

HTML5 | VLAN | SOGo | LVM2 | Swap Space | KVM | Clonezilla

LINUXTM JOURNAL

Since 1994: The Original Magazine of the Linux Community

System Administration

LVM2 Snapshots
for Data Backup

Build Your
Own SAN
with AoE

Turn Your
Linux Box
into a VLAN
Smart
Switch

Configure
Swap Space for
Stability and
Performance

CLONEZILLA

High-Performance
Open-Source Cloning

SOGo

a Real Exchange
Replacement

Manage KVM
with Virtual
Machine
Manager

Fault
Tolerance
with Ethernet
Bonding

JANUARY 2011 | ISSUE 201
www.linuxjournal.com



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!

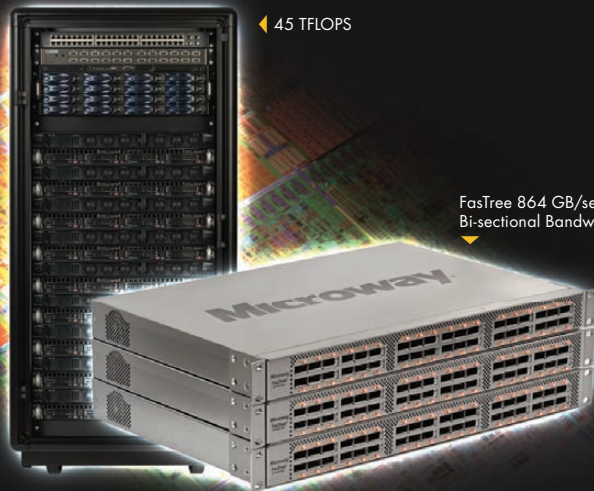
www.microway.com/quickquote
508-746-7341



2.5 TFLOPS

10 TFLOPS

5 TFLOPS



4.5 TFLOPS

FasTree 864 GB/sec
Bi-sectional Bandwidth

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™

WEBSITE PLANS ON SALE!

**ALL WEB HOSTING
PACKAGES JUST:**

\$3.99
per month*
For the first 3 months!

Hurry – offer ends 12/31/2010!



Whether you're a beginner or a professional, 1&1 offers a full range of website solutions to suit your needs. For a limited time, we're offering all web hosting packages at one incredible low price. Website building tools, unlimited traffic, and search engine marketing dollars are included with all packages.

Go to www.1and1.com to choose your package!

DOMAIN OFFERS:

.info only \$0.99 first year* **.com only \$4.99** first year*



Free Web
Marketing Tools



90-Day Money
Back Guarantee



24/7 Toll-Free
Support



Call 1-877-GO-1AND1 or visit us now

www.1and1.com

* Offers valid through December 31, 2010. 12 month minimum contract term applies for web hosting offers. Setup fee and other terms and conditions may apply. Domain offers valid first year only. After first year, standard pricing applies. 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.

CONTENTS

JANUARY 2011

Issue 201



FEATURES

SYSTEM ADMINISTRATION

44

SOGo— Open-Source Groupware

A real, transparent
Microsoft Exchange
replacement.

Ludovic Marcotte

50

Use AoE to Build Your Own SAN

It never goes down and
doesn't break the bank.

Greg Bledsoe

54

VLAN Support in Linux

Of course it can,
it's Linux.

Henry Van Styn

60

Archiving Data with Snapshots in LVM2

The snapshot feature
within LVM2 eases the
burden of retrieving
archived data.

Petros Koutoupis

ON THE COVER

- Clonezilla: High-Performance Open-Source Cloning, p. 68
- SOGo: a Real Exchange Replacement, p. 44
- LVM2 Snapshots for Data Backup, p. 60
- Build Your Own SAN with AoE, p. 50
- Turn Your Linux Box into a VLAN Smart Switch, p. 54
- Configure Swap Space for Stability and Performance, p. 64
- Manage KVM with Virtual Machine Manager, p. 73
- Fault Tolerance with Ethernet Bonding, p. 32

STORAGE WITHOUT BOUNDARIES

Imagine what you can achieve with Aberdeen.

Aberdeen's AberSAN Z-Series scalable storage platform brings the simplicity of network attached storage (NAS) to the SAN environment, by utilizing the innovative ZFS file system.

Featuring the high performance of the Intel Xeon processor 5600 series, the AberSAN Z-Series offers enterprise level benefits at entry level pricing, delivering the best bang for the buck.

Who gives you the best bang for the buck?

	NetApp FAS 2050	EMC NS120	Aberdeen AberSAN Z20
Deduplication	✓	✓	✓
Thin Provisioning	✓	✓	✓
VMware® Ready Certified	✓	✓	✓
Citrix® StorageLink™ Support	✓	✓	✓
Optional HA Clustering	✓	✓	✓
Optional Virtualized SAN	✓	✓	✓
Optional Synchronous Replication	✓	✓	✓
Optional Fibre Channel Target	✓	✓	✓
Unified Storage: NFS, CIFS, iSCSI	✓	✓	✓
Unlimited Snapshots	✗	✗	✓
Unlimited Array Size	✗	✗	✓
Hybrid Storage Pooling	✗	✗	✓
Integrated Search	✗	✗	✓
File System	WAFL 64-bit	UxFS 64-bit	ZFS 128-bit
RAID Level Support	4 and DP	5 and 6	5, 6 and Z
Maximum Storage Capacity	104TB	240TB	Over 1,000TB
Warranty	3 Years	3 Years	5 Years
Published Starting MSRP	Not Available	Not Available	\$8,995



Above specific configurations obtained from the respective websites on Sept. 13, 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. All trademarks are the property of their respective owners. All rights reserved. For terms and conditions, please see www.aberdeeninc.com/abopoly/abterms.htm. IJ036

888-297-7409
www.aberdeeninc.com/lj036

CONTENTS

JANUARY 2011 Issue 201

COLUMNS

- 20** Reuven M. Lerner's
At the Forge
HTML5
- 24** Dave Taylor's
Work the Shell
Calculating Mortgage Rates
- 26** Mick Bauer's
Paranoid Penguin
Building a Transparent Firewall with Linux, Part V
- 32** Kyle Rankin's
Hack and /
Bond, Ethernet Bond
- 80** Doc Searls'
EOF
What Does Linux Want?

INDEPTH

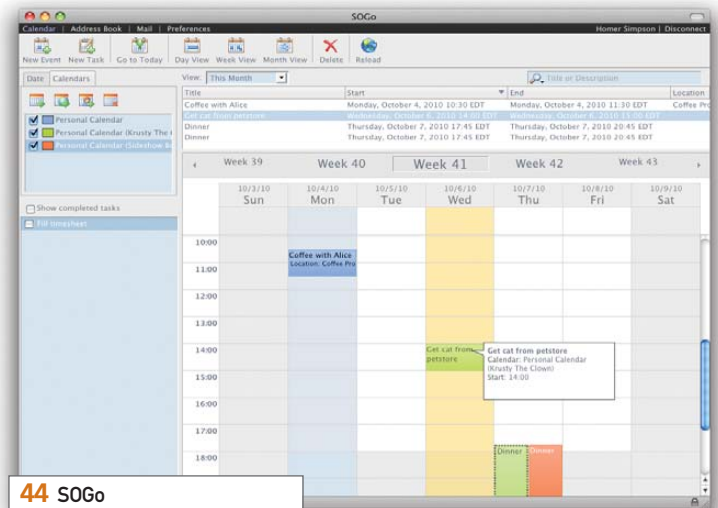
- 64** Linux Swap Space
What you really need to know about configuring swap space.
Tony Kay
- 68** Clonezilla: Build, Clone, Repeat
Clonezilla and DRBL: enterprise-class imaging made easy.
Jeremiah Bowling
- 73** Managing KVM Deployments with Virt-Manager
It's so easy, even a caveman can do it!
Michael J. Hammel

IN EVERY ISSUE

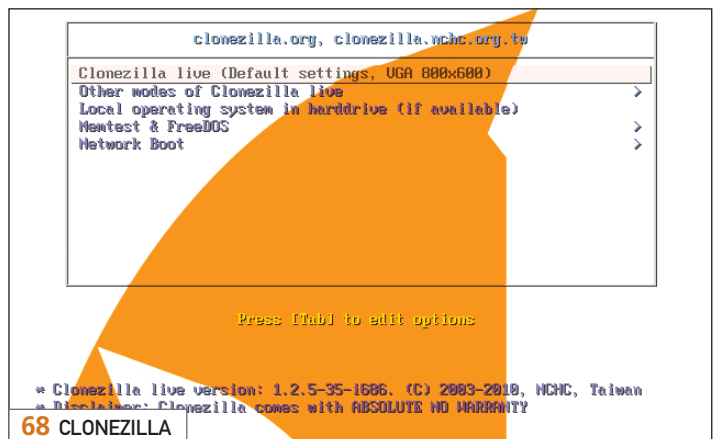
- 8** Current_Issue.tar.gz
- 10** Letters
- 14** UPFRONT
- 38** New Products
- 40** New Projects
- 65** Advertisers Index
- 78** Marketplace



40 SOFA



44 SOGo



68 CLONEZILLA

USPS *LINUX JOURNAL* (ISSN 1075-3583) (USPS 12854) is published monthly by Belltown Media, Inc., 2121 Sage Road, Ste. 310, Houston, TX 77056 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 Pitney Bowes, P.O. Box 25542, London, ON N6C 6B2

MPLS for the masses

\$39⁹⁵



Usually MPLS routers cost more than \$1000, but not anymore. MikroTik gives you the ability to use MPLS in any network. No more big box prices for MPLS! A chicken in every pot!

MPLS stands for Multi Protocol Label Switching. It can be used to replace IP routing - packet forwarding decision is no longer based on fields in IP header and routing table, but on labels that are attached to the packet.

MPLS makes it easy to create “virtual links” between nodes on the network, regardless of the protocol of their encapsulated data. It is a highly scalable, protocol agnostic, data-carrying mechanism. MPLS allows one to create end-to-end circuits across any type of transport medium, using any protocol.

Features:

- Label Distribution Protocol for IPv4
- Virtual Private Lan Service
 - * VPLS LDP signaling
 - * VPLS MP-BGP based autodiscovery and signaling
 - * split-horizon bridging
- RSVP TE Tunnels
 - * explicit paths
 - * CSPF path selection
 - * OSPF extensions for TE tunnels
- Virtual Routing and Forwarding
- MP-BGP based MPLS IP VPN
- OSPF and RIP as CE-PE protocols

Benefits:

- higher speed forwarding in network core
- ability to implement transparent L2 and L3 VPNs (VPLS & VRF)
- reduced VPN overhead compared to legacy tunneling solutions
- traffic engineering to implement QoS and optimize network usage
- ability for the ISP to create VPNs without user interaction
- separate tunnels for voice, video, or data

All MikroTik RouterBOARDS support MPLS, including the **RB750** which costs \$39.95. The RB750 is a SOHO router with a 400MHz Atheros CPU, five ethernet ports, plastic case and PSU. With MPLS, RB750 is capable of wire speed throughput for 1000byte packets and up, maximum 80000 pps with smaller packets.

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

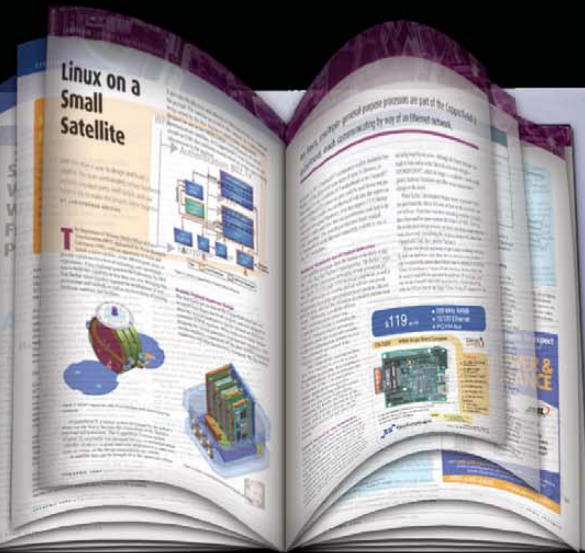
Download to your computer for
convenient offline reading

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

Art Director Garrick Antikajian
garrick@linuxjournal.com

Products Editor James Gray
newproducts@linuxjournal.com

Editor Emeritus Don Marti
dmarti@linuxjournal.com

Technical Editor Michael Baxter
mab@cruzio.com

Senior Columnist Reuven Lerner
reuven@lerner.co.il

Security Editor Mick Bauer
mick@visi.com

Hack Editor Kyle Rankin
lj@greenfly.net

Virtual Editor Bill Childers
bill.childers@linuxjournal.com

Contributing Editors

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

Proofreader Geri Gale

Publisher Carlie Fairchild
publisher@linuxjournal.com

General Manager Rebecca Cassity
rebecca@linuxjournal.com

Senior Sales Manager Joseph Krack
joseph@linuxjournal.com

Associate Publisher Mark Irgang
mark@linuxjournal.com

Webmistress Katherine Druckman
webmistress@linuxjournal.com

Accountant Candy Beauchamp
acct@linuxjournal.com

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

Editorial Advisory Panel

Brad Abram Baillio • Nick Baronian • Hari Boukis • Steve Case
Kalyana Krishna Chadalavada • Brian Conner • Caleb S. Cullen • Keir Davis
Michael Eager • Nick Faltys • Dennis Franklin Frey • Alicia Gibb
Victor Gregorio • Philip Jacob • Jay Kruiuzenga • 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.



we get now geeks think.

PATIENT MRI EXAM

Turbocharged

CPU Cores: 64

Clock Speed: 6.66 GHz

Bandwidth: 100 Gbps

Refresh Rate: 240 Hz

Storage: 32 PB

Latency: 0.002 ms

Packet Loss: 0.00%

Load Avg: 0.01

S. BEACH
GEEK, IMA
023Y MALE
1 800 7419939
23:59:59

R

L

Linux Journal Magazine Exclusive Offer*

15% OFF

Call **1.888.840.9091** | serverbeach.com

Sign up for any dedicated server at ServerBeach and get 15% off*. Use the promo code: **LJ15OFF** when ordering.

* Offer expires December 31st, 2010.

Terms and conditions:

© 2010 ServerBeach, a PEER 1 Company. Not responsible for errors or omissions in typography or photography. This is a limited time offer and is subject to change without notice. Call for details.



SHAWN POWERS

Administrate Me

As a system administrator, one of my favorite things is to be ignored. No, it's not due to a latent social anxiety disorder or anything; it's just because when the sysadmin doesn't hear from anyone, it means things are working. In fact, if things are going really well, we can forward our phones and spend the afternoon on a beach somewhere. No one would ever miss us.

Sadly, that's not usually how things go. Call it job security, call it bad karma, or just blame Bill Gates—for whatever reason, computers break. Even when they don't break, they get old and wear out. In fact, for most of our workdays (and nights), we system administrators spend our lives in a paranoid state ready for the whole world to fall apart. When that happens, everyone suddenly remembers the sysadmin and suddenly is angry with him or her. That's where this issue of *Linux Journal* comes into play. With our system administration issue, we try to fortify your paranoia with redundancy, calm your nerves with best practices and teach the fine art of telling the future to determine when a failure is about to happen. If we do a really good job, you might even learn a few ways to prevent disaster before it strikes at all.

Mick Bauer ends his series on building a transparent firewall. The best offense against outside attack is a strong defense, and Mick will make you paranoid enough to make sure your firewall is top notch. Kyle Rankin gets into the networking act this month as well and shows how to bond Ethernet ports together for redundancy or speed. So many servers come with multiple Ethernet ports, it's a waste not to take advantage of them.

When you have a server with multiple bonded NICs, it certainly makes sense to add storage to it as well. Greg Bledsoe describes how to use AoE (ATA over Ethernet) to build your own SAN at a fraction of the cost of buying one. With hard drives connected directly to your network, it takes out a single point of failure and also allows a gradual expansion without the need for buying new chassis.

With Linux acting as a firewall, and Linux acting as a SAN, why not add one more possibility to the mix? Henry Van Styn not only shows us how to turn our Linux box into a switch, but also how to use VLANs in that switch. VLANs are a powerful way to

secure network traffic, and with Linux acting as a switch, it also can use the security of VLANs in addition to its other abilities.

But, that's just the networking part of this issue. There's lots more to being a sysadmin than filtering a few packets. Michael J. Hammel shows how to manage KVM deployments with virt-manager. Hardware virtualization is a powerful tool, and thanks to virt-manager, those VMs can be configured with a nice GUI tool from any Linux computer that can access the VM host.

Sometimes it's not just virtual machines that need to be installed, however, and that's where Clonezilla comes in. Jeremiah Bowling demonstrates the ins and outs of Clonezilla, a powerful cloning tool that makes imaging new computers a breeze. When imaging, of course, it's important that your original image is exactly how you want it. Tony Kay's article on Linux swap space is something you'll want to read before creating your master image. We don't usually think much about swap space, but it's more than just a safety net if you happen to run out of RAM. And, of course, no system administrator would be caught dead without backups—lots and lots of backups. Petros Koutoupis explains how to take snapshots with LVM2, which is a neat way to take zero downtime snapshots of your Linux system.

We certainly haven't left out our non-sysadmins this month though. If you've been looking for a viable, open-source replacement for Microsoft Exchange, SOGo might be just the thing you're looking for. Sure, it takes some system administration to install it, but once it's going, SOGo is a tool for the end user. Ludovic Marcotte covers the features of this powerful groupware alternative. When you add to that our regular cast of programmers, like Dave Taylor scripting mortgage calculations and Reuven M. Lerner delving into HTML5, this issue is bound to please. For now, I'm going to take this issue and head to the beach. Don't worry; I'll forward my phones in case anything catastrophic happens. Otherwise, I doubt anyone will miss me. ■

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](https://www.freenode.net) IRC channel on Freenode.net.

SharePoint Comes Back to San Francisco!

Attend



SPTechCon

The SharePoint
Technology Conference

Feb. 7-9, 2011 → San Francisco

Hyatt Regency Burlingame

Choose from over
90 Classes
& Workshops!

Learn from the most experienced
SharePoint experts in the industry!

Over 55 exhibiting companies!

“Great! Well-planned, the classes were great for
my organization’s needs.”

Uka Udeh, IT Specialist, DDOT

Register NOW for
Early Bird Discounts!

DIAMOND SPONSOR	PLATINUM SPONSORS		
 Unleashing the Power of SharePoint™	 QUEST SOFTWARE Smart Systems Management	 Kodak	 BITKOO
	 EPC GROUP.NET Empowering Networks	 commvault	

GOLD SPONSORS	SILVER SPONSORS	
 susQtech	 NINTEX	 IRON MOUNTAIN
 TITUS	 SharePoint Engine	 harmon.ie
 conceptSearch Inc	 K2	 SHAREPOINT 911 THE SHAREPOINT AUTHORITY
 boldbrick	 Component One	
 SYNERGY		
 AXcelera		
 MacroView		
 unilytics		
 Atalasoftware		



A BZ Media Event

Follow us at twitter.com/SPTechCon

Full Technical Program Announced!
www.sptechcon.com

letters



Is 54G as Good as 54GL?

Regarding Mick Bauer's "Building a Transparent Firewall with Linux" series in the August, September, November and December 2010 issues: is the 54G*L* a must? I have been upgrading regular 54Gs successfully with DD-WRT for quite a while. Would you say that once OpenWrt is on the device, 54G is as good as 54GL? I have a few of those here, mostly v.2 and v.3, which is why I am asking. Looking forward to your reply.

--
Lee

Mick Bauer replies: Excellent question. The L is Linksys' way of indicating that the device is designed to run Linux, which means you have the manufacturer's assurance that it's possible to load OpenWrt (or DD-WRT or another OpenWrt derivative) and that the device's particular wireless chipset, bridge circuitry and so forth will be recognized under Linux.

However, OpenWrt runs on many other Linksys devices, not just those with an L in the model number. The real measure is the device's status in the OpenWrt Table of Hardware (wiki.openwrt.org/toh/start) and, of course, your own experience. If you've had success running DD-WRT or OpenWrt on a Linksys WTR-54G, obviously it works!

Note that early versions of the 54G shipped with Linux firmware; the WRT54G was the very first device OpenWrt ever supported.

Tech Tips, September 2010

Tanmay Mande's "Download Bandwidth Usage" brought back the excitement of my younger MS-DOS computing days. Like my wheat-eating rodent, which neither Ubuntu nor Fedora now catches, I am getting a little long in the tooth. I am a lone SUSE/Puppy/Slackware user eschewing the 4M Company (Mighty Monopolist Mogul Microsoft). My tooth length means I have absolutely no desire to re-invent the wheel by learning the myriad *nix programming languages and writing my own utilities like Tanmay's download measurer, although if I could find a Bash tome in a second-hand bookshop, learning Bash scripting is worth the effort. After much delay and hassle getting my window into CyberEarth open, I couldn't find a down-/upload meter addon for Firefox. When running SUSE, the download iptables setup script has to be run in a root text window each time after booting my router; boot.local doesn't work as Tanmay implies. The newer distros, with the exception of Slackware, are no longer DIY-learn-as-you-go-along Linuxes, having become too complicated and WinDoze-user friendly. Tanmay Mande, thank you for making my day!

--
Wilfred Tritscher

Tweeting from a Script

In the November 2010 issue of *Linux Journal*, Jamie Popkin mentions using curl to post to Twitter to alert him when someone cancels a user session on his children's computer [see Jamie's article "Controlling Your Linux System with a Smartphone"]. Unfortunately, this method of posting to Twitter no longer works, as it has moved to the OAuth spec. Luckily, a Google search on "Twitter OAuth command line" returns several sites that explain how to get it working again.

--
John Grafton

How to Get My Netflix Fix?

Okay, it bugs me to no end that Netflix does not work on Linux. Now answer me this, which really boggles my mind. Why is it that the Roku device, which runs on Linux, can stream videos, but I can't do the same thing in Ubuntu 9.10? There must be a way to get my Netflix fix without having a dual-boot PC.

--
jpm1973

It is absolutely true that the Linux-powered Roku can stream Netflix, but the Linux desktop cannot. I don't know of any way to "fake" it either. I've tried both Wine and whining, and neither seems to help. Hopefully, Netflix will fix this travesty, but I'm not terribly hopeful.—Ed.

Another Way to Count Files

Regarding Dave Taylor's November 2010, Work the Shell column: another way to count files and avoid grepping away the "No such file" error message is to use "echo pattern*":

```
pat=$1
matches=$(echo $pat*)
if [ "$matches" != "$pat*" ] ; then
    matches=$(echo $matches | wc -w)
else
    matches=0
fi
```

--
Michael Eager

Dave Taylor replies: True. I never really think of using echo in this sort of situation. Good idea!

Happy 200th Issue

It might be a bit late now, but better late than never. Anyway, happy 200th issue! What a milestone! I think every issue is a milestone reached, but the next big ones I'm looking forward to are the June 2013 and July 2014 issues. The first is for *LJ*'s 20th anniversary, and the other is for the 0x100th issue (I don't always use my ten [1010B] fingers to count). I might be one or two issues off, but it doesn't matter. I enjoy every single one!

Some people write songs or poems to

celebrate milestones, but other people like me write code. This one is for you:

```
issue = 200L;
while (1 == 1) {
    publish(issue);
    if (mod (issue, 100L) == 0) {
        celebrate();
        bragAboutIt();
    }
    enjoy(issue);
    issue++;
}
```

--
JSchiavon

```
10 PRINT "Thank You!"
20 GOTO 10
—Ed.
```

Poor Practices Taught in Work the Shell?

I try to ignore most of Dave Taylor's Work the Shell columns, but I find it hard to keep silent when I see repeated examples that teach poor bash programming practices.

In the November 2010 column, all of the examples fail to quote arguments passed to commands like mv, which is sure to blow up the instant the script is run on a file with a space in the filename. Additionally, the same example needlessly calls sed, when bash parameter expansion in the style "yy\${name#xx}" or "\${name/xx/yy}" would do fine. Why not teach the readers these useful features of bash?

Additionally, the author goes to great pains to handle the case where no files match the pattern, including spawning three separate processes and filtering on the error message output of one of those commands. A much better practice in bash would have been simply to enable nullglob with the command `shopt -s nullglob`, which allows glob expansion to expand to the null string. Using nullglob would have been a simple, more efficient and more reliable way to avoid any iterations of the loop if there were no matches.

--
Bob Bell

Dave Taylor replies: Thanks for your note, Bob. It's inevitable that if you get two programmers in the room, there'll be at least three opinions floating around. Your "poor practices" is my constant demonstrations of "get it done, get on with your job" efficiency. Having written books about programming, I know that you can spend a huge amount of time fine-tuning any given code snippet to make it optimal, but in the real world, in a production environment—particularly with shell script programming—it's about solving the problem. Often the solution is lazy, inefficient or even partially broken. But if it does the job, you can move on to the next task. Agreed?

In terms of filenames with spaces in them, you're right. I have been involved with UNIX and Linux for so long that I feel physically ill if I put a space in a filename (only partially kidding!), so I often forget to ensure that my demonstration scripts work for filenames that include spaces. Frankly, Linux is not a very spaces-in-filenames-friendly environment anyway.

Again, thanks for your note. As always, I encourage readers to use what I write as a launching platform, a place to learn more about shell script programming, get ideas for different ways to solve things and enjoy a puzzle solved, even if the solution isn't necessarily a) the most optimal or b) how you'd do it!

Wi-Fi on the Ben NanoNote

I was reading your journal today in the library, and I read among other things Daniel Bartholomew's article on the Ben NanoNote in the October 2010 issue.

Here are some links where you can see that some microSD-Wi-Fi adapters work with Ben NanoNote: en.qi-hardware.com/wiki/Ben_NanoNote_Wi-Fi and www.linux.com/news/embedded-mobile/netbooks/296251:a-review-ben-nanonote-gets-small-with-embedded-linux.

