife IM'd me from work saying that she couldn't see the bird bath due to snow, but a mourning dove kept popping its head up over the snowbank, and it made her laugh. The amount of joy I've gotten from the entire project, tech and building, has been incredible. Thanks again—Shawn Powers

Loved Kyle's DNS Article

Two questions: 1) Can we get more like this? I am interested in better security for my personal machine as well as my server on the Net. [See "Own Your DNS Data" by Kyle Rankin in the February 2014 issue.] 2) Regarding DNS specifically, is there a way to test and verify where my queries are going to?

Thanks for the great magazine!
—Shawn Freeman

Kyle Rankin replies: *I'm glad you liked the column. To answer your first question, I'm devoting at least the first part of this year to security and privacy issues in my column, so you can expect more security content in the coming months.*

With respect to where your DNS queries are going, the most definitive way to see where DNS traffic is going is to use a low-level tool like tcpdump or wireshark that can capture packets leaving your system. Since DNS communicates over port 53, you should be able to filter out the rest of your traffic and just view DNS packets. If you don't need a definitive answer, or aren't interested in something that low-level, you always can look at /etc/resolv.conf for a list of nameservers, and if it only lists 127.0.0.1 and you haven't set it to that yourself yet because you are running your own DNS server on the same machine, you may be using a tool like resolvconf on your system and may have to dig through the resolvconf configuration to track down what actual name servers you are using.

WRITE *LJ* A LETTER

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

PHOTO OF THE MONTH

Remember, send your Linux-related photos to ljeditor@linuxjournal.com!

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 980985, Houston, TX 77098 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

Recently, someone suggested rewriting Linux in **Perl** as a way to improve the design and make it more organized and uniform. In particular, the person said that Linux relied on the **big kernel lock** (BKL) longer than other free OSes like **FreeBSD**.

It's fun to speculate on who might have posted that. Clearly, he or she knows enough about Linux development to know what the big lock was, and that other OSes had it for less time, but the person is so full of bile against Linux that he or she would snipe that an interpreted language would be faster. Let's see...what big software company employs people to work on Linux because it hates Linux?

Anyway, **Theodore Ts'o** had an interesting, and actually relevant, response. He said, "Linux had the BKL longer because it has had SMP longer than its competitors. Linux got rid of the last of the BKL in mid-2012. As of 2013, FreeBSD, NetBSD and OpenBSD still have the giant lock (BSD's equivalent of the BKL) in some of their subsystems." And he added, "Linux has had much better scalability than the *BSD's for much of the past couple years, with SGI using Linux on systems with hundreds of processors, and with people using Linux on 32 and 64 processors systems for the past decade. In contrast, FreeBSD was boasting in 2013 of improving its 16 processor scalability."

He also remarked that his favorite was when someone "suggested porting BSD 4.3 to Emacs LISP, so that you could run your entire system under GNU Emacs."

It's fun to think about all the different trolls that have appeared through the years. But actually, there are some pretty valid reasons for proprietary OS companies to be bitter. Linux may not dominate the desktop, but it still powers millions upon millions of servers that help make the Internet what it is. And, those servers represent massive amounts of lost revenue for proprietary OS companies.

Victor Porton wanted to improve the security of the **SELinux sandbox**, and started a discussion on how to do that. The problem was that hostile code could break out of the sandbox too easily, primarily by spawning child processes. He wanted the sandbox to keep better track of child processes by adding an ID to each process. In his vision, the ID could not be abandoned and would allow the sandbox to reap all child processes after the parent terminated.

There was a lively discussion. Someone suggested that **cgroups** (Linux Control Groups) could use their resource-limiting features to constrain processes within the sandbox. But **Andy Lutomirski** objected, saying that cgroups was already a horrible failure, and was getting worse, not better.

Andy suggested using the spanking-new **subreaper** to accomplish his goal. The subreaper, introduced in Linux 3.4, tracked process ancestry and delivered child-process exit status to the nearest living ancestor of that process, even if the child process' immediate parent already had terminated.

According to Andy, the subreaper could implement a new command to kill all descendants of a given process. This would ensure that no process could slide undetected out of the sandbox's grasp.

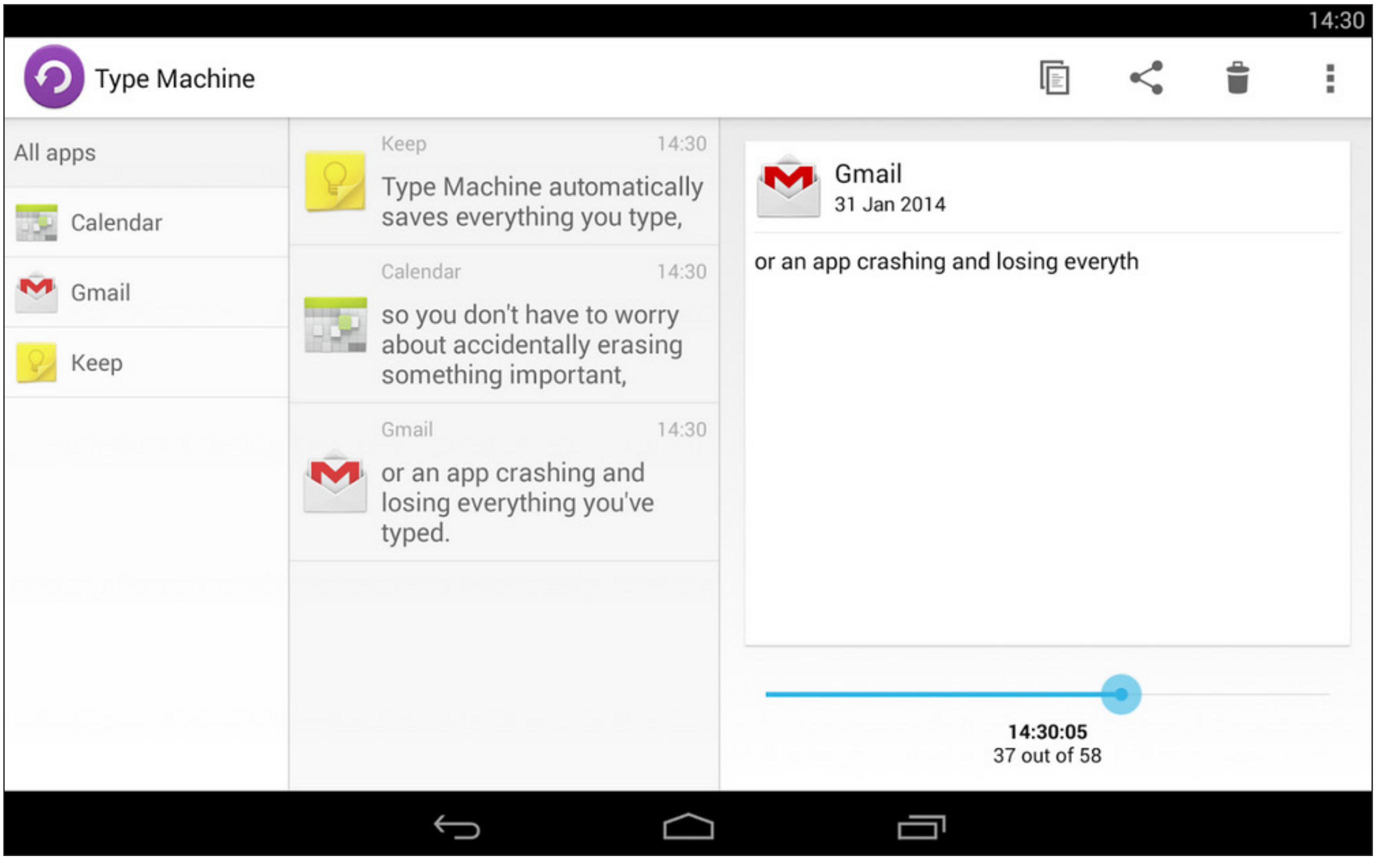
Joshua Brindle also suggested using a **seccomp filter** for sandboxed processes. Seccomp filters could restrict the system calls a given process could use. Joshua suggested that processes in the sandbox be restricted from using any system calls that might allow them to escape. Unfortunately, this wouldn't work for Victor's particular use case, which required the sandbox to span a network successfully.

Ultimately, Victor found no more appealing solution than the cgroups idea, and he offered a thorough description of how he wanted to proceed in that direction. The discussion ended there, but probably at least one of the various approaches to the SELinux sandbox will result in improved security.

There's been some question regarding the status of the **2.6.34 stable tree**. Apparently, **Aaro Koskinen** noticed that it hadn't been updated in a while, and **H. Peter Anvin** confirmed that it had been more than a year since the last update—during which time the 2.6.32 stable series had seen a new release. Peter said, "I'm worrying if people think that security patches are still being backported if in fact they aren't."

Paul Gortmaker, the 2.6.34.x tree maintainer, replied that there was another release in the works that would come out within a couple weeks, "with a focus on just clear CVE like fixes and hence a relatively smaller queue size (i.e., nothing like 200 patches etc.)". But he added that the tree would shortly reach its end of life and would not receive any more updates beyond that point.—**ZACK BROWN**

Android Candy: Control-Z for Your Phone!



Keep 14:30
Type Machine automatically saves everything you type,

Calendar 14:30
so you don't have to worry about accidentally erasing something important,

Gmail 14:30
or an app crashing and losing everything you've typed.

Gmail
31 Jan 2014

or an app crashing and losing everyth

14:30:05
37 out of 58

(Screenshot from the Google Play Store)

I never have a Twitter app crash in the middle of a Tweet. That wouldn't be too terrible to deal with. No, for me, it seems my e-mail application decides to crash after I've spent 20 minutes thumbing out a reply while sitting in a crowded airport. If you've ever lost a love letter, term paper, shopping list or world-class Facebook post, Type Machine is the perfect app for you.

It costs $1.99 in the Google Play Store, and automatically keeps track of the last text typed in every native Android application.

It has some great features that satisfy even the most privacy-concerned individuals:

- No unnecessary permissions.
- Supports a PIN number to lock typing history.
- Apps can be blacklisted so no input is recorded.
- History is pruned automatically.
- Password fields are never recorded.

The best part about Type Machine is that it works automatically in the background, and you never need to think about it—until you do.

I'll admit, the thought of installing a keystroke logger on my own device was a little creepy at first. I've never read the "this app requires these permissions" screen more carefully than when installing Type Machine. That said, I've had it only a couple days, and I've already used it to retrieve a Twitter update that got lost amid a program crash. If you have a particularly crash-prone phone, or if you just prefer not to risk the possibility of a lost e-mail, check out Type Machine in the Google Play store: **https://play.google.com/store/apps/details?id=fi.rojekti.typemachine**.

—SHAWN POWERS

They Said It

There's more to life than books, you know. But not much more.
—Morrissey

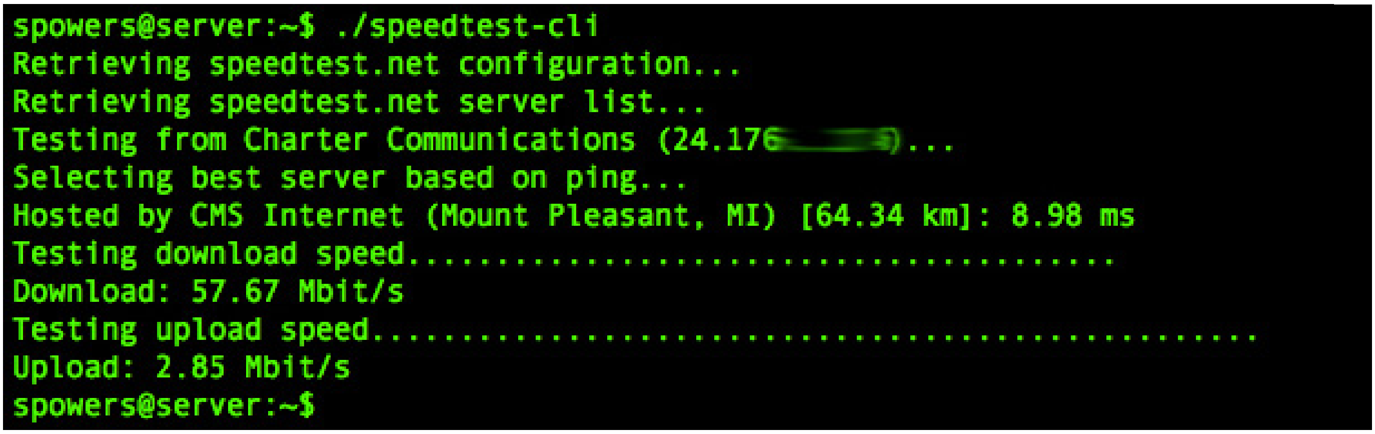
Don't cry because it's over, smile because it happened.
—Dr. Seuss

Be yourself; everyone else is already taken.
—Oscar Wilde

Two things are infinite: the universe and human stupidity; and I'm not sure about the universe.
—Albert Einstein

Most people work just hard enough not to get fired and get paid just enough money not to quit.
—George Carlin

Speed Test for Nerds



```
spowers@server:~$ ./speedtest-cli
Retrieving speedtest.net configuration...
Retrieving speedtest.net server list...
Testing from Charter Communications (24.176[redacted])...
Selecting best server based on ping...
Hosted by CMS Internet (Mount Pleasant, MI) [64.34 km]: 8.98 ms
Testing download speed........................................
Download: 57.67 Mbit/s
Testing upload speed..................................................
Upload: 2.85 Mbit/s
spowers@server:~$
```

Most people with Internet access in their houses have visited a speed-test Web site to make sure they're getting somewhere close to the speed they're overpaying for. I'm paying more than $100 a month for my business-class connection from Charter, so on a regular basis, I make sure I'm getting the advertised speed. (I seldom get the advertised speed, but the margin of error is acceptable. I guess.)

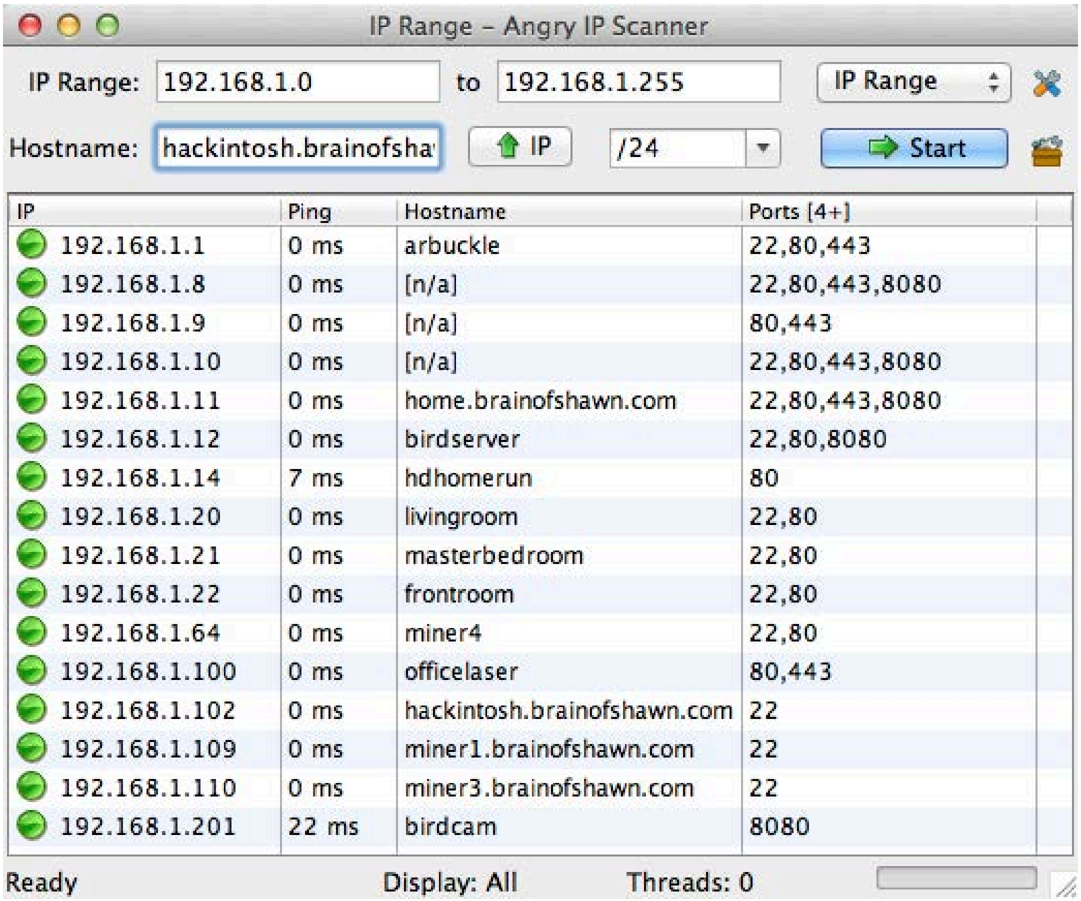
One of the frustrations with Internet speed tests is that most of them require Adobe Flash to work. Even those sites that don't require Flash do require a rather robust (and frivolous) GUI that I find annoying at best. If you're anything like me, you'd like a simple command-line tool that gives you your speed. If you're truly like me, that last sentence just sparked notions of automated scripts e-mailing results via timed cron jobs at different times during the day. Welcome to the club; we're all nerds here.

Thankfully, my friend Charlie K. (I won't use his last name, because I didn't ask him if I could) posted a link on Google Plus to the speedtest-cli program. The project is on GitHub at **https://github.com/sivel/speedtest-cli**, and to get the Python-based program, simply do this:

```
# wget -O speedtest-cli \
  https://raw.github.com/sivel/speedtest-cli/master/speedtest_cli.py
# chmod +x speedtest-cli
```

Then execute the script `./speedtest-cli` to get your results. There are advanced options as well, but a simple execution of the script will provide your speed results. You can see the results of my supposed 80/5 business connection in the screenshot.—**SHAWN POWERS**

Non-Linux FOSS: Angry IP



IP Range: 192.168.1.0 to 192.168.1.255 IP Range

Hostname: hackintosh.brainofsha /24

IP	Ping	Hostname	Ports [4+]
192.168.1.1	0 ms	arbuckle	22,80,443
192.168.1.8	0 ms	[n/a]	22,80,443,8080
192.168.1.9	0 ms	[n/a]	80,443
192.168.1.10	0 ms	[n/a]	22,80,443,8080
192.168.1.11	0 ms	home.brainofshawn.com	22,80,443,8080
192.168.1.12	0 ms	birdserver	22,80,8080
192.168.1.14	7 ms	hdhomerun	80
192.168.1.20	0 ms	livingroom	22,80
192.168.1.21	0 ms	masterbedroom	22,80
192.168.1.22	0 ms	frontroom	22,80
192.168.1.64	0 ms	miner4	22,80
192.168.1.100	0 ms	officelaser	80,443
192.168.1.102	0 ms	hackintosh.brainofshawn.com	22
192.168.1.109	0 ms	miner1.brainofshawn.com	22
192.168.1.110	0 ms	miner3.brainofshawn.com	22
192.168.1.201	22 ms	birdcam	8080

Ready Display: All Threads: 0

The de facto standard for port scanning always has been the venerable Nmap program. The command-line tool is indeed very powerful, but I've only ever seen it work with Linux, and every time I use it, I need to read the man page to figure out the command flags.

Windows users have been able to use the "Angry IP Scanner" tool for quite some time, and recently, the program (since version 3) has become truly cross-platform. If you need to scan for open ports on a specific host or on an entire network, the Angry IP Scanner (or just ipscan) tool is fast, robust and, of course, open source.

Grab a copy of this awesome little FOSS tool from its Web site at **http://www.angryip.org** or directly from SourceForge at **http://ipscan.sf.net**. Just remember, port scanning is one of those skills that can be used for good or evil—be sure you're wearing your white hat!—**SHAWN POWERS**

Numerical Python

For the past few months, I've been covering different software packages for scientific computations. For my next several articles, I'm going to be focusing on using Python to come up with your own algorithms for your scientific problems. Python seems to be completely taking over the scientific communities for developing new code, so it is a good idea to have a solid working knowledge of the fundamentals so you can build solutions to your own problems.

In this article, I start with NumPY (**http://www.numpy.org**). My next article will cover SciPy, and then I'll look at some of the more complicated modules available in the following article.

So, let's start with the root module from which almost all other scientific modules are built, NumPY. Out of the box, Python supports real numbers and integers. You also can create complicated data structures with lists, sets and so on. This makes it very easy to write algorithms to solve scientific problems. But, just diving in naively, without paying attention to what is happening under the hood, will lead to inefficient code. This is true with all programming languages, not just Python. Most scientific code needs to squeeze every last available cycle out of the hardware. One of the things to remember about Python is that it is a dynamic language where almost all functions and operators are polymorphic. This means that Python doesn't really know what needs to be done, at a hardware level, until it hits that operation. Unfortunately, this rules out any optimizations that can be made by rearranging operations to take advantage of how they are stored in memory and cache.

One property of Python that causes a much larger problem is polymorphism. In this case, Python needs to check the operands of any operator or function to see what type it is, decide whether this particular operand or function can handle these data types, then use the correct form of the operand or function to do the actual operation. In most cases, this is not really an issue because modern computers have become so fast. But in many scientific algorithms, you end up applying the same operations to thousands, or millions, of data points. A simple example is just taking the square of the first

100,000 numbers:

```
import time
a = range(100000)
c = []
starttime = time.clock()
for b in a:
   c.append(b*b)
endtime = time.clock()
print "Total time for loop: ", (endtime - starttime)
```

This little program uses the `range` function to create a list of the first 100,000 integers. You need to import the `time` module to get access to the system clock to do timing measurements. Running this on my desktop takes approximately 0.037775 seconds (remember always to make several measurements, and take the average). It takes this much time because for every iteration of the loop, Python needs to check the type of the b variable before trying to use the multiplication operator. What can you do if you have even more complicated algorithms?

This is where NumPY comes in. The key element that NumPY introduces is an N-dimensional array object. The great flexibility of Python lists, allowing all sorts of different types of elements, comes at a computational cost. NumPY arrays deal with this cost by introducing restrictions. Arrays can be multidimensional, and they must all be the same data type. Once this is done, you can start to take some shortcuts by taking advantage of the fact that you already know what the type of the elements is. It adds extra overloading functions for the common operators and functions to help optimize uses of arrays.

All of the normal arithmetic operators work on NumPY arrays in an element-wise fashion. So, to get the squares of the elements in an array, it would be as simple as `array1 * array1`.

NumPY also uses external standard, optimized libraries written in C or FORTRAN to handle many of the actual manipulations on these array data types. This is handled by libraries like BLAS or lapack. Python simply does an initial check of each of the arrays in question, then hands them as a single object to the external library. The external library does all of the hard work, then hands back a single object containing the result. This removes the need for Python to check each element when using the loop notation above. Using NumPY, the earlier example becomes:

```
import numpy
import time
a = numpy.arange(1000000)
starttime = time.clock()
```

```
c = a * a
endtime = time.clock()
print "Total time used: ", (endtime - starttime)
```

Running this toy code results in an average run time of 0.011167 seconds for this squaring operation. So the time is cut by one third, and the readability of the code is simplified by getting rid of the loop construct.

I've dealt only with one-dimensional arrays so far, but NumPY supports multidimensional arrays just as easily. If you want to define a two-dimensional array, or a matrix, you can set up a small one with something like this:

```
a = numpy.array([[1,2,3,4], [2,3,4,5]])
```

Basically, you are creating a list of lists, where each of the sub-lists is each of the rows of your matrix. This will work only if each of the sub-lists is the same size—that is, each of the rows has the same number of columns. You can get the total number of elements in this matrix, with the property `a.size`. The dimensions of the matrix are stored in the property `a.shape`. In this case, the size is 8, and the shape is (2, 4), or two rows and four columns. What shape did the array in the earlier example have? If you go ahead and check, you should see that the shape is (1000000). The other key properties of these arrays are:

- ndim: the number of dimensions.
- dtype: the data type of the elements.
- itemsize: the size in bytes of each element.
- data: the buffer that stores the actual data.

You also can reshape arrays. So if you wanted to take the earlier example of the first 100,000 integers and turn it into a three-dimensional array, you could do something like this:

```
old_array = numpy.arange(100000)
new_array = old_array.reshape(10,100,100)
```

This will give you a new 3-D array laid out into a 10x100x100 element cube.

Now, let's look at some of the other functions available to apply to arrays. If you remember from earlier, all of the standard arithmetic operations are overloaded to operate on arrays one element at a time. But, most matrix programming languages use the multiplication element to mean matrix multiplication. This is something to keep in mind when you start using Python. To get a true matrix multiplication, you need to use the `dot()` function. If you have two matrices, A and B, you can multiply them with `numpy.dot(A, B)`.

Many of the standard mathematical functions, like cosine, sine, square root and so on, are provided by NumPY and are called universal functions. Just like with the arithmetic operators, these universal functions are applied element-wise across the array. In science, several common functions are used. You can get the transpose of a matrix with the object function `a.transpose()`. If you need to get the inverse of a matrix, there is the module function `inv()`, so you would execute `numpy.inv(a)`. The trace is also a module function, given by `numpy.trace(a)`.

