

LINUX JOURNAL™

Since 1994: The Original Magazine of the Linux Community

APRIL 2010 | ISSUE 192 | www.linuxjournal.com

Use Mongoose
for a Simple
Web Server

Take Rich
Internet
Apps to the
Next Level
with jsormdb

Flash
Application
Development
with
Flex Builder

Software Development



Test Web
Apps with
Selenium

Interview—
Rich Hickey,
Creator of
Clojure



REVIEWED:
Motorola DROID

Troubleshooting
Tips for Sysadmins





Consistency Breeds Predictability

Security Blanket ensures enterprise-wide security policy configuration.

How much time do you spend hardening all of your Linux and Solaris servers (assuming you are hardening all of them)? Are you confident that they are all locked down to a consistent, secure state?

Security Blanket automates the lock down process so that you can regain control of your time and have confidence that your servers are secure. Security Blanket will assess your server environment against industry standard security guidelines (such as DISA STIGs, CIS, and SANS Top 20 Security Risks) and automatically configure the OS to comply with security policy.

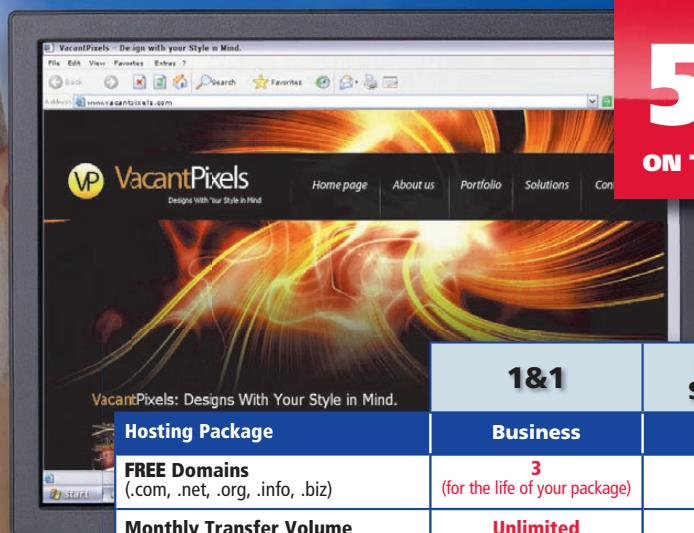
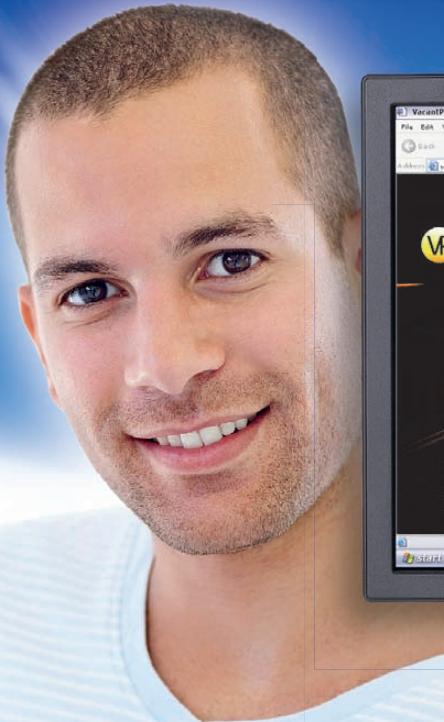
Saving you time, reducing security risk, and consistently complying with policy is what Security Blanket is all about!

Try it for free at www.TrustedCS.com/SecurityBlanket or call us at 1-866-230-1317 for more information.



QUALITY

WEB HOSTING AT THE BEST PRICE!



"Working with startup companies on a tight budget, my clients know an online presence is needed. I use 1&1 exclusively to satisfy their requests for reliable hosting at affordable rates."

Lance Ochs, www.vacantpixels.com

6 MONTHS

50% OFF

ON THE 1&1 BUSINESS PACKAGE*

	1&1	Network Solutions	Go Daddy
Hosting Package	Business	Small	Unlimited
FREE Domains .com, .net, .org, .info, .biz)	3 (for the life of your package)	1 (first year only)	—
Monthly Transfer Volume	Unlimited	3,000 GB	Unlimited
Mailbox Size	2,000 MB	1,000 MB	1,000 MB
Private Domain Registration	FREE	\$9.00/year	—
Search Engine Submission	✓	✓	Extra charge applies
Money Back Guarantee	90 days	30 days	—
Special Offer	6 MONTHS 50% OFF	—	—
Price Per Month	\$9.99 \$4.99 for 6 months*	\$9.96	\$14.24
First Year Hosting Total (with discount)	\$89.91/year	\$128.52/year	\$170.88/year

As the world's largest web host, with over 20 years of experience and state-of-the-art data centers worldwide, 1&1 is dedicated to security, innovation and value when it comes to your website.



99.9%
Uptime!



**90-Day Money Back
Guarantee!**



**24/7 Toll Free
Support**

More special offers are available online. For details, visit www.1and1.com

*Offer valid as of March 1, 2010. 12 month minimum contract term applies. Setup fee and other terms and conditions may apply. Visit www.1and1.com for full promotional offer details. Program and pricing specifications and availability subject to change without notice. 1&1 and the 1&1 logo are trademarks of 1&1 Internet AG, all other trademarks are the property of their respective owners. © 2010 1&1 Internet, Inc. All rights reserved.



Call **1-877-GO-1AND1**
Visit us now **www.1and1.com**

1&1

®

CONTENTS

APRIL 2010 Issue 192

FEATURES

SOFTWARE DEVELOPMENT

50

DEVELOPING FLASH APPLICATIONS WITH FLEX BUILDER

Flash application development on Linux.

Carl Fink



56

MONGOOSE: AN EMBEDDABLE WEB SERVER IN C

A Web server in a single C file.

Michael J. Hammel



62

WEB APPLICATION TESTING WITH SELENIUM

Find the bugs before you release!

Alexander Sirotnik

68

JSORMDB—AN EMBEDDED JAVASCRIPT DATABASE

Taking Rich Internet Applications to the next level.

Avi Deitcher



ON THE COVER

- USE MONGOOSE FOR A SIMPLE WEB SERVER, P. 56
- TAKE RICH INTERNET APPS TO THE NEXT LEVEL WITH JSORMDB, P. 68
- FLASH APPLICATION DEVELOPMENT WITH FLEX BUILDER, P. 50
- TEST WEB APPS WITH SELENIUM, P. 62
- INTERVIEW—RICH HICKEY, CREATOR OF CLOJURE, P. 36
- REVIEWED: MOTOROLA DROID, P. 46
- TROUBLESHOOTING TIPS FOR SYSADMINS, P. 32



ABERDEEN
SOURCES AND STORAGE

WHAT'S THE DEAL WITH THOSE GUYS?

Sometimes you have to ask, "What are they thinking?"

Aberdeen gets it. Businesses are in need of cost effective, reliable, high performance, customizable servers that feature enterprise level benefits with entry level pricing.



Who gives you the best bang for the buck?

	Dell PowerEdge R710	HP ProLiant DL380 G6	Aberdeen Stirling 267
VMware® Ready Certified	✓	✓	✓
Windows Server® 2008 Models	✓	✓	✓
Linux OS Models	✓	✓	✓
Redundant Power	✓	✓	✓
Hardware RAID 0, 1, 5 & 6	✓	✓	✓
SAS / SATA Drive Support	✓	✓	✓
Available with 2TB Drives	✓	✗	✓
Out of Band RAID Management	✗	✗	✓
JBOD Storage Expansion	✗	✗	✓
Dual Intel® Xeon® Processors	E5504 2GHz	E5504 2GHz	E5504 2GHz
Memory	6GB	6GB	6GB
PCI-E Expansion Slots	4	6	7
Hot-Swap Drive Bays	6	6	8
Maximum Capacity	12TB	6TB	16TB
Configured Capacity	3TB	3TB	3TB
Warranty	3 Years	3 Years	5 Years
Price	\$4,462	\$5,338	\$3,995



**Powerful.
Intelligent.**

Prices for the above specific configurations obtained from the respective websites on Jan. 27, 2010. Intel, Intel Logo, Intel Inside, Intel Inside Logo, Pentium, Xeon, and Xeon Inside are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. For terms and conditions, please see www.aberdeeninc.com/abpoly/abterms.htm. IJ033

888-297-7409
www.aberdeeninc.com/ij033

CONTENTS APRIL 2010

Issue 192

COLUMNS

20 REUVEN M. LERNER'S AT THE FORGE

NoSQL? I'd Prefer SomeSQL

28 MICK BAUER'S PARANOID PENGUIN

Linux VPNs with OpenVPN,
Part III

24 DAVE TAYLOR'S WORK THE SHELL

Our Twitter Autoresponder
Goes Live!

32 KYLE RANKIN'S HACK AND /

Linux Troubleshooting, Part II:
Local Network

36 DIRK ELMENDORF'S ECONOMY SIZE GEEK

Interview with Rich Hickey,
Creator of Clojure

80 DOC SEARLS' EOF

Is Android the "Top" Linux?

REVIEW

46 THE MOTOROLA DROID

Brian Conner

IN EVERY ISSUE

8 CURRENT_ISSUE.TAR.GZ

10 LETTERS

14 UPFRONT

40 NEW PRODUCTS

42 NEW PROJECTS

65 ADVERTISERS INDEX

78 MARKETPLACE



46 MOTOROLA DROID



36 RICH HICKEY



44 AUDIOPREVIEW

Next Month: ENTERTAINMENT

Let's get serious. On second thought, let's not. Let's talk Entertainment—that's more fun. And, what's more entertaining than Linux? Running entertainment software on Linux, that's what!

Next month, we look at five of the more-popular music players for Linux, and we compare the media-center applications MythTV and XBMC. But, we don't stop there. We also look at Handbrake for doing video ripping and conversion. Plus, we review the Nokia N900, an entertaining device even when you're not using it for entertainment.

And, if you think running entertainment software is entertaining, wait till you read about the Linux Laptop Orchestra (L2Ork) and find out how entertaining it is actually to create music with Linux.

USPS *LINUX JOURNAL* (ISSN 1075-3583) (USPS 12854) is published monthly by Belltown Media, Inc., 2211 Norfolk, Ste 514, Houston, TX 77098 USA. Periodicals postage paid at Houston, Texas and at additional mailing offices. Cover price is \$5.99 US. Subscription rate is \$29.50/year in the United States, \$39.50 in Canada and Mexico, \$69.50 elsewhere. POSTMASTER: Please send address changes to *Linux Journal*, PO Box 16476, North Hollywood, CA 91615. Subscriptions start with the next issue. Canada Post: Publications Mail Agreement #41549519. Canada Returns to be sent to Blechip International, P.O. Box 25542, London, ON N6C 6B2

Power your planet.

We live on a planet where nearly 6 terabytes of information are being exchanged over the Internet every second, and where billions of connected people are surpassed in number, only by trillions of connected objects and devices. Why then is the average server in the average business running at only 10% utilization? It's hard enough for businesses to meet the demands of a smarter planet today, much less the unforeseen demands of tomorrow. The new POWER7 Systems™ from IBM are not simply servers—they're fully integrated systems with the ability to run hundreds of virtual servers, helping you drive up to 90% utilization. These next-generation systems integrate massive parallel processing, throughput computing and analytics capabilities to optimize for the complex workloads of an increasingly data-driven world. Learn how to power your planet at ibm.com/poweryourplanet

Smarter systems for a Smarter Planet.



LINUX JOURNAL™

Since 1994: The Original Magazine of the Linux Community

DIGITAL EDITION NOW AVAILABLE!

Read it first

Get the latest issue before it
hits the newsstand

Keyword searchable

Find a topic or name
in seconds

Paperless archives

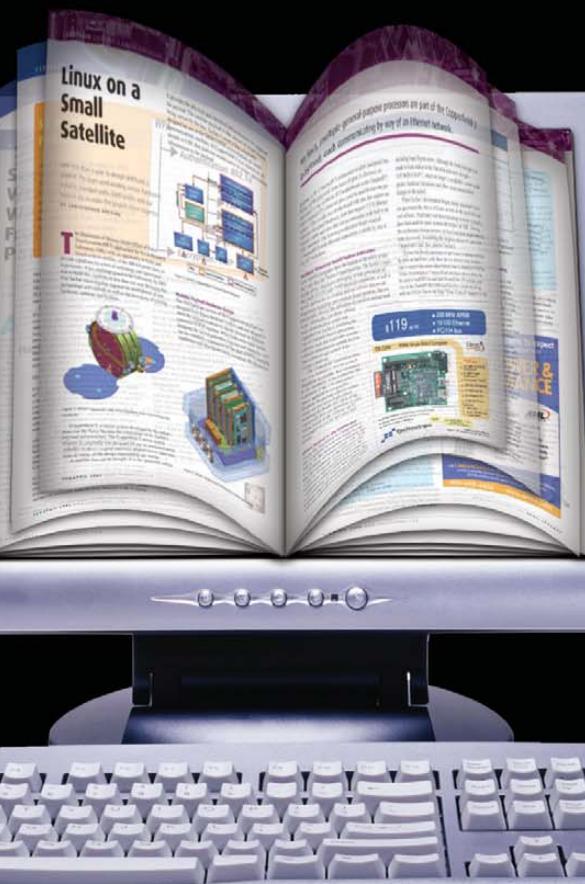
Download to your computer for
convenient offline reading

Same great magazine

Read each issue in
high-quality PDF

Try a Sample Issue!

www.linuxjournal.com/DLISSUE



LINUX JOURNAL

Executive Editor	Jill Franklin jill@linuxjournal.com
Senior Editor	Doc Searls doc@linuxjournal.com
Associate Editor	Shawn Powers shawn@linuxjournal.com
Associate Editor	Mitch Frazier mitch@linuxjournal.com
Art Director	Garrison Antikajian garrison@linuxjournal.com
Products Editor	James Gray newproducts@linuxjournal.com
News Editor	Justin Ryan news@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

David A. Bandel • Ibrahim Haddad • Robert Love • Zack Brown • Dave Phillips • Marco Fioretti
Ludovic Marcotte • Paul Barry • Paul McKenney • Dave Taylor • Dirk Elmendorf

Proofreader Geri Gale

Publisher Carlie Fairchild
publisher@linuxjournal.com

General Manager Rebecca Cassity
rebecca@linuxjournal.com

Senior Print Media Sales Manager Joseph Krack
joseph@linuxjournal.com

Digital Media Sales Manager Michael Beasley
michael@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 Ballo • Nick Baranian • Hari Boukis • Steve Case
Kalyana Krishna Chadalavada • Brian Conner • Caleb S. Cullen • Keir Davis
Michael Eager • Nick Faltry • Dennis Franklin Frey • Alicia Gibb
Victor Gregorio • Philip Jacob • Jay Kruizinga • 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
PHONE: +1 818-487-2089
FAX: +1 818-487-4550
TOLL-FREE: 1-888-66-LINUX
MAIL: PO Box 16476, North Hollywood, CA 91615-9911 USA
Please allow 4-6 weeks for processing address changes and orders
PRINTED IN USA

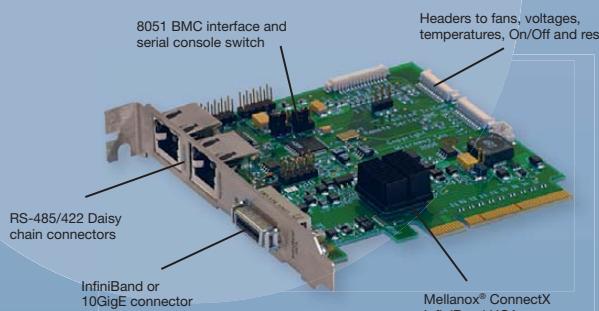
LINUX is a registered trademark of Linus Torvalds.



Your Applications Will Run Faster With Next Generation Microway Solutions!

TriCom™ X

- QDR/DDR InfiniBand HCA
- ConnectX™ Technology
- 1μsec Latency
- Switchless Serial Console
- NodeWatch™ Remote Management



Teraflop GPU Computing

For Workstations and HPC Clusters

- NVIDIA® Tesla™ GPU with 240 Cores on One Chip
 - CUDA™ SDK
- NVIDIA® Quadro® Professional Graphics
- AMD® FireStream™ GPU
 - Stream SDK with Brook+



NumberSmasher®

Large Memory Scalable SMP Server

- Scales to 1 TB of Virtual Shared Memory
- Up to 128 CPU Cores
- 8U System Includes 32 Quad Core CPUs
- QDR 1 μsec Backplane



FasTree™ X

- Mellanox® InfiniScale™ IV Technology
- QDR/DDR InfiniBand Switches
- Modular Design
- 4 GB/sec Bandwidth per Port
- QSFP Interconnects
- InfiniScope™ Real Time Diagnostics

Call the HPC Experts at Microway to Design Your Next High-Reliability Linux Cluster or InfiniBand Fabric.

508-746-7341

Sign up for Microway's Newsletter at
www.microway.com

Microway
Technology you can count on™



SHAWN POWERS

Focusing on the Softies

This month, we focus on software development. Thankfully, for those of us not terribly gifted in such endeavors, Linux allows a rather wide definition of the subject. Although it may not allow for my flavor of soft development (sculpting mashed potatoes), it does include anything from shell scripting to Web pages, and kturtle to binary.

If you're fed up with all the existing programming languages and you want something completely new, you may want to look at Dirk Elmendorf's interview with Rich Hickey. Rich created Clojure, and although it doesn't claim to re-invent the wheel, it does put a new spin on existing concepts. Don't worry if new languages scare you. This month, Dave Taylor takes us back to the trusty shell scripting world, and we see the Twitter program we've been working on in action. Using tools like AWK may not seem cutting edge, but having a Twitter bot respond to you takes some getting used to!

Both Reuven M. Lerner and Avi Deitcher talk about databases this month. Reuven talks about the NoSQL movement, and Avi discusses jsormdb. Although the former might arguably not be about a database, but rather a lack of one, jsormdb is most assuredly a JavaScript database that can be embedded into your applications. Sometimes a database isn't enough though, and you need a full-blown Web server for your application. Michael J. Hammel shows us Mongoose, which is exactly that. Mongoose is a fully embeddable Web server written in C.

Honestly, the programming many of us are exposed to all have their GUI pieces on the Web. Carl Fink shows us how to use Flex Builder to make Flash applications—in Linux! Adobe's Flash is cross-platform, but often creating those applications requires Windows or OS X. With Flex Builder, that's no longer the case. Add Adobe's AIR technology, and even standalone applications can be created while never leaving the penguiny comfort of Linux. If the notion of developing Web applications on Linux concerns you (it really shouldn't), perhaps Alexander Sirokin's article on Selenium will ease your mind a bit.

Using Selenium, a programmer can test scripts written in a variety of programming languages, including Perl, Java, C# and others.

If you are like me, and your mashed-potato-sculpting skills are better than your programming skills, this issue won't leave you feeling lumpy. Although we don't have any sculpting tutorials, we do have a bunch of other interesting topics with real a-peel. (Okay, enough with the potato jokes.) Brian Conner reviews the Motorola DROID this month, and as a DROID owner myself, I can easily say it's a review you'll want to read. You'll also want to read Anton Borisov's interview with three important women in Linux. Valerie Aurora, Sarah Sharp and Stormy Peters all weigh in on what it's like to be a woman in the Open Source world.

We've also got our regular cast of columnists helping us hone our geek skills. Kyle Rankin continues his series on troubleshooting. This month, he debugs two troublesome servers, oddly named shawn and bill. I'll leave it to the reader to determine which server is more troublesome. Be sure to keep in mind that shawn and bill are servers with communication problems; otherwise, Kyle's article starts to look like a *Linux Journal* group-therapy session.

Mick Bauer continues his series on OpenVPN, and if you have to work remotely or possibly don't trust those folks on your network, you'll want to catch this month's article. Or, perhaps total anonymity is your cup of tea. John Knight covers that in our New Projects column, where he talks about Tor. It's not the fastest way to transmit data, but it's hard to find a more anonymous way.

So whether you're a programmer with your mouth watering for this programming issue or a fellow mashed-potato fan with your mouth watering for bacon and chives, this issue should satiate your needs. Just remember not to lick the pages, people frown on that.■

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.

YOUR HIGH PERFORMANCE COMPUTING HAS ARRIVED.

The ServersDirect® Systems with the Intel® Xeon® Processor helps you simplify computing operations, accelerate performance and accomplish more in less time

STARTING AT **\$899****ENTRY LEVEL INTELLIGENT SERVER**

SDR-S1341-T00 is among our most cost-effective 1U Xeon Servers, and it is ideal for large high-performance computing deployments

STARTING AT **\$959****APPLICATION SERVER**

Refresh your servers with new **SDR-S1337-T02** powered by Intel® Xeon® processor 5500 series, based on intelligent performance, automated energy efficiency and flexible virtualization.

**SDR-S1343-T04**
STARTING AT **\$1,099****1U INTEL® XEON® PROCESSORS 5500 SERIES SERVER W/ 4X 3.5" HOT-SWAP SATA DRIVE BAYS**

- Supermicro 1U Rackmount Server with 560W Power Supply
- Supermicro Server Board w/Intel® 5520 Chipset
- Support up to Dual Intel® 5500 series Xeon® Quad/Dual-Core, with QPI up to 6.4 GT/s
- Support up to 96GB DDR3 1333/ 1066/ 800MHz ECC Reg.DIMM
- 4x 3.5" Hot-swap SATA Drive Bays
- Intel® 82576 Dual-Port Gigabit Ethernet Controller

**SDR-S2311-T08**
STARTING AT **\$1,159****2U INTEL® XEON® PROCESSORS 5500 SERIES SERVER W/ 8X 3.5" HOT-SWAP SAS/SATA BAYS**

- Supermicro 2U Rackmount Server with 560W Power Supply
- Supermicro Server Board w/Intel® 5500 Chipset
- Support up to Dual Intel® 5500 series Xeon® Quad/Dual-Core, with QPI up to 6.4 GT/s
- Support up to 24GB DDR3 1333/ 1066/ 800MHz ECC Reg.DIMM
- 8x 3.5" Hot-swap SATA Drive Bays
- Dual Intel® 82574L Gigabit Ethernet Controller

**SDP-IP308-T10**
STARTING AT **\$1,599****PEDESTAL INTEL® XEON® PROCESSORS 5500 SERIES SERVER W/ 10X HOT-SWAP (OPT.) SATA BAYS**

- Intel Pedestal Chassis w/ 750W (1+1) Power Supply
- Supermicro Server Board w/Intel® 5520 Chipsets
- Support up to Dual Intel® 5500 series Xeon® Quad/Dual-Core, with QPI up to 6.4 GT/s
- Support up to 96GB DDR3 1333/ 1066/ 800MHz ECC Reg./unbuffered DIMM
- Option 10x 3.5" Hot-swap SATA Bays
- Intel® 8257EB Dual-port Gigabit Ethernet Controller

**SDR-S4313-T24**
STARTING AT **\$1,899****4U INTEL® XEON® PROCESSORS 5500 SERIES SERVER W/ 24X 3.5" HOT-SWAP SAS/SATA BAYS**

- Supermicro 4U Rackmount 900W (1+1) Red. Power Supply
- Supermicro Server Board w/ Dual Intel® 5520 Chipsets
- Support up to Dual Intel® 5500 series Xeon® Quad/Dual-Core, with QPI up to 6.4 GT/s
- Support up to 144GB DDR3 1333/ 1066/ 800MHz ECC Reg. DIMM
- 24x 3.5" Hot-swap SATA Drive Bay
- Intel® 82576 Dual-port Gigabit Ethernet Controller

**SDR-S3305-T16**
STARTING AT **\$1,979****3U INTEL® XEON® PROCESSORS 5500 SERIES SERVER W/ 16X 3.5" HOT-SWAP SAS/SATA BAYS**

- 3U Rackmount Server with 1+1 900W Red. Power Supply
- Supermicro Server Board w/ Dual Intel® 5520 Chipsets
- Support up to Dual Intel® 5500 series Xeon® Quad/Dual-Core, with QPI up to 6.4 GT/s
- Support up to 96GB DDR3 1333/ 1066/ 800MHz ECC Reg.DIMM
- 16x Hot-swap SAS/SATA Drive Bays
- Intel® Dual 82576 Dual-Port Gigabit Ethernet (4 ports)

**SDR-C9303-T50**
STARTING AT **\$4,339****9U INTEL® XEON® PROCESSORS 5500 NEHALEM SERIES SERVER W/ 50X HOT-SWAP SATA II / SAS BAYS**

- 9U Chassis with 1620W Redundant Power Supply
- Supermicro Server Board w/ Dual Intel® 5520 Chipsets
- Support up to Dual Intel® 5500 series Xeon® Quad/Dual-Core, with QPI up to 6.4 GT/s
- Support up to 144GB DDR3 1333/ 1066/ 800MHz ECC Reg. DIMM
- 50 x 3.5"Internal SATA Drives Trays
- Intel® 82576 Dual-port Gigabit Ethernet Controller

**SERVERS DIRECT CAN HELP YOU CONFIGURE YOUR NEXT HIGH PERFORMANCE SERVER SYSTEM - CALL US TODAY!**

Our flexible on-line products configurator allows you to source a custom solution, or call and our product experts are standing by to help you to assemble systems that require a little extra. Servers Direct - your direct source for scalable, cost effective solutions.

1.877.727.7886 / www.ServersDirect.com

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, Pentium, and Pentium III Xeon are trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

**Powerful.
Intelligent.**

letters



Error in Distance Calculation

The distance calculation in Dave Taylor's December 2009 Work the Shell is in error because he's not using enough significant digits. Six digits usually are not sufficient to define the position of a place or object accurately in lat/lon coordinates. In my experience, it typically takes nine digits to get sub-meter accuracy.

--
Michael Wallace

Dave Taylor replies: Ahh, I'll have to check that out. Thanks for the tip!

National Debt = 1,587 Miles

I like your National Debt figure tally, which used to appear in the *LJ* Index. It is a healthy reminder of the country's economic burden. However, it is difficult to grasp how much money it is. Here is my favorite method to "get it". A \$100 bill is approximately 0.1mm thick. So in a cm, you have 100 bills = \$10,000. In a meter, you can stack a million dollars, or 1.6 billion dollars in a mile. So a National Debt of 2.54 trillion dollars is a stack of 1,587 miles of \$100 bills! If you split it in two and put it on a road, you will have almost 800 miles of \$100 bills. If you drive your car on it, you will run out of gas before reaching the end!

--
Carlos Vidal

Cloud Computing

I devoured Mick Bauer's Paranoid Penguin column "Linux Security Challenges 2010" in the January 2010 issue. How I wish the people where I work had one-tenth of his insightful common-sense view! But, I think one challenge is missing, user education. Although the IT department must bear the ultimate responsibility for security, I think end users should be held accountable. Computer basics should be on the hiring criteria for any position that requires the use of a computer, and basic training should be compulsory for new hires, as should annual "what's new" sessions. My company offers free training in just about everything except computer know-how, and it considers including its computer/e-mail policy statement in the new-hire packet satisfactory. As a result, many users don't know what copy/paste does, but they use company laptops for everything from on-line shopping to browsing porn sites. They have no motivation to change, because their behavior has no consequences. But if doing something stupid costs properly trained employees bonus points, they might think twice before clicking.

--
Peter Bratton

Linux and DRM

If you ask me why I keep a Windows desktop, the answer is still three letters, DRM. I have two desktops: my Ubuntu 9.10 (which is very good and does just about everything) and my W7 desktop (which does everything with DRM). Even with the CrossOver Office Plugin, I can't run Amazon Unbox, which allows me to watch my favorite TV shows over and over.

I have been a Linux user since RH 6.2, and I love the way things have come along. There have been few DRM issues, but again, until we make some kind of headway with all the major distributors, I will have to keep a Windoze desktop of some kind. I now run W7. It's half the bloated warthog of its predecessor, but it's still not my beloved GNOME desktop. (Yes, I really love GNOME.)

The day I can run the Amazon Unbox client, or the day Amazon ports Unbox over to Linux (and, of course, somebody works out a deal with Netflix) is the day I give up this machine of great power (Intel i7) and make it my new, true desktop.

My Linux desktop is highly functional and a flawless media server and print server for the house. Now, if we could only get the major streaming media people to help us out.

--
Mark Kaufman

I know exactly what you mean. Linux itself is quite functional; it's just all the services from others that are causing the problem. I have the same issue with DRM, and while I try diligently to avoid it, there are times when I'm forced into such things. Netflix, for example, runs great on the Linux-based Roku device, but for those of us with Linux desktops, we can't watch streaming video! So, yes, I feel your pain.—Ed.

Google Chrome, Linux and Web Applications

Recently, I decided to download Google Chrome for Ubuntu after the almost stealthy release last month. I was blown away by the many great features and other hoopla we've been eagerly awaiting. One surprise feature, which is of extreme interest to myself as a Web developer, is that Google Chrome has an App Mode where it runs a Web site in its own application window, à la Mozilla's Prism.

There's one problem though. The .desktop files it creates to do so are garbage and won't install into /usr/share/applications without hacking permissions on the folder, which will lead to constant messages indicating the change in permissions when applications are installed/updated. So, not to be defeated by such a simple problem, I came up with what I'd like to think is an elegant solution—a Python app I call chromify.

It's a simple application that makes up for Google Chrome's shortcoming in creating clean .desktop files. For now,

it's just a CLI program, but it's still very feature-rich and uses a Python module so that other interfaces can be used/developed (that is, CGI or a GUI) to create Google Chrome Web apps.

Downloads, installation guides and a wiki are available at code.google.com/p/chromify.

Note: I'm not trying to advertise here. This was something of an extreme annoyance for me, and I know it likely is for others. This is more of an "I thought I'd share" letter than anything else.

--
G. John Schutte

Whoa, that's awesome. Prism with Firefox has been rather wonky of late. I do like Chrome's App Mode, and although I haven't had a chance to play with it enough to discover the issues with its .desktop files, it sounds like it's pretty annoying! Thanks for not only fixing the problem, but also for sharing your solution. Hooray for open source!—Ed.

OpenVPN

I loved Mick Bauer's "Linux VPNs with OpenVPN" article in the February 2010 issue, and I think it is long overdue. OpenVPN is a great technology. I'm just wondering if Mick meant VPN where VLAN is listed on page 28, as these are two completely different types of network technology. This also brings up a good idea—any chance on doing an article describing how to configure Linux for use with VLANs?

--
Aaron

Mick Bauer replies: You're quite right, both instances of the term *VLAN* in this article (in the same paragraph) were misprints; I meant to say *VPN*. I hope this didn't cause too much confusion!

You're also right that *VLANs* (Virtual LANs) would make a good topic. Actually, Paul Frieden covered *VLANs* in 2004 on the LJ Web site, in his article "VLANs on Linux" (www.linuxjournal.com/article/7268), including its use with *iptables*. But, I've been thinking of covering this topic myself in the context of a larger article on building Linux broadband routers using OpenWRT—stay tuned for the (probable) upcoming

series. Thanks both for the correction and the suggestion!

Comparing Files

Joey Bernard's review of diff and its relatives on page 16 of the February 2010 issue was useful, but he didn't mention sdiff, which is the basis of my favorite way of comparing two text files.

The problem with `diff -y` is that the side-by-side format isn't very convenient. I prefer to use the following:

```
sdiff -s -w156 file1 file2 | more
```

This places the lines of file1 above those of file2, if they differ. Small differences really stand out in this format. The `-w 156` makes corresponding columns line up in an 80-column text window.

--
Andrew T. Young

Clarification

Thanks for the wonderful article "Running Remote Applications" by Michael J. Hammel in the February 2010 issue of *Linux Journal*.

In the article, it says "Because VNC is based on the tile architecture, where rectangles of frame buffer memory are resent if they have been updated, any compression that improves the transfer of tiles will have serious performance implications." I am confused. Are you saying that if tile compression of the frame buffer segments is implemented that the performance would be worse as opposed to non-tile compression? One would think that frame buffer compression would improve bandwidth usage but may, in fact, increase latency—what do you think?

By the way, it would be great if a follow-up article could be done on remote multimedia in concert with remote desktop—I was thinking about UPNP AV and DLNA.

--
Daya Cooppan

Michael J. Hammel replies: I'm not completely sure about this. I think this is information I found during my research and not in my experimentation, but I thought it worthwhile to include in the article. Compression, in general, should increase latency while reducing bandwidth. However, my interpretation of my own

writing is that the client side potentially sees tile compression from the server as raw pixel changes and, therefore, asks for all tiles repeatedly. If this is true, compression potentially increases bandwidth by resending more tiles than what was actually updated. I'm not sure this is actually the case, however. My recent use of VNC using *virt-viewer* and friends doesn't seem to show this, although I can't verify (at the moment) if compression is being used. In essence, I can't find my notes that say where this came from. At best, it is accurate based on specific implementations. My apologies for not being able to back it up further.