So, the device also can be used, for example, for Web surfing, but it would be better if one managed to embed a USB-Wi-Fi adapter into the device (not so

TS-WIFIBOX-2

A Complete Solution for 802.11g WiFi Applications



qty 1 **\$185**



Powered by a
250 MHz ARM9 CPU

- ❖ Low power (3.2 watts), fanless
- ❖ Power via 5-12 VDC, USB, PoE (opt.)
- ❖ 64MB DDR-RAM
- ❖ 256MB ultra-reliable XNAND drive
- ❖ Micro-SD Card slot
- ❖ RS-232, RS-485, CAN, RTC
- ❖ Header with SPI and 11 DIO
- ❖ 480Mbit/s USB, Ethernet, PoE option
- ❖ Boots Linux 2.6.24 in < 3 seconds
- ❖ Un-brickable, boots from SD or flash
- ❖ Customizable FPGA - 5K LUT
- ❖ Optional DIN mountable enclosure

Ideal for gateway or firewall, protocol converter, web server, WiFi audio, and unattended remote applications

- ❖ Over 25 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

Spaces in Filenames

Love your magazine. I am an IBMer, and a lot of us are moving to Linux (mainly because Windows XP is 32-bit, and Linux is 64-bit).

I had a question on Dave Taylor's article "Scripting Common File Rename Operations" in the November 2010 issue. I like this script, but does it handle spaces in filenames?

Here is a common script that I have, but it fails because of spaces:

```
for n in $(find . -name ".copyarea.db"); do rm -f $n; done
```

None of the Linux commands like spaces very much. And, when I try to wrap things in double quotes, that doesn't work either. Any ideas? Thanks.

--
Doug

Dave Taylor replies: Doug, you're absolutely right. UNIX/Linux does not like spaces in filenames. It's a pain, to say the least.

One easy solution that works much of the time is simply to ensure you surround any occurrence of a filename with double quotes in your scripts (not single quotes, those would stop the "\$" variable from being expanded). So, try this:

```
for n in $(find . -name ".copyarea.db"); do rm -f "$n"; done
```

You also should look at the `-print0` option in `find`, and consider a non-`for` loop, like this:

```
find . -name ".copyarea.db -print0 | xargs -0 /bin/rm -f
```

I've not tested it, but that should do the same thing and considerably faster! Cheers, and thanks for writing.

Finding Program Files in Your Path

I often have wondered if there was a command on my system that had to do with something I was doing and would end up manually listing the directories and grepping for it. I put together a little script to make this easy and called it `findip` for `find-in-path`. It makes looking, for example, for all program files having to do with volume—that is, `findip volum` lists all files with `volum` in their names. Here's the script:

```
# find in path ignore case
NAME=$1
find $(echo $PATH|tr ':' ' ') -iname "*${NAME}*"
exit 0
```

Thanks for a very helpful and interesting publication.

--
Alex

I love the flexibility of Linux scripting. Congrats on bending the command line to your will!—Ed.

impossible if the Ben NanoNote does have a USB-host on it).

Thanks for the article. It was enjoying to read it and made me happy for those people who make things (especially hardware) copyleft.

I first will try to hack my (Linux-based) Eee PC, then maybe try hacking on a Ben NanoNote (when more hacks are described on the Internet). Thanks for the fine journal!

--
Oussama

Daniel Bartholomew replies: Yes, there are some Wi-Fi adapters that work with the Ben NanoNote, but based on what I've read, you need to recompile the kernel to include the appropriate drivers, and there are issues (hotplug not being supported yet is one). I agree that it would be much better to have a wireless chipset and antenna integrated into the NanoNote. I look forward to seeing the second generation of the NanoNote and hope wireless networking is part of it. That said, there's nothing stopping people with the appropriate skills from creating their own version of the NanoNote and to include whatever hardware they have the ability to include (wireless or otherwise). That is what makes copyleft hardware so awesome. Thanks for your comments and happy hacking!

rename.sh

In the November 2010's Work the Shell column by Dave Taylor, I read "`ls -l` actually generates an error message: `ls: No such file or directory`". This is not always true. On my PC, it generates an error message in the Dutch language (I live in Belgium, and Dutch is my native language): `Bestand of map bestaat niet`. So, the line `grep -v "No such file"` doesn't work here. I think a `LANG=C` should be added.

Also, I have downloaded the complete `rename` script (10885.tgz), but there are some errors in it.

For example, at line 14 of the `rename.sh`-script, I see `echo "`. I think there is a second quote missing. There is also a "case" without an "esac" and a "do" without a "done" in the script. I think that part should be something like this:

```
for i; do
```

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.

```
case "$i"
in
-n ) renumber=1 ; shift ;;
-p ) fixpng=1 ; shift ;;
-t ) doit=0 ; shift ;;
-- ) shift ; break ;;
esac
done
```

After these changes, the script is working for me.

-- Jan Wagemakers

Dave Taylor replies: *Thanks for the reminder about the issues associated with localized Linux! I can't explain why the code was missing some elements though. Might be a transmission hiccup. It worked on my test system before I sent it in. Really.*

Giving Up on Linux

The reason I write is that maybe you can change my mind. I've been a Linux user since 2005. First Mandrake, then openSUSE and the last Ubuntu 10.04 (since 6.06). I fought for open-source software, and I honestly believed in it, until I was fed up with trying to make Linux work for me the way I needed and was spending more time researching, building and compiling, than actually doing what I needed to do.

For basic Web surfing and typing documents, it's okay. The stability and security of Linux is remarkable, and so on. When it came to movie editing, I had a lot of problems. After that came the problem of ripping CDs. It was taking so long that I had to go back to Vista to rip and transfer the data to Linux. When I tried to rip it to Ogg files, my music did not sound right. It sounded as if there were sequences missing. Later, it seemed that I often was losing the metadata of my MP3s, and for me, that was important, because I'm converting sermons for my church from CDs to MP3s and uploading them to the Web site. Finally, burning CDs with Brasero was too buggy, so I used K3B instead. Again, for writing simple data it was good, but when I wanted to burn a sermon on a CD, instead of the metadata or information that was supposed to appear, it was some foreign Asian language.

I could have lived without the movie

editing, but for me, CDs, MP3s, metadata and CD-burning problems are too much. So you see, I was spending more time trying to fix these problems than actually getting my work done. So, unfortunately, I had to switch *all* of my work back to Windows Vista.

I can't believe what I'm saying, but Windows Vista does the job right, the first time.

Maybe I should try another distro. I don't know anymore. I do not want to go through another process of learning a new distro and spending hours again trying to fix it. I was very sick the past year and a half, so I'm exhausted, and I need things to just work sometimes.

So if you can help, I would appreciate it. If not, it's okay, but my ride in the Linux community is over.

--
JF

First, I will freely admit I feel your pain. I struggle with movie editing (a big part of my job here at Linux Journal), and at times, I feel like I'm spending more time preparing to work than actually working. Since Linux is developed by people basically for themselves (that's a gross generalization), often the most common things work the best out of the box.

Sometimes I find that manipulating the underlying command-line tools via shell scripts works better for me. This is especially true with video format conversion and so on. Although I personally haven't done much with CD burning, it's possible the same is true.

If you're still interested in trying another distro, perhaps one aimed specifically at media creation may be tweaked a bit better for your needs. Ubuntu Studio is one of those options, but there probably are others as well.

Finally, I'm not sure if you're familiar with the support available on-line for such things. You mentioned Ubuntu is your distro of choice lately, and the Ubuntu forums are really great for getting support from both peers and developers. In the end, of course, it's up to you to use whatever system meets your needs. We won't judge you!—Ed.

diff -u

WHAT'S NEW IN KERNEL DEVELOPMENT

In the ongoing saga of **big-kernel-lock** removal, we seem to be getting very, very close to having no more BKL. Its remaining users in the linux-next tree seem to be down to less than 20, and most of those are in old code that has no current maintainer to take care of it. Filesystem code, networking code and a couple drivers are all that remain of the once ubiquitous big kernel lock.

Arnd Bergmann has posted a complete list of offending code, and a large group of developers have piled on top of it.

The truly brilliant part of ditching the BKL is that a lot of people felt completely stymied by it for a long time. It was absolutely everywhere, and there didn't seem to be any viable alternative. Then the wonderful decision was made to push all the occurrences of the big kernel lock out to the periphery of the kernel—in other words, out to where only one thing would depend on any particular instance. Before that decision, the BKL was deep in the interior, where any given instance of it would be relied on by whole regions of code, each relying on different aspects of it. By pushing it out to the periphery, all of a sudden it became clear which features of the BKL were actually necessary for which of its various users.

As it turned out, most BKL users could make do with a much simpler locking structure, although a few had more complex needs. But, that whole period of time is really cool, because the problem went from being super-intractable, to being pretty well tractable, and by now, it's almost tracted right out of existence.

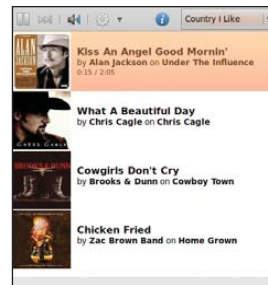
If you do any kernel development, you'll notice the **LKML-Reference** tag in every git patch submission, but it might not make total sense to you in this age of on-line e-mail clients, if

you haven't read www.faqs.org/rfcs/rfc5322.html lately. Recently, it came up on **linux-kernel**, that maybe the LKML-Reference tag should just be a URL linking to the linux-kernel post corresponding to that git submission, as stored in any one of many linux-kernel Web archives. But, as several folks pointed out at the time, URLs come and go. Message-ID headers are eternal.

In the old days, the question of which **GCC** version to support was largely a question of which GCC version managed to piss off **Linus Torvalds**. Back then, the question tended to be, "How long can we keep compiling with this extremely old version of GCC, now that the GCC developers have implemented this weird behavior we don't like?" Nowadays, the question is more along the lines of, "How soon can we stop supporting kernel compilations under this very old version of GCC that is quirky and requires us to maintain weird stuff in the kernel?"

The question came up recently of whether it would be okay to stop supporting the 3.3 and 3.4 versions of GCC. It turns out that even though GCC 3.3.3 has problems that make it not work right with the latest kernels, certain distributions ship with a patched version that they also call GCC 3.3.3. So, any attempt to alert users to the potential breakage would cause only further confusion. Meanwhile, GCC 3.4 apparently is being used by most embedded systems developers, for whatever reason. And, GCC 3.4 also is a lot faster on ARM architectures than more recent versions of GCC are. So for the moment at least, it seems that those are the oldest versions of GCC still supported for kernel compiling. You also can use more recent GCC versions if you like.—**ZACK BROWN**

Pithos



I love Pandora Radio. I really do. Unfortunately, the Web browser is an awkward interface for listening. Oh sure, it's effective and works fine. But, it's missing things like media key support, and if you accidentally close your browser window, BOOM—no music.

Enter Pithos. With a very simplistic interface, Pithos doesn't do much, but what it does, it does very well—plays Pandora Radio. If your keyboard has multimedia keys, you'll appreciate Pithos' media key support as well. If your phone rings, simply press the pause button on your keyboard to pause the music regardless of whether the application has focus.

If you like Pandora Radio, check out Pithos. It's worth the download: kevinmehall.net/p/pithos.—**SHAWN POWERS**

kevinmehall.net/p/pithos.—**SHAWN POWERS**

They Said It

Never underestimate the determination of a kid who is time rich and cash poor.
—**Cory Doctorow, *Little Brother*, 2008**

For a list of all the ways technology has failed to improve the quality of life, please press three.
—**Alice Kahn**

Technology adds nothing to art.
—**Penn Jillette, *WIRED* magazine, 1993**

Technology...the knack of so arranging the world that we don't have to experience it.
—**Max Frisch**

Imagine if every Thursday your shoes exploded if you tied them the usual way. This happens to us all the time with computers, and nobody thinks of complaining.
—**Jef Raskin, interviewed in *Doctor Dobbs' Journal***

If computers get too powerful, we can organize them into a committee—that will do them in.
—**Bradley's Bromide**

NON-LINUX FOSS



There may be a battle in the Linux world regarding what instant-messaging client is the best (Pidgin or Empathy), but in the OS X world, the battle never really started. Adium is an OS X-native, open-source application based on the libpurple library. Although there is a native Pidgin client for OS X, it's not nearly as polished and stable as Adium. With Apple's reputation for solid, elegant programs, the Open Source community really showed it up with this answer to Apple's iChat program. Adium wins on multiple levels, and its source code is as close as a click away. If you ever use OS X and want to try some quality open-source goodness running on the "other" proprietary operating system, check out Adium: www.adium.im.

—SHAWN POWERS

BackupPC

Some tools are so amazing, but unfortunately, if no one ever talks about them, many folks never hear of them. One of those programs is BackupPC. You may have heard Kyle Rankin and myself talk about BackupPC on the *Linux Journal Insider* podcast, or perhaps you've seen us write about it here in *Linux Journal* before. If you haven't checked it out, you owe it to yourself to do so. BackupPC has some great features:

Host	User	Full Count	Full Age (days)	Full Size (GB)	Speed MB/sec	Incr Count	Incr Age (days)	Last Backup (days)	State	Last Attempt
Hosts with good Backups										
There are 19 hosts that have been backed up, for a total of:										
• 167 Full backups of total size 2490.17GB (prior to pooling and compression).										
• 194 incr backups of total size 100.36GB (prior to pooling and compression).										
accounts		9	3.2	3.97	1.73	11	1.1	1.2	idle	idle
adrian_rsync		9	1.6	0.09	5.34	10	0.2	0.2	idle	done
adrian_rsync		10	0.1	14.10	0.99	11	1.5	0.1	idle	done
busbuntu		9	3.1	18.29	1.54	11	1.1	1.1	idle	idle
calvin		9	0.4	9.12	1.82	11	3.6	0.4	idle	idle
dbert		6	28.6	26.54	2.36	1	32.2	28.4	idle	backup failed (No Ret (dumpe for share 2rc)
emperor_rsync		9	3.2	2.81	2.68	11	1.2	1.2	idle	idle
hobbes		9	18.8	23.12	0.82	12	7.1	7.1	idle	never disabled
hose		9	2.8	40.62	2.30	11	0.2	0.2	idle	done
itso		9	1.6	8.88	4.93	10	0.2	0.2	idle	done
kenyon_rsync		9	3.8	65.86	4.00	11	1.5	1.5	idle	idle
leo		9	3.0	35.21	2.16	10	1.0	1.0	idle	idle
sebast		9	1.6	0.26	1.29	10	0.2	0.2	idle	done
smorris		9	3.2	5.25	2.85	11	1.2	1.2	idle	idle
stef		9	3.2	17.10	5.33	11	1.2	1.2	idle	idle
thead		8	4.1	0.16	1.92	10	1.2	1.2	idle	idle
tracuradex		10	0.2	15.74	2.29	11	1.5	0.2	idle	done
tyler_rsync		9	1.6	0.12	2.15	10	0.2	0.2	idle	done
unsmooth		9	0.2	13.34	3.10	11	1.5	0.2	idle	done
Hosts with no Backups										
There are 2 hosts with no backups.										
adrian		0		0.00		0			idle	backup failed (No Ret (dumpe for share 0drive)
hose		0		0.00		0			idle	backup in progress

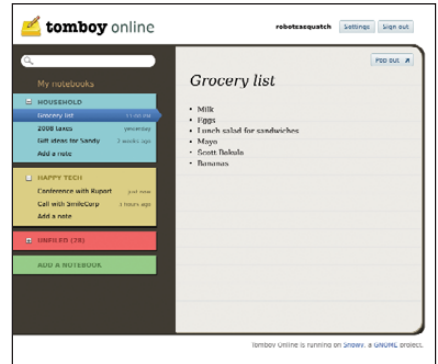
- Hard drive-based backups, no tape swapping.
- Support for NFS, SSH, SMB and rsync.
- Hard linking to save valuable disk space.
- Individual files can be restored in place in real time.
- Powerful and simple Web interface.

- E-mail notification on errors.
- Free!

BackupPC is one of those projects that doesn't get updated terribly often. It doesn't have flashy graphics. It doesn't require constant maintenance. It just works, and it works well. Check it out: backupp.c.sourceforge.net.

—SHAWN POWERS

Snowy—If Ubuntu One Leaves You Feeling Cold



Canonical has quite a cloud service going with Ubuntu One. It syncs files, music, preferences, contacts and many more things. But, it's only partially free, and the server is completely private and proprietary. The philosophy of keeping the server bits completely closed turns off many of us in the Open Source community. Yes, Canonical has every right to develop Ubuntu One as it sees fit, but as users, we also have the right to look elsewhere. Enter Snowy. No, it's not an all-encompassing replacement for Ubuntu One, but Snowy is designed as a free (in both senses) syncing solution for Tomboy Notes.

If you want to access your Tomboy Notes on-line as well as from several computers while keeping them all in sync, keep your eye on Snowy. The developers hope to get a free syncing service on-line, powered by Snowy, by the time Gnome 3.0 launches. Of course, you also can download Snowy yourself and host your notes on your own server, which is exactly what we hoped Ubuntu One would be when we first heard about it. Check out the progress at live.gnome.org/Snowy.

—SHAWN POWERS

Linux for Science

Welcome to a new series of short articles on using Linux and open-source software for science. The tools available for free (as in beer) have lowered the cost of participating in computational science. The freedom (as in speech) you have when using these tools allows you to build upon the work of others, advancing the level of knowledge for everyone.

In this series, I'll look at some small science experiments anyone can do at home, and then consider some tool or other available on Linux that you can use to analyze the results. In some purely theoretical spheres (like quantum chemistry or general relativity), I'll just look at the tool alone and how to use it without the benefit of an accompanying experiment.

The first experiment is a classic—the simple pendulum (en.wikipedia.org/wiki/Pendulum). When you look at a simple pendulum, there are two obvious parameters you can change: the mass of the pendulum bob and the length of the string. A simple way to do this at home is to use a bolt with nuts. You can

tie a string to the head of a bolt and tie the other end to a pivot point, like a shower-curtain rod. Then, you simply can add more weight by adding nuts to the bolt. This is the basic experimental setup for this article.

The data to collect is the time it takes for each oscillation (one full back-and-forth motion). Because you will want to figure out which parameters affect the pendulum, you'll need to do this for several different masses and lengths. To help get consistent times, let's actually time how long it takes for ten oscillations. To average out any reaction time issues in the time taking, let's do three of these measurements and take the average. You should end up with something like Table 1.

Table 1. Pendulum Data

Length (cm)	Weight (g)	Time (s)
18.8	102.0	0.9
18.8	118.5	0.9
18.8	135.0	0.9
18.8	151.5	0.9
37.6	102.0	1.3
37.6	118.5	1.3
37.6	135.0	1.3
37.6	151.5	1.3
57.6	102.0	1.5
57.6	118.5	1.5
57.6	135.0	1.5
57.6	151.5	1.5
88.8	102.0	1.9
88.8	118.5	1.9
88.8	135.0	1.9
88.8	151.5	1.9

Now that you have the data, what can you learn from it? To do some basic analysis, let's look at Scilab (www.scilab.org). This is a MATLAB-like application that can be used for data analysis and graphing. Installing on Ubuntu, or other Debian-based distributions, is as simple as:

```
sudo apt-get install scilab
```

On startup, you should see something like Figure 1.

Usually, the first thing you'll want to do is graph your data to see whether any correlations jump out at you. To do that, you need to get your data into Scilab. The most natural format is three vectors (length, mass and time), with one row for each measurement you made. In Scilab, this would look like the following:

```
height = [18.8, 18.8, 18.8, 18.8,
          37.6, 37.6, 37.6, 37.6,
          57.6, 57.6, 57.6, 57.6,
          88.8, 88.8, 88.8, 88.8];
weight = [102.0, 118.5, 135.0, 151.5,
          102.0, 118.5, 135.0, 151.5,
          102.0, 118.5, 135.0, 151.5,
          102.0, 118.5, 135.0, 151.5];
times = [0.9, 0.9, 0.9, 0.9,
         1.3, 1.3, 1.3, 1.3,
         1.5, 1.5, 1.5, 1.5,
         1.9, 1.9, 1.9, 1.9];
```

You probably will want to use this data over and over again, doing different types of analysis. To do this most simply, you can store these lines in a separate file and load it into your Scilab environment when you want to use it. You just need to call `exec()` to load and run these variable assignments. For this example, load the data with:

```
exec("~/pendulum1.sce");
```

You can see individual elements of this data using the `disp()` function. To see the first value in the times vector, you would use what's shown in Figure 2. To do a simple 2-D plot, say, of height vs. times, simply execute:

```
plot(height, times);
```

This doesn't look very descriptive, so let's add some text to explain what this graph shows. You can set labels and titles for your graph with the `xtitle` command:

```
xtitle("Pendulum Height vs Time", "Height(cm)", "Time(s)");
```

This produces a graph that looks like Figure 3. But, you have three pieces of data, or three dimensions. If you want to produce a 3-D graph, use:

```
surf(height, weight, times);
```

This produces a surface plot of the data. Because this experiment seems so clear, you won't actually need a full 3-D plot. All of this data visualization points to weight not really

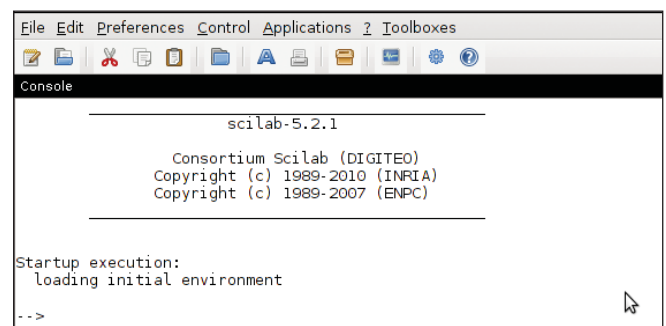


Figure 1. Scilab Startup

```

File Edit Preferences Control Applications ? Toolboxes
Console
-----
scilab-5.2.1

Consortium Scilab (DIGITEO)
Copyright (c) 1989-2010 (INRIA)
Copyright (c) 1989-2007 (ENPC)
-----

Startup execution:
Loading initial environment
-->exec("~/pendulum1.sce");
-->disp(times(1));

0.9
-->

```

Figure 2. First Value in the Times Vector

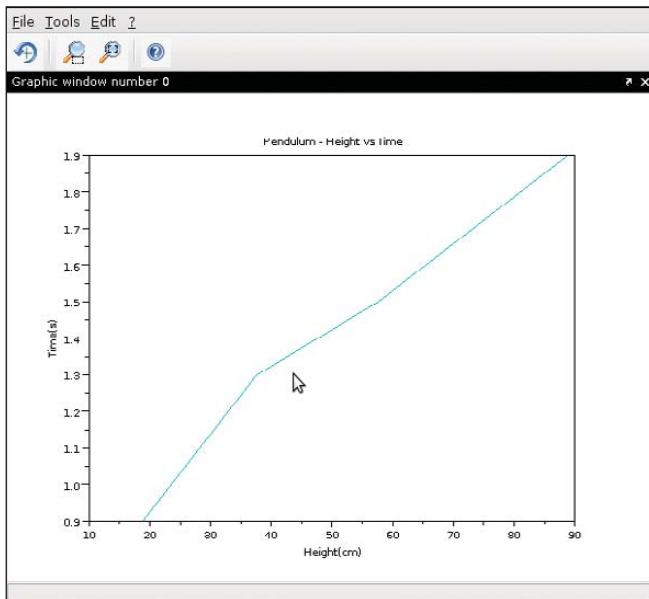


Figure 3. Pendulum Height vs. Time

having any influence on time. So, let's focus on the relationship between the length of the pendulum and the time. The graph looks like an almost straight line, so let's assume that it is and see where we get with it. The formula for a straight line is $y=a+bx$. The typical thing to do is to try to fit a "closest" straight line to the data you have. The term for this is linear regression. Luckily, Scilab has a function for this called regress(). In this case, simply execute:

```

coeffs = regress(height, times);
disp(coeffs(1));
disp(coeffs(2));

```

This ends up looking like Figure 4. From this, you can see that the slope of the straight line you just fit to your data is 0.0137626 s/cm. Does this make sense? Let's look at some theory to check out that number.

According to theory, you should have plotted the square of



ASA COMPUTERS

Want your business to be productive?

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

" Since 1989 Integration and service with pride "

ASA 1U Series



Intel® Atom™ processors D510 and D410 based platform and are designed for embedded industrial PC (IPC) applications. These quiet, energy saving solutions make ideal network appliances, print and email servers.

ASA 2U Series

QPI, Intel® Xeon® Processor 5500 Series in a high-density 2U form-factor are ideal for network infrastructure, front-end enterprise, and minimal downtime cluster server systems maximum upto 4 nodes in 2U.



ASA 3U Series



Server series excel under iSCSI/NAS/JBOD environments, the 3U servers support high availability storage and mission critical business applications.

ASA 4U Series

Optimized for enterprise-level high-capacity storage applications, features 36x (24 front + 12 rear) 3.5" Hot-swap HDD bays, reliable and hassle-free maintenance storage system.