Even more complicated functions are available. You can solve systems of equations with NumPY. If you have a matrix of coefficients given by a, and a vector of numbers representing the right-hand side of your equations given by y, you can solve this system with `numpy.solve(a,y)`. In many situations, you may be interested in finding the eigenvalues and eigenfunctions of a given system. If so, you can use `numpy.eig(array1)` to get those values.

The last thing I want to look at here is a class that provides even more shortcuts, at the cost of more restrictions. Matrices (2-D arrays) are so prevalent that NumPY provides a special class to optimize operations using them as much as possible. To create a new matrix, say a 2x2 matrix, you would write:

```
A = numpy.matrix('1.0, 2.0; 3.0, 4.0')
```

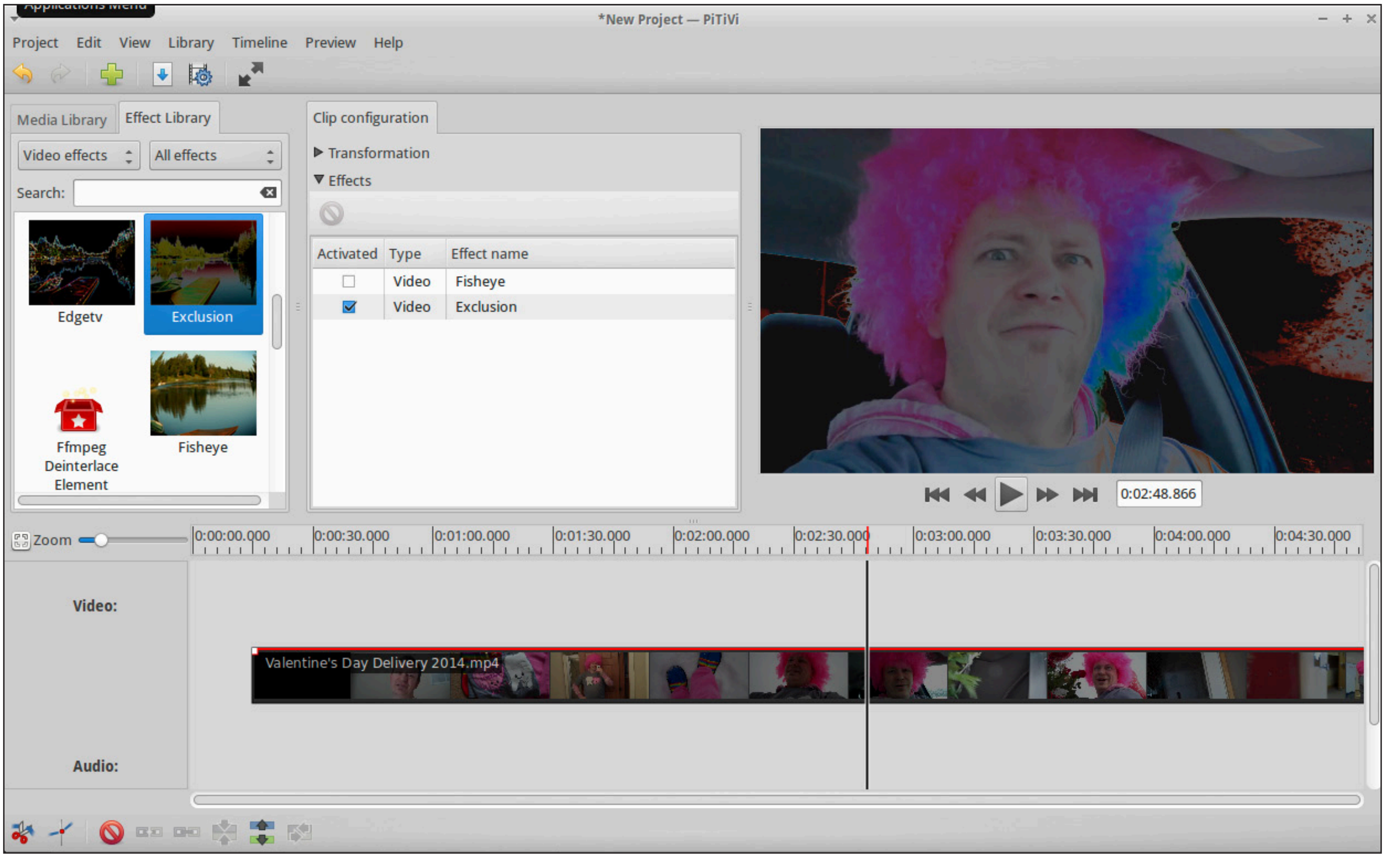
Now, you can get the transpose with just `A.T`. Similarly, the inverse is found with `A.I`. The multiplication operation will do actual matrix multiplication when you use matrix objects. So, given two matrix objects A and B, you can do matrix multiplication with `A*B`. The solve function still works as expected on systems of equations that are defined using matrix objects. Lots of tips and tricks available on the NumPY Web site, which is well worth a look, especially as you start out.

This short introduction should get you started in thinking of Python as a viable possibility in "real" numerical computations. The NumPY module provides a very strong foundation to build up complex scientific workflows. Next month, I'll look at one of the available modules, SciPY. Until then, play with all of the raw number-crunching possibilities provided by NumPY.

—JOEY BERNARD

Pro Video Editing with Pitivi

LINUX JOURNAL EDITORS' CHOICE



Activated	Type	Effect name
☐	Video	Fisheye
☑	Video	Exclusion

Several decent video editors are available on the Linux platform. Kdenlive, OpenShot, Cinelerra and Pitivi are those that come to mind as "big players" in an admittedly small market. I've used them all through the years, with varying levels of success. A frustration of mine is that invariably, I end up using a proprietary video editing suite like iMovie or Final Cut Pro when I have to do a larger project. As an open-source enthusiast, that doesn't settle well with me.

Although I'm honestly not sure Pitivi is the best choice for Linux-based video editing, I truly can say that its current fundraising push is impressive. The "kickstarter" concept is old hat by now, but that doesn't mean a well-planned campaign isn't still a great idea. The Pitivi team is trying to raise enough money to put serious coding time into the program and get to the 1.0 release. That's only the first step of the journey, however, because after the solid 1.0 foundation is complete, future features will be added according to contribution and user-base voting.

I'm confident to say that Pitivi is currently a great choice for video editing on Linux. If the fundraising campaign works out well, it soon may be the clear leader in stability and functionality. Thanks to the combination of an incredible product plus a game plan to get even better, Pitivi is this month's Editors' Choice. If you want to be a part of Pitivi's future, check out the fundraiser page at **http://fundraiser.pitivi.org**. If you want to test the program itself, you can download it today for your favorite distribution at **http://www.pitivi.org**.—**SHAWN POWERS**

REUVEN M. LERNER

Geolocation

Standardizing addresses, finding users and customizing content are just three ways geolocation can improve your Web applications.

There's an old saying in the real-estate business that the three most important things in a property are location, location and location. We can assume this is still true when it comes to real estate, but it also is increasingly true when it comes to Web applications. A number of my recent consulting projects have included, in one way or another, the need to work with addresses and locations of various sorts.

This shouldn't come as too much of a surprise, given the many ways that the Web is becoming the way we communicate, store information and work. It gives me a warm (if somewhat creepy) feeling when a site I go to wishes me a "good morning", because it knows it is now morning in my part of the world. It's useful when a mapping program starts off by displaying my current location as a default. And as the person running various applications, I like the fact that I can learn basic geographical information about my users—both so I can offer additional services while simultaneously receiving useful data.

Working with street addresses, location coordinates and the like falls under the umbrella of "geolocation". So in this article, I review a few of the technologies and options that use geolocation and offer some suggestions as to how you can include such features in your own Web applications.

Which Server?

The first thing to realize when it comes to geolocation is that you're almost certainly not going to be able to do it alone. Sure, given enormous amounts of time and money, you probably could figure out the locations and street addresses of most people in the world, but you're unlikely to do this. This means you're going to have to connect to one or more companies that owns and distributes mapping information via an API, such as Google, Bing (Microsoft) or similar.

There are free and open-source alternatives to commercial map providers, such as **http://freegeoip.net** and **http://www.openstreetmap.org**. However, the APIs of the commercial offerings are richer and seem to be better supported. Even some of the free services will require or expect that you have an API key, for which you need to register. This allows them to track how many requests you are making and to limit your usage unless you pay for a commercial tier. Although it is useful and nice to work with open-source tools, the remainder of this column assumes you are working with a commercial provider.

Note that some API libraries provide a single interface to multiple servers for both street addresses and IP addresses. For example, the Geocoder gem for Ruby (written and maintained by Alex Reisner) lets you choose from a number of different mapping providers and also from a number of IP address providers, defaulting to Google and freegeoip.net, respectively. You then can decide whether to use free or commercial services, or a mix of them, depending on your use case.

It's also important to remember that the accuracy is far from 100%. For example, I decided to look up an old address of mine from when I was living in Skokie, Illinois. I wrote a small Ruby program to do this:

```
require 'geocoder'
Geocoder.search('9120b niles center road skokie il')
```

Google, the default decoding system, almost immediately returned with a better-formatted version of the address, along with a great deal of other information. I was able to get the address out of the system:

```
Geocoder.search('9120b niles center road skokie
➥il')[0].formatted_address
    => "9120C Niles Center Road, Skokie, IL 60076, USA"
```

Now, the fact is that the B and C units in that particular townhouse are right next to one another. And it's likely that if I were to look on a map, or even send mail to one of those addresses, the difference would be obvious. But as you can see, the address returned from Google is not necessarily the right one.

One of the nice things about Google's API is that it includes a large number of locations around the world. For example, I can look up my current address:

```
Geocoder.search('14 migdal oz street modiin israel')
```

But in this case, I don't get an

address that matches mine, but rather the overall entry for my city of Modi'in. I actually don't even get back a single entry, but rather three, each of which represents Modi'in in a different way, with a slightly different spelling. The differences between the entries is most obvious if I ask for the coordinates from each of the three returned result objects:

```
Geocoder.search('14 migdal oz street modiin
 ➥israel').map {|a| a.geometry['location'] }
=> [
    [0] { "lat" => 31.90912, "lng" => 35.002462 },
    [1] { "lat" => 31.890267, "lng" => 35.010397 },
    [2] { "lat" => 31.893661, "lng" => 34.96079 }
]
```

For many purposes, these coordinates are all close enough. However, if you are creating an application that depends on exact precision, such as a GIS navigation application, you're likely going to need to compare different services and perhaps even perform multiple queries, taking the result that most closely matches the location of interest.

Addresses and Coordinates

You now have seen several examples of how you easily can perform geocoding with the Geocoder Ruby gem. Given an address, you can invoke the "search" class method on the Geocoder object, getting an array of Geocoder result objects, containing various pieces of information about the resulting address. Even if there is only a single result, you will receive an array. And, the Google API tries hard to match something. It returned a result for "1 Main Street, Fredonia", but returned an empty array when I entered "1 zzz street, yyy qqq".

The result object contains a great deal of information. If I am interested in a standardized version of the address, I can invoke the "address_components" method on the result object, which will return an array of hashes containing the street number, street name, village name and so forth. This portion of the result contains more information than you need to address an envelope in the United States—for example, it includes the county and city name, as well as the state and postal code. You can grab these pieces of information separately or invoke methods that pull them together. I can use the "formatted_address" method (shown above) to get a complete address or the "street_address" method to get just the most important parts.

Several of the applications I've written for clients during the past few years have used geocoding APIs to standardize the addresses, ensuring that they have an "official" address that meets US specifications. This also helps avoid misspellings and other errors that can cause trouble down the road. Thus, even when a user enters his or her own address, we run it through a geocoding facility and store the result of this search. (It's probably a good idea to store the originally entered address as well.)

Instead of (or in addition to) an address, you'll often want to get the coordinates, including longitude and latitude. Because coordinates give an exact location on the globe, you can use them in a variety of places that aren't tied to individual addresses, like mapping software or GIS databases (such as PostGIS, a GIS extension to PostgreSQL). If I have the coordinates of a particular place, I then can draw that on a map with a great deal of accuracy. Two of my clients in recent years have asked that I hide users' addresses when displayed on maps for privacy reasons. Making up addresses (for example, changing "123 Main Street" to "456 Main Street") is almost guaranteed to cause trouble and failure, but changing the coordinates by a small random factor has worked quite well.

Geocoding IP Addresses

Although some of my geocoding work has involved taking addresses from the user's input, much of it has been just the opposite—trying to figure out the user's location and then doing something with that information. In other words, I would like to take the user's IP address and use that to pinpoint the user's location.

The first thing to realize is that the need for such things has been reduced, to some degree at least, by the HTML5 API for geolocation. That API, implemented on the client and in JavaScript, allows an application to ask the browser to report its current location. (The standard requires that a browser ask the user before sending location information.) You then can use the information inside a Web page using JavaScript or invoke an Ajax call to send that information to the server, where it can be parsed and used.

On one recent project, I wasn't interested in bothering the user with geolocation information or in using that information within a Web application. Rather, I wanted to review an application's logs and summarize the countries from which visitors had come. To do this, I needed to review each logfile entry and then look for each IP address, determining its country.

Now, note that this kind of information can be grossly inaccurate. For example, I'm currently writing this in my local public library in Modi'in, Israel, and my IP address is being reported as 81.218.200.112 to the outside world. I can ask Geocoder to tell me where it thinks I am:

```
result = Geocoder.search('81.218.200.112')
```

And unfortunately, it has no idea, other than the fact that I'm in Israel:

```
result[0].country
  => "Israel"
result[0].city
  => ""
```

According to **http://www.iplocation.net**, which offers IP location to individual visitors, it thinks I'm in Petach Tikva—a fine city, but a 40-minute drive from where I'm sitting. That's because the geolocation is looking for the telecom facility, or the provider, rather than the specific location where I'm located.

So you always should take IP geolocation with a healthy dose of skepticism. Moreover, plenty of IP addresses are not in geolocation databases. Others are tied to companies or to services (for example, Google's searchbot) that will come to your site and make requests, but for which there is no location. And then there are the visitors who come to your site via cellular phones and services, which often can be national in scope and thus fail to provide an accurate reading.

That said, if you are interested in finding out general information about your users—their countries of origin and time zones—then IP location can work quite well. As you can see, the Geocoder gem lets you use the same class method, "search", to request information about an IP address. It figures out whether you're entering an IP address, coordinates or a street address, and handles it accordingly. For one recent project, I was able to provide interesting information and analytics about the countries from which people came, simply by running the IP addresses through an IP geolocation library.

As a general rule, you never should perform such an action in real time, when the user is coming to your site. You are much better off running a background task or an hourly cron job. As you collect and store the IP location information, you almost certainly should store it in a database, or at least cache it, to avoid too many requests to the geolocation service.

If you do end up using Geocoder with Rails, you get a "location" method that you can invoke on the "request" object, allowing you to get the user's information automatically, via IP address. I have not tested this to see if calling the "location" method significantly adds to the response time, or if it is somehow handled in a separate thread, or by turning to a cached copy of the data on the local server, but it would be wise to check the performance hit before putting it into production.

Configuration

I have barely mentioned configuration so far, because I've found that Geocoder works so well out of the box. That said, there have been times when I have wanted or needed to reconfigure it. Fortunately, configuration is quite simple and straightforward, and is accomplished by invoking the Geocoder.configure class method.

For example, while working on this article at the library, I found that the Wi-Fi connection was quite slow—so much so that even simple API calls were timing out. I was pleasantly surprised to find that the Geocoder gem was smart enough to realize the problem was a timeout and suggest that I might want to avoid the timeout by invoking Geocoder.configure. Now, that's the sort of error message I would like to see more often! So, I invoked:

```
Geocoder.configure(:timeout => 1000)
```

And sure enough, my future calls worked just fine, even if they took a while to execute.

You always can get the current configuration settings by invoking Geocoder.location without any options. This returns a hash with all of the name-value pairs associated with the configuration system.

First, if you want to use a different geocoding API than the default of Google, you can do so by changing the "lookup" parameter in the configuration system:

```
Geocoder.configure(lookup: :nominatim)
```

Now, the results from a search will not be an array of Geocoder::Result::Google (which is what you received before), but rather Geocoder::Result::Nominatim. Each result object has a different set of methods and attributes, which means you cannot simply swap one

API for another. The methods and data available reflect the information received from the geocoding API as best as possible.

Summary

Geolocation is far from perfectly accurate. However, this lack of precision doesn't mean that you should avoid using it in your applications. Whether you want to give localized greetings to your users, standardize addresses or create summaries and reports of who has been accessing your application, geocoding is a technique you likely will find useful for many of your applications. As easy as the commercial and free APIs may be to use, the existence of such open-source libraries like Geocoder makes it even easier.■

Reuven M. Lerner, a longtime Web developer, consultant and trainer, is completing his PhD in learning sciences at Northwestern University. You can learn about his on-line programming courses, subscribe to his newsletter or contact him at http://lerner.co.il.

Send comments or feedback via http://www.linuxjournal.com/contact or to ljeditor@linuxjournal.com.

Resources

The home page for the Ruby Geocoder gem is at **https://github.com/alexreisner/geocoder**. The gem is still under active development, and the GitHub page includes a great deal of documentation and examples.

The open-source and free geocoding site **http://freegeoip.net** and the **http://www.openstreetmap.org** application (which is building a map of the world that anyone can use) are both worth visiting, and perhaps even incorporating into your application.

If you are a Python user, you should look at the pygeocoder package, available at PyPI (**https://pypi.python.org/pypi**), which does similar things to the Geocoder Ruby gem discussed in this column.

Finally, if you are interested in storing the results of geolocation in a database, you should look into PostGIS (**http://postgis.org**), an extension to the PostgreSQL database that includes GIS. I am still taking my first steps with PostGIS, but the book *PostGIS in Action* written by Regina Obe and Leo Hsu, and published by Manning, provides a useful introduction and tutorial.

The Best SharePoint Training in the World!

Choose from more than 80 classes and tutorials!

"This is the place to be if you want to know about SharePoint. Not just technical, but business-related information as well."

—David Pileggi, Lead Consultant, Portal Solutions

"SPTechCon has lots of excellent real-world knowledge, bonding opportunities and the ability to interface with leaders in the space."

—David Miller, Director, IT, Cubic Transportation Systems

SPTechCon

The SharePoint Technology Conference

April 22-25, 2014

San Francisco Hilton

Bolster your career by becoming a SharePoint Master!

- Learn from SharePoint experts, including dozens of SharePoint MVPs and Certified SharePoint Professionals
- Master document management
- Study SharePoint governance
- Find out about SharePoint 2013
- Learn how to create applications for SharePoint that solve real business problems
- Exchange SharePoint tips and tricks with colleagues
- Test-drive SharePoint solutions in the Exhibit Hall

If you or your team needs Microsoft SharePoint training, come to SPTechCon San Francisco!

Register Early and SAVE! → www.sptechcon.com

A BZ Media Event @SPTechCon

Central Contractor Registration

SPTechCon™ is a trademark of BZ Media LLC. SharePoint® is a registered trademark of Microsoft.

DAVE TAYLOR

Accumulating Brains in *Zombie Dice*

Dave continues to build a *Zombie Dice* game, and in this installment, he deals with runners and counting up how many brains you've harvested while trying to avoid those pesky gunshots.

For the record, this is my 100th column for *Linux Journal*. One hundred columns. That's more than eight years of me writing about shell scripts and you, dear reader, reading my articles. It seems like a good partnership, and I hope we can stick together into the next decade, which is only a dozen or two columns down the road. That first column was titled "Getting Started with Redirection", and it explored the differences between >, >>, < and << in shell scripts. I think we've come a long way since then!

And on this occasion of commemorating my 100th column, I again invite you to send me e-mail with your ideas for interesting column topics. If you can dream it up, and it's not insanely complicated, you can write it as a shell script. Enough of that mushy stuff though—let's get back to work!

In my last article, I started writing a game that simulated some elements of the Jackson Games dice game *Zombie Dice*. As homework, you might want to pick up the game at Target or Walmart next time you're there, or get it at **http://amazon.com**. It's fun, easy and great with kids (or with my kids, at least, who can appreciate a few zombies as a source of entertainment).

The game revolves around rolling dice and seeking to maximize one accumulation of items (brains) while avoiding getting shot at too many times (gunshots). There are three

The entire purpose of this segment is to set the “color” variable correctly. Who knew that’d require so much code?

different kinds of dice with varying levels of challenges—green, yellow and red—and various nuances about how many you can roll and how. To learn more, check out last month’s column or look at the instructions that came with your canister of *Zombie Dice*.

The script, when last I hacked it, could simulate a roll while also randomly choosing between the different dice:

```
$ ./zdice.sh
    rolled green die: brain
    rolled red die: runner
    rolled red die: brain
```

To start, let’s add the smarts in the script to know what a “runner” is. “Dude, what is a runner?” I can hear you ask. Okay, okay, a runner is essentially an ambiguous roll, and if you opt to roll your dice again, you roll that specific die, whether it’s green, yellow or red. The brains and gunshots you pull aside and accumulate, by contrast. So on the above roll, you’d put the two brain dice aside, re-roll the red “runner” and randomly pick two of the remaining dice to make the three required for another roll. In code, you can simulate this by assigning specific values to the `diceroll[]` array. If the specific roll status is set to zero, it’s a new roll of a new die. If not, it’s the numeric value associated with the die color and produces a re-roll:

```
if [ ${diceroll[$rollcount]} -eq 0 ] ; then
  pick_color
else
  echo "  dice $rollcount was a runner last time, \
      rerolling the same color die again:"
  color=${diceroll[$rollcount]}
  diceroll[$rollcount]=0      # reset for next roll
fi
```

The entire purpose of this segment is to set the “color” variable correctly. Who knew that’d require so much code?

So that you have some status output though, it’s the kind of information that might well be removed once you’ve thoroughly tested the production code.

You'll then have this snippet:

```
roll_die $color
echo "    rolled ${colorname[$color]} die: ${nameof[$roll]}"
```

This is an example of debug output. I've written about debugging shell scripts in previous columns if you want to check the archives to learn more about this essential element of script development.

There's not much work in the add_score subroutine, as you'll see shortly, but it fits in here, after the dice rolls. Then there's additional code below to show how you're doing with this roll, and this will be where the challenge of accumulating brains and gunshot scores will happen, along with the test of whether you've had too many gunshots and have lost.

One more step, the mirror of the earlier code: if the roll (variable $roll, set in subroutine roll_die) is a runner, then the diceroll[] value needs to be set appropriately so it can be differentiated from a non-runner roll:

```
if [ $roll -eq $RUNNER ] ; then
  diceroll[$rollcount]=$color;
fi
```

This is all neatly wrapped up in a for loop that gives three rolls with the simple expedient of:

```
for roll count in 1 2 3
do
   all the code shown above goes here
done
```

Let's have a closer look at the add_score subroutine, then the loop that lets you decide after each three-dice roll whether you want to stop or continue. Remember, your goal is to get the most brains without dying of three gunshot wounds:

```
function add_score
{
  # Add the current roll to the score so far
  # Only need to score brains and gunshots

  case $roll in
    $BRAIN ) brains=$(( $brains + 1 ))  ;;
    $SHOT  )  shots=$(( $shots + 1 ))  ;;
  esac
}
```

It's short and sweet, actually. Notice that as with many case statements, there actually are three possible values ($RUNNER is the third possible value), but only two are addressed. That's fine; the third value just drops through. The trick with good coding, of course, is to know that's going to happen.

If you haven't seen it before, the $(( )) notation is a convenience for invoking an equation-solving part of the Bash shell.

If you haven't seen it before, the `$(( ))` notation is a convenience for invoking an equation-solving part of the Bash shell. It's similar to `$( expr $shots + 1 )`, but it executes faster because it doesn't require instantiation of a subshell.

From a functional perspective, `add_score` shouldn't test to see if you have accumulated the 13 brains needed to win or have been hit three or more times with gunshots—a lose scenario. Those show up a bit later in the script instead, just below the runner-savvy dice-rolling code shown earlier.

The code now invokes the routine, shown in my last article, that displays your current score:

```
show_score
```

Now for the big question: did you accumulate three or more gunshots? If you did, `show_score` will have returned a non-zero status, which can then be tested:

```
if [ $? -ne 0 ] ; then
   echo "BOOM. You died. But you did get to roll \
       $totalrolls times and eat $brains brains."
   exit 0
fi
```

If you survive that test, you're still alive! Hurray.

And that's where I'm going to stop for this month. Next month, I'll explore the rest of the brains/gunshots testing rules and have a functional one-player game. But that's not much fun, so I'll delve into creating a computer player too, suggesting some algorithmic strategies that should make it a decent player.

Meanwhile, go check out the real *Zombie Dice* game from Steve Jackson Games: **http://www.sjgames.com**.■

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.

Send comments or feedback via http://www.linuxjournal.com/contact or to ljeditor@linuxjournal.com.