As far as a follow-up article on UPNP AV and DLNA, I'm not familiar with either at the moment, but I'll certainly look into them. I love to find new technologies to play with, and if I can speak intelligently about them, I'll certainly propose it to LJ (if I can just remember to hang onto my research notes—sigh). I believe Red Hat's recent announcement on SPICE is related to this problem space, and I'm planning on investigating that as well. Thanks for the feedback!

Linux on the Desktop

Linux has provided my desktop environment 99.99% of the time since somewhere around the days of Red Hat v3 (that's mid-1990s) at work (college computer technology instructor, retired for about five years now) and at home. I also used it on my laptop and several lab servers at work.

Applixware was the first office suite I used. Then came StarOffice and currently OpenOffice.org. Regarding Bruce Byfield's article "OpenOffice.org vs. Microsoft Office" in the February 2010 issue, he makes the statement: "Similarly, Impress lacks the ability to use the pointer to draw on the screen during a presentation." This feature has been present for some years before I retired. To use it, one simply must enable it in the Slide Show→Slide Show Settings menu. The three-pane window of Impress was never an issue either. The best solution for me was a dual-head system. Any geek worthy of the name should be able to scrounge up a second monitor and a dual-head video card. The extra screen space is very useful.

My only gripe with Linux is that some-

[LETTERS]

where between Fedora 8 and 9 the dual head, aka multiple monitor, facility was broken for my video card, the Matrox G450. I have reported the bug as new releases come along, but it is still broken. I continue to use the latest release of Fedora for some things, as I am right now, but when I need to be productive (for example, programming, image editing, Web site development), it's back to my Fedora 8 system.

I encourage others to try OpenOffice.org whenever the opportunity arises. It has served my needs well. What the future will bring to the desktop, only time will tell. But as for me and my data and apps, I do *not* want them blended with the stuff out on the Net. Eat local, compute local. If it comes from far away, I want to know.

--
Paul Almquist

PHOTO OF THE MONTH

Have a photo you'd like to share with *LJ* readers? Send your submission to publisher@linuxjournal.com. If we run yours in the magazine, we'll send you a free T-shirt.



In the Heart of Darkness, Penguins! (On Seattle's quay.) Submitted by Curtis Vaughan.

Thanks for the correction on OpenOffice.org. As far as the Matrox G450 goes, I find it amazingly ironic that it doesn't work well, considering the Matrox cards used to be the only video cards a Linux user could rely on! Hopefully, things will straighten out for your trusty G450 soon.—Ed.

Linux Remote Desktop

I liked Michael J. Hammel's article "Running Remote Applications" in the February 2010 issue, but there's an easier way to get a remote desktop under Linux. I use FreeNX (<https://help.ubuntu.com/community/FreeNX> is a good place to start) provided by NoMachine (www.nomachine.com). It's faster than VNC and secure. I connect to my desktop computer at home from my workplace, and it's almost like sitting in front of the computer. Better still, it allows you to suspend a session and reconnect to it later, so when I come back to work the next day, I can resume the session, and all the windows I had open are still ready to go, including whatever pages I had open in Firefox.

I sound like I work for NoMachine, but I just like it that much. It is free software.

--
Marcus Huculak

Michael J. Hammel replies: I'd heard of FreeNX and had looked at NoMachine some time back, but I simply didn't remember it before working on the article. Being an old-timer, I was using XDMCP. Even VNC was new to me when I started into the article. FreeNX certainly sounds like it's worth investigating in the future.

LINUX JOURNAL

At Your Service

MAGAZINE

PRINT SUBSCRIPTIONS: Renewing your subscription, changing your address, paying your invoice, viewing your account details or other subscription inquiries can instantly be done on-line, www.linuxjournal.com/subs. Alternatively, within the U.S. and Canada, you may call us toll-free 1-888-66-LINUX (54689), or internationally +1-818-487-2089. E-mail us at subs@linuxjournal.com or reach us via postal mail, Linux Journal, PO Box 16476, North Hollywood, CA 91615-9911 USA. Please remember to include your complete name and address when contacting us.

DIGITAL SUBSCRIPTIONS: Digital subscriptions of *Linux Journal* are now available and delivered as PDFs anywhere in the world for one low cost. Visit www.linuxjournal.com/digital for more information or use the contact information above for any digital magazine customer service inquiries.

LETTERS TO THE EDITOR: We welcome your letters and encourage you to submit them at 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, www.linuxjournal.com/author.

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, www.linuxjournal.com/advertising. Contact us directly for further information, ads@linuxjournal.com or +1 713-344-1956 ext. 2.

ON-LINE

WEB SITE: Read exclusive on-line-only content on *Linux Journal's* Web site, www.linuxjournal.com. Also, select articles from the print magazine are available on-line. Magazine subscribers, digital or print, receive full access to issue archives; please contact Customer Service for further information, subs@linuxjournal.com.

FREE e-NEWSLETTERS: Each week, *Linux Journal* editors will tell you what's hot in the world of Linux. Receive late-breaking news, technical tips and tricks, and links to in-depth stories featured on www.linuxjournal.com. Subscribe for free today, www.linuxjournal.com/eneWSletters.

Save the Date!

USENIX Federated Conferences Week

June 22–25, 2010, Boston, MA

www.usenix.org/events/#june10

USENIX ATC '10

2010 USENIX Annual
Technical Conference

Wednesday–Friday, June 23–25

www.usenix.org/atc10

REGISTRATION
DISCOUNTS
AVAILABLE!

WebApps '10

USENIX Conference on Web
Application Development

Wednesday, June 23

www.usenix.org/webapps10

WOSN 2010

3rd Workshop on Online
Social Networks

Tuesday, June 22

www.usenix.org/wosn10

HotStorage '10

2nd Workshop on Hot Topics
in Storage and File Systems

Tuesday, June 22

www.usenix.org/hotstorage10

HotCloud '10

2nd USENIX Workshop on
Hot Topics in Cloud Computing

Tuesday, June 22

www.usenix.org/hotcloud10

And more!

Visit www.usenix.org/events/#june10 for new workshop
announcements and registration information.

Stay Connected...

USENIX

www.usenix.org

 <http://www.usenix.org/facebook>

 <http://www.usenix.org/linkedin>

 <http://twitter.com/usenix>

 <http://blogs.usenix.org>

diff -u

WHAT'S NEW IN KERNEL DEVELOPMENT

Rafal Milecki is reworking the **backlight device class**.

Apparently, tons of strange configurations are cropping up, where laptops have multiple monitors attached in different ways, and the current code is just not cutting it anymore. He's been considering how best to rewrite the whole thing.

Henrique de Moraes Holschuh also is very interested in the idea of doing this, and the two of them seem to be getting into it together. **Richard Purdie**, the current backlight code maintainer, so far is not involved, but it's clear that some changes are coming soon to the whole backlight device class.

GPL violations can be difficult to track down, because the violators often release only binary versions of the software. But, **Gerhard Wiesinger** felt he'd uncovered a violation in **VMware's ESX server**. VMware's own documentation apparently reported that some of its drivers had been based on versions of Linux drivers that had to have been under the GPL at the time VMware based its work on them. And, because the drivers were statically linked with the ESX server, Gerhard felt that this meant the entire ESX server was bound to the GPL and had to be released under its provisions. The truth of the matter isn't yet known, and accusations aren't proof. But, it's good to take note of these potential issues as they come up.

Sam Ravnborg no longer is maintaining **kbuild**. He handed maintainership off to **Michał Marek**, who seems very excited to get started with it. One of the first things to happen after the announcement was that

Arkadiusz Miskiewicz reported

a problem with color no longer being displayed in the kbuild menus. Under Michał's new leadership, **Nir Tzachar** tackled the problem and uncovered its vile roots. Best of luck to Michał and the rest of them on advancing kbuild.

Joe Moo (his real name's Paul) has started a new **SourceForge** project to help users test the latest Linux **kernel patches**. The SourceForge page is sourceforge.net/projects/thelinuxtesters. The kernels that get the most testing, of course, are the official releases—those 2.6.36, 2.6.37 and other kernels that **Linus Torvalds** puts out once in a while. Even more valuable to Linux kernel stability, however, would be for more people to test the current working version of Linus' tree—the code at the very tip of the git repository at any given time. Joe's project tries to make it easier for people in general to do that.

The **ReiserFS** code has been so filled with the **big kernel lock**, that **Frédéric Weisbecker** had to create his own git repository just to deal with rooting out all occurrences of the BKL in ReiserFS and destroying them. Linus Torvalds recently asked about the status of that, and **Ingo Molnar** said Frédéric had been creating a huge dust vortex around his workstation, yanking BKL after BKL out of the code with his bare hands!

Thomas Gleixner also had been monitoring and reviewing Frédéric's work, and Frédéric also announced that very, very soon Linus would get a pull request that would carry all the BKL-yanking into the official tree.

—ZACK BROWN



Talking on Cell Phones Is So 2009

Like many families, the Powers' house has a "family plan" for our cell service. A year ago, when we estimated the minutes we'd need on a shared plan, 1,400 per month seemed like a good amount. Six months into our contract, we had more than 6,000 rollover minutes (yes, we use AT&T). To compensate, and save some money, we lowered our plan to 700 shared minutes. This Christmas, when our oldest daughter got a cell phone, we discovered we had several thousand new rollover minutes, even with the reduced plan!

Perhaps data-only cell phones will be here sooner than later. Certainly with things like Google Voice, Skype, Gizmo5 and any number of other VoIP services, seamless transition from Wi-Fi to 3G makes sense. As long as we can send text messages, pictures of cats and answer the occasional phone call via VoIP, why do we need traditional voice services? As someone who works in a building with no cell-phone coverage, but plenty of Wi-Fi bandwidth, I certainly would welcome accessibility from indoors. How about you? Would you take a cell phone with a zero-minute plan?

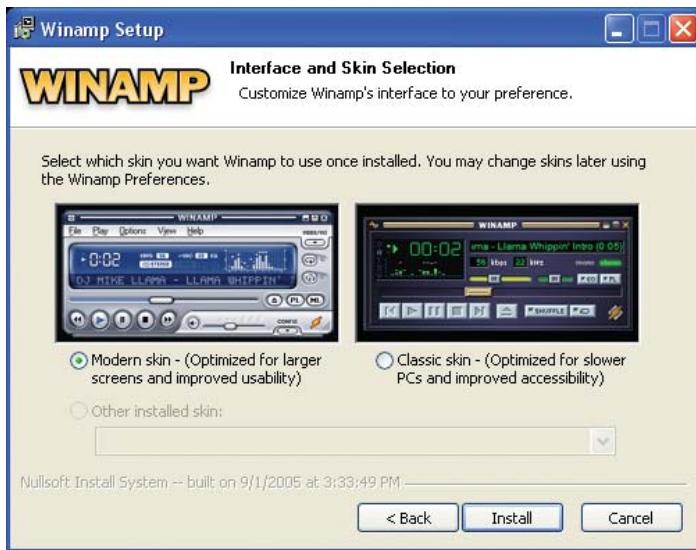
—SHAWN POWERS

NON-LINUX FOSS

Windows users expect an installer program when they get new software. But, creating a full-featured installer could end up taking more time than your application did. Fortunately though, creating installers is largely an automated task on Windows these days, as there are numerous tools to assist you in creating a GUI installer for your application. One of them is the Nullsoft

Scriptable Install System (NSIS). NSIS was developed by Nullsoft, the developers of the Winamp media player. NSIS is free and open-source software.

NSIS installers are built from installer scripts. These scripts control the actions of the installer when it is run. The scripts allow the developer



NSIS Installer for Winamp

to create sophisticated installers that can deal with complex installation tasks, such as handling upgrades differently based on the already-installed version. NSIS includes a plugin system that allows the installer to be customized to include custom steps and interfaces. Plugins can be written in languages that can produce a Windows DLL.

Once the installer scripts are ready, they are compiled with the NSIS compiler, and an installation program is created. One of the nice features of NSIS is that the script compiler can be built and run on any POSIX-compliant platform (such as Linux, BSD and UNIX), allowing you to build your installer on a system you actually like using.

NSIS is actively maintained, well documented, and it runs on all modern versions of Windows (from Windows 95 on). The latest release at the time of this writing is NSIS version 2.46.

—MITCH FRAZIER

New LinuxJournal.com

You may have noticed something a little different about your favorite on-line destination. We've made a major overhaul and invite you to make yourself at home and try out the new features. We encourage you to create a user account if you haven't already. Logged-in users can keep track of all their favorite articles, connect with other readers and editors, and participate in our growing on-line community.

You'll find all the same great articles, blogs and videos, and we hope you enjoy the new look and layout. Additionally, look for new features to be added throughout the year. If you use Linux, LinuxJournal.com should be your home page.

See you on the interwebs!

—KATHERINE DRUCKMAN



LJ Index April 2010

- Percent of US smartphone Web traffic from iPhones: **55**
- Percent of US smartphone Web traffic from Android phones: **20**
- Percent of US smartphone Web traffic from Research In Motion (RIM) OS phones: **12**
- Percent of US smartphone Web traffic from webOS (Palm) phones: **5**
- Percent of US smartphone Web traffic from Windows Mobile OS phones: **4**
- Percent of US smartphone Web traffic from Palm OS phones: **1**
- Percent of worldwide smartphone Web traffic from iPhones: **50**
- Percent of worldwide smartphone Web traffic from Symbian OS phones: **15**
- Percent of worldwide smartphone Web traffic from Android phones: **11**
- Percent of worldwide smartphone Web traffic from RIM OS phones: **7**
- Percent of worldwide smartphone Web traffic from Windows Mobile OS phones: **3**
- Percent of worldwide smartphone Web traffic from webOS (Palm) phones: **2**
- Percent of worldwide smartphone Web traffic from Palm OS phones: **1**
- Percent of worldwide smartphones sold by Nokia: **45**
- Percent of worldwide smartphones sold by RIM: **13**
- Percent of worldwide smartphones sold by Apple: **5**
- Percent of worldwide smartphones sold by HTC: **4**
- Percent of worldwide smartphones sold by Fujitsu: **4**
- Millions of smartphones sold per year worldwide: **145.6**
- Millions of cell phones sold per year worldwide (including smartphones): **1,076.5**

Sources: 1–13: October 2009 Mobile Metrics Report | 14–20: Gartner Research

Stupid afio Tricks

I've already covered tar and all the wonderful ways to use it, but it's not the only tool at your disposal. Another popular backup tool is afio. Depending on your distribution, it already may be installed. On Ubuntu, for example, run the following to install it:

```
sudo apt-get install afio
```

Now you have a fairly powerful tool at your disposal for archiving files and making backups.

By default, afio reads and writes the files being archived on standard input and standard output. This means you can create a list of files to archive with another program, like find, and pipe it to afio to do the actual archive. Once you have your list of files, you can apply five basic commands to those files:

- -o — create an archive.
- -i — install (or unpack) an archive.
- -t — test (or list) the files stored in an archive.
- -r — verify the files stored in an archive against the filesystem.
- -p — copy the files to a given directory location.

If you want to create a simple archive of all of your C source code files, execute:

```
find . -name *.c -print | afio -o -Z source_code
```

When you want to extract these files again, execute:

```
afio -i -Z source_code
```

When you run afio as a regular user, all file paths are stored with no leading /. This means when you unpack an archive, it unpacks in the current directory. The idea is to avoid overwriting system files accidentally. To keep the leading /, use the command-line option -x. Running afio as the superuser reverses this behavior—any leading / is maintained, and you need to use the command-line option -X to get the usual behavior of stripping the leading /.

If space is at a premium, afio also can compress your archive, just like tar, with the -Z command-line option. There is one very big difference, however. When you compress a tar archive, the entire archive file is compressed. This means if one part of the file is corrupted, you potentially could lose all the files in the archive. When you compress an afio archive, the archived files actually are compressed individually. So if one file becomes corrupted, by whatever means, you won't lose any of the other files in the archive.

When you compress an archive, afio uses gzip by default.

You can tell gzip what compression factor to use with the -G *num* command-line option, where *num* is the amount of compression gzip will apply to the archived files. This is a number between 0 (for no compression) and 9 (for maximum compression), with a default of 6.

You may need to balance how much CPU and how much IO time is being used during the compression phase. If so, you can limit when compression is to be used. The command-line option -T *threshold* tells afio not to try compressing a file unless it is at least *threshold* bytes in size. The default setting is -T 0k, so afio tries to compress all files, no matter how small. At the other end of the spectrum, you may want to limit how large a file can be before afio tries to compress it. You can do that with the command-line option -2 *max*, where *max* is the maximum file size. The default in this case is -2 200m, so afio won't try to compress files larger than 200MB.

What if you don't want to use gzip as your compression method? You can change this by using the command-line option -P *progname*, where *progname* is the name of the executable to use to do the compression.

If you need to hand options in to this alternate program, use the -Q *opt* option. You need to use separate -Q options for each option you want to hand in to the alternate program. Because afio simply executes this alternate program, you can run anything at this stage, such as an encryption program, allowing you to encrypt your archive. To encrypt your archive using PGP, execute:

```
export PGPPASSFD=3
find . -name *.c -print | afio -ovz -Z -U -P
→pgp -Q -fc -Q +verbose=0 -3 3 archive 3<passphrasefile
```

This runs PGP on all files in the archive as they are added.

The last afio trick is having the ability to interact with archives on external systems in a similar manner as you would with tar. The format looks like this:

```
[user@host [%rsh] [=afio]:file
```

The option user@ is the user name you would use to access the external system. The default communications mechanism is rsh, but you can change that to ssh with the option %ssh. You can define the command to use on the external system with the option =afio. You can use this if the executable is named something else or in an odd location. So, if you want to archive all your source code files onto an external server over ssh, you could execute:

```
find . -name *.c -print | afio -o -Z user@server%ssh:archive
```

Using afio, go forth and ensure that you have proper backups of all of your important information.

—JOEY BERNARD

Chumby, the Next Generation

If you're unfamiliar with the Chumby, you might want to go back to the May 2008 issue of *Linux Journal*. Daniel Bartholomew showed us all about the cuddly little gadget and explained why we might want one of our very own. The folks over at www.chumby.com still sell the original Chumby device, but they've come out with a new model, the Chumby One. I'm rather fond of the numbering scheme they chose, because that would mean the original Chumby is number zero. If the next model is the Chumby 10, they will get extra geek points!

How does the new revision stack up? Quite frankly, it's great. Although it may have lost the rounded edges and squishy case, the Chumby One adds some welcome features:

- A dedicated volume knob, for quick-and-simple volume control.
- Rechargeable battery for Chumby uninterrupted mobility (battery not included).
- FM radio.
- Beefed-up CPU (454MHz).



I'll admit I always thought a Chumby would be more of a novelty than a useful gadget; however, I was pleasantly surprised to see its rotating "widget" display happily keeps me informed and entertained during my workday. It can stream Pandora (and other) music in the background, display LOLcats and show you the weather. At night, it makes an extremely useful clock radio. Because it has a rechargeable battery, toting it around isn't a problem. If you've been tempted to pick up a Chumby, but were concerned it would just collect dust, you might be pleasantly surprised!

—SHAWN POWERS

They Said It

The nice thing about standards is that there are so many to choose from.

—Andrew S. Tanenbaum

Where is all the knowledge we lost with information?

—T.S. Elliot

For years there has been a theory that millions of monkeys typing at random on millions of typewriters would reproduce the entire works of Shakespeare. The Internet has proven this theory to be untrue.

—Anonymous

Computers make it easier to do a lot of things, but most of the things they make it easier to do don't need to be done.

—Andy Rooney

In the old days, people robbed stage-coaches and knocked off armored trucks. Now they're knocking off servers.

—Richard Power

If you have any trouble sounding condescending, find a UNIX user to show you how it's done.

—Scott Adams

Security vs. Convenience

Although my intent is not to start the next GNOME/KDE-level war, it seems there must be a happy medium between total desktop insecurity and total desktop unusability. Linux offers so many ways to secure data that it's important to realize it's okay for folks to have different needs and desires. Sure, there are some basic security measures we all should take—things like:

- Don't write your password on a sticky note fastened to your monitor.
- Don't leave your e-mail account logged in on a public computer.
- Keep your system updated.
- Do have a password.

- Don't use "password" as your password.

Apart from that, and I'm sure a few other common-sense practices, security is different for different users and different situations. Take the password scenario—it's very good to have a complex password. But, if your screensaver kicks on every three minutes of inactivity and requires you to type that complex password, your security measures have taken you hostage.

Now before I get hate mail (you know who you are, you've likely already started an e-mail to me), let me assure you, I'm not advocating insecure computer practices. What I am advocating is freedom. If I want my laptop to auto-log in, and not lock the screen with the screensaver, as long as it's only my data being exposed, it should be okay. Sadly, when it



comes to freedom, we need to let people have the freedom to do dumb things too. And now if you'll excuse me, I need to try to remember my luggage combination, "1, 2, 3, 4, 5..." .

—SHAWN POWERS

Women in Linux: Interview with Sarah Sharp, Stormy Peters and Valerie Aurora

Software development, and particularly open-source software development, attracts men more often than women. The FLOSS Project has conducted some interesting studies related to gender and open-source software:

- Free/Libre and Open Source Software: Policy Support (FLOSS Project): www.flosspolis.org
- Free/Libre/Open Source Software Report: Gender: Integrated Report of Findings: www.flosspolis.org/deliverables/D16HTML/FLOSSPOLIS-D16-Gender_Integrated_Report_of_Findings.htm
- Free/Libre/Open Source Software Report: Gender: Policy Recommendations: www.flosspolis.org/deliverables/D17HTML/FLOSSPOLIS-D17-Gender_Policy_Recommendations.htm

Here, my goal is not to explain or solve the problem, but instead to talk with three women who are active in open-source development.

AB: Could you please introduce yourselves?

SS: I'm Sarah Sharp, the maintainer for the Linux kernel driver for USB 3.0 host controllers. I work with the Linux USB kernel developers and other Linux developers in Intel's Open Source Technology Center.

SP: I'm Stormy Peters, the Executive Director of the GNOME Foundation. The GNOME Foundation exists to help the GNOME Project fulfill its mission of a free desktop accessible to everyone.

VA: I'm Valerie Aurora, the filesystems developer at Red Hat, consultant and sometimes a writer.

AB: Do you remember the very first time when you heard about Linux and your first experience with it?

SP: I don't remember when I first heard about Linux. I think it was probably around 1996. I do remember when I first heard about Web browsers—I said it sounded like a waste of time. I said the same thing about Twitter. Now I use them both every day and find them very useful.

I actually was more interested in GNOME than Linux. I worked in the HP-UX lab and was looking at seeing whether we could put GNOME on HP-UX. This was an intentional move from CDE to a more-featured GNOME. Later, I helped start the Linux lab at HP.

SS: My boyfriend Jamey was the first one to introduce me to Linux. We were both university students at the time. I was taking a C programming class, and we had to compile our work with gcc on a Linux

system. There were Linux computers on campus, but I wanted to work on my projects from home. I tried using the instructor's directions for SSHing into a box, but the SSH client I was using didn't handle scroll back properly. My boyfriend told me I could install Cygwin on Windows and walked me through installing it. I finally got tired of Cygwin, and he convinced me to dual-boot my computer with Debian Linux.

I do remember asking Jamey a lot of questions and eventually trying to Google for answers in order to learn things independently.

VA: My first experience with Linux was a long time ago! Around 1996, my university started installing Slackware on most of the fast new i386 PCs in the computer lab. I really liked Linux, in part because our previous UNIX machines were Sun pizza boxes running SunOS 4.1.4, and the i386 blew them out of the water. They also didn't have that distinctive annoying hard-disk whine that the Sun boxes had. I used Linux to read e-mail, write programs and browse the Web. I spent a lot of time customizing my window manager.

AB: Do you agree with the statement that universities (scientific institutions) are the best places to be a cradle for technology, like Linux, and eventually promoters of it?

SP: I do remember being very jealous of today's college students who can carry Linux on a laptop back to their dorm rooms. I had to walk across campus to use



Valerie Aurora



Sarah Sharp



Stormy Peters

the Solaris systems we learned on, and nobody could afford one of their own.

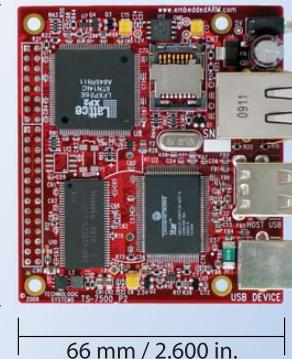
I do think universities are a good place to promote and encourage free software, including Linux. Students look for low cost and good solutions. They then take what they learn to their workplaces.

SS: I believe that universities and colleges are a good place to introduce people to Linux *development*, but you have to start much younger to introduce people to *using* Linux. I would love to see programs to get middle-school-aged girls into using Linux and programming small projects or

TS-7500 Embedded Computer

Faster. Smaller. Cheaper.

Qu. 100 \$84



Powered by a
250 MHz ARM9 CPU

- » Low power, fanless, < 2 watts
- » 64MB DDR-RAM
- » 4MB NOR Flash
- » Micro-SD Card slot - SDHC
- » USB 2.0 480Mbit/s host (2) slave (1)
- » 10/100 Ethernet
- » Boots Linux in less than 3 seconds
- » Customizable FPGA - 5K LUT
- » Power-over-Ethernet ready
- » Optional battery backed RTC
- » Watchdog Timer
- » 8 TTL UART
- » 33 DIO, SPI, I²C

 Dev Kit provides out-of-box development + extra features

- » Over 20 years in business
- » Never discontinued a product
- » Engineers on Tech Support
- » Open Source Vision
- » Custom configurations and designs w/ excellent pricing and turn-around time
- » Most products ship next day



We use our stuff.

visit our TS-7800 powered website at

www.embeddedARM.com

(480) 837-5200

even just installing Linux on older boxes. See creativity-online.com/news/little-girls-get-their-hands-on-arduino-136751 and www.gnomejournal.org/article/88/the-un-scary-screwdriver.

VA: What I do think is that having universities involved in open-source development is excellent for their computer science students' job prospects, and it increases open-source development significantly. In my case, I learned how to develop Linux and open source at my university, New Mexico Tech. I graduated from the university already knowing how to download open-source code, compile it and modify it. Our computer science department as a whole developed some significant portions of the Linux kernel. Working with open source was a key part of my class work every semester, and I can't think of any other way than university support for open source to get four years of experience working on open source before graduating.

What is shocking to me is how seldom university computer science programs take advantage of this opportunity to teach and train students at the same time. I can't believe how often professors teach operating systems with a "teaching OS" instead of Linux or use some proprietary IDE that abstracts away the practical side of programming. As far as I can tell, it is all about the personal preferences of individual professors, and it is a matter of luck whether you get a professor who uses open source.

AB: Can women bring an inspiring atmosphere that positively influences productivity in the workplace?

SP: I think where you work and the atmosphere around you contributes greatly to how happy you are and then to how productive you are. I wouldn't say that only women bring this atmosphere, but rather they bring that diversity in general.

AB: How often do you hear men saying something like: "Wow, you're familiar with Linux! That's great!"?

SP: Are you asking me if males find it surprising that I use Linux? I think males are more surprised when I'm not in a techie group.

SP: Hmm. I'm much more likely to get "GNOME, Linux, what's that?" I used to get "you don't look like a computer person!" I hated that one—especially when they told me that was a compliment.

AB: Do you think this reaction was caused by a computer industry stereotype—only by men and for men?

SP: Absolutely it was caused by a

stereotype. And, instead of feeling like a compliment it felt like they were limiting me.

AB: In your opinion, does the Linux community suffer from a lack of women-contributors?

VA: I have been involved in the Linux community at one level or another for about ten years now and have thought a lot about why women stay away from and/or leave Linux development. One thing I am sure about is that the Linux community suffers from a leak of people who don't want to work in an incredibly hostile environment. Leaving aside arguments about genetically based differences in male and female personalities, men are trained and encouraged to hold their own in aggressive, adversarial and argumentative situations, and women are discouraged from the same. (I was personally trained to be more assertive and challenging than most women, but it's still not a pleasant experience, to say the least.)

In the end, the Linux development culture is heavily biased toward people who thrive on conflict. A widely held belief is that the adversarial culture results in higher-quality software, but that sounds a little bit like rationalization to me.

AB: How do we attract more women to contribute?

SS: So many people have opinions on this. Most of them boil down to this: don't be a jerk. Stop allowing sexist/racist comments on IRC, e-mail or in conference presentations. Criticize the code, not the person. Women really want to get their first submissions "right" the first time, and are discouraged when they don't know everything. I've found it was really good to have a mentor I could trust with my emotions to review code before sending it to a public mailing list. Perhaps a group of women could step up to review code or be mentors for new developers. It's a difficult problem to solve, and I don't think there is one big solution to this.

SP: I think it's best solved by encouraging more women to join through mentors, programs directed at women and general awareness and encouragement in the forums where the community congregates. I think the Free and Open Source community would benefit from women in the community—not only would it add diversity but also twice as many developers! Women would benefit from having more career options—a really fun and rewarding one.

Image credits: Jiang Jiang (Beijing).

—ANTON BORISOV



REUVEN M. LERNER

NoSQL? I'd Prefer SomeSQL

Are non-relational databases the future? Reuven examines the NoSQL movement and what it means for Web apps.

When I started to develop Web applications in the early 1990s, I sometimes found myself needing to store persistent data. At the time, the main options available were files—plain-text files, binary files or DBM (now Berkeley DB) files.

Fortunately, the languages I was using for Web development at the time (mainly Perl and Tcl) made it fairly easy to use any of those options. However, Web developers like myself quickly discovered the problems inherent in these technologies, particularly on large sites. The lack of intelligent, fine-grained locking meant that a popular application would be forced to wait before it could read from, or write to, an external file. Moreover, storing the data in this way prevented us from retrieving it intelligently or selectively. You could, of course, read the data into your program and produce your own reports, but each report required not just its own query, but its own program.

So, you can imagine how my life changed dramatically when I was introduced to relational database systems. Simply put, a relational database is built out of many two-dimensional tables, with each table containing one or more columns (the attributes) and one or more rows (the records). You could relate these tables to one another (thus, making the database "relational") with foreign keys, such that a column in one table would represent a record in another table.

When you store data in a relational database, you suddenly are able to stop thinking in terms of how your data is stored and more in terms of how it is structured. Is the database server keeping information in binary format, in ASCII or in something altogether different? Who knows? And furthermore, who cares? So long as I can store and retrieve my data easily and efficiently, I'll treat the database as a black box, concentrating on writing my application.

SQL, the query language that works with relational databases, encourages this kind of thinking. When you use SQL to query a database, you describe the data you want to retrieve, rather than telling the computer how to retrieve it. True, now there are procedural extensions to SQL, and you also can write programs in non-SQL languages that manipulate the data in further ways. But, even they allow you to loop over, manipulate and perform calculations on the retrieved data. The data itself is still in the

database, and you still describe how and what you want to retrieve in the declarative language of SQL.

Relational databases became popular because they offered so many advantages to programmers. They offered stability and security. They offered transactions, allowing programmers to ensure that a set of operations either all happened atomically or that they were rolled back to their state before the query began to execute. It was fairly easy to map many types of data onto two-dimensional tables. Databases gained all sorts of capabilities, including the ability to check the integrity of the data they are storing. And, they even gained the ability to distribute the load across multiple computers, albeit with some degree of difficulty.

Open-source relational databases, such as MySQL and PostgreSQL, began to challenge their commercial competitors. When I began writing this column in 1996, MySQL had not yet been released under an open-source license, and although PostgreSQL was fully open source, it had a number of limitations. So, it was expected that if you were interested in using a relational database, you almost certainly would fork over money for one, paying Oracle or one of its smaller competitors many thousands of dollars. Today, the opposite is the case. Open-source databases are more than adequate for Web applications. Moreover, it is basically expected that a Web application will use a relational database as a back end. If you say you are a Web developer, it is basically expected that you have some experience with SQL, as well as the care and feeling of a relational database.

So What's Wrong?

I've been using relational databases for years, and I'm quite comfortable with them. Indeed, as I've sometimes told my clients, I see the world through two-dimensional tables. Breaking problems down into tables and the relations among them is something I do nearly every day. Yet, even I recognize that a database is a tool, and that not every tool is appropriate for every task. Saying every Web application should use a relational database is like saying all programs should be written in C (or whatever your favorite programming language is at the moment). As the old adage goes, if your only tool is a hammer, every problem looks like a nail.

And, there are indeed some issues with relational databases that have come to the foreground during the past few years. Perhaps the biggest problem many developers face is the "impedance mismatch" between objects, with which we enjoy writing our programs, and tables, in which relational databases function. If you want to store your object in a database, you have several options, none of which are particularly good, and all of which lose some of the semantic value that is the whole point of objects. True, you can treat each object attribute as a column, each instance as a row and each class as a table, but many corner cases don't map well to this, even with an excellent object-relational mapper (ORM) library. One of the reasons I enjoy working with Ruby on Rails is the large degree to which ActiveRecord is able to paper over these differences, but even with all of ActiveRecord's smarts, some problems can't be solved easily with objects, where it's required to massage my data model or even to drop down into SQL.