ASA Blade Series



Blade server with, Highest computing density (20 DP nodes and 2.56TB mempry in 7U) Fastest and Most Cost-Effective Networking Solution (Infiniband DDR/QDR support)

E-mail - sales@asacomputers.com
Call - 1800-REAL-PCS



ASA Computers, Inc
645 National Ave,
Mountain View, CA 94043
www.asacomputers.com



**Powerful.
Intelligent.**

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

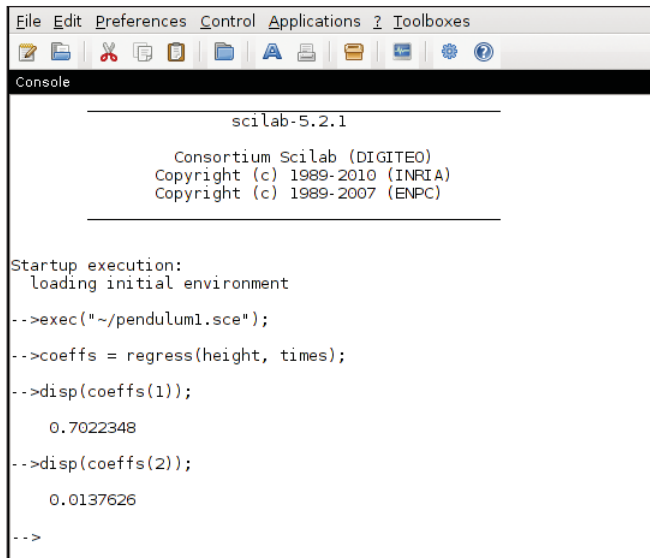


Figure 4. Using Scilab's regress() Function

the time values against the length of the pendulum. To get the square of the time values, use:

```
times = times .* times;
```

This multiplies the first entry of the time vector to itself, the second entry to itself (and so on), down the entire vector. So, the new vector times contains the square of each entry in the vector times. If you now do a linear regression with times instead of times, you get the following result:

```
a = 0.1081958
b = 0.0390888
```

According to theory, the value of a should be given by $((2 * \pi)^2 / g)$, where g is the acceleration due to gravity. According to Scilab, the value is:

```
ans = (2 * 3.14159)^2 / (9.81 * 100);
disp(ans);
```

You need to adjust the value of g by a factor of 100 to

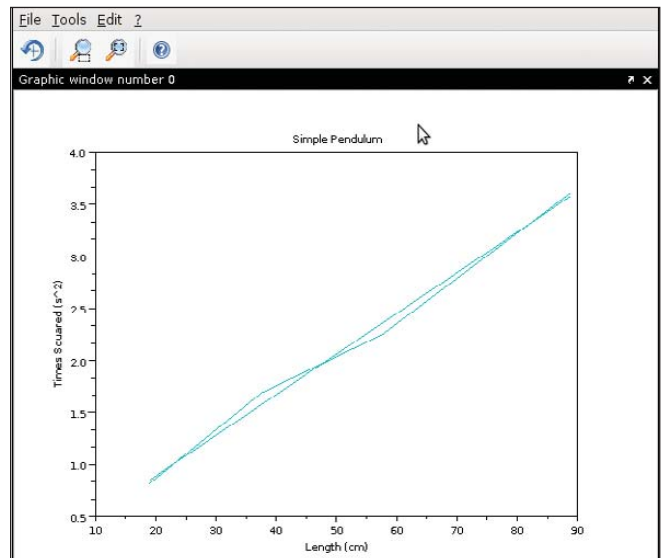


Figure 5. Simple Pendulum Graph

change it to the correct units of cm, and this gives you 0.0402430. To the second decimal place, this gives you 0.04 from the experiment and 0.04 from theory. What does this look like graphically? You can generate the two graphs with:

```
plot(height, times);
plot(height, 0.1081958 + 0.0390888*height);
xtitle("Simple Pendulum", "Length (cm)", "Times Squared (s^2)");
```

This looks like Figure 5. It seems like a reasonably close match, considering that the spread of measured pendulum lengths covers only 70cm. If you made measurements with pendulum lengths over a larger range of values, you likely will see an even closer match to theory. But, as an easy example experiment to show a small list of Scilab functions, you've already seen that simple pendulums seem to follow theory reasonably well.

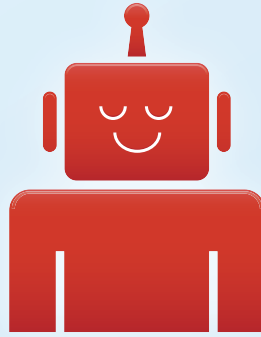
Next month, I'll introduce Maxima and take a look at the math behind the theory of the simple pendulum, seeing whether I can derive the usual results as taught in introductory physics.

If you want me to cover any specific areas of computational science, please contact me with your ideas.—**JOEY BERNARD**

LinuxJournal.com—a Fantastic Sysadmin Resource

As you flip through the pages of this month's *Linux Journal*, you will find a wealth of resources to help you with a variety of system administration tasks. When you have thoroughly absorbed it all, and your head is in danger of exploding, I'd like to offer you an additional resource that will help you achieve total sysadmin Nirvana—LinuxJournal.com. Visit www.linuxjournal.com/tag/sysadmin to find a constant flow of tips and tricks, industry commentary and news. You'll stay up to date on trends and discover new tools to add to your bag of tricks with the help of LinuxJournal.com.

If you are a Linux sysadmin, slap a bookmark on that page and visit often. Please also consider leaving a comment while you're there. We'd like to hear your feedback, as well as have the opportunity to interact with you, our readers. Best of all, you can contribute your own tips, tricks and hacks. Just e-mail webeditor@linuxjournal.com if you have something to share, and you could be published at LinuxJournal.com. We look forward to hearing from you!—**KATHERINE DRUCKMAN**



Lullabot™

Learn Drupal & jQuery

FROM THE COMFORT OF
YOUR LIVING ROOM



The Lullabot Learning Series includes everything you need to become a Drupal & jQuery expert from the comfort of your living room! The videos are available in both DVD format and high-definition video download.

Purchase the videos at <http://store.lullabot.com>



REUVEN M. LERNER

HTML5

HTML5 is coming, and it's going to change the way you develop Web apps. Read on to find out how.

One of the amazing things about the Web always has been the relative ease with which you can create a site. All you have to do is learn a few HTML tags, create a file that uses those tags, and *voilà*, you have a one-page site. Learn another tag or two, and you now can create multipage sites that contain images, links to other pages and all sorts of other goodies. Really, it doesn't take more than a few hours to learn the basics of HTML.

The problem is that this shallow learning curve has long masked the fact that HTML's vocabulary hasn't kept up with the times. Yes, it's very easy to create a site, but when you want to start styling the site, things become a bit more complex. And, I don't mean they become complex because CSS is hard for many people to understand (which it is). Rather, things become complex because styles often are attached to span and div tags, which means that pages end up containing many span and div tags, each with its own ID and/or class, so that you can style each thing just right.

Now, there's nothing technically wrong with having dozens of div tags on a page of HTML. But, at a time when Tim Berners-Lee and others are talking about the "semantic Web", and a growing number of computers (rather than people) are trying to retrieve and parse documents on the Web, it seems odd to stick with such a limited vocabulary.

While I'm complaining, it's hard to believe that HTML forms barely have changed since I started using them in 1993. Given the number of sites that ask you to enter e-mail addresses or dates, you would think that HTML forms would include special provisions for these types of inputs, rather than force people to use text fields for each one.

In fact, there are a whole bunch of problems with HTML in its current form, and with many of the CSS selectors that allow you to style it nicely. HTML has been a part of my life for so long, some of those problems didn't even occur to me until I started to think about them in greater depth. Fortunately for the Web though, I'm not the one in charge of such thinking. After a number of fits and starts, the HTML5 specification (and a few other related specifications, such as CSS3), which includes no small number of improvements, is beginning to gain popularity.

The good news is that HTML5 (and friends) has a great deal to offer Web developers. Indeed, I've already switched all of my new development to use HTML5, and I'm hoping to back-port applications where possible. However, HTML5 is a catchall phrase

for a huge number of different tags, functions and CSS selectors—and, each browser manufacturer is implementing these piecemeal, with no guarantee of 100% compliance in the near future.

This puts Web developers in the unenviable position of being able to enjoy a great deal of new functionality, but also constantly having to check to see whether the user's browser can take advantage of that functionality. I find this to be a bit ironic. For years, many of us have touted Web and server-side applications as a way of getting around the headaches associated with the compatibility issues that plague desktop applications. Although browser compatibility always has been an issue of sorts, the problems we've seen to date pale in comparison with our current issues, in part because the HTML5 elements are more advanced than we've seen before. (The closest equivalent would be the browsers that lacked support for forms, back in the early days of the Web.) At the same time, JavaScript is now mature enough to provide us with a way to test for the presence of these features, either on our own or by using a third-party library.

This month, I'm looking at some of the promise HTML5 brings to the table, with a particular emphasis on some of the syntax, elements and attributes that are coming into play. I also explain how to use Modernizr, an open-source JavaScript library that automates testing for various HTML5 features, so you can use an alternative.

I should note one subject that has been at the center of much public discussion of HTML5 that has to do with video and audio formats. These topics certainly are important and of interest, but they also are complicated, in terms of browser compatibility and licensing issues. It's true that HTML5 simplifies the use of video in many ways, but it's still a complex issue with no truly simple resolution. If you're interested in this subject, I suggest you look at one or more of the books mentioned in the Resources section of this article.

Doctypes and Tags

If you're like me, the first thing you do when you create the first standards-compliant HTML (or XHTML) page on a new site is copy the doctype from an existing site. The doctype not only provides hints to the user's browser, indicating what it can and should expect, but it also provides a standard against which you can check the validity of your HTML. If you fail to provide a doctype, not only are you failing to hitch your wagon to any standard, but you're also telling

Microsoft's browsers that they should operate in "quirks mode", which explicitly ignores standards and will wreak havoc on your HTML and CSS.

Modern HTML declarations are long-winded and easy to get wrong, so I never type them myself, but rather copy them, either from an existing project or from someone else on the Web. Fortunately, HTML5 simplifies this enormously. Generally, you just have to put the following at the top of the page:

```
<!DOCTYPE html>
```

After the doctype, the HTML document looks and works much like you might expect. For example, a document has a <head> section, typically containing the document title, links to stylesheets, metatags and imported JavaScript libraries. Perhaps the most common metatag you will use is the one determining the character encoding; nowadays, just about everyone should use UTF-8, which you can specify as:

```
<meta charset="utf-8" />
```

Following the <head> section is the <body> section, which contains the page's actual content. Tags continue to work as they did before, but the rules are somewhat relaxed. For example, you no longer need to quote attribute values that lack whitespace (although I think it's a good idea to do so), and you can omit the self-closing trailing slash on such tags as .

If you are tired of using <div> to divide up your page, and use an "id" attribute of "header", "footer" or "sidebar", cheer up, you're not alone. Google apparently did some statistical analysis of Web pages and determined that a huge number of sites use divs to set up their headers and footers, among other things. In order to make HTML5 more semantically expressive, the specification includes a number of new sectional tags, such as <section>, <article>, <header> and <footer>. You even can indicate that a particular set of links are for navigation, such as a menu bar, by putting them inside a <nav> tag. Note that these new tags don't change anything other than the semantics, as well as the ability to style them by tag, rather than by ID. Although this won't necessarily change the technical layout of pages, it will make them easier to read and understand, and it also will make it easier for search engines to parse and deal with documents without having to look at IDs and classes, which are arbitrary in any event.

HTML Form Elements

If you ever have created a Web application, you undoubtedly have needed to ask people to enter their e-mail addresses or telephone numbers. In both cases, you needed to use a text field, and then presumably check (using JavaScript, a server-side program or both)

to ensure that the e-mail addresses were valid, or that the phone numbers contained only valid characters. Fortunately for users and developers alike, HTML5 specifies a number of new types of input and form elements, from fields specifically meant for telephone numbers and e-mail addresses, to color pickers and sliders.

Before HTML5, if you wanted someone to enter a URL, you would use a simple text field:

```
<input type="text" name="homepage" />
```

In HTML5, you can use the special url input type:

```
<input type="url" name="homepage" />
```

Many Web sites pre-populate text fields with a hint or a description of what should be entered there (for example, "Enter a search term here"), which disappears when you first click in the field. HTML5 automates this, allowing you to provide "placeholder" text:

```
<input type="text" name="search" placeholder="Search here" />
```

You even can restrict the values you will allow in various fields, using regular expressions, by setting the "pattern" attribute. You also can ask users to input dates and times without having to specify six separate inputs:

```
<input type="datetime" name="datetime" />
```

These form elements have more than just semantic value. The browser itself then can monitor the elements to ensure that submitted data is valid. This doesn't remove the need for server-side validation and checking, but it does greatly simplify things.

All of this sounds wonderful and will make HTML forms more useful and valid than they have been to date. But, and you knew there had to be a catch, support for these HTML form elements is very spotty and inconsistent across browsers. Opera seems to be the leader in supporting them, with Apple's iPhone/iPad following, and the Safari browser coming afterward. Support for such terms in Firefox is nearly nonexistent. Now, it's true that when a browser is asked to render an input tag it doesn't recognize, it produces a text field, which generally is fine. And, it's possible to get around the problematic cases, such as date- and color-pickers, with JavaScript. But, I'm still frustrated that it will be some time before we will see it implemented.

Canvas

Another killer HTML5 feature, the "canvas", has been in the works for several years already. It provides a 2-D drawing area that you can control (writing and reading) using JavaScript. The canvas makes it easy to create

graphs, charts and even simple drawing programs, with functions that support the creation of various shapes, lines, images and even gradient fills.

I tend to be a text-oriented kind of guy who relies on designers to do much of the graphics in Web sites. Nevertheless, it's clear to me that the canvas, which is supported by recent versions of Safari and Firefox and will be included in Internet Explorer 9, will open up many possibilities—not only for displaying graphs and charts, but also for drawing under and over text and graphics, and allowing for new, mouse-based interactions and information display within the browser.

Geolocation

One of my favorite new features in HTML5 is geolocation. To date, geolocation has been a very iffy business, often depending on IP addresses. This often produces results that aren't quite accurate—for example, most IP-based geolocation libraries indicate that my house is in the city of Lod, about a 20-minute drive from my city of Modi'in. Now, that's not too far off if you realize how large the world is, but it's not quite good enough for geolocated applications, from supermarket finders to friend locators.

Although HTML5 doesn't officially include a geolocation standard, it's being developed and released along with HTML5, so I feel comfortable lumping it together with the standard. (And I'm not the first one to do so.) By providing functionality in the browser, it's possible for a JavaScript program to grab your current location:

```
navigator.geolocation.getCurrentPosition(function(position) {
    alert("longitude = " + position.coords.longitude + ",
    ↪latitude = " + position.coords.latitude);
});
```

There are a variety of additional pieces of functionality, including some that will help you track speed and movement, for applications that help with navigation. And, I can't tell you why, but the geolocation in HTML5 browsers consistently has been more accurate than the simple IP-address locators. That alone is good news for people planning to rely on such functionality for application development.

Now, if the general availability of geolocation information to any Web application gives you goosebumps, rest assured that the good folks creating these standards have tried to ensure your privacy. When the geolocation function is invoked, and before it returns any results, the user is presented with a dialog box in the browser, asking if it's okay to share location information. This message specifically indicates the site that is asking for the data, blocking its execution until the user responds.

In case you're wondering, the fact that you can get location information in JavaScript doesn't mean you're forced to write all your software in the client using JavaScript. You always can take the geolocation information and send it to the server using an Ajax call.

Geolocation depends on being able to rely on this functionality existing in the browser. At the time of this writing, there is full support in a variety of browsers, but not in Internet Explorer or Opera. On such systems, you might need to contact the server to perform an IP-based geolocation server call, with all of the issues that raises.

That said, I'm rather excited about the introduction of geolocation in future browsers, and I'm looking forward to see what applications crop up as a result.

Modernizr

As I indicated earlier, it's nice to know that HTML5 is being rolled out in stages, and that we don't need to

Resources

HTML5 has become a hot topic lately, leading to a large number of books, blog postings and even magazine articles (like this one!) on the topic. Additionally, the W3C has a number of standards and papers about HTML5.

The best book I've read on the subject isn't even a proper book at the time of this writing, but rather a free on-line resource written by Mark Pilgrim. If you're familiar with Pilgrim's previous work, such as *Dive into Python*, you know that his writing is excellent. Not surprisingly, this was the first resource to which I turned to bone up on HTML5, and it continues to be my favorite combination of tutorial and reference.

I have read two other books on the subject, each of which introduces things in its own way. From the Pragmatic Programmers

comes *HTML5 and CSS3* by Brian Hogan. This book is well written, providing (I think) a gentler introduction to the subject than other books. Apress has its own book, called *Pro HTML5 Programming* by Peter Lubbers, Brian Albers and Frank Salim. This last book also is aimed at beginners, but I found its examples to be less useful than those in the other books.

There have been a number of good articles and blog postings about HTML5 as well. One particularly interesting one is www.smashingmagazine.com/2010/09/23/html5-the-facts-and-the-myths.

Finally, the Modernizr library is at www.modernizr.com. Full documentation about what it provides is on that site, with terrific detail about what information is available to your application.

wait for a complete implementation to be ready, which might take a number of years. However, it also means that each browser supports a slightly different subset of the HTML5 standard, which spells trouble for Web developers aiming to address users with a uniform platform.

Fortunately, there are ways to check with the current browser to see whether it supports each of the features offered by HTML5. However, you probably want to be able to concentrate on developing your application, rather than creating useful and reliable tests. The open-source Modernizr JavaScript framework is simple to download and install (since it's a single .js file), and it allows you to query the browser from within your program, checking to see what functionality is there. For example, if you want to know whether geolocation is supported, you can say:

```
if (Modernizr.geolocation) {  
    navigator.geolocation.getCurrentPosition(handle_position);  
}  
else  
{  
    alert("Ack! I have no way of knowing where you are!");  
}
```

Although Modernizr can be a terrific help in identifying what

features are available, it doesn't solve the real problem—namely, gracefully handling the lack of such features. I realize Modernizr isn't designed to take on such responsibility, but perhaps someone in the jQuery community (or elsewhere) will create a library (post-Modernizr?) that goes one step beyond Modernizr, allowing us to paper over the differences between browsers, much as Prototype and jQuery did for basic JavaScript functionality several years ago.

Conclusion

HTML5 is coming, and some would say it's already here. If you are creating a Web application, you will do both yourself and your users a big favor by using HTML5. However, doing so does raise questions about which features you can and will include, and which browsers you intend to support. We've seen how Modernizr can help smooth over these differences and keep the Web a universal medium, but doing so will take a bit of work. Next month, I'll look at some features aimed at making the Web a more complete application framework, namely Web sockets (for interprocess communication), workers (for background threading) and local storage. ■

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

Small, Portable Devices with Ubuntu Linux

Small Form Factor Intel® Atom™ Platform

No fans, no moving parts. Just quiet, reliable operation.
Incredibly compact and full featured; no compromises.



VESA-Mountable NVIDIA® ION/ION2 System

Compact, lightweight system with GeForce® Graphics.
Flexible storage options (dual HDD support) and WiFi.

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](http://www.logicsupply.com/linux)

LOGIC
SUPPLY



DAVE TAYLOR

Calculating Mortgage Rates

Shell scripting math with bc.

I used to work at Hewlett-Packard years ago (pre-Fiorina), and back then, one of our mantras was the “next bench syndrome”. You’ve probably heard of it: build things that you’d want to use. It’s logical, and although often we aren’t in that position, it’s certainly the basis of many great hobby projects.

It’s also the basis for this month’s script project. As I write this column, the house I’m in is “under contract” to sell, and I have an offer in on a new place that’s four miles away and lots bigger—more room for those servers, natch! You can’t talk about buying a house, townhouse, condo or even apartment without talking about getting a loan from a friendly bank—at least, I don’t have a spare half-mil burning a hole in my bank account!

When looking at houses, the key question always becomes “how much can I afford?”, and the variables that go into that equation are the amount of your down payment, the duration of the loan, the interest rate the bank is going to charge you and, of course, the base cost of the house you’re considering.

There are more factors, including mortgage insurance if your down payment is less than 20% of the house price, taxes, points, closing costs and so on, but we’re going to skip those because, well, they’re bafflingly complex and way beyond the scope of this column—or magazine!

The basic calculation we need is rather complicated:

$$M = P [i((1 + i)**n)] / [((1 + i)**n) - 1]$$

In this calculation, M is the required monthly payment; n is the total number of monthly payments (on a 30-year mortgage, it’s 30 * 12); P is the principal/amount of the loan itself, and i is the annual interest rate divided by 12 for a monthly interest rate.

So, let’s say the current mortgage rate at your bank is 5.00% APR. That means i would be 5/12 = 0.42.

Math in a Shell Script

How does this convert into a shell script? Surprisingly, it’s not too difficult to duplicate the formula if we tap the power of bc and do some intermediate calculations to make it easy to work with and ensure we’ve no mathematical errors.

The real trick is to make sure we are getting sufficient precision from the calculations, so we

don’t see egregious rounding errors, which multiply dramatically with this sort of formula.

To accomplish that, each time I call bc, I’ll specify scale=6, which gives six post-decimal digits of precision. That means the 5/12 calculation above would be calculated more correctly as 0.426667.

The script starts by having a default duration of 30 years and a default interest rate of 4.85% APR, which is, at the time of this writing, a typical rate. Here’s how I encode those:

```
dfltduration=30 # 30 year loan
dfltint=4.85 # 4.85% APR is the default
pmts=$(( $dfltduration * 12 ))
```

The script will expect two variables to be specified, although only one is required, principal and actual interest rate:

```
princ=$1 ; int=$2
```

That’s a lazy way to assign those variables because, as should be immediately obvious, if they’re not specified, we now have empty variables. My general approach to writing scripts is always to start with the core functionality, then graft on all the error tests and friendly errors. I bet you’re the same.

This gets a bit tricky because we need to normalize percentages; otherwise, users will assume they’re entering 5.25%, and the formula will calculate payments based on 525% (for example, 5.25 not 0.0525). I wrap it all up in one:

```
if [ -z "$int" ] ; then
    int="$(echo "scale=6; $dfltint / (100*12)" | bc -q)"
else
    int="$(echo "scale=6; $int / (100*12)" | bc -q)"
fi
```

That’s the setup. Now the calculations can be configured as a variable prior to pushing it at bc, for ease of debugging:

```
calculation="$princ * ( $int * ((1 + $int) ^ $pmts) ) /
↳( ((1 + $int) ^ $pmts) - 1)"
```

Before I go further, let’s look at an example. Say

I want a loan for a \$300,000 house at 4.85% APR for a typical 30-year mortgage. Here's what the above formula actually looks like:

```
300000 * ( .004041 * ((1 + .004041) ^ 360) ) /  
( ((1 + .004041) ^ 360) - 1)
```

Hopefully, all of those numbers make sense given my explanation so far.

The last step is simply to run the calculation by feeding it to bc:

```
payment="$(echo "scale=6; $calculation" | bc -q)"
```

This still is not quite right because the resultant value ends up looking like 1582.859358 / month, which is a bit confusing when we're used to two digits after the decimal point!

I could tweak that within bc, but changing the scale has the bad side effect of reducing the accuracy of all calculations. Instead, here's a simple tweak that I've shown before and like:

```
payment="$(echo "$payment" | rev | cut -c5- | rev)"
```

I'll let you figure out what that does. Finally, the output:

```
echo Payments: \$$payment/month for $pmts payments to pay off $princ
```

Let's run a few calculations given this functional script:

```
$ mortgage-calc.sh 300000  
Payments: $1582.85/month for 360 payments to pay off 300000
```

```
$ mortgage-calc.sh 300000 7  
Payments: $1995.78/month for 360 payments to pay off 300000
```

```
$ mortgage-calc.sh 500000  
Payments: $2638.09/month for 360 payments to pay off 500000
```

You can see that even a few points of interest (7% instead of 4.85% makes a dramatic difference in those monthly payments).

Now, to find a really nice house to buy! ■

Dave Taylor has been hacking shell scripts for a really long time, 30 years. 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.



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



**Powerful.
Intelligent.**



As the Chief Financial Officer for Silicon Mechanics, Steve is an Expert where value is concerned. That's why he's pictured here with the Rackform iServ R143.

The R143 is a flexible and affordable 1U server. It features an Intel® Xeon® Processor 3400 Series, with powerful features like Turbo Boost, Hyper-Threading, and DDR3 memory. This processor is also available in a low-voltage version, which can optimize power usage and help contain energy costs. With 6 DDR3 DIMM sockets, 2 Gigabit Ethernet adapters, a PCIe expansion slot, and 4 hot-swap SAS/SATA drive bays, the R143 can handle a lot more than entry-level workloads. With a price that starts around \$1250, you don't have to be a CFO to understand the value.

When you partner with Silicon Mechanics, you get more than a flexible, affordable entry-level server — you get an Expert like Steve.

For more information about the
Rackform iServ R143
visit www.siliconmechanics.com/R143

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.



MICK BAUER

Building a Transparent Firewall with Linux, Part V

Build a transparent firewall using an ordinary PC.

Dear readers, I appear to have set a Paranoid Penguin record—six months spent on one article series. (It has consisted of five installments, with a one-month break between the second and third pieces.) But, we’ve covered a lot of ground: transparent firewall concepts and design principles; how to install OpenWrt on a Linksys WRT54GL router; how to compile a custom OpenWrt system image; how to configure networking and iptables bridging on OpenWrt; and, of course, how to replace the native OpenWrt firewall script with a customized iptables script that works in bridging mode. This month, I conclude the series by showing how to achieve the same thing using an ordinary PC running Ubuntu 10.04.

Hardware Considerations

At this late stage in the series, I assume you know what a transparent firewall is and where you might want to deploy it on your network. But since I haven’t said a word about PC hardware since Part II (in the September 2010 issue of *LJ*), it’s worth repeating a couple points I made then about selecting network hardware, especially on laptops.

If it were ten years ago, I’d be talking about internal PCI network adapters for desktop/tower

I love it when a cheap, simple device not only “just works” under Linux, but also performs well, don’t you?

systems and PCMCIA (PC-card) interfaces for laptops. Nowadays, your system almost certainly has an Ethernet card built in and needs only one more to act as a firewall (unless you want a third “DMZ” network, but that’s beyond the scope of this series—I’m assuming you’re firewalling off just one part of your network).

If you have a desktop or tower system with a free PCI slot, you’ve got a plethora of good choices for Linux-compatible Ethernet cards. But, if you have a laptop, or if your PCI slots are all populated, you’ll want an external USB Ethernet interface.

Here’s the part I mentioned earlier: be sure to

select a USB Ethernet interface that supports USB 2.0, because USB 1.1 runs at only 12Mbps and USB 1.0 at 1.5Mbps. (USB 2.0 runs at 480Mbps—plenty fast unless your LAN runs Gigabit Ethernet.) Obviously, you also want an interface that is supported under Linux.