KYLE RANKIN

Tails above the Rest, Part II

Now that you have Tails installed, let's start using it.

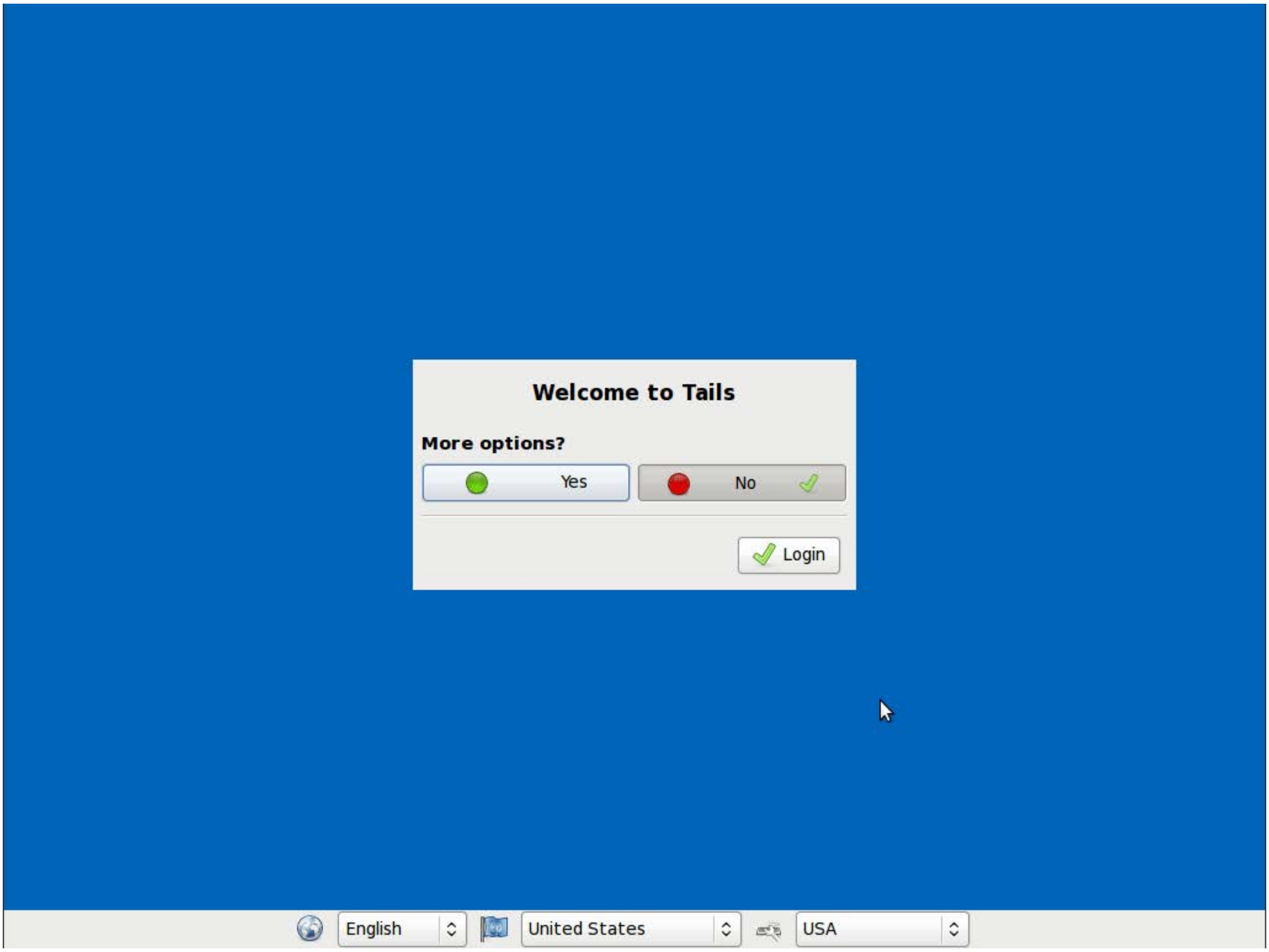
I'm halfway through what will likely be a three-part series on the Tails live disk. In the first column, I introduced Tails as a special distribution of Linux, based on Debian, that puts all sorts of privacy- and security-enhancing tools in a live disk you can boot anywhere. Then I talked about how to download and install the distribution securely on a CD or USB disk. In this article, I'm going to follow up with a general overview of the Tails desktop and highlight some of the software you are most likely to use within it. In my next column, I'll cover some of the more advanced features of Tails, including the persistent disk and encryption.

Tails Limitations

Before I talk too much about the security features of Tails, I think it's important to highlight the limitations that Tails has. Although Tails is incredibly useful and makes it much easier to use the Internet securely, it still isn't a magical solution that will solve all of your privacy problems. Before you use Tails, it's important to know where its limitations are and beyond that, mistakes that you might make that could remove some of the protections Tails does have.

Tails uses Tor to anonymize your Internet use, but that within itself has limitations. First, Tails doesn't attempt to hide the fact that you are using Tor or Tails, so if others can sniff the traffic leaving your network, while they may not be able to tell what Web sites you are browsing, they still can tell you are using Tor itself. So, if you are in a situation where you may get into trouble for using Tor, Tails out of the box won't protect you. Second, although traffic between you and Tor and between Tor nodes is encrypted, traffic that leaves Tor is not necessarily encrypted. Tails, like the Tor browser bundle, adds extensions to its Web browser to attempt to use HTTPS-encrypted sites whenever possible, but if you send an unencrypted e-mail or browse to an unencrypted Web site, the traffic leaving Tor still would be unencrypted. Along the same lines, you also still may be vulnerable to man-in-the-middle attacks launched from a malicious Tor exit node itself or from an attacker between the Tor exit node and the site you want to visit, so you still need to pay attention to any certificate warnings you see in your browser.

Generally speaking, Tails doesn't scrub your Internet traffic or any documents you create for any identifying metadata. If you decide to log in to a social-networking site



Welcome to Tails

More options?

Figure 1. The Tails Pre-Desktop Prompt

from inside Tails and then browse to other sites that integrate with that login, even though those sites will see that the traffic came from a Tor exit node and not from your personal computer, cookies and other identifying metadata from the social-networking site will out you. Generally speaking, you don't want to do anything within a single browsing session in Tails that may link on-line identities (like an e-mail account, social-networking login and the like) that you don't want linked. Likewise, if you write a document or edit a photo within Tails, it won't automatically remove any metadata that contains identifying information.

The Tails Desktop

Before you get to the Tails desktop itself, you are greeted with a login prompt (Figure 1) that asks if you'd like more options. These additional options allow you to use persistent volumes, set administrator passwords

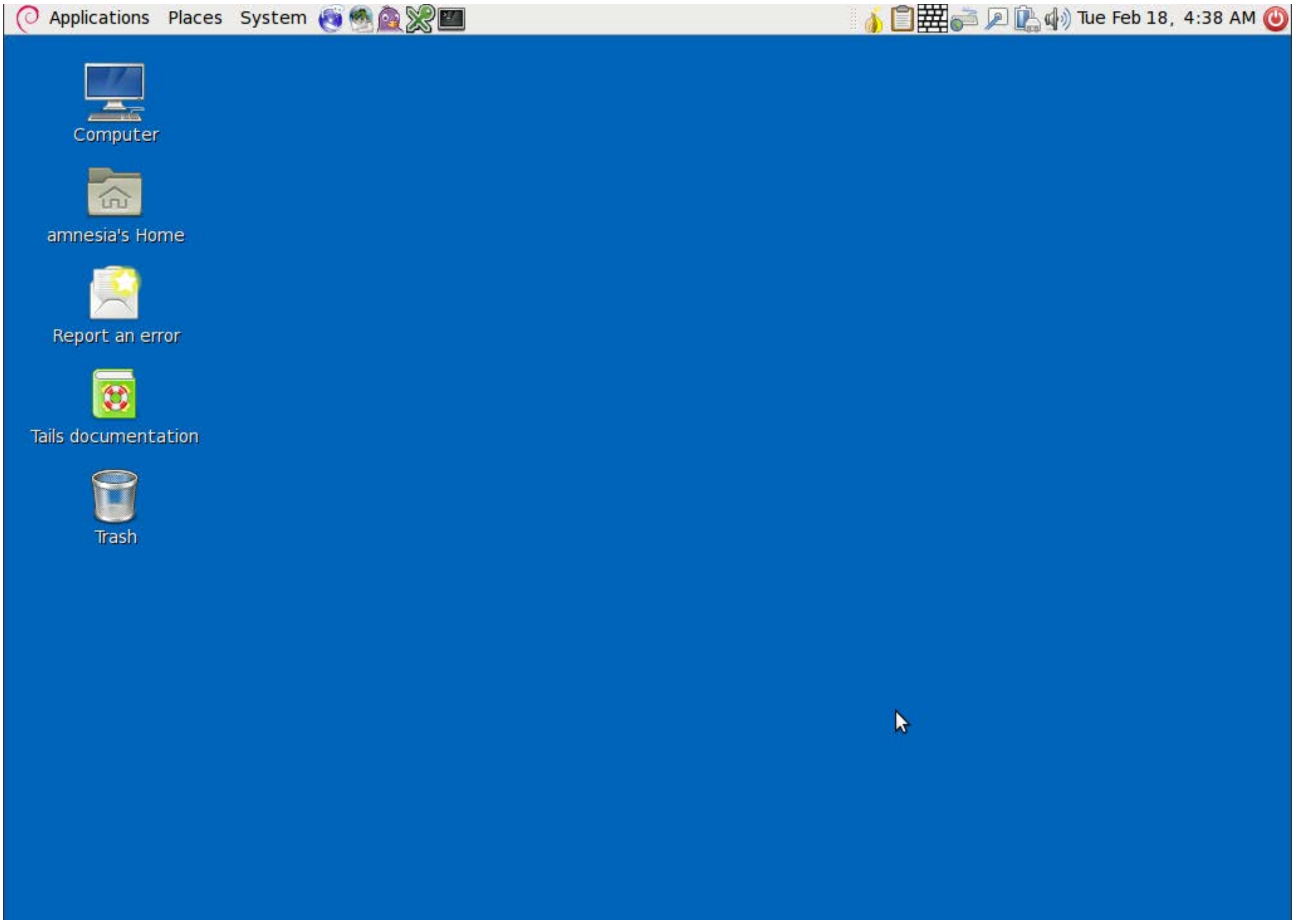


Figure 2. Default Tails Desktop

and go into incognito Windows mode. But, I'll cover more advanced features in a follow-up column, so in the meantime, just click Login.

Tails uses the all-too-familiar GNOME 2 desktop (Figure 2) with a panel along the top containing Applications, Places and System menus; a few icons for application shortcuts; and a notification area to the far right that lists the time along with icons, so you can see the status of the network, Tor, your battery (if you are on a laptop), a PGP applet and even an on-screen keyboard you can use to enter passwords if you suspect your computer might have a keylogger installed.

Tor and the Iceweasel Web Browser

The Tor Vidalia front-end application shows up in the notification area as an onion icon. The moment that Tails connects to a network, it will attempt to start up Tor, and this icon will change from yellow to green once Tor is fully up and configured. You can double-click the icon to open the Vidalia control panel to reset your Tor connection or view the current network. Once Tor is ready, Tails also will launch a Web browser configured much like the one in the Tor browser bundle with privacy-enhancing settings and plugins like NoScript (disables JavaScript), HTTPS-anywhere (attempts to connect to the HTTPS version of a Web page by default) as well as plugins so the browser uses Tor.

Like with the Tor Browser Bundle, all the sites you browse in the default browser go over the Tor network. The browser also uses search engines like Start Page in the default search bar. Start Page returns Google results but acts as a proxy to help anonymize your search queries. Don't be surprised if you sometimes get Web pages localized in a foreign language—Tor may route you over an exit node in a different country, and often sites try to be helpful and set the default language based on where they think you are from. If for some reason you need to use a Web browser outside Tor (for instance, so you can authenticate to an active portal on hotel Wi-Fi), there also is an unsafe browser option you can launch that bypasses Tor. Just be sure to close the browser once you are done so you don't mistakenly use it when you intend to browse over Tor.

Pidgin

Beyond browsing, instant messaging is another communication tool that could benefit from some privacy. Tails includes the Pidgin instant-messaging client and by default enables only the

Of course, it's worth saying that if you do access a personal e-mail account without using SSL encryption, even over the Tor network, someone who is sniffing the traffic coming from that Tor exit node, or sniffing traffic coming into your e-mail provider, will be able to correlate your account with that particular Tor session.

communication plugins for IRC and XMPP, as they are considered to have a decent security track record with respect to fixing security bugs. Each time you start Tails, it creates a random English-sounding user name for Pidgin to help aid in your anonymity. In addition, it includes the OTR (Off the Record) plugin that helps you have private IM conversations by not only encrypting the communication end to end, but it also authenticates the person you are chatting with, has forward secrecy, and even adds a deniability element to make it difficult outside the conversation to prove who said what (there's more information about how OTR achieves this at **http://www.cypherpunks.ca/otr**).

Other Applications

I'll cover e-mail in more depth in a follow-up column where I discuss encryption, but Tails includes the Claws mail client that you can use to access any personal e-mail accounts. Of course, it's worth saying that if you do access a personal e-mail account without using SSL encryption, even over the Tor network, someone who is sniffing the traffic coming from that Tor exit node, or sniffing traffic coming into your e-mail provider, will be able to correlate your account with that particular Tor session.

Beyond e-mail, Tails also includes the OpenOffice.org productivity suite, so you can work on documents and spreadsheets, the GIMP for image editing, and Audacity so you can listen to and edit audio files. Many people could very well spend their entire day within Tails and get work done.

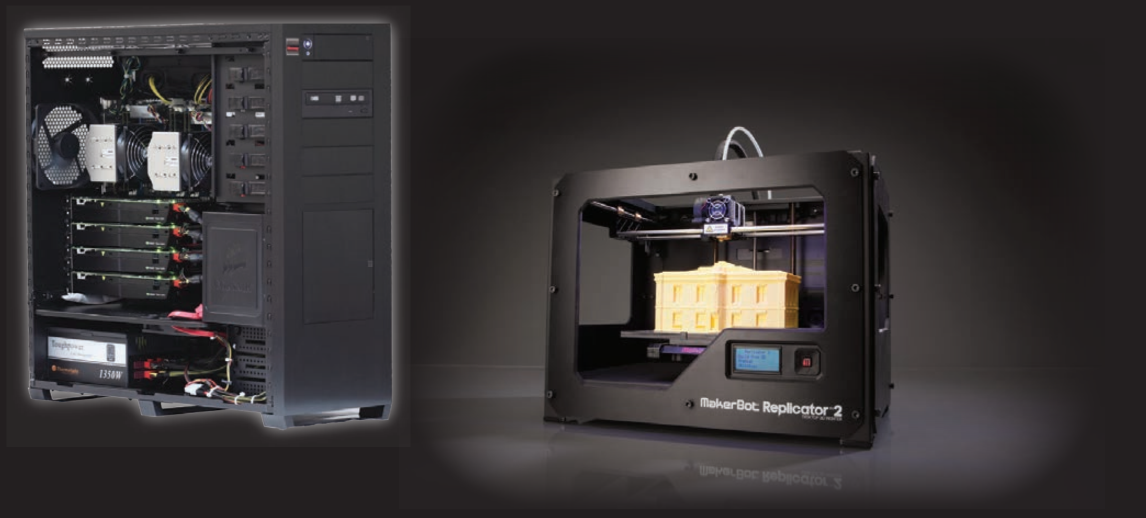
Shutdown

Once you are done with Tails, you can log out and select to reboot or shut down the computer. In either case, since anything that might identify you resides only in RAM, Tails makes a point to wipe the contents of RAM before it completely shuts down. I've noticed on my computers that this results in strange artifacts showing up on the screen during that process, but once it's done, Tails will shut down safely, and you can remove the DVD or USB drive.

This covers just some of the basic usage of Tails, but in my next column, I'll cover some of the more advanced uses, including persistent disks, encryption and some of the other internal Tails tools that are dense enough topics that they deserve their own treatment. In the meantime, enjoy your safe and private Internet browsing.■

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.

Send comments or feedback via http://www.linuxjournal.com/contact or to ljeditor@linuxjournal.com.



SHAWN POWERS

Your Media, Served Two Ways

XBMC or Plex? Why not both?

My family likes to watch TV. Yes, I know that makes us the cliché American family, over-saturated with media and slowly disconnecting ourselves from humanity, but that's just the way we are. We also are a family who takes our passions very seriously, so in the Powers home, movies and television shows are archived and accessible from pretty much any device in any room (and even over the Internet). Granted, we have more than 40 terabytes of storage in the basement, gigabit Ethernet in almost every room and a robust multi-access point wireless network that allows us to roam around the property without ever disconnecting. We're basically one step away from becoming the Borg.

Although many would consider our multimedia entertainment system overkill, the technology itself scales to meet any need. So in this article, I want to talk about our two main methods for watching media: Plex and XBMC. The former is what we use for mobile viewing, and the latter is for our set-top boxes. Yes, Plex can be used on a television, and XBMC can be used on a mobile device, but the two seem to excel at one thing over the other, so we use both.

XBMC—the Hardware

I think the current "best hardware" for building an XBMC set-top box is probably a Cubox-i from **http://www.solidrun.com**. A variety of models are available, but all run XBMC very well. They also include an infrared receiver for a remote control, but in my experience, success using that included receiver is inconsistent.

The other very popular option is a Raspberry Pi unit running OpenElec from an SD card. I have two of them running in our house, and apart from a slightly slow menu interface, they play quite well, even 1080p content. Many people purchase a Zotac Zbox

I’ve had a dozen or more varieties of XBMC and Windows MCE remotes through the years, and hands down the best experience has been with an inexpensive remote/USB receiver combination from Ortek.

or Intel NUC system for XBMC, but if all you plan to do is run XBMC, I think both options are fairly pricey. If you have cause to do more with your set-top box than just run XBMC (for instance, if you want to run a TV tuner or something), go with one of the more expensive non-ARM-based devices. If you just want to stream audio and video, I really recommend one of the less expensive options.

The Remote

Depending on what system you choose, you might end up with a remote out of the box. Chances are, however, you’ll need to purchase one. In the case of the Cubox-i, it’s possible to program any old remote you have lying around the house to be the controller for your XBMC install, but in my experience, it’s fairly difficult, and the result isn’t always worth it.

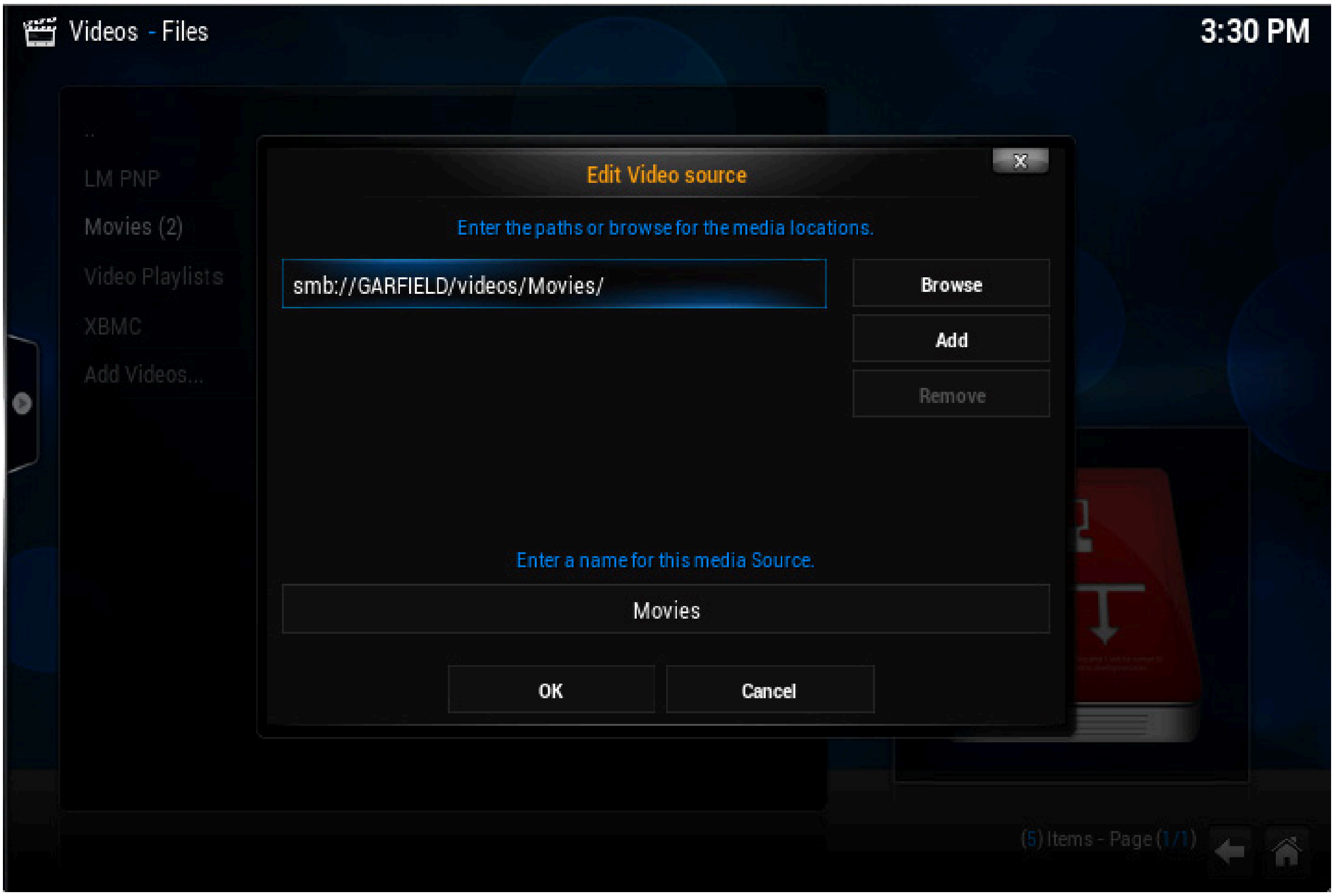
I’ve had a dozen or more varieties of XBMC and Windows MCE remotes through the years, and hands down the best experience has been with an inexpensive remote/USB receiver combination from Ortek. It costs less than $15 and emulates a mouse along with performing flawlessly as an XBMC remote. It’s even eligible for Amazon Prime if you’re a Prime subscriber: **http://www.amazon.com/Ortek-Windows-Infrared-Receiver-Ultimate/dp/B00224ZDFY**.

Most Windows MCE remotes will work with XBMC out of the box with no additional configuration required. That’s the case with the Ortec remote I mentioned earlier, but it’s probably true of any MCE remote you might have.

The Screen

This would seem like the easiest decision to make. Because XBMC runs on a computer, the screen can be any monitor or HDMI-capable television. Our experience has been that volume can be challenging, however, so either plan on a really nice set of amplified speakers to go along with your monitor (for small installs), or get a flat-screen television with decent volume. XBMC itself will let you adjust the volume with the remote (*such* a nice feature and one of my favorite things about XBMC), so you can leave the TV or speaker volume fairly high, needing to use the remote only for adjustments from the couch. I hate multiple remotes, so being able to turn the volume up and down via XBMC software is really awesome.

As far as which option is better, realize it depends on the need. In my living room, I have a huge flat-screen TV and use an audio receiver with big speakers. In my bedroom, I have a 720p television using its surprisingly loud built-in speakers. And then in my front room, I have a 22" monitor and a set of computer speakers. The front room setup is by far the most quiet, but since it's basically a reading room, I most often play a video of a fireplace anyway, so volume isn't an issue. (If you watch the monthly *LJ* issue intro videos, I record them in the front room in front of my XBMC machine.) The important thing is



Enter the paths or browse for the media locations.

smb://GARFIELD/videos/Movies/

Enter a name for this media Source.