This problem has become bigger in the past few years, with the growth of documents and objects that contain many different types of data, some of which are unpredictable or contain highly variable content. In response to these needs, database servers have begun to incorporate functions that previously

were considered to be outside the database domain, such as full-text search (which is extremely useful) and XML handling (which I still find puzzling and somewhat inappropriate). Regardless of your attitude toward such features, it does seem a bit strange to set up a database server just so you can store and retrieve files. There must be better ways to store such files, without all the administrative and computational overhead associated with a database server.

Another problem has to do with the greater and greater performance requirements of Web applications. I tend to dismiss talk of "scalable" languages and frameworks, in part because I believe that any language or framework can scale, given sufficient hardware. Instead of worrying about the scalability of a particular language, I would concentrate on the scalability of an architecture—and that is certainly a weak point for relational database servers. Modern Web applications using a "share-nothing" approach easily can scale up to hundreds or even thousands of Web servers, each handling part of the load. But if each of those servers needs to communicate with a database server on the back end, the database might become overwhelmed at some point. There are solutions to this problem, even among the open-source databases, but they are far from dead simple to install and configure.

Fanless HD Media Players with Ubuntu Linux

Ruggedized NVIDIA® ION™ System

No fans, no moving parts. Just quiet, reliable operation.
NVIDIA® GeForce® 9400M graphics for 1080p playback.



Low-Profile Intel® Atom™ Embedded System

Small footprint platform featuring solid state USB storage.
Broadcom Hardware Decoder enables 1080p support.

Value only an Industry Leader can provide.

Selecting a complete, dedicated platform from Logic Supply is simple: Pre-configured systems perfect for both business & desktop use, Linux development services for greater system customization, and a wealth of online resources all within a few clicks.

Learn More > www.logicsupply.com/linux

NoSQL? I'd Prefer SomeSQL

The rise of increasingly variable document types, combined with the scalability issues associated with Web applications, has led many in the Web developer community to push for NoSQL, a variety of solutions that are united only in their rejection of the relational database model. The term itself was coined in 2009 by Eric Evans, who wanted to describe the collection of non-relational tools that were rising in popularity. As I write this in early 2010, we are seeing a veritable NoSQL revolution, with a number of prominent Web developers and sites switching away from relational database servers to other systems.

The reasons for the switch to these various NoSQL systems vary with the site and developer, but they echo the reasons that I outlined above: a good match with the objects used to program the application, the flexibility that a document-oriented database provides and the easier replication that a NoSQL system offers. These systems are far from identical in approach, communication or performance, but they each have their own backers and are released under an open-source license. Besides, many of these systems are currently in use on high-performance Web sites, such as Facebook and LinkedIn, so if your site is smaller than those, the odds are good that these solutions will work for you too. Another advantage is that the most popular of these systems offer a number of language bindings, such that it doesn't matter whether you're using Ruby, Python, Java or PHP.

The names I hear most often when discussing NoSQL solutions are CouchDB and MongoDB, with others (such as Tokyo Tyrant) gaining a great deal of mention. CouchDB came to prominence as much for its implementation in Erlang and its use of JSON as a query and response format, as for its database capabilities. CouchDB allows you to use the famous map-reduce algorithm on your database, write functions in JavaScript and send queries over HTTP using REST. MongoDB, by contrast, is written in C++ and uses its own object-oriented system (BSON) to handle queries. I'm still mulling over the differences between the two and trying to understand where each of them might have an advantage.

There is no doubt in my mind that these sorts of NoSQL solutions might be very useful and do indeed solve many problems. But, a large part of me wonders if we really should be throwing away 30 years of accumulated knowledge about relational databases. Moreover, relational databases were designed to ensure data integrity, not just the fast storage and retrieval of data. And, although these modern NoSQL solutions might indeed be fast and flexible, their lack of integrity checking does worry me somewhat. Maybe I'm old-fashioned, but I like the fact that I can define a relational database table

to be NOT NULL or UNIQUE and, thus, know that the data contained within exists, is unique or both. I like knowing that each record in my database has a unique identifier. I like being able to traverse foreign keys, such that when one record points to another, I know there will be something on the other end, rather than the database equivalent of a null pointer. I'm also curious to see how such databases avoid duplication of data, or what the NoSQL equivalent of "normalization" might be.

So, I believe the people who think that NoSQL is a complete solution for all application storage needs are exaggerating quite a bit and are ignoring a great deal of wisdom associated with relational databases. Databases work so well, and are so easy to work with nowadays, that it seems a shame to put them out to pasture because they are having trouble handling loads of the top tier of Web applications.

Indeed, there are a variety of problems for which NoSQL offerings seem to have no solution, at least for now. They don't pay any attention to data integrity, be it a lack of null data or ensuring that only certain values will be allowed to be stored. They don't offer any way of associating objects to one another on multiple dimensions, which I find to be one of the most attractive elements of the relational model. And the query languages are far from standardized, meaning that moving from one NoSQL solution to another requires translating the query from one language to another.

That said, there is a growing class of Web applications for which relational databases are a poor fit and for which a document-oriented approach might indeed be superior. My dissertation software, which works with a large number of text documents, might well be a good example of a program that would benefit from a document-oriented approach, and I already have begun to explore such alternatives. However, even if I switch part of my software to use a NoSQL solution, it will be for a portion of the data storage, not for the entirety. In this way, I hope to benefit from the best of both worlds, using a relational database where it makes sense and a document database where it provides the superior solution.

My feeling is that the NoSQL movement, at least in the image its name projects, is a bit extreme and fails to appreciate the advantages and sophistication of many relational solutions. I would prefer a more nuanced approach, which might be called SomeSQL, in which non-relational databases are seen as additional tools, which can (hopefully) be integrated into a larger data-storage solution. After all, memcached and relational databases have managed to live in harmony for a number of years already. With a bit of planning, and with the right tools in

place, I would argue that a combination of SQL and NoSQL could be extraordinarily powerful.

There is, of course, the chance that the NoSQL people are completely right, and that we are seeing the decline of one technology in favor of another. Perhaps we're at the leading edge of a new revolution, the database equivalent of the growth of high-level "scripting" languages in the 1990s. But, I'd prefer to see this as a new technology that Web developers will have to fit into their toolboxes, and which they will need to understand when designing and writing new Web applications.

Over the coming months, I plan to explore some of the better-known NoSQL solutions. I will look at them from the perspective of a developer—what do they offer, and why would I want to use this over a relational database or its NoSQL rivals? I cannot yet say which of these technologies will be my favorite or which would be appropriate for your applications. But, I do believe that the SomeSQL approach is one that has the potential to improve our applications' flexibility and performance dramatically, at the expense of the data-storage simplicity to which we have grown accustomed during the past few decades.■

Reuven M. Lerner, a longtime Web/database developer and consultant, is a PhD candidate in learning sciences at Northwestern University, studying on-line learning communities. He recently returned (with his wife and three children) to their home in Modi'in, Israel, after four years in the Chicago area.

Resources

It's hard to summarize all of the writing that has been done about NoSQL to date. Many videos and presentations from the NoSQL conference in mid-2009 are linked to from Johan Oskarsson's blog posting at blog.oskarsson.nu/2009/06/nosql-debrief.html.

Additional information about CouchDB is available at couchdb.apache.org, and additional information about MongoDB is at mongodb.org.

For some interesting comments reacting to (and rejecting) the NoSQL movement, you might want to read one or more of the following: www.eflorenzano.com/blog/post/my-thoughts-nosql, cacm.acm.org/blogs/blog-cacm/50678-the-nosql-discussion-has-nothing-to-do-with-sql/fulltext, codemonkeyism.com/dark-side-nosql and bjclark.me/2009/08/04/nosql-if-only-it-was-that-easy.



visit us at www.siliconmechanics.com
or call us toll free at 866-352-1173

As a Senior Account Executive for Silicon Mechanics, Michael collaborates with customers to expertly match hardware with processing needs. Lately he has been inviting a good many of those customers to have a close look at the Hyperform HPCg R2504, powered by NVIDIA Tesla. This workstation has earned its place among our most popular products for very good reasons.

We start with two Quad-Core Intel® Xeon® Processor 5500 Series CPUs, for fast, reliable, energy-efficient processing. Then we add up to four NVIDIA Tesla C1060 GPUs, to dramatically accelerate parallel processing for applications like ray tracing and finite element analysis. With dual-IOH design, the system provides non-blocking connectivity between the GPUs and CPUs to maximize system performance. Include up to 96GB of registered DDR3 memory and you have the power of a cluster in a workstation form factor at a price you don't want to miss.

When you partner with Silicon Mechanics, you get more than collaborative service and affordable performance—you get an expert like Michael.

For configuration and pricing on the
Hyperform HPCg R2504 visit
www.siliconmechanics.com/R2504



**Powerful.
Intelligent.**



Expert included.

Silicon Mechanics and the Silicon Mechanics logo are registered trademarks of Silicon Mechanics, Inc. Intel, the Intel logo, Xeon, and Xeon Inside, are trademarks or registered trademarks of Intel Corporation in the US and other countries.



DAVE TAYLOR

Our Twitter Autoreponder Goes Live!

Some final fixes to the Twitterbot script we developed last month.

I **can't believe it**, this is my 52nd column. That means I've been writing for *Linux Journal* for almost four and a half years. Hopefully, you've been reading my column just as long and enjoying our monthly forays into the world of shell script programming. On the tech side, quite a bit has changed in the last four and a half years. But on the Linux/shell side, it's surprisingly similar to how it was when I wrote my first column.

Last month, we continued to build a Twitter autoreplier script that could read and parse Twitter messages (aka tweets). We got it working and wrapped up the column by realizing we actually needed to capture the unique tweet ID in addition to name and message, so we could ensure that the script kept track of what it had or hadn't answered.

Where We Are Now

The script keeps track of tweets by ID and knows both how to parse the incoming Twitter stream and how to remember if it has seen a one-word tweet request or not. Run it once, and I see:

```
Twitter user @jlight asked for the time
@jlight the time on our server is LOCALTIME
```

The next time I run it, just a few minutes later, I see:

```
Twitter user @truss asked for the time
@truss the time on our server is LOCALTIME
Twitter user @tlady asked what our address in tweet 7395272164
@tlady we're located at 123 University Avenue, Anywhere USA
```

It looks good, but there's a problem in the script, because one of the output diagnostic lines is:

```
Twitter user @ asked for the time
@ the time on our server is LOCALTIME
```

Somehow it's not identifying the user ID for this particular user. After a quick analysis of the actual Twitter.com data, it appears that the first tweet comes out of the parser section without an associated user ID.

To debug this, first get a copy of the script to follow along (the script from last month is at <ftp://linuxjournal.com/pub/lj/listings/issue191/10695.tgz>). In the while loop, I'll add this line to aid in debugging:

```
echo got name = $name, id = $id, and msg = $msg
```

Now when I run the script, here's what I see:

```
got name = , id = 7395437583, and msg = VERY cool
got name = spin, id = 7395333666, and msg = time
got name = astrong, id = 7395281516, and msg = time
got name = truss, id = 7395281011, and msg = time
```

Clearly something's wrong, but what?

Debugging a Complicated Script

One reason I like to use temp files in scripts rather than having incredibly long and complicated pipes is for debugging this sort of problem.

Recall that the main parsing work is done by curl feeding its output to grep, then a sequence of sed invocations and finally a quick call to awk:

```
$curl -u "davetaylor:$pw" $inurl | \
grep -E '<screen_name>|<text>|<id>' | \
sed 's/@DaveTaylor //;s/ <text>//;s/<\/text>//' | \
sed 's/ *<screen_name>//;s/<\/screen_name>//' | \
sed 's/ *<id>//;s/<\/id>//' | \
awk '{ if (NR % 4 == 0) {
        printf ("name=%s; ", $0)
    }
    else if (NR % 4 == 1) {
        printf ("id=%s; ", $0)
    }
    else if (NR % 4 == 2) {
        print "msg=\"\" $0 \"\""
    }
}' > $temp
```

Adding the command `more $temp` immediately after this means we can eyeball the data stream and see what's different about the first and second lines

(as the second is parsed properly). Here's what I see:

```
id=7395681235; msg="African or European?"  
name=jeffrey; id=7395672894; msg="North Hall IStage"
```

Note that there's no name= field on the first message. My theory? There's a logic error in the awk statement that's causing it to skip the first entry somehow.

To test that assumption, I'll temporarily replace the entire awk script with another that outputs the record number (mod 4) followed by the data line:

```
awk '{ print (NR % 4), $0 }' > $temp
```

The result is exactly what we were expecting, which is a bit confusing:

```
1 7395934047  
2 we are at the MGM as well!  
3 14171725  
0 sideline  
1 7395681235
```

```
2 African or European?  
3 14712874  
0 jeffrey
```

Here, Twitter user sideline has sent "we are at the MGM as well!", and jeffrey sent the message "African or European?".

What's Really Wrong?

The problem isn't that the data is being eaten, it's that the awk script is pairing the name information with the wrong tweet. Let's re-examine the awk script:

```
awk '{ if (NR % 4 == 0) {  
    printf ("name=%s; ", $0)  
}  
else if (NR % 4 == 1) {  
    printf ("id=%s; ", $0)  
}  
else if (NR % 4 == 2) {  
    print "msg=\"\" $0 \"\""  
}  
}'
```



visit us at www.siliconmechanics.com
or call us toll free at 866-352-1173

Meet Victoria (on the right). She is the Silicon Mechanics marketing expert responsible for the events and promotions that keep our customers informed about new products and technologies. She's pictured here with her twin sister Veronica, an industrial designer, to help us make a point about what makes twin servers so popular. Victoria and Veronica are twins, but they don't look exactly alike and they don't do the same job. Twin servers are two servers in a single 1U chassis: they can be configured differently, and they handle their own individual workloads.



For more information about the
Rackform iServ R4410 visit
www.siliconmechanics.com/R4410



**Powerful.
Intelligent.**

With the introduction of the Rackform iServ R4410 from Silicon Mechanics, twin power has reached a whole new level: the twin². A twin² is a 2U 4-node system. It supports four swappable, full-featured nodes in a 2U chassis with redundant power. In each node you'll find 2 of the new Intel® Xeon® 5500 Series processors, 12 DDR3 DIMM slots, 3 hot-swap drives, and an integrated dual-port GigE adapter. Integrated InfiniBand is also available with the R4410-IB. Unmatched density and state-of-the-art processors make the R4410 a superior choice for high-performance computing, and Victoria is spreading the word with enthusiasm.

When you partner with Silicon Mechanics, you get more than the latest and greatest in density, performance, and energy efficiency—you get an expert like Victoria.

Expert included.

Silicon Mechanics and the Silicon Mechanics logo are registered trademarks of Silicon Mechanics, Inc. Intel, the Intel logo, Xeon, and Xeon Inside, are trademarks or registered trademarks of Intel Corporation in the US and other countries.

`NR%4=0` is correctly tagged as the name, `NR%4=1` as the message ID, `NR%4=2` as the msg, and `NR%4=3` is skipped. (It's the Twitter user ID, not the tweet ID. It might be useful in a different context, but not for what we're doing.)

The problem is subtle, but it becomes obvious when you compare what the parser is generating against the actual tweets in the Twitter stream. We saw the first two like this:

```
id=7395681235; msg="African or European?"  
name=jeffrey; id=7395672894; msg="North Hall IStage"
```

But in fact, the tweet "African or European?" was sent by jeffrey, and the "North Hall IStage" was

The problem is subtle, but it becomes obvious when you compare what the parser is generating against the actual tweets in the Twitter stream.

sent by the user identified in the subsequent line of parsed and formatted data.

Conclusion? We're splitting the data lines in the wrong place. Instead of adding the carriage return after `NR%4==2` (it's subtle, we use `print` instead of `printf`), we actually should be adding it after the match for `NR%4==0`, like this:

```
awk '{ if (NR % 4 == 0) {  
    printf ("name=%s;\n", $0)  
} else if (NR % 4 == 1) {  
    printf ("id=%s; ",$0)  
} else if (NR % 4 == 2) {  
    printf ("msg=\"%s\"; ", $0) }  
'
```

Now, let's try that statement again:

```
id=7395681235; msg="African or European?"; name=jeffrey;  
id=7395672894; msg="North Hall IStage"; name=sideline;
```

Ah, that's the ticket!

Fixed It, Now Let's Clean Everything Up

With the problem solved, I'll remove the added debug statements and unleash the listener beast:

```
got name = jeffrey, id = 7395681235, and msg = African or European?  
got name = sideline, id = 7395672894, and msg = North Hall IStage  
got name = Genuine, id = 7395669466, and msg = ummmmm I know
```

Perfect. Bug debugged!

Now when we run the script, it correctly sees only the new tweets since it was last run, and it responds only to those it understands:

```
Twitter user @Larkin asked for the time
```

```
@Larkin the time on our server is LOCALTIME  
Twitter user @jennyj asked for the time  
@jennyj the time on our server is LOCALTIME
```

Run the script again, and it sees only what's newer yet:

```
Twitter user @NoA asked for directions in tweet 7396527668  
@NoA directions to our office are here: SOMEADDRESS
```

Perfect! Now, a tiny tweak. As we've debugged things, I have set the variable `tweet` to `/bin/echo`, so as not to flood my followers with unnecessary messages. Change it back to the `tweet.sh` script (as developed in an earlier

series of columns), and the script actually responds with tweets.

The first run looks like this:

```
$ sh tweet-listen.sh  
Twitter user @mosa asked for directions in tweet 7396566048  
(sent tweet @mosa directions to our office are here: SOMEADDRESS)  
Twitter user @xwatch asked for the time  
(sent tweet @xwatch the time on our server is TIME)  
Twitter user @NoA asked for directions in tweet 7396527668  
(sent tweet @NoA directions to our office are here: SOMEADDRESS)
```

To ensure that it won't answer more than once to a tweet query, I'll run the script again:

```
$ sh tweet-listen.sh  
$
```

That's it! Now one tiny additional task is left, to add it to crontab so that it'll be an active listener, which is done by having it run every two minutes with the line:

```
*2 * * * * bash $SCRIPTS/davesbot/tweet-listen.sh
```

That's all there is to it. Congratulations, we've just built a fully featured Twitterbot.

If you'd like to test it, it has its own account on Twitter, `@davesbot`. Start by sending a 2–3 word message, and it'll tell you what it can do. Grab the final source code from the LinuxJournal.com site at ftp.linuxjournal.com/pub/lj/listings/issue192/10711.tgz.■

Dave Taylor has been hacking shell scripts for a really long time. He's the author of the popular *Wicked Cool Shell Scripts* and can be found on Twitter as `@DaveTaylor` and more generally at www.DaveTaylorOnline.com.

O'REILLY®

MySQL

Conference & Expo

APRIL 12–15, 2010

Santa Clara Convention Center
Santa Clara, California

Information Unleashed

The 6th annual **MySQL Conference & Expo**, presented by O'Reilly Media, happening April 12 – 15, 2010 in Santa Clara, CA, brings over 2,000 open source and database enthusiasts together to harness the power of MySQL and celebrate the huge MySQL ecosystem.

Take advantage of this rare opportunity to meet face to face with a huge core group of engineers who are in the process of developing MySQL. Share ideas and get your toughest questions answered from MySQL experts. Bring your team, your laptops, and leave with new insights in how to build high performance scale-out applications using MySQL.

The quality of the sessions and tutorials at the **MySQL Conference & Expo**, is very high. Year after year, the most frequent comment the conference team receives from attendees is that it is sometimes very difficult to choose which great simultaneous session to attend. From novice to expert-level sessions, developer to DBA, business to technical, the sessions provide expert insight into solving architectural, design, and development issues your organization faces.



Register Now and Save 15% when you use discount code: mys10ljr
conferences.oreilly.com/mysql



MICK BAUER

Linux VPNs with OpenVPN, Part III

Secure remote networking, the OpenVPN way.

In my previous two columns, I began a series on building Linux-based Virtual Private Network (VPN) solutions using OpenVPN. When I left off last time, I had gotten as far through the OpenVPN server configuration process as creating a simple Public Key Infrastructure (PKI), using it to generate server and client certificates, and creating a few other "support" files involved in building OpenVPN tunnels. In so doing, I worked my way down just the first third or so of the example OpenVPN server configuration, but those PKI/crypto-related configuration parameters represent the most complicated part of OpenVPN configuration tasks.

This month, I describe the rest of that server

Listing 1. Server's server.ovpn File

```
port 1194
proto udp
dev tun

ca 2.0/keys/ca.crt
cert 2.0/keys/server.crt
key 2.0/keys/server.key # This file should be kept secret
dh 2.0/keys/dh1024.pem
tls-auth 2.0/keys/ta.key 0

server 10.31.33.0 255.255.255.0
ifconfig-pool-persist ipp.txt
push "redirect-gateway def1 bypass-dhcp"

keepalive 10 120

cipher BF-CBC      # Blowfish (default)
comp-lzo
max-clients 2

user nobody
group nogroup
persist-key
persist-tun

status openvpn-status.log
verb 3
mute 20
```

configuration file and show a corresponding OpenVPN client configuration file (which I'll dissect next month). I also show how to start both server and client processes, although debugging, firewall considerations and other finer points also will need to wait until my next column.

Have no fear—I think you'll find this installment to be plenty action-packed in its own right. Let's get to it!

OpenVPN Server Configuration, Continued

Normally at this point in a multipart series, I'd review at least some details from the prior month's column, but that won't work this time. Last month's article covered a lot of ground, and this month's needs to cover still more. Suffice it to say that I began dissecting an example OpenVPN server configuration file, /etc/openvpn/server.ovpn (Listing 1).

I got as far as generating the files referenced in the ca, cert, key, dh and tls-auth lines, using a combination of OpenVPN "easy-rsa" helper scripts (located in /usr/share/doc/openvpn/examples/easy-rsa/2.0) and the commands openvpn and openssl. I'm going to continue describing Listing 1's parameters, assuming that the aforementioned certificate, key and other helper files are in place.

So, having set up the basic port/protocol/device settings and cryptography-related settings, let's move on to settings that will determine what happens once a client successfully establishes an authenticated, encrypted tunnel. The first such setting is server.

server actually is a helper directive. It expands to an entire block of other parameters. Rather than slogging through all those additional parameters, let's just say that the server directive takes two parameters: a network address and a netmask. For each tunnel established by clients on the port we specified earlier, the OpenVPN server process will carve a little 30-bit subnet from the specified IP space, assign itself the first host IP address in that subrange as its local tunnel endpoint and assign the other host IP in the 30-bit subnet to the connecting client as its remote tunnel endpoint.

In the example, I've specified the network address 10.31.33.0 with a netmask of 255.255.255.0,

which translates to the range of IP addresses from 10.31.33.1 to 10.31.33.254. When the first tunnel is established, the server will use 10.31.33.1 as its local tunnel endpoint address and assign 10.31.33.2 to the client to use as the remote tunnel endpoint address. (10.31.33.0 is that subnet's network address, and 10.31.33.3 is its broadcast address.)

For the next client to connect, the server will use 10.31.33.5 as its tunnel endpoint and will assign 10.31.33.6 as the client's tunnel endpoint (with 10.31.33.4 and 10.31.33.7 as network and broadcast addresses, respectively). Get it?

This isn't the most efficient use of an IP range. The server needs a different local IP address for each tunnel it builds, and for each tunnel, the server essentially wastes two others (for network and broadcast addresses). Preceding the server directive with the line `topology subnet` will cause the server to use the first IP in its `[server [network address] [netmask]]` range for its local tunnel IP for *all* tunnels and for client tunnel-endpoint IPs to be allocated from the *entire* remainder of possible IPs in that range, as though all remote tunnel endpoints were IP addresses on the same LAN.

This isn't the default behavior, because it's new to OpenVPN 2.1. The "subnet" topology isn't supported by earlier versions or by Windows clients using version 8.1 or lower of the TAP-Win32 driver. Note that if undeclared (as in Listing 1), the `topology` parameter has a default value of `net30`, which results in server's specified IP range being split up into 30-bit subnets as described above.

Continuing on in Listing 1, next comes `ifconfig-pool-persist`, which specifies a file in which to store associations between tunnel clients' Common Names (usually their hostnames, as specified in their respective client certificates) and the IP addresses the server assigns to their tunnels. Although this doesn't guarantee a given client will receive the same tunnel IP every time it connects, it does allow clients to use the `--persist-tun` option in their client configurations, which keeps tunnel sessions open across disruptions in service (OpenVPN server daemon restarts, network problems and so forth).

Next comes the statement `push "redirect-gateway def1 bypass-dhcp"`. The `push` directive causes the subsequent double-quotation-mark-enclosed string to be run on the client as though it was part of the client's local configuration file. In this case, the server will push the `redirect-gateway` parameter to all clients, with the effect that each time a client connects, the client's local default gateway, DNS servers and other network parameters normally provided by DHCP will be overridden by the server's settings for those things.

This effectively enforces a VPN policy of "local-subnet-only split tunneling". For those of you new to VPNs, a split tunnel configuration is one in which clients can use their VPN tunnels to connect to some things and their local (non-tunneled) Internet connection to connect to other things.

As I've said in previous columns though, forcing clients to use the *remote* network's infrastructure (DNS servers, Internet uplink and so forth) makes it much harder for attackers connected to a client's local network, which might be an untrusted environment like a coffee-shop wireless hotspot, to perform various kinds of eavesdropping, session-hijacking and man-in-the-middle attacks.

Even with this setting, a client still will be *able* to connect to some things on the local network. It just won't be able to use it as a *route* for anything but connecting back to your OpenVPN server. Again, it's good policy to configure clients to leverage as much of your trusted network's infrastructure as possible.

After the `push "redirect-gateway..."` directive comes `keepalive 10 120`. Like `server`, `keepalive` is a helper directive that expands to a list of other parameters. Again for the sake of brevity, let me summarize the effect of the example line: every ten seconds, the server will check to see that each client is still connected, and if no reply is received from a given client over any 120-second period, it will assume that client is unreachable at its last known IP address.

For example, if the server sends a query to a particular tunnel client at 9:00:00 and gets a reply, but another at 9:00:10 to which there's no reply, and also receives no reply to any of 11 more queries sent out at 9:00:20, 9:00:30 and so on until 9:02:00, then at 9:02:00 (after 120 seconds of no replies), the server will conclude the client system is unreachable.

At this point, the server will attempt to re-resolve the remote client's name, on the assumption that its IP address may have changed (due to DHCP lease renewal, for example) and, thus, re-establish the tunnel session.

The aforementioned infrastructure settings, such as DNS servers, by the way, will be read by the server's `openvpn` process from `/etc/resolv.conf`, the server's running routing table and so forth—no OpenVPN configuration parameters are necessary for those settings unless you want them to be different from the server's. (For now, let's assume you don't!)

I just spent a fair amount of ink on only a handful of settings. But I think this is warranted given that `server` and `keepalive` are helper directives that expand to many more settings and given that we're now done with the network configuration portion of our server configuration.

The next parameter is a simple one: `cipher BF-CBC`, which specifies that each tunnel's data payload will be encrypted with the Blowfish cipher, using 128-bit keys, in Cipher Block Chaining mode (CBC mode makes it harder for an attacker to brute-force-decrypt isolated parts of a given session). BF-CBC is the default setting for `cipher`, so technically, I don't need to specify this, but it's an important setting. You can use the command `openvpn --show-ciphers` to see a list of all supported cipher values and their default key sizes.

`comp-lzo` is even simpler. It tells OpenVPN to compress all session data using the LZO compression algorithm, unless a given portion of data appears to be compressed already (for example, if a JPEG image or a ZIP file is being transferred), in which case OpenVPN won't compress until it detects a return to noncompressed session content. This adaptive behavior helps minimize the data padding that results from trying to compress already-compressed data. Because LZO is a fast algorithm, this is a good setting. Its cost in CPU overhead is generally more than compensated for by the amount of network bandwidth (and, thus, other CPU cycles) it conserves.

The next setting, `max-clients 2`, specifies that a maximum of two tunnels may be active at one time. If you have only one or two users, there's no good reason to allow more than one or two concurrent tunnels. In my own testing, however, I've found that setting this all the way down to 1 can cause problems even if you have only one user, probably due to how OpenVPN handles tunnel persistence (see `keepalive` above).

The next four settings are interrelated. `user` and `group` specify the names of an unprivileged user account and group (`nobody` and `nogroup`, respectively), for the OpenVPN server daemon to demote itself to after opening necessary tun/tap devices, reading its configuration file, certificates and keys, and other root-only startup actions.

For this to work properly, you also need to set `persist-key` and `persist-tun`. `persist-key` causes OpenVPN to keep key file contents cached in memory across daemon interruptions (like those caused by tunnels being broken and re-established). `persist-tun` causes OpenVPN to keep any tun/tap devices that were opened on startup, open across the same kinds of daemon restarts.

With `user` and `group` set to unprivileged user and group, if you were to skip declaring `persist-key` or `persist-tun`, the OpenVPN daemon would lack the necessary privileges to re-read protected key files or re-open the tun or tap device.

You could, of course, skip the user and group settings. Those settings, however, lessen the impact of some unforeseen buffer-overflow vulnerability. It

can make the difference from an attacker gaining an unprivileged shell and gaining a root shell. Unfortunately, you can't assume that just because OpenVPN has had a good track record so far with respect to lacking many significant security vulnerabilities, that it never will have any!

The last three settings in Listing 1 concern logging. `status` specifies a file to which OpenVPN will write daemon status updates periodically, regardless of actual activity. Unlike most log files, each time this file is updated, OpenVPN will overwrite the previous message. This is what the file `/etc/openvpn/openvpn-status.log` on my OpenVPN server says right now:

```
OpenVPN CLIENT LIST
Updated,Fri Jan 1 21:55:11 2010
Common Name,Real Address,Bytes Received,Bytes Sent,Connected Since
minion2,192.168.20.1:36491,125761,103329,Fri Jan 1 17:56:21 2010
ROUTING TABLE
Virtual Address,Common Name,Real Address,Last Ref
10.31.33.6,minion2,192.168.20.1:36491,Fri Jan 1 20:54:03 2010
GLOBAL STATS
Max bcast/mcast queue length,0
END
```

As you can see, there's only one client currently connected (`minion2`), with one corresponding route table entry.

Moving on back in Listing 1, verb 3 sets the overall logging-verbosity level to 3 out of a possible range of 0 (no logging except major errors) and 11 (the most verbose debugging output possible). The default value is 1, but 3 is much more useful for getting things set up and working properly, without presenting any particular danger of log files growing too huge too quickly.

This is especially true with `mute 20` set, which tells OpenVPN never to log the same message (in a given event category) more than 20 times in a row.

On my Ubuntu system, OpenVPN writes all its messages to `/var/log/daemon` if the `openvpn` command is executed with the `--daemon` flag, which causes it to run as a background (daemon) process. If you run `openvpn` without `--daemon`, it runs in the foreground and logs all messages to the console or terminal window you started it in (tying up that console in the process, but this is a very handy way to run OpenVPN during initial setup and testing).

Running OpenVPN as a Server Daemon

Now that I've covered a sample server configuration file in depth, let's fire up our OpenVPN daemon in server mode! This, as you'll see, is the easy part.

OpenVPN uses a single command, `openvpn`, for everything. Precisely what any given OpenVPN instance does depends on how you start it. As

you've already seen, some startup parameters, like `--show-ciphers`, cause the `openvpn` command to give certain information and then exit. Other parameters tell it to remain active, listening for incoming client connections (`--mode server`) or attempting to establish and maintain a tunnel to some server, as a client (`--mode client`).

If you execute `openvpn` with the `--config` parameter followed by the name of a configuration file, OpenVPN will start itself configured with all parameters in that file. For example, you could create a configuration file containing just the parameter `show-ciphers` (parameters must start with a `--` if specified in a command line, but the `--` is omitted for all parameters within configuration files).

More commonly, as with Listing 1, we use configuration files for server-mode and client-mode startup. I mentioned that the `server` helper directive expands into a list of other parameters; the first of these is `mode server`.

Thus, to start OpenVPN as a persistent server daemon running the configuration file `/etc/openvpn/server.ovpn`, shown in Listing 1, use this command:

```
sudo openvpn --config ./server.ovpn
```

Note the relative path for the file `server.ovpn`. If that file resides in `/etc/openvpn`, you'd need to run the above command from within that directory. Note also the use of `sudo`. On non-Ubuntu systems, you might instead `su` to root before running this command. Regardless, OpenVPN must be run as root in order to read its server key file, to open the tun device and so forth, even though as configured in Listing 1 it subsequently will demote itself to user `nobody` and group ID `nogroup`.

Did you notice I omitted the `--daemon` flag on that command line? Again, you can use that flag to tell OpenVPN to run in the background (like a quiet, well-behaved daemon) and log its messages to `/var/log/daemon.log`, but you first may want to make sure everything's working properly.

Configuring the Client

At this point, I had hoped I'd be able to give you a detailed walk-through of client configuration, but I'm out of space for now, so that will need to wait until next time. But, I won't leave you completely hanging. Listing 2 shows a sample client configuration file, `client.ovpn`, that corresponds to Listing 1's `server.ovpn` file.