As a general rule, I don’t like to shill specific products, but in the course of writing these articles, I had excellent experiences with the D-Link DUB-E100, a USB 2.0, Fast Ethernet (100Mbps) interface. It’s supported under Linux by the `usbnet` and `asix` kernel modules. Chances are, your Linux system automatically will detect a DUB-E100 interface and load both modules. I love it when a cheap, simple device not only “just works” under Linux, but also performs well, don’t you?

Configuring Ethernet Bridging

You’ll remember from the previous two installments that in order to support iptables in bridging mode, your Linux kernel needs to be compiled with `CONFIG_BRIDGE_NETFILTER=1`, and your `/etc/sysctl.conf` file either needs to *not* contain any entries for the following settings or have them set to “1”:

```
net.bridge.bridge-nf-call-arptables=0
net.bridge.bridge-nf-call-ip6tables=0
net.bridge.bridge-nf-call-iptables=0
```

Well, if you’re an Ubuntu user, you don’t have to worry. Unlike OpenWrt, the stock Ubuntu kernels already have `CONFIG_BRIDGE_NETFILTER` support compiled in, and its default `/etc/sysctl.conf` file is just fine without needing any editing by you. Odds are, this is true for Debian and other Debian derivatives as well, although I haven’t had a chance to verify it myself.

One thing you probably *will* have to do, however, is install the command `brctl` by way of either Debian’s/Ubuntu’s `bridge-utils` package or whatever package through which your non-Debian-derived distribution of choice provides the `brctl` command. This is seldom a default package, so if entering the command which `brctl` doesn’t yield a path to the `brctl` command, you need to install it.

As with OpenWrt, however, you will not need the `ebtables` (Ethernet Bridging tables) command, unless you want to filter network traffic based on

Networking Tips: GNOME vs. You

Normally, any computer you're configuring to act as a network device or server should not run the X Window System for reasons of performance and security. But if, for some reason, such as testing, you want to set up bridging on Ubuntu 10.04 Desktop (or any other GNOME-based distribution), you need to be aware of a few things.

Traditionally, Ubuntu and other Debian derivatives store network interface configurations in the file `/etc/network/interfaces`. However, GNOME's Network Manager system automatically configures any interface not explicitly described in that file.

In theory, this should mean that if you specify interface and bridge configurations in `/etc/network/interfaces`, you shouldn't have to worry about Network Manager overriding or otherwise conflicting with those settings. But in practice, at least in my own experience on Ubuntu 10.04, you're better off *disabling* Network Manager altogether in the System→Preferences→Startup Applications applet, if you want to set up persistent bridge settings in `/etc/network/interfaces`.

To *completely* disable Network Manager, you also need to open the System→Preferences→Network Connections control panel and delete all connection profiles under the Wired tab. Even if Network Manager is disabled as a startup service, Ubuntu will read network configuration information set by this control panel, resulting in strange interactions with `/etc/network/interfaces`.

On my test system, even after disabling the Network Manager service, setting up `/etc/network/interfaces` as shown in Listing 1 and stopping and restarting `/etc/init.d/networking`, `eth2` kept showing up in my routing table with the *same IP address* as `br0`, even though `br0` should have been the only interface with *any* IP address (let alone a route). Clearing out `eth2`'s entry in Network Connections and again restarting networking fixed the problem.

To kill the running Network Manager processes, first find its process ID using `ps auxw | grep nm-applet`. Then, do `sudo kill -9 [PID]` (substituting [PID] with the process ID, of course) to shut down Network Manager. This is a good point to configure networking manually by editing `/etc/network/interfaces` (`sudo vi /etc/network/interfaces`). Finally, restart networking by entering `sudo /etc/init.d/networking restart`.

Ethernet header information, such as MAC (hardware) address and other very low-level criteria. Nothing I describe in this series requires `iptables`, just plain-old `iptables`.

If you've got two viable Ethernet interfaces, if your kernel supports `iptables` in bridging mode, and if your system has `bridge-utils` installed, you're ready to set up bridging! On Ubuntu Server and other Debian-derived, nongraphical systems, this involves changes to only one file, `/etc/network/interfaces`—unless, that is, your window manager controls networking. See the sidebar Networking Tips: GNOME vs. You for instructions on disabling GNOME's Network Manager system.

So, let's examine a network configuration for bridged `eth1` and `eth2` interfaces. (To you fans of Fedora, Red Hat, SUSE and other non-Debian-ish distributions, I apologize for my recent Ubuntu-centrism. But hopefully, what follows here gives you the *gist* of what you need to do within your respective distribution's

manual-network-configuration schemes.)

Listing 1 shows my Ubuntu 10.04 firewall's `/etc/network/interfaces` file. My test system is actually an Ubuntu 10.04 Desktop system, but I've disabled Network Manager as described in the sidebar.

The first part of Listing 1 shows settings for `lo`, a virtual network interface used by local processes to communicate with each other. I've explicitly assigned `lo` its customary IP address `127.0.0.1` and subnet mask `255.0.0.0`.

The rest of Listing 1 gives the configuration for `br0`, my logical bridge interface. First, I set the bridge interface's IP address to `10.0.0.253` with a netmask of `255.255.255.0`, just as I did with `OpenWrt`. Note that when you associate physical network interfaces with a logical bridge interface, the bridge interface gets an IP address, but the physical interfaces do *not*. They are, at that point, just ports on a bridge.

Note that on my test system, `eth1` and `eth2` are

Listing 1. /etc/network/interfaces

```

auto lo
iface lo inet loopback
address 127.0.0.1
netmask 255.0.0.0

auto br0
iface br0 inet static
address 10.0.0.253
netmask 255.255.255.0
pre-up ifconfig eth1 down
pre-up ifconfig eth2 down
pre-up brctl addbr br0
pre-up brctl addif br0 eth1
pre-up brctl addif br0 eth2
pre-up ifconfig eth1 0.0.0.0
pre-up ifconfig eth2 0.0.0.0
post-down ifconfig eth1 down
post-down ifconfig eth2 down
post-down ifconfig br0 down
post-down brctl delif br0 eth1
post-down brctl delif br0 eth2
post-down brctl delbr br0

```

the names assigned to my two USB D-Link DUB-E100 interfaces. It's actually more likely you'd use your machine's built-in Ethernet interface (probably named eth0), and that any second interface you'd add would be named eth1. When in doubt, run the command `tail -f /var/log/messages` before attaching your second interface to see what name your system assigns to it. You also can type `sudo`

If you've got two viable Ethernet interfaces, if your kernel supports iptables in bridging mode, and if your system has bridge-utils installed, you're ready to set up bridging!

`ifconfig -a` to get detailed information on all network interfaces present, even ones that are down.

Continuing the analysis of Listing 1, after I configure the bridge IP address and netmask, I actually bring *down* the two physical interfaces I'm going to bridge, before invoking the `brctl` command to create the bridge (br0) and add each interface (eth1 and eth2) to it. The last step in bringing the bridge up is to assign to both physical interfaces, eth1 and eth2, the reserved address 0.0.0.0, which has the effect of allowing each of those interfaces to receive *any* packet that reaches it—which is to say,

having an interface listen on IP address 0.0.0.0 makes that interface promiscuous. This is a necessary behavior of switch ports. It does *not* mean all packets entering on one port will be forwarded to some other port automatically; it merely means that all packets entering that port will be read and processed by the kernel.

The "post-down" statements in Listing 1, obviously enough, all concern breaking down the bridge cleanly on shutdown.

Once you've restarted networking with a `sudo /etc/init.d/networking restart`, your system should begin bridging between its two physical interfaces. You should test this by connecting one interface on your Linux bridge/firewall to your Internet-connected LAN and connecting the other interface to some test system. The test system shouldn't have any problem connecting through to your LAN and reaching the Internet, as though there were no Linux bridge in between—at least, not yet it shouldn't. But, we'll take care of that!

Configuring iptables in Bridging Mode

Now it's time to configure the Linux bridge with the same firewall policy I implemented under OpenWrt. Listing 2 shows last month's custom iptables script, adapted for use as an Ubuntu init script. (Actually, we're going to run it from the new "upstart" system rather than `init`, but more on that shortly.) Space doesn't permit a detailed walk-through of this script, but the heart of Listing 2 is the "do_start" routine, which sets all three default chains (INPUT, FORWARD and OUTPUT) to a default DROP policy and loads the firewall rules. The example rule set enforces this policy:

- Hosts on the local LAN may send DHCP requests through the firewall and receive their replies.
- Hosts on the local LAN may connect to the firewall using Secure Shell.
- Only the local Web proxy may send HTTP/HTTPS requests and receive their replies.
- Hosts on the local LAN may send DNS requests through the firewall and receive their replies.

This policy assumes that the network's DHCP and DNS servers are on the other side of the firewall from the LAN clients, but that its Web proxy is on the same side of the firewall as those clients.

You may recall that with OpenWrt, the state-tracking module that allows the kernel to track tcp and even some udp applications by transaction state, rather than one packet at a time, induces a significant performance hit. Although that's almost certainly not so big

an issue on a PC-based firewall that has enough RAM and a fast enough CPU, I'm going to leave it to you to figure out how to add state tracking to the script in

Listing 2; it isn't difficult at all!

I have, however, added some lines at the end of the "do_start" routine to log all dropped packets.

Listing 2. Custom iptables Startup Script

```
#!/bin/sh
### BEGIN INIT INFO
# Provides:          iptables_custom
# Required-Start:   $networking
# Required-Stop:
# Default-Start:
# Default-Stop:    0 6
# Short-Description: Custom bridged iptables rules
### END INIT INFO

PATH=/sbin:/bin
IPTABLES=/sbin/iptables
LOCALIP=10.0.0.253
LOCALLAN=10.0.0.0/24
WEBPROXY=10.0.0.111

. /lib/lsb/init-functions

do_start () {
    log_action_msg "Loading custom bridged iptables rules"

    # Flush active rules, custom tables
    $IPTABLES --flush
    $IPTABLES --delete-chain

    # Set default-deny policies for all three default tables
    $IPTABLES -P INPUT DROP
    $IPTABLES -P FORWARD DROP
    $IPTABLES -P OUTPUT DROP

    # Don't restrict loopback (local process intercommunication)
    $IPTABLES -A INPUT -i lo -j ACCEPT
    $IPTABLES -A OUTPUT -o lo -j ACCEPT

    # Block attempts at spoofed loopback traffic
    $IPTABLES -A INPUT -s $LOCALIP -j DROP

    # pass DHCP queries and responses
    $IPTABLES -A FORWARD -p udp --sport 68 --dport 67 -j ACCEPT
    $IPTABLES -A FORWARD -p udp --sport 67 --dport 68 -j ACCEPT

    # Allow SSH to firewall from the local LAN
    $IPTABLES -A INPUT -p tcp -s $LOCALLAN --dport 22 -j ACCEPT
    $IPTABLES -A OUTPUT -p tcp --sport 22 -j ACCEPT

    # pass HTTP and HTTPS traffic only to/from the web proxy
    $IPTABLES -A FORWARD -p tcp -s $WEBPROXY --dport 80 -j ACCEPT
    $IPTABLES -A FORWARD -p tcp --sport 80 -d $WEBPROXY -j ACCEPT
    $IPTABLES -A FORWARD -p tcp -s $WEBPROXY --dport 443 -j ACCEPT
    $IPTABLES -A FORWARD -p tcp --sport 443 -d $WEBPROXY -j ACCEPT

    # pass DNS queries and their replies
    $IPTABLES -A FORWARD -p udp -s $LOCALLAN --dport 53 -j ACCEPT
    $IPTABLES -A FORWARD -p tcp -s $LOCALLAN --dport 53 -j ACCEPT
    $IPTABLES -A FORWARD -p udp --sport 53 -d $LOCALLAN -j ACCEPT
    $IPTABLES -A FORWARD -p tcp --sport 53 -d $LOCALLAN -j ACCEPT

    # cleanup-rules
    $IPTABLES -A INPUT -j LOG --log-prefix "Dropped by default
    =>(INPUT):"
    $IPTABLES -A INPUT -j DROP
    $IPTABLES -A OUTPUT -j LOG --log-prefix "Dropped by default
    =>(OUTPUT):"
    $IPTABLES -A OUTPUT -j DROP
    $IPTABLES -A FORWARD -j LOG --log-prefix "Dropped by default
    =>(FORWARD):"
    $IPTABLES -A FORWARD -j DROP
}

do_unload () {
    $IPTABLES --flush
    $IPTABLES -P INPUT ACCEPT
    $IPTABLES -P FORWARD ACCEPT
    $IPTABLES -P OUTPUT ACCEPT
}

case "$1" in
    start)
        do_start
        ;;
    restart|reload|force-reload)
        echo "Reloading bridging iptables rules"
        do_unload
        do_start
        ;;
    stop)
        echo "DANGER: Unloading firewall's Packet Filters!"
        do_unload
        ;;
    *)
        echo "Usage: $0 start|stop|restart" >&2
        exit 3
        ;;
esac
```

Listing 3. Upstart Configuration File for iptables_custom

```
# iptables_custom

description    "iptables_custom"
author        "Mick Bauer <mick@wiremonkeys.org>"

start on (starting network-interface
         or starting network-manager
         or starting networking)

stop on runlevel [!023456]

console output

pre-start exec /etc/init.d/iptables_custom start
post-stop exec /etc/init.d/iptables_custom stop
```

Although logging on OpenWrt is *especially* problematic due to the limited virtual disk capacity on the routers on which it runs, this is just too important a feature to leave out on a proper PC-based firewall. On most Linux systems, firewall events are logged to the file `/var/log/messages`, but if you can't find any there, they instead may be written to `/var/log/kernel` or some other file under `/var/log`.

Enabling the Firewall Script

As you may be aware, Ubuntu has adopted a new startup script system. The old one, the `init` system, still works, and if you prefer, you can enable the

Although logging on OpenWrt is *especially* problematic due to the limited virtual disk capacity on the routers on which it runs, this is just too important a feature to leave out on a proper PC-based firewall.

script in Listing 2 the old-school way by making it executable and creating `rc.d` links by running this command:

```
bash-$ sudo update-rc.d iptables_custom start 36 2 3 4 5 .
      =>stop 98 0 1 6
```

However, I recommend you take the plunge into the world of the newer "upstart" system by skipping `update-rc.d` and instead adding the following script, `iptables_custom.conf` (Listing 3),

to `/etc/init` (*not* `/etc/init.d`).

Rather than requiring you to figure out which start/stop number to assign to your "rc." links, upstart lets you just specify what needs to start beforehand (in this example: `network-interface`, `network-manager` or `networking`). As you can see, this `iptables_custom.conf` file then invokes `/etc/init.d/iptables_custom`, as listed in Listing 2, to do the actual work of loading or unloading rules. For that reason, `/etc/init.d/iptables_custom` must be executable whether you use it as an `init` script or an upstart job.

After saving your `/etc/init/iptables_custom.conf` file, you must enable it with this command:

```
bash-$ sudo initctl reload-configuration
```

Now you either can reboot or enter this command to load the firewall rules:

```
bash-$ sudo initctl start iptables_custom
```

Conclusion

And that, in one easy procedure, is how to create a bridging firewall using a Linux PC! I hope I've explained all of this clearly enough for you to figure out how to make it meet your specific needs. I also hope you found the previous few months' foray into OpenWrt to be worthwhile.

The Paranoid Penguin will return in a couple months, after I've had a short break. In the meantime, go forth and secure things! ■

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

Peter de Jong's iptables script for the upstart `init` system is available at 4pdj.nl/2010/01/11/custom-firewall-under-ubuntu-karmic-koala-with-upstart.

See also my book: Bauer, Michael D. *Linux Server Security*, second edition. Sebastopol, California: O'Reilly Media, 2005. Chapter 3 explains iptables in detail, and Appendix A contains two complete iptables scripts. Although focused on "local" ("personal") firewalls rather than Internet or LAN firewalls, this material nonetheless should be helpful to iptables beginners.

SOUTHWEST DRUPAL SUMMIT

HOUSTON, TX

JANUARY 27-28



2 DAYS
OF DRUPAL
DEVELOPMENT
BUSINESS &
DESIGN

REGISTER NOW!

WWW.SWDRUPALSUMMIT.COM

MAGNOLIA HOTEL ★ 1100 TEXAS AVE ★ HOUSTON TX 77002

See what the **Drupal Content Management System and Framework** can do for you. The event brings Drupal experts, novices, and business leaders together to share successes, explore opportunities, and learn more about why and how Drupal is making headlines across the world as a superior enterprise-level web application platform.

REGISTER
EARLY!
VISIT US
ONLINE



WHEN IS IT
JANUARY 27-28, 2011

WHERE TO GO
MAGNOLIA HOTEL - HOUSTON, TEXAS

HOW TO BE A SPONSOR
CONTACT JHAYS@NEOSPIRE.NET OR
KATHERINE@LINUXJOURNAL.COM

KEYNOTE SPEAKER
ANGELA BYRON, LULLABOT.COM

Angela Byron is the Drupal 7 core maintainer, recipient of the 2008 Google- O'Reilly Open Source Award for Best Contributor, and an Open Source evangelist who lives and breathes Drupal. We are so excited to welcome her to Houston!



SPONSORED BY

NEOSPIRE **LINUX**
MANAGED HOSTING JOURNAL

 Lullabot™



SPECIAL HOTEL GROUP RATE: \$139/NIGHT
CALL 1.888.915.1110 AND ASK FOR DRUPAL CONFERENCE GROUP
EARLY REGISTRATION IS AVAILABLE



KYLE RANKIN

Bond, Ethernet Bond

Configure Ethernet bonding and get a license to kill a network interface without any downtime.

As a sysadmin, one of the most important virtues you should foster is tolerance. Now, although it's important for vi and Emacs users and GNOME and KDE users to live in harmony, what I want to focus on here is *fault* tolerance. Everybody has faults, but when your server's uptime is on the line, you should do everything you can to make sure it can survive power faults, network faults, hardware faults and even your faults. In this column, I describe a basic fault-tolerance procedure that's so simple to implement, all sysadmins should add it to their servers: Ethernet bonding.

Live and Let Interfaces Die

The basic idea behind Ethernet bonding is to combine two or more Ethernet ports on your machine, such that if one Ethernet port loses its connection, another bonded port can take over the traffic with zero or minimal downtime. The fact is that these days, the bulk of the services on a server require a network connection to be

On top of basic fault tolerance, you also can use certain Ethernet bonding modes to provide load balancing as well and get more bandwidth than a single interface could provide.

useful, so if you set up multiple Ethernet ports that are connected to redundant switches, you can conceivably survive a NIC failure, a cable failure, a bad switch port or even a switch failure, and your server will stay up.

On top of basic fault tolerance, you also can use certain Ethernet bonding modes to provide load balancing as well and get more bandwidth than a single interface could provide. Ethernet bonding is a feature that is part of the Linux kernel, and it can provide a number of different behaviors based on which bonding mode you choose. All of the Ethernet bonding information can be found in the Documentation/networking/bonding.txt file included with the Linux kernel source. Below, I provide an excerpt from that documentation that lists the 007 bonding modes:

- **balance-rr** or 0 — round-robin policy: transmit packets in sequential order from the first available slave through the last. This mode provides load balancing and fault tolerance.

- **active-backup** or 1 — active-backup policy: only one slave in the bond is active. A different slave becomes active if, and only if, the active slave fails. The bond's MAC address is externally visible on only one port (network adapter) to avoid confusing the switch.

- **balance-xor** or 2 — XOR policy: transmit based on the selected transmit hash policy. The default policy is a simple [(source MAC address XOR'd with destination MAC address) modulo slave count]. Alternate transmit policies may be selected via the `xmit_hash_policy` option, described below. This mode provides load balancing and fault tolerance.

- **broadcast** or 3 — broadcast policy: transmits everything on all slave interfaces. This mode provides fault tolerance.

- **802.3ad** or 4 — IEEE 802.3ad dynamic link aggregation: creates aggregation groups that share the same speed and duplex settings. Utilizes all slaves in the active aggregator according to the 802.3ad specification.

- **balance-tlb** or 5 — adaptive transmit load balancing: channel bonding that does not require any special switch support. The outgoing traffic is distributed according to the current load (computed relative to the speed) on each slave. Incoming traffic is received by the current slave. If the receiving slave fails, another slave takes over the MAC address of the failed receiving slave.

- **balance-alb** or 6 — adaptive load balancing: includes `balance-tlb` plus receive load balancing (`rlb`) for IPv4 traffic and does not require any special switch support. The receive load balancing is achieved by ARP negotiation. The bonding driver intercepts the ARP Replies sent by the local system on their way out and overwrites the source hardware address with the unique hardware address of one of the slaves in the bond such that different peers use different hardware addresses for the server.

Now that you've seen all the choices, the real question is which bonding mode should you choose? To be honest, that's a difficult question to answer, because it depends so much on your network and what you want to accomplish. What I recommend is

to set up a test network and simulate a failure by unplugging a cable while you ping the server from another host. What I've found is that different modes handle failure differently, especially in the case of a switch that takes some time to re-enable a port that has been unplugged or a switch that has been rebooted. Depending on the bonding mode you choose, those situations might result in no downtime or a 30-second outage. For my examples in this column, I chose bonding mode 1 because although it provides only fault tolerance, it also has only one port enabled at a time, so it should be relatively safe to experiment with on most switches.

Note: the bonding mode is set at the time the bonding module is loaded, so if you want to experiment with different bonding modes, you will at least have to unload and reload the module or at most reboot the server.

Although Ethernet bonding is accomplished through a kernel module and a utility called `ifenslave`, the method you use to configure kernel module settings and the way networks are configured can differ between Linux distributions. For this column, I talk about how to set this up for both Red Hat- and

Debian-based distributions, as that should cover the broadest range of servers. The first step is to make sure that the `ifenslave` program is installed. Red Hat servers should have this program installed already, but on Debian-based systems, you might need to install the `ifenslave` package.

For Your Files Only

The next step is to configure the bonding module with the bonding mode you want to use, along with any other options you might want to set for that module. On a Red Hat system, you will edit either `/etc/modprobe.conf` (for a 2.6 kernel) or `/etc/modules.conf` (for an older 2.4 kernel). On a Debian-based system, edit or create the `/etc/modprobe.d/aliases` file. In either case, add the following lines:

```
alias bond0 bonding
options bonding mode=1 miimon=100
```

The alias line will associate the `bond0` network interface with the bonding module. If you intend on having multiple bonded interfaces (such as on a system with four or more NICs), you will need to add an extra

Powerful: Rhino



Rhino M6500/E6500

- Dell Precision M6500 w/ Core i7 Quad (8 core)
- Dell Latitude E6500 w/ 2.2-3.0 GHz Core 2 Duo
- Up to 17" WUXGA LCD w/ X@1920x1200
- NVidia Quadro FX 3800M
- 80-500 GB hard drive
- Up to 16 GB RAM (1333 MHz)
- DVD±RW or Blu-ray
- 802.11a/g/n
- Starts at \$1240

- High performance NVidia 3-D on a WUXGA RGB/LED
- High performance Core i7 Quad CPUs, 16 GB RAM
- Ultimate configurability — choose your laptop's features
- One year Linux tech support — phone and email
- Three year manufacturer's on-site warranty
- Choice of pre-installed Linux distribution:



Tablet: Raven



Raven X200 Tablet

- ThinkPad X200 tablet by Lenovo
- 12.1" WXGA w/ X@1280x800
- 1.2-1.86 GHz Core 2 Duo
- Up to 8 GB RAM
- 80-500 GB hard drive / 256 GB SSD
- Pen/stylus input to screen
- Dynamic screen rotation
- Starts at \$2080

Rugged: Tarantula



Tarantula CF-30

- Panasonic Toughbook CF-30
- Fully rugged MIL-SPEC-810F tested: drops, dust, moisture & more
- 13.3" XGA TouchScreen
- 1.6 GHz Core 2 Duo
- Up to 8 GB RAM
- 80-500 GB hard drive
- Call for quote

EmperorLinux

...where Linux & laptops converge

www.EmperorLinux.com

1-888-651-6686



Listing 1. Bond Script for Red Hat Users

```

# bond bond0 eth0 eth1

#!/usr/bin/perl

# bond -- create a bonded interface out of one or
# more physical interfaces
# Created by Kyle Rankin
#

my $bond_interface = shift;
my @interfaces = @ARGV;
my $network_scripts_path = '/etc/sysconfig/network-scripts/';
my $bond_mode=1;
my $bond_miimon=100;
my $bond_max=2;

usage() unless (@ARGV);
if($#interfaces < 1){
    usage("ERROR: You must have at least 2 interfaces to bond!");
}

system("/etc/init.d/network stop");

config_bond_master($bond_interface, $interfaces[0]);
foreach(@interfaces){
    config_bond_slave($bond_interface, $_);
}
config_modules($bond_interface, $bond_miimon, $bond_mode);

system("/etc/init.d/network start") or die
    =>"Couldn't start networking: $!\n";

sub usage
{
    $error = shift;
    print "$error\n" if($error);
    print "Usage: $0 bond_interface interface1 interface2 [...] \n";
    print "\nbond_interface will use the network
    =>settings of interface1\n";
    exit
}

sub config_bond_master
{
    my $bond_interface = shift;
    my $main_interface = shift;
    my $netconfig_ref = get_network_config($main_interface);

    open CONFIG, "> $network_scripts_path/ifcfg-$bond_interface"
    =>or die "Can't open
    =>$network_scripts_path/ifcfg-$bond_interface: $!\n";

    print CONFIG "DEVICE=$bond_interface\n";
    foreach(keys %$netconfig_ref){
        unless($_ eq "HWADDR" || $_ eq "DEVICE"){
            print CONFIG "$_=$netconfig_ref{$_}\n";
        }
    }
    close CONFIG;
}

sub config_bond_slave
{
    my $bond_interface = shift;
    my $slave_interface = shift;
    my $netconfig_ref = get_network_config($slave_interface);

    open CONFIG, "> $network_scripts_path/ifcfg-$slave_interface"
    =>or die "Can't open
    =>$network_scripts_path/ifcfg-$slave_interface: $!\n";

```

alias line for bond1 or any other interfaces. The options line allows me to set my bonding mode to 1 as well as set miimon (how often the kernel will check the link state of the interface in milliseconds).