Movies

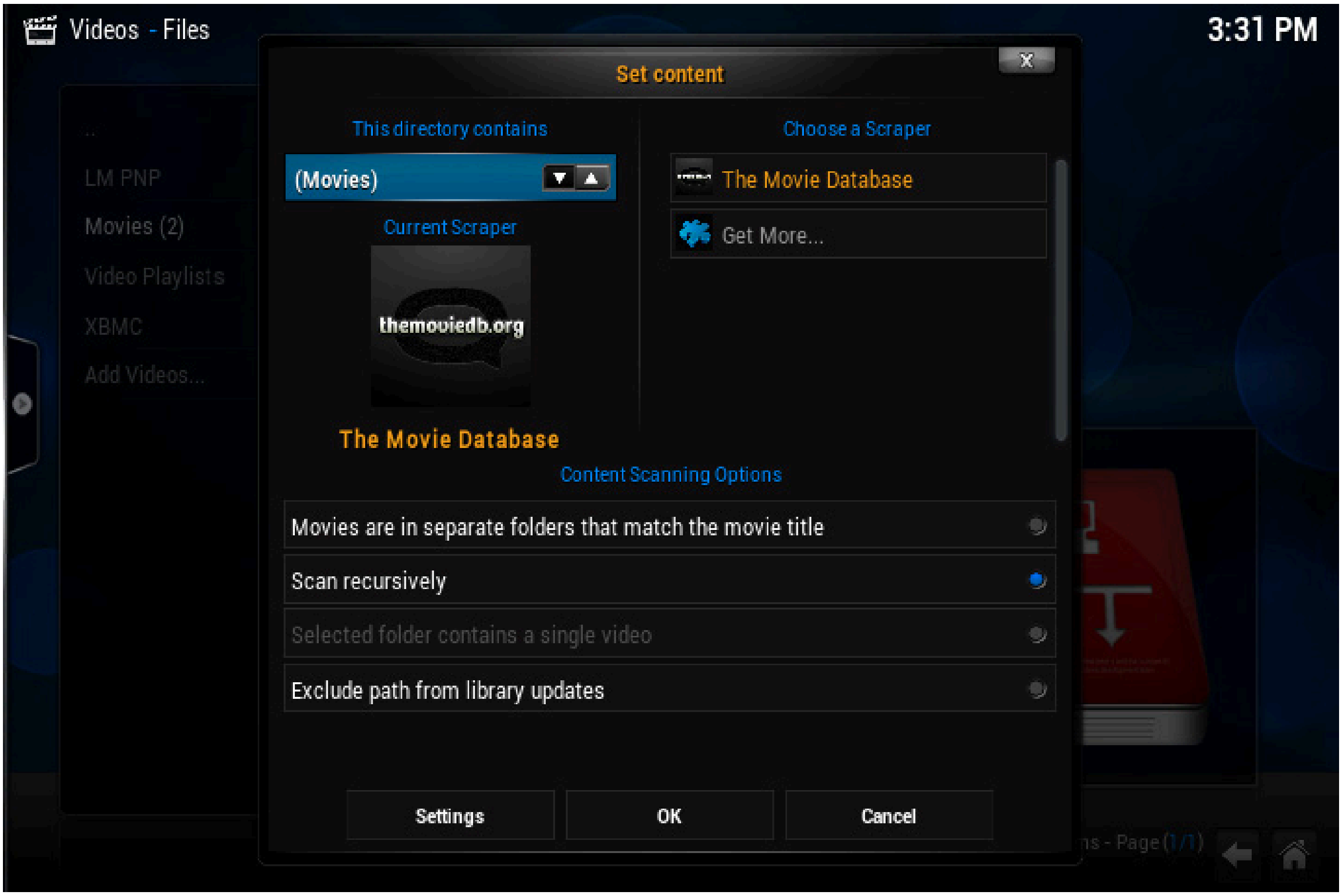
Figure 1. Adding a video source is much more fun if you have a USB keyboard attached.

to plan ahead so you have enough volume. It's easy to turn things down if they're too loud, but much more difficult to get louder than "10" on the dial of a small speaker.

XBMC + MySQL

Setting up XBMC is fairly straightforward. I actually recommend using OpenELEC (**http://www.openelec.tv**) as your operating system, because it works so well on a wide variety of hardware. OpenELEC on a Raspberry Pi functions almost identically to OpenELEC on a Zotac box. I'm not going to talk too much about actually installing OpenELEC—just follow the directions and you should be fine. The configuration, on the other hand, has to be done in a specific way if you want the XBMC machines to stay in sync.

Step 1—Adding Sources: The reason it's beneficial to share a common MySQL database among



Set content

This directory contains: (Movies)

Content Scanning Options

- Movies are in separate folders that match the movie title ○
- Scan recursively ●
- Selected folder contains a single video ○
- Exclude path from library updates ○

Figure 2. Be sure to select the type of video files in your source, so the scraper can download your metadata.

all your XBMC machines is that you can start a show in one room, and then move to another room and pick up where you left off. It also means metadata is shared between all the XBMC machines, and the library is updated once for your entire system instead of redundantly by every individual XBMC device.

I recommend setting up your video sources on a single XBMC machine (Figure 1) and making sure the video type is correct. In the configuration screen, you choose whether a particular source is Movies or Television, and the metadata scrapers do their work accordingly (Figure 2). One of the important things about a MySQL back end with XBMC is that every device has to have the exact same sources set up. Rather than add sources over and over, I prefer to get one machine completely set up and then copy the configuration files to the other units.

Step 2—Adding MySQL: Although adding sources can be done with the built-in menu system, MySQL is something that must be done from the configuration file. The first step, however, is getting a MySQL server ready. I assume you can install a MySQL server from your distro's repositories, so let's move on to configuring MySQL for XBMC. This is a little more complex than you'd think, and I found the easiest way is to give XBMC all the permissions it might want and then lock things down afterward. So basically, you create a super-powerful MySQL user named "xbmc"—on the command line, type the following:

```
$ mysql -u root -p
$ (type the password of your MySQL root user when prompted)
mysql> CREATE USER 'xbmc' IDENTIFIED BY 'xbmc'; (then press return)
mysql> GRANT ALL ON *.* TO 'xbmc'; (again press return)
mysql> quit (press return to exit the mysql program)
```

I'm sure half of you are screaming at how insecure it is to create such a powerful MySQL user, especially with such a horrible user/password combination. The other half of you is probably unconscious from fainting. I do *not* recommend you leave this user with all this functionality, but at first, it works best if XBMC can create the database and tables it wants. *After* everything is working, feel free to lock down the permissions a bit.

Step 3—Configuring XBMC: Armed with the IP address of your MySQL server, you either need to SSH in to the running XBMC device or somehow get to the configuration files. If you're using a Raspberry Pi with an SD card, you could mount the SD card on your workstation and edit the files from there. The important thing is to make sure you're in the right place. With OpenELEC, you want to edit the files in /storage/.xbmc/userdata (you should see a few files in there if you're in the right place). The sources.xml file contains the source information from the step above, and you'll want to copy that somewhere safe so you can use it on any subsequent installs of XBMC. (Remember, the sources have to match exactly, and copying the sources.xml is the easiest way to accomplish that.)

You most likely will not have a file named advancedsettings.xml in your folder. If you do, you'll need to add the information I'll talk about in a moment. You probably don't have one, however, so you'll need to create it and put the following text inside (matching your settings of course):

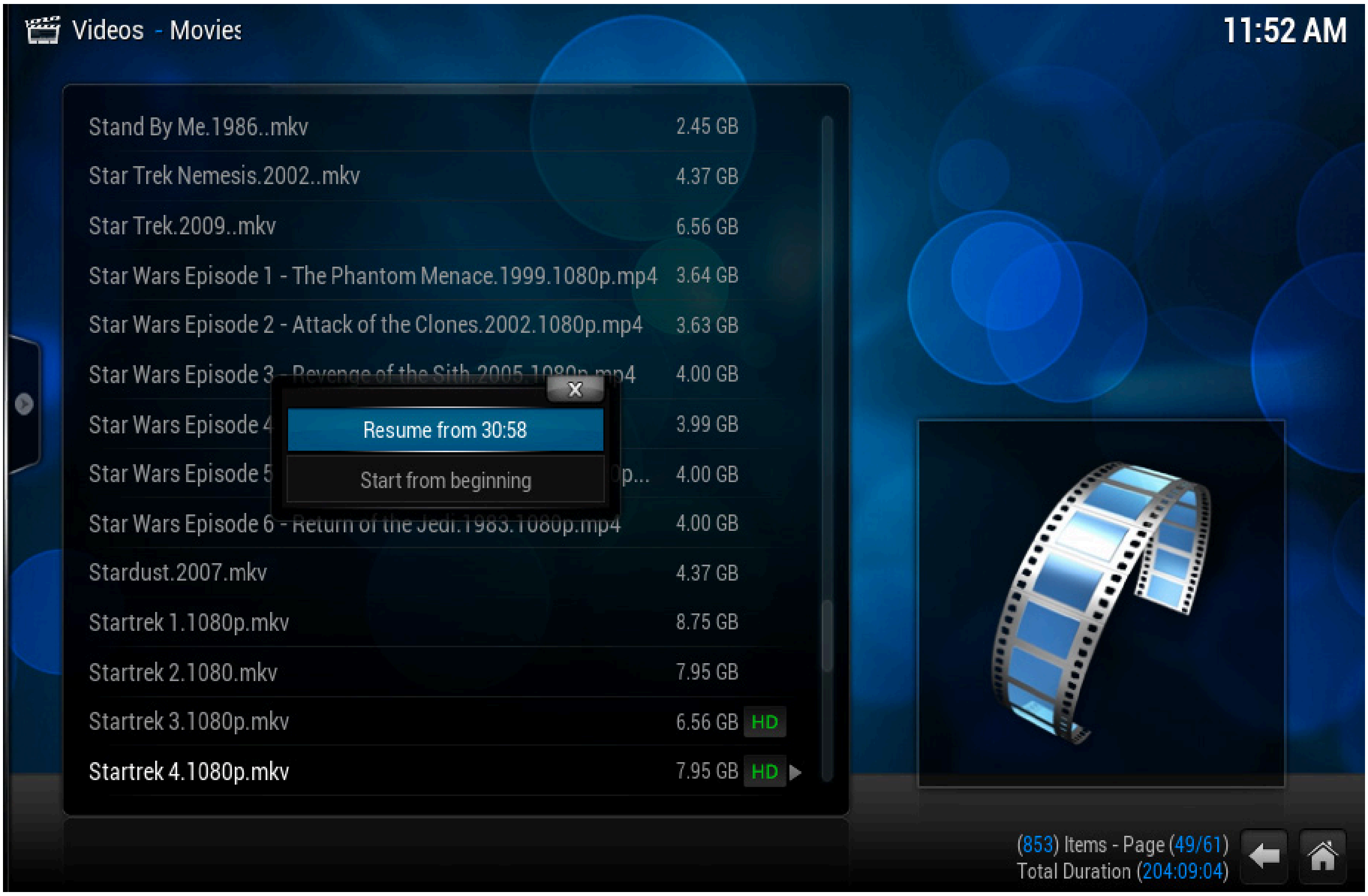
```
<advancedsettings>
    <videodatabase>
        <type>mysql</type>
        <host>192.168.1.11</host>
        <port>3306</port>
        <user>xbmc</user>
        <pass>xbmc</pass>
    </videodatabase>
</advancedsettings>
```

That's really all there is to set up. When you reboot your XBMC (after saving the advancedsettings.xml file into /storage/.xbmc/userdata/), the metadata scrapers should start filling your MySQL database with all the information, including where you leave off watching a particular movie. What MySQL database, you ask? Well, that's why you created the super-powerful xbmc user. XBMC should create a database on the server automatically, named something like MyVideos75, which I personally think is a strange naming convention, but whatever.

I probably should note that you can do the same for XBMC music and have it create a database for that too. I just never play music via XBMC, so it hasn't been something I've been interested in doing. Once you see the database has been created and is being populated with data from XBMC, you can feel free to lock down permissions a bit to make sure the xbmc user has access only to the specific database.

Wash, Rinse, Repeat

Now that the hard work is done, subsequent XBMC installs will be much, much simpler. Just install the base system, and then copy the sources.xml and advancedsettings.xml from your original system to the new system. Reboot the new system, and everything should be configured perfectly with shared metadata. To test it out, start watching a movie



Videos - Movies

11:52 AM

Stand By Me.1986..mkv	2.45 GB
Star Trek Nemesis.2002..mkv	4.37 GB
Star Trek.2009..mkv	6.56 GB
Star Wars Episode 1 - The Phantom Menace.1999.1080p.mp4	3.64 GB
Star Wars Episode 2 - Attack of the Clones.2002.1080p.mp4	3.63 GB
Star Wars Episode 3 - Revenge of the Sith.2005.1080p.mp4	4.00 GB
Star Wars Episode 4	3.99 GB
Star Wars Episode 5 ...p...	4.00 GB
Star Wars Episode 6 - Return of the Jedi.1983.1080p.mp4	4.00 GB
Stardust.2007.mkv	4.37 GB
Startrek 1.1080p.mkv	8.75 GB
Startrek 2.1080.mkv	7.95 GB
Startrek 3.1080p.mkv	6.56 GB HD
Startrek 4.1080p.mkv	7.95 GB HD

(853) Items - Page (49/61)
Total Duration (204:09:04)

Figure 3. Resuming in the next room means getting kicked off the big TV doesn't result in losing your place!

from the first XBMC system, and halfway through, press stop. Then go to the second system, and start to play the same video. You should be asked if you want to continue from where you left off, even though you're on a completely new device (Figure 3).

Having several different versions of XBMC (Frodo and Gotham, for instance) should work, but I've always kept my XBMC boxes at the same version to make sure nothing weird goes on between versions. I've tried many, many (oh so many) different set-top programs for watching videos on a television, and nothing compares to XBMC. It's smooth, attractive and works very well. When it comes to mobile viewing, however, XBMC isn't even close to the best option. Mobile viewing is where Plex really shines.

Plex, Oh So Cool

Unlike XBMC, Plex has a server back end much like MythTV does. This back-end/front-end combination means that a server can handle many clients (front ends) and transcode video appropriate for the device and bandwidth. Whereas XBMC plays video files from local or network locations, Plex creates streams of video for front-end devices in real time. Granted, sometimes those streams are just a direct transfer of the raw video, but Plex is smart and sends video that is appropriate for the front end.

The first step is to install Plexmediaserver. This is a Linux (or other) based program that runs on a server and creates the transcoded streams. Because video transcoding is involved, a beefy server is recommended. Head over to **https://plex.tv/downloads** and get the Plexmediaserver .deb or .rpm that matches your system. Install the package with your distribution's package manager, and let the server

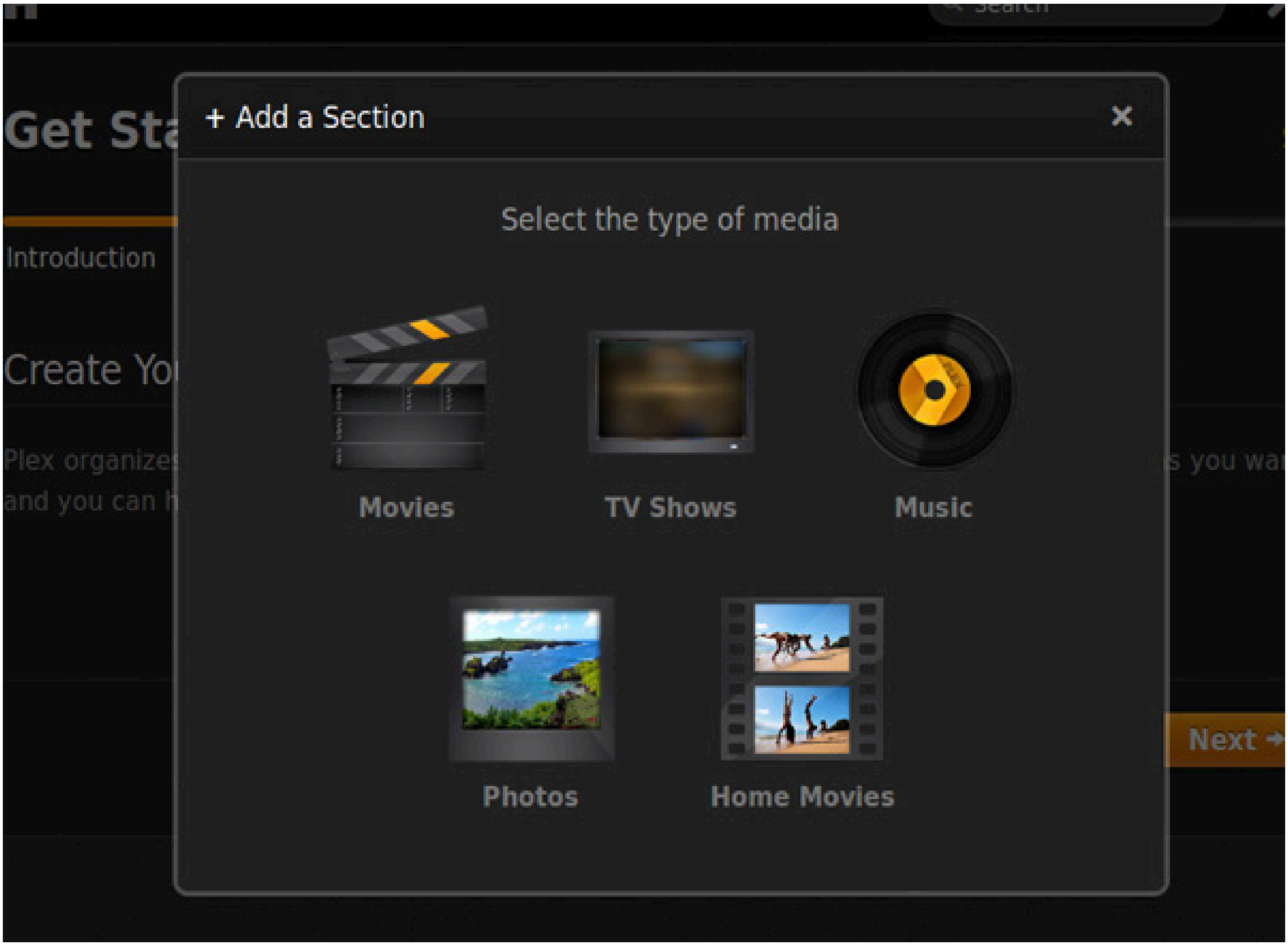
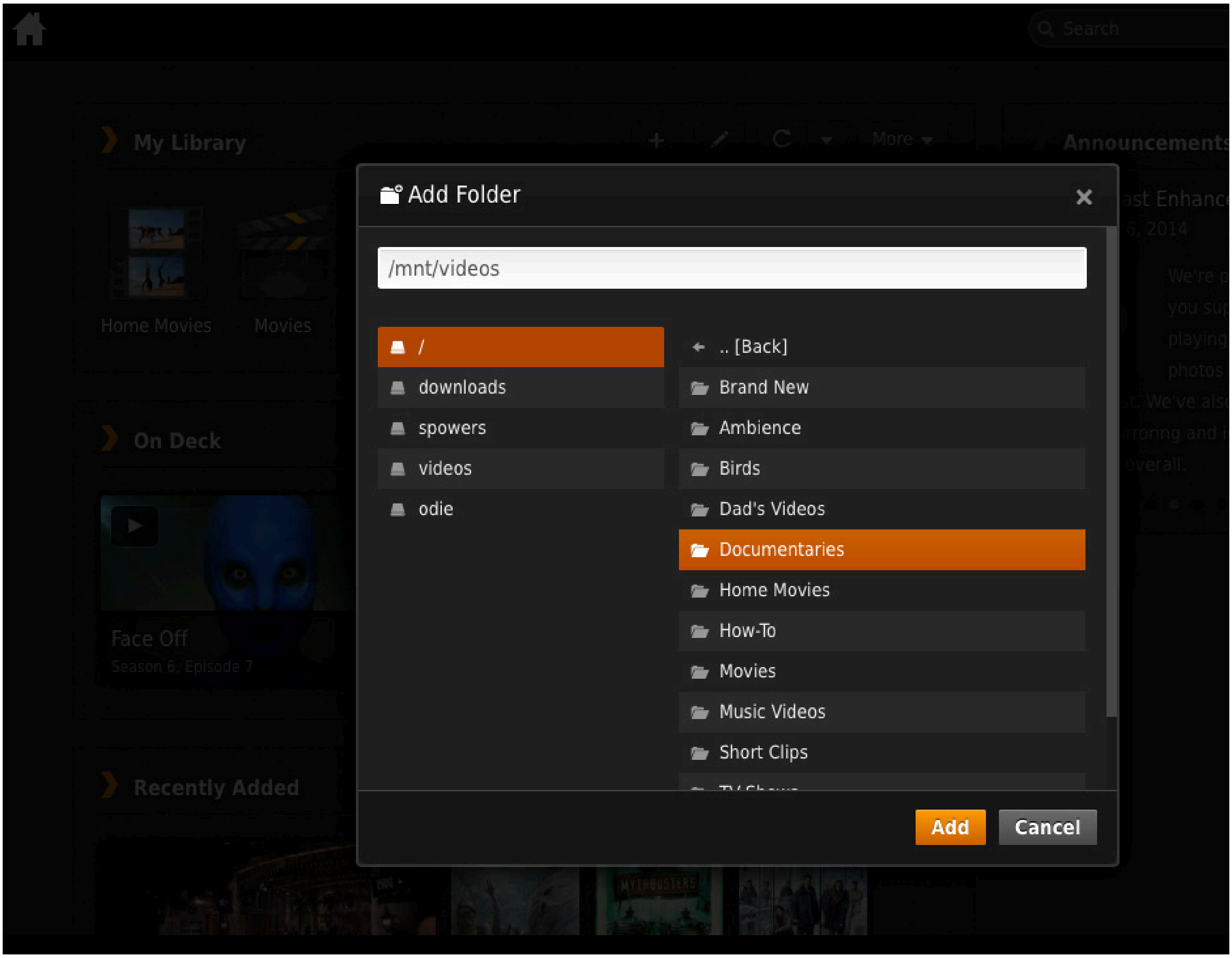


Figure 4. As with XBMC, choosing the source type means proper metadata scraping.



Add Folder

/mnt/videos

Figure 5. Plex requires videos to be in a local folder, but that folder certainly can be a network-mounted server.

start up. Configuration is all done over the Web, so once it's installed and running, head on over to http://your.server.ip:32400/manage/, and walk through the initial setup process. After you name your server and walk through the initial wizard, you'll need to add sources (Figure 4). This is slightly different from XBMC, because Plexmediaserver requires you to specify local folder locations. If you don't have it installed on the same machine you store your files, that's not a big deal, it just means you'll need to mount the remote file server on your Plexmediaserver's filesystem. That's actually what I do, and I never have a problem with slowdowns. Figure 5 shows my actual Plexmediaserver, and you can see the

NFS-mounted folders are easy to add.

Once you add all your various media locations and types, Plex will churn away for hours or days compiling its metadata. You still can watch videos during this time, but you might not see the pretty metadata for a while.

Oh Let Me Count the Ways

Plex is flexible. I mentioned the mobile applications available for Android and iOS earlier, and a few months back, the Android app was "Editors Choice" for its awesomeness. When you set up your Plexmediaserver, you'll be prompted to create an account with Plex. That account will allow you to log in to your server remotely and stream from your home video collection to your mobile device. Plex also allows you to use the awesome Plexweb interface (which again, I've written about before) by visiting **http://my.plexapp.com** and logging in.

If you start to love Plex as much as I do, you might be tempted to purchase a PlexPass, which is a subscription service giving early access to new features and a more robust platform. Currently, being a PlexPass subscriber means you can sync content locally to your mobile devices, so you can watch them without needing access to a network. I'm a PlexPass member, but honestly, I very rarely take advantage of any features it provides.

My Guilt-Ridden Conclusion

If anything, you realize by now that I watch far too much television. I agree with you, but I don't plan on stopping any time soon. I've recently integrated Live TV into our XBMC experience with a digital TV tuner and network-based PVR. Once my guilt wears off, perhaps I'll write about that process as well. Live TV is a relatively new feature for XBMC and setting it up is challenging at best. If you're a movie buff or just love setting up nerdy multimedia, you can't go wrong with the one-two punch of XBMC and Plex. Have fun, and don't forget the popcorn!■

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.

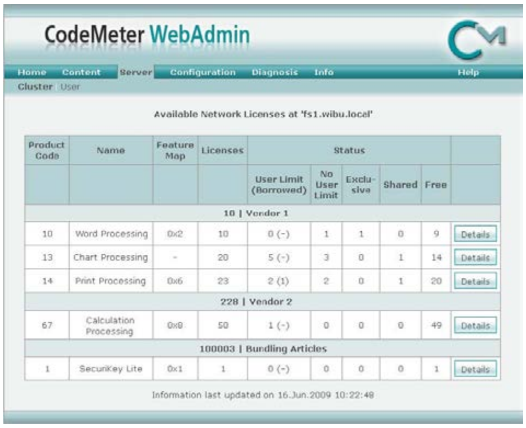
Send comments or feedback via http://www.linuxjournal.com/contact or to ljeditor@linuxjournal.com.