Much of this should be familiar. Other parts you can figure out via the `openvpn(8)` man page. In the meantime, feel free to experiment. To run OpenVPN in client mode on a client computer,

Listing 2. Client's `iwazaru.ovpn` File

```
client
dev tun
proto udp

remote 1.2.3.4 1194

resolv-retry infinite
nobind

user nobody
group nogroup
persist-key
persist-tun

mute-replay-warnings

ca ca.crt
cert minion.crt
key minion.key

ns-cert-type server
tls-auth ta.key 1

cipher BF-CBC
comp-lzo

verb 3
mute 20
```

use this command:

```
sudo openvpn --config ./iwazaru.ovpn --daemon openvpn-client
```

One parting tip for you experimenters: you'll need to disable or reconfigure any local iptables (firewall) rules you've got running on either your server or client systems. I'll discuss iptables considerations in the next column in this series, and I'll continue where we left off this time. Until then, be safe!■

Mick Bauer (darth.elmo@wiremonkeys.org) is Network Security Architect for one of the US's largest banks. He is the author of the O'Reilly book *Linux Server Security*, 2nd edition (formerly called *Building Secure Servers With Linux*), an occasional presenter at information security conferences and composer of the "Network Engineering Polka".

Resources

Official OpenVPN Home Page: www.openvpn.net

Ubuntu Community OpenVPN Page:
<https://help.ubuntu.com/community/OpenVPN>



KYLE RANKIN

Linux Troubleshooting, Part II: Local Network

Last month, I discussed localhost troubleshooting, and this month, I extend troubleshooting to your local network. Find out why shawn can't talk to bill.

This column is the second in a series dedicated to one of my favorite subjects: troubleshooting. Because my column is generally aimed more at tips and tricks and less on philosophy and design, I'm not going to talk much about overall approaches to problem solving. Instead, in this series, I describe some general classes of problems you might find on a Linux system, and then I discuss how to use common tools, most of which probably already are on your system, to isolate and resolve each class of problem.

In the first column, I talked about how to diagnose high-load issues on a server, but the fact is that these days, just about every Linux computer is connected to a network, and a large number of the problems you have are based in the network. This month, I focus on local network troubleshooting, and although I am writing from the perspective of servers, most of these steps will apply to any Linux machine on a network. Also, because the goal of this article is to show how to become better at troubleshooting, I list each step from the lowest level on up. In real life, I'd probably skip ahead here and there to make the troubleshooting process faster.

bill Is Down

The generic problem I cover here is how to track down the root cause when one machine can't communicate with another machine on the same network. For this example, let's assume I have two servers named bill and shawn. The server shawn is trying to communicate with bill over port 25 (port 25 is used for sending e-mail over SMTP), but wouldn't you know it, bill isn't responding.

Does shawn or bill Have a Problem?

One of the first things I might do in a scenario like this is find another machine on the same network and try to connect with bill from there. If I can talk to bill from another machine on the same network, the problem is most likely with shawn or with the network in between shawn and bill. If I have the same problem from another machine on the same network, it's more likely that the problem is with

bill, so I would start troubleshooting from there. Just so I can discuss more troubleshooting steps, let's start troubleshooting from shawn.

One of the most embarrassing things in troubleshooting is to waste an hour only to find out that something wasn't plugged in. So the first step I perform is to make sure that shawn is plugged in to the network. Although I could inspect the port physically on the server, if the server were in a different city, I might run a program like ethtool. ethtool gives you a lot of different diagnostics on your Ethernet devices. By default, all you have to do is run ethtool as root and pass the Ethernet device you want to check as an argument. In many cases this will be eth0:

```
$ sudo ethtool eth0
Settings for eth0:
Supported ports: [ TP ]
Supported link modes: 10baseT/Half 10baseT/Full
                           100baseT/Half 100baseT/Full
                           1000baseT/Half 1000baseT/Full
Supports auto-negotiation: Yes
Advertised link modes: 10baseT/Half 10baseT/Full
                           100baseT/Half 100baseT/Full
                           1000baseT/Half 1000baseT/Full
Advertised auto-negotiation: Yes
Speed: 100Mb/s
Duplex: Full
Port: Twisted Pair
PHYAD: 0
Transceiver: internal
Auto-negotiation: on
Supports Wake-on: pg
Wake-on: d
Current message level: 0x000000ff (255)
Link detected: yes
```

As you can see, ethtool gives all sorts of information, including the fact that this machine supports 10 base T, 100 base T and gigabit networking speeds, but it currently communicates at 100 base T, full duplex. To check for a link, just look at the very last line that says "Link detected". As you can

see in my example, link is detected, so my cable is plugged in and I can move on.

Before I move past ethtool completely, it's worth mentioning that it does a lot more than just diagnose link problems. A common problem I've found on networks is a host with slower-than-normal network speeds. Often you'll see this crop up after a reboot or a power outage. What often happens is that when the interface connects to the network, it will try to auto-negotiate the fastest speed it can. Sometimes auto-negotiation doesn't work correctly, in which case the interface might fail back to half duplex mode or might even fail back to 10 base T! If you know that your network can support 100 base T at full duplex, you can use ethtool to disable auto-negotiation and force full duplex. To do this for eth0, you would type:

```
$ sudo ethtool -s eth0 autoneg off duplex full
```

Test Local IP Settings

After we have confirmed that shawn is plugged in, the next step is to confirm that eth0 on shawn is

configured correctly. To do that, I would use the ifconfig command with eth0 as an argument. I should get back all of the network information I need to determine whether eth0 is set up correctly on shawn:

```
$ ifconfig eth0
eth0      Link encap:Ethernet HWaddr 00:17:42:c0:ff:ee
          inet addr:10.1.1.9 Bcast:10.1.1.255 Mask:255.255.255.0
             inet6 addr: fe80::217:42ff:fe1f:18be/64 Scope:Link
               UP BROADCAST MULTICAST MTU:1500 Metric:1
             RX packets:1 errors:0 dropped:0 overruns:0 frame:0
             TX packets:11 errors:0 dropped:0 overruns:0 carrier:0
                collisions:0 txqueuelen:1000
              RX bytes:229 (229.0 B) TX bytes:2178 (2.1 KB)
```

There is a lot of output in that command, but the first line I would look at is the second line of output. There I can see that eth0's IP address is 10.1.1.9 and that its subnet mask is 255.255.255.0. If the machine were supposed to have a different IP or subnet mask from what I see here, that potentially could be the cause of the problem. If eth0 didn't

The image features a stack of colorful books in the background, with the Texas Linux Fest logo overlaid. The logo consists of the words "TEXAS LINUX FEST" in large, stylized yellow letters, with a small crown icon above the "T". Below the main text, there are two smaller "31" icons. In the bottom right corner, the date "April 10, Austin TX" is written in yellow. The overall background is dark, making the yellow text stand out.

Mark your calendar now....

The premiere regional Linux and open source conference for the Lone Star State!

Register online today at www.texaslinuxfest.org

April 10, Austin TX

Photo from <http://www.flickr.com/photos/luxpixel/2905283558/>

have an IP or subnet mask configured at all, I might run `ifup eth0` to bring up the interface, or I might look into the local network settings (`/etc/network/interfaces` on a Debian or Ubuntu machine, `/etc/sysconfig/network-scripts/ifcfg-eth0` on a Red Hat-based machine) to see if anything is set incorrectly. If I can't seem to get the interface to come up, and this host gets its IP from DHCP, I might have to move my troubleshooting focus to the DHCP server.

Test the Local Subnet

After you have confirmed that the interface is on the network and should be able to communicate, the next step is to test whether you can access another host on the same subnet—specifically the gateway if you have one configured. Why? Well, if you can't talk to a host on the same subnet, especially if you can't talk to the gateway, there's no point in testing communications with hosts outside of your local subnet. First, I will use the `route` command to see what gateway is configured, and then I will use `ping` to see whether I can access the gateway:

```
$ sudo route -n
Kernel IP routing table
Destination     Gateway      Genmask        Flags Metric Ref  Use Iface
10.1.1.0        *           255.255.255.0   U     0      0      0 eth0
default         10.1.1.1    0.0.0.0       UG    100    0      0 eth0
```

In this example, I have a very basic routing table, and the line that begins with the word `default` defines my default gateway: 10.1.1.1. Be sure to use the `-n` option with `route` in this step. Without the `-n` option, `route` will try to resolve any IP addresses it lists into hostnames. Besides the fact that `route` will execute faster with `-n`, if you have network problems, you might not even be able to talk to your DNS server, plus DNS troubleshooting is a topic for another column.

Because I see that the gateway is 10.1.1.1, I would use the `ping` command to confirm that I can communicate with that gateway:

```
$ ping -c 5 10.1.1.1
PING 10.1.1.1 (10.1.1.1) 56(84) bytes of data.
64 bytes from 10.1.1.1: icmp_seq=1 ttl=64 time=3.13 ms
64 bytes from 10.1.1.1: icmp_seq=2 ttl=64 time=1.43 ms
64 bytes from 10.1.1.1: icmp_seq=3 ttl=64 time=1.79 ms
64 bytes from 10.1.1.1: icmp_seq=5 ttl=64 time=1.50 ms

--- 10.1.1.1 ping statistics ---
5 packets transmitted, 4 received, 20% packet loss, time 4020ms
rtt min/avg/max/mdev = 1.436/1.966/3.132/0.686 ms
```

This output tells me that my machine can at

least talk with the gateway and presumably with the rest of the 10.1.1.x network. Now, if I couldn't talk to the gateway, that could mean my network administrator is being annoying and blocking ICMP packets. If that's the case, I would just choose another machine on the same subnet (10.1.1.2-10.1.1.254) and try to ping it instead. If I am the network administrator (and therefore not blocking ICMP), or if ICMP isn't being blocked for some other reason, the problem at this phase could be some sort of VLAN issue that I would have to resolve on the network switch itself.

If you run the `route` command and don't find a default gateway set, you might be tempted to conclude that's the source of the problem. Be careful! That conclusion might be premature. See, if shawn and bill are on the same subnet, I don't need a default gateway configured for those servers to communicate. I'm not going to get into how to calculate subnets in this column, but suffice it to say in my example, if shawn has an IP of 10.1.1.9 and a subnet mask of 255.255.255.0, bill could have an IP of 10.1.1.1 through 10.1.1.254 and be on the same subnet. In that case, I might just ping bill directly. Ideally, I would have a third host on the same subnet I also could ping. That way if bill doesn't respond, but another host on the same subnet responds, I can narrow in on bill as the likely source of the problem.

Next: Probe bill's Ports

If bill is responding to ping, the next step is to test whether port 25 is even open on bill. There are a few different methods for doing this, but `telnet` is one of the easiest and is likely already to be installed on your machines. Let's assume bill has an IP of 10.1.1.17; I would type:

```
$ telnet 10.1.1.17 25
Trying 10.1.1.17...
telnet: Unable to connect to remote host: Connection refused
```

If `telnet` doesn't complain about `Connection refused`, but instead starts outputting SMTP commands, then congratulations, you don't have a networking problem! On the downside, this means you probably have some sort of SMTP problem, which might be more of a pain to troubleshoot. If `telnet` complains with `Connection refused`, either port 25 is down on the remote machine (possibly the SMTP service on bill isn't running or isn't listening on that port), or a firewall is blocking you. This is where a tool like `nmap` can be handy, and it's one of the reasons I often use `nmap` instead of `telnet` when I want to test whether a port is available.

You see, many firewalls are configured to block ports by dropping packets with no reply. Because



Want your business to be more productive?

The ASA Servers powered by the Intel Xeon Processor provide the quality and dependability to keep up with your growing business.

Hardware Systems for the Open Source Community - Since 1989.

(Linux, FreeBSD, NetBSD, OpenBSD, Solaris, MS, etc.)

1U Server - ASA1401i



- 1TB Storage Installed. Max – 3TB.
- Intel Dual core 5030 CPU (Qty-1), Max-2 CPUs
- 1GB 667MGZ FBDDIMMs Installed.
- Supports 16GB FBDDIMM.
- 4X250GB htswap SATA-II Drives Installed.
- 4 port SATA-II RAID controller.
- 2X10/100/1000 LAN onboard.

2U Server - ASA2121i



- 4TB Storage Installed. Max – 12TB.
- Intel Dual core 5050 CPU.
- 1GB 667MGZ FBDDIMMs Installed.
- Supports 16GB FBDDIMM.
- 16 port SATA-II RAID controller.
- 16X250GB htswap SATA-II Drives Installed.
- 2X10/100/1000 LAN onboard.
- 800w Red PS.

3U Server - ASA3161i



- 4TB Storage Installed. Max – 12TB.
- Intel Dual core 5050 CPU.
- 1GB 667MGZ FBDDIMMs Installed.
- Supports 16GB FBDDIMM.
- 16 port SATA-II RAID controller.
- 16X250GB htswap SATA-II Drives Installed.
- 2X10/100/1000 LAN onboard.
- 800w Red PS.

5U Server - ASA5241i



- 6TB Storage Installed. Max – 18TB.
- Intel Dual core 5050 CPU.
- 4GB 667MGZ FBDDIMMs Installed.
- Supports 16GB FBDDIMM.
- 24X250GB htswap SATA-II Drives Installed.
- 24 port SATA-II RAID, CARD/BBU.
- 2X10/100/1000 LAN onboard.
- 930w Red PS.

8U Server - ASA8421i



- 10TB Storage Installed. Max – 30TB.
- Intel Dual core 5050 CPU.
- Quantity 42 Installed.
- 1GB 667MGZ FBDDIMMs.
- Supports 32GB FBDDIMM.
- 40X250GB htswap SATA-II Drives Installed.
- 2X12 Port SATA-II Multilane RAID controller.
- 1X16 Port SATA-II Multilane RAID controller.
- 2X10/100/1000 LAN onboard.
- 1300 W Red PS.

All systems installed and tested with user's choice of Linux distribution (free). ASA Colocation—\$75 per month



2354 Calle Del Mundo,
Santa Clara, CA 95054
www.asacomputers.com
Email: sales@asacomputers.com
P: 1-800-REAL-PCS | FAX: 408-654-2910

Intel®, Intel® Xeon™, Intel Inside®, Intel® Itanium® and the Intel Inside® logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Prices and availability subject to change without notice.
Not responsible for typographic errors.



Powerful.
Efficient.

normally a server would send a basic reply back to let you know the port is closed, if the packet is dropped instead, nmap will flag it as filtered instead of closed:

```
$ nmap -p 25 10.1.1.17
```

```
Starting Nmap 5.00 ( http://nmap.org ) at 2010-01-04 20:20 PST
Interesting ports on 10.1.1.17:
PORT      STATE    SERVICE
25/tcp     filtered  smtp
```

In this case, nmap says the port is filtered, which tells me there is a firewall blocking this port. If these machines were on different subnets, there might be a firewall in between the networks restricting access. Because I know these machines are on the same subnet, I would assume that there is some iptables firewall configured on bill that needs to be checked.

Test bill Directly

Let's assume we think the problem is on bill. After I've performed the same network troubleshooting on bill that I have on shawn, the next step is to log in to bill and test whether port 25 is open and listening for connections. For this, I will use the netstat tool. netstat can be used to output all sorts of information about network connections on the machine. In this case though, I will just use the -lnp options to list listening ports and the processes that have the ports open, then I will grep for the port I'm interested in, port 25:

```
$ sudo netstat -lnp | grep :25
tcp        0      0 0.0.0.0:25      0.0.0.0:*          LISTEN      1878/master
```

The column I want to pay the most attention to here is the fourth column that lists what local address is open on port 25. In this case, I can see it is set to 0.0.0.0:25, which means bill is listening to port 25 connections on all available interfaces. If I had set up the mail server to listen only on eth0, this would be set to 10.1.1.17:25. If, on the other hand, I saw this was set to 127.0.0.1:25, I might have found the cause of the problem: the mail server was set to listen only to the localhost address (127.0.0.1) and isn't listening for any connections from the outside network. In that situation, I would reconfigure my mail server so that it listens on eth0. If I got no output from the above command, I would know my problem is that my server isn't running at all (or isn't set to listen on port 25). Then, I'd need to start my mail server and troubleshoot why it stopped running to begin with, or why it isn't listening on the right port.

As you can see, network troubleshooting can lead you in all sorts of interesting directions. Even now I've barely scratched the surface. In my next column, I'll extend network troubleshooting beyond the local network and touch on how to track down routing and DNS problems from your local networks to the Internet itself. ■

Kyle Rankin is a Systems Architect 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.



DIRK ELMENDORF

Interview with Rich Hickey, Creator of Clojure

An in-depth look at the new language from the man himself.

This month's column diverges from the normal pattern of covering my struggles with technology. When I started this column, I was really looking forward to the Software Development issue, because I was hoping to spotlight a language I am learning—Clojure (pronounced like "closure"). But, I found myself still in the process of bending my brain around it as my deadline loomed, so instead of trying to teach you while I am still learning, I decided it would be more interesting to hear from the man who created the language instead, Rich Hickey.



Rich Hickey, Creator of Clojure

DE: What did you do before you started the Clojure project?

RH: I'm a consultant, so I work on various things. I think the big thing I've done recently is I worked on the national exit poll.

DE: What other languages did you use before inventing your own?

RH: I was a C++ developer for a long time. I taught it at NYU for a little while. I worked in scheduling systems and broadcast automation all in C++. Then I moved from that to Java and C#. At the same time, I also started doing some Common Lisp.

DE: Most people who create their own language start from scratch—what was it that drew you to do a Lisp?

RH: I discovered Lisp after ten years of C++ and said to myself, "What have I been doing with my life?" I realized it was just much more productive for me. I fell in love with it right away. Basically, I said to myself, "At some point in my life, this is really what I want to be doing." I spent several years subsequent to that continuing to do my commercial work in Java and C#, but all the while hoping at some point to be able to get Lisp into the mix. That's what drove me to doing Clojure, because I saw that either .NET or the JVM were requirements for all the customers I was encountering. Lisp really couldn't reach those platforms, so I decided to make a Lisp that could.

DE: Why put it on the JVM?

RH: It's designed to be useful for the work I was doing, where you have customers that have requirements that things run on one of these standard platforms—platform not meaning the OS but these infrastructures, like the JVM or .NET.

DE: And putting it on the JVM, did that get you out of doing a lot of the low-level stuff?

RH: Absolutely! I mean as a separate concern from the practicality of being able to access all the libraries, being on the JVM gives you a really nice separation of concerns between the high-level language, which is what I got to focus on, and the runtime, which is something that the JVM does an excellent job with. It's got a great garbage collector and a very sophisticated runtime compilation infrastructure. So as a language developer, you don't have to worry about emitting machine code or anything that low level. But, I'm not on the JVM just to get a free ride in terms of making it easier to implement. It

was very much a part of the design of the language that you be able to touch and reach the underlying host platform, because there's a lot of value there.

DE: Libraries are part of that value, right?

RH: The amount of libraries available are really fantastic. That meant as a new language, Clojure had a ton of libraries right out of the gate. People didn't have to wait for a Clojure library for sockets or for talking to the database or for doing a GUI. They had libraries for those things right away. So they were productive with Clojure right away.

DE: You seem very pragmatic. You talked about productivity and libraries and how you get to use this on a daily basis. That seems like a different view of what a language is and how it fits.

RH: It's a little bit different. There's no taking away from Common Lisp being very powerful. I just think that it was standardized back before the Internet was commonplace. Then it was hard to move because there was a change in the marketplace in the 1990s that shut down artificial intelligence and a bunch of other things that were where Lisp lived. It wasn't really about the language, but it did cause some stagnation. So coming out fresh, not being compatible, really gave me a blank slate. Being a commercial developer, I knew what I needed to have be present in order for it to be practical.

DE: Clojure is not object-oriented, why not?

RH: Well it's not object-oriented the way Java, C# or C++ is. That's not really the way you would structure things. It is in some ways a rejection of my heritage, as an object-oriented programmer. I think after having done it for two decades, I don't believe in it anymore. I just don't think it's the right way to start. It can help you organize your code, but it brings along with it some complexity that I have found in real systems always ends up biting you—and it is related to mutability. By default, an object-oriented program is a graph of mutable objects. That's a very, very difficult thing to think about, debug and keep running. I've worked on very big object-oriented systems, and you always essentially run into the problems related to that architecture. I think that even before you get to concurrency, there are complexity problems with mutable objects that basically affect every large object-oriented application. When you add in concurrency, those problems become much clearer.

So a functional approach was something that I had already started doing, even in programs I was writing in C#. For instance, there were parts of the national exit poll system that were very functional, even though it's a C# system, because the way to defend yourself against this complexity is to write in a more functional style with a lot more immutability. The problem is that

it's not very idiomatic in C# or Java to do so. I wanted to make a language where it was—where the default was to do the right thing. When you needed mutability, there would be a good story about how to do that compatibly with concurrency.

DE: So is that why you have the transaction system built in?

RH: Once you say by default everything should be immutable, real systems can't work that way everywhere. There have to be places where people can see the effect as if something were changing. So how do you mimic that? The Clojure system says, "Well, we're gonna take something immutable and put it into a cell or a reference, and we'll swap out what's in that reference from one immutable thing to another." And, that will look like change to the program. It's a little bit different from a variable in that you're promising yourself the only thing you'll ever put in there is an immutable thing. But, that gives you a great separation of concerns. Because then you have these immutable values that most of your program is manipulating. You can say about those references, "This is what happens in the concurrent program." And, one of the things could be transactions, and that's the STM (Software Transactional Memory).

So it's a nice story. You can switch back and forth between the different constructs depending on how much sharing there will be in your application. But, yes, that's why it's built in. Because without it, you just leave people wondering, "Okay, well, when I need to change, what do I do? Do I go back to locks?", and that kind of thing.

DE: Clojure introduces a lot of different ideas all at once. How do you get started?

RH: Well, you start small. I think the way in for most people is to see the way Clojure uses associative data structures (maps in Clojure). They're called dictionaries or hashes in other languages. So people that have used Python and Ruby have gotten a good feel for working with objects as if they were just generic dictionaries.

You will be facing challenges because some of the things that you're used to aren't there. On the other hand, you also are handed a lot of things that are much easier and clearer and fit together better. So you get a little bit of a bump to try to change your habits, but you're getting immediate payback in a program that's easier to understand, easier to test and much closer to the way you're thinking about the problem.

DE: Is that the power you see—that it makes it easier for you to code the problem you're trying to tackle clearly?

RH: Absolutely. I mean, that's really the great thing about a Lisp or one of the highest-level functional

languages like a Haskell, the program pretty much consists of stuff that matters. Anything that doesn't matter, you can make go away. Whereas, if you look at a Java, C# or C++ program, it's full of things that don't matter—things that are there because of the way you have to do things or the syntax and the language.

DE: Well, I noticed that in May 2009 you guys announced Clojure as 1.0. It seems like open-source projects often linger below 1.0. How did you decide this is 1.0?

RH: I think one of the nice things about being a Lisp is that most of the additions you make to the language aren't really in the language, they're just libraries. It is very easy to add and not break anything. Clojure has grown really rapidly and yet in a non-disruptive way, because new things are just new things. If you don't use them, they don't impact you.

DE: How does your process for evolving the language work?

RH: It's not very formal. I'm trying to stay in touch with what people are doing and what's working and what needs refinement. Sometimes there are performance things I want to tackle. Other times there are new things in the libraries people need that I'll tackle. It's not like there's a big road map, because I don't really believe in getting too far ahead of yourself. Things are done incrementally.

DE: How does code get into Clojure? Is it just you, or is there a team?

RH: On the core stuff it's me. There's one other person who can apply patches that I've approved. Then contrib is much wider. In contrib, I think there are about a dozen people who can commit and there are 100 registered contributors.

DE: And how do you register to become a contributor?

RH: You sign a contributor's agreement. It's a lot like the one that Sun used, and then you're a contributor.

DE: If I want to make a difference in the language where do I get started?

RH: Well you should become familiar with what Clojure has got already. Usually, you'll get good at the core of Clojure, and then you'll start building apps and using parts of contrib. Then perhaps you could find something that contrib isn't covering and contribute a whole new library there. Or, you'll find something in contrib that could be enhanced. You can talk to the person who owns that part and say, "Hey, I've got ideas", and work together. It's not difficult. There's not a lot of hurdles. The core language being much more focused I think is critical, because I

don't think languages really are built by teams of people usually. You have to have a vision, and you really don't want to be going in more than one direction at a time. In particular for Clojure, it's essential to me that the core stays very small. Being simple at the very core of things is part of what makes it good.

DE: Are you surprised by the growth?

RH: It's out of control. When I released it, I had the realistic expectation that if 10–100 people used it, that would be amazing. Because that's all you can expect. But for some reason, it took off. It was not something I anticipated. While we're talking, we're at 2,999 members of the Google group. Who could know that was going to happen? It's been crazy. Very crazy.

DE: Is it exciting for you to kind of get some validation that you weren't the only person who wanted a Lisp you could use?

RH: Yeah, I think there's the Lisp aspect, and then there are plenty of people using Clojure that Lisp is a kind of hurdle for initially. They're not coming looking for the Lisp part. They want the dynamic development. They want the immutability. They want a good concurrency story. They want functional programming. The Lisp part is not the appeal.

DE: I noticed in one of your posts talking about Clojure in Clojure. What does that mean?

RH: Well, Clojure was written from scratch. So I started writing it in Java. A lot of the bootstrap, the underpinnings of Clojure, were written in Java. The basic data structures were all written in Java, and the first compiler was written in Java. Then, once you had a compiler and the data structures, you could write the rest of the language in Clojure. So what I want to do is go back to those parts that are written in Java and rewrite them in Clojure now that Clojure exists. Recently, I've been doing a lot of work so it has the features and performance needed to do even the lowest-level parts of implementing itself. We can go back and re-implement Clojure in Clojure.

DE: So is the goal to make it so that you have more kind of idiomatic flexibility? Or just for completeness?

RH: Well, the goal is to get rid of the Java code. It demonstrates that Clojure has sufficient performance and expressiveness to do everything the Java part did. The other part is it will make moving forward a lot easier. David Miller has been porting all the Java to C# to port it to .NET. So when there's a lot of Java, there's a lot of C#. When most of Clojure is written in Clojure, only a very little bit will be Java-specific. And, only that little bit would need to be ported to .NET. In addition, I think a really important target for Clojure will be JavaScript. So moving Clojure mostly to

Clojure means reducing the footprint of a port to a very small amount, and I think that's a valuable goal.

DE: When you move to these other kinds of things, do you worry about the low-level stuff you have counted on the JVM for?

RH: The whole thing is not to try to make all these platforms the same. It's to say, "Well, I know how to accomplish things quickly in Clojure." And, when my target platform is the browser, I can leverage my knowledge about Clojure to do that as well. You'll still want the same kind of ability to access the hosts that you have for Java when you're in JavaScript. You end up with a body of expertise. You will have libraries that will work in all those places that are not host-specific. So, more and easier portability is an objective of Clojure in Clojure. And, it'll just be more fun for people to hack on it.

DE: What is your typical Clojure user like?

RH: There isn't a typical user. It's very much split among people with very different backgrounds. You have people who are solid Lispers who are looking forward to being able to reach the rest of the world easily, who actually know very little Java or none. Then you have Java people who are looking to get that expressiveness and agility, but they want to know it will be solid enough to do everything they were doing in Java in terms of performance and threading. We get people from Ruby and Python who know they love dynamic languages already, but have found performance or other issues there. Or they're just trying to up their game and learn more about functional programming. And then we'll get people from the functional camp (Haskell or ML) who are looking for the practicality of being on a platform like the JVM. What's great is they can all help each other.

DE: They all have a different piece. Your platform provides something that they can each bring and they each need.

RH: That's a great way to put it. It's made for a fantastic community where somebody who's struggling to learn the functional programming side can turn right around and be the expert about CLASSPATH for somebody else because he knows Java. So that's been great. I just can't say enough about the community—it's fantastic! I think there's just a tremendous amount going on. A lot of libraries. The numbers and the growth are a big part of the appeal of Clojure. There are books. One book is out, and more books are on the way. So I think it's very approachable now in terms of getting a lot of help.

DE: What do you see that's exciting for you about the future?

RH: I think there are lots of challenges still to

come, for people who are looking to try to take advantage of all these CPUs they are going to have. I'm excited about Clojure in Clojure and growing that. In general, I'm trying to work on the general problem of dealing with time in programs. I think that's part of what Clojure is about. I still have a lot of ideas around that that I haven't implemented yet.

DE: When you say time in programs, what do you mean?

RH: Well, when we talk about things changing, and concurrency and mutability and things like that, those are all about time. I don't think the languages that we've had up till now have been explicit enough about time. So we've run into a lot of problems, because we're not thinking about time explicitly. Clojure makes time explicit. I want to do more work on that, because I think that will be a really important thing as we move to these multicore platforms.

DE: So are you trying to create something that is missing from other languages?

RH: Well, I mean I'm borrowing as much as I can from wherever I can for sure. I definitely stand on the shoulders of giants. But in this area, I never claim any novelty for Clojure. Clojure is mostly about trying to take existing good ideas that may not have been put together and put them together. It's certainly missing from the more traditional languages like Java, C# and C++. But, there are other ways to address it that are present in Erlang and Haskell, for instance.

DE: Any final thoughts?

RH: I would encourage everybody to try it out. It's a welcoming community, and beginners are always welcome and treated nicely. And, we're happy to have more users.■

Dirk Elmendorf is cofounder of Rackspace, some-time home-brewer, longtime Linux advocate and even longer-time programmer.

Resources

If you are interested in learning more, head to the main Web site (clojure.org).

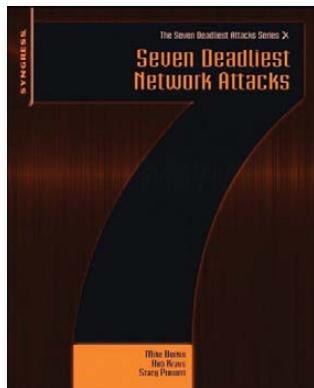
You also can check out a number of presentations at clojure.blip.tv. I really enjoyed the "Clojure for Lisp Programmers", even though I am very new to Lisp.

If you are interested in learning more about Hickey's view of time in programs, check out his "Persistent Data Structures and Managed References" at InfoQ: www.infoq.com/presentations/Value-Identity-State-Rich-Hickey.