On Your Distribution's Network Service

Like with module configuration, different distributions handle network configuration quite differently, and that's true for bonded interfaces as well. So, at this point, it's best if I describe each system separately.

From Red Hat with Love

Red Hat network configuration is managed via files under `/etc/sysconfig/network-scripts`. Each interface has its own configuration file preceded by `ifcfg-`, so that the configuration for `eth0` can be found in `/etc/sysconfig/network-scripts/ifcfg-eth0`. To configure bonding, you simply can use the basic network settings you would have for your regular interface, only now they will be found in `ifcfg-bond0`:

```

DEVICE=bond0
NETMASK=255.255.255.0
GATEWAY=192.168.19.1
BOOTPROTO=static
IPADDR=192.168.19.64
HOSTNAME=goldfinger.example.net
ONBOOT=yes

```

Next, each interface you want to use for `bond0` needs to be configured. In my case, if I wanted to bond `eth0` and `eth1`, I would put the following into `ifcfg-eth0`:

```

DEVICE=eth0
USERCTL=no
ONBOOT=yes
MASTER=bond0
SLAVE=yes
BOOTPROTO=none

```

and the following into `ifcfg-eth1`:

```

    print CONFIG <<"EOC";
DEVICE=$slave_interface
USERCTL=no
ONBOOT=yes
MASTER=$bond_interface
SLAVE=yes
BOOTPROTO=none
EOC
    if($$netconfig_ref{'HWADDR'}){
        print CONFIG "HWADDR=$$netconfig_ref{'HWADDR'}";
    }
}

# This subroutine returns a hash with key-value pairs matching
# the network configuration for the interface passed as an
# argument according to the configuration file in
# /etc/sysconfig/network-scripts/ifcfg-interface
sub get_network_config
{
    my $interface = shift;
    my %netconfig;
    open(CONFIG, "$network_scripts_path/ifcfg-$interface")
    or die "Can't open
    =>$network_scripts_path/ifcfg-$interface: $!\n";
    while(<CONFIG>)
    {
        chomp;
        ($key, $value) = split '=';
        $netconfig{uc($key)} = $value;
    }
    close CONFIG;
    return \%netconfig;
}

sub config_modules
{

```

```

    my $bond_interface = shift;
    my $bond_miimon = shift;
    my $bond_mode = shift;
    my $bond_options_configured = 0;
    my $bond_alias_configured = 0;

    if(-f "/etc/modprobe.conf"){ # for 2.6 kernels
        $module_config = "/etc/modprobe.conf";
    }
    else {
        $module_config = "/etc/modules.conf";
    }
    open CONFIG, "$module_config" or die
    =>"Can't open $module_config: $!\n";
    while(<CONFIG>){
        if(/options bonding/){ $bond_options_configured = 1; }
        if(/alias $bond_interface bonding/){
            =>$bond_alias_configured = 1; }
        }
    close CONFIG;

    open CONFIG, ">> $module_config" or die
    =>"Can't open $module_config: $!\n";
    unless($bond_alias_configured)
    {
        print CONFIG "alias $bond_interface bonding\n";
    }
    unless($bond_options_configured)
    {
        print CONFIG "options bonding
        =>miimon=$bond_miimon mode=$bond_mode max_bonds=$bond_max\n";
    }
    close CONFIG;
}
}

```

```

DEVICE=eth1
USERCTL=no
ONBOOT=yes
MASTER=bond0
SLAVE=yes
BOOTPROTO=none

```

Finally, type `service network restart` as root to restart your network service with the new bonded interface. From this point on, you can treat `ifcfg-bond0` as your main configuration file for any network changes (and files like `route-bond0` to configure static routes, for instance). To make this even easier for you, I've included a script for Red Hat users that automates this entire process. Run the script with the name of the bonded interface you want to use (such as `bond0`), follow it with the list of interfaces you want to bond, and it will set up the modules and network interfaces for you automatically based on the configuration it finds in the first interface (such as `eth0`) that you list. So, for instance, to

set up the above configuration, I would make sure that `ifcfg-eth0` had the network settings I wanted to use, and then I would run the script shown in Listing 1.

Debian Is Forever

As you might imagine, Debian's network configuration is quite different from Red Hat's. Unfortunately, I don't have a Perl script to automate the process for Debian users, but as you will see, it's so simple, a script isn't necessary. All the network configuration for Debian-based servers can be found in `/etc/network/interfaces`. Here's a sample interfaces file:

```

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static

```

LINUXTM JOURNAL

Linux News
and Headlines
Delivered
To You



Linux Journal
topical RSS feeds
AVAILABLE

http://www.linuxjournal.com/rss_feeds

Once the bonded interface is enabled, you can ping the server from a remote host and test that it fails over when you unplug one of the Ethernet cables.

```
address 192.168.19.64
netmask 255.255.255.0
gateway 192.168.19.1
```

To configure a bonded interface, I simply comment out all of the configuration settings for eth0 and create a new configuration for bond0 that copies all of eth0's settings. The only change I make is the addition of an extra parameter called slaves that lists which interfaces should be used for this bonded interface:

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
#auto eth0
#iface eth0 inet static
# address 192.168.19.64
# netmask 255.255.255.0
# gateway 192.168.19.1

auto bond0
iface bond0 inet static
address 192.168.19.64
netmask 255.255.255.0
gateway 192.168.19.1
slaves eth0 eth1
```

Once you have made the changes, type `sudo service networking restart` or `sudo /etc/init.d/networking restart` to restart your network interface.

No matter whether you use Red Hat or Debian, once you have configured the bonded interface, you can use the `ifconfig` command to see that it has been configured:

```
$ sudo ifconfig
bond0      Link encap:Ethernet HWaddr 00:0c:29:28:13:3b
           inet addr:192.168.19.64 Bcast:192.168.0.255
           Mask:255.255.255.0
           inet6 addr: fe80::20c:29ff:fe28:133b/64 Scope:Link
           UP BROADCAST RUNNING MASTER MULTICAST MTU:1500 Metric:1
           RX packets:38 errors:0 dropped:0 overruns:0 frame:0
           TX packets:43 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:0
           RX bytes:16644 (16.2 KB) TX bytes:3282 (3.2 KB)

eth0      Link encap:Ethernet HWaddr 00:0c:29:28:13:3b
```

```
UP BROADCAST RUNNING SLAVE MULTICAST MTU:1500 Metric:1
RX packets:37 errors:0 dropped:0 overruns:0 frame:0
TX packets:43 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:16584 (16.1 KB) TX bytes:3282 (3.2 KB)
Interrupt:17 Base address:0x1400
```

```
eth1 Link encap:Ethernet HWaddr 00:0c:29:28:13:3b
UP BROADCAST RUNNING SLAVE MULTICAST MTU:1500 Metric:1
RX packets:1 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:60 (60.0 B) TX bytes:0 (0.0 B)
Interrupt:18 Base address:0x1480
```

```
lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

Once the bonded interface is enabled, you can ping the server from a remote host and test that it fails over when you unplug one of the Ethernet cables. Any failures should be logged both in `dmesg (/var/log/dmesg)` and in the `syslog (/var/log/messages or /var/log/syslog)` and would look something like this:

```
Oct 04 16:43:28 goldfinger kernel: [ 2901.700054] eth0: link down
Oct 04 16:43:29 goldfinger kernel: [ 2901.731190] bonding: bond0:
link status definitely down for interface eth0, disabling it
Oct 04 16:43:29 goldfinger kernel: [ 2901.731300] bonding: bond0:
making interface eth1 the new active one.
```

As I said earlier, I highly recommend you experiment with each bonding mode and with different types of failures, so you can see how each handles both failures and recoveries on your network. When your system is more tolerant of failures, you'll find you are more tolerant of your system. ■

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.



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



**Powerful.
Intelligent.**



Charles heads the web development team here at Silicon Mechanics: he's responsible for the configurators and power calculator on our site, among other things. As a software expert, he offers a useful perspective on our server and storage products.

When asked what he would do if he had a Rackform iServ R413 configured with 4 8-core Intel® Xeon® Processor 7500 Series CPUs and 32 DDR3 DIMMs, he said, "32 virtual machines . . . one per core . . . in one rack unit." But he didn't stop there.

Charles knows that to make the best use of a server with that kind of processing horsepower in a virtualized environment, he needs I/O to match. He paired the 4P server with a Storform iServ R516 storage server, configured with 24 2.5-inch Intel X25-E solid state drives. Think of it as a developer's dream team: multi-core processing and high memory counts for blistering performance, and high-performance storage for blazing I/O speed.

Need a "dream team" of your own? Talk to the Experts at Silicon Mechanics for the perfect match.

When you partner with Silicon Mechanics, you get more than just the power and performance of the latest Intel technologies—you get an Expert like Charles.

For more information about the
Rackform iServ R413
visit www.siliconmechanics.com/R413

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.

Opera Browser

The new version 11 of Opera, the veteran of Web browsers for Linux, took the next step in its evolution by adding extension functionality. Developers can author, upload and share these browser add-ons, which utilize Opera's APIs, as well as Web standards like HTML5 and JavaScript. These standards allow extensions made for other browsers to be tweaked and shared with 50 million Opera desktop users. Previously, extensions were limited to Opera Widgets and Opera Unite applications. Opera checks all extensions before they are made public to ensure that the catalog of extensions is free from defects and malicious software. Opera 11 runs on Linux, Mac OS and Windows.

www.opera.com/browser/next



Linutop OS

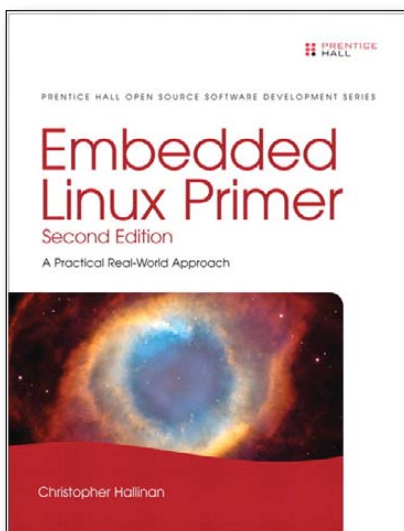
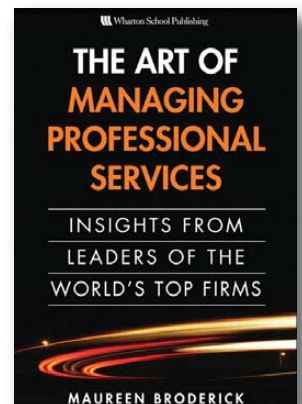
The miserly and miniature Linutop PC just got a brain implant in the form of the new Linutop OS 4.0, a small and secure Ubuntu-based operating system. The OS also works on other x86 PCs. Version 4's key new addition is the kiosk configuration, which allows for quick-and-simple customized configuration—for example, in secured public Internet access, digital signage, network monitoring, thin-client use, mini-servers and low-cost desktops in virtualized environments. Other benefits include a small, 700MB footprint, low power requirements and the ability to back up the entire OS on a USB drive.

www.linutop.com

Maureen Broderick's *The Art of Managing Professional Services* (Wharton School)

Farming out services is something nearly all of us have engaged in at some level. If you think you've got the stuff to break out beyond frantic calls from grandma, you may want to pick up Maureen Broderick's *The Art of Managing Professional Services: Insights from Leaders of the World's Top Firms*. The book is a guide to building and managing a professional service firm. According to Broderick, aspects like infrastructure, governance, talent acquisition and retention, compensation and financial management vary significantly from traditional corporate environments. Furthermore, conventional management advice doesn't offer all the answers, and mainstream business gurus rarely address the unique challenges facing professional service firm leaders. Insights are offered based on 130 in-depth interviews with leaders of the world's top firms.

www.informit.com



Christopher Hallinan's *Embedded Linux Primer* (Prentice Hall)

If embedded Linux is an arrow you want to add to your quiver, take your aim at the new 2nd edition of Christopher Hallinan's popular book *Embedded Linux Primer: A Practical Real-World Approach*. The publisher Prentice Hall bills the title as "the definitive real-world guide to building efficient, high-value, embedded systems with Linux". This new edition has been updated to include the newest Linux kernels, capabilities, tools and hardware support, including advanced multicore processors. Topics covered include kernel configuration and initialization, bootloaders, device drivers, filesystems, BusyBox utilities, real-time configuration and system analysis. This edition adds new content on UDEV, USB and open-source build systems.

www.phptr.com



Joyent's Smart Technology Platform

By releasing its new Smart Technology Platform, a cloud hosting solution for Linux and Windows, Joyent has Amazon's EC2 focused squarely in the crosshairs. Joyent differentiates its product from other cloud platforms by offering "an environment optimized for Web application development" that delivers higher performance in key areas, such as disk and memory I/O, CPU speed and network latency, as well as being "pound-for-pound the most affordable solution on the market for the performance delivered". The Joyent platform also comes bundled with a full set of integrated solutions.

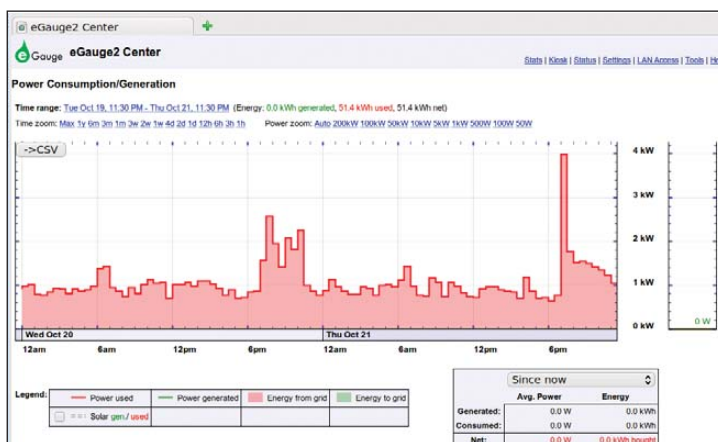
www.joyent.com

SoleraTec's Phoenix RSM Module for OnSSI System

The data protection solution provider SoleraTec has expanded the functionality of its flagship OnSSI video management system with the new Phoenix RSM Module. This new module extracts and replicates video feeds at high speed while maintaining the original video quality and resolution. By replicating video into the Phoenix RSM forensic storage solution, organizations can greatly expand their retention abilities and provide stronger, comprehensive management of all recorded video. Users quickly can search and play back video footage, regardless of when the video was recorded or where it is stored. Organizations can capture and store huge amounts of video surveillance footage that easily can be searched and retrieved for playback. The optional H.264 lossless format reduces storage requirements up to 70% while preserving original image quality, allowing for greatly increased content on the same storage space.

www.SoleraTec.com

SoleraTec™



eGauge Systems' eGauge2

Inside sources at eGauge Systems tipped us off to the fact that their new eGauge2 Web-based electric energy and power meter is powered by Linux. eGauge2, now 40% smaller than its predecessor, is used to measure and record whole-house consumption, renewable (such as solar and wind) energy generation, and individual loads, such as appliances or geothermal system pumps and backup heaters. It can measure up to 12 circuits on up to 3-phases (120V–480V, 50–60Hz). The data can be viewed on any Web-enabled device through the built-in Web-server, and the device records the most recent 30 years of data in its built-in solid-state memory. The measurements also can be accessed through BACnet and/or the recorded data can be shared with Google PowerMeter.

www.egauge.net

Graebert's ARES Commander Edition 2D/3D CAD

Given how good we Linux geeks have been this year, Santa Claus (via his agents at Graebert GmbH) is treating us to a new CAD application, namely ARES Commander Edition Version 1.0 2D/3D CAD. Graebert says that ARES Commander Edition is a powerful and affordable 3D CAD solution that is fully capable of supporting both AEC (Architecture/Engineering/Construction) and MCAD (Mechanical CAD) and the ability to exchange files across all three supported OS platforms, Linux, Mac OS and Windows seamlessly. Users also have a choice of experiencing a fully Linux-specific UI or a more tailored UI that matches the Windows version, both of which are fully command-compatible with AutoCAD. A free 30-day trial is available for download from Graebert's Web site.

www.graebert.com

Gräbert™ CAD ANYWHERE.

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

Fresh from the Labs

SOFA—Statistics Open For All

www.sofastatistics.com

If statistics is your game, and you're chasing an easy-to-use and comprehensive package that outputs great-looking charts, look no further. According to the Web site: "SOFA is a user-friendly statistics, analysis and reporting program. It is free, with an emphasis on ease of use, learn as you go and beautiful output. SOFA lets you display results in an attractive format ready to share."

Installation Binary packages are available for Linux, Windows and Mac (with Linux at the top of the list). Sadly, the Linux binary is only for Ubuntu, but the obligatory source also is available. Ubuntu users can grab the .deb and work things out for themselves, but the source is a bit trickier. At the time of this writing, the installation process was in a state of flux, so project maintainer Grant Paton-Simpson will have some special instructions up at the Web site for *LJ* readers when this article is printed.

As far as library requirements, here's what Grant told me you need:

- python (>= 2.6.2).
- wx-common (>= 2.8.9.2).
- python-wxversion (>= 2.8.9.2).
- python-wxgtk2.8 (>= 2.8.9.2).
- python-numpy (>= 1:1.2.1).
- python-pysqlite2 (>= 1.0.1).
- python-mysqldb (>= 1.2.2).
- python-pygresql (>= 1:4.0).
- python-matplotlib (>= 0.98.5.2).
- python-webkit (>= 1.0.0).

Once the program is installed, you should be able to find SOFA Statistics in your menu; otherwise, you'll need to run it from a terminal. If you need to use the command line, enter:

```
$ python
/usr/share/pyshared/sofa/start.py
```

This path may be different on some distributions, and Grant may have made a link to a bin directory by the time this article is published (meaning you could start SOFA Statistics with a simple one-word command).

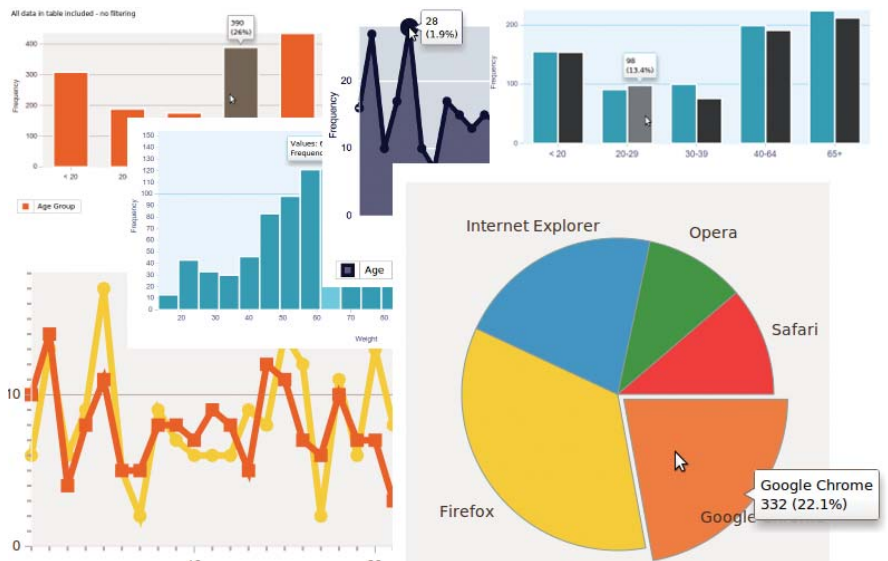
Usage Grant has gone to a lot of effort making some excellent video tutorials, and there's no way I can improve upon them, so instead, I concentrate on highlighting cool features here. Again, Grant appears to be one step ahead of me in that he's provided a default set of preloaded values you can use to explore the project with ease, rather than going through the laborious process of first having to learn how to enter data and then making it display something meaningful. For now, let's look at the three main sections: Report Tables, Charts and Statistics.

Under Report Tables, choose some random settings under Table Type, provide names for Title and Subtitle, and choose some of the available data fields with the Add button. Now click Run, and a swank new table is presented to you. Don't like the aesthetics? No problem. The Style output using... drop-down box lets you change the border to something more pleasing—a nice touch.



SOFA Statistics provides a highly flexible visualization system for analyzing complex data.

The pièce de résistance is probably the Charts section. This is where you can play around with the charts you see here in the screenshots, and more and more chart types are being added over time. Whether you want a bar graph, pie chart, line graph or something like a Scatterplot configuration, chances are it's doable.



A montage of some of SOFA's beautiful graphs and charts, generated on the fly.

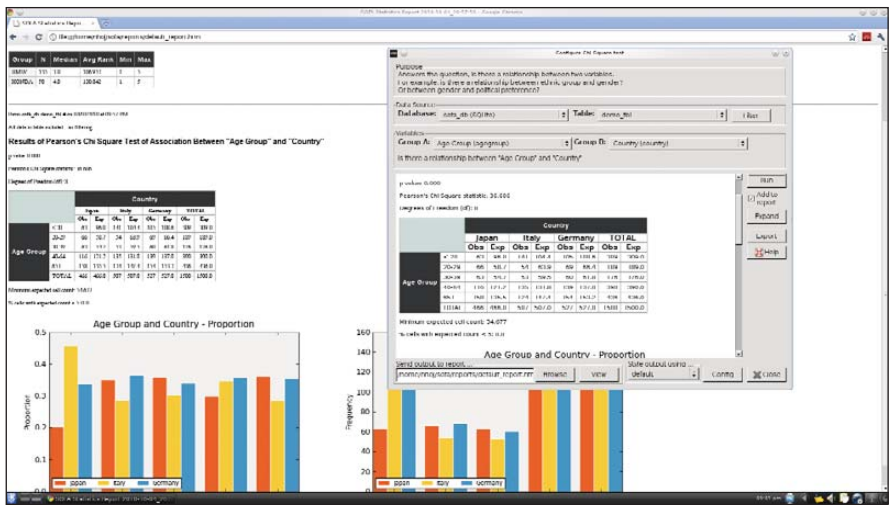


Don't your applications deserve the **BEST** PROTECTION?



Combining **High Availability** with **efficient Data Replication** to ensure **100% Business Continuity** for your **Mission Critical Apps!**

For more information visit <http://us.sios.com/>



My favorite feature of SOFA's is its ability to generate HTML pages of your work dynamically, which can be viewed by anyone with a browser.

Play with some values in the Variables section, choose a Chart Type, click Run and a beautiful chart appears.

The Statistics section is where the elegance of design and data flow really come into play. This section is a bit beyond me, but here you can run statistical tests on your data, with a focus on the kind of tests most users need, most of the time. You can choose from common

opened instantly by anyone (like your coworkers) on their own computers, without needing to install SOFA Statistics. Plus, the information they see will be presented professionally with some impressive graphics to boot.

Although SOFA Statistics is still in its slightly buggy developmental stage, project maintainer Grant Paton-Simpson has shown an impressive grasp of what

Whether you want a bar graph, pie chart, line graph or something like a Scatterplot configuration, chances are it's doable.

tests, such as ANOVA or Chi Square, or run through a check list of choices to choose what's right for you. Click Configure Test on the right, and you'll be presented with the final screen.

From here, you can choose which variables and groupings you want to test against. And finally, click Run. This section gives you the most impressive of the readouts and provides a comprehensive bundle of tables and graphs of analyzed statistics.

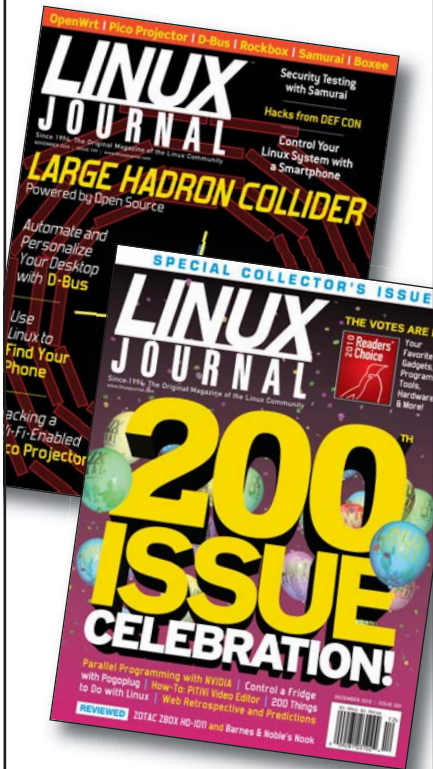
However, one of the most impressive and practical features under all three of these main sections is the Send output to... feature, with its View button. Here you actually can view each page of output in any Web browser in HTML format. This gives the project some instant credibility and practicality in that any work you do in SOFA can be

needs to be included in SOFA, from the small touches to the big. My hope is that this program becomes an adopted industry standard of sorts, mentioned in everyday conversation by organization workers the world over. And, given its free and multiplatform nature, combined with a very canny coder and designer, this hope of mine may not be an unrealistic one.

Cube Escape—Mind-Bending 3-D Mazing

code.google.com/p/cube-escape

Fans of unique puzzle games should check out *Cube Escape*—a really interesting variation on the traditional maze games you've come to expect. According to the Web site: "You are inside a cube made up of numerous shells, with a maze

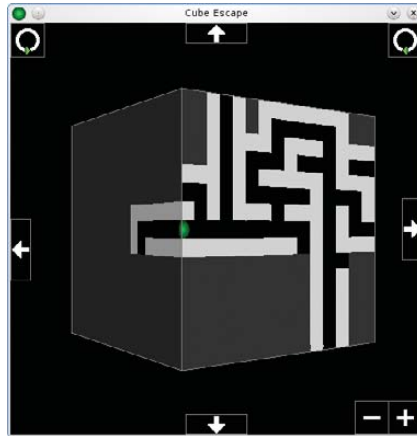


LINUX JOURNAL™

Since 1994: The Original Magazine of the Linux Community

SUBSCRIBE TODAY!