SUSE's kGraft

SUSE opines that the differentiating factor in its new live, runtime patching solution for the Linux kernel, called kGraft, is its status as the only competing solution in the upstream Linux kernel. By extension, none of the other major Linux distributions provide updates this way. kGraft, developed by SUSE Labs, makes it easier for IT staff to install critical security and other patches without system downtime—the holy grail of uptime. The net benefit for enterprise Linux users is a significant enhancement of uptime in mission-critical environments. SUSE adds that although kGraft is, by choice, limited to replacing whole functions and constants they reference, this does not limit the set of code patches that can be applied significantly. kGraft will offer tools to assist in creating the live patch modules, identifying which functions need to be replaced based on a patch and creating the patch module source code.

http://www.suse.com

Wibu-Systems' CodeMeter



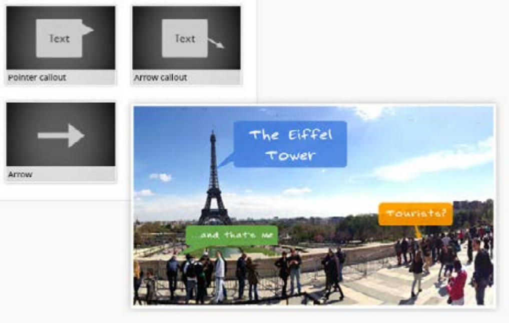
Available Network Licenses at 'fs1.wibu.local'

Product Code	Name	Feature Map	Licenses	Status					
				User Limit (Borrowed)	No User Limit	Exclu-sive	Shared	Free	
10 | Vendor 1									
10	Word Processing	0x2	10	0 (-)	1	1	0	9	Details
13	Chart Processing	-	20	5 (-)	3	0	1	14	Details
14	Print Processing	0x6	23	2 (1)	2	0	1	20	Details
228 | Vendor 2									
67	Calculation Processing	0x8	50	1 (-)	0	0	0	49	Details
100003 | Bundling Articles									
1	SecuriKey Lite	0x1	1	0 (-)	0	0	0	1	Details

Information last updated on 16.Jun.2009 10:22:48

Faced with increasing complexity in network environments and software agreements, IT staff members are challenged to keep accurate accounting of their software assets, contracts and entitlements. The CodeMeter software copy protection and licensing tool is a solution that simplifies this complexity. The new CodeMeter v5.10 now provides software developers with a graphical display in its WebAdmin tool that enables network-license customers to have an accurate accounting of all allocated licenses and users as well as rejected requests. End users simply can refresh their browsers to get instant updates of network license usage. CodeMeter's protection against software piracy, tampering and unauthorized usage can be software- or hardware-based and can support a wide range of licensing models. With CodeMeter, asserts Wibu-Systems, IT staff members can reduce application costs, allocate licenses efficiently and improve license security.

http://www.wibusystemsusa.com



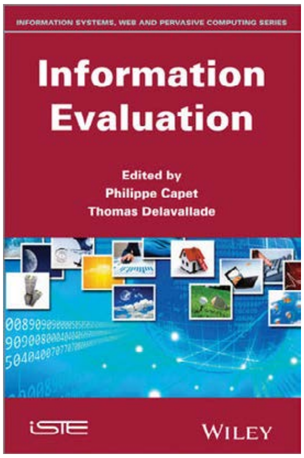
WeVideo

To its developers, WeVideo is significantly more than a mere collaborative video editing platform—it is a storytelling movement. Because WeVideo is cloud-based, social video editing is possible on the platform—that is, people joining together on-line to create a video project. Inspired by customer feedback, WeVideo's newest release includes one of the most requested features: Text Customization. Callouts also have been improved, and performance and rendering improvements are making the WeVideo experience smoother and more trouble-free. Exported files now are faster and of higher quality. The Android Video Editor, which WeVideo calls best of breed, now enjoys support for 720p and 1080p exports, Android 4.4 and Emoji symbols. Finally, the WeVideo Academy library of tips and step-by-step video lessons continues to grow, helping users more easily create the stories they want to tell.

http://www.wevideo.com

Philippe Capet and Thomas Delavallade's *Information Evaluation* (Wiley)

If you are browsing for a "For Dummies" book, please don't read on. On the other hand, if you are in the market for a real intellectual challenge on, say, the philosophical nature of knowledge and how humans process it, this is for you. The new book *Information Evaluation* edited by Philippe Capet and Thomas Delavallade explores how we humans view information and filter it based on our own personal beliefs and convictions. We bestow upon a piece of information a certain amount of confidence on its provenance and credibility. Capet and Delavallade seek to understand and explain how these judgments are conceived, in what context and to what end. Spanning the approaches offered by philosophy, military intelligence, algorithmics and information science, this book presents the concepts of information and the confidence placed in it. It further reveals ways to evaluate information for the good of the military, economic intelligence and, more globally, the informational monitoring by governments and businesses.


INFORMATION SYSTEMS, WEB AND PERVASIVE COMPUTING SERIES

Information Evaluation

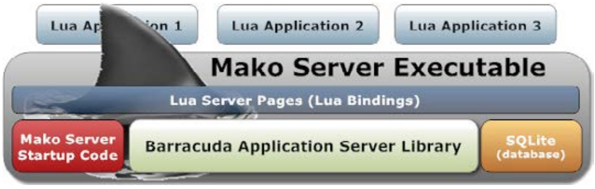
Edited by
Philippe Capet
Thomas Delavallade

ISTE

WILEY

http://www.wiley.com

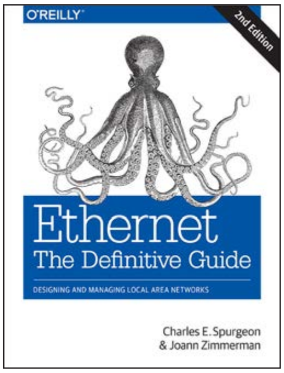
Real Time Logic's Mako Server



As a compact application and Web server, Real Time Logic's Mako Server helps developers rapidly design server-side Web applications. The Mako Server provides a bare-bones Web application server environment from which developers can design and implement complete, custom solutions. The latest Mako Server now includes the C source code, which enables developers to compile the Web application server for any embedded Linux platform. In addition, Real Time Logic's BarracudaDrive—a personal cloud server solution letting people set up and operate their own secure file-sharing site—has been released as a plugin for Mako Server. Releasing BarracudaDrive source code enables developers and technical users to compile the source code for many specialized embedded Linux platforms such as OpenWrt, dd-wrt, CuBox and so on.

http://makoserver.net

Charles E. Spurgeon and Joann Zimmerman's *Ethernet: The Definitive Guide*, 2nd edition (O'Reilly Media)



The updated 2nd edition to Charles E. Spurgeon and Joann Zimmerman's *Ethernet: The Definitive Guide* is classic O'Reilly—practical, to the point, with the quintessential animal on the cover. In *Ethernet*, readers discover what it takes to build, manage and scale Ethernet LANs, from basic Ethernet operation to network management. Further, they'll learn the answers to common questions, such as "What can I do to make sure that my Ethernet network works as well as possible?" "When do I need to upgrade to higher speed Ethernet, and how do I do that?" "How do Ethernet switches work, and how can I use them to build larger networks?" "How can I manage the network, what problems should I be looking for, and how can I troubleshoot the system when problems arise?" This thoroughly revised 2nd edition includes descriptions of the most widely used Ethernet media systems, including 10, 40 and 100 Gigabit Ethernet, as well as a complete glossary of terms used throughout the book and a resource list.

http://www.oreilly.com

Untangle Inc.'s Next Generation Firewall

Because the operating margins of small- and medium-size businesses are so slim, these operations can't afford downtime. To improve business continuity for these SMBs, Untangle Inc. developed the open-source Next Generation (NG) Firewall, which is now in version 10.1. NG Firewall is a next-generation platform for deploying network-based applications. The platform unites these applications around a common GUI, database and reporting. NG Firewall's applications inspect network traffic simultaneously, greatly reducing the resource requirements of each individual application. The new version 10.1 now provides high availability via virtual router redundancy protocol (VRRP). It allows organizations to deploy multiple Untangle servers as hot or cold backups, and in the event of a system failure, one of the backup servers will take over. Network users will continue to have uninterrupted access to the Internet, minimizing the impact of an outage.

http://www.untangle.com

Stealth.com Inc.'s LPC-630F PC

Stealth.com's LittlePC products are targeted at demanding applications, such as HMI, embedded control, digital signs, kiosks, process control, mobile navigation, thin clients and data acquisition. The company's latest addition to the LittlePC line is the LPC-630F, a high-performance, fanless, small-form computer with third-generation Core i7 processing power. Stealth.com says that the LPC-630F is loaded with features generally found in systems many times its diminutive size. The machines are encapsulated in a rugged extruded aluminum chassis performing as a heat sink to dissipate heat build-up. Its compact size 7.9" x 7.9" x 2.56" (200 x 200 x 65 mm) makes it ideal for space-challenged applications. Systems are compatible with Linux and Microsoft Windows and can be custom configured to meet the exact needs of the OEM or end user.

http://www.stealth.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.

A PROCESS FOR MANAGING AND CUSTOMIZING HPC OPERATING SYSTEMS

I've just seen the largest group of computers I've ever seen, and they expect me to manage this into an HPC cluster?!?!?! Don't panic, we have a process you can implement that will help manage communication and change of the software on the system.

DAVID BROWN

High-performance computing (HPC) for the past ten years has been dominated by thousands of Linux servers connected by a uniform networking infrastructure. The defining theme for an HPC cluster lies in the uniformity of the cluster. This uniformity is most important at the application level: communication between all systems in the cluster must be the same, the hardware must be the same, and the operating system must be the same. Any differences in any of these features must be presented as a choice to the user. The uniformity and consistency of running software on an HPC cluster is of utmost importance and separates HPC clusters from other Linux clusters.

The uniformity also persists over time. Upgrades and security fixes should never affect application correctness or performance. However, security concerns in HPC environments require updates to be applied in a timely fashion. These two requirements are conflicting and need to be managed by well documented processes that involve testing and regular outages.

A process for managing these requirements was developed at the Environmental Molecular Sciences Laboratory (EMSL) during the past ten years. EMSL supports HPC for the United States Department of Energy (DOE) and the open science community. This process gives EMSL an edge in maintaining a secure platform for large computational chemistry simulations that complement instrument research done at EMSL.

UPGRADES AND SECURITY FIXES SHOULD NEVER AFFECT APPLICATION CORRECTNESS OR PERFORMANCE.

REQUIREMENTS

The process developed at EMSL to maintain HPC clusters has roots in standard software testing models. The process involves three phases: build testing, integration testing and production. These phases have their own requirements both in hardware, software and organization. Other important systems include configuration management, continuous monitoring and repository management. All of these systems have well defined roles to play in the overall process and need dedicated hardware, not part of the production cluster, to support them.

The build integration phase requires two primary components: package repository management and continuous integration software. These two components interact and give software developers and system administrators knowledge of bugs in individual pieces of software before those updates affect integration testing. This form of testing is important to automate for critical applications because it helps facilitate communication between operations and development teams.

The integration testing phase requires a test cluster that is close to matching the production cluster. The primary difference between the production and test clusters, for HPC, is scale. The test cluster should have a lower number, but at least one, of every Linux host in the production cluster, including configuration management and continuous monitoring. Furthermore, the Linux hosts should be as close to matching production configuration as possible. Any deviations between the production and test clusters' configuration, both in hardware and software, should be well documented. This document will help define the accepted technical risks that might be encountered during production outages.

The production cluster is the culmination of all the preparations done in the build and testing phases. Leading up to the outages, documented tasks during the outage should be identified along with planned operating system upgrades. Storing these documents should be easily accessible for both developers and management to see as well as easy for operational staff to modify and track issues. Along with the plan, documented processes for moving configuration management and continuous monitoring from testing to production also should be followed.

We have identified some required infrastructure needed to support and automate the process for managing your own Linux HPC operating system. During the build integration phase, a dedicated build system is needed along with package management and continuous integration software. The integration testing phase requires test cluster hardware and continuous monitoring and configuration management software. Finally, the production cluster also should integrate with configuration management and continuous monitoring software.

Several systems are not covered here but are critical to integrate into the process. Site-specific backup solutions should be considered for every phase of the process. Furthermore, automated provisioning systems also should be considered for use with this process. At EMSL we have used both, but it's certainly not required by the process; it just makes sleep better at night.

THE CONTENT OF THE OVERLAY REPOSITORY IS SPECIFIC TO THE CRITICAL APPLICATIONS IN THE DISTRIBUTION THAT NEED TO BE MANAGED SEPARATELY.

BUILD PHASE

The build phase is the start of the process. There are three inputs into the system: binary packages, source code packages and tickets. These three inputs produce three outputs: a set of base repositories, a set of patches for upstream contribution and an overlay repository of modified packages. These inputs and outputs provide the operating system fixes needed for your site while contributing them back to the communities that support them. To understand this process completely, let me to break down the components and talk about their requirements.

The package repository management system is utilized throughout the process but first appears in the build phase. The package repository management system should be able to download binary package repositories from an upstream distribution. It also should be able to keep those downloaded repositories in sync with the upstream distribution. The first set of repositories should be a local copy of the upstream distribution, including updates, synchronized daily. As an added feature, the package repository management system also should be able to remove certain packages selectively from being downloaded. This feature complements the contents of the overlay repository. The overlay repository is the place where custom builds of the packages get put to enhance the base distribution.

The content of the overlay repository is specific to the critical applications in the distribution that need to be managed separately. For example, HPC sites might be more concerned about the kernel build, openfabrics enterprise distribution (OFED) and software that implements the message passing interface (MPI). This software is then removed from the base distribution and added back in an overlay repository. Furthermore, there can be multiple overlay repositories. For example, security concerns may dictate that the kernel needs to be managed separately from the rest of the distribution. Having the kernel in a separate overlay repository means that the testing phase can be skipped with minimal impact and still maintain a secure cluster.

The packages that are in the overlay repository are patched to match the needs of the organization. The continuous integration system should be used to patch the specific packages and maintain the build with future updates. These patches should be issued back to the upstream distribution along with good reasons why this patch was needed. Some of these patches may get accepted by upstream developers and make it into the distribution while others may take years to make it due to policy decisions on the part of the distribution maintainers.

Another job of the continuous integration system is to support the continuous build and testing of additions to the distribution that are not supported. These additions may be site-specific applications or open-source software not supported by the distribution. Many open-source software projects support compatibility with enterprise distributions but do not seek distribution inclusion because of financial project support reasons.

The final piece to the build phase is the ticket-tracking system. This system provides package developers information into what needs to be fixed and how. These tickets may come directly from users or from cluster administrators. Furthermore, the users and cluster administrators may use completely different ticketing systems. This piece of the process helps facilitate communication between groups. Having a list of tickets allows objective discussion about priority and makes sure tickets are not forgotten. Tickets may stay open for years or days, depending on priority and rate of ticket creation. The tickets do not stop with the package developers; the cluster administrators use this system in further phases.

The package management and continuous integration systems are automated processes, while the ticket-tracking system requires human interaction. These systems can be deployed on a single host. However, there is a requirement that three copies of the package repositories be present for the later phases of the process. Furthermore, there are features of the continuous integration system that integrate with the ticket-tracking systems. Enabling this feature does require a certain level of stability in the continuous integration build process. Many of the specifics in these systems are not covered here and will be covered later.

TEST PHASE

The integration testing phase requires the package repository management, continuous monitoring and configuration management systems. These three systems help maintain the test cluster in a state that integration testing be done by some automated processes. Furthermore, the test cluster hardware configuration should represent all critical aspects of the production cluster such that it mitigates risk to production clusters.

The package repository management system does play a role in all three phases of the process. This is the first phase of the process where the packages with additions are tested in production-like configurations. The daily repositories, including the overlay repository, are synchronized to a set of testing repositories to be included in the test cluster. This synchronization ensures a consistent environment to perform tests.

Every time updates are synchronized to the testing repositories, a set of integration tests should be performed on the test cluster. These tests should be designed to simulate the usage of the production cluster. It's important to focus the tests on critical user-level applications and parts of the operating system you have replaced and put into the overlay repository. The continuous integration system can run these tests and alert on failures.

Failures in the integration tests should be reported in the ticket-tracking system. This is one of the paths to complete the circle of development. Other tickets include deployment and re-install issues. Complex internal infrastructure in the production cluster also may present upgrade issues, and those issues also should be tracked. The test cluster also should be managed by the same procedures as the production cluster. The procedures should be practiced on the test cluster to minimize tickets before updates get deployed to the production cluster. All of these tasks should be performed in repetition until the addition of new tickets is reduced to minimize the risk to the production cluster.

The configuration management and continuous monitoring systems are set up in a similar way between the test and production environments. These two systems help maintain the production state from inadvertent hardware or software changes and, thus, need to be tested when deliberate changes are made. These changes need to be integrated into the production environment easily. So, maintaining the configuration for these two systems in a source code management repository that supports branching and merging also is prudent. This allows for standard processes for making changes and pushing those changes between testing and production environments.

When the number of tickets have been reduced and it is time to push the changes to the production cluster, five things come out of this phase: an updated set of package repositories, a set of tasks that need to be done during an outage, updated procedures to be used on the production cluster, changes that need to be merged to production for the configuration management and continuous monitoring systems. Both the package developers and the cluster administrators need to collaborate on the procedural changes and the outage tasks. This collaboration works well in a wiki environment that is internal to both groups. These outputs conclude the integration testing phase of the process.

PRODUCTION PHASE

The production phase of the process takes the results from the integration testing phase and applies them to the production cluster. This phase utilizes all of the same processes as the testing phase, with a few modifications. Furthermore, this is the phase where users get to affect change in the process. There is also an increase in more formal communication methods through software between groups. The final outputs of this phase feed back to help complete the development and testing cycle as well. After this phase is completed, the process is finished, and the updated production environment will be maintainable.

The first part of this phase is the replication of what was done with the package repositories. However, this phase requires that production copies

Table 1. Components and Examples of Open-Source Software That Would Meet the Requirements within the Process

COMPONENT	OPEN-SOURCE SOFTWARE
Continuous Monitoring	Nagios, Simple Event Correlator, Auditd
Package Repository Management	Cobbler
Continuous Integration	Jenkins, Hudson
Ticket Tracking	Trac, Bugzilla
Wiki Documentation	Trac, Drupal WordPress
Provisioning	Cobbler
Configuration Management	Cfengine, Chef, Salt, Puppet
Backup Software	Bacula

of the repositories be synchronized from the integration testing repositories. This is the final set of package repositories required by the process. Furthermore, production configuration of the continuous monitoring and configuration management system also should be created from the integration testing configuration of the respective systems.

Users of the production cluster get input into the process during this phase. Depending on the users' requirements, this may be a different instance of the ticket-tracking system or the same one as used by the package developers and cluster administrators. Either way, it's important to track this input so it makes it through the process without getting forgotten.

Communication is key to this part of the process. From the testing phase, we know what tasks need to be completed on the production cluster during an outage and how long they should be expected to take. This helps management determine cost and benefit of the outage to determine a path forward. There is also continued communication required during outages when differences between the production and test clusters bring unexpected issues. These issues should be mitigated quickly, but tickets should be issued to ensure proper resolution of the issue so it never happens again.

There is always an importance of being prepared for production cluster outages. However, it is impossible to be completely prepared for every possible contingency. The differences between the test cluster and production cluster configuration will help to define the highest risks to any particular outage. It is critical to communicate these risks and any changes that might be impacted by those risks to management prior to outages.

CONCLUSION

The process described here does seem like a lot of overhead, and it may seem not applicable to your situation. However, the process does have specific circumstances where the testing phase can be skipped. Furthermore, this process is generic enough to be scaled to your needs. There are many pieces of open-source or proprietary software that can meet the requirements of this process.

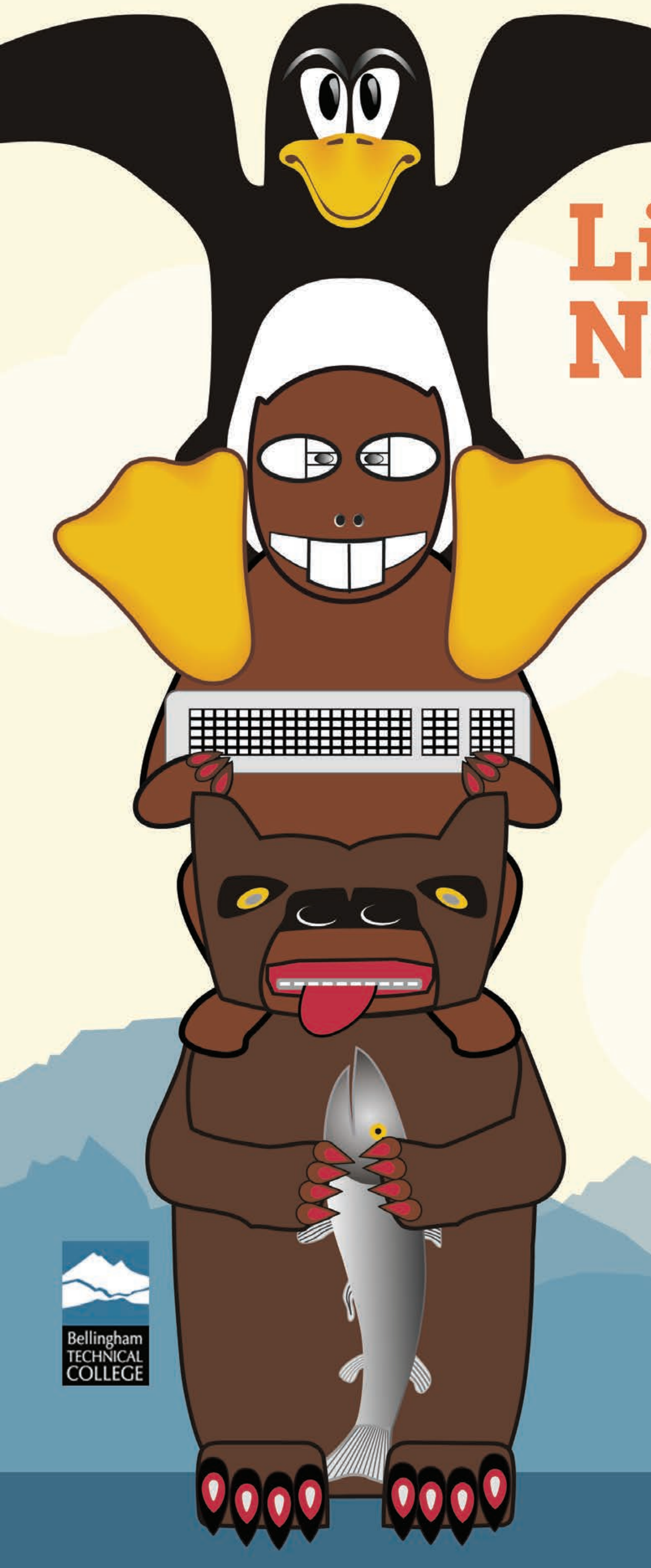
Skipping the testing phase process easily can be done by pulling the critical applications into separate overlay repositories so they can be managed separately. Then make sure the process for getting updates is put into the continuous integration system. This may just be a Web site download that pulls the appropriate software into its respective overlay repository. Then simply synchronize the overlay repository to test and then production. This is done immediately to push security updates to production systems.

Similarly with production configuration changes, many times unexpected issues during an outage demand that configuration changes be made directly to production systems. These changes should be able to be made directly in the production configuration management system then merged back to test when the outage is over. If the changes to production need development to be made more generic, this should happen in the build and integration testing phase. The final solution then should be pushed to production during an outage.

In conclusion, the process described here is simply suggestive in nature. If the process needs to be modified to get things working again, do so. However, after fixing issues, keep in mind the part of the process the issue relates to and inject what has been done into the process. This process is generic and flexible to manage these sorts of changes as well as keep systems updated while managing communication through well defined systems. ■

David Brown is a high-performance computing system administrator with a B.S. in Computer Science from Washington State University. He has worked at the Pacific Northwest National Laboratory (PNNL) in the Environmental and Molecular Sciences Laboratory (EMSL) since January, 2007. He also is a Fedora Package Maintainer and supports several scientific and administrative packages that are used in HPC environments. He has experience in high-performance filesystems (Lustre) and cloud technology (OpenStack).

Send comments or feedback via http://www.linuxjournal.com/contact or to ljeditor@linuxjournal.com.



LinuxFest Northwest

Bellingham, WA
April 26th & 27th

- 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!

Bellingham
TECHNICAL
COLLEGE

How YARN Changed Hadoop Job Scheduling

Got Cluster Scheduling? It's old hat for HPC admins, but have you ever wondered how Hadoop does workload management and job scheduling?

ADAM DIAZ

Scheduling means different things depending on the audience. To many in the business world, scheduling is synonymous with workflow management. Workflow management is the coordinated execution of a collection of scripts or programs for a business workflow with monitoring, logging and execution guarantees built in to a WYSIWYG editor. Tools like Platform Process Manager come to mind as an example. To others, scheduling is about process or network scheduling. In the distributed computing world, scheduling means job scheduling, or more correctly, workload management.

Workload management is not only about how a specific unit of work is submitted, packaged and scheduled, but it's also about how it runs, handles failures and returns results. The HPC definition is fairly close to the Hadoop definition of scheduling. One interesting way that HPC scheduling and resource management cross paths is within the Hadoop on Demand project. The Torque resource manager and Maui Meta Scheduler both were used for scheduling in the Hadoop on Demand project during Hadoop's early days at Yahoo.

This article compares and contrasts the historically robust field of HPC workload management with the rapidly evolving field of job scheduling happening in Hadoop today.

Both HPC and Hadoop can be called distributed computing, but they diverge rapidly architecturally. HPC is a typical share-everything architecture with compute nodes sharing common storage. In this case, the data for each job has to be moved to the node via the shared storage system. A shared storage layer makes writing job scripts a little easier, but it also injects the need for more expensive storage technologies. The share-everything paradigm also creates an ever-increasing demand on the network with scale. HPC centers quickly realize they must move to higher speed networking technology to support parallel workloads at scale.