AdaCore's CodePeer

Make your Ada code live up to the language's elegant name with AdaCore's new CodePeer, a source code analysis tool that detects runtime and logic errors in Ada programs. As a code reviewer, CodePeer identifies constructs that are likely to lead to runtime errors, such as buffer overflows, and flags legal but suspect code typical of logic errors. Additionally, AdaCore says that its tool goes "beyond the capabilities of typical static analysis tools", producing a detailed analysis of each subprogram, including pre- and postconditions, which allows for early detection of potential bugs and vulnerabilities. CodePeer can be used both during system development or as part of a systematic code review process. Finally, it can be used either as a standalone tool or fully integrated into the GNAT Pro Ada development environment.

www.adacore.com



Seven Deadliest Attacks Series (Syngress)

First there were seven brides for seven brothers, and now there are seven tech books for seven strains of security geek—all courtesy of Syngress. The publisher is promising a whopping seven books in the *Seven Deadliest Attacks Series*, each with its own focus on a specific type of security breach. The titles include: *Seven Deadliest Microsoft Attacks* by Rob Kraus, Brian Barber, Mike Borkin and Naomi Alpern; *Seven Deadliest Network Attacks* by Stacy Prowell, Rob Kraus and Mike Borkin; *Seven Deadliest USB Attacks* by Brian Anderson and Barbara Anderson; *Seven Deadliest Wireless Technologies Attacks* by Brad Haines; *Seven Deadliest Social Network Attacks* by Carl Timm and Richard Perez; *Seven Deadliest Web Application Attacks* by Mike Shema; and (almost there) *Seven Deadliest Unified Communications Attacks* by Dan York. Each book covers the anatomy of the seven respective attacks, as well as how to get rid of and defend against them.

www.syngress.com

SoleraTec's Phoenix RSM

Keep the burglars at bay with SoleraTec's new and improved Phoenix RSM, an overarching forensic video surveillance management system. The RSM part stands for Phoenix's Record, Store and Manage capabilities. Other product capabilities include searches in video surveillance environments; incrementally scalable three-medium (hard disk, digital computer tape and optical) on-line and off-line multitier storage; and unlimited retention and support for an unlimited number of cameras, servers and users. Furthermore, video from all connected cameras can be reviewed, investigated and exported with client tools that run on Windows, Linux and Mac OS. Features added to the new version include one-step centralized camera configuration, simplified camera policy management and support for QuickTime and VLC media players.

www.SoleraTec.com



Libelium's Wasp mote

Wasp mote from Libelium is a modular platform for wireless sensor networks that enables environmental monitoring in adverse conditions and remote locations with its radio range of up to 40km. The sensors are intended for deployment in fire and flood detection and other environmental monitoring applications. Wasp mote networks can communicate to the external world via GPRS or in situations with very difficult wireless connectivity, such as mines. Each sensor device can store more than 21 million different sensor measurements in its internal memory. Wasp mote's four power modes—on, sleep, deep sleep and hibernation—enable a device to function for up to three years without recharging the battery, while a small solar panel can allow it to run indefinitely. Special boards that enable detection of gases and physical events (such as pressure, impact, vibration, temperature and so on) can be integrated. Open-source API and programming environment are available.

www.libelium.com/wasp mote



Bart Kummel's *Apache MyFaces 1.2 Web Application Development* (Packt)

The folks at Packt Publishing recently released Bart Kummel's new book *Apache MyFaces 1.2 Web Application Development*, a work that teaches readers how to build appealing Web interfaces with the open-source Apache MyFaces framework. Written as a step-by-step, example-driven tutorial, Kummel's book teaches concepts such as assuring re-usability of code, building consistent-looking and usable pages with Trinidad components, applying advanced components from the Tomahawk library, enabling AJAX functionality without writing JavaScript code, creating dynamic applications that utilize Trinidad's skinning capabilities and preventing the duplication of validation rules. The book also contains scores of tips and tricks based on experience with MyFaces in real-life projects.

www.packtpub.com

Active Media Products' 600X Pro Series CompactFlash Cards

The geeks at Active Media Products weren't satisfied with the performance of CompactFlash cards in digital photography applications, so they made their own. The company's 600X Pro line of CF cards, which write up to 90MB per second, aims to free the memory card's hitherto role as bottleneck in shooting action sequences with DSLRs firing up to 10 frames per second. Active Media also says that the cards support 0–70°C operating temperatures and are rugged and reliable enough to take into the field. Capacities range from 8GB to 64GB.

www.activemp.com



Cyberoam iView Appliances

Cyberoam iView, an open-source logging and reporting solution, has recently become available in a convenient appliance form. The product caters to the logging/reporting requirements of SMBs and distributed enterprises, delivering a comprehensive view of network activity across dispersed geographical locations. Cyberoam describes the iView appliances as quick-to-deploy and easy-to-manage preloaded hardware devices with terabyte-storage space, RAID technology, redundancy and high levels of storage reliability. The appliance further enables organizations to gain complete visibility into network activity with real-time security and access reports related to top virus attacks, spam recipients, Web users and more, reinforcing organization-wide network security and data confidentiality. It also offers archiving to meet forensic requirements.

www.cyberoam.com

Perforce's Software Configuration Management System

Perforce came out swinging in the new year, announcing a new version 2009.2 of its Software Configuration Management (SCM) System. SCM is a tool that versions and manages source code and digital assets for enterprises of all sizes. The most significant addition to 2009.2 is shelving—that is, real-time metadata replication and additional functionality for working off-line. This feature enables developers to cache modified files in the Perforce Server without first having to check them in as a versioned change. Users, thus, can pass pending changes to managers as part of code review or approval workflows, share works in progress with another team member or workstation, test changes in a distributed build environment, and put aside an effort when a higher priority task arrives.

www.perforce.com

PERFORCE
SOFTWARE

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

Fresh from the Labs

Tor—Anonymity On-line

<https://www.torproject.org>

We've covered Tor in *LJ* before (see Kyle Rankin's "Browse the Web without a Trace", January 2008), but that was some time ago, and this subject seems to be more timely with each passing day. Also, with Tor being at only 0.2.x status, it still qualifies as software in development, so I'm justified in featuring it this month.

For those not in the know, Tor stands for The Onion Router, and its roots go all the way back to the US Naval Research Laboratory, Tor's original sponsors. It then became an EFF (Electronic Frontier Foundation) project until 2005, and it now has moved up to being its own nonprofit research/education organization: the Tor Project.

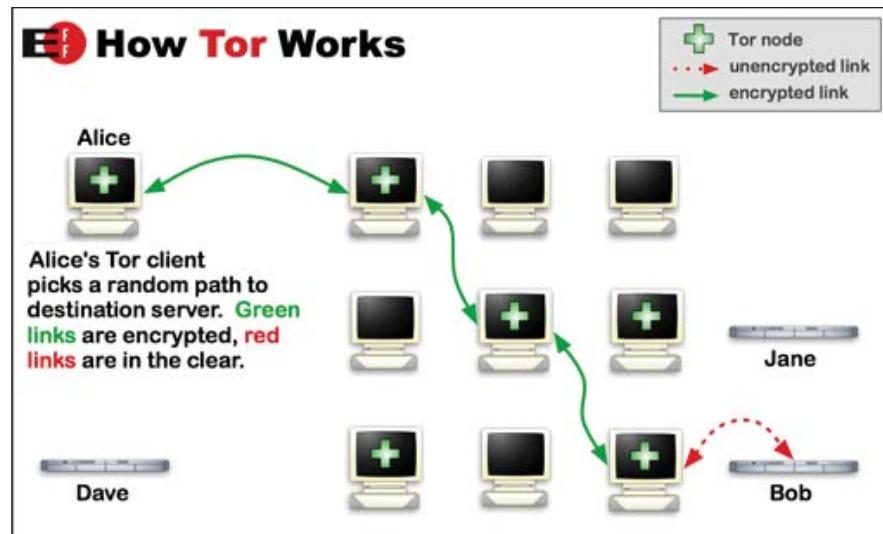
The essential idea is that your original IP address is masked by passing it through numerous special routers, designed to avoid keeping records, until the original source has been lost and the receiving end knows only about the last Tor box it encounters. To quote Tor's man page:

Users choose a source-routed path through a set of nodes and negotiate a "virtual circuit" through the network, in which each node knows its predecessor and successor, but no others. Traffic flowing down the circuit is unwrapped by a symmetric key at each node, which reveals the downstream node.

Basically, Tor provides a distributed network of servers ("onion routers"). Users bounce their TCP streams—Web traffic, FTP, SSH and so on—around the routers, and recipients, observers and even the routers themselves have difficulty tracking the source of the stream.

However, all that may be a bit headache-inducing, and the Tor Web site explains things in human terms quite nicely:

Tor is free software and an open network that helps you defend against a form of network



Tor takes a clever approach to anonymity, deliberately losing IP addresses as it bounces from server to server.

surveillance that threatens personal freedom and privacy, confidential business activities and relationships, and state security known as traffic analysis.

Tor protects you by bouncing your communications around a distributed network of relays run by volunteers all around the world: it prevents somebody watching your Internet connection from learning what sites you visit, and it prevents the sites you visit from learning your physical location. Tor works with many of your existing applications, including Web browsers, instant-messaging clients, remote login and other applications based on the TCP protocol.

Installation and Usage Surprisingly, there aren't many strange library requirements for Tor; it may install straightforwardly on many systems. The only missing library that got in the way was libevent, and installing libevent-dev (which selects the other needed libevent libraries along with it at the time) sorted this out. However, Tor recommends using the program Polipo, but I'll get to that in a moment.

To install Tor, head to the download page where source and binaries are



Tor can be a bit hard to understand at first, but if you look around, many tools can help you along the way, such as TorK and even custom distributions built around using Tor.

available. You can figure out the binaries yourself, but for those using source, grab the latest tarball, extract it, and open a terminal in the new folder. Enter the usual commands:

```
$ ./configure  
$ make
```

If your distro uses sudo:

```
$ sudo make install
```

If your distro doesn't:

```
$ su  
# make install
```

To set up Tor for Web browsing, at this point, you have to install Polipo. This is in most distros' repositories, so you can decide how you want to install Polipo yourself. I'll quote Tor's documentation from here:

Polipo is a caching Web proxy that does http pipelining well, so it's well suited for Tor's latencies. Make sure to get at least Polipo 1.0.4, since earlier versions lack the SOCKS support required to use Polipo with Tor.

Once you've installed Polipo (either from package or from source), you will need to configure Polipo to use Tor. Grab our Polipo configuration for Tor and put it in place of your current polipo config file (for example, /etc/polipo/config or ~/.polipo). You'll need to restart Polipo for the changes to take effect. For example:

/etc/init.d/polipo
restart.

If you prefer, you can instead use Privoxy with this sample Privoxy configuration. But, since the config files both use port 8118, you shouldn't run both Polipo and Privoxy at the same time.

Configure Your Applications to Use Tor

After installing Tor and Polipo, you need to configure your applications to use them. The first step is to set up Web browsing.

You should use Tor with Firefox and Torbutton for the best safety. Simply install the Torbutton plugin, restart Firefox, and you're all set (the Torbutton plugin for Firefox is available at <https://addons.mozilla.org/firefox/2275>).

To Torify other applications that support HTTP proxies, just point them at Polipo (that is, localhost port 8118). To use SOCKS directly (for instant messaging, Jabber, IRC and so on), you can point your application directly at Tor (localhost port 9050), but see the FAQ entry for why this may be dangerous. For applications that support neither SOCKS nor HTTP, take a look at tsocks or socat.

It's really hard to do justice to Tor in this small space, so I hope I've at least pointed you in a useful direction and haven't made any glaring errors. It really is worth heading to the Web site to understand it more fully. Speaking of the Web site, here's an appeal from the Tor folks themselves:

Tor's security improves as its user base grows and as more people volunteer to run relays. (It isn't nearly as hard to set up as you

Presenting
CIO
Thought Leader
Forum

Co-located with

CLOUD COMPUTING
CONFERENCE
AN IT PARADIGM SHIFT

ASTERISK
OPEN
TELEPHONY
CONFERENCE

IT360 SILVER SPONSOR: **MEUK** MEDIA PARTNER: **itWorld Canada** PREMIER MEDIA SPONSORS: **COMPUTERWORLD** **NetworkWorld** **THE GLOBE AND MAIL** **CDN**
IN COOPERATION WITH: **CIO Executive Council** Leaders Shaping the Future of Business PLATINUM ASSOCIATION SPONSOR: **CATAAlliance** CERTIFICATION SPONSOR: **Procept** ASSOCIATES LTD.
CONFERENCE SPONSORS: **INFO-Tech** **research group** **CIO CANADA** **TORONTO** Economic Development, Culture & Tourism

IT professionals and business executives maximize your success: www.it360.ca

METRO TORONTO CONVENTION CENTRE, CANADA
CONFERENCE & EXPO : 7 APRIL 2010

might think and can significantly enhance your own security.) If running a relay isn't for you, we need help with many other aspects of the project, and we need funds to continue making the Tor network faster and easier to use while maintaining good security.

Information is becoming increasingly unsafe, and certain governments and corporations are becoming increasingly invasive regarding personal data. It's time that Net users started taking more care with their information, and Tor is an interesting technology that I'm sure will continue to become more relevant over time.

audiopreview—Multimedia Previewer

audiopreview.codealpha.net

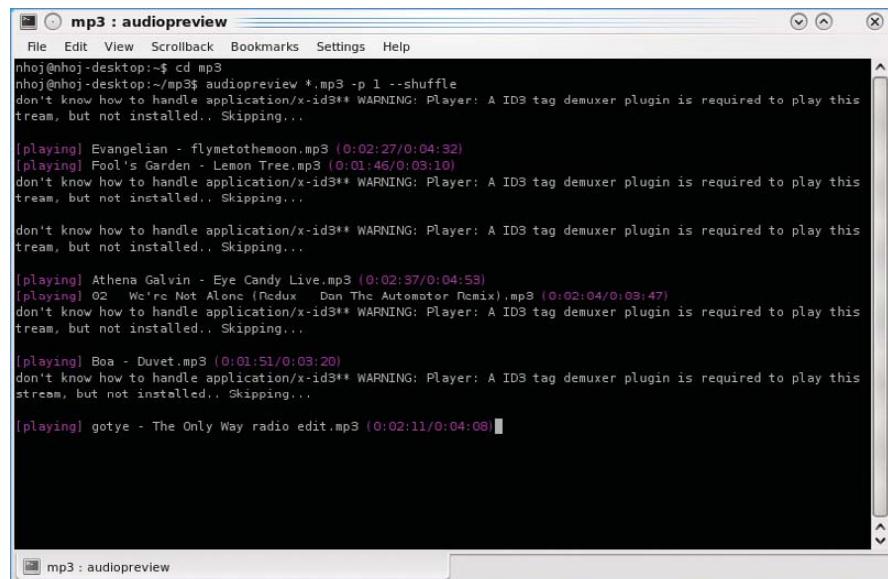
I love niche programs, especially in the area of multimedia. If you're like me, you probably have a folder full of MP3s and Ogggs collected from the last ten years that's reached the point where you've forgotten half the files in there. This month, I stumbled upon the charming little command-line program, audiopreview. To quote the project's Freshmeat entry:

audiopreview is a command-line tool that plays previews of many audio file types (Ogg, MP3, etc.), video file types (AVI, MPEG, Real, etc.), and Internet streams. It also can be used as a regular command-line media file player (that is, play the files entirely like yauap or mpc123 would).

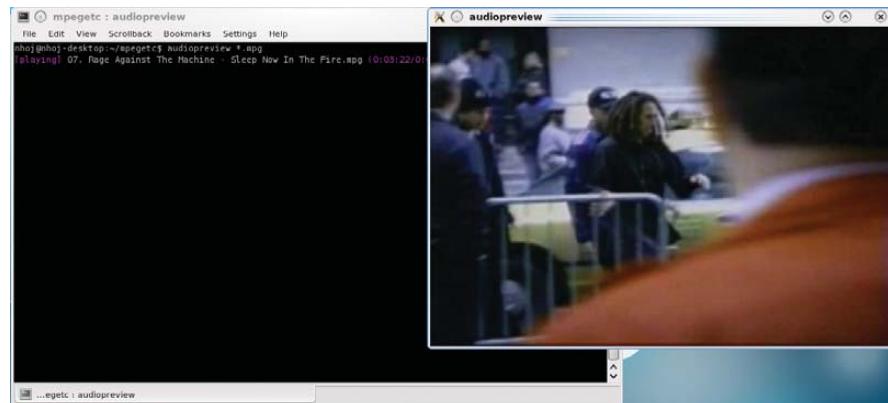
Installation Packages for audiopreview are available in Debian/Ubuntu format or the usual source. If you're running with the source, according to the man page, you need the following libraries: gstreamer0.10-plugins-base, gstreamer0.10-plugins-good, gstreamer0.10-plugins-bad and gstreamer0.10-plugins-ugly.

I found I also had to install the intltool library to get past the configure script. Once you have the library side of things sorted out, compile the program with the usual:

```
$ ./configure  
$ make
```



audiopreview is a simple and easy command-line program for previewing large numbers of music files.



Although the name may suggest otherwise, audiopreview also plays video files.

If your distro uses sudo:

```
$ sudo make install
```

If your distro doesn't:

```
$ su  
# make install
```

Usage Using the actual command can be as simple as entering the folder where the files you want to hear are located and entering:

```
$ audiopreview *
```

(The * is used to indicate all the files in a folder.)

Once the program is running,

you'll be greeted with a simple track listing, along with other relevant information in purple. As far as controls go, the spacebar pauses and unpauses the stream, N plays the next stream, and P plays the previous stream. R restarts the current stream, and Q stops playing and exits the program.

That's the basic usage out of the way, but let's refine it with some command-line switches to hone your usage. For new Linux users, these are added at the end of the command, like this:

```
$ audiopreview files-to-play --switch
```

If you plan on using audiopreview to play a whole song instead of segments, use the switch --entirely or -e.

If you want audiopreview to start over again after the last song has been played, use `--loop` or `-1`.

As mentioned previously, audiopreview also can play some video formats. However, this being a command-line program, there's a good chance you may not have X running. If so, you'll want to disable the video to avoid errors. To do so, enter `--no-video`.

The default starting position for each file seems to be random, which might become annoying for those looking for more specific sections of a song. Thankfully, you can specify which section of a song you want to hear with a simple numerical switch. Add: `--position=POSITION` or `-p POSITION`, and replace POSITION with the numbers 0, 1, 2 or 3. 0 sets the position to the beginning, 1 to the middle, 2 to the end, and 3 makes the start position random.

Last but not least, engage the all-important shuffle function with `--shuffle` or `-S`. For example:

```
$ audiopreview *.mp3 -p 1 --shuffle
```

The above command plays all the MP3 files in a directory, sets the starting position to the middle of a song and shuffles the order in which they're played.

You can work out the rest from here, but honestly, do yourself a favor and check out the man page with:

```
$ man audiopreview
```

Ultimately, audiopreview fills a nice little niche that will appeal to anyone sorting through large collections of music (and some video) files. DJs in particular will find this of real use, but I found it great for rediscovering songs I hadn't listened to in years. Love it. ■

John Knight is a 25-year-old, drumming- and climbing-obsessed maniac from the world's most isolated city—Perth, Western Australia. He can usually be found either buried in an Audacity screen or thrashing a kick-drum beyond recognition.

Brewing something fresh, innovative or mind-bending? Send e-mail to newprojects@linuxjournal.com.

Projects at a Glance

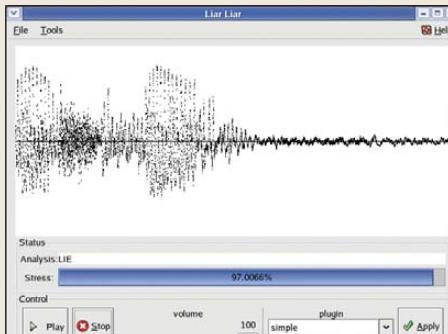
LiarLiar

liarliar.sourceforge.net

I'm dying to cover this project, but installing it is proving to be a pain! It sounds great though: "LiarLiar is a voice-stress analysis tool for Linux. Voice-stress analysis, an alternative to the polygraph as a method for lie detection, is already widely used in police and insurance fraud investigations.

LiarLiar's main purpose is to detect stress in a person's voice.

Higher stress levels can be an indication that the person is not being truthful." I'm having problems with library requirements, and I can't reach the developers. If anyone out there can help get this working, please send me an e-mail!



LiarLiar

Storybook

storybook.intertec.ch

I ran out of space this month, but I hope to cover this next time. Any creative writers should check this project out. "Storybook is a free (open-source) novel-writing tool for creative writers, novelists and authors which will help you to keep an overview of multiple plot lines while writing books, novels or other written works." It can store all the information about your characters and locations in one place, as well as manage chapters, scenes, characters and locations.

Linux News
and Headlines
Delivered
To You



Linux Journal
topical RSS feeds
available

HARDWARE

The Motorola DROID

The Motorola DROID is a serious contender in the smartphone world. Android just keeps getting better. BRIAN CONNER

"But does it run Linux?" is a common refrain in some circles whenever a shiny new gadget hits the streets. In the case of the Motorola DROID, I am pleased to say it does, in the form of Google's Android mobile operating system version 2.0 (running Linux kernel version 2.6.29).

With its big, beautiful screen, fast processor, full qwerty keyboard and utilitarian looks, is the DROID a serious contender for the smartphone crown or just another pretender that's too little, too late?

Call Quality and Experience

Let's cover the basics first: call quality. It is a phone, after all. As a longtime user of another carrier and having had no prior experience with Verizon, I was pleasantly surprised by just how good the DROID (and the Verizon network) sounded. The improvement was noticeable to people I speak to regularly. The external speaker is well suited to its primary roles of speakerphone and ringtone playback, and it actually does a really nice job in media playback as well (an obvious lack of punch in the low end notwithstanding).

The dial pad is clean and well spaced, with large buttons. Once connected to a call, the buttons for doing useful actions (speaker, mute, three-way call and so on) are right where you need them to be (Figure 1). There isn't much else to say here. Making and receiving calls is straightforward and simple.

Screen

The screen is the first thing that catches your eye, and it definitely warrants the attention—in a word: stunning. In a side-by-side comparison with that other smartphone, the DROID screen is bigger, brighter, crisper and more clear (Figure 2).

Scrolling and dragging are smooth and fluid. The haptic feedback (feedback technology that interacts with one's sense of touch, for example, via vibration), where implemented, provides just the



Figure 1. DROID Screen during a Call

right amount of "buzz" without being distracting. The response to tapping is good, as long as you remember to use your fingertip and not your fingernail. "Tap" response does seem to taper off a bit near the top of the phone (when holding it in portrait mode), but I've noticed this really only in one application, so I'm not sure if it is the screen or the app.

One thing the DROID screen doesn't do is multitouch (multitouch technology allows user interaction by touching the screen in more than one place at the same time). This "shortcoming" has been well documented, but I haven't really found it to be a big deal. I am sure my opinion would be different if I were used to using a phone that did support multitouch.

Keyboard

Although not quite a holy war like with vi vs. emacs, the presence (or

absence) of a physical keyboard is definitely a polarizing factor among smartphone aficionados. Based on my previous experience with touchscreens, I knew going in that I wanted a physical keyboard, and this one, for the most part, hasn't disappointed.

To expose the keyboard, simply slide the screen up. There's no springs or hinges, just a satisfying click when fully opened or closed. The lack of hinges or springs is, in my opinion, a positive, because it lessens the chances of having a "loose screen" result from a worn-out mechanism. It also means less moving parts to break.

The keyboard itself takes up about three-quarters of the full width of the phone, with the remaining quarter being lost to the slide mechanisms, the microphone and a directional pad, which functions just like the arrow keys on a standard keyboard (Figure 3). The presence of



Figure 2. DROID Main (Home) Panel

the directional pad, much like the "chin" on the G1, makes right-hand placement a bit awkward initially; however, it took only an hour or so of use to adjust. A person with smaller hands may need a bit more practice to adjust, however.

All of the keys are the same size, with the exception of the spacebar, which is three keys wide. The lack of a physical space between the keys was

a concern initially, but is no longer. Each key is bubbled up slightly in the center. This subtle rise in the center of each key makes it easier to differentiate between one key and the next as your thumb slides across the board.

The keyboard is a standard qwerty layout, with each key having an alternate function. The presence of both an Alt and a Shift key in both bottom corners makes it very easy to switch lowercase, uppercase, numbers and symbols. The inclusion of physical keys that replicate the touchscreen Search and Menu buttons is a nice touch, as it allows you to use these functions while keeping your hands in typing position.

The most unusual thing about this keyboard is the presence of two unused "keys", one each in the absolute bottom left and right corners. It'd be great if they were user-programmable, but alas, they don't even click. Although this isn't really the kind of thing you expect to see on a finished product, I haven't found myself reaching for a key that isn't there, perhaps with the exception of having a Ctrl key available when using an SSH client.

In addition to the physical keyboard, there are also four buttons on the touchscreen: Back, Menu, Home and Search. Back does exactly what you'd expect, as does Search. Home drops you back to the main (center) panel from whatever you were doing, and Menu opens the options

and settings menu in most applications.

Besides the keyboard and soft keys, the DROID has three additional physical buttons: the power/lock switch, a volume toggle and the camera button. The power/lock button functions exactly as it should. The same can be said of the volume toggle, which is smart enough to adjust the volume of whatever you're doing at the time (that is, call volume, ringer volume and media playback volume). The placement of the volume toggle does make it difficult to adjust the volume when the keyboard is exposed.

Camera

I've seen some very nice photos taken with cell phones, and I've seen an equal number, if not more, of really bad ones. Because of this, I really had no way to gauge my expectations for the DROID camera. Now that I've had some time with it, I think the best way to describe it is that the camera can, in certain situations, take really nice photos.

The hardware is certainly there: 5 megapixel camera, dual function flash and a physical shutter button that is located on the top-right side—just like a point-and-shoot camera when holding the phone in landscape mode.

It is the software and interface where things aren't as positive. Upon first launch, I was very disappointed to see that the camera interface was inconsistent with every other application. A push of the Menu button while in camera mode doesn't launch the settings/options menu. Instead, it launches another menu from which you can access the settings. When opened, the settings panel slides out from the left of the screen, instead of from the bottom like in all other apps. Those were my initial thoughts, but after using this for a while, I am starting to think that the arrangement of the menus and controls was a deliberate attempt to make them easier to use when holding the phone as a camera (landscape). If that's the case, kudos to the Android development team for considering how it will be used and not just blindly making all the apps the same. My praise for them is diminished, however, by the fact that it still doesn't work in a manner consistent with what you would expect. As an example, say you've opened the settings panel, then selected the flash settings submenu and changed your flash-mode preference. At this point,



Figure 3. The DROID Open Showing the Keyboard and Browser

tapping the back button closes out all menus and takes you back to the main camera mode. In contrast, my expectation (and the behavior for every other app on the phone) is that tapping the back button while within a menu tree simply takes you up one menu level. This may not sound like a big deal, but the extra steps are a drag if you're trying to change more than one setting in a hurry.

Although the menu navigation can be tedious, the most unfortunate and disappointing drawback with this camera is the lag between button push and shutter fire when in autofocus mode. The lag actually seems more pronounced when using the physical shutter button than when using the soft (on-screen) button. This lag can be avoided by switching the focus from auto to infinity. The trade-off is, of course, that you lose the benefits of the autofocus. I don't know if the lag is something that can be fixed in software, but if it is, let's hope the developers are paying attention.

Regarding shooting mode, the camera has all the usual ones you'd expect on a typical point-and-shoot camera: auto, portrait, action, snow, beach, indoor and so on. It also has the usual palette of color effects: black and white, sepia, negative, red, blue or green tint and so on.

This certainly isn't meant to be a harsh indictment of the DROID camera. As I said, in the right circumstances, this camera definitely is capable of taking a nice photograph. But, let's keep in mind that this is a camera that's been added on to a phone and not the other way around. At the end of the day, you aren't going to replace your DSLR with the DROID (or any other phone for that matter).

Battery, Storage, Headphones and USB

The battery and memory card are housed in a compartment accessible through a removable panel on the back of the phone, and both are user-replaceable. The memory type is microSD.

The headphone jack takes a standard 3.5mm plug and is mounted in such as a way to require no special adapters or extenders.

Data transfer and battery charging are done via USB. The port on the phone accepts a standard micro USB 5-pin male connector.

These may seem like minor details;

however, they bear mentioning because they are features that aren't always found on other smartphones. These choices, I believe, are a nod to consumers who are tired of over-paying for accessories and tired of guessing which USB cable in the drawer has the right connector for the device at hand.

Interface

The DROID has three panels (think desktops) to hold the most commonly used applications and widgets. Switching between them is as simple as a flick of the finger. When started, the center panel is populated mostly with communications apps (Phone, Contacts, Gmail, Text, Gtalk, Calendar and so on) and, of course, the search widget. Access to the full menu of applications is available via a shade window, which slides in from the bottom of the screen (Figure 4). A long press on an open area of the panel brings up a menu from which additional application shortcuts or widgets can be added. A long press, hold and drag is all that is needed to move icons around on a panel, or from panel to panel. A long press and hold also temporarily turns the application menu button into a trash can, should you decide to remove an app or widget from the panel.

One thing that makes the Android interface really stand out is the Notifications shade. The unsuspecting gray bar along the top of the screen that displays time, battery status, signal strength and network status also displays application notification icons (new voice-mail, missed call, unread e-mail and so on). With a flick, however, the shade unrolls to fill the screen and provides additional information about your notifications (Figure 5). At a glance, you can see the first sentence of unread e-mail messages, texts or tweets. You also get details on downloads that have completed, application updates that are available or Wi-Fi networks in the area. A simple tap on a notification will take you right to the application. I was pleasantly surprised to discover that the notification system works equally well with third-party apps as it does with Google applications.

Google Integration and Search

Because this is an Android phone, you would expect integration with Google's various services to be tight, and it is for some services. For others, it's nonexis-



Figure 4. The DROID Main (Home) Panel with Application Panel Shade

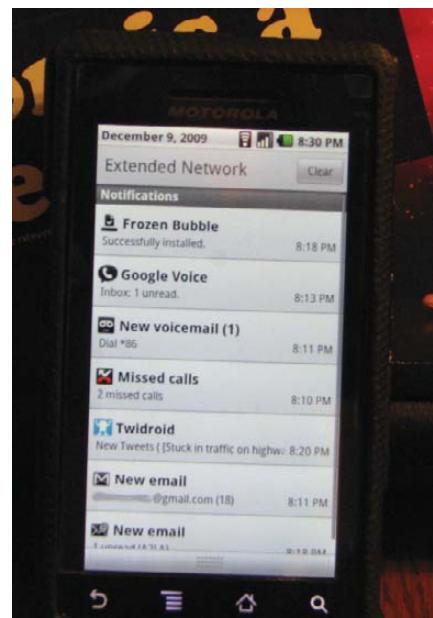


Figure 5. The DROID Notifications Shade

tent or inconsistent. A single sign-on the first day I had the DROID gave me Gmail, Google Calendar and Gtalk at my fingertips without a second thought. Because of this seamless integration, I was expecting similar ease when visiting Google Docs and Google Reader via the native browser, but I was disappointed.

I concluded that this was a security measure related to the fact that the first three are standalone apps, while the last two run in-browser. Yes, this explanation made sense until the first time I visited Google.com and discovered that I was signed in by default. The bigger question is, of course, why native apps for Docs and Reader aren't available.

Like anything from Google, the search functionality is quite good. The interface is consistent across all apps (even third-party) and uses predictive text, as well as your search history, to begin presenting results as soon as you start typing. Voice search performs well, provided you speak clearly and limit background noise.

Multitasking

The ability to run multiple applications at the same time is another feature that sets DROID apart from most of the other smartphones on the market. The classic example of opening a Web page, leaving to check e-mail and returning to find the page fully loaded works as advertised. As a more challenging test, I started Google Navigation and had it provide me turn-by-turn directions for my route. I then started playback of the *Linux Journal* Insider podcast. I wasn't sure how DROID would manage the conflicting audio streams, but it did the sensible thing by allowing the navigation prompts to interrupt Shawn and Kyle at the required times.

Apps and Widgets

The DROID comes pre-installed with a fairly well-rounded collection of productivity, entertainment and social-media applications. More notable than the applications that were included, however, were those that weren't. The three most glaring omissions being a weather application (or widget), a Twitter application and a file browser. Fortunately, good third-party solutions exist for all three.

Regarding third-party applications, the Android apps store has convinced me that more isn't always better. After many hours browsing the store, it appears as though there is enough variety to cover just about everything you'd want to do with your phone (and probably a few things you wouldn't). For me, I've had no trouble finding applications for most of the things I've wanted to do. The one shortcoming is the availability of applications for certain popular Web

services. This is beginning to change, however, even in the short time that DROID has been on the market.

In searching the app store, I've come across a couple applications developed by Google that aren't on the phone by default, but probably should be included. The first is Listen, a podcast aggregator that includes the very nice feature of allowing downloads right to the DROID over 3G or Wi-Fi (although the app warns of the potential for data-overage charges if you use 3G) and even allows you to begin playback before the download is completed. The second is a client for Voice, Google's telephony power tool, that is available by invitation only. Google Voice users know it's a nice way of managing your voice calling and SMS through one common interface. Once available to the public, Google Voice definitely should be a standard part of the Android operating system.

The DROID also comes supplied with a small collection of useful widgets. The search widget, of course, is included, and

it provides fast access to search. The widget I've found useful is the one for power management. It includes status indicators and toggles for Wi-Fi, Bluetooth, GPS, sync and screen brightness.

Conclusion

The all-around conclusion is that the DROID is a solid phone. Unlike some of its Android-based predecessors, the hardware finally has the muscle to implement the platform the way the developers intended. Only time will tell if the DROID is a serious contender for the smartphone crown, but if you're a devoted user of Google's cloud services (Gmail, Calendar and so on), you'd be hard-pressed to find a phone that does a better job of integrating these services into your daily routine. ■

Brian Conner, an Internet junkie working for a small nonprofit in western Maryland, blows off the steam of helping to manage a proprietary OS environment by poking and prodding at his favorite distro: Slackware. When not in front of a monitor, Brian can be found photographing his two beautiful daughters (and his beautiful wife), enjoying college football and reading.

DEDICATED SERVERS WITH BLAZING FAST CONNECTIONS

BLADE SERVERS

Gigabit ports / MULTI-Gig options

High-capacity bandwidth plans, including:

- * 3000 GB/month for \$200
- * 5000 GB/month for \$375
- * 10000 GB/month for \$800

Custom clusters with private VLANs

Flexible storage and RAID options

Intel Premium Partner

Numerous OS choices (Linux or Windows)

FREE 24x7 "6-Star" support

www.CARI.NET/LJ
888.221.5902

carnet
Better Servers. Better Service

Developing Flash Applications with

FLEX BUILDER

Flex Builder is built on top of Eclipse, and it allows you to develop Flash applications on Linux.

CARL FINK

I miss programming. For years, I developed software for large companies, first using Lotus Notes and then Macromedia Flash. I have spent the past three years in a totally nonprogramming job, developing Web sites, writing whitepapers and otherwise doing nontechnical things. I like writing, but I miss programming. I wanted to get back into development, and as a *Linux Journal* writer, I wanted to use free software if possible. The thing is, my background is all in not only proprietary but also highly specialized software. I have never programmed in C or Java or Perl or any of the other popular general-purpose languages. I certainly could learn one, but I would prefer not to start from scratch with a whole new system at this stage of my career.