WWW.LINUXJOURNAL.COM/SUBSCRIBE

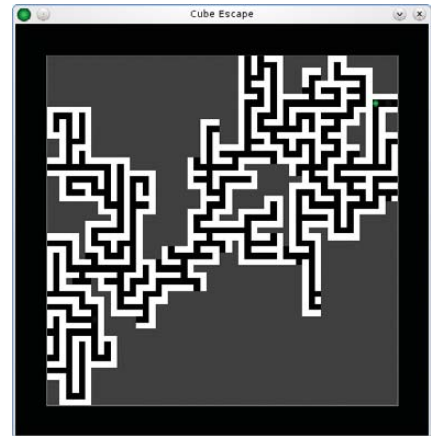


Cube Escape is a brain-melting maze game that takes place on a cube, where you escape from multiple layers in the process.

etched on the surface of each shell.

Escape the cube by traveling through the mazes, including over cube edges, until you find an upward exit home. When you reach the red exit, you win.”

Installation Running this game actually is very easy, as you don't have to compile it (assuming you're on an Intel x86 architecture). If you're not on an Intel-based distro, fear not; compilation is also very easy. Although binaries are available at places such as playdeb.net (and you can work out how to install those yourself), the source tarball is so easy that I'll just run with that.



Later levels can become seriously complex, making 3-D navigation incredibly difficult.

```
$ ./cube-escape
```

If you're running on another architecture, simply delete the current `cube-escape` file and enter this command:

```
$ make
```

Now you can run the game with the same command as above.

Usage Inside the game, the controls and game play are pretty simple. Start a new game, and using the default settings for now, click OK and the game starts.

You'll be presented with a small white

Fans of unique puzzle games should check out *Cube Escape*—a really interesting variation on the traditional maze games you've come to expect.

As far as library requirements go, the documentation says you need the following

- **SDL** (www.libsdl.org/download-1.2.php).
- **SDL_ttf** (www.libsdl.org/projects/SDL_ttf).
- **SDL_image** (www.libsdl.org/projects/SDL_image).

Grab the latest tarball from the Web site and extract it. Open a terminal in the new folder, and if you're on an x86 machine (including AMD64 and the like), run the program with the following command:

box, with the player represented as a green ball inside a black pathway. Basic controls consist of the arrow keys for movement, and the Enter key engages the colored portals to ascend and descend between levels, as well as the red portals that finish the game.

At this point, I recommend right-clicking in the black space outside the maze. A set of controls appears (which can be disabled again by another right-click) that control your view of the cubic maze. Currently, you are looking at one side of the cube, but click any of the arrows on the top, bottom, left and right of the window, and you can flip the cube around, exploring all six sides

of the cube before moving on. This is handy for checking which direction your needed portals are on, so I highly recommend you do so!

If you look in the corners of your game window, at the bottom right of the screen are some zoom controls. At the top left and top right are rotational controls, so you actually can rotate the view of the cube, instead of just changing between cube faces.

As far as the actual gameplay goes, you start on some very basic levels with little detail, zoomed in quite closely to your character. Find your way to the green portal, and you'll ascend to the next layer. At the top layer is the red portal to finish, and the blue portals let you descend layers (I'm not sure why you would though, unless other gameplay mechanics are in the pipeline).

You'll notice gray sections on the cube. These are the unexplored areas, and they light up and reveal bits of maze the more you explore, staying that way if you're heading back (the game would be very hard without this gameplay mechanic

as you'd keep covering old ground).

Once you've come to grips with the game, you may want to increase the difficulty. When you start a new game, the Options screen has a number of variables you can change, such as how many levels you want, which level to start on, how far the exit distance is from the starting portal and so on.

Although the gameplay of *Cube Escape* will speak for itself with any genuine geek (myself included), OSS projects have a habit of evolving into something bigger, and what I'm really looking forward to are the mutations that inevitably will take place.

The game may take place on a 3-D cube, but most of the time, this 3-D world isn't readily apparent. If you turn off the Advanced Graphics option with its cube flipping, you would realize the game takes place on a cube only after hours of playing time. I know it's shallow, but if some whiz-kid OpenGL programmer used some perspective tricks to show something like a cube floating in space, with some graphical hints toward the game taking

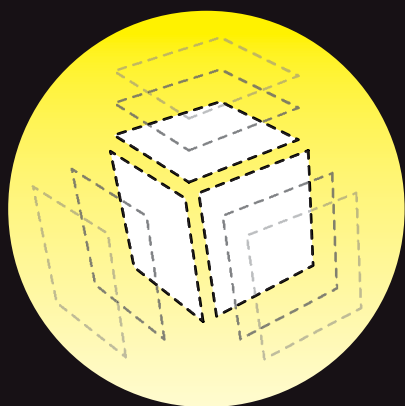
place over multiple layers, the appeal of the game would become readily apparent to any casual observer.

I think there's also some potential for modifications, such as a "time attack" mode, split-screen multiplayer races, or some kind of gameplay mechanic that would make use of the descent function, utilizing each level completely and multiple times (instead of just ascending).

I'm not criticizing the game though—far from it! I think this game has a solid design principle at heart that easily could be extended upon. *Cube Escape* may become one of those cult-following games that spawns a thousand variants. Get modding, people. ■

John Knight is a 26-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.

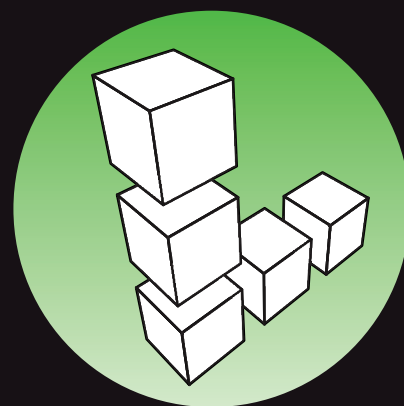
Brewing something fresh, innovative or mind-bending? Send e-mail to newprojects@linuxjournal.com.



Develop.



Deploy.



Scale.

Full root access on your own virtual server for as little as \$19.95/mo

Multiple Linux distributions to choose from • Web-based deployment • Five geographically diverse data centers • Dedicated IP address • Premium bandwidth providers • 4 core SMP Xen instances • Out of band console access • Private back-end network for clustering • IP fail-over support for high availability • Easily upgrade or add additional Linodes • Free managed DNS

For more information visit www.linode.com or call us at 609-593-7103



linode.com

SOGGo

Open-Source Groupware

The current state of SOGo and its integration capabilities with desktop and mobile clients.

LUDOVIC MARCOTTE

More than two years have passed since I last wrote about SOGo for *Linux Journal*. Since then, the project has matured greatly and now is being used successfully in tons of large-scale deployments. SOGo's strengths are really about scalability; component re-usability (IMAP, LDAP, SMTP and database server); compliance to standards, such as CalDAV and CardDAV; and excellent interoperability with popular applications, such as Mozilla Thunderbird, Apple iCal/Address Book and Microsoft Outlook. The latter is supported through a close collaboration between the SOGo and OpenChange Projects.

Since my last article, there have been more fluctuations in the world of open-source groupware, including:

- CalDAV and CardDAV protocols have gained major popularity, and extensions, such as Collection Synchronization for WebDAV and CalDAV Scheduling Extensions to WebDAV, are making those finally usable in large-scale enterprise environments.
- Mozilla Messaging finally released Thunderbird 3 with a greatly improved Mozilla Lightning extension.

- Zimbra gained tons of popularity, but Yahoo sold it to VMware for a major loss.
- OpenGroupware.org, Open-Xchange, Citadel, Kolab, eGroupWare and Bedework are fading away.
- Chandler and Bongo virtually died.
- Apple is actively integrating open standards in its calendar and address-book applications.
- DAViCal and OpenChange have emerged as great solutions to consider.
- Funambol has positioned itself as the favorite synchronization middleware.

The first incarnation of SOGo appeared with v1.0 in March 2009, and versions up to 1.3.2 quickly followed. This article presents the current state of SOGo, its integration capabilities with desktop and mobile clients and provides simple instructions to get a workable installation quickly.

Installation

Installing SOGo is very easy, as packages are available for a variety of GNU/Linux-based distributions, such as Red Hat Enterprise (and CentOS), Debian and Ubuntu. An Ubuntu-based virtual appliance also is available, which provides a complete out-of-the-box testing environment of SOGo. Here, I provide high-level installation and configuration instructions. For more in-depth instructions, look at the official SOGo documentation, which covers everything, including instructions for desktop clients and mobile devices.

SOGo leverages virtually every part of your infrastructure. It makes use of the following:

- Your LDAP server for authentication, public address books and groups extraction.
- Your database server to store events, tasks and contacts. A database server also can be used for authentication and shared address books. MySQL, Oracle and PostgreSQL are supported.
- Your IMAP server to store/retrieve e-mail. When available, SOGo also makes use of advanced capabilities, such as IMAP ACLs, Sieve, shared folders and more.
- Your SMTP server to send e-mail.
- Your HTTP server to proxy requests to the SOGo server and perform SSL encryption.

Of course, when you do not have such components in your environment, best-of-breed ones can be used from the free and

```
mysql> CREATE DATABASE sogo CHARSET='UTF8';
mysql> CREATE USER 'sogo'@'localhost' IDENTIFIED BY 'secret';
mysql> GRANT ALL PRIVILEGES ON sogo.* TO 'sogo'@'localhost';
```

SOGo's database will be used to store events, tasks, contacts and user preferences. To keep this installation as simple as possible, let's also use MySQL for user authentication. To make this work, create a database table holding user information, and add three test users all with the same MD5-encrypted password ("secret"):

```
mysql> USE sogo;
mysql> CREATE TABLE sogo_users (c_uid VARCHAR(10) PRIMARY KEY,
  ↳c_name VARCHAR(10), c_password VARCHAR(32),
  ↳c_cn VARCHAR(128), mail VARCHAR(128));
mysql> INSERT INTO sogo_users VALUES ('alice', 'alice',
  ↳MD5('secret'), 'Alice Thompson', 'alice@acme.com');
mysql> INSERT INTO sogo_users VALUES ('bob', 'bob', MD5('secret'),
  ↳'Bob Smith', 'bob@acme.com');
mysql> INSERT INTO sogo_users VALUES ('chris', 'chris', MD5('secret'),
  ↳'Chris Cooper', 'chris@acme.com');
```

In a read-world environment, your database table probably would be much more complex than that. It also might be a database view on your existing information, and in many cases, an LDAP server will be used for authentication and user information retrieval instead. SOGo supports multiple authentication sources, so you also could have an LDAP server for authentication used together with an SQL view on your CRM contacts that would be exposed as an address book to all your SOGo users.

SOGo supports multiple authentication sources, so you also could have an LDAP server for authentication used together with an SQL view on your CRM contacts that would be exposed as an address book to all your SOGo users.

Open Source community. SOGo will literally transform those loosely coupled components into a single and coherent groupware solution, which then can be accessed from your favorite Web browser or from a variety of desktop and mobile clients.

Assuming you have an Ubuntu 10.04 LTS installation and you prefer MySQL, let's proceed with the installation of SOGo and its dependencies. First, add SOGo's repository to your APT sources list, and resynchronize the package index files from their sources:

```
% sudo su -
% echo 'deb http://inverse.ca/ubuntu lucid main'
  ↳>> /etc/apt/sources.list
% apt-get update
```

Then, install SOGo, its dependencies, Apache and MySQL:

```
% apt-get install sogo sope4.9-gd11-mysql apache2 mysql-server
```

If you get Apache startup errors after the installation, ignore them. Next, create the SOGo database and the required user:

```
mysql -h localhost -u root -p
```

Next, let Apache proxy requests to SOGo. Because SOGo is not a fully compliant HTTP server, you should use Apache (or any other HTTP server) in front of it by enabling some required modules:

```
% a2enmod proxy
% a2enmod proxy_http
% a2enmod headers
```

Once enabled, you must configure SOGo so that it uses your newly created MySQL database. Again, in this example, let's not configure the Web mail part of SOGo to keep the configuration as simple as possible. Nonetheless, if you do have a working IMAP server installation (Cyrus IMAP Server and Dovecot are recommended), it will work out of the box if SOGo and your IMAP server use the same authentication source.

Proceed with SOGo's configuration under the "sogo" user. This is very important as the sogod daemon will run under the sogo user, so the preferences created by the defaults utility must belong to the sogo user:

```
% sudo su - sogo
```

FEATURE SOGo—Open-Source Groupware

```
% defaults write sogo SOGoTimeZone "America/Montreal"
% defaults write sogo SOGoMailDomain "acme.com"
% defaults write sogo SOGoLanguage English
% defaults write sogo SOGoUserSources '{(canAuthenticate = YES;
↳displayName = "SOGo Users"; id = users; isAddressBook = YES;
↳type = sql; userPasswordAlgorithm = md5; viewURL =
↳"mysql://sogo:secret@127.0.0.1:3306/sogo/sogo_users");}'
% defaults write sogo SOGoProfileURL
↳"mysql://root:secret@127.0.0.1:3306/sogo/sogo_user_profile"
% defaults write sogo OCSFolderInfoURL
↳"mysql://root:secret@127.0.0.1:3306/sogo/sogo_folder_info"
% defaults write sogo SOGoAppointmentSendEmailNotifications NO
% defaults write sogo SOGoLoginModule Calendar
% exit
```

Finally, modify the `/etc/apache2/conf.d/SOGo.conf` configuration file to use localhost with no SSL. So replace the following:

```
RequestHeader set "x-webobjects-server-port" "443"
RequestHeader set "x-webobjects-server-name" "yourhostname"
RequestHeader set "x-webobjects-server-url" "https://yourhostname"
```

with:

```
RequestHeader set "x-webobjects-server-port" "80"
RequestHeader set "x-webobjects-server-name" "localhost"
RequestHeader set "x-webobjects-server-url" "http://localhost"
```

Then, restart Apache and SOGo:

```
% /etc/init.d/apache2 restart
% /etc/init.d/sogo restart
```

If you want to use an IP address or a real DNS name to access SOGo, you must adjust this accordingly. The `"x-webobjects-server-url"` value will become the official URL to access your SOGo system. Now, from the same machine on which you performed the above steps, open your favorite Web browser and access `http://localhost/SOGo`. You should be able to log in with any of the three users created above.

Desktop Clients

Through the standard CalDAV and CardDAV protocols, SOGo supports desktop clients, such as Mozilla Thunderbird, Apple iCal and Apple Address Book, very well.

Mozilla Thunderbird, combined with the Lightning calendar extension, is the preferred client to use with SOGo. Version 2 and 3.1 of Thunderbird are supported. Thunderbird is the preferred desktop client as SOGo's Web interface shares most of its look and feel and functionality with Thunderbird. Moreover, two extensions can be installed together with Lightning to perfect the integration: the SOGo Connector and SOGo Integrator extensions. The former adds more capabilities to Thunderbird (such as CardDAV support, CalDAV ACL and so on), and the latter adds features that are vertical to SOGo (such as calendars, address-book sharing capabilities and automatic discovery, preferences synchronization and more).

Using the SOGo Integrator extension requires editing one file in the extension file subtree to specify where the SOGo server is located. This is done by hand. In an enterprise environment, this

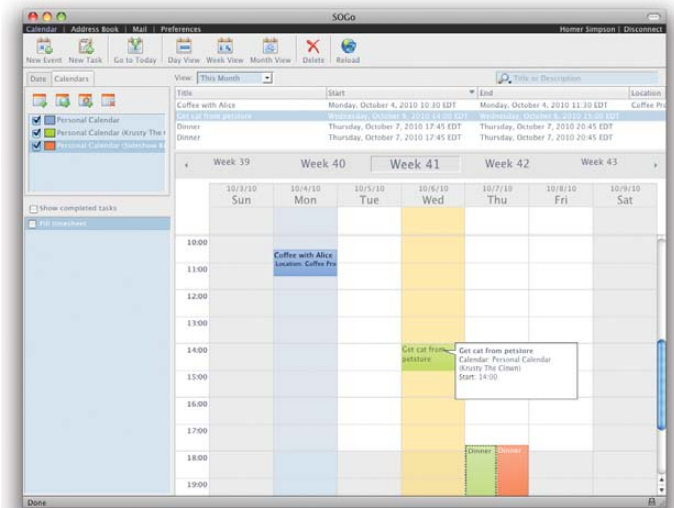


Figure 1. SOGo Web Interface

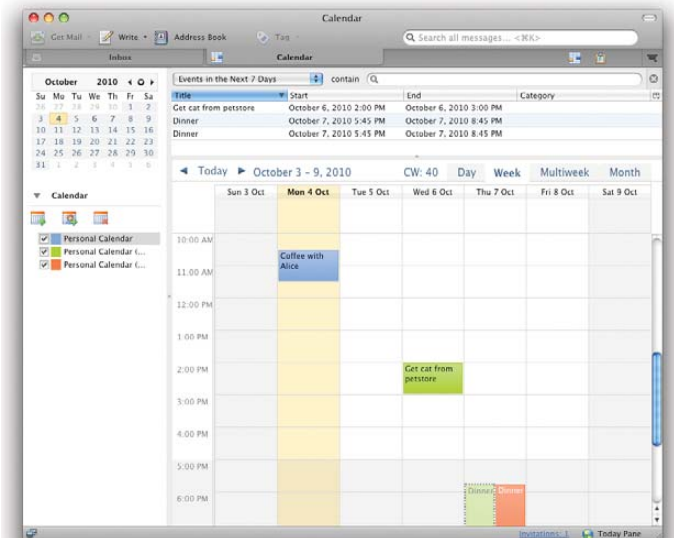


Figure 2. Mozilla Thunderbird

step is required only once per release, because the updates are expected to propagate automatically. Uncompress (using a ZIP or jar tool) the SOGo Integrator XPI, and locate the `extensions.rdf` file. This file is used for locating the extension update server and the SOGo server (let's consider those to be the same for the moment). There is a line starting with a `Seq` tag and with an attribute named `isi:updateURL`. Replace the host part of that URL with the SOGo server to which you want to connect, which again should be identical to the `x-webobjects-server-url`. For example, one would replace the following:

```
<Seq about="http://inverse.ca/sogo-integrator/extensions"
↳isi:updateURL="http://sogo-demo.inverse.ca/plugins/
↳updates.php?plugin=%ITEM_ID%&version=%ITEM_VERSION%&
↳platform=%PLATFORM%">
```

with:

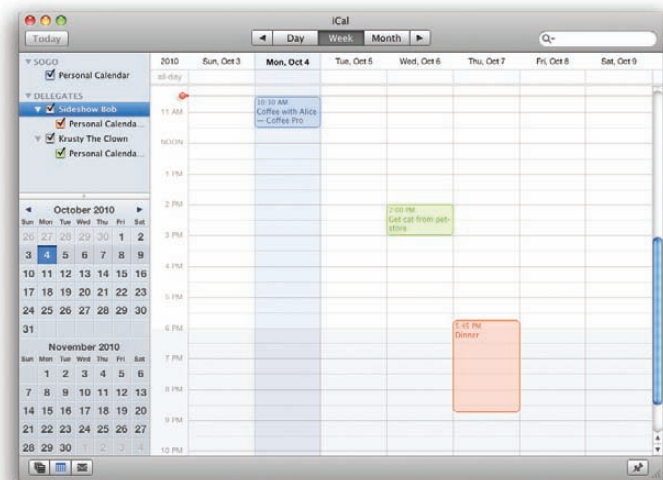


Figure 3. Apple iCal 3

```
<Seq about="http://inverse.ca/sogo-integrator/extensions"
  >isi:updateURL="https://sogo.acme.com/plugins/
  >updates.php?plugin=%ITEM_ID%&version=%ITEM_VERSION%&
  >platform=%PLATFORM%">
```

if the SOGo server is accessible from <https://sogo.acme.com/SOGO>. Once you are done modifying the configuration file, save your changes and reconstruct the XPI. As for the extension update server, it can be configured to install or uninstall Mozilla Thunderbird extensions automatically. You also can push Thunderbird settings to all your user base. Installation and configuration is documented in the “Mozilla Thunderbird—Installation and Configuration Guide”.

On Mac OS X, if you prefer Apple’s closed-source applications, you easily can use Apple iCal 3 and iCal 4 with SOGo. All features will be available, including calendar sharing and delegation, due to SOGo’s excellent compatibility with the CalDAV protocol and its implementation of some Apple-specific extensions. Since Mac OS X 10.6, it’s also possible to use Address Book with SOGo through the CardDAV protocol in order to access your contacts. When you combine those two applications with Apple Mail, a cohesive environment is created with collaboration possibilities with other users on other platforms.

The popularity gained by the CalDAV and CardDAV quickly exposed fundamental flaws in both protocols, and the Collection Synchronization for WebDAV and CalDAV Scheduling Extensions to WebDAV were created to eliminate those. The former introduces a token-based approach to DAV resources synchronization. So, instead of letting the DAV client ask for the ETag of every single item in a collection to see what has changed on the server, the client actually sends a sync token and gets in return the references of changed items from a collection. This makes the whole synchronization process of large calendars or address books very fast.

The second extension actually moves all the scheduling logic required in calendaring applications (inviting attendees, checking availabilities and so on) to the server. This avoids client-side implementation bugs and reduces client-to-server communications, which can be slow on high-latency connections.

SOGo implements those two extensions quite nicely, and

Mozilla Lightning supports both while Apple iCal limits itself to CalDAV Scheduling Extensions to WebDAV.

OpenChange Integration

To provide SOGo connectivity options to Microsoft Outlook users, SOGo makes use of the OpenChange Project. The project was founded in 2003 by Julien Kerihuel in the context of his EPITECH Innovative Project. Tightly integrated with Samba 4, the OpenChange solution is divided into three subprojects:

1. libmapi: a client-side library that can be used in existing messaging clients (Evolution, Akonadi, Mailody and so on) and offers native compatibility with Exchange server.
2. mapiproxy: a transparent gateway/proxy used to accelerate the communication between Outlook clients and an Exchange server.
3. OpenChange Server: a full implementation of the Exchange protocols that features pluggable storage providers.

The third subproject is what really interests us here. SOGo developers have created a storage provider for OpenChange that makes use of SOGo libraries in order to reuse all the business logic associated to address books, calendars and e-mail management. Microsoft Outlook communicates directly and natively to



PostgreSQL 9.0



**PostgreSQL 9.0
High Performance**

Accelerate your PostgreSQL system

Gregory Smith



**PostgreSQL 9
Admin Cookbook**

Over 80 recipes to help you run an efficient PostgreSQL 9.0 database

Simon Riggs

www.2ndQuadrant.com/books

24x7 Support, Tuning, Replication, Migration
 email: info@2ndQuadrant.us



2ndQuadrant **Professional PostgreSQL**

FEATURE SOGo—Open-Source Groupware

OpenChange (because it behaves like any Microsoft Exchange server), which in turn uses the SOGo storage provider to access all the information SOGo handles.

This makes SOGo a real, transparent Microsoft Exchange replacement, because it does not force Outlook users to use costly and hard-to-maintain MAPI connectors, which often are limited in terms of functionality. A proof of concept was released in October 2010 with many capabilities for e-mail, contacts, events and tasks. The project is being worked on actively, and by the time you read this article, a well-working version should be available for public consumption.

Samba, being vastly popular in numerous organizations and its ambitious rewrite to become an Active Directory-compatible Domain Controller, might eventually position a Samba 4, OpenChange and SOGo combo as a turnkey solution for lots of organizations requiring well-integrated directory services, file and printing services and a collaboration solution on top of it.

Mobile Devices

Although the Web interface or desktop client applications will satisfy most users, the high popularity of mobile devices, the increasing mobility of users and the need to access events, tasks, contacts or e-mail from everywhere can't be neglected by any groupware solution.

Through its Funambol connector, SOGo can synchronize

events, contacts and tasks fully with any SyncML-capable device. The Funambol Project is composed of a SyncML server and clients for devices with no built-in SyncML support, such as Research In Motion BlackBerry, Microsoft Windows Mobile, Symbian S60 or Google Android. The server part, which is a middleware, can reuse all data from SOGo using the Funambol SOGo Connector. All those free components enable billions of SyncML-capable devices to synchronize to the SOGo platform.

The Funambol middleware is a self-contained Java application. Installation is as easy as downloading Funambol Server, the Funambol SOGo Connector and creating the sync sources. Full instructions are provided in the "SOGo—Installation and Configuration Guide".

Apple iPhone users can configure their phones to use CalDAV and CardDAV and have access to their calendars and address books whenever they want. On Google Android-based devices, it also is possible to use the freely available Hypermatix CalDAV client to access calendar information from SOGo.

Configuration instructions for mobile devices are available in the "Mobile Devices—Installation and Configuration Guide".

Conclusion

As discussed in this article, the industry is moving toward open standards, such as CalDAV and CardDAV, to support collaborative applications, which SOGo supports well. Hopefully, this trend will continue, if not improve.

The native compatibility SOGo offers to Outlook users using OpenChange will be a major step in dislodging Microsoft's biggest enterprise lock-in—Exchange.

SOGo is not a finished product, and it will continue to evolve. Developers actively are improving the OpenChange and Exchange integration, implementing more Apple extensions, such as file attachments for meetings, as well as adding scripting capabilities to SOGo. The virtual appliance for SOGo, called ZEG for "Zero Effort Groupware", is a good way to try the sogo application and download it. ■

Ludovic Marcotte (lmarcotte@inverse.ca) holds a Bachelor's degree in Computer Science from the University of Montréal. Currently, he's having fun at Inverse, Inc., an open-source IT consulting company located in downtown Montréal that specializes in the development and deployment of PacketFence and SOGo.