Hadoop, on the other hand, functions in a share-nothing architecture, meaning that data is stored on individual nodes using local disk. Hadoop moves work to the data and leverages inexpensive and rapid local storage (JBOD) as much as possible. A local storage architecture scales nearly linearly due to the proportional increase in CPU, disk and I/O capacity as node count increases. A fiber network is a nice option with Hadoop, but two bonded 1GbE interfaces or a single 10GbE in many cases is fast enough. Using the slowest practical networking technology provides a net savings to a project budget.

From a Hadoop philosophy, funds really should be allocated for additional data nodes. The same can be said about CPU, memory and the drives themselves. Adding nodes is what makes the entire cluster both more parallel in operation as well as more resistant to failure. The use of mid-range componentry, also called commodity hardware is what makes it affordable.

Until recently, Hadoop itself was a paradigm restricted mainly to MapReduce. Users have attempted to stretch the model of MapReduce to fit an ever-expanding list of use cases well beyond its intended roots. The authors of Hadoop addressed the need to grow Hadoop beyond MapReduce architecturally by decoupling the resource management features built in to MapReduce from the programming model of MapReduce.

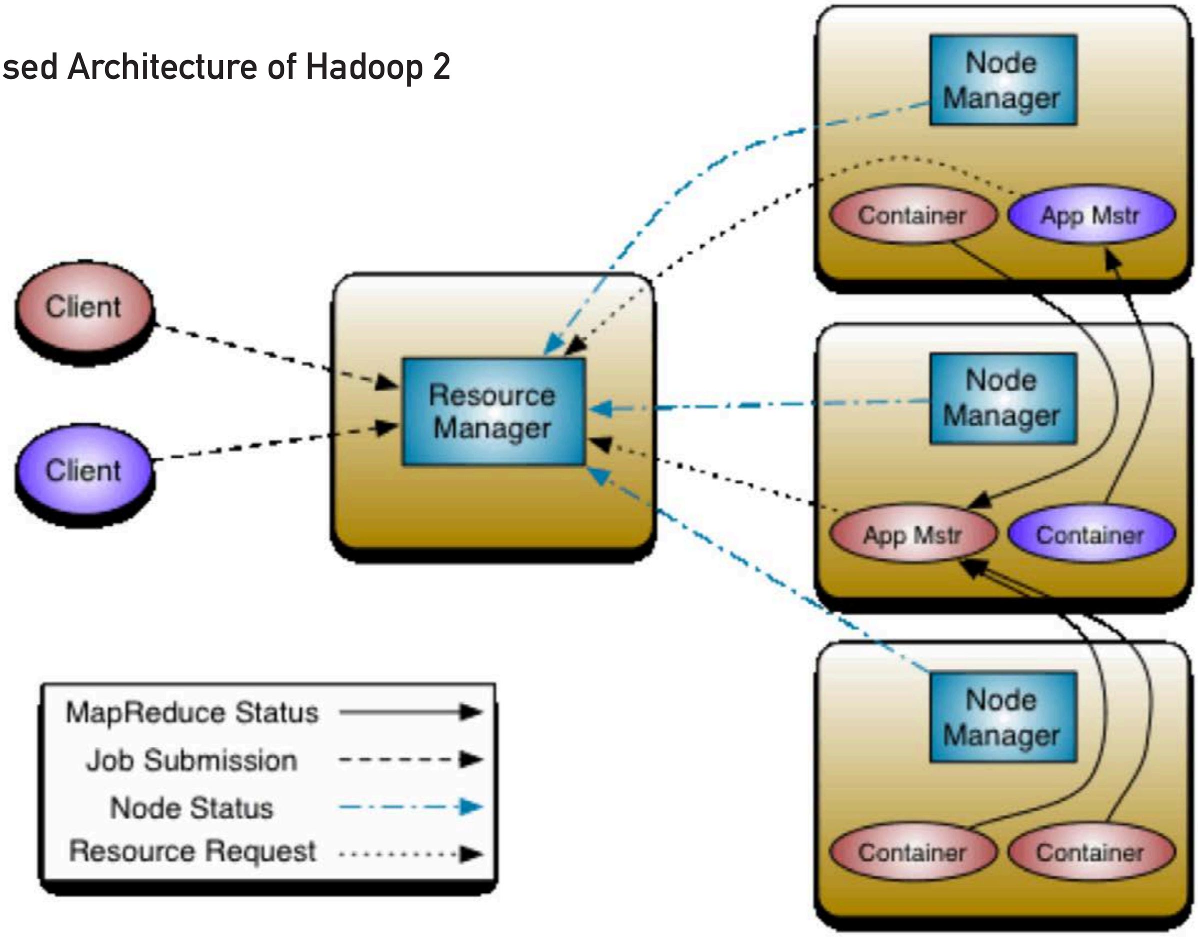
The new resource manager is referred to as YARN. YARN stands for Yet Another Resource Negotiator and was introduced in the ASF JIRA MAPREDUCE-279. The YARN-based architecture of Hadoop 2 allows for alternate programming paradigms within Hadoop. The architecture uses a master node dæmon called a Resource Manager consisting of two parts, a scheduler and Application Manager.

The scheduler is commonly called a pure scheduler in that it is only managing resource availability from the node manager on the data nodes. It also enforces scheduling policy as it is defined in the configuration files. The scheduler functions to schedule containers that are customizable collections of resources.

The Application Master is itself a container, albeit a special one, sometimes called container 0. The Application Master is responsible for launching subsequent containers as required by the job. The second part of the Resource Manager, called the Application Manager, receives job submissions and manages launching the Application Master. The Application Manager handles failures of the Application Master, while the Application Master handles failures of job containers. The Application Master then is really an application-specific container charged with management of containers running the actual tasks of the job.

Refactoring of resource management from the programming model of MapReduce makes Hadoop clusters more generic. Under

Figure 1.
YARN-Based Architecture of Hadoop 2



YARN, MapReduce is one type of available application running in a YARN container. Other types of applications now can be written generically to run on YARN including well-known applications like HBase, Storm and even MPI applications. The progress of MPI support can be seen in the Hamster project and a project called mpich2-yarn available on GitHub. YARN then moves from being a scheduler to an operating system for the Hadoop supporting multiple applications on a distributed architecture.

Architecturally, HPC workload management has many similarities to Hadoop workload management. Depending on the HPC workload management technology used, there is a set of master nodes containing cluster-controlling dæmons for accepting and scheduling jobs. The master node(s) in many cases contains special configurations including sharing of important cluster data via networked storage to eliminate SPOF of master services. On the worker node side, there exists one or more dæmons running to accept jobs and

Table 1. Comparison of Scheduling Options

POLICY OR FEATURE	HPC	HADOOP
FIFO	Available	Available
Fair Share	Available	Available
Time-Based Policies	Available	Technology Gap
Preemption	Available	Available
Exclusive Placement	Available	Technology Gap
Custom Algorithms	Available	Available
SLA- or QoS-Based	Available	Technology Gap
Round-Robin	Available	Technology Gap
Static and Dynamic Resources	Available	Available
Node Labeling	Available	Coming Soon
Custom Resources	Available	Technology Gap

report resource availability to the master node dæmons. Technologies from HPC, like Platform LSF and PBS Professional as well as other open-source variants like SLURM and Torque, are commonly seen in HPC.

These technologies are much older than Hadoop, and in terms of scheduling policy, they are more mature. They tend to share some basic tenets of scheduling policy that the Hadoop community is either in the process of addressing or has already.

First-In First-Out Scheduling

Many times this is the default policy used when a workload manager is first installed. As the name suggests, FIFO operates like a line or queue at a movie theatre.

Fair Share

Fair Share is a scheduling policy that attempts to allocate cluster resources fairly to jobs based upon a fixed number of shares per user or group. Fair share is implemented differently based upon the exact cluster resource management software used, but most systems have the concept of ordering jobs to be run in an attempt to even out the use of resources for all users. The specific ordering can be based upon a fixed number of shares or a percentage capacity of resources along with policies for an individual queue or a hierarchy of queues.

Time-Based Policies

Time-based policies come in a few different varieties. Queue-level time-based policies might be used to alter the configuration of a queue based upon time of day including allowing jobs to

Exclusive placement is important when users want to ensure that there is absolutely no contention for resources with other jobs within the selected nodes.

be submitted (enqueued) but not dispatched to nodes. Time-based policies enable concepts like using a cluster for a specific workload during business hours and an alternate workload overnight. Other time-based policies include dedicating the entire cluster or portion of a cluster for a specific use for a length of time. Additionally, draining a cluster of submitted jobs for maintenance windows is common.

Preemption

Preemption is the idea that some jobs can take the place of others that are currently running. Preemption is usually based upon the priority level of the job itself. The preempted job may be simply killed, suspended or possibly just requeued. All of these options come with benefits and disadvantages. Preemption in general tends to cause many internal political challenges but none as much as preemption by killing. Setting submitted high-priority work simply to be the next job to run when resources become available tends to balance the needs of high-priority work without the disruption potentially caused by a kill-style preemption model. An additional alternative would be to automate job requeue of preempted jobs instead of killing them. The best way to do preemption is intimately related to the workload profile.

Exclusive Job Placement

Exclusively placing jobs onto a node is an important job placement policy. Exclusively placing a job on a node means that no subsequent job could be placed on a node once a job is assigned to it. Exclusive placement is important when users want to ensure that there is absolutely no contention for resources with other jobs within the selected nodes. Users might request this type of placement when rendering video or graphics where memory is the rate-limiting factor in total wall time.

Exclusive placement can be enabled on most systems by matching the job resource request to encompass an entire single node. To do this, submitting users have to know specific hardware details of nodes in the cluster, and this approach also assumes node homogeneity. In many cases, users have no knowledge of the exact configuration of nodes, or there may be some level of heterogeneity across nodes in the cluster. Using a resource manager with a language for job submission that includes a client resource request flag to allow exclusive placement of jobs is highly desirable.

Custom Algorithms

Advanced cluster users eventually find that creating their own algorithm for custom job placement becomes required. In practice, these algorithms tend to be highly secret and bound to some proprietary process specific to the owner's vertical line of business. An example of a custom algorithm might include assigning specific jobs an immediate high priority based upon an organizational goal or specific project.

SLA- or QoS-Based Policies

Many times it is difficult to guarantee a job will complete within a required window. Most workload management systems have a direct or indirect way to configure scheduling policy such that jobs are guaranteed to finish within given time constraints. Alternatively, there may be ways to define custom qualities used to build scheduling policy.

Round-Robin Placement

Round-robin placement will take jobs from each queue in a specific order, usually within a single scheduling cycle. The queues are ordered by priority in most systems, but the exact behavior can be tricky depending upon the additional options used (for example, strict ordering in PBS Professional).

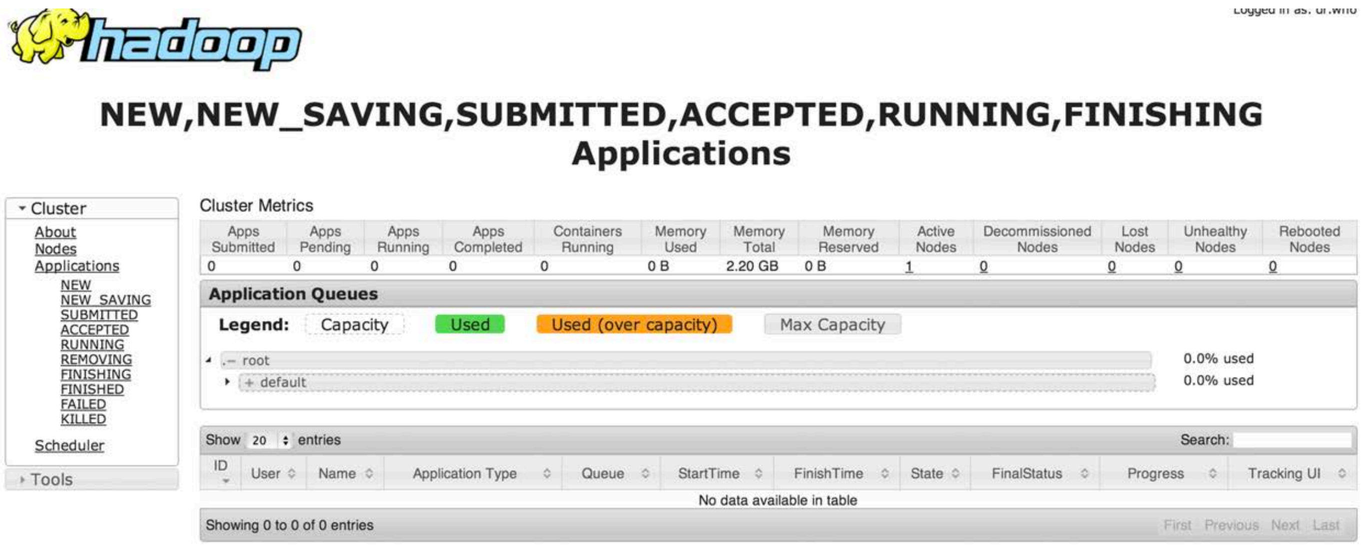
HPC Workload Manager Resource Types

Workload managers use resource request languages to help the scheduler place work on nodes. Many job placement scenarios include the specification of static or built-in resources as well as the ability to use custom-style resources defined using a script. Resource types tend to reflect programming primitives like boolean, numerical and string as well as properties like static and dynamic to reflect the nature of the values. Some of these resource types are assigned specifically to hosts while others have to do with shared resources in a cluster like software licenses or allocation management (cluster use credits or chargebacks). These resources are all important in a multitenant distributed computing environment.

Hadoop Scheduling Policy

Hadoop currently makes use of mainly CPU and memory. There are additional selection criteria one can make when specifying container requests. The Application Master can specify a hostname, rack placement information and priority. Over time, Hadoop will benefit from a more mature resource specification design similar to HPC. One such use case would be a boolean host resource to specify placement of containers onto nodes with specific hardware (for example, a GPU). Even though very robust placement of containers can be accomplished in the Java code of the Application Master, resources requests probably need to be made more generic and available at a higher level (that is, during submission time via a common client). YARN allows for what it calls static resources from the submitting client and dynamic resources as those defined at runtime by the Application Master.

There are two built-in scheduling policies for Hadoop (excluding FIFO) at this time, but scheduling, like



NEW,NEW_SAVING,SUBMITTED,ACCEPTED,RUNNING,FINISHING Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
0	0	0	0	0	0 B	2.20 GB	0 B	1	0	0	0	0

Application Queues

Legend: Capacity Used Used (over capacity) Max Capacity

.– root 0.0% used

+ default 0.0% used

Show 20 entries Search:

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI

No data available in table

Showing 0 to 0 of 0 entries First Previous Next Last

Figure 2. The scheduler page of the Resource Manager Web interface showing queue configuration and data on running applications.

most things in Hadoop, is pluggable. Setting `yarn.resourcemanager.scheduler.class` to the desired class in the configuration yarn-site.xml file can alter the specific scheduling type used. Custom scheduling policy classes can be defined here as well.

Scheduling policy for a Hadoop cluster is easy to access via a Web browser. Simply navigate to http://ResourceManager:port/cluster/scheduler using the Resource Manager hostname or IP and the correct port for the distribution of Hadoop being used.

FIFO

This is the standard first-in first-out method one might expect as a default scheduling option. It operates by accepting jobs and dispatching them in order received.

Capacity Scheduler

Hadoop's Capacity Scheduler was designed to provide minimum levels of resource availability to multiple users on the same cluster (aka multitenancy). Part of the power of Hadoop is having many nodes. The more worker nodes provided in a single cluster, the more resilient it is to failures. In large organizations with independent budgets, individual department heads might think it best to set up individual clusters to obtain resource isolation. Multitenancy can be accomplished logically using the Capacity Scheduler. The benefit of this design is not only better cluster utilization but also the improvement of system stability. Using more nodes decreases the importance of any one node in a node loss scenario by spreading out data as well as increasing cluster data and compute capacity.

The Capacity Scheduler functions through a series of queues. This includes hierarchical queues each with properties associated to direct the sharing of resources. The main resources include memory and CPU at this time. When writing an Application Master, the container requests can include resource requests, such as node resource (memory and CPU), a specific host name, a specific rack and a priority.

The capacity-scheduler.xml file contains the definition of queues and their properties. The settings in this file include capacity and percentage maximums along with total number of jobs allowed to be running at one time. In a multitenant environment, multiple child queues can be created below the root queue. Each queue configuration contains a share of resources to be consumed by itself or shared with its children.

It's also common to see the use of access control lists for users of queues. Each queue in this case would receive a minimum capacity guaranteed by the scheduler. When other queues are below their capacity, another queue can use additional resources up to its configured maximum (hard limit).

Configurable preemption was added in Hadoop 2.1 for the capacity scheduler via ASF JIRA YARN-569. On the other hand, the complete isolation of resources so that no one job (AM or its containers) impedes the progress of another is accomplished in an operating-system-dependent way. Yes, Hadoop has matured so it will even run on Windows. For Linux, resource isolation is done via cgroups (control groups) and on Windows using job control. Future enhancements may even include the use of virtualization technologies, such as XEN and KVM, for resource isolation.

Fair Scheduler

The Fair Scheduler is another pluggable scheduling functionality for Hadoop under YARN. The Capacity and Fair Scheduler operate in a very similar manner although their nomenclature differs. Both systems schedule by memory and CPU; both systems use queues (previously called Pools) and attempt to provide a framework for sharing a common collection of resources. Fair Scheduler uses the concept of a minimum number of shares to enforce a minimum amount of resource availability with excess resources being shared with other queues. There are many similarities but a few nice unique features as well. Scheduling policy itself is customizable by queue and can include three options including FIFO, Fair Share (by memory) and a Dominant Resource Fairness (using both CPU and memory) that does its best to balance the needs of divergent workloads over time.

The yarn-site.xml file can include a number of Fair Scheduler settings including the default queue. A unique setting includes an option to turn on preemption that was previously preemption by killing and now includes a work-saving preemption option. One of the most important options in yarn-site.xml includes the allocation file location. The allocation file details queues, resource allotments as well as a queue-specific scheduling algorithm in XML.

The YARN Scheduler Load Simulator is a convenient tool for investigating options for scheduling via the options available to Hadoop.

YARN Scheduler Load Simulator

How should one choose between the two main options available? More important, how are the configurations tuned for optimal performance? The YARN Scheduler Load Simulator is a convenient tool for investigating options for scheduling via the options available to Hadoop. The simulator works with a real Resource Manager but simulates the Node Manager and Application Masters so that a fully distributed cluster is not required to analyze scheduling policy. One of the new configuration best practices should be possible to include time for scheduler tuning when initially setting up a new Hadoop cluster. This can be followed by analysis of scheduling policy at an interval going forward for continued optimization. Regardless of what type of scheduling is selected or how it is configured, there now is a tool to help each group determine what is best for its needs.

Scheduler simulation is a very technical field of study and something commercial HPC workload offerings have desperately needed for years. It is exciting to see a concrete method for analysis of Hadoop workloads, especially considering the effect a small change can make in throughput and utilization on a distributed system.

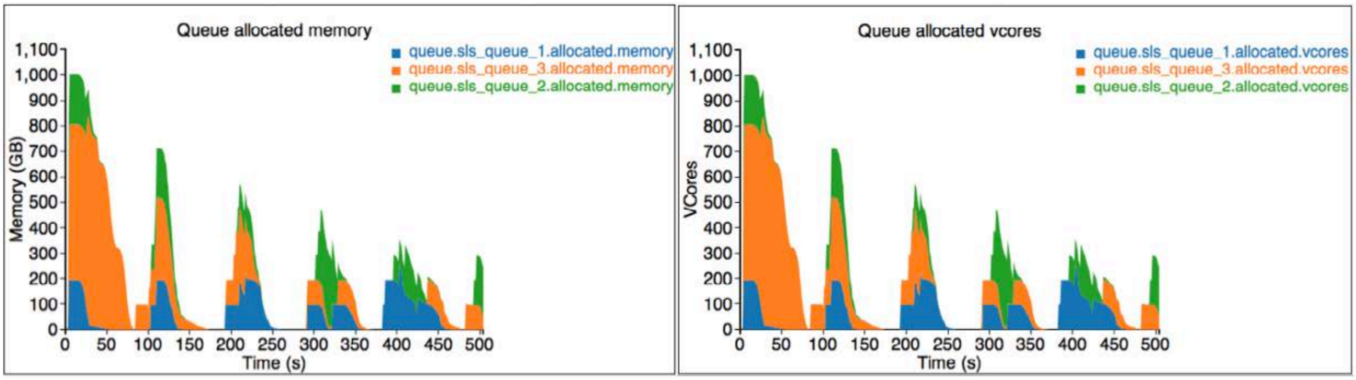


Figure 3. YARN Scheduler Simulator output showing memory and vcores for a queue.

Conclusions

Hadoop workload scheduling, much like the rest of Hadoop, is growing by leaps and bounds. With each release, more resource types and scheduling features become available, and it is exciting to see the convergence of Internet-scale distributed computing with the field of HPC that has been available for many years. One might argue some features from HPC workload management are needed in Hadoop. Examples, such as SLA-based scheduling and time-based policies are important operational examples of policies administrators expect. From a resource perspective, additional resource types also are needed.

The pace at which the open-source model innovates surely will close the gaps very soon. The participation of multiple groups and contributors in a meritocracy-based system drives not only the pace of innovation but quality as well.■

Adam Diaz is a longtime Linux geek and fan of distributed/parallel systems. Adam cut his teeth working for companies like Platform Computing, Altair Engineering and a handful of startups. His current endeavor is with Hortonworks helping companies make use of Hadoop. He can be reached at http://www.techtonka.com.

Resources

Original YARN JIRA: **https://issues.apache.org/jira/browse/MAPREDUCE-279**

Hamster Project: **https://issues.apache.org/jira/browse/MAPREDUCE-2911**

mpich2-yarn: **https://github.com/clarkyzl/mpich2-yarn**

Apache Capacity Scheduler Site: **http://hadoop.apache.org/docs/r2.2.0/hadoop-yarn/hadoop-yarn-site/CapacityScheduler.html**

Capacity Scheduler Preemption: **https://issues.apache.org/jira/browse/YARN-569**

Apache Fair Scheduler Site: **http://hadoop.apache.org/docs/r2.2.0/hadoop-yarn/hadoop-yarn-site/FairScheduler.html**

Work-Saving Preemption: **https://issues.apache.org/jira/browse/YARN-568**

YARN Scheduler Load Simulator: **https://issues.apache.org/jira/browse/YARN-1021**

YARN Scheduler Load Simulator Demo: **http://youtu.be/6thLi8q0qLE**

Linux Containers and the **Future Cloud**

HPC and lightweight virtualization with Linux-based containers.

RAMI ROSEN

Linux-based container infrastructure is an emerging cloud technology based on fast and lightweight process virtualization. It provides its users an environment as close as possible to a standard Linux distribution. As opposed to para-virtualization solutions (Xen) and hardware virtualization solutions (KVM), which provide virtual machines (VMs), containers do not create other instances of the operating system kernel. Due to the fact that containers are more lightweight than VMs, you can achieve higher densities with containers than with VMs on the same host (practically speaking, you can deploy more instances of containers than of VMs on the same host).

Another advantage of containers over VMs is that starting and shutting down a container is much faster than starting and shutting down a VM. All containers under a host are running under the same kernel, as opposed to virtualization solutions like Xen or KVM where each VM runs its own kernel. Sometimes the constraint of running under the same kernel in all containers under a given host can be considered a drawback. Moreover, you cannot run BSD, Solaris, OS/x or Windows in a Linux-based container, and sometimes this fact also can be considered a drawback.

The idea of process-level virtualization in itself is not new, and it already was implemented by Solaris Zones as well as BSD jails quite a few years ago. Other open-source projects implementing process-level virtualization have existed for several years. However, they required custom kernels, which was often a major setback. Full and stable support for Linux-based containers on mainstream kernels by the LXC project is relatively recent, as you will see in this article. This makes containers more attractive for the cloud infrastructure. More and more hosting and cloud services companies are adopting Linux-based container solutions. In this article, I describe some open-source Linux-based container projects and the kernel features they use, and show some usage examples. I also describe the Docker tool for creating LXC containers.

The underlying infrastructure of modern Linux-based containers consists mainly of two kernel features: namespaces and cgroups. There are six types of namespaces, which provide per-process isolation of the following operating system resources: filesystems (MNT), UTS, IPC, PID, network and user namespaces (user namespaces allow mapping of UIDs and GIDs between a user namespace and the global namespace of the host). By using network namespaces, for example, each process can have its own instance of the network stack (network interfaces, sockets, routing tables and routing rules, netfilter rules and so on).

Creating a network namespace is very simple and can be done with the following `iproute` command: `ip netns add myns1`. With the `ip netns command`, it also is easy to move one network interface from one network namespace to another, to monitor the creation and deletion of network namespaces, to find out to which network namespace a specified process belongs and so on. Quite similarly, when using the MNT namespace, when mounting a filesystem, other processes will not see this mount, and when working with PID namespaces, you will see by running the `ps` command from that PID namespace only processes that were created from that PID namespace.