One way forward was to create Flash applications in Flex. Flash originally was created as an animation player. Its scripting language, ActionScript, originally was a way to script object animation, but it has developed into a full-featured object-oriented programming environment for general-purpose applications. It is an implementation of the ECMAScript 4 draft, meaning that it is compatible with the newest JavaScript engines. However, many traditionally trained developers find Flash's animation-oriented timeline development system confusing and unfamiliar. The solution was the development of Flex, which allows programmers to create Flash applications using more standard tools and the more familiar metaphor of drawing controls on a form and assigning them behaviors.

After Adobe bought Macromedia, it released the Flex SDK

under the Mozilla Public License, so it now is possible to develop Flash applications using entirely free software. It also has released an alpha version of its Eclipse-based Flex Builder development environment for Linux. Flex Builder is released under a proprietary license.

Flex applications also can be defined using MXML, an XML dialect that is used to lay out the user interface and other aspects of the program, such as data bindings. Behaviors still are defined using ActionScript.

Flash applications generally run in the browser. They offer many of the advantages of AJAX or Silverlight applications, including a stateful client that can update specific items without reloading the entire page, and the Flash Player sandboxes applications in much the same way that a Java applet is restricted for security reasons.

Another innovation, AIR (Adobe Integrated Runtime) lets applications run off-line. AIR lets ActionScript developers create true freestanding programs that do not require a Web browser. However, they still are restricted by the Flash "sandbox", which limits what changes can be made on the local system. AIR apps also can include HTML and JavaScript.

Perhaps the two best-known AIR applications are the Pandora Internet radio player and TweetDeck, which streamlines the Twitter experience. Both work on Linux.

In this article, I demonstrate how to create a simple Flash application using Adobe Flex Builder on a Linux system. In a follow-up article, I'll move on to totally open-source development using Project Sprouts.

Installing Flex Builder 3 Alpha for Linux

Flex is an Eclipse-based environment. In order to use it, you must have certain prerequisite software installed: Eclipse 3.3.x, Sun JRE 1.5 or newer and Mozilla Firefox 3.0.

Note that the system requirements refer specifically to Eclipse 3.3. If you use a higher version, installation will succeed, but Eclipse will fail to open code editors. I installed version 3.3.2 from eclipse.org (see Resources) in my \$HOME directory. You can install 3.3 alongside 3.5 on the same computer, as long as you start version 3.3 to use Flex. Simply untar the download and place it anywhere in the filesystem. I put it in \$HOME/eclipse.

I was able to use Flex Builder with Mozilla Firefox version 3.5, however, rather than the called-for 3.0 without problems. One Firefox tip: I use the NoScript plugin. At first, I thought the context-sensitive help in Flex Builder wasn't working, but it turns out that I had to allow scripts

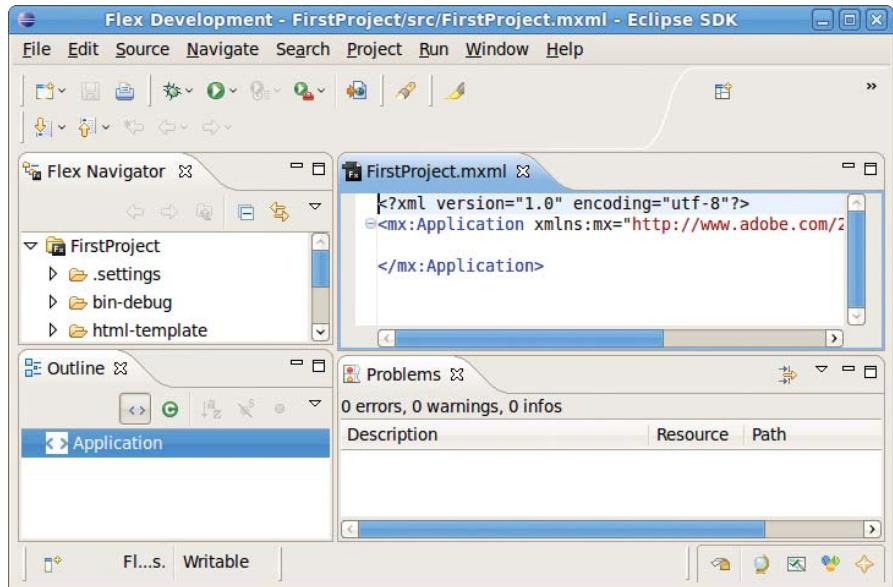


Figure 1. Flex Builder

from 127.0.0.1:51296.

Also, note that you *must* install the Sun JRE. GCJ will not work with Flex.

To make debugging work, you must download and install the debugging version of the Flash Player (see

An advertisement for Black Hat Europe 2010. The background features a dark, red-tinted image of a person's profile, possibly wearing a mask. Overlaid text includes "DECRYPTING SECURITY INDE", "black hat® europe", "DIGITAL SELF DEFENSE", "(april 12 - 15)", "BARCELONA, SPAIN", "HOTEL REY JUAN CARLOS", and a descriptive paragraph about the event's purpose and registration information.

DECRYPTING SECURITY INDE

black hat® europe

DIGITAL SELF DEFENSE

(april 12 - 15)

BARCELONA, SPAIN

HOTEL REY JUAN CARLOS

Understanding the increasingly complex threats posed to an enterprise is a daunting task for today's security professional. The knowledge to secure an enterprise against those threats is invaluable. Come to Black Hat and learn from the industry's best. Register for Black Hat Europe today at www.blackhat.com

Listing 1. Quiz Program

```
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
    layout="absolute">
<mx:TextArea width="75%" height="75%">
<mx:text>
    Question 1: Which strip is this grouchy but
    good-hearted fighter the star of?
</mx:text>
</mx:TextArea>
</mx:Application>
```

Resources). Amusingly, when you try to run Adobe's installer on Ubuntu 9.10 (Karmic Koala), it complains because you don't have libc6 "higher" than 2.3. In fact, Karmic ships with version 2.10 (read as "2 dot 10"), which is higher than 2.3 in version-speak but not in normal numbering. I edited the script to remove the version check by commenting out these lines:

```
#GLIBCSTATUS=`check_glibc`
#case $GLIBCSTATUS in
#  invalid-glibc)
#    exit_glibc
#    ;;
#esac
```

With those edits, the install completed without further problems.

You can download the Flex alpha from Adobe's Web site (see Resources), and you need to create a free account first. Once

GNOME terminal (Figure 1).

The first time you run Eclipse after installing Flex Builder, you must create a new Workspace. Simply click File→Switch Workspace→Other and create a new folder.

Flex Builder for Linux, as an alpha, is missing several features present in the Windows and Mac versions:

- Design View
- States View
- Refactoring
- Data Wizards

The solution was the development of Flex, which allows programmers to create Flash applications using more standard tools and the more familiar metaphor of drawing controls on a form and assigning them behaviors.

you have it downloaded, do a chmod u+x on the file and run the downloaded file to install. Flex Builder uses a Windows-style graphical wizard installer. I installed into /home/carlf/AdobeFlexBuilderLinux, which meant I did not need to become superuser to complete the installation.

To use Flex Builder, simply start Eclipse. Being old-school, I did this by typing ./eclipse/eclipse & in a

- Cold Fusion Data Services Wizard
- Web Services Introspection
- Profiler

Depending on the type of project you are planning, these features may be either critical or unimportant.

Because this was my first experi-

Listing 2. Quiz Program with Font and Layout Fixes

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Example for LJ article. --&gt;
&lt;mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
    layout="absolute"&gt;
&lt;mx:TextArea width="75%" height="75%"&gt;
    paddingTop="10" paddingBottom="10"
    paddingLeft="10" paddingRight="10"
    id="Question" editable="false"&gt;
&lt;mx:htmlText&gt;
&lt;! [CDATA[
&lt;font size="+3" face="Arial"&gt;
&lt;b&gt;
    Question 1: Which strip is this grouchy but
    good-hearted fighter the star of?
&lt;/b&gt;
&lt;/font&gt;
]]&gt;
&lt;/mx:htmlText&gt;
&lt;/mx:TextArea&gt;
&lt;/mx:Application&gt;</pre>
```

ence with Eclipse, I took time to review the Eclipse tutorials before closing the Welcome screen. To switch from the default Java development environment to Flex, click Windows→Open Perspective→Other, and select Flex Development. Now, create a Flex project by clicking File→New→Flex Project. I chose to create a browser-based SWF file and named it "FirstProject".

For this first simple application, I decided to create a simple Internet quiz that asks the user some questions, then supplies a "Webcomics IQ" score (I'm a big fan of Webcomics). This let me avoid having to worry about server database access on my first project. For this project, I need to use MXML to draw a simple form, which contains a question (text field) and four possible answers (radio buttons), along with a Next button. When the user clicks Next, the next question is displayed in the text field. After the last question, the score is displayed.

Because Flex Builder for Linux lacks a GUI painter (the Design View is absent), I created the components by typing MXML code into the editor. First, I write the text of the first question.

When run, the program looks like Figure 2.

As you see, the text is very small. You can set the text size by using

Listing 3. Quiz Program with First ActionScript Code

```
<?xml version="1.0" encoding="utf-8"?>
<!--Example for LJ article. -->
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
    layout="vertical"
    backgroundColor="#FFFFFF">
<mx:TextArea id="Question"
    width="100%" height="15%"
    paddingTop="10" paddingBottom="10"
    paddingLeft="10" paddingRight="10"
    editable="false"
    backgroundColor="#FFFFFF" borderColor="#FFFFFF">
<mx:htmlText>
<![CDATA[
<font size="+4" face="Arial">
<b>
    Question 1: Which strip is this grouchy but
    good-hearted fighter the star of?
</b>
</font>
]]>
</mx:htmlText>
</mx:TextArea>
<mx:VBox paddingLeft="150"
    backgroundColor="#FFFFFF" width="100%">
<mx:RadioButton id="a1" groupName="Answers"
    label="Belkar Bitterleaf"
    width="400" paddingRight="20" />
<mx:RadioButton id="a2" groupName="Answers"
    label="Gilgamesh Wulfenbach"
    width="400" paddingRight="20" />
<mx:RadioButton id="a3" groupName="Answers"
    label="Roy Greenhilt"
    width="400" paddingRight="20" />
<mx:RadioButton id="a4" groupName="Answers"
    label="Frank Mangle"
    width="400" paddingRight="20" />
<mx:Button id="nextButton"
    label="Next" click="parseanswers();"/>
</mx:VBox>
<mx:Script>
<![CDATA[
public function parseanswers(): void
{
    import mx.controls.Alert;
    if (a3.selected) {
        Alert.show('Yes, the answer is ' + a3.label,
            'Right!', mx.controls.Alert.OK);
    }
    else {
        Alert.show('Sorry, no.', 'Wrong', mx.controls.Alert.OK);
    }
}]]>
</mx:Script>
</mx:Application>
```

htmltext instead of text. I also corrected the problem that the text is too close to the borders of the movie by adding padding. I assigned an ID (name) to the control, so I can refer to it in scripts, and I made it non-editable, which then gives us Listing 2.

I still need to add the answer selections as radio buttons and a Next button. In Listing 3, I have added our first bit of ActionScript, a function that evaluates whether the correct answer is selected and gives immediate feedback by way of a dialog box. Anything other than MXML in a project file is best kept inside CDATA tags, which prevent Flex from parsing it as XML. This applies to both ActionScript and HTML. ActionScript also can be stored in external files and loaded at runtime or during compilation.

Running the program now produces a single question, and clicking Next produces a simple message box (Figure 3).

The dialog and other controls don't look "standard" for most operating

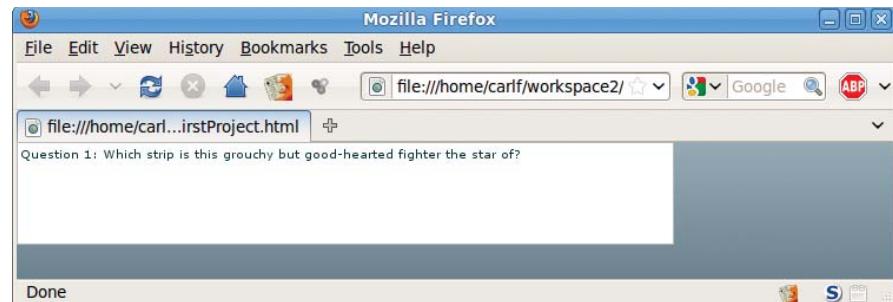


Figure 2. First Run of the Quiz Program as Seen in Firefox



Figure 3. Quiz Program with Answer-Checking

FEATURE Developing Flash Applications with Flex Builder

Listing 4. Quiz Program with More Questions

```
<?xml version="1.0" encoding="utf-8"?>
<!--Example for LJ article. -->
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
    layout="vertical"
    backgroundColor="#FFFFFF">
<mx:TextArea id="Question"
    width="100%" height="15%"
    paddingTop="10" paddingBottom="10"
    paddingLeft="10" paddingRight="10"
    editable="false" backgroundColor="#FFFFFF"
    borderColor="#FFFFFF"
    creationComplete="initApp();">
<mx:htmlText>
<! [CDATA[
<font size="+4" face="Arial">
<b>
    Question 1: Which strip is this grouchy but
    good-hearted fighter the star of?
</b>
</font>
]]>
</mx:htmlText>
</mx:TextArea>
<mx:VBox paddingLeft="150"
    backgroundColor="#FFFFFF" width="100%">
<mx:RadioButtonGroup id="Answers" />
<mx:RadioButton id="a1" groupName="Answers" value="1"
    label="Belkar Bitterleaf"
    width="400" paddingRight="20" />
<mx:RadioButton id="a2" groupName="Answers" value="2"
    label="Gilgamesh Wulfenbach"
    width="400" paddingRight="20" />
<mx:RadioButton id="a3" groupName="Answers" value="3"
    label="Roy Greenhilt"
    width="400" paddingRight="20" />
<mx:RadioButton id="a4" groupName="Answers" value="4"
    label="Frank Mangle"
    width="400" paddingRight="20" />
<mx:Button id="nextButton"
    label="Next" click="parseanswers();"/>
</mx:VBox>
<mx:Script>
<! [CDATA[
import mx.controls.Alert;
// Define here to make variables global.
var currentQuestion:int = 0;
var totalRight:int = 0;
// Initialize the array of questions.
// - First sub-array is question text
// - Second through fifth are possible answers
// - Sixth is the correct answer for that question.
var questions:Array = new Array(
    new Array("This grouchy but ... Order of the Stick.",
        "In Kevin and Kell, why ... secretaries?",
        "Which of these strips is NOT a stick-figure comic?"),
    new Array("Belkar Bitterleaf",
        "High metabolism means lots of work done.",
        "Questionable Content"),
    new Array("Gilgamesh Wulfenback",
        "Hollow bones mean no carpal tunnel",
        "No Time for Life"),
    new Array("Roy Greenhilt",
        "Excellent language skills of parrots",
        "xkcd"),
    new Array("Frank Mangle",
        "Flight lets them double as messengers",
        "Cyanide and Happiness"),
    new Array(3,2,1));
private function initApp(): void
// Initializes the first question, removing the default text
// that's there when the controls are created
{
    // testing
    Question.text=questions[0][currentQuestion];
    a1.label=questions[1][currentQuestion];
    a2.label=questions[2][currentQuestion];
    a3.label=questions[3][currentQuestion];
    a4.label=questions[4][currentQuestion];
}
public function parseanswers():void
// Function runs whenever the user clicks the Next button.
// Updates the score and puts up the new question, except on
// the final question, where it displays the user's final tally.
{
    if (Answers.selectedValue == questions[5][currentQuestion]) {
        totalRight++;
    }
    // OK, Next was pressed so it's time to update the
    // screen. Test whether this was the last question,
    if (currentQuestion >= 2) {
        // That was the last question, time to report
        // results. For simplicity I will use
        // the Alert function here
        Alert.show('You got '+ totalRight
            + ' questions right out of '
            + 3, 'Score', mx.controls.Alert.OK);
        // Since the quiz is over, I disable all the
        // controls on the screen. In a polished
        // version, I will handle the end of quiz by switching
        // to a different Flash file showing the score only.
        Question.enabled=false;
        Answers.enabled=false;
        nextButton.enabled=false;
    }
    else {
        // OK, that wasn't the last question, so update
        // the question, all four answers and
        // increment currentQuestion
        currentQuestion++;
        Question.text=questions[0][currentQuestion];
        a1.label=questions[1][currentQuestion];
        a2.label=questions[2][currentQuestion];
        a3.label=questions[3][currentQuestion];
        a4.label=questions[4][currentQuestion];
        Answers.selection = null;
    }
}
]]>
</mx:Script>
</mx:Application>
```

systems, and developers will want to customize them. Flex and Flash support various “skinning” techniques that make it simple to change the appearance of controls, but those are beyond the scope of this article.

Obviously, this version of the quiz is only for testing purposes. It has one question and no provision for tabulating results. Now, it's time to create more questions. Because I'm deliberately not connecting to a server-side database for this article, I simply declared an array of data directly in the program's code.

It's a peculiarity of ActionScript (like its parent, ECMAScript) that it doesn't directly support multidimensional arrays. The workaround is to declare an array of arrays, as shown in Listing 4.

I made my question arrays and index variable global. I know it's frowned on, but it's convenient. Global variables must be defined outside all functions, so here I defined them immediately at the beginning of the code, before any function definitions.

In the application definition, I added the MXML `creationComplete="initApp();"`, which says to run the function initApp after the form is initialized. initApp replaces the default text of the question and answers with the contents of the first column of the array.

For this article, the application is complete (Listing 4 shows the full, final code).

Flex is commercial software, totally nonfree (as in speech). It retails for \$249. I worked with the trial version, which is free as in beer. It's labeled an alpha, but it worked extremely well. It literally never crashed. The missing features listed above didn't affect me much, but I'm not an experienced Eclipse or Flex user who might be depending on those things.

Adobe hasn't announced any plans to release Flex Builder 4 for Linux. However, it did just extend the free license of the Flex 3 alpha for more than a year. I was intrigued to find a Flex Builder project at Google Code, fb4linux (see Resources). A programmer is trying to single-handedly convert the Windows version to run on Linux. I installed it, and it seems to work surprisingly well. Unfortunately, the “More info” link leads to sole developer eshangrao's personal site, which is written in Chinese. Because I can't read Chinese, I can't say much more about it. The project is distributed with the original Adobe license and still requires a license code from Adobe to work.

In my next article on this discussion of Linux-based Flash/Flex development, I will evaluate Sprouts, a Ruby-centered development environment. Unlike Flex Builder, Sprouts is released under an open-source license (the MIT license). Sprouts is a command-line-only compiler and debugger, built around the Rake build language. I will give an overview of using Sprouts, how to port Flex Builder projects into the Sprouts environment, and I'll also finish the work on my Webcomics quiz, reading the questions from a server-based database and improving the appearance of the screens.■

Having been a programmer, writer, trainer, teacher and several other things, Carl Fink is not what you'd call a specialist. You can read his blog at nitpicking.com.

Resources

Adobe AIR: www.adobe.com/products/air

Eclipse Platform Downloads (use 3.3.2 for Flex Builder 3 alpha): archive.eclipse.org/eclipse/downloads/index.php

Debug Version of Flash Player 10: download.macromedia.com/pub/flashplayer/updaters/10/flash_player_10_linux_dev.tar.gz

Flex Builder 3 Alpha for Linux (also includes installation instructions): labs.adobe.com/technologies/flex/flexbuilder_linux

Flex Builder 4 for Linux: code.google.com/p/fb4linux

Project Sprouts: projectsprouts.org

Flex Examples (very helpful): blog.flexexamples.com

EMBEDDED SERVER



• Fanless x86 500MHz/1GHz CPU
• 512MB/1GB DDR2 RAM On Board
• 4GB Compact Flash Disk
• 10/100 Base-T Ethernet
• Reliable (No CPU Fan or Disk Drive)
• Two RS-232 Ports
• Four USB 2.0 Ports
• On Board Audio
• Dimensions: 4.9 x 4.7 x 1.7" (125 x 120 x 44mm)

2.6 KERNEL

Standard SIB
(Server-In-a-Box)
Starting at \$305
Quantity 1.

• Locked Compact Flash Access
• Analog SVGA 3D Video
• Optional Wireless LAN
• EMAC Linux 2.6 Kernel
• XP Embedded & WinCE 6.0

www.emacinc.com/servers/standard_sib.htm

Since 1985
OVER
25
YEARS OF
SINGLE BOARD
SOLUTIONS

EMAC, inc.
EQUIPMENT MONITOR AND CONTROL

Phone: (618) 529-4525 • Fax: (618) 457-0110 • www.emacinc.com

MONGOOSE

an Embeddable Web Server in C

MONGOOSE provides a Web server that can be embedded in your application, and it consists of a single C source file (and a header file) that you can compile and link with your application code. If you need a simple Web server, Mongoose may be the solution.

Web services are all the rage in development circles, but full-featured application servers like JBoss are terrible overkill for small system solutions. In many situations, simple RESTful interfaces suffice without the need for complex containers. Additionally, embedded single file implementations may be preferred over collections of scripts that rely on an external interpreter, such as Python or PHP.

Mongoose is an MIT-licensed, embeddable Web server contained within a single C module library that can be embedded in a program to provide basic Web services. Its lightweight approach hides the power of a fully threaded system capable of serving both static and dynamic content over multiple ports using standard and secure HTTP protocols. It supports CGI and SSI, access control lists and digest authorization. File transfers are supported using code from example server implementations.

Along with the embeddable C library, the Mongoose package includes a front end that turns the module into a full-fledged server capable of serving files from a user-specified document root. The ready-to-run server supports all configurable options from the command line as well as from a text configuration file. Although this configuration provides a method to bring Mongoose up quickly, Mongoose's power comes from writing custom front ends to the Mongoose library with callbacks for specific REST-styled URLs.

In this article, I introduce the Mongoose library API, show how it can be configured for multiple uses and provide an example implementation that serves up static pages utilizing digest authentication. This article is aimed at developers with a knowledge of C programming. Although not required, familiarity with HTML and CSS also is useful.

MICHAEL J. HAMMEL

Structure of a Mongoose-Based Server

A Mongoose-based server provides a main() function that parses command-line arguments, initializes the Mongoose library and then spins in a loop waiting for an exit event. The command-line arguments are server-specific but typically represent values to be passed to the Mongoose library through the mg_set_option() function. Mongoose initialization requires creating an initial Mongoose context, setting library options and establishing callback functions for authorization, error and URI handling. The main function then spins forever while the Mongoose master thread handles incoming connections on the configured network ports. The front-end code is responsible for handling signals to support shutdown operations, including notifying the Mongoose library to stop gracefully.

The Mongoose library is threaded and extremely simple to use. The initial context starts the master thread, which waits for incoming connections. New connections are queued, and worker threads are started to handle them. Processing of the connection occurs in the worker thread. Here, the connection request is analyzed to determine how processing will proceed. For performance sake, worker threads are started as needed and remain after a connection closes to process any additional queued incoming requests without the overhead of starting new threads (Figure 1).

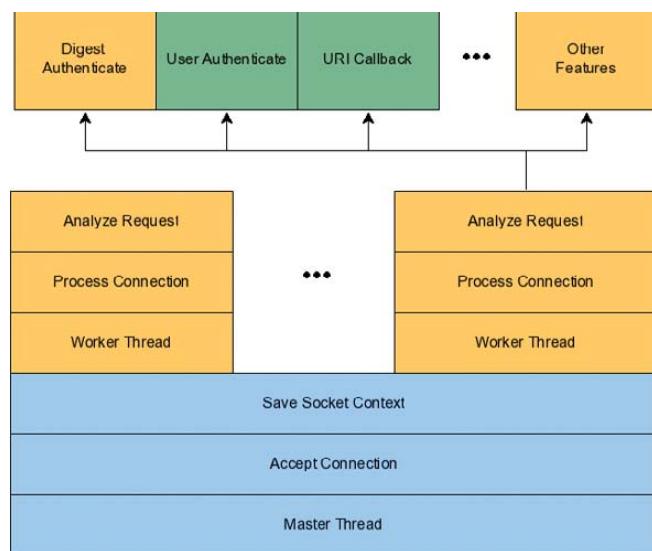


Figure 1. Green boxes are user functions, and all the rest are implemented in the Mongoose library.

In the worker thread, the incoming request is analyzed to determine what should happen next. Mongoose supports various HTTP requests, such as PUT, POST and DELETE. However, for simple REST services, the most important feature Mongoose supports is callbacks for URIs, error handling and authentication.

Hello, World: Initialization and Callbacks

Mongoose initialization is handled within a single user-defined function called from the main() function. This initialization function performs three mandatory operations: start the Mongoose initial context (mg_start), set options on that context (mg_set_option) and specify URI callbacks

(mg_set_uri_callback):

```
void mongooseMgrInit()
{
    struct mg_context *ctx;
    ctx = mg_start();
    mg_set_option(ctx, "ports", port);
    mg_set_uri_callback(ctx, "/*", &uriHandler, NULL);
}
```

The Mongoose API documentation is sparse, and the available options are not obvious. Fortunately, the man page for the default front end documents command-line options, which in turn map directly to most of the available options that can be set with mg_set_option(). The -A option to the default server, used to edit a digest authentication file, is not supported by the Mongoose library. The default server supports most, but not all, available Mongoose library options via the command line. The known_options array in mongoose.c defines the list of options directly supported by the Mongoose library.

All arguments to mg_set_option() options are character strings. Mongoose converts them to appropriate formats as needed. For example, the port number for the ports option must be specified as one or more character strings separated by commas, with SSL ports identified with the letter s appended to the port number. Some options can be disabled at compile time. To disable SSL options, define NO_SSL. To disable CGI options, define NO_CGI.

Callbacks, set with mg_set_uri_callback(), are functions that handle specific URI requests. The asterisk is used as a path wild card. In this simple example, there is a single callback handler that handles all URI requests starting at the root path for the Web server.

To complete this example of the Mongoose equivalent of the “Hello, World” program, all that is required is a function for printing a page back to the requesting Web browser:

```
void uriHandler()
{
    mg_printf(conn,
        "HTTP/1.1 200 OK\r\n"
        "Content-Type: text/html\r\n\r\n"
        "<html>\r\n"
        "<body>\r\n"
        "Hello, World!\r\n"
        "</body>\r\n"
        "</html>\r\n"
    );
}
```

The first two lines are HTTP headers. These are not

NOTE: Arguments to mg_set_option() are case-sensitive. For a quick reference of the Mongoose library options, refer to the command-line options shown in the Mongoose Manual on the Web site, using the same case, but remove the dash prefix.

necessarily required for such a simple example, but if you do include them, remember to include a blank line between the Content-Type header and the start of the page content.

Two functions in the Mongoose library are used for sending results back to the browser: mg_printf() and mg_write(). The former provides printf() semantics for sending data back to the client, and the latter provides no limits on the amount of data that can be sent. If the server needs to know that the client closed the connection before all data was returned, or if the server needs to send more than MAX_REQUEST_SIZE (8Kb) of data, mg_write() should be used. Note that the API documentation says the maximum size for mg_printf() is 16Kb, but the Mongoose library source code defaults MAX_REQUEST_SIZE to 8Kb. Multiple mg_printf() or mg_write() calls are possible within a single callback; however, once the callback returns, the connection is closed by the worker thread.

Authentication and Authorization

In the Web world, *authentication* refers to validation of an incoming request as having come from a known entity. All that is required is that the entity identify itself with tokens kept by the system. In the case of digest authentication, that means a user name, password and realm. A realm is a symbolic name allowing the same user name/password to have different meanings to different areas of a server URI namespace. In practice, users need remember only the user name/password combination. The realm is managed by the server. Digest authentication has additional complexities related to how the server and client communicate, but from the standpoint of Mongoose users, this is not required knowledge. In summary, authentication is used to identify a user.

Authorization refers to the verification that an authenticated entity has permission to do what it is attempting to do. Although people may have a proper login to a server, they may not have permission to view certain areas of the Web site. Access to specific server functionality is handled by authorization.

Mongoose provides built-in support for digest authentication. If configured, a file containing a user name, password and realm is stored within reach of the server at runtime. The server checks this file for authentication based on HTTP Authentication headers from the browser. A global authentication file can be configured as well as per-URI authentication files. Mongoose users can generate these files using Apache's htdigest program. The location of the file is set during Mongoose initialization using the mg_set_option() function. The realm defaults to "mydomain.com" if not specified. Digest authentication is not required.

The auth_gpass option sets the location of a global authentication file. This file is used to authenticate requests for any URI. The argument for this option is the path to the file. To set authentication for specific URIs, use the protect option. The argument to this option is a collection of comma-separated URI=PATH pairs, with URI being relative to the Web server and containing wild cards, and PATH being a path to the authentication file to use for that URI. Paths should be fully qualified or relative to the directory from which the Mongoose-based server is started.

Listing 1. The fully qualified path to the digest authentication file and associated realm are set in options, and a callback is specified to perform server-specific authorization.

```
#define DOCROOT    "/docs"
#define HTPASSWD   "./htpasswd"
#define port        "8083"
void mongooseMgrInit()
{
    struct mg_context *ctx;
    char *ptr = NULL;
    char *documentRoot[PATH_MAX];
    char htpath[PATH_MAX];

    ptr = getcwd(NULL, 0);
    memset(documentRoot, 0, PATH_MAX);
    strcpy(documentRoot, ptr);
    documentRoot = strncat(documentRoot, DOCROOT);

    memset(htpath, 0, PATH_MAX);
    strcpy(htpath, documentRoot);
    strncat(htpath, "./htpasswd");

    ctx = mg_start();
    mg_set_option(ctx, "ports", port);
    mg_set_uri_callback(ctx, "/*", &uriHandler, NULL);

    mg_set_option(ctx, "auth_gpass", htpath);
    mg_set_option(ctx, "auth_realm",
                  "mongoose-example.com");
    mg_set_auth_callback(ctx, "/*", &authorize, NULL);
}
```

If an authentication option is set for a requested URI, Mongoose will tell the client browser to open a login dialog. The Mongoose library processes the login information from the user before passing control to the appropriate callback, if any. Once the browser user is authenticated, the only way to log out is to request the authentication again. It turns out that this form of authentication requires cookies to implement the logout process and force another login. Alternatively, cookies can be used to implement a page with HTML forms for the purpose of login and logout outside the use of digest authentication. If logout or a server-side form is required for login, digest authentication probably should not be set with Mongoose options. Digest authentication still can be used manually, but Mongoose does not expose API functions for this purpose.

Along with authentication, authorization can be implemented using a callback registered with the mg_set_auth_callback() function. The registered function is called before each URI callback to allow the server code to determine whether the incoming request should be authorized to access the requested URI. If authorization is granted, this function calls mg_authorize() on the provided mg_connection. If this is not done, Mongoose assumes authorization is not granted and will not call the configured callback for the requested URI:

```

static
void authorize(
    struct mg_connection *conn,
    const struct mg_request_info *ri,
    void *data)
{
    const char *cookie, *domain;
    cookie = mg_get_header(conn, "Cookie");
    uri = ri->uri;

    if ( (strcmp(ri->uri, "/") == 0) ||
        (strncmp(ri->uri, "/images", 7) == 0) )
    {
        mg_authorize(conn);
    }
    else if (strncmp(ri->uri, "/logout", 7) == 0)
    {
        ... Verify login cookie ...
        ... redirect to front page ...
    }
    else if (cookie != NULL &&
              strstr(cookie, "UUID=") != NULL)
    {
        ... Get value from the cookie, if any ...
        if ( ...cookie okay ... )
            mg_authorize(conn);
        else
            ... redirect to /logout ...
    }
}

```

mg_authorize(conn);

else

... redirect to /logout ...

}

}

Note the arguments to the authorize() function. The first argument is the connection information. The second is a pointer to request information pulled from the incoming HTTP request. The third argument points to data provided when the callback was registered with mg_set_auth_callback(). These same arguments are used when URI callback functions are called.

In this example authorization function, any request for the front page or the images directory within the document root are authorized automatically. This allows images referenced in CSS, for example, to be retrieved by the browser without having to be inspected by this function or by having a registered URI callback for images. If no callback is registered for a URI, Mongoose attempts to serve the file found at the specified URI under the document root.

If the URI is the logout page, the login cookie is checked for, and if found, the user is redirected to the login page where that cookie is removed. If the cookie is not found, the server can redirect to the front page anyway or perform some other appropriate action.

IN THE VERY NEAR FUTURE

~~In the future when cars fly, geeks and fans will come together for one thing...~~

...FREE BEER.
(well, that, and a gathering of tech, sci-fi, and fantasy enthusiasts.)

STARRING:

karl SCHROEDER howard TAYLER joe BROCKMEIER marcel GAGNÉ

...and the virtual presence of spider & jeanne ROBINSON

PENGUICON2010

THE YEAR WE MAKE IMPACT

APRIL 30 - MAY 2, TROY MI
details and registration @ www.penguicon.org

FEATURE Mongoose: an Embeddable Web Server in C

The next test looks for a specific cookie, in this case named "UUID". If this is found and has the correct value, the request is authorized. Otherwise, the user is redirected to the logout page, which in turn cleans up the login cookie and presents the login page again.

The mg_request_info structure is defined in mongoose.h and is filled by the worker thread with information gleaned from the HTTP request. This includes information such as the request method (POST, PUT, GET and so forth), a normalized URI, query string, post data and the IP address from which the request originated. It also includes an array holding the set of HTTP headers, which is how cookies are retrieved.

Cookies

The mg_get_header() function is used to retrieve a named header from the mg_request_info's mg_header array. Cookies are set in a header before the start of the document content:

```
mg_printf(conn,
    "HTTP/1.1 200 OK\r\n"
    "Set-Cookie: UUID=MGOOSE;\r\n"
    "Set-Cookie: LOGIN=;\r\n"
    "Content-Type: text/html\r\n\r\n"
    ...
}
```

The 1994–2009 Archive CD, back issues, and more!

www.LinuxJournalStore.com

This code would set the UUID cookie and clear the LOGIN cookie on the browser. Retrieving a particular cookie requires pulling the cookie header and parsing it for the desired cookie name/value pair:

```
static
char *getCookieParam(const char *cookie, char *param)
{
    char *start = NULL;
    char *end   = NULL;
    char *value = NULL;
    int length;

    if ( (cookie!=NULL) &&
        ((start=strstr(cookie, param)) != NULL) )
    {
        if ( (end=strstr(start, "; ")) != NULL )
            length = end-start;
        else
            length = strlen(start);
        value = malloc(length+1);
        memset(value, 0, length+1);
        strncpy(value, start, length);
    }
    return value;
}
```

This function will retrieve both the name of the cookie and its value, if any, as NAME=VALUE. The returned character string must be freed by the caller, however.

Serving Static Files

When Mongoose encounters a URI without a registered callback, it attempts to open the specified file and send it back to the client. Static HTML files can be written and stored in a document root configured with the root option. To allow retrieving directory listings of directories under the document root, set the dir_list option to yes. This option defaults to no. The directory list setting is a global configuration, so either no directory listings are allowed, or all of them can be seen.

Listing 2. Because the dir_list option does not match a true value, it will disable directory listings.