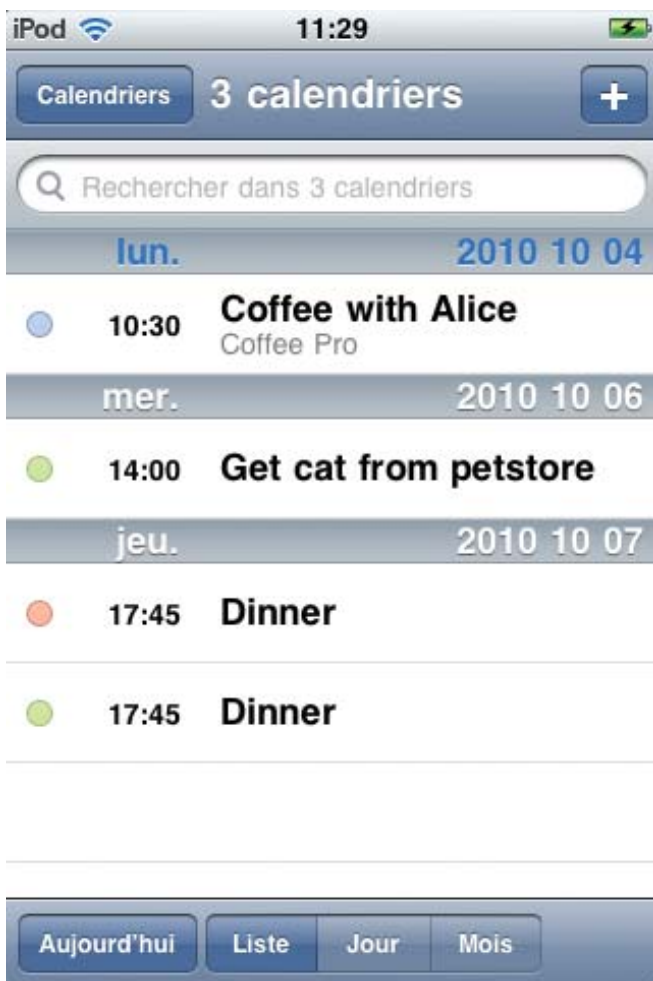


Figure 4. Apple iPhone OS 4

Resources

SOGo and Its Documentation: www.sogo.nu

OpenChange: www.openchange.org

Hypermatix CalDAV Client for Android: www.hypermatix.com/products/calendar_sync_for_android

Collection Synchronization for WebDAV: tools.ietf.org/html/draft-daboo-webdav-sync-03

CalDAV Scheduling Extensions to WebDAV: tools.ietf.org/html/draft-desruisseaux-caldav-sched-08

Funambol: www.funambol.com

SCALE 9X

The Ninth Annual
Southern California Linux Expo

Mark your calendars!

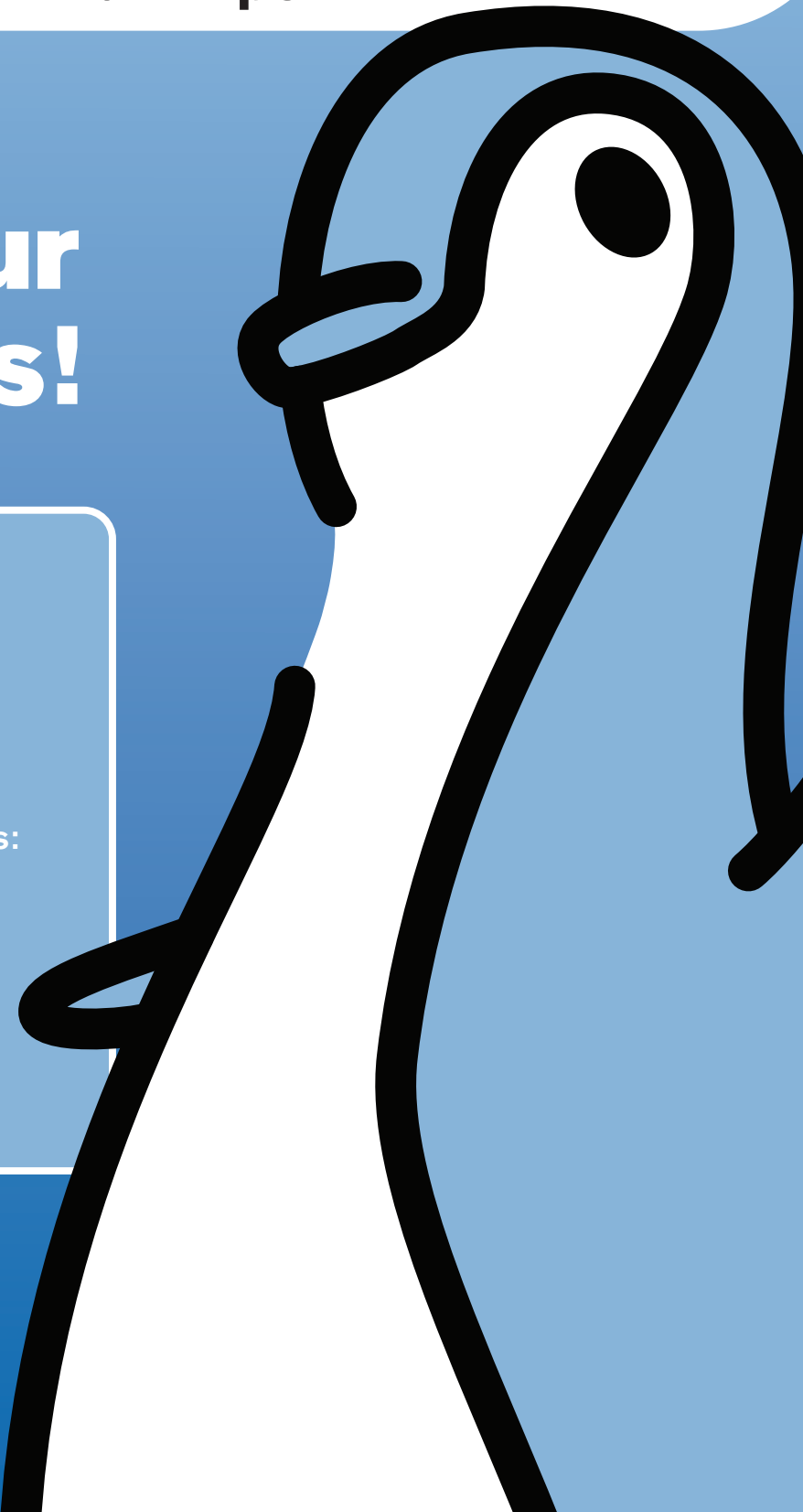
The 9th Annual
Southern California Linux Expo
is coming!

February 25-27, 2011

Expanded to five session tracks:
Beginner, Developer, SysAdmin
and two General Interest!

New, larger venue at the
Hilton Hotel @ LAX
Los Angeles, California

<http://www.socallinuxexpo.org>
Use Promo code LJAD for a 30% discount
on admission to SCALE



Use AoE *to Build Your Own* SAN

Using AoE (ATA over Ethernet) you can build a SAN (Storage Area Network) for a pittance and deliver performance that will blow your users' socks off.

GREG BLEDSOE

When I first heard of the ATA over Ethernet (AoE) protocol, I got excited about the possibilities. Sending ATA commands directly over an Ethernet physical layer offers tremendous simplicity, flexibility and low overhead that potentially could result in astonishing performance. As the rumors of official acceptance into the Linux kernel grew louder, I waited with bated breath. I believed this would be game-changing for

the storage market. When it made its way into the official kernel tree and 2.6.11 was released in early 2005 with built-in AoE support, it was all I could do not to stand up and cheer.

The obvious question is what possibly could warrant such excitement? We already had Fibre Channel and iSCSI. Why throw another technology in? Storage was almost a completely mature market, right? You might as well ask why people got excited

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.

about Voice over IP (VoIP) not long before that. The farsighted who understood the economics of open source could see where this was going to take us, and that place is disruptive. We were going to move the bar to a price point traditional vendors would find impossible to match, and it would all get turned on its head—just like VoIP. VoIP makes providing yourself with low-cost high-quality phone services essentially free if you are nerdy enough, and it allows you to become a telecom for small businesses (the business that I'm in). Similarly, AoE makes SAN available to anyone, allowing someone with some technical skill and a little capital to compete with the likes of EMC and IBM. You can, right now, literally build your own SAN with AoE initiator and target for less than \$100 dollars. See the how-to in the Resources section to do just that.

Essentially, AoE is an open-standards-

away the performance of Fibre Channel SAN at a fraction of the cost. Imagine that it was simple enough to make complex redundant designs easy to build and manage. Now imagine someone already did all that. Now open your eyes and realize that it's all true. Coraid has been fine-tuning its product offerings and stands poised to revolutionize the price to performance and manageability of network storage. The latest results show at least a five-to-one price-to-performance advantage over other technologies.

Being a consummate Linux Nerd, I found that using command-line tools to configure things a comforting respite from convoluted GUI tools from other vendors. Originally, with Coraid's gear, you could use only SATA and SATA II disks, which, although more than adequate for the vast majority of applications, was admittedly a bit of a disadvantage over SCSI-based solutions that could take advantage of

You can, right now, literally build your own SAN with AoE initiator and target for less than \$100 dollars.

based protocol that allows direct network access to disk drives by client hosts. That means AoE allows you to let a disk or RAID array on one box be interacted with by the IDE/ATA/SATA driver on another box, using the ATA commands that SATA disks use natively to do it, only over an Ethernet network—and to do it very efficiently. You build an array on one box and export it as a block device that multiple clients can see and access. You can export a disk, an array, a single partition, a disk or array split up into multiple partitions, or even a loopback device that contains an encrypted block device to any number of clients. Drivers for targets and initiators (hosts and clients) exist for all major operating systems, and it is native in the mainstream Linux kernel now.

Now, imagine that you could build a box designed specifically to be an AoE target and export your arrays as block devices over your Ethernet network. Imagine that you optimized a kernel to do that and only that, put enough processor and memory in a shelf-style chassis running your optimized kernel and AoE target driver and found that by so doing, you could throw in commodity disks and blow

SCSI's higher RPM and throughput performance along with the higher MTBF and component reliability that comes from a SCSI disk's tighter manufacturing tolerances. Coraid was nice enough to loan me some demo gear to test its next generation of solutions.

I discovered that Coraid's latest SRX series of shelves allow you to use a wider variety of interfaces, including SSD and SAS with 6GB SAS on the horizon, which makes higher-end drives available for your architecture for those applications that require it. Using 16 15k rpm SAS drives in RAID 10, I was not able to devise a test that even challenged the demo gear I had performance-wise. To see what it can do aggregating four 10-gigabit or six 1-gigabit ports, check out independent lab ESG's report found in the Resources section. It's a touch mind-boggling. I also learned that Coraid is building a point-and-click GUI interface to sit on top of the CLI, but honestly, I didn't care as long as it doesn't take my geek-friendly command line from me.

It is worth stating that some security groups have attacked the relative simplicity of the AoE protocol and asserted this

FEATURE Use AoE to Build Your Own SAN

makes it “insecure”, because it has no strong authentication mechanisms and potentially could be hijacked or subjected to other hijinks with simple techniques like MAC address spoofing. The saving grace is that AoE is not routable, meaning that people would have to plug their malicious host physically, directly in to your Ethernet LAN segment in order to be a threat, so the security of your AoE architecture is entirely dependent on the physical security of your Ethernet switches. So, this will be important to keep in mind in terms of physical placement and access to your gear. AoE rides on a lower network layer than IP, and the IP layer is what makes TCP/IP and the Internet routable, but I strongly suspect that if I needed to export an AoE block device farther than I could reach with a strand of fiber and media converters, that I could work something out with GRE (Generic Route Encapsulation) and VPN, although this is not something I’ve tested. Also keep in mind that when your clients see the AoE block devices, nothing in the AoE protocol stops you from mounting your filesystems on multiple machines, but with most filesystems, this will have devastating effects on your data integrity. So, it is up to you to build in precautions against this.

AoE overlaps with Fibre Channel and iSCSI in many ways. Its main advantage over them lies in its simplicity. The AoE specification is 12 pages long, while iSCSI clocks in at 276 pages and Fibre Channel is no longer than even a single specification, but has multiple versions in multiple parts, each hundreds of pages long with the newest releases breaking all backward-compatibility. If you want to perform an exercise in brain-bending masochism, attempt to build a Fibre Channel SAN with a mere five clients, no single point of failure and automated failover between all components. Not only is there a good chance you will go bankrupt and still be unable to do it, the complexity will make it virtually impossible to scale and a nightmare you can’t wake from to manage. iSCSI would make that task somewhat easier and less expensive, but you would find performance suffers as the overhead of TCP/IP header processing drags down

your solution. For finding the sweet spot of manageability, flexibility and performance, you won’t beat AoE gear—that’s before you even take affordability into account.

To demonstrate that, I’m going to take you through how I built a “never go down” solution for our environment using Coraid’s older SR shelves and CLN21 failover kit. Coraid is discontinuing the exact front-end gear that I bought, but building the equivalent is not terribly difficult and how-tos already exist. I don’t intend simply to re-create the how-tos, such as the ones listed in the Resources section here, so I will just hit the highlights to demonstrate some of the considerations and how easy this build really was.

The design goals I needed to meet were as follows. The item of first priority was high availability of customer media, data and application information. There should be no single point of failure, and failover should be automatic in all failure cases. Performance must allow for voice mail to be recorded from a large number of phone calls and many concurrent customers to access and play that voice mail from many application servers simultaneously. In addition, I would store application data on the SAN as well. Given expected growth over several years, meeting the availability requirements was going to be more difficult than meeting the performance requirements.

After considering and pricing several options, I decided to go with two of Coraid’s 16-disk shelves and its failover kit that comes with two Debian-based servers for NFS/CIFS gateways and a STONITH- (Shoot The Other Node In The Head) ready power supply. After deciding on this, the next decisions were what disks to use and what Ethernet gear to use. After a lot of consideration, I went with the Western Digital RE3 line for the performance and relatively high MTBF for SATA II gear. I chose 500 gigabyte disks for their cost/gigabyte (at the time I bought them) and ease of acquisition and availability. I started out with ten disks per shelf configured in RAID 10 arrays. These arrays have been going for about two years now, and so far, I’ve lost only one

of the original disks. I also discovered how easy and uneventful it is to replace disks on these shelves.

The other big consideration is that you definitely want to make sure your client NIC and switches do jumbo Ethernet frames, as the ability to aggregate ATA commands and data blocks on return will do wonders for your performance, so check out the list of jumbo-capable gear in the Resources section. I wound up buying four older Cisco Catalysts running IOS that I was able to secure inexpensively and run with a 9K MTU, which gives me two to run in production with two spares on standby.

The configuration on the gateway servers combines the two arrays, one RAID 10 array on each shelf, into a RAID 1 array. There is a performance penalty for this, as every write must be duplicated to both devices, and this doubles the overhead, but availability was the paramount concern here. Either component can fail, and your array on the gateway will degrade but continue to run until you can repair the failed shelf and/or disks. The two gateway servers are connected via a heartbeat over serial cable, so in the event of the failure of the primary, within seconds, the backup comes on-line, mounts the AoE block devices into the RAID 1, and after a pause, clients keep working as if nothing has happened. Exactly how to set this up is detailed in the Resources section.

I tested this in a variety of scenarios, and the biggest snag I hit was when each device was single-homed to one switch. In this scenario, one gateway and one shelf are on one switch, and the other server and the other shelf are on the other switch with a trunk port between switches. The problem occurs if you lose either switch. This causes things to hang indefinitely unless you do some manual tuning to the arpping utility and scripts used for failover. I had been a bit afraid of that, so I had to figure out how to dual-home all the components and make sure I had multiple uplinks between switches.

The uplinks were easy enough, just plug in two cables, tune spanning tree, and let STP figure it out. Of course, spanning tree doesn’t “fail soft”, so you can use a redundant trunk in IOS or something

The latest results show at least a five-to-one price-to-performance advantage over other technologies.

If you want to perform an exercise in brain-bending masochism, attempt to build a Fibre Channel SAN with a mere five clients, no single point of failure and automated failover between all components.

of that sort as well. Once I'd gotten that out of the way, I turned my attention to what was sure to be the most difficult part of this setup. Having tried to do multiport configurations with iSCSI and Fibre Channel, I was really dreading setting up dual homing with AoE. Here's my harrowing tale of getting it to work.

First, plug an additional port from each shelf in to each switch and turn the ports on in the configuration. Second, plug a second port on the gateways in to the AoE switches, tune its MTU, and turn it on in the configuration. Next comes the hard part. Wipe the sweat from your brow, call it a day, and go brag about how you mastered multipath AoE. Yes, it is that easy. Coraid's gear and the AoE protocol automatically discover devices and paths to them using a query-packet mechanism that makes the setup brain-dead simple, using a round-robin approach to sending packets when it finds multiple paths between the target and the client. Once you have this in place, you could lose one of each of your components: gateway, links, switches and shelves, and continue to run.

Configuring a LUN and exporting it is super simple and covered in multiple how-tos that can be found in multiple places. You log in to the shelves from a device running AoE on the same Ethernet segment with a utility called cec (Coraid Ethernet console) and issue a sequence of commands (the example below is for a different set of drives than those mentioned above):

```
SR shelf 1> show -l
1.0 500GB up
1.1 500GB up
1.2 500GB up
1.3 500GB up
1.4 500GB up
1.5 500GB up
1.6 500GB up
1.7 500GB up
1.8 500GB up
1.9 500GB up
1.10 500GB up
1.11 500GB up
```

```
1.12 500GB up
1.13 500GB up
1.14 500GB up
1.15 500GB up
```

```
SR shelf 1> make 0 raid10 1.0-15
```

```
SR shelf 1> list -l
0 4000GB offline
0.0 4000GB raid10 normal
0.0.0 normal
...
```

```
SR shelf 1> online 0
```

```
SR shelf 1> online
0 4000GB online
```

Of course, you could create any number of LUNs here using any combination of RAID 0,1,5,6,10 JBOD or any other supported RAID type.

To use your new LUN on a connected server, it is as simple as:

```
client:/# aoe-discover
client:/# aoe-stat
e1.0 4000GB eth0 up

client:/# mkfs.ext4 /dev/etherd/e1.0
```

```
client:/# mount /dev/etherd/e1.0 /mnt/aoe
```

It really is just as easy as that. I hope you can see the flexibility and the power this approach affords—simple management for complex architectures. I have validated this architecture and this methodology by using it in production for the last two years. There is so much more that can be done, and so much more I plan to do. If you remember my last article regarding VirtualBox in the October 2010 issue of *LJ*, you know that my next project is to move our virtual machine images onto AoE back ends to complete the requirements to allow me to be able to teleport running virtual machines between virtual hosts in our production environment. My next project after that will be to experiment with AoE and GFS (global filesystem) to eliminate the gateway server and give multiple servers access to the same LUNs at the same time. Should be fun! ■

Greg Bledsoe is the Manager of Technical Operations for a Standout VoIP Startup, Aptela (www.aptela.com), an author, husband, father to six children, wine enthusiast, amateur philosopher and general know-it-all who welcomes your comments and criticism at lj@bledsoehome.net.

Resources

"ATA Over Ethernet: Putting Hard Drives on the LAN" by Ed L. Cashin, *LJ*, June 2005: www.linuxjournal.com/article/8149

How to Build a Low-Cost SAN: howtoforge.net/how-to-build-a-low-cost-san

"Simplest Ethernet Storage" by Chris Mellor: www.theregister.co.uk/2010/08/10/coraid_esg

The CLN Failover Kit HOWTO by Ed. L. Cashin: support.coraid.com/support/cln/ft/failover-kit.html

"Getting Started with Heartbeat" by Daniel Bartholomew, *LJ*, November 2007: www.linuxjournal.com/article/9838

Jumbo Frame Clean Networking Gear: www.uoregon.edu/~joe/jumbo-clean-gear.html

VLAN Support in Linux

Add flexibility and take Ethernet networking to the next level by turning your Linux box into a VLAN Smart Switch.

HENRY VAN STYN

It's no surprise that Linux makes a great router and firewall. A lesser-known fact is that you also can use Linux as an Ethernet bridge and VLAN switch, and that these features are similarly powerful, mature and refined. Just like Linux can do anything a PIX or SonicWall can do, it also can do anything a managed VLAN "Smart Switch" can do—and more.

In this article, I give a brief overview of 802.1Q VLAN technology and configurations and then explain how you can configure Linux to interface directly with VLAN topologies. I describe how a VLAN switch can help you add virtual Ethernet interfaces to a Linux box, saving you the need to buy hardware or even reboot. I also look at solving some of those small-scale network annoyances. Do you think your Linux firewall has to be located near your Internet connection and have two network cards? Read on.

Why VLAN?

When you hear the term VLAN, large-scale corporate or campus networks might come to mind. Easing the burden of maintaining these types of networks was one of the primary reasons VLAN technology originally was developed.

VLANs allow network topology to be rearranged on demand—purely in software—without the need to move physical cables. VLANs also allow multiple separate layer-2 networks to share the same physical link, allowing for more flexible and cost-effective cabling layouts. VLANs let you do more with less.

Take the example of a large spread-out network with multiple LANs and data closets. Without the benefit of VLANs, the only way you can move a device to a new LAN is if it's accessible in the same data closet (where the device connects) as the old LAN. If it's not, you have no option other than pulling a new cable or physically moving the device to a location where the new LAN is accessible. With VLANs, however, this is a simple configuration change.

These are the types of benefits and applications that usually are associated with VLANs, but there are many more

scenarios beyond those that are useful in all sized networks, even small ones.

Because VLAN switches historically have been expensive, their use has been limited to larger networks and larger budgets. But in recent years, prices have dropped and availability has increased with brands like Netgear and Linksys entering the market.

Today, VLAN switches are cheap (less than \$100) and are starting to become commonplace. I suspect that in a few more years, it'll be hard to find a switch without VLAN support, just like it's hard to find a "hub" today.

802.1Q VLAN Primer

The purpose of VLAN (Virtual LAN) is to build LANs from individual ports instead of entire switches. A VLAN config can be as simple as groupings of ports on a single switch. The switch just prevents the ports of separate groups (VLANs) from talking to each other. This is no different from having two, three or more unconnected switches, but you need only one physical switch, and it's up to you how to divide the ports among the "virtual switches" (three ports here, eight ports there and so on).

When you want to extend this concept across multiple switches, things become more complicated. The switches need a standard way to cooperate and keep track of which traffic belongs to which VLAN. The purpose is still the same—build LANs from individual ports instead of entire switches, even if the ports are spread across multiple switches and even multiple geographic locations.

You can think of VLAN switches as a natural evolution of Ethernet devices, with its ancestors being the switch and the hub. The fundamental difference between a switch and a hub is that a

switch makes decisions. It won't send packets to ports where it knows the destination MAC can't be found. Switches automatically learn about valid port/MAC mappings in real time as they process packets (and store that information in their "ARP cache").

A VLAN switch adds another condition on top of this. It won't send packets to ports (the "egress port" or sink) that aren't "members" of the VLAN to which the packet belongs. This is based on the VLAN ID (VID) of the packet, which is a number between 1 and 4096.

If a packet doesn't already have a VID, it is assigned one based on the port on which it arrived (the "ingress port" or source). This is the Primary VID (PVID) of the port. Each switch port can be a member of multiple VLANs, one of which must be configured as its PVID.

The VID is stored in an extra 4-byte header that is added to the packet called the Tag. Adding a Tag to a packet is called Tagging. Only VLAN devices know what to do with Tagged packets; normal Ethernet devices don't expect to see them. Unless the packet is being sent to another VLAN switch, the Tag needs to be removed before it is sent. This Untagging is done after the switch determines the egress port.

If a port is connected to another VLAN switch, the Tags need to be preserved, so the next switch can identify the VLANs of the packets and handle them accordingly. When a packet has to cross multiple switches in a VLAN, all subsequent switches rely on the VID that was assigned to the packet by the first switch that received it.

All packets start out as Untagged when they enter the network, and they also should always end as

IEEE 802.1Q

Although various manufacturers initially created other proprietary VLAN formats, the prevailing standard in use today is 802.1Q. In a nutshell, 802.1Q provides a simple way for multiple VLAN switches to cooperate by attaching VLAN-specific data directly to the headers of individual Ethernet packets.

IEEE 802.1Q is an open standard, which means, theoretically, that all compatible devices can interoperate regardless of manufacturer. Linux VLANs are based on 802.1Q, and almost any switch that advertises "VLAN" will support this standard.

NOTE:

In the case of a VLAN with only a single switch, no Tagged packets should be sent or received. However, it's still useful to think of the Tagging and Untagging as occurring:

- Packet arrives and is Tagged according to the PVID of the ingress port.
- Egress port is determined based on the VID in the Tag.
- Packet is Untagged and sent.

Untagged when they leave the network and arrive at their destination. Along their journey, if they cross a VLAN network, they will be Tagged with a VID, switched according to this VID by one or more VLAN switches, and then finally Untagged by the last VLAN switch.

If you've been keeping track, you know there are three things you need to configure for each port of each switch:

- Member VLANs (list of VLANs).
- PVID (must be one of the member VLANs).
- Whether packets should be left Tagged or Untagged when sent (egress).

With one or more switches, you can achieve any VLAN topology by selectively configuring the above three settings on each port.

Linux as a Switch (aka Bridge)

A VLAN switch is really just a normal switch with some extended functionality. Before you can have a VLAN switch, you first need to have a normal switch. Fortunately, Linux already has full support for this—it's just not called "switching".

The functionality that makes Linux what you would think of as a "switch" is called bridging, a more specific and accurate term,

The fundamental difference between a switch and a hub is that a switch makes decisions.

because it's based on the official bridge standard, IEEE 802.1D.

For all practical purposes, switches and bridges are the same thing. Switch is a loose term coined by the industry that means different things for different products (for example, \$10 switches usually don't fully support 802.1D, while \$500 switches usually at least support 802.1D, plus lots of other features).

An 802.1Q VLAN switch is really a VLAN bridge, because 802.1Q as a standard just extends 802.1D (all devices that support 802.1Q must also support 802.1D). Technically, VLAN bridge is the correct terminology, but very few people would know what that means.

Configuring Bridges

In order to use bridges in Linux, you need a kernel compiled with CONFIG_BRIDGE and the userland package bridge-utils. I suggest you also add the ebtables kernel options and userland tool.

Think of each Ethernet interface in your system as a one-port switch. An Ethernet interface already performs the same basic functions as a switch—forwarding packets, maintaining an ARP cache and so on—but on a single port without the need or capability to decide to which other port(s) a packet should be sent.

Linux's bridging code elegantly plugs in to and extends the existing functionality by letting you define bridges as virtual Ethernet interfaces that bundle one or more regular Ethernet

interfaces. Each interface within the bridge is a port. In operation, this is exactly like ports of a switch.