The cgroups subsystem provides resource management and accounting. It lets you define easily, for example, the maximum memory that a process may use. This is done by using cgroups VFS operations. The cgroups project was started by two Google developers, Paul Menage and Rohit Seth, back in 2006, and it initially was called "process containers". Neither namespaces nor cgroups intervene in critical paths of the kernel, and thus they do not incur a high performance penalty, except for the memory cgroup, which can incur significant overhead under some workloads.

Linux-Based Containers

Basically, a container is a Linux process (or several processes) that has special features and that runs in an isolated environment, configured on the host. You might sometimes encounter terms like Virtual Environment (VE) and Virtual Private Server (VPS) for a container.

The features of this container depend on how the container is configured and on which Linux-based container is used, as Linux-based containers are implemented differently in several projects. I mention the most important ones in this article:

- OpenVZ: the origins of the OpenVZ project are in a proprietary server virtualization solution called Virtuozzo, which originally was started by a company called SWsoft, founded in 1997. In 2005, a part of the Virtuozzo product was released as an open-source project, and it was called OpenVZ. Later, in 2008, SWsoft merged with a company called Parallels. OpenVZ is used for providing hosting and cloud services, and it is the basis of the Parallels Cloud Server. Like Virtuozzo, OpenVZ also is based on a modified Linux kernel. In addition, it has command-line tools (primarily `vzctl`) for management of containers, and it makes use of templates to create containers for various Linux distributions. OpenVZ also can run on some unmodified kernels, but with a reduced feature set. The OpenVZ project is intended to be fully mainlined in the future, but that could take quite a long time.

- Google containers: in 2013, Google released the open-source version of its container stack, lmctfy (which stands for Let Me Contain That For You). Right now, it's still in the beta stage. The lmctfy project is based on using cgroups. Currently, Google containers do not use the kernel namespaces feature, which is used by other Linux-based container projects, but using this feature is on the Google container project roadmap.

- Linux-VServer: an open-source project that was first publicly released in 2001, it provides a way to partition resources securely on a host. The host should run a modified kernel.

- LXC: the LXC (LinuX Containers) project provides a set of userspace tools and utilities to manage Linux containers. Many LXC contributors are from the OpenVZ team. As opposed to OpenVZ, it runs on an unmodified kernel. LXC is fully written in userspace and supports bindings in other programming languages like Python, Lua and Go. It is available in most popular distributions, such as Fedora, Ubuntu, Debian and more. Red Hat Enterprise Linux 6 (RHEL 6) introduced Linux containers as a technical preview. You can run Linux containers on architectures other than x86, such as ARM (there are several how-tos on the Web for running containers on Raspberry PI, for example).

I also should mention the libvirt-lxc driver, with which you can manage containers. This is done by defining an XML configuration file and then running `virsh start`, `virsh console` and `visrh destroy` to run, access and destroy the container, respectively. Note that there is no common code between libvirt-lxc and the userspace LXC project.

LXC Container Management

First, you should verify that your

host supports LXC by running `lxc-checkconfig`. If everything is okay, you can create a container by using one of several ready-made templates for creating containers. In lxc-0.9, there are 11 such templates, mostly for popular Linux distributions. You easily can tailor these templates according to your requirements, if needed. So, for example, you can create a Fedora container called fedoraCT with:

```
lxc-create -t fedora -n fedoraCT
```

The container will be created by default under /var/lib/lxc/fedoraCT. You can set a different path for the generated container by adding the `--lxcpath PATH` option.

The `-t` option specifies the name of the template to be used, (`fedora` in this case), and the `-n` option specifies the name of the container (`fedoraCT` in this case). Note that you also can create containers of other distributions on Fedora, for example of Ubuntu (you need the `debootstrap` package for it). Not all combinations are guaranteed to work.

You can pass parameters to `lxc-create` after adding `--`. For example, you can create an older release of several distributions with the `-R` or `-r` option, depending on the distribution template. To create an older Fedora container on a host running Fedora 20, you can run:

```
lxc-create -t fedora -n fedora19 -- -R 19
```

You can remove the installation of an LXC container from the filesystem with:

```
lxc-destroy -n fedoraCT
```

For most templates, when a template is used for the first time, several required package files are downloaded and cached on disk under /var/cache/lxc. These files are used when creating a new container with that same template, and as a result, creating a container that uses the same template will be faster next time.

You can start the container you created with:

```
lxc-start -n fedoraCT
```

And stop it with:

```
lxc-stop -n fedoraCT
```

The signal used by `lxc-stop` is SIGPWR by default. In order to use SIGKILL in the earlier example, you should add `-k` to `lxc-stop`:

```
lxc-stop -n fedoraCT -k
```

You also can start a container as a dæmon by adding `-d`, and then log on into it with `lxc-console`, like this:

```
lxc-start -d -n fedoraCT
lxc-console -n fedoraCT
```

The first `lxc-console` that you run for a given container will connect you to tty1. If tty1 already is in use (because that's the second lxc-console that you run for that container), you will be connected to tty2 and so on. Keep in mind that the maximum number of ttys is configured by the `lxc.tty` entry in the container configuration file.

You can make a snapshot of a non-running container with:

```
lxc-snapshot -n fedoraCT
```

This will create a snapshot under /var/lib/lxcsnaps/fedoraCT. The first snapshot you create will be called `snap0`; the second one will be called `snap1` and so on. You can time-restore the snapshot at a later time with the `-r` option—for example:

```
lxc-snapshot -n fedoraCT -r snap0 restoredFdoraCT
```

You can list the snapshots with:

```
lxc-snapshot -L -n fedoraCT
```

You can display the running containers by running:

```
lxc-ls --active
```

Managing containers also can be done via scripts, using scripting languages. For example, this short Python script starts the fedoraCT container:

```
#!/usr/bin/python3

import lxc

container = lxc.Container("fedoraCT")
container.start()
```

Container Configuration

A default config file is generated for every newly created container. This config file is created, by default, in /var/lib/lxc/<containerName>/config, but you can alter that using the `--lxcpath PATH` option. You can configure various container parameters, such as network parameters, cgroups parameters, device parameters and more. Here are some examples of popular configuration items for the container config file:

- You can set various cgroups parameters by setting values to the lxc.cgroup.[subsystem name] entries in the config file. The subsystem name is the name of the cgroup controller. For example, configuring the maximum memory a container can use to be 256MB is done by setting lxc.cgroup.memory.limit_in_bytes to be 256MB.

- You can configure the container hostname by setting lxc.utsname.

- There are five types of network interfaces that you can set with the lxc.network.type parameter: empty, veth, vlan, macvlan and phys. Using veth is very common in order to be able to connect a container to the outside world. By using phys, you can move network interfaces from the host network namespace to the container network namespace.

- There are features that can be used for hardening the security of LXC containers. You can avoid some specified system calls from being called from within a container by setting a secure computing mode, or seccomp, policy with the lxc.seccomp entry in the configuration file. You also can remove capabilities from a container with the lxc.cap.drop entry. For example, setting lxc.cap.drop = sys_module will create a container without the CAP_SYS_MDOULE capability. Trying to run insmod from inside this container will fail. You also can define Apparmor and SELinux profiles for your container. You can find examples in the LXC README and in man 5 lxc.conf.

Docker

Docker is an open-source project that automates the creation and deployment of containers. Docker first was released in March 2013 with Apache License Version 2.0. It started as an internal project by a Platform-as-a-Service (PaaS) company called dotCloud at the time, and now called Docker Inc. The initial prototype was written in Python; later the whole project was rewritten in Go, a programming language that was developed first at Google. In September 2013, Red Hat announced that it will collaborate with Docker Inc. for Red Hat Enterprise Linux and for the Red Hat OpenShift platform. Docker requires Linux kernel 3.8 (or above). On RHEL systems, Docker runs on the 2.6.32 kernel, as necessary patches have been backported.

Docker utilizes the LXC toolkit and as such is currently available only for

Linux. It runs on distributions like Ubuntu 12.04, 13.04; Fedora 19 and 20; RHEL 6.5 and above; and on cloud platforms like Amazon EC2, Google Compute Engine and Rackspace.

Docker images can be stored on a public repository and can be downloaded with the `docker pull` command—for example, `docker pull ubuntu` or `docker pull busybox`.

To display the images available on your host, you can use the `docker images` command. You can narrow the command for a specific type of images (fedora, for example) with `docker images fedora`.

On Fedora, running a Fedora docker container is simple; after installing the `docker-io package`, you simply start the docker dæmon with `systemctl start docker`, and then you can start a Fedora docker container with `docker run -i -t fedora /bin/bash`.

Docker has git-like capabilities for handling containers. Changes you make in a container are lost if you destroy the container, unless you commit your changes (much like you do in git) with `docker commit <containerId> <containerName/containerTag>`. These images can be uploaded to a public registry, and they are available for downloading by anyone who wants to download them. Alternatively, you can set a private Docker repository.

Docker is able to create a snapshot using the kernel device mapper feature. In earlier versions, before Docker version 0.7, it was done using AUFS (union filesystem). Docker 0.7 adds "storage plugins", so people can switch between device mapper and AUFS (if their kernel supports it), so that Docker can run on RHEL releases that do not support AUFS.

You can create images by running commands manually and committing the resulting container, but you also can describe them with a Dockerfile. Just like a Makefile will compile code into a binary executable, a Dockerfile will build a ready-to-run container image from simple instructions. The command to build an image from a Dockerfile is `docker build`. There is a tutorial about Dockerfiles and their command syntax on the Docker Web site. For example, the following short Dockerfile is for installing the `iperf` package for a Fedora image:

```
FROM fedora
MAINTAINER Rami Rosen
RUN yum install -y iperf
```

You can upload and store your images for free on the Docker public index. Just like with GitHub, storing public images is free and just requires you to register an account.

The Checkpoint/Restore Feature

The CRIU (checkpoint/restore in userspace) project is implemented mostly in userspace, and there are more than 100 little patches scattered in the kernel for supporting it. There were several attempts to implement Checkpoint/Restore in kernel space solely, some of them by the OpenVZ project. The kernel community rejected all of them though, as they were too complex.

The Checkpoint/Restore feature enables saving a process state in several image files and restoring this process from the point at which it was frozen, on the same host or on a different host at a later time. This process also can be an LXC container. The image files are created using Google's protocol buffer (PB) format. The Checkpoint/Restore feature enables performing maintenance tasks, such as upgrading a kernel or hardware maintenance on that host after checkpointing its applications to persistent storage. Later on, the applications are restored on that host.

Another feature that is very important in HPC is load balancing using live migration. The Checkpoint/Restore feature also can be used for creating incremental snapshots, which can be used after a crash occurs. As mentioned earlier, some kernel patches were needed for supporting CRIU; here are some of them:

- A new system call named `kcmp()` was added; it compares two processes to determine if they share a kernel resource.
- A socket monitoring interface called `sock_diag` was added to UNIX sockets in order to be able to find the peer of a UNIX domain socket. Before this change, the `ss` tool, which relied on parsing of `/proc` entries, did not show this information.
- A TCP connection repair mode was added.
- A `procfs` entry was added (/proc/PID/map_files).

Let's look at a simple example of using the `criu` tool. First, you should check whether your kernel supports Checkpoint/Restore, by running `criu check --ms`. Look for a response that says `"Looks good."`

Basically, checkpointing is done by:

```
criu dump -t <pid>
```

You can specify a folder where the process state files will be saved by adding `-D folderName`.

You can restore with `criu restore <pid>`.

Summary

In this article, I've described what Linux-based containers are, and I briefly explained the underlying cgroups and namespaces kernel features. I have discussed some Linux-based container projects, focusing on the promising and popular LXC project. I also looked at the LXC-based Docker engine, which provides an easy and convenient way to create and deploy LXC containers. Several hands-on examples showed how simple it is to configure, manage and deploy LXC containers with the userspace LXC tools and the Docker tools.

Due to the advantages of the LXC and the Docker open-source projects, and due to the convenient and simple tools to create, deploy and configure LXC containers, as described in this article, we presumably will see more and more cloud infrastructures that will integrate LXC containers instead of using virtual machines in the near future. However, as explained in this article, solutions like Xen or KVM have several advantages over Linux-based containers and still are needed, so they probably will not disappear from the cloud infrastructure in the next few years.

Send comments or feedback via http://www.linuxjournal.com/contact or to ljeditor@linuxjournal.com.

Acknowledgements

Thanks to Jérôme Petazzoni from Docker Inc. and to Michael H. Warfield for reviewing this article.■

Resources

Google Containers: https://github.com/google/lmctfy

OpenVZ: http://openvz.org/Main_Page

Linux-VServer: http://linux-vserver.org

LXC: http://linuxcontainers.org

libvirt-lxc: http://libvirt.org/drvlxc.html

Docker: https://www.docker.io

Docker Public Registry: https://index.docker.io

Rami Rosen is a kernel developer and the author of *Linux Kernel Networking: Implementation and Theory*, Apress, 648 pages, 2013. From time to time, he gives voluntary lectures for Israeli LUGs and writes articles about Linux. You can visit his homepage at http://ramirose.wix.com/ramirosen.

WEBCASTS

Learn the 5 Critical Success Factors to Accelerate IT Service Delivery in a Cloud-Enabled Data Center

Today's organizations face an unparalleled rate of change. Cloud-enabled data centers are increasingly seen as a way to accelerate IT service delivery and increase utilization of resources while reducing operating expenses. Building a cloud starts with virtualizing your IT environment, but an end-to-end cloud orchestration solution is key to optimizing the cloud to drive real productivity gains.

> **http://lnxjr.nl/IBM5factors**

Modernizing SAP Environments with Minimum Risk—a Path to Big Data

Sponsor: SAP | Topic: Big Data

Is the data explosion in today's world a liability or a competitive advantage for your business? Exploiting massive amounts of data to make sound business decisions is a business imperative for success and a high priority for many firms. With rapid advances in x86 processing power and storage, enterprise application and database workloads are increasingly being moved from UNIX to Linux as part of IT modernization efforts. Modernizing application environments has numerous TCO and ROI benefits but the transformation needs to be managed carefully and performed with minimal downtime. Join this webinar to hear from top IDC analyst, Richard Villars, about the path you can start taking now to enable your organization to get the benefits of turning data into actionable insights with exciting x86 technology.

> **http://lnxjr.nl/modsap**

WHITE PAPERS

White Paper: JBoss Enterprise Application Platform for OpenShift Enterprise

Sponsor: DLT Solutions

Red Hat's® JBoss Enterprise Application Platform for OpenShift Enterprise offering provides IT organizations with a simple and straightforward way to deploy and manage Java applications. This optional OpenShift Enterprise component further extends the developer and manageability benefits inherent in JBoss Enterprise Application Platform for on-premise cloud environments.

Unlike other multi-product offerings, this is not a bundling of two separate products. JBoss Enterprise Middleware has been hosted on the OpenShift public offering for more than 18 months. And many capabilities and features of JBoss Enterprise Application Platform 6 and JBoss Developer Studio 5 (which is also included in this offering) are based upon that experience.

This real-world understanding of how application servers operate and function in cloud environments is now available in this single on-premise offering, JBoss Enterprise Application Platform for OpenShift Enterprise, for enterprises looking for cloud benefits within their own datacenters.

> **http://lnxjr.nl/jbossapp**

WHITE PAPERS

Linux Management with Red Hat Satellite: Measuring Business Impact and ROI

Sponsor: Red Hat | Topic: Linux Management

Linux has become a key foundation for supporting today's rapidly growing IT environments. Linux is being used to deploy business applications and databases, trading on its reputation as a low-cost operating environment. For many IT organizations, Linux is a mainstay for deploying Web servers and has evolved from handling basic file, print, and utility workloads to running mission-critical applications and databases, physically, virtually, and in the cloud. As Linux grows in importance in terms of value to the business, managing Linux environments to high standards of service quality — availability, security, and performance — becomes an essential requirement for business success.

> **http://lnxjr.nl/RHS-ROI**

Standardized Operating Environments for IT Efficiency

Sponsor: Red Hat

The Red Hat® Standard Operating Environment SOE helps you define, deploy, and maintain Red Hat Enterprise Linux® and third-party applications as an SOE. The SOE is fully aligned with your requirements as an effective and managed process, and fully integrated with your IT environment and processes.

Benefits of an SOE:

SOE is a specification for a tested, standard selection of computer hardware, software, and their configuration for use on computers within an organization. The modular nature of the Red Hat SOE lets you select the most appropriate solutions to address your business' IT needs.

SOE leads to:

- Dramatically reduced deployment time.
- Software deployed and configured in a standardized manner.
- Simplified maintenance due to standardization.
- Increased stability and reduced support and management costs.
- There are many benefits to having an SOE within larger environments, such as:
  - Less total cost of ownership (TCO) for the IT environment.
  - More effective support.
  - Faster deployment times.
  - Standardization.

> **http://lnxjr.nl/RH-SOE**

Using MySQL's Built-in Replication

See the benefits MySQL replication can bring to your environment.

BRIAN TRAPP

My first Linux-based project at work was a Web-based LAMP (Linux/Apache/MySQL/Perl) stack that provided engineering reports and analysis for my department. Over time, that evolved into Tomcat/MySQL/Java (which doesn't acronym nicely at all) and moved past my department to the whole facility, and eventually to several different manufacturing locations. As my user count increased, so grew the expectation of 24x7 uptime, and I needed to get serious about redundancy for my single database instance.

MySQL's built-in replication option seemed like a logical choice, as it seemed easy to configure and would provide an additional level of redundancy. I was pleasantly surprised at how easy it was to enable replication, and after two years of use in production, my only regret is that I didn't implement it sooner.

This article is written based on the 5.5 version of MySQL, but it also will work for 5.6. Although 5.6 reached "generally available" status in February 2013, there are still a *lot* of 5.5 installations out there in production.

What Is Replication?

MySQL's replication is a way for one or more slave servers to stay in sync with a single master server. At a high level, this is accomplished by having the master log all database changes to a local file. Each slave knows its master log filename and position

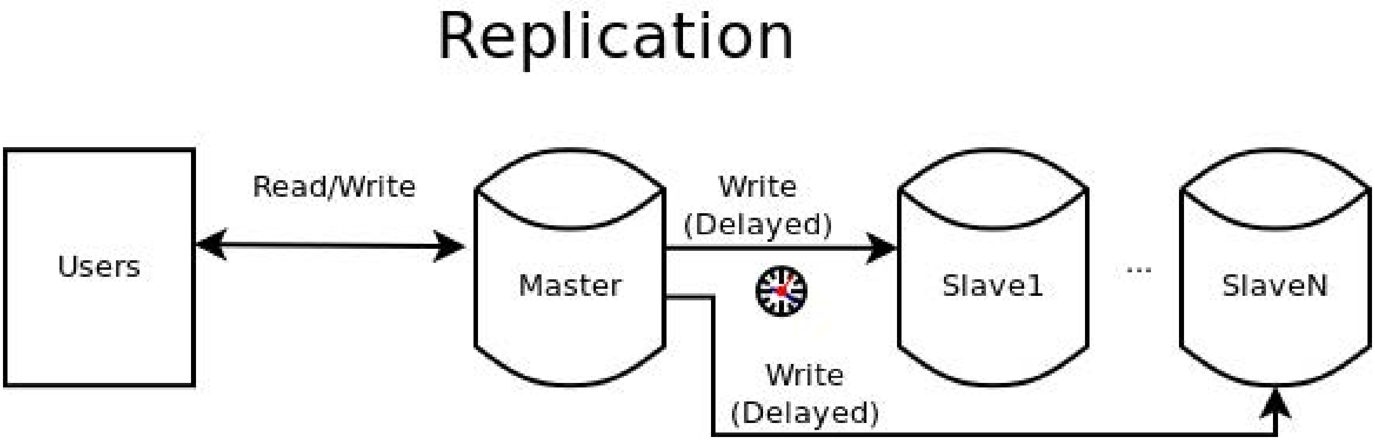


Figure 1. Example Replication Flow

within that file, and routinely queries the master for any new changes. As changes are found, the slave modifies its copy of the database and moves its master log file pointer accordingly. This means the slaves will process changes in the same order as executed on the master, although each slave may be at a slightly different position in the change log.

Replication is quite different from clustering. Replication can be enabled in the standard MySQL server installation ("MySQL Community Edition") by enabling a few control file options, but clustering requires installing an entirely different product ("MySQL Cluster"). In clustering, a set of machines synchronize reads and writes across all the machines in the cluster—all machines in the cluster are essentially identical. In replication, all writes occur solely on the master. Reads can be performed on any machine, with the understanding that a slave may be slightly behind the master. In my experience, slave machine delays are typically less than a second.

Benefits of Replication

Impact-Free Database Backups: Replication alone is *not* a backup strategy. From personal experience, I can assure you that silly SQL mistakes on the master will propagate to your slaves faster than you ever thought possible, but replication *can* ensure that the process of making backups won't affect your master server. Creating a quality database backup is a resource-intensive operation, and it usually requires a write lock for the duration of the backup in order to ensure consistency. Prior to configuring replication on our system, our production server would use MySQL's mysqldump tool to

With the rising popularity of data-mining techniques, it's entirely possible that you may want to start using your operational, near real-time database to do some hefty batch reporting.

create backups at 6am and 6pm. This worked well at first, but as users requested more types of data and longer retention windows, the amount of backup time passed the point where users would accept write interruption. The increased I/O load during backups also would degrade normal application performance, further annoying our users.

After introducing clustering, the entire backup process can be moved onto one of the slaves. The slave can be write-locked for the whole backup process with no write or I/O impact to the master. Once the slave is finished creating the backup, it releases the write lock and quickly catches back up to the master.

Batch Reporting and Load Balancing: With the rising popularity of data-mining techniques, it's entirely possible that you may want to start using your operational, near real-time database to do some hefty batch reporting. Though these types of reports typically don't require write locks during processing, enough of them can degrade performance while they're crunching data. Prior to implementing replication, we manually scheduled batch jobs to run at traditionally low usage time slots, but this was error-prone and became significantly more complex as we started supporting users from other time zones.

After implementing replication, we have offloaded all of these batch reports to a data-mining server with its own MySQL slave instance. Batch reports and data mining now can be scheduled at the user's convenience, not just when the server is idle.

Direct End-User Access: One of the downsides of making a really awesome database is that eventually other folks will want to get their fingers into that database too. Each additional user increases the base system load and the risk that their queries may misbehave. You could develop a bunch of Web services and insist that they use those instead of directly querying the database, but that is a lot of extra work, even more maintenance and slows down innovation. On the other hand, giving even the most experienced users direct access could lead to SQL mistakes that stress the database. MySQL has a nice facility for limiting user queries with resource limits, but even that doesn't guarantee zero impacts to your coveted production database.

After replication, you can offload this read-only traffic to one of your slaves. It's even possible to let the end users create their own MySQL with very little input required on your end.

Hot Failover Capability: Once you have replication configured, you can start to leverage the hot failover capabilities built in to MySQL's Connector/J JDBC driver. With a properly structured URL, the JDBC infrastructure automatically can sense that the master database is down and failover to the slave in read-only mode. Once the master database comes back up, connections automatically will start using the master again—no application outage required. For applications where the bulk of operations are reads, this can be a real lifesaver! Enabling this failover option is incredibly simple, as shown in the following JDBC URL example: jdbc:mysql://master.mydomain.org,slave1.mydomain.org:3306/MYDATABASE.

Although the default options probably are sufficient for most cases, there are plenty of more advanced options for fine-tuning. See the Connector/J documentation for more information.

Configuring Replication

Hopefully, you're sold on the benefits that replication can provide and are ready to start.

Replication Types: The first important configuration decision is choosing between the three main replication strategies offered by MySQL: statement-based (SBR), row-based (RBR) or mixed-based replication (MBR). In MySQL 5.5, SBR is the default.

At a high level, you can think of statement-based replication as simply copying any SQL statements that modify content on the master (insert, update, delete and so on) to a log file. For example, if the master sees `DELETE FROM ME.FOO WHERE BAR=1`, it puts that SQL in the replication command logs. This logging style has been in use the longest and usually will result in smaller log files than RBR. It is possible, however, to write non-deterministic SQL that will cause RBR to fail. Some obvious examples would be updating a field to a random number (`UPDATE ME.FOO SET BAR=RAND();`) or deleting the first X rows of a table with no order specified.

Row-based logging takes a very different approach. Instead of logging the raw SQL, the master instead will log changes to specific rows. For example, an `UPDATE ME.FOO SET BAR=1 WHERE BAR>5;` on the master actually would be logged as many different unique row-level change events (imagine commands like: `set BAR=1 where ROW_ID=4`, `set BAR=1 where ROW_ID=14` and so on). This technique protects you from ambiguous SQL like the RAND() and unordered delete examples above but at the cost of larger log files.

In mixed-based mode, the server will switch between statement or row-based logging based on the type of statement. For example, `DELETE FROM FOO` is best done as an SBR type command, while row-level updates or non-deterministic statements would be more efficient as RBRs.