```
void mongooseMgrInit()
{
    ...
    mg_set_option(ctx, "dir", documentRoot);
    mg_set_option(ctx, "dir_list", "0");
    ...
}
```

Server Logging

Mongoose provides access and error logging. The files are appended on restart of the server. Mongoose provides two functions available from the API that are documented in the Mongoose API page on the Web site: mg_set_error_callback()

NOTE: Options that allow yes/no or true/false values are tested against the case-insensitive string values of "1", "yes", "true" or "ja". Any other value is interpreted as no or false.

and mg_set_log_callback. These callbacks have a slightly different configuration from URL callbacks:

```
void mongooseMgrInit()
{
    ...
    mg_set_error_callback(ctx, 404, show404, NULL)
    mg_set_log_callback(ctx, logger)
    ...
}
```

The error callback sets callbacks for the error codes from 0 to 1000. These map to HTTP error codes, such as 404 when a requested URI does not exist. When this callback function is called, the function can print a custom error page. The log callback is called any time the Mongoose server library wants to log something.

The source code for the sample server implemented using mongoose can be found at <ftp://linuxjournal.com/pub/lj/listings/issue192/10680.tgz>. It includes a single page with an image and CSS.

Summary

The Mongoose Project is stable and in use by a number of developers; however, the Google forums for it are littered with spam. Don't let this inconvenience prevent you from utilizing what is a well-designed and implemented Web server library.

This introduction to Mongoose covers the basics for creating a lightweight embedded Web server without covering the full breadth of Mongoose features, such as CGI or SSL. The ease of use of this library should make it plain that these extended features will require little additional knowledge of Mongoose and free developers to build custom Web servers.■

Michael J. Hammel is a Principal Software Engineer for Colorado Engineering, Inc. (CEI), in Colorado Springs, Colorado, with more than 20 years of software development and management experience. He has written more than 100 articles for numerous on-line and print magazines and is the author of three books on The GIMP, the premier open-source graphics editing package.

Resources

Mongoose: code.google.com/p/mongoose

Example Source: <ftp://linuxjournal.com/pub/lj/listings/issue192/10680.tgz>



Linux - FreeBSD - x86 Solaris - MS etc.



Proven technology. Proven reliability.

When you can't afford to take chances with your business data or productivity, rely on a GS-1245 Server powered by the Intel® Xeon® Processors.

Quad Core Woodcrest



2 Nodes & Up to 16 Cores - in 1U

Ideal for high density clustering in standard 1U form factor. Up to 16 Cores for high CPU needs. Easy to configure failover nodes. Features:

- 1U rack-optimized chassis (1.75in.)
- Up to 2 Quad Core Intel® Xeon® Woodcrest per Node with 1600 MHz system bus
- Up to 16 Woodcrest Cores Per 1U rackspace
- Up to 64GB DDR2.667 & 533 SDRAM Fully Buffered DIMM (FB-DIMM) Per Node
- Dual-port Gigabit Ethernet Per Node
- 2 SATA Removable HDD Per Node
- 1 (x8) PCI_Express Per Node



Servers :: Storage :: Appliances

Genstor Systems, Inc.

780 Montague Express. # 604
San Jose, CA 95131

[Www.genstor.com](http://www.genstor.com)
Email: sales@genstor.com
Phone: 1-877-25 SERVER or 1-408-383-0120



Intel®, Intel® Xeon®, Intel® Inside® are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Selenium Core provides basic Web application testing, and Selenium RC allows you to write test scripts in other programming languages, such as Perl, Java, C# and others.

Web Application Testing with SeleniUM

ALEXANDER SIROTKIN

Web 2.0 may pale in comparison to the days of the dot-com bubble and Web 1.0 from a financial point of view, but from a technical point of view, it's light-years ahead. As a Web developer, you find yourself designing more complex and more demanding Web applications than your bubble 1.0 predecessors ever dared to dream of. This is the fun part. The less fun part is trying to test those feature-rich applications. The prospect of manual testing does not thrill any developer, and the multitude of browsers you need to test with makes it an even bigger nightmare. Figure 1, which is based on the browser market-share statistics provided by Net Applications at the time of this writing, illustrates quite clearly the state of the browser war 2.0.

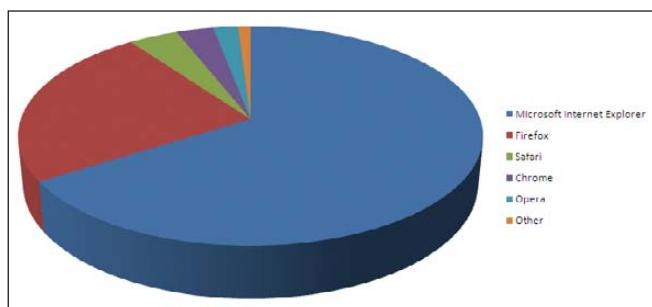


Figure 1. Browser Market Share

The breakdown is as follows:

- Microsoft Internet Explorer: 66%
- Firefox: 24%
- Safari: 4%
- Chrome: 3%
- Opera: 2%
- Other: 1%

Although estimates vary from different companies that monitor Web usage traffic, the conclusion is obvious: you no longer can afford to test with a single browser. And, note that Microsoft IE, which accounts for about 66% of the traffic, actually is three very different browsers: IE6, IE7 and IE8, which are known to render Web sites differently.

The situation is expected to get worse as mobile broadband Internet becomes cheaper and more common, adding more browsers you'll have to test with, sometimes with limited capabilities and nonstandard resolution.

The good news is that you are not facing this problem alone, and there are some advanced automatic Web application testing frameworks that can reduce the burden significantly.

Selenium to the Rescue

Selenium is much more than an average Web site unit-testing application. It is actually a set of tools, consisting of the following:

1. Selenium Core
2. Selenium Remote Control (RC)
3. Selenium Integrated Development Environment (IDE)
4. Selenium Grid

Selenium Core provides basic testing functionality. It is implemented in JavaScript and can be deployed either standalone (in which case it has to be installed on the Web server) or more commonly, as part of Selenium RC, IDE or Grid, which all use Selenium Core engine. Figure 2 shows the relationships between the Core and other Selenium projects.

Selenium RC is a Java-based command-line server that launches browsers and sends test commands to Selenium Core. You can write your tests, implemented as Selenium RC clients, in a variety of programming languages, such as Perl, Java, C# and others. If you are not afraid of some basic programming, this is the most powerful way to use Selenium.

Selenium IDE is a Firefox plugin that allows you to create tests using a graphical tool. These tests can be executed either

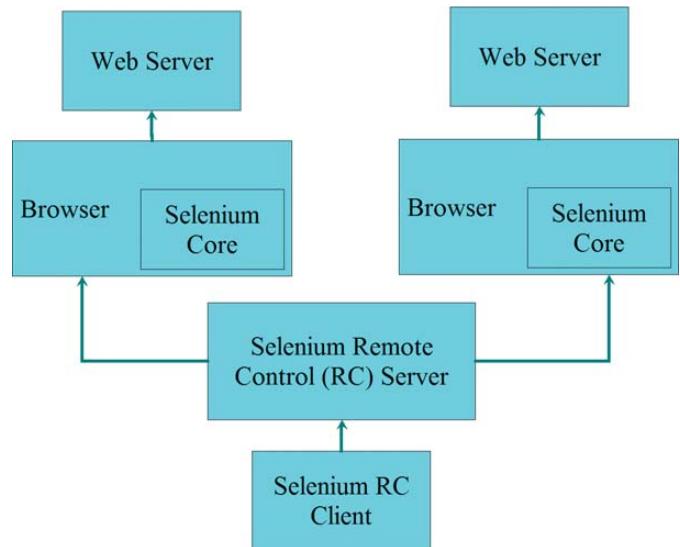


Figure 2. Selenium Architecture

from the IDE itself or exported in many programming languages and executed automatically as Selenium RC clients.

Selenium Grid solves the scalability problem by coordinating many Selenium RC instances, allowing you to run multiple tests in parallel on different machines.



101 Tech Tips DVD by *Linux Journal* is a series of one-minute how-to videos with shortcuts, tips and cool tricks for Linux users of all levels of expertise. Hosted by *Linux Journal* editor Shawn Powers, this DVD also features fun video extras, DRM-free copies of each tech tip for playing on your computer, as well as an entire year (2009 PDFs) of *Linux Journal* itself!

**Buy yours today
for just \$29.95.**

FEATURE Web Application Testing with Selenium

Selenium RC

I usually use Selenium RC, and I recommend it even to people with only a little bit of programming background. The API is simple, and the flexibility it provides is well worth the learning curve involved. RC consists of a server and client parts, which is a bit confusing, as the server has nothing to do with the Web server you are testing, but rather drives the Web browser used for testing. The server is a Java command-line application and should be executed as follows:

```
java -jar selenium-server.jar
```

The --help command-line switch will give you a full listing of supported options, but usually the defaults are sufficient. You also can invoke it programmatically from Java. The server will wait for client connections on port 4444 by default. The client can be written in one of the following languages: Perl, Python, PHP, Ruby, Java, C# and Erlang. The list is quite impressive. In fact, before I discovered Selenium, I was not aware that anybody outside Ericsson was using Erlang. The following sample Perl code shows a rather basic Selenium test that opens a Firefox browser and performs a search for the word Selenium using Google:

```
use strict;
use warnings;
use Test::WWW::Selenium;

my $sel = Test::WWW::Selenium->new(
    host          => "localhost",
    port          => 4444,
    browser       => "*firefox",
    browser_url   => "http://www.google.com",
    default_names => 1);

$sel->open("/");
$sel->type("q", "selenium");
$sel->click("btnG");
$sel->wait_for_page_to_load(60000);

print "$sel->get_title()\n";

$sel->stop();
```

The code above opens a new Selenium session, connects to the Selenium server running on the same machine as the client (the code in the listing) on port 4444, opens a Firefox browser, goes to the http://www.google.com URL, types the word selenium into the query field and clicks Google's "I'm Feeling Lucky" button. Both query and search elements are identified by their HTML element ID or name, q and btnG, respectively (examine the source code for Google's main page). The script waits for the page to load, prints the page title and shuts down the Selenium session.

Selenium Core

Selenium Core is the heart of Selenium. It is a collection of JavaScript functions that are used by all Selenium projects, but it is rarely used on its own. If you take a look at the HTML source for the Web site under test (using View→Page source in Firefox or with whichever browser you told Selenium to

execute), you will notice some additional JavaScript code that looks like the following:

```
<script
  type="text/javascript"
  src="/selenium-server/core/scripts/selenium-browserbot.js">
</script>
```

This is actually the Selenium Core code. The way Selenium "injects" itself into the Web page can vary (more on this later); however, when Selenium Core is used directly, this code has to be added to your HTML manually. Naturally, it requires access to the Web server that's serving the page, which is not always available. The biggest advantage of this mode is that it will work reliably with all browsers—something that other methods of using Selenium cannot guarantee.

Nevertheless, standalone Selenium Core is rarely used these days and may be deprecated in the future, so for the rest of this article, I concentrate on Selenium RC and IDE, which are the preferred methods of using Selenium.

Selenium IDE

Contrary to Selenium RC, for which you write your test cases using some programming language, the Selenium IDE allows you to develop your Selenium test cases in a graphical environment by simply interacting with the Web application under test as your users would. It probably is the easiest way to develop test cases. Selenium IDE is implemented as a Firefox plugin. The IDE not only records the Selenium commands that

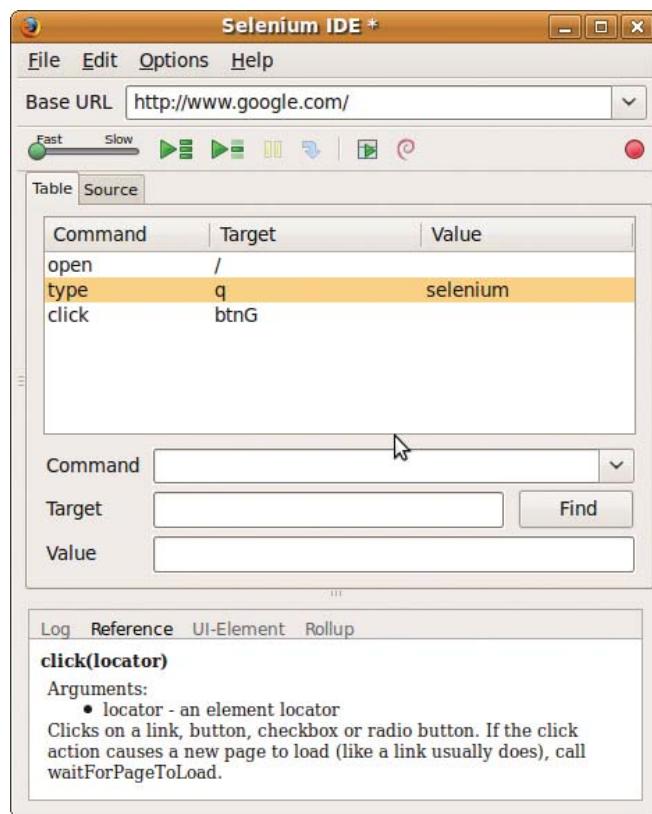


Figure 3. Selenium IDE Screen Capture

Advertiser Index

CHECK OUT OUR BUYER'S GUIDE ON-LINE.
Go to www.linuxjournal.com/buyersguide where you can learn more about our advertisers or link directly to their Web sites.

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

Advertiser	Page #	Advertiser	Page #
1&1 INTERNET, INC. www.oneandone.com	1	LINUX FEST NORTHWEST www.linuxfestnorthwest.org	71
7TH INTERNATIONAL PENGUIN CONFERENCE www.PenguinConference.org	78	LOGIC SUPPLY, INC. www.logicsupply.com	21
ABERDEEN, LLC www.aberdeencinc.com	3	LULLABOT www.lullabot.com	77
ASA COMPUTERS, INC. www.asacomputers.com	35	MICROWAY, INC. www.microway.com	C4, 7
CARINET www.cari.net	49	O'REILLY MySQL USERS CONFERENCE conferences.oreilly.com/mysql	27
COSSFEST www.cossfest.ca	73	PENGUICON www.penguicon.org	59
DIGI-KEY CORPORATION www.digi-key.com	79	POLYWELL COMPUTERS, INC. www.polywell.com	78, 79
EMAC, INC. www.emacinc.com	55	SERVERS DIRECT www.serversdirect.com	9
FLAGG MANAGEMENT www.flaggmgmt.com/linux	69	SILICON MECHANICS www.siliconmechanics.com	23, 25
FOURTH GENERATION SOFTWARE SOLUTIONS www.fourthgeneration.com	79	TECHNOLIC SYSTEMS www.embeddedx86.com	19
GECAD TECHNOLOGIES/AXIGEN www.axigen.com	78	TECHWEB LIVE EVENTS (BLACKHAT) www.blackhat.com	51
GENSTOR SYSTEMS, INC. www.genstor.com	61	TEK-X tek.phparch.com	75
GUTSY GEEKS www.gutsygeeks.com	76	TEXAS LINUX FEST www.texaslinuxfest.org	33
IBM www.ibm.com	5	TRUSTED COMPUTER SOLUTIONS www.TrustedCS.com/SecurityBlanket	C2, 79
IT360 www.it360.ca	43	USENIX ASSOCIATION www.usenix.com/events/lisa08	13
IXSYSTEMS, INC. www.ixsystems.com	C3	UTILIKILTS www.utilikilts.com	78

ATTENTION ADVERTISERS

July 2010 Issue #195 Deadlines

Space Close: April 26; Material Close: May 4

Theme: Enterprise

BONUS DISTRIBUTIONS: USENIX Annual Tech, O'Reilly's Velocity, Sun's JavaOne, Community One

Print: contact Joseph Krack, +1-713-344-1956 ext. 118, joseph@linuxjournal.com

On-line: contact Michael Beasley, +1-713-344-1956 ext. 119, michael@linuxjournal.com

correspond to your interactions with the browser, but it also allows you to modify their arguments and add more commands to the recorded test case. The plugin has a context menu that allows you to select a UI (User Interface) element from the browser's currently displayed page and then select a Selenium command from a drop-down list, automatically adding relevant parameters according to the context of the selected UI element.

After installation, the IDE is accessible via Tools→Selenium IDE in the Firefox menu. To illustrate the IDE functionality, let's re-create the above Google search test case using the IDE. In order to do this, you simply should run the IDE, check that the record button is on, open the <http://www.google.com> page, type selenium and click Google Search. As you type, the Selenium IDE captures what you do (Figure 3).

Selenium recorded the manually entered commands. This test case now can be played back using the IDE, saved as HTML or exported in a variety of formats, including ready-to-run code snippets for Selenium RC. The exported test case looks very similar to the handwritten code above, although in this case, I exported it in Python in order to demonstrate some of Selenium's capabilities:

```
from selenium import selenium
import unittest, time, re

class googletest(unittest.TestCase):
    def setUp(self):
        self.verifyErrors = []
        self.selenium = selenium(
            "localhost", 4444,
            "*chrome", "http://www.google.com/")
        self.selenium.start()

    def test_googletest(self):
        sel = self.selenium
        sel.open("/")
        sel.type("q", "selenium")
        sel.click("btnG")

    def tearDown(self):
        self.selenium.stop()
        self.assertEqual([], self.verifyErrors)

if __name__ == "__main__":
    unittest.main()
```

Note that in addition to using a unit-testing framework, the code above differs from the Perl example by using the "*chrome" browser string instead of "*firefox". This is one of the most confusing issues with Selenium, and it deserves a section of its own here.

Browser String Parameter

Before I go into the really confusing details of this parameter, it is important to understand that even though Selenium is not a new project, it still is under active development, and as it sometimes happens with open-source projects, some of the development effort that went into introducing new features might have been better spent debugging old ones and making

it more stable. As a result, in certain configurations, some Selenium features might not work as expected, and others may not work at all.

The browser string parameter not only specifies the browser Selenium will work with, but also the method Selenium uses to control the browser and the mode of communication between the Selenium server and the Selenium Core running inside the browser. And, yes, there are multiple modes for some browsers. Not all modes are implemented for every browser, and some Selenium commands will not work with certain modes. To add to the confusion, the default modes sometimes change between different Selenium versions. In Selenium version 1.0 "`*firefox`" is an alias for "`*chrome`" and "`*iexplore`" for "`*iehta`".

This explains why the automatically generated Python code produced a different value for the browser string than was used in the manually produced Perl code, even though both tests used Firefox. Both the "`*chrome`" and "`*iehta`" browser profiles implement a "native" approach in which Selenium

By design, the Proxy Injection mode works with all browsers, even those that were not tested with Selenium, as long as the browser supports JavaScript and an HTTP proxy.

uses a browser-specific method to "inject" the Selenium Core JavaScript code and to control the browser, as opposed to a Proxy Injection (PI) mode, which is generic and, at least in theory, should work with all browsers.

When Proxy Injection mode is enabled, Selenium RC, which has a built-in HTTP proxy server, configures the custom profile of the browser under test to work with its local proxy. Every HTTP response returned by the proxy server has had the Selenium Core JavaScript code "injected" into the `<head>` HTML element.

By design, the Proxy Injection mode works with all browsers, even those that were not tested with Selenium, as long as the browser supports JavaScript and an HTTP proxy. Not only that, but it also allows circumventing the "same origin policy"—that is, it allows you to test different sites on different domains during the same Selenium session with browsers for which the "native" mode is not complete, for instance Opera. At this point, you may think that this is the mode you should use for your tests, but unfortunately, during Selenium's development the developers discovered that some important functionality is very hard to implement in PI mode. As a result, most developers gradually switched to the so-called native mode, even though it requires a separate implementation for every browser. As a result, the PI mode is poorly maintained and quite buggy.

As you can see, even though it is often praised for its multiple browser testing capability, in reality, browsers other than Firefox and Internet Explorer are poorly supported. At the end of the day, if you have to ensure that your Web site looks and works correctly under all important browsers, your best bet may be a browser screenshot tool, such as BrowserShots or BrowserSeal. BrowserShots is a Web-based screenshot service supporting a wide range of browsers that unfortunately suffers from long response times. BrowserSeal, on the other hand,

is a standalone application running on your PC optimized for Web site screenshots. It produces screenshots in a few seconds, compared with a few minutes for BrowserShots; however, there is only a Windows version available at the time of this writing.

Selenium Grid

At first glance, it looks like Selenium RC supports enough parallelism that any additional distributed processing capability would not be needed. After all, a single Selenium RC server allows you to open a number of parallel sessions (that is, drive a number of browsers at the same time) and a single Selenium RC client. In addition to being able to work with multiple concurrent sessions of one server, it can communicate with multiple servers at the same time.

However, in practice, running more than six browsers on the same Selenium RC server is not advisable due to performance issues. Additionally, managing a large number of Selenium RC servers is a major headache and does not scale

very well. This is where Selenium Grid can help.

Selenium Grid introduces another component to the Selenium architecture—Selenium Hub, which manages a pool of available Selenium Remote Control entities and is responsible for the following:

- Transparently allocating a Selenium RC entity to a specific test.
- Limiting the number of concurrent test runs on each Remote Control.
- Shielding the tests from the actual grid infrastructure.

As far as your RC client programming is concerned, the move from Selenium RC to Grid requires minimal code changes. All you have to do is to change the infamous browser string parameter. For instance, change "`*firefox`" to something like "`Firefox on Windows`" or "`Safari on Mac`".

Selenium Hub's configuration is a bit more complex. First, you have to modify the `grid_configuration.yml` file. Let's say you want to use two RC instances—one with Firefox on Linux and another with Internet Explorer on Windows. In that case, your configuration file will look like this:

```
hub:  
  port: 4444  
  environments:  
    - name: "Firefox on Linux"  
      browser: "*firefox"  
    - name: "IE on Windows"  
      browser: "*iexplore"
```

After that, you should use ant to launch the Selenium Hub by running `ant launch-hub` on the hub machine. The RC instances are created by running the following commands, one on a Linux machine and one on a Windows machine.

On the Linux machine:

```
ant -Denvironment="Firefox on Linux" \
-DhubURL=http://<hub-IP-address>:4444 \
launch-remote-control
```

On the Windows machine:

```
ant -Denvironment="IE on Windows" \
-DhubURL=http://<hub-IP-address>:4444 \
launch-remote-control
```

After that, you can code your RC client to use any of the above RC server instances via the hub.

Selenium API

Comprehensive description of the Selenium API is beyond the scope of this article, but the list below demonstrates what the framework is capable of:

- `$sel->click($locator)` — Clicks on a link, button, check box or radio button.
- `$sel->context_menu($locator)` — Simulates opening the context menu for the specified element (as might happen if a user right-clicks on the element).
- `$sel->focus($locator)` — Moves the focus to the specified element.
- `$sel->key_press($locator, $key_sequence)` — Simulates a user pressing and releasing a key.
- `$sel->mouse_over($locator)` — Simulates a user hovering the mouse over the specified element.
- `$sel->type($locator, $value)` — Sets the value of an input field, as though you typed it in.
- `$sel->check($locator)` — Checks a toggle button (check box/radio).
- `$sel->select($select_locator, $option_locator)` — Selects an option from a drop-down menu using an option locator.
- `$sel->submit($form_locator)` — Submits the specified form.
- `$sel->open($url)` — Opens a URL in the test frame.
- `$sel->open_window($url, $window_id)` — Opens a pop-up window.
- `$sel->go_back()` — Simulates a user clicking the back button in the browser.

- `$sel->get_location()` — Gets the absolute URL of the current page.
- `$sel->get_body_text()` — Gets the entire text of the page.
- `$sel->get_text($locator)` — Gets the text of an element.
- `$sel->get_selected_indexes($select_locator)` — Gets all option indexes for the selected options in the specified select or multi-select element.
- `$sel->get_all_links()` — Returns the IDs of all links on the page.
- `$sel->wait_for_condition($script, $timeout)` — Runs the specified JavaScript snippet repeatedly until it evaluates to "true".
- `$sel->get_cookie()` — Returns all cookies for the current page under test.
- `$sel->wait_for_text_present($text, $timeout)` — Waits until \$text is present in the HTML source.

For more details, check the Perl API link at the end of the article. API documentation for other languages is available as well. If you find that the API is lacking somewhere, you always can extend it by executing your own JavaScript functions using `$sel->get_eval($script)`.

Summary

As you've seen, Selenium is a powerful Web application testing tool that supports many different languages and has a number of different frameworks built around the Selenium Core. It's an open-source project with an active community, which, at the time of this writing, is working full steam on version 2.0, which will include many new exciting features.■

Alexander (Sasha) Sirotnik has more than ten years' experience in software, operating systems and networking. He currently works on the LTE (Long Term Evolution) Project at Comsys Mobile and lives with his wife and kid in Tel-Aviv, Israel. Alexander can be reached via e-mail at [sasha.sirotnik \[AT\] gmail.com](mailto:sasha.sirotnik@comsysmobile.com).

Resources

Selenium Web Site: seleniumhq.org

Selenium Perl API Documentation:
release.seleniumhq.org/selenium-remote-control/1.0-beta-2/doc/perl/WWW-Selenium.html

BrowserShots Cross-Browser Screenshot Service:
browsershots.org

BrowserSeal Fast Cross-Browser Screenshot Application:
www.browserseal.com

jsormdb

an Embedded JavaScript Database

With the rise of Web 2.0 and Rich Internet Applications, the need for a solid JavaScript-based data management paradigm has become much more acute. Using jsormdb, you can develop your applications with the same data-driven paradigm you use on server-side applications.

AVI DEITCHER

The (almost) standardization and acceptance of JavaScript as a browser-side development language has enabled the rapid growth of dynamic Web applications. These applications often look and feel as snappy as native desktop applications. Add the excellent open-source client-side development frameworks, like jQuery and ExtJS, along with the communicative power of AJAX, and you have an incredibly powerful and largely open browser-side development platform.

It has gotten so powerful, in fact, that applications are living entirely within the browser, with the server acting as a simple file store. One excellent example is password managers, like Clipperz and jkPassword. These work entirely in the browser to provide the user interface and encryption. The only services provided by the server are serving the static HTML and JavaScript files, and persisting the already-browser-side-encrypted file.

As the client side grows, and self-contained applications like these password managers increase, the need for basic services that are normally available in a development environment increase. In a normal development environment, the ability to manage data reliably is one of the first and key issues tackled. Sometimes, it leads to full-blown client-server database systems, like Oracle or MySQL. In other instances, it leads to embedded databases—for example, Apache Derby, which are critical for managing data within a single instance of an application.

Unfortunately, our browser development platform has lacked any such data management system. As it turns out, the availability of such a system is even more critical in the browser environment than in a server environment. For complete browser applications, the entire data set and transaction semantics are local and cannot depend on a server. For more traditional server-driven applications, the need is even more acute. Whereas a server can rely on redundancy, high bandwidth and low latency to its data store, the browser has none of these attributes. It becomes critically important for the browser application developer to be able to perform the majority of the application's data activities locally, sending the result to the server with the lowest frequency and bandwidth possible.

Enter a new class of data stores, embedded JavaScript databases. This article introduces jsormdb, an advanced JavaScript database engine. jsormdb includes most of the features you, as a developer, would expect in an embedded database system, as well as many that make sense only within the context of a browser environment. I describe jsormdb and its basic usage, including creating a database, loading data, modifying and querying the data, and transactions. jsormdb also supports event signaling and, critical for a browser-side data store, persisting changes or the entire data set to the server, but those are the subject of another article.



2010 HIGH PERFORMANCE COMPUTING LINUX FINANCIAL MARKETS Show and Conference

APRIL 19, 2010 (MONDAY)

www.flaggmgmt.com/linux

ROOSEVELT HOTEL, NYC

Madison Ave and 45th St, next to Grand Central Station

Show is free. Register today.
\$295. Save \$100. The
Conference
Low cost

The need for speed.

See new Wall Street speakers talking about speed, low latency, cost-saving strategies.

Register today for the full conference and save \$100.

Wall Street and the financial markets are looking for ways for traders to speed their orders to market.

This IT priority combines with lowering costs, making users more productive, streamlining business processes.

High Frequency Trading, High Performance Computing, and increased volume of transactions are creating pressure for IT departments to seek out technology and lower transaction costs.

2010 IT budgets are projected to increase by 6%. This is an incentive for major firms to see, examine and compare systems at our 2010 Annual High Performance Computing Linux Showcase.

The Show will feature High Performance Computing, High Frequency Trading, Linux and Open Source, Virtualization, Cloud Computing, Grid, Blade and Cluster computing and other cost saving technologies.

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

IBM, Oracle, Intel, Cisco, Blade, ScaleMP, among other leading companies will be showing their newest products on-the-show floor live.

Register in advance for the full conference program which includes general sessions, drill down sessions, an industry luncheon, coffee breaks, exclusive viewing times in the exhibits, and more. Save \$100. \$295 in advance. \$395 on site.

Don't have time for the full Conference? Attend the free Show from 8 am-4 pm in the convenient Roosevelt Hotel at Grand Central Station. Plan to attend the free show, but you must register in advance at: <http://www.flaggmgmt.com/linux>

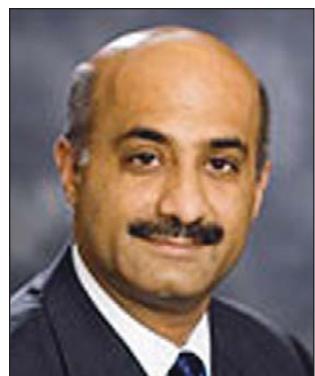
Featured Speakers Announced



Andy Bechtolsheim
Founder, Chief Development Officer and Chairman, Arista Networks, Menlo Park, CA



Stanley Young
CEO of NYSE Technologies & co-CIO of NYSE Euronext, New York, NY



Vikram Mehta
CEO, BLADE Network Technologies, Santa Clara, CA



2010 Gold Sponsors



ORACLE

BLADE



2010 Silver Sponsor

ScaleMP™

NETWORK TECHNOLOGIES

LINUX JOURNAL

HPC



Inside Market Data

LINUX MAGAZINE

waters

SECURITIES INDUSTRY

DWT

WINDOWS IN FINANCIAL SERVICES

WSTA

Show & Conference:
Flagg Management Inc
353 Lexington Ave, NY10016
(212) 286 0333 flaggmgmt@msn.com

Show Hours: Mon, Apr 19 8 - 4:00
Conference Hours: 8:30 - 4:50

Visit: www.flaggmgmt.com/linux

FEATURE jsormdb: an Embedded JavaScript Database

Why a Database?

As anyone who has done even a moderate amount of development knows, data storage structures are, in many ways, the fundamental building block of any application. There are several good reasons for this:

1. Performance: choice of data structure can have a significant impact on application performance. For example, using a linked list versus a serial array is slower to access elements in the middle of the list, but much faster at list reorder and element insertion/addition/removal.
2. Usability: a standard data structure makes it much easier for someone else (or you) to understand what your application does and how.
3. Separation of concerns: by separating data structure and storage from business logic, you are able to work with each separately.

A good database structure addresses those concerns, while providing the following features:

1. Queries: you will want to query your data to discover which elements of the data—for example the first, 35th and 60th records—match certain arbitrary criteria. Of course, big RDBMS systems excel at this, and even have their own language for doing so, SQL. Of course you’re not about to embed Oracle or MySQL type systems into a JavaScript environment, just to support local temporary queries.
2. Indexing: if you go to an arbitrary array or table of data and ask it to return all the records where the third field is equal to 2, you (or your data store engine) will need to go to each and every record and check whether the third field is equal to 2. This is called a full table scan, and it is terribly inefficient. For fields that are checked frequently, you would much prefer a more efficient method, one that does not require checking every single record linearly. This is known as indexing.
3. Transactions: in a simple world, events are single-stage and one-way. For example, login is a simple process: you enter your credentials and are either in or out. In the real world, however, events usually are multistage. For example, in order to transfer \$100 from your checking account to your savings account, you need to deduct \$100 from your checking account and add \$100 to your savings account. Ideally, both steps will succeed. If both fail, it is not great, but you can live with it. However, if only one of the two succeeds, either you (if it is the deduction) or your bank (if it is the addition) will be very unhappy.
4. Events: sometimes, you want the database to tell you whether something of significance has occurred. For example, you may want to know that a certain table has been updated or perhaps that an account balance has dropped below a certain threshold. Many databases support events, often known as triggers, that cause reactions.

Because I am discussing the browser environment, one additional feature is important: persistence. A browser environment is a transient one; the moment users close their windows,

all of the data is lost. Because the browser application relies on the server to provide permanence, you want a way to handle both loading your local data store from, as well as persisting our local changes back to, that server seamlessly.

jsormdb provides a solution to all of these problems. When configured correctly, the presentation and business logic within your application can treat jsormdb as the *entire* data store, leaving jsormdb to handle persistence to/from the server. Without jsormdb, your application looks like Figure 1.

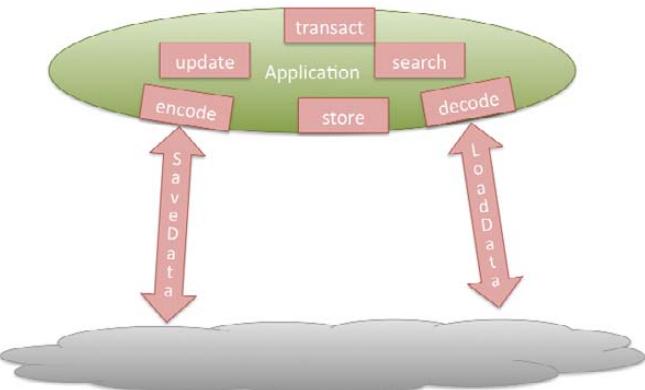


Figure 1. Web Application without a Database

With jsormdb, you can work with a much cleaner design (Figure 2).

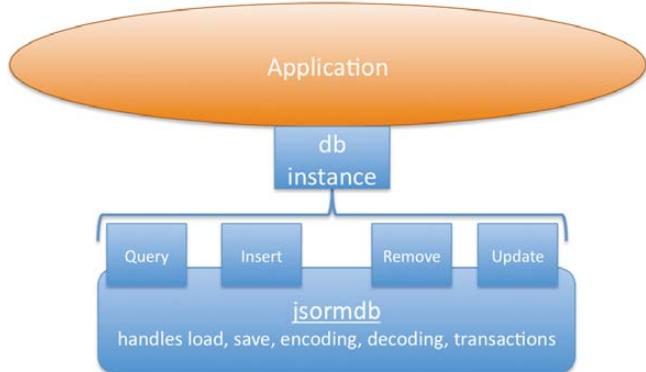


Figure 2. Web Application with a Database

How It Works

The jsormdb library introduces several concepts and implementations that are necessary for creating and using a database. Note that all of these are detailed in the JavaScriptDoc distributed with jsormdb, in the doc folder.

1. The database: this class is JSORM.db.db(), and it represents a single database instance.
2. The parser: responsible for taking input data and transforming it into records that are stored in an instance of JSORM.db.db or taking entries in a jsormdb database and transforming them into an acceptable output format. For example, the data you load into the library may be in JSON, XML, JavaScript objects or even Pig Latin. jsormdb does not care, provided you provide a parser that can convert



LinuxFest Northwest

**April 24-25 2010
Bellingham, WA**

Grassroots linux gathering 

Exhibits of all flavors 

Presentations of all levels 

Prizes, after parties 

FREE admission & parking 

FREE open source software 

Bring the whole family! 

Hosted By



linuxfestnorthwest.org

FEATURE jsormdb: an Embedded JavaScript Database

between your preferred encoding and an array of native JavaScript objects. jsormdb comes with a parser for JSON and Objects; an XML parser is being developed.

3. The channel: responsible for loading data from or saving data to a remote data store. An HTTP channel can use AJAX to retrieve data from and post data to the server from which the Web page itself was loaded. You also can use another jsormdb as a source for your database, but a channel for that is unnecessary.

Putting it all together, you can create a database instance JSORM.db.db that retrieves data via HTTP from your server using JSORM.db.channel.http and parses it in JSON format using JSORM.db.parser.json. Figure 3 shows how all the pieces work together.

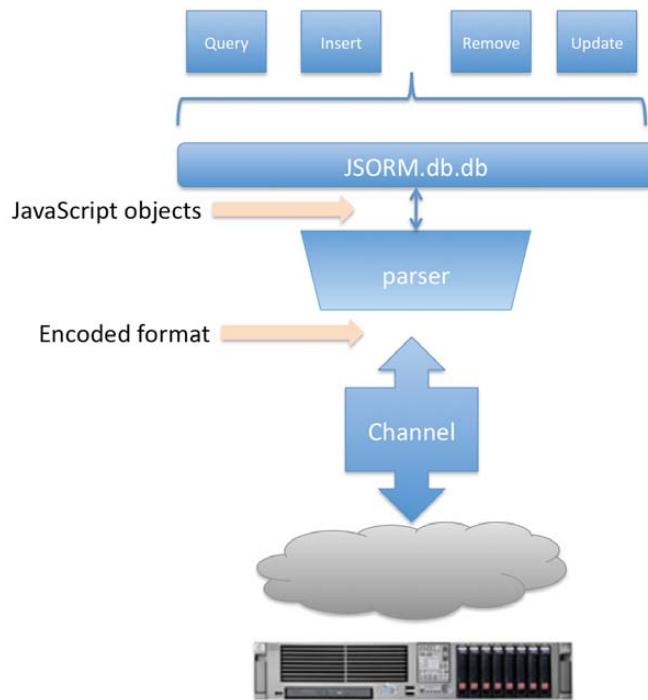


Figure 3. Fetching and Parsing Data from the Server

It is important to note that all classes follow Douglas Crockford's principles and are instantiated directly, without the "new" keyword:

```
var parser = JSORM.db.parser.json(); // right
var parser = new JSORM.db.parser.json(); // WRONG!
```

As much as possible, jsormdb semantics follow those of classical SQL. Thus, adding records is an insert; modifying records is an update, and so forth.

Installation

Installing jsormdb involves a few simple steps.

- 1) Download and unzip the library from the jsorm site, where it is available as a zip file from the Download link.
- 2) Install the library. The download includes two versions of the library. jsormdb.js is minified at just under 25KB. jsormdb-src.js

is not minified, is used primarily for debugging and is 77KB. You can reduce their sizes further with gzip. You need to install the library you want to use in a path accessible to your browsers. For the purposes of this example, install file jsormdb.js in the same directory as your Web page.

3) Include the library in your Web page. Normally this is done in the header as follows:

```
<script type="text/javascript" src="jsormdb.js"></script>
```

Creation

Now that you have downloaded and installed the library, as well as included it in your page, you are ready to create a database.

In the simplest form, you create a database by simply instantiating it:

```
var db = JSORM.db.db();
```

Although this creates a database, you may want to add some initial configuration parameters. For example, you may want to indicate the parser and/or the channel to use, or even to load some data directly.

Loading Data

jsormdb supports loading data in two ways: directly, using raw data, and remotely, via a channel:

```
var conf, db;
// to use a channel and parser
conf = {
  channel: JSORM.db.channel.http({updateUrl: "/send/text.php",
                                    loadUrl:   "/receive/text.json"}),
  parser:  JSORM.db.parser.json()
}
db = JSORM.db.db(conf);

// to load data directly
conf = {data: [{name: "Joe",   age: 25},
               {name: "Jill",   age: 30},
               {name: "James",  age: 35}]}
db = JSORM.db.db(conf);
```

JSORM.db.db has many options for instantiation. See the API docs or the Wiki entry, both of which are listed in the Resources for this article.

Whichever manner you choose to load data, jsormdb expects the data passed to it to be an array of simple JavaScript object literals. jsormdb is not rigid about the data structure, and it does not care what fields exist on each object. Both of the following are equally valid:

```
data = [ {name: "Joe",   age: 25},
         {name: "Jill",   age: 30},
         {name: "James",  age: 35}];
```

```
data = [ {name:      "Joe",   age: 25},
         {firstName: "Jill"}, 
         {surname:   "James", city: "London"}];
```

An important note about the records is that each can have

a type. When a record has a type, it is specially marked and can be searched more easily along with others of the same type. This can improve search times greatly, akin to putting records in different tables in an RDBMS:

```
data = [
  {name: "Joe", age: 25, type: "person"},  
  {name: "Jill", age: 30, type: "person"},  
  {name: "James", age: 35, type: "person"},  
  {name: "Fiat", color: "yellow", type: "car"},  
  {name: "Ferrari", color: "red", type: "car"},  
  {name: "GM", color: "white", type: "car",  
    status: "bankrupt"}  
];
```

Querying Data

In order to make the database useful, you need to be able to retrieve records you inserted or loaded—that is, you need to query the database. To do this, you simply call db.find(query). The results will be an array of JavaScript objects that match your query. If no records match, an empty array is returned. If the query itself is invalid, a null object is returned.

The query parameter itself is an object with two fields: “where” and “fields”. The where field informs the database what you need to match in order to retrieve the record. It can be a simple match, or it can be a compound match, joining

multiple simple or compound matches into a single larger one. The fields field can be used to restrict which fields are returned from the records that are found:

```
var where, results;  
  
// simple, retrieves all records where the name field equals "John"  
where = {field: "name", compares: "equals", value: "John"};  
results = db.find({where: where});  
  
// compound, retrieves all records where name equals "John"  
// or name equals "Jack"  
where = {join: 'or',  
  terms: [  
    {field: "name", compares: "equals", value: "John"},  
    {field: "name", compares: "equals", value: "Jack"}]  
};  
results = db.find({where: where});
```

Compound terms can be joined by ‘and’ or ‘or’. Simple terms can match on any field and can compare the field using one of many conditions, such as “equals”, “in”, “starts”, “gt” and so on. The API docs and Wiki entry, listed in the Resources for this article, have a complete list.

Finally, you can restrict the search, at any level, by the type of record to retrieve. The type field, if available, is always indexed, leading to much faster searches:

Western Canada's Premier Open Source Festival

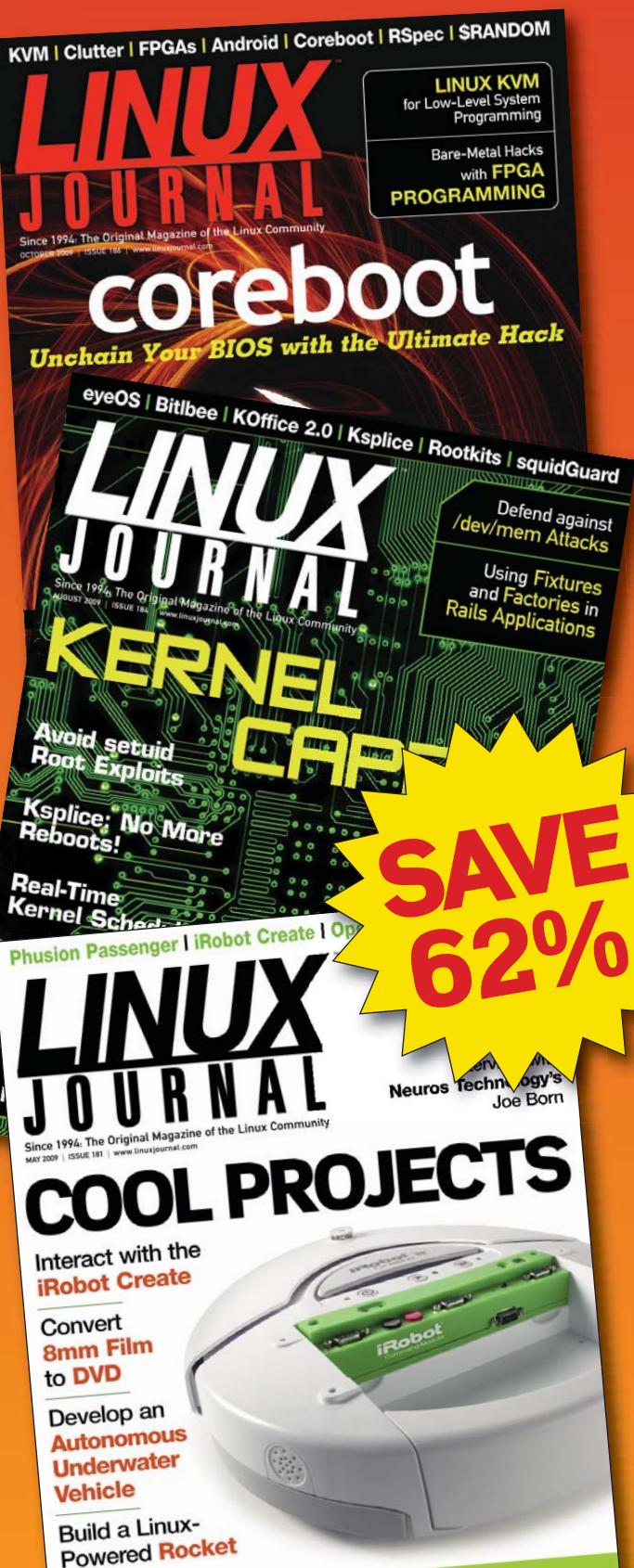


COSSFEST.CA
CALGARY OPEN SOURCE
SYSTEMS FESTIVAL

> Business Applications, Security
> System Administration, Programming
> Workshops, Contests & More

> April 9th and 10th 2010 - Coast Plaza Hotel - Calgary, Alberta !
> Get all the details -> <http://www.cossfest.ca>

If You Use Linux, You Should Be Reading **LINUX JOURNAL**™



- » In-depth information providing a full 360-degree look at featured topics relating to Linux
- » Tools, tips and tricks you will use today as well as relevant information for the future
- » Advice and inspiration for getting the most out of your Linux system
- » Instructional how-tos will save you time and money

Get *Linux Journal* delivered to your door monthly for 1 year for only \$29.50! Plus, you will receive a free gift with your subscription.

SUBSCRIBE NOW AT:
WWW.LINUXJOURNAL.COM/SUBSCRIBE

Offer valid in US only. Newsstand price per issue is \$5.99 USD; Canada/Mexico annual price is \$39.50 USD; International annual price is \$69.50. Free gift valued at \$5.99. Prepaid in US funds. First issue will arrive in 4-6 weeks. Sign up for, renew, or manage your subscription on-line, www.linuxjournal.com/subscribe.

```
// all records of type "car" where age >= 12
where = {field: "age", compares: "ge", value: 12, type: "car"};
results = db.find({where: where});
```

It is important to note that the results of a db.find(query) will return a copy of the records, not the originals themselves. Thus, it is safe to modify the returned results at will.

Modifying Data

You can modify data in one of several ways: remove records, add records or change records.

Adding records is fairly straightforward. Simply call db.insert(data), with the data an array of JavaScript object literals:

```
data = [{name: "Jack", age: 80},
        {name: "Sam", age: 22},
        {city: "Paris", type: "location"}]
db.insert(data);
```

Where these records actually will physically be inserted into the jsormdb database is irrelevant, just as it is in any true database. All that matters is that the records are inserted, and that you can retrieve them.

To remove records, just call db.remove(query). The query parameter is exactly the same as in db.find(). All records that match the where will be removed immediately.

To change records, just call db.update(data,query). The query parameter is exactly the same as in db.find(). The data parameter is a single JavaScript object that has the fields to update. All records whose fields match the where will be updated:

```
// for every record where the age >= 40, change the age to be 35
var update, where;
where = {field: "age", compares: "ge", age: 40};
update = {age: 35};
db.update(update,where);
```

Transactions

As noted earlier, transactions are crucial to anything other than trivial events. jsormdb provides advanced transaction processing to allow you to manage your changes properly.

Transactions are always enabled. From the moment you load a database, a new transaction is started. All the changes you make—update, remove, insert—are tracked by jsormdb. When you have reached the end of your transaction, you must either commit the changes or reject them.

If you commit the changes, all of the change tracking is thrown away, and a new transaction is started. From that point forward, you cannot undo any of your previous changes. On the other hand, if you reject the changes, all of the changes from the beginning of the transaction—either the last load or the last commit or reject—are undone. Additionally, if you want, you



OVER 50 High caliber **talks & tutorials**.

[HTTP://TEK.PHPARCH.COM](http://TEK.PHPARCH.COM)

Conference tickets are available in limited quantity. Get yours today.

FEATURE jsormdb: an Embedded JavaScript Database

can reject only *some* of your changes. For example, if you have made eight changes in this transaction, and you want to undo only the last four, you can do so. This is particularly useful in the user-interface environment. For example, if you have written a Web 2.0 spreadsheet application with jsormdb as your data store, you probably want to give users the ability to undo each of their changes, one by one, in reverse order, probably using Ctrl-Z on Windows and Linux or Cmd-Z on Mac. Until jsormdb, you would have to code the tracking of these changes manually. Now, you can simply delegate this function to jsormdb. Each time users click Undo, they reject exactly one change.

The following example starts with three records, adds two more, modifies one and removes one:

```
var data, where, db, recs;

// create and load the database
data = [{name: "Joe", age: 25},
        {name: "Jill", age: 30},
        {name: "James", age: 35}];
db = JSORM.db.db({data: data});

// add records
db.insert([{name: "Karl", age: 40}, {name: "Karyn"}]);

// modify Joe
```



The first and only radio show broadcast in the USA dedicated exclusively to spreading the word about the LINUX OPERATING SYSTEM and FOSS.

GUTSY GEEKS
COMPUTER SHOW

gutsygeeks.com

```
db.update({data: {age: 26},
           where:
             {field: "name", compares: "equals", value: "Joe"}});

// remove James
db.remove({where:
            {field: "name", compares: "equals", value: "James"}});

// get all of the data
recs = db.find();
// recs = [{name: "Joe", age: 26},
//          {name: "Jill", age: 30},
//          {name: "Karl", age: 40},
//          {name: "Karyn"}]

// we can commit, reject or partially reject
db.commit(); // all changes are saved and a new transaction starts
// OR
db.reject(); // all changes are rolled back;
              // db.find() returns [{name: "Joe", age: 25},
              //                    {name: "Jill", age: 30},
              //                    {name: "James", age: 35}]
// OR
db.reject(1); // just the removal of James is rolled back
```

Last but not least, commit() can cause jsormdb to update the server with its new data in one of several formats, and it even can update itself based on the server's response. This persistence of changes, which uses jsormdb to mediate between browser-side business logic and presentation on the one hand and server-side storage on the other, is the subject of another article.

Summary

In summary, jsormdb provides Web browser application developers the ability to isolate data management from business and presentation logic cleanly, to utilize full transaction semantics easily and to query, update and modify data, including indexing, simply and efficiently. ■

Avi Deitcher is an operations and technology consultant based in New York who has been involved in technology since the days of the Z80 and Apple II. He has a BS in Electrical Engineering from Columbia University and an MBA from Duke University, and can be reached at avi@atomicinc.com.

Resources

jsorm Site: jsorm.com

jsorm Wiki: jsorm.com/wiki

jsorm API Docs: jsorm.com/doc/api

jsormdb Samples: jsorm.com/doc/samples/jsormdb.html

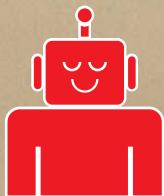
Douglas Crockford's JavaScript Site: www.crockford.com

MySQL: www.mysql.org

Oracle: www.oracle.com

Lullabot-Powered

The most super powered sites in the world are created in Drupal, by you and Lullabot.



Lullabot™ New Lullabot Learning Series training DVDs at Lullabot.com

Suzi Arnold
Director of New Media
Sony Music

axigen
MAIL SERVER

The Mail Server
for IT Professionals

www.axigen.com

American made Utility Kilts for Everyday Wear

UTILIKILTS.com

VII 7th International
Penguin Conference

August 30 - September 3, 2010
Boston, MA

Sponsorship opportunities now available.
www.PenguinConference.org

Polywell Storage Servers

More Choices, Excellent Service, Great Prices!

9015N

Quiet Storage NAS/SAN/iSCSI

8TB \$1,999	- Dual Gigabit LAN
12TB \$2,599	- RAID-5, 6, 0, 1, 10
30TB \$6,599	- Hot Swap, Hot Spare
	- Linux, Windows, Mac
	- E-mail Notification
	- Tower or Rackmount

5048A

4U24A

2012A

LD-001

5U-48Bay 96TB Storage Server

4U-24Bay 48TB
RAID-6, NAS/iSCSI/SAN Storage
Mix SAS / SATA, 4 Giga / 10Gbit LAN

2U-12Bay 24TB
RAID-6, NAS/iSCSI/SAN Storage
Mix SAS / SATA, 4 GigaLAN

1U945GCL2

Mini-1U Server \$499
Intel Dual-Core Processor, 2 x 500G RAID
Dual GigaLAN, 4GB DDR2 RAM

Silent Eco Green PC
The Best Terminal PC
Intel® / AMD® x86 Processor
Energy efficient, Quiet and Low
Voltage Platform. starts at \$199

Polywell OEM Services, Your Virtual Manufacturer

Prototype Development with Linux/FreeBSD Support
Small Scale to Mass Production Manufacturing
Fulfillment, Shipping and RMA Repairs

■ 20 Years of Customer Satisfaction
■ 5-Year Warranty, Industry's Longest
■ First Class Customer Service

888.765.9686
linuxsales@polywell.com
www.polywell.com/us/Lx

POLYWELL

INNOVATION ON THE GO
ORDER YOUR BEAGLE BOARD FROM DIGIKEY.COM



AVAILABLE EXCLUSIVELY AT DIGI-KEY
beagleboard

only \$149⁰⁰

**LOW-COST, NO FAN,
SINGLE-BOARD
COMPUTER**

**Digi-Key[®]
CORPORATION**

www.digikey.com

**ERP/Accounting
Distribution/Mfg
Software for Linux**
Source Code, Development Tools

>> Free Eval License <<

Fitrix
Affordable, Adaptable ERP Software

fitrix.com/linux2
800.374.6157

Automated OS Lock Down for Linux and Solaris

Are you using scripts to lock down your operating systems? Security Blanket automatically locks down your OS to meet industry (DISA STIGS, CIS, SANS, etc.) or customized standards.



For a Free Trial of
Security Blanket visit
[http://www.trustedcs.com/
SecurityBlanket/SecurityBlanket-Try-Out.html](http://www.trustedcs.com/SecurityBlanket/SecurityBlanket-Try-Out.html)

Polywell Mini-PCs



ITX with VESA / Wallmount



ITX-50C



ITX-10A 1.4" ThinPC



ITX-20A with SlimDVD



Fanless Slim PC with Intel[®] 1.6GHz 1W Mobile Atom[™] Processor, DC12V Power-in, 1GB DDR2 RAM starts at **\$199**
Supports SATA Hard Drive or Solid State Drive (SSD). Optional PCI RISER Slot for TV Tuner or other Add-on Device on ITX-30A



ITX-1000C with 4LAN and WiFi Option

- 23 Years of Customer Satisfaction

Polywell Computers, Inc

Over 250 Mini-ITX Models Available:

- NVIDIA[®] GeForce 8200/8100 with AMD[®] Athlon/Phenom Processor
- NVIDIA[®] GeForce 9300/7100/7050 with Intel[®] Core 2 Duo Processor
- PCI, PCIe, MiniPCIe Slot for TV Tuner or Industrial Add-on
- Custom Design Chassis for Small to Mid Size OEM Project

- 5-Year Warranty, Industry's Longest

1461 San Mateo Ave, South San Francisco, CA 94080 650.583.7222 Fax: 650.583.1974

888.765.9686

linuxsales@polywell.com

polywell.com/us/Lx



Is Android the “Top” Linux?

The Linux-based OS is in a good position to win a much bigger category than desktops. DOC SEARLS



Linux on the desktop has been around for a long time. So were jeeps, before large numbers of them started showing up in suburban driveways, in the form of SUVs. At the time of this writing (early January 2010), desktop Linux is still a jeep at the pre-SUV stage. It also has been a decade (or more) since many of us (myself included) began predicting that personal Linux computer sales would hockey-stick in the next year. Alas, the market share estimates are still low, even if they’re not flat: 2.14% (from W3Counter), 0.67%, (from StatCounter) and 1.02% (from NetApplications).

As always, there are encouraging signs. CES this year featured lots of Linux-based devices. Lenovo showed the IdeaPad U1 (an odd but inventive Linux-Windows hybrid) and the Skylight, which is due to be released in April 2010. HP floated the Mini 5102, offering Linux as one of several operating system options. These are all in the Netbook class, which has a lot in common with MP3 players before the iPod came along, and smartphones before iPhones came along—meaning that Netbooks are ripe for an Apple offering that changes the whole game.

The rumor is that Apple will come out with a “slate” or a “tablet” or something else that will be old news by the time you read this. Apple’s offering, however, matters less than its strategy, which Keith Hopper (keithhopper.com), a technologist with NPR, explains this way:

Apple’s genius is finding an immature market where progress is logjammed, and building a whole new vertical silo around a piece of gotta-have-it hardware. They did this with the iPod in the MP3 player market, and they did it again with the iPhone in the smartphone market. Apple was brilliant to recognize that the MP3 player and mobile markets were actually immature and suffered from compatibility, usability, and performance issues that could be addressed by slick and robust vertical integration. Never mind that this was bad news

for many customers, who in essence get locked-in by the integration.

But, while Apple goes vertical, Google goes horizontal. For example, Google created Wave to break the logjam in several categories at once, including instant messaging and group editing. Google created Chrome and Gears to provide rich client-side support for Web-based apps. All are open-source platform planks, both for Google and for people who want to use them (and improve them). They go horizontal so everybody (including Google) can build all kinds of vertical stuff on them.

Those platform planks are the Google Chrome OS, about which the company blogged this, last July:

Google Chrome OS is an open-source, lightweight operating system that will initially be targeted at Netbooks. Later this year we will open-source its code, and Netbooks running Google Chrome OS will be available for consumers in the second half of 2010....

Google Chrome OS will run on both x86 as well as ARM chips, and we are working with multiple OEMs to bring a number of Netbooks to market next year. The software architecture is simple—Google Chrome running within a new windowing system on top of a Linux kernel. For application developers, the Web is the platform....And of course, these apps will run not only on Google Chrome OS, but on any standards-based browser on Windows, Mac and Linux, thereby giving developers the largest user base of any platform.

They added:

Google Chrome OS is a new project, separate from Android. Android was designed from the beginning to

work across a variety of devices from phones to set-top boxes to Netbooks. Google Chrome OS is being created for people who spend most of their time on the Web....

But, while Chrome OS has to compete with Windows, Mac OS X and a zillion Linux distros, Android has an easier job—and potentially a much larger marketplace. Although Symbian, iPhone, RIM and Windows Mobile all have much bigger market shares than Android, they still are mostly held captive to partnerships between phone makers and carriers. Android was too, with early versions, but that changed with Nexus One, which launched in January 2010. Nexus One is the closest thing yet to a white-box phone design—one anybody can use, with any carrier, and improve at the base code level as well. Think of it as the smartphone equivalent of an SUV that anybody can make.

Application portfolios (among which Google’s are becoming essential) will help sell units, but Nexus One’s independence from makers and carriers is what will expand the market’s envelope. The Nexus One is not made just for Verizon or Vodafone. It’s horizontal, and it’s open. You can add value to it with more than just apps.

The Nexus One isn’t quite as slick as the iPhone, but its evolutionary path (and therefore that of Android) has none of the iPhone’s proprietary roadblocks and speed bumps. While Google has surely annoyed “partners”, such as Motorola (whose Android-based DROID was upstaged by the Nexus One), it also has opened a path toward a liberated smartphone future. As the Nexus One improves (to the Nexus Two and Three?), it’s on track to score big in a place that’s even more personal than desks and laps: the hands, pockets and purses of phone users everywhere.■

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.

Orion II iX-N4236

Powerful 4U Orion II Storage Series

- ✓ Outstanding performance
- ✓ Excellent cooling efficiency
- ✓ Up to 72 TB in 4U, unparalleled storage density
- ✓ Up to 432TB in 20U of rack space utilizing optional Orion II JBOD Expansion Units

To order today call: **1-800-820-BSDI**



Notable features include:

- Dual Intel® 64-Bit Socket 1366 Quad-Core or Dual-Core, Intel® Xeon® Processor 5500 Series
- 4U Storage Server Chassis with up to 72 TB storage capacity
- 36 x 3.5 Hot-Swap SAS/SATA HDDs (24 front side + 12 rear side)
- 1400 W (1+1) Redundant High Efficiency Power Supply (Gold level 93%+ power efficiency)

- Dual Intel® 5520 chipsets with QuickPath Interconnect (QPI) up to 6.4 GT/s
- Up to 144GB DDR3 1333/1066/800 MHz ECC Registered DIMM/24 GB Unbuffered DIMM
- 2 (x16) PCI-E 2.0, 4 (x8) PCI-E 2.0 (1 in x16 slot), 1 (x4) PCI-E (in x8 slot)
- Intel® 82576 Dual-port Gigabit Ethernet Controller

iXsystems Introduces the Orion II 4U Storage Solution

The iX-N4236 boasts energy efficient technology and maximum, high density storage capacity, creating a 4U powerhouse with superior cooling.

The Orion II has **thirty-six hot-swappable SAS/SATA drive bays**, providing 50% more storage density than its predecessor. By delivering high-end storage density within a single machine, iXsystems cuts operating costs and reduces energy requirements.

Storage sizes for the iX-N4236 are customizable, with 250GB, 500GB, 750GB, 1TB, and 2TB hard drives available. Utilizing 2TB Enterprise-class drives, the Orion II provides 72TB of storage within 4U of rack space. For environments requiring maximum storage capacity and efficiency, 45 drive 4U JBOD expansion units are available, each contributing up to 90TB of additional storage. By adding a maximum of four JBOD expansion units, the Orion II offers a staggering capacity of 432TB in only 20U of rack space.

Powerful Intel® Xeon® 5500 Series Processors have a light footprint, while creating a perfect environment for intense virtualization, video streaming, and management of storage-hungry applications. Energy efficient DDR3 RAM complements the other power saving components while still providing 18 slots and up to 144GB of memory overall.

100% cooling redundancy, efficient airflow, and intelligent chassis design ensure that even under the heaviest of workloads, the Orion II remains at an optimal temperature, while still drawing less power than other servers in its class. With a 1400 W Gold Level (93%+ efficient) power supply, the entire system works together to efficiently manage power draw and heat loss.

For more information or to request a quote, visit:

<http://www.iXsystems.com/Orion2>



More TFLOPS, Fewer WATTS

Microway delivers the fastest and greenest floating point throughput in history

Enhanced GPU Computing with Tesla Fermi

- ▶ 480 Core NVIDIA® Tesla™ Fermi GPUs deliver 1.2 TFLOP single precision & 600 GFLOP double precision performance!
- ▶ New Tesla C2050 adds 3GB ECC protected memory
- ▶ New Tesla C2070 adds 6GB ECC protected memory
- ▶ Tesla Pre-Configured Clusters with S2070 4 GPU servers
- ▶ WhisperStation - PSC with up to 4 Fermi GPUs
- ▶ OctoPuter™ with up to 8 Fermi GPUs and 144GB memory

New Processors

- ▶ 12 Core AMD Opterons with quad channel DDR3 memory
- ▶ 8 Core Intel Xeons with quad channel DDR3 memory
- ▶ Superior bandwidth with faster, wider CPU memory busses
- ▶ Increased efficiency for memory-bound floating point algorithms

Configure your next Cluster today!

www.microway.com/quickquote
508-746-7341



FasTree™ QDR InfiniBand Switches and HCAs

- ▶ 36 Port, 40 Gb/s, Low Cost Fabrics
- ▶ Compact, Scalable, Modular Architecture
- ▶ Ideal for Building Expandable Clusters and Fabrics
- ▶ MPI Link-Checker™ and InfiniScope™ Network Diagnostics

Achieve the Optimal Fabric Design for your Specific MPI Application with ProSim™ Fabric Simulator

Now you can observe the real time communication coherency of your algorithms. Use this information to evaluate whether your codes have the potential to suffer from congestion. Feeding observed data into our IB fabric queuing-theory simulator lets you examine latency and bi-sectional bandwidth tradeoffs in fabric topologies.



GSA Schedule
Contract Number:
GS-35F-0431N

Microway
Technology you can count on™