This article has more of the gory details: **http://dev.mysql.com/doc/refman/5.5/en/replication-sbr-rbr.html** (and some additional pros and cons). I chose SBR because it's easier to understand; I didn't have any of the weird conditions where SBR doesn't work well, and the smaller log size was appealing.

Master Configuration: If you're setting up MySQL from scratch, install the rpm or debs as usual. (Be sure to run `mysql_secure_installation` to make your installation a bit more secure. While writing this article, I used the Employees test database and creation script from this url: **https://launchpad.net/test-db**. The employees-db-full file has both the data and the database creation scripts. Don't forget to uncomment the bind-address line to make the database server listen to more than just localhost.)

Adding replication to the master is pretty basic: the only required entries are the server-id (a unique, nonzero integer—1 is a good choice) and the binary log name pattern. I've included two lines to help auto-expire old replication logs and two additional lines suggested for the InnoDB table type.

Here's an example of configuring the

master's [mysqld] section of my.cnf:

```
[mysqld]
...
#Assign a unique server-id non-zero integer to each machine.
server-id=1
log-bin=mysql-bin.log              #Enables the log
#Log size/time limits
expire_logs_days = 14
max_binlog_size = 100M
#These next 2 lines are recommended for InnoDB tables
#(now the default)
innodb_flush_log_at_trx_commit=1
sync_binlog=1
```

It's probably a good idea to have replication use a dedicated ID and password, so create an ID and give it "REPLICATION SLAVE" authority:

```
you@master:~$ mysql -u $SQLID -p
mysql> CREATE USER 'replid'@'%.mydomain.org' IDENTIFIED BY 'replpw';
mysql> GRANT REPLICATION SLAVE ON *.* TO 'replid'@'%.mydomain.org';
mysql> FLUSH PRIVILEGES;
```

That's it! Once configured, restart the master, and you should start to see mysql-bin.### files being created as changes are made to the tables.

Creating a Master Snapshot: If you have existing data on the master (a likely situation), you'll need to make a one-time snapshot for a known master log file and log file position, so that the slaves will have a precise point from which to start.

Warning: this will lock all writes while mysqldump is creating the backup, so perform this step during an acceptable maintenance window!

Step 1: open up a MySQL client connection to the master, write-lock the database, and get the master log position. Leave this client connection open until the backup is complete:

```
mysql> FLUSH TABLES WITH READ LOCK;
mysql> SHOW MASTER STATUS;
+------------------+----------+--------------+------------------+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+------------------+----------+--------------+------------------+
| mysql-bin.000274 | 43891079 |              |                  |
+------------------+----------+--------------+------------------+
```

Note both the file and position somewhere safe, and leave this client connection open until the database dump is complete.

Step 2: from a different terminal on the master, use mysqldump to create a database backup:

```
you@master~:$ time mysqldump -u $SQLID -p --events
➥--all-databases --master-data > masterdump.sql
```

The `--master-data` option automatically records the master log file and position from the master and will set these values on the slave when imported. If you create a snapshot without this option, you'll need to record the master filename and position via `SHOW MASTER STATUS` in the MySQL client connection from step 1. The `--events` flag is needed to suppress a relatively new warning in MySQL 5.5.30 and up.

Step 3: once the database backup is complete, exit the MySQL client connection from step 1 to release the read lock.

Slave Configuration: Perform a normal MySQL installation on the slave machine, but add the following to the mysqld section of my.cnf:

```
[mysqld]
...
server-id=2 #Use a UNIQUE ID for each slave. Don't repeat server-ids
```

Restart the MySQL server on the slave, then open a MySQL client connection to the slave and set the master hostname, user ID and password:

```
you@slave1:~$ mysql -u $SQLID -p
mysql> CHANGE MASTER TO MASTER_HOST='master.mydomain.org';
mysql> CHANGE MASTER TO MASTER_USER='replid';
mysql> CHANGE MASTER TO MASTER_PASSWORD='replpw';
```

Importing the Master Snapshot: If you have data to import from the master, you either can copy the SQL dump to the slave and import it there:

```
you@slave1:~$ mysql --user=$SQLID -p < masterdump.sql
```

Or, if your user ID has remote access, you could load it over the network, saving a file copy step:

```
you@master~$ mysql --user=$SQLID -p -h slave1 < masterdump.sql
```

This will take a while. If you haven't used mysqldump before, importing a backup into a new database takes significantly longer than creating the backup in the first place. For my database, creating a backup takes around ten minutes, while creating a new database from that same backup takes about 75 minutes.

Starting Replication: Once the master and each slave is configured, and you've imported any existing data, open a MySQL client connection to each slave and start replication:

```
mysql> START SLAVE;
mysql> SHOW SLAVE STATUS \G
*************************** 1. row ***************************
               Slave_IO_State: Waiting for master to send event
                  Master_Host: master.mydomain.org
                  Master_User: replid
                  Master_Port: 3306
                Connect_Retry: 60
```

```
              Master_Log_File: mysql-bin.000274
          Read_Master_Log_Pos: 44915942
               Relay_Log_File: slavename-relay-bin.000263
                Relay_Log_Pos: 44916088
        Relay_Master_Log_File: mysql-bin.000274
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
...snip...
        Seconds_Behind_Master: 0
...snip...
             Master_Server_Id: 1
1 row in set (0.00 sec)
```

If `Master_log_file` or `Read_Master_Log_Pos` is not specified, you may have forgotten the `--master-data` option to `mysqldump`. The `Read_Master_Log_Pos` should match or be larger than the position you recorded when creating the initial dump.

The most important lines above are:

- `Slave_IO_Running: Yes`
- `Slave_SQL_Running: Yes`
- `Seconds_Behind_Master: 0`

If either `Slave_IO` or `Slave_SQL` are `No`, something went wrong with replication. Check the MySQL logs. `Seconds_Behind_Master` can fluctuate, but in my experience, it's almost always zero once you reach steady state.

Replication Tips and Tricks

- **Temporary tables:** temporary tables created using the `TEMPORARY` keyword are excluded from replication automatically. If your application has pseudo-temporary tables that you don't want replicated, you'll have to use a configuration option to exclude them by a table name pattern. As an example, my application has many large user datasets that are just inputs to data-mining queries. Although they need to stay around longer than the length of just one session, they don't need to be present in any backups or even for hot failover. To exclude tables like TMPTBL_1, TMPTBL_FOO from replication, update the master's my.cnf file with the following option: `replicate-wild-ignore-table=MYSCHEMA.TMPTBL%`.

- **Don't modify the slaves:** if you alter the slave tables in a way that would make a subsequent SQL command on the master fail on the slave, replication will *stop*. For example, if the master gets the SQL statement `DROP TABLE MYSCHEMA.FOO` and table FOO doesn't exist on the slave, replication will fail. In my two years of using replication, this type of self-inflicted error has been the only thing I've seen that breaks replication.

- **Monitoring the slaves:** write a script to monitor the response of `SHOW SLAVE STATUS` on each of your slaves, checking to make sure `Slave_IO_Running` and `Slave_SQL_Running` are both `Yes` and that `Seconds_Behind_Master` is something you're comfortable with.

- **Monitoring the master:** Connector/J's hot failover capability is a great feature. Don't forget to disable that for any monitoring scripts though. At my first planned outage, all my database health-monitoring scripts happily failed over to the slave machine!

What's Coming Next?

Although this article was written using MySQL 5.5, version 5.6.10 reached generally available (GA) status in February 2013. Version 5.6 brings some interesting new choices, including Delayed Replication: a slave can now be told to stay a deliberate amount of time behind the master. This could be interesting and provide a chance to recover quickly from an accidental delete or update. It also includes Global Transaction Identifiers (GTIDs): GTIDs would eliminate the need to tell each slave the master log file and log file position. Instead, the server assigns each modifying statement a time-ordered ID so the slave just needs to know the ID of the last transaction it processed. This should simplify slave setup and snapshot synchronization further.

Conclusion

MySQL enterprise-level replication features are easy to configure and can help solve a wide range of performance, reliability and data security problems. Although each application's database needs are different, I hope that covering the benefits I've seen will encourage you to give it a try on a database of your own.■

Brian Trapp serves up a spicy gumbo of Web-based yield reporting and analysis tools for hungry semiconductor engineers at one of the leading semiconductor research and development consortiums. His signature dish has a Java base with a dash of JavaScript, Perl, Bash and R, and his kitchen has been powered by Linux ever since 1998. He works from home in Buffalo, New York, which is a shame only because that doesn't really fit the whole chef metaphor.

Register and Save $100 on Conference, only $295. $395 Onsite.

11th Annual

2014 High Performance Computing Linux for Wall Street Show and Conference

April 7, 2014 (Monday) **Roosevelt Hotel, NYC**
Madison Ave and 45th St, next to Grand Central Station

HPC, Data Centers, Networks, Low Latency, Big Data, Cloud, Optimization, Linux, at 2014 HPC, April 7, Roosevelt, NYC. Wall Street will be coming to see these systems live at the show.

Wall Street and New York will be coming to the 11th Annual Wall Street IT marketplace at one time and one place in New York, April 7, Monday.

Register and attend this major IT event covering HPC, Data Centers, Networks, Switches, Low Latency, Big Data, Cloud, Optimization, Linux, Cost Savings, and Operational Efficiencies.

Our Show is an efficient one-day showcase and networking opportunity.

Register in advance and save $100. Includes general sessions, drill-down sessions, industry luncheon, exclusive show viewing times, post-show receptions. In advance: $295. On-site: $395.

Don't have time for the full Conference? Attend the free Show. Register in advance at: www.flaggmgmt.com/linux

Wall Street IT speakers and Gold Sponsors will lead drill-down sessions in the Grand Ballroom of the convenient Roosevelt Hotel.

Show Hours: Mon, April 7		8:00 - 4:00
Conference Hours: Mon, April 7		8:30 - 4:50

Leading vendors will be introducing their latest financial services systems at this One-Day Conference for Wall Street registrants.

2014 Sponsors

CISCO · IBM · redhat · hp · ARISTA · SAP · DataDirect NETWORKS · SanDisk

LINUX NEW MEDIA The Pulse of Open Source · LINUX PRO MAGAZINE · ADMIN Network & Security · TABOR COMMUNICATIONS · MARKETSMEDIA

LINUX JOURNAL · HPC In the Cloud · datanami · HPC wire · SourceMedia · scientific computing world

Show & Conference: Flagg Management Inc
353 Lexington Avenue, New York 10016
(212) 286 0333 fax: (212) 286 0086
flaggmgmt@msn.com

www.flaggmgmt.com/linux

DOC SEARLS

Opening Minds to the Spheres Among Us

Linux can't be understood in terms of hierarchy. Neither can the Internet. That's because both are examples of heterarchy at work.

Flatland, an 1880 novella by Edwin A. Abbott, is about a world with just two dimensions, inhabited by lines and polygons (**http://en.wikipedia.org/wiki/Flatland**). Trouble starts when a sphere shows up.

For business, the same kind of trouble started when Linux and the Internet showed up in the mid-1990s. No matter how useful Linux and the Internet prove to be, business still has trouble getting its head around a virtual world composed of end points that are all autonomous, self-empowered and at zero functional distance from each other. The best geometric figure for that world is a giant hollow sphere composed of boundless smarts on its outside—the nodes of its network—and no controlling entity in the middle.

Business can't see that sphere when its head stays inside the flat triangle we call hierarchy (**http://www.merriam-webster.com/dictionary/hierarchy**). Even complex hierarchies, such as those of giant corporations, governments and military organizations, are depicted as two-dimensional and roughly triangular org charts (**http://en.wikipedia.org/wiki/Organizational_chart**), all flat

No matter how useful Linux and the Internet prove to be, business still has trouble getting its head around a virtual world composed of end points that are all autonomous, self-empowered and at zero functional distance from each other.

as a whiteboard.

The word *hierarchy* is older than English. Born as the Greek *hierarches* ("leader of sacred rites", **http://en.wikipedia.org/wiki/Hierarch**), it evolved into Latin (*hierarchia*, **http://en.wiktionary.org/wiki/hierarchia#Latin**) and then through French into the Middle English *jerarchie* before arriving at its present English spelling in the 14th century. The current (February 7, 2014) edit of Wikipedia's hierarchy article adds this technical jive to the dictionary definition (**http://en.wikipedia.org/w/index.php?title=Hierarchy&oldid=594383135**):

> Abstractly, a hierarchy can be modeled mathematically (**http://en.wikipedia.org/wiki/Mathematical_model**) as a rooted tree (**http://en.wikipedia.org/wiki/Rooted_tree#rooted_tree**): the root of the tree forms the top level, and the children of a given vertex are at the same level, below their common parent. However, a rooted tree does not allow for items to be "at the same level as" one another, since a tree prohibits cycles. To accommodate this, a hierarchy can be modeled using a graph (**http://en.wikipedia.org/wiki/Graph_(mathematics)**) or a pre-order relation (**http://en.wikipedia.org/wiki/Preorder**) on the set of items. Alternatively, items of like type can be grouped together, and the hierarchy can be modeled using a partial order relation (**http://en.wikipedia.org/wiki/Partial_order#Formal_definition**) on the set of sets-of-like-items.

Got root? If so, you're at the top of a hierarchy like the one described in that paragraph, and possess power that fans down through directory paths. But your power is over your own system—one node on the surface of that giant sphere.

Yet the purpose of that hierarchy in kernel space is to support the absence of hierarchy in userspace.

The Linux kernel itself is produced by a hierarchy that Kernel.org describes as "Linus Torvalds with assistance from a loosely-knit team of hackers across the Net" (**https://www.kernel.org/doc/linux/README**). At the top is Linus, above a few dozen maintainers (**https://www.kernel.org/doc/linux/MAINTAINERS**). Below that are any number of patch submitters. Yet the purpose of that hierarchy in kernel space is to support the absence of hierarchy in userspace. Unlike operating systems from Apple, Microsoft and Google (which bases Android on Linux), Linux serves no corporate agenda (**http://www.linuxjournal.com/content/linux-now-slave-corporate-masters**), meaning it belongs to no hierarchy. The same goes for the Internet in which Linux was born and throughout which it continues to grow and evolve.

Even though it supports an infinite variety of hierarchies, the Internet is not hierarchical. Instead, like Linux, it belongs to a far less talked about class of organizational being: heterarchy (**http://www.thefreedictionary.com/heterarchy**). Derived from the Greek *heteros* (other, different) and *arche* (sovereignty), heterarchy has relatively few examples, none of which make full sense in hierarchical terms.

Adriana Lukas (**http://www.mediainfluencer.net/biography**) explains, "Heterarchy poses an alternative to hierarchy itself, rather than another evolutionary stage of hierarchy. One comes from scarcity and the other from abundance." Business and government are both built in hierarchical forms to manage scarcities. Meanwhile, virtual worlds created by digital technologies and the Internet's protocols are abundant beyond full reckoning, and as alien to scarcity-based mentalities as a sphere to a polygon.

To help get our heads around this sphere, here are Adriana's "Five laws of heterarchy", with brief explanations of each:

1. **Collapse of functions at the node level.** "For a heterarchy to exist and persist, each node has to be able to perform certain base functions, meaning no hard-wired distinctions or divisions of role and functionality among the nodes."

2. **Freedom and ability to bypass, that is, to choose a different path and thus avoid obstacles, control and imposition of hierarchy via backdoor.** "An example of this is John Gillmore's (**http://en.wikipedia.org/wiki/John_Gilmore_(activist)**) familiar statement, 'The Net interprets censorship as damage and routes around it'" (**http://en.wikipedia.org/wiki/John_Gilmore_(activist)#Internet_censorship**).

3. **Decentralized and distributable resources.** "This is the difference between having to go to the post office to make a phone call and the ability to make a call at home or on your mobile phone. This law is important because having to use a centralized resource, apart from restricting nodes' autonomy, opens up opportunities for control of that resource and imposition of a hierarchy."

4. **Abundance of resources or at least abundance of the most important resource that enables to create and maintain the network.** "Don't try to build a heterarchy around a scarce resource, because once that resource is controlled, heterarchy

Advertiser Index

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

ADVERTISER	URL	PAGE #
11th Annual HPC Linux for Wall Street Conference	http://www.flaggmgmt.com/linux/	107
Drupalize.me	http://www.drupalize.me	117
EmperorLinux	http://www.emperorlinux.com	31
LinuxFest Northwest	http://linuxfestnorthwest.org/	73
Microway	http://www.microway.com	3, 49
Percona MySQL Conf	http://www.percona.com/live	115
SharePoint TechCon	http://www.sptechcon.com/	39
Silicon Mechanics	http://www.siliconmechanics.com	7
USENIX NSDI 2014	https://www.usenix.org/conference/nsdi14	113

I suspect that the laws of both hierarchy and heterarchy apply throughout nature, and often dissolve each other.

disintegrates. An example of an abundant resource is information online. Digital format makes information duplicable and therefore abundant."

5. **The marginal cost of communication needs to be zero or near zero.** "This is important because in a peer-to-peer network it takes many more information exchanges to negotiate transactions."

She also says "asymmetrical balance appears to be a general feature of peer-to-peer/heterarchical networks", and provides two examples to explain what she means:

> TCP/IP turns any server into an originator, relaying party or recipient of a message. The relationships between nodes are not symmetrical, but each server at any time can perform any role. Hence the term asymmetrical balance.
>
> BitTorrent is another example—once a user starts downloading a file, he or she is automatically uploading it too, that's the inbuilt balance. But the downloads and uploads are not reciprocal: a different set of nodes are providing files for me to download and different set of nodes are downloading from me, so it's asymmetrical.

Adriana credits *World of Ends* (**http://worldofends.com**) with influencing some of her thinking on heterarchy, which is still in a formative stage. *World of Ends* in turn was influenced by thesis #7 of *The Cluetrain Manifesto* (**http://cluetrain.com**): "Hyperlinks subvert hierarchy", which was coined by David Weinberger (**http://hyperorg.com**). Hyperlinks, Adriana says, are heterarchical, exemplifying all five of her laws. As a kind of corollary to David's thesis, Adriana adds, "Networks dissolve hierarchy."

I suspect that the laws of both

usenix
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

UPCOMING CONFERENCES

For a complete list of USENIX and USENIX co-sponsored events, see www.usenix.org/conferences

FAST '14: 12th USENIX Conference on File and Storage Technologies

February 17–20, 2014, Santa Clara, CA, USA
www.usenix.org/conference/fast14

2014 USENIX Research in Linux File and Storage Technologies Summit
In conjunction with FAST '14
February 20, 2014, Mountain View, CA, USA
Submissions due: January 17, 2014

NSDI '14: 11th USENIX Symposium on Networked Systems Design and Implementation

April 2–4, 2014, Seattle, WA, USA
www.usenix.org/conference/nsdi14

2014 USENIX Federated Conferences Week

June 17–20, 2014, Philadelphia, PA, USA

USENIX ATC '14: 2014 USENIX Annual Technical Conference
www.usenix.org/conference/atc14
Paper titles and abstracts due January 28, 2014

HotCloud '14: 6th USENIX Workshop on Hot Topics in Cloud Computing

WiAC '14: 2014 USENIX Women in Advanced Computing Summit

HotStorage '14: 6th USENIX Workshop on Hot Topics in Storage and File Systems

UCMS '14: 2014 USENIX Configuration Management Summit

ICAC '14: 11th International Conference on Autonomic Computing

USRE '14: 2014 USENIX Summit on Release Engineering

Do you know about the USENIX Open Access Policy?

USENIX is the first computing association to offer free and open access to all of our conferences proceedings and videos. We stand by our mission to foster excellence and innovation while supporting research with a practical bias. Your membership fees play a major role in making this endeavor successful.

Please help us support open access.
Renew your USENIX membership and ask your colleagues to join or renew today!

www.usenix.org/membership

23rd USENIX Security Symposium

August 20–22, 2014, San Diego, CA, USA
www.usenix.org/conference/usenixsecurity14
Submissions due: Thursday, February 27, 2014

Workshops Co-located with USENIX Security '14

EVT/WOTE '14: 2014 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections

USENIX Journal of Election Technology and Systems (JETS)
Published in conjunction with EVT/WOTE
www.usenix.org/jets
Submissions for Volume 2, Issue 2, due: December 5, 2013
Submissions for Volume 2, Issue 3, due: April 8, 2014

HotSec '14: 2014 USENIX Summit on Hot Topics in Security

FOCI '14: 4th USENIX Workshop on Free and Open Communications on the Internet

HealthTech '14: 2014 USENIX Workshop on Health Information Technologies
Safety, Security, Privacy, and Interoperability of Health Information Technologies

CSET '14: 7th Workshop on Cyber Security Experimentation and Test

WOOT '14: 8th USENIX Workshop on Offensive Technologies

OSDI '14: 11th USENIX Symposium on Operating Systems Design and Implementation

October 6–8, 2014, Broomfield, CO, USA
www.usenix.org/conference/osdi14
Abstract registration due April 24, 2014

Co-located with OSDI '14:

Diversity '14: 2014 Workshop on Diversity in Systems Research

LISA '14: 28th Large Installation System Administration Conference

November 9–14, 2014, Seattle, WA, USA
https://www.usenix.org/conference/lisa14
Submissions due: April 14, 2014

Stay Connected...

twitter.com/usenix
www.usenix.org/facebook
www.usenix.org/youtube
www.usenix.org/linkedin
www.usenix.org/gplus
www.usenix.org/blog

hierarchy and heterarchy apply throughout nature, and often dissolve each other. We see this happening live in higher education today. On the hierarchical side, universities value the abundance of knowledge in the world, yet organize knowledge sharing within hierarchical systems that are thick with command, control and costs. On the heterarchical side, universities are having their hierarchical gears stripped (**http://www.shirky.com/weblog/2014/01/there-isnt-enough-money-to-keep-educating-adults-the-way-were-doing-it**) by the ability of pretty much anybody to learn pretty much anything by connecting to published knowledge (and other people), across the vast heterarchy of the Internet. (The challenges and opportunities here are expressed in the title of David Weinberger's latest 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*, **http://www.toobigtoknow.com**.) The room is a heterarchy.

I also believe freedom thrives in heterarchy. Consider the freedoms listed in The Free Software Definition (**https://www.gnu.org/philosophy/free-sw.html**) and how well they align with Adriana's five:

- The freedom to run the program, for any purpose (freedom 0).
- The freedom to study how the program works, and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbor (freedom 2).
- The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

These are embodied in the General Public License (GPL, **https://www.gnu.org/copyleft/gpl.html**), which Linux adopted in the beginning (with Version 2, **https://www.gnu.org/licenses/old-licenses/gpl-2.0.html**). That was 24 years ago. Today business still has trouble grokking free software and the GPL, because nothing in either suggests ways to manage scarcity. From the standpoint of hierarchy, the

fact that Linux has floated $trillions in economic activity is irrelevant. When positive externalities are external to the model framing one's view, they remain out of sight.

The same is true of the Internet. The difference is that nearly all of us access the Internet through ISPs that make it scarce to some degree. Thus, by controlling pathways in the network (and violating Adriana's fourth law), ISPs dissolve some of the Internet's virtues as a heterarchy, and with it the capacity of the Internet to float $trillions in economic activity, rather than mere $billions for ISPs alone.

It helps to recall that the Internet was not born as a "telecommunications service" or an "information service" (the US legal classifications for telephony and cable) or even a "service" at all. It is a state of connectedness made possible by protocols that were not created for billing purposes. Instead they were created—whether their authors knew it or not—in compliance with the laws of heterarchy.

It is interesting that ways have been found to make Linux scarce to some degree, through cloud service providers, such as Amazon and Rackspace. One appeal of these services is that, as Jamie Zawinski once put it, "Linux is only free if your time has no value" (**http://www.jwz.org/doc/linux.html**). Time is scarce for all of us, even when we operate in heterarchies. Yet operating Linux in clouds has become cheaper and cheaper. It is now reasonable to assume that, at some point, the marginal cost of cloud computing and storage will get so close to zero that the fifth law of heterarchy will apply, along with the other four.

And maybe that will happen with the Internet as well, in due time. Once the economic benefits of cheap access to the free and open Internet become fully obvious, business will rush toward the same equilibrium between hierarchy and heterarchy.

But we'll reach that stage a lot faster if we get our collective heads out of the triangles in Flatland and wrapped around the vast spheres of achievement and opportunity floating in our midst. Understanding heterarchy should help with that.■

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.

Send comments or feedback via
http://www.linuxjournal.com/contact
or to ljeditor@linuxjournal.com.

```
 *
 * Usage: $(obj).insertAtCursor(text);
 *   obj = a textarea or textfield
 *   text = a string to insert
 */
$.fn.extend({
  insertAtCaret: function(myValue){
    return $(this).each(function(){
      // If target element is hidden, don't do it.
      if ($(this).is(':hidden') || $(this).parents(':hidde
        return;
      }
      if (document.selection) {
        this.fo
        sel = d
        sel.tex
        this.fo
      }
      else if (
        var sta
        var end
        var scr
        this.va
        this.fo
        // High
```