The userland tool for administering bridges is brctl. Here's how you would set up a new bridge comprising eth0 and eth1:

```
brctl addbr br0
brctl addif eth0
brctl addif eth1
ip link set br0 up
```

Once you run these commands, you'll have a new Ethernet interface named br0 that is the aggregate of both eth0 and eth1. For typical usage, you wouldn't configure IP addresses on eth0 and eth1 anymore—you would now use br0 instead.

The best way to understand how this works is to imagine br0 as a physical Ethernet interface in your box that's plugged in to a three-port

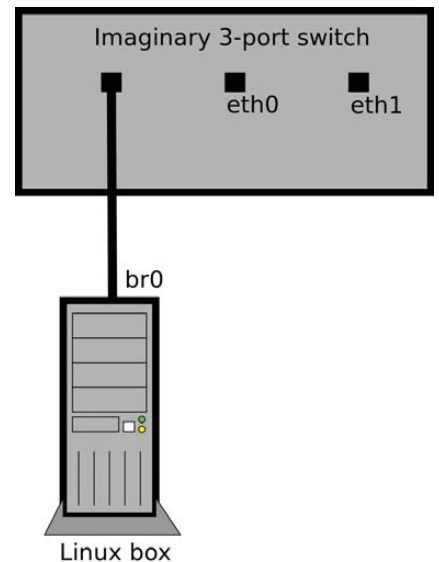


Figure 1. Imaginary Three-Port Switch Created by a Bridge

switch on your desk. Because br0 is plugged in to one of the ports, this would leave the switch with two remaining ports—eth0 and eth1 are these two switch ports (Figure 1).

Packets will pass between the interfaces/ports, the bridge will learn and maintain an ARP cache, and like a switch, it will decide to which ports each packet should be forwarded.

But, unlike a normal switch external to your system, you own and control all of the ports. You now can use any and all of the tools available under Linux to create the ultimate managed switch!

Because you still have access to the underlying Ethernet interfaces, you can do things like sniffing with tcpdump or snort on ports individually. Using the ebtables package, you can filter the Ethernet packets that pass through your switch with the same control and precision as iptables for IP packets.

Topics such as ebtables are beyond the scope of this article, but see the ebtables man page and Web site (see Resources).

Configuring VLAN Interfaces

VLAN support requires a kernel compiled with CONFIG_VLAN_8021Q and the vlan userland package (I suggest you also enable CONFIG_BRIDGE_EBT_VLAN so you can match VLANs in ebtables rules).

Use the vconfig tool to create virtual VLAN interfaces based on the combination of a physical Ethernet interface and a specific VLAN ID. These interfaces can be used like any other Ethernet interface on your system.

Run the following commands to add a new interface associated with eth0 and VID 5:

```
vconfig add eth0 5
ip link set eth0.5 up
```

This will create the virtual interface eth0.5, which will have the following special VLAN-specific behaviors:

- Packets sent from eth0.5 will be Tagged with VID 5 and sent from eth0.
- Packets received on eth0 Tagged with VID 5 will show up on eth0.5 as normal (that is, Untagged) packets.

Only packets that were Tagged with VID 5 will arrive on the virtual VLAN interface.

Bringing It All Together

The biggest difference between Linux and an off-the-shelf VLAN switch is that Linux can participate as a host on the network rather than just forward packets for other hosts. Because the Linux box itself can be the endpoint of network communications, the configuration approach is different from that of a typical VLAN switch.

Instead of setting VLAN membership for each port, each port/VID combination gets its own virtual eth interface. By adding these interfaces and optionally bridging them with physical interfaces, you can create any desired VLAN configuration.

There is no per-port PVID setting in Linux. It is implicit based on to which VLAN interface(s) the physical ingress interface is bridged. Packets are Tagged if they are sent out on a virtual VLAN interface according to the VID of that interface. Tagging and Untagging operations happen automatically as packets flow between physical and virtual interfaces of a given bridge. Remember that the PVID setting is relevant only when forwarding packets that were received as Untagged.

With a typical VLAN switch there is only one bridge (the switch itself), of which every port is a member. Traffic segmentation is achieved with separate per-port ingress (PVID) and egress VLAN membership rules. Because Linux can have multiple bridges, the PVID setting is unnecessary.

These details are simply convention; the effective configurations are still the same across all VLAN platforms. It sounds more complicated than it actually is. The best way to understand all this is with some real-world examples.

Join Existing VLANs

Let's say you have a Linux box with a single physical interface (eth0) that you want to join to three existing VLANs: VIDs 10, 20 and 30. First, you need to verify the configuration of the existing switch/port into which you will plug the Linux box. It needs to be a member of all three VLANs, with Tagging on for all three VLANs. Next, run these commands on the Linux box:

```
ip link set eth0 up
vconfig add eth0 10
ip link set eth0.10 up
vconfig add eth0 20
ip link set eth0.20 up
```

```
vconfig add eth0 30
ip link set eth0.30 up
```

You then can use eth0.10, eth0.20 and eth0.30 as normal interfaces (add IP addresses, run dhclient and so on). These will behave just like normal physical interfaces connected to each of the VLANs. There is only one physical interface in this example, so there is no need to define a bridge.

Extend Existing VLANs

Let's say you want to use the Linux box in the above example to connect a non-VLAN-aware laptop to VLAN 20. You'll need to add another physical interface (eth1), and then bridge it with eth0.20. I'm naming the bridge vlan20, but you can name it anything:

```
brctl addbr vlan20
ip link set vlan20 up
brctl addif vlan20 eth0.20
ip link set eth1 up
brctl addif vlan20 eth1
```

Now eth1 is a port on VLAN 20, and you can plug in the laptop (or a whole switch to connect multiple devices). Any devices connected through eth1 will see VLAN 20 as a normal Ethernet network (Untagged packets), as shown in Figure 2.

The implied PVID of eth1 is 20 because it's bridged with

Ultra Small Panel PC

PPC-E4

- Fanless ARM9 200MHz CPU
- 3 Serial Ports & SPI
- Open Frame Design
- 2 USB 2.0 Host Ports
- 10/100 BaseT Ethernet
- Audio Beeper
- Micro SD Flash Card Interface
- Battery Backed Real Time Clock
- 64 MB Flash & 64 MB RAM
- Linux with Eclipse IDE or WinCE 6.0
- JTAG for Debugging with Real-Time Trace
- WQVGA (480 x 272) Resolution TFT LCD with Touch Screen
- Four 12-Bit A/Ds, Two 16-Bit & One 32-Bit Timer/Counters



The PPC-E4, an ultra compact Panel PC with a 4.3 inch WQVGA (480 x 272) TFT color LCD and a resistive touch screen. The dimensions of the PPC-E4 are 4.8" by 3.0", about the same dimensions as that of popular touch cell phones. The PPC-E4 is small enough to fit in a 2U rack enclosure. **Price is \$345 at quantity 1.**

For more info visit: www.emacinc.com/panel_pc/ppc_e4.htm

Since 1985
OVER
24
YEARS OF
SINGLE BOARD
SOLUTIONS

EMAC, inc.

EQUIPMENT MONITOR AND CONTROL

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

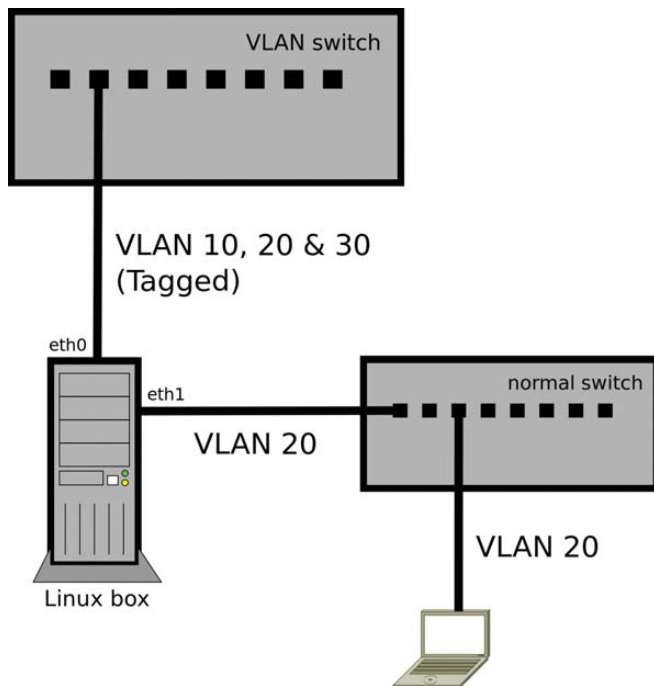


Figure 2. Extended VLAN Configuration

that virtual VLAN interface. You're not creating any VLAN interfaces on eth1 (such as eth1.20), because you don't want it to send or receive Tagged packets. It's the bridge with eth0.20 that makes eth1 a "member" of the VLAN.

As with any bridge config, you'll also need to stop using eth0.20 as a configured interface and start using vlan20 in its place.

Single Interface Firewall

The typical configuration of a Linux box as a firewall/gateway is to have two physical interfaces, with one connected to the Internet router (public side) and the other connected to the internal LAN switch (private side), as shown in Figure 3.

But, what if the Internet router and switch/patch panel are inside a wiring closet where there is no room to install a Linux box, and every possible location to put it has only a single jack/cable?

VLANs make this no problem. Instead of installing the Linux box physically in between the public and private networks, you can install a small off-the-shelf VLAN switch, configured with two VLANs (VIDs 1 and 2).

Configure one port as a member of both VLANs with Tagging on. You'll plug the Linux box in to this port. This should be the only port configured with Tagging, because it's the only port that will talk to another VLAN device (the Linux box). Every other port will be set to Untagged.

Configure another port of the switch as a member of VLAN 2 only (Untagged, PVID set to 2). You'll plug the Internet router in to this port.

Leave the rest of the ports on VLAN 1 only (Untagged, PVID set to 1). These are the ports for all the hosts on the private network. If there are more hosts than ports, you can plug in another switch or switches (non-VLAN) to any of these VLAN 1 ports to service the rest of the hosts.

The Linux box needs only one physical interface (eth0). Run

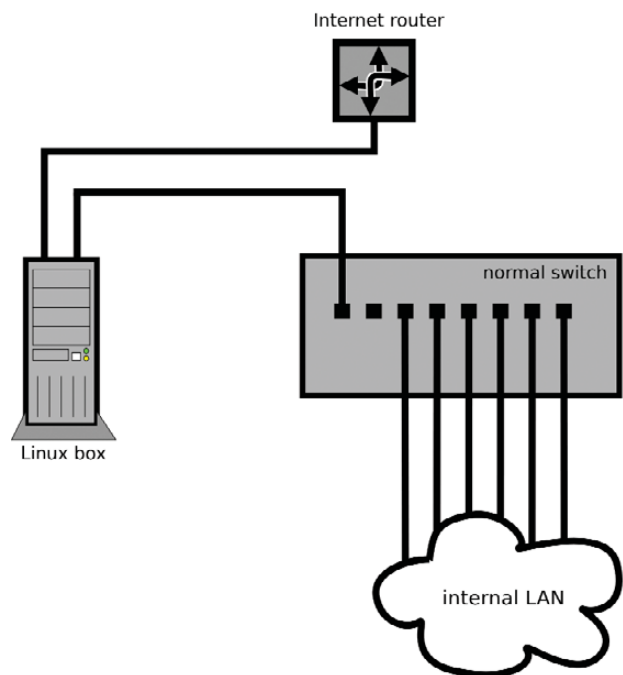


Figure 3. Typical Dual-Interface Firewall Configuration

these commands to configure the VLANs:

```
ip link set eth0 up
vconfig add eth0 1
ip link set eth0.1 up
vconfig add eth0 2
ip link set eth0.2 up
```

Just like in the first example, you now would configure your IP addresses and firewall normally, using eth0.1 as the interface on the private network and eth0.2 as the interface on the public network (Figure 4).

As in the first example, because there is only one physical interface in the Linux box, there is no need to define a bridge.

The VLAN switch ports in this example are acting like interfaces of the Linux box. You easily can extend this concept for other applications and scenarios. Using a 24-port VLAN switch, you could have the equivalent of 23 Ethernet interfaces in a Linux box if you created 23 separate VLANs. The 24th port would be used to connect the Linux box to the switch and would need to Tag all the packets for the 23 VLANs.

Testing

You can use tcpdump to see Tagged and Untagged packets on the wire and to make sure traffic is showing up on the expected interfaces. Use the -e option to view the Ethernet header info (which shows 802.1Q Tags) and the -i option to sniff on a specific interface. For example, run this command to show traffic for VLAN 10:

```
tcpdump -e -i eth0.10
```

You should see normal traffic without VLAN Tags. If VLAN 10 contains more than a few hosts, you should at least start seeing ARP and other normal broadcast packets (like any switched network, you

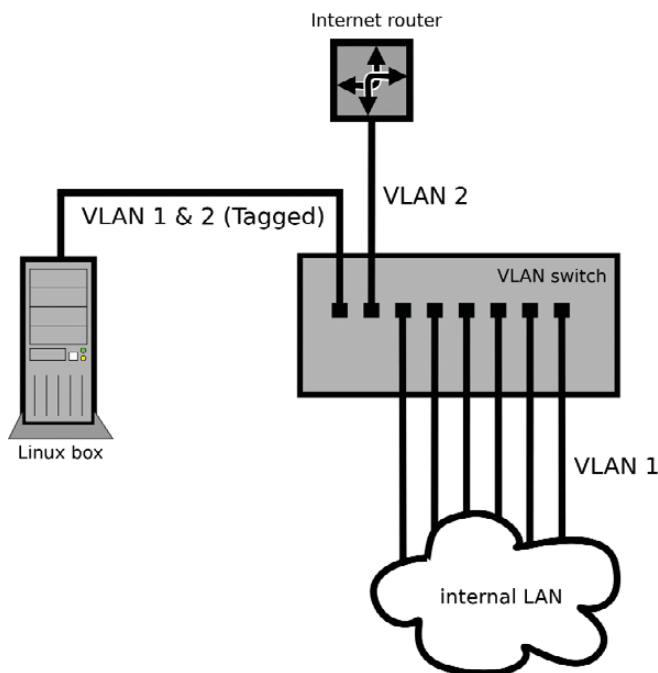


Figure 4. Single-Interface Firewall Configuration with VLANs

won't see unicast packets not addressed to your host/bridge).

If the eth0.10 VLAN interface is working correctly above, you should see the Tagged 802.1Q packets if you look at the traffic on the underlying physical interface, eth0:

```
tcpdump -e -i eth0
```

If you run this command at the same time as the eth0.10 capture, you should see the Tagged version of the same packets (as well as packets for any other VLAN interfaces set up on eth0).■

Henry Van Styn is the founder of IntelliTree Solutions, an IT consulting and software development firm located in Cincinnati, Ohio. Henry has been developing software and solutions for more than ten years, ranging from sophisticated Web applications to low-level network and system utilities. He is the author of Strong Branch Linux, an in-house server distribution based on Gentoo. Henry can be contacted at www.intellitree.com.

Resources

802.1Q VLAN Implementation for Linux (vlan package):
www.candelatech.com/~greear/vlan.html

bridge-utils: www.linuxfoundation.org/collaborate/workgroups/networking/bridge

ebtables: ebtables.sourceforge.net

IEEE 802.1Q-2005—Virtual Bridged Local Area Networks:
standards.ieee.org/getieee802/download/802.1Q-2005.pdf

IEEE 802.1D-2004—Media Access Control (MAC) Bridges:
standards.ieee.org/getieee802/download/802.1D-2004.pdf



Linux - FreeBSD - OpenSolaris - etc.

Proven Technology.

Proven Reliability.

When you can't afford to take chances with your business data or productivity, rely on a Genstor server customized to your specifications.

POWER

PERFORMANCE

Fly into the Cloud with Genstor Systems



- Up to 48 cores in a 1U.
- AMD Opteron 6100 series.
- Single high-efficiency power supply.
- Up to 512GB DDR3 memory.
- Ideal as front end processing servers.



- Up to 12 cores in a 2U
- Dual redundant high efficiency power.
- Up to 96GB DDR3 memory.
- Server Power Capping via Intel Intelligent Power Node Manager.
- Ideal as front end processing and/or storage.



- Up to 4 GPU cards.
- Dual redundant high efficiency power.
- Up to 192GB DDR3 memory
- Up to 24 cores using 2 CPUs.
- Up to 8 3.5" disks.

Genstor Systems, Inc.



780 Montague Express. # 604
 San Jose, CA 95131
www.genstor.com
 E-mail: sales@genstor.com
 Phone: 877-25 SERVER
 408-383-0120

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

Archiving Data with Snapshots in LVM2

Simplify the time-consuming data backup process with zero downtime using LVM2 snapshot.

PETROS KOUTOUPIS

Sometimes we use a technology even though we're unaware of its full features and capabilities and how they may be able to benefit us. One such feature is the data snapshot. The snapshot is a single state (that is, a copy) of a storage volume at a particular point in time. A volume can refer to a disk device or partition. The snapshot is primarily a data backup technology. Directed toward larger storage capacities, utilizing such a technique has advantages. For instance, full backups of an entire volume can take a long time and also use large amounts of storage space, even for files that will remain unchanged for some time to come. Also, when performing a data backup of entire volumes or subsets of volumes in a symmetric multiprocessing environment, write operations still may continue to modify the file data on that volume, preventing atomicity and, in turn, possibly leading to data corruption. There are ways around the latter in which the volume can be taken off-line or marked as read-only prior to the archival process, but in high-availability production environments, that may never be an option. This is where the snapshot comes in.

Used to avoid downtime and retain atomicity, the snapshot provides a read-only copy of a specified volume at a specific point in time, usually implemented by a copy-on-write mechanism. Some vendors and software implementations are known to support write commands via a concept known as branching snapshots, in which diverging versions of data are created via an

extremely complex system of pointers, all based on the original snapshot. When you write to a snapshot or the original volume, the write will not be seen by the other. The way this works is when a volume marked for snapshot gets written to and data is modified, the original and unchanged data block(s) or file data (in the case of a file-based snapshot) will be copied to the space allocated for the snapshot. After all original and unmodified data are copied over to the snapshot, the original volume will be updated with the new data. When the snapshot volume needs to be mounted, using a system of pointers, the snapshot will reference the parent volume with the original data saved in the snapshot. With such a technique, it now becomes possible to archive valuable data incrementally without losing productivity or the risk of suffering from any data corruption.

The use of snapshot technologies can be seen in a variety of environments, ranging from external storage controllers, filesystems, virtual machines (such as VMware, VirtualBox and so on), databases and even volume managers, which is the focus of this article. Here, I cover the snapshot feature found in LVM2 and how to manage it, all from the command line.

NOTE:
LVM2 refers to a collection of user-space tools that provide logical volume management on Linux.

The Linux Logical Volume Manager

The second generation of the Linux Logical Volume Manager (LVM2) is a logical volume manager capable of pooling multiple storage devices together to represent a single volume or volumes, either in a striped or mirrored fashion. Everything is created and managed on a layer-by-layer basis. First is the physical volume. It is followed by the volume group and then the mountable logical volume itself. Most mainstream Linux distributions usually have the LVM2 userland tools preinstalled. If you find that it's not installed on your distribution, download and install it via your distribution's package repository.

The idea is almost similar in concept to the Redundant Array of Independent Disks (RAID), and although LVM2 does not support any parity-driven striping, it still adds additional value. For instance, LVM2 allows for the uninterrupted addition, removal and replacement of storage devices. It makes for easy dynamic resizing of volume groups and logical volumes. Most important, it supports the snapshot—the focus of this article. As of LVM2, write operations are supported to snapshot volumes.

As mentioned earlier, LVM2 volumes utilize a layered structure—that is, physical volumes (or PVs) must be created from

The idea is almost similar in concept to the Redundant Array of Independent Disks (RAID), and although LVM2 does not support any parity-driven striping, it still adds additional value.

a physical disk device. This can be accomplished with the `pvcreate` command followed by the list of physical partitions to label for LVM2 usage:

```
$ sudo pvcreate /dev/sda1 /dev/sdb1 /dev/sdc1 /dev/sdd1
```

With the newly labeled physical volumes, volume groups (or VGs) need to be created with the `vgcreate` command, followed by a name for the volume group and then a list of all physical volumes to use:

```
$ sudo vgcreate vg0 /dev/sda1 /dev/sdb1 /dev/sdc1 /dev/sdd1
Volume group "vg0" successfully created
```

By default, the volume groups are located in the `/dev` directory path. It is with this volume group that logical volumes (or LVs) can be created, formatted with a filesystem, mounted and, in turn, used for file I/O operations. The best feature of creating logical volumes is that you can use some or all available capacity of the VG. For instance, if a 1GB LV needs to be created from the 4GB VG, the `lvcreate` command needs to be used followed by the name of the VG and then a size for the LV. When an LV is created, it will create a node name for accessibility in the `/dev` directory path under the volume group's name:

```
$ sudo lvcreate --name /dev/vg0/test_vg --size 1G
Logical volume "test_vg" created
```

The example above showcases the creation of a nonredundant LV. To create an LV with mirroring capabilities, invoke the `lvcreate` command with the `-m` option. The example below creates a 500MB-mirrored LV:

```
$ sudo lvcreate --size 500M -m1 --name mirrorlv vg0
  /dev/sda1 /dev/sdb1 /dev/sdc1 /dev/sdd1
Logical volume "mirrorlv" created
```

You can remove logical volumes, volume groups and physical volumes easily with the `lvremove`, `vgremove` and `pvremove` commands followed by their respective volume names:

```
$ sudo lvremove /dev/vg0/test_vg
Do you really want to remove active logical volume "test_vg"? [y/n]: y
Logical volume "test_vg" successfully removed
```

Note that a list of all logical volumes, volume groups and physical volumes with detailed volume information can be displayed with the `lvdisplay`, `vgdisplay` and `pvdisplay` commands.

LVM2 Snapshots

Now that I've covered a brief summary of how LVM2 is structured and managed, it's time to focus on the snapshot feature. It is worth noting that the LVM2 snapshot feature can be used only on LVM2-managed logical volumes. Assuming that an LV already

exists, possibly the partition for the `/` directory path, a second LV needs to be created for the snapshot of the original logical volume. With regard to size, another great feature of the snapshot is that the snapshot volume does not have to be equal in size to the original volume. The size even can be half or less than the original volume, allowing only that many changes of data to be backed up. By default, LVM2 will disable the snapshot automatically if the snapshot LV ever gets filled. The amount of storage space necessary is dependent on the usage of the snapshot. If the snapshot size equals the size of the original LV, it never will overflow, and snapshot service will not be interrupted. In the worst-case scenario, if it is found that space is running out on the snapshot, the LV always can be resized dynamically to a larger capacity.

Define the size to allocate for the snapshot. Create the snapshot on the desired VG by using the `lvcreate` command, with the size followed by the snapshot switch, the name for the snapshot and the VG. In this example, only 500MB are allocated for modified data. Realistically, this is not an ideal size to use (it's too small but serves its purpose here):

```
$ sudo lvcreate -L500M -s -n rootsnapshot /dev/VolGroup/lv_root
Logical volume "rootsnapshot" created
```

The `lvdisplay` command displays all details pertaining to the snapshot LV. One detail to keep an eye on is the "Allocated to snapshot" value. In this example, it is set to 0.06%:

```
$ sudo lvdisplay /dev/VolGroup/rootsnapshot
--- Logical volume ---
LV Name                /dev/VolGroup/rootsnapshot
VG Name                VolGroup
LV UUID                kAc3Iq-Gn3e-pBWs-KC9V-bFi8-0fHr-SsdRLR
```

Listing 1. Mounting and Listing of the Snapshot-Enabled Volume

```
$ sudo mkdir -p /mnt/VolGroup/rootsnapshot
$ sudo mount /dev/VolGroup/rootsnapshot /mnt/VolGroup/rootsnapshot/
$ ls -l /mnt/VolGroup/rootsnapshot/
total 124
dr-xr-xr-x.  2  root root  4096  May 15 07:45  bin
drwxr-xr-x.  2  root root  4096  May 15 06:59  boot
drwxr-xr-x.  9  root root  4096  Sep 26 06:12  cgroup
drwxr-xr-x.  2  root root  4096  May 15 06:59  dev
drwxr-xr-x. 116 root root 12288 Sep 26 06:19  etc
drwxr-xr-x.  3  root root  4096  May 15 08:10  home
dr-xr-xr-x. 17  root root 12288 May 15 07:42  lib
drwx-----  2  root root 16384  May 15 06:58  lost+found
drwxr-xr-x.  3  root root  4096  Sep 26 06:15  media
dr-xr-xr-x.  2  root root  4096  Sep 26 06:13  misc
drwxr-xr-x.  3  root root  4096  Dec  4 2009  mnt
dr-xr-xr-x.  2  root root  4096  Sep 26 06:13  net
drwxr-xr-x.  2  root root  4096  Dec  4 2009  opt
drwxr-xr-x.  2  root root  4096  May 15 06:59  proc
dr-xr-x-...  4  root root  4096  Aug 31 15:54  root
dr-xr-xr-x.  2  root root 12288 May 15 07:48  sbin
drwxr-xr-x.  2  root root  4096  May 15 07:02  selinux
drwxr-xr-x.  2  root root  4096  Dec  4 2009  srv
drwxr-xr-x.  2  root root  4096  May 15 06:59  sys
drwxrwxrwt. 15  root root  4096  Sep 26 06:27  tmp
drwxr-xr-x. 14  root root  4096  May 15 07:14  usr
drwxr-xr-x. 22  root root  4096  May 15 07:48  var
```

```
LV Write Access      read/write
LV snapshot status  active destination for /dev/VolGroup/lv_root
LV Status           available
# open              0
LV Size             5.51 GiB
Current LE          1410
COW-table size      500.00 MiB
COW-table LE        125
Allocated to snapshot 0.06%
Snapshot chunk size 4.00 KiB
Segments            1
Allocation           inherit
Read ahead sectors  auto
- currently set to 256
Block device        253:3
```

If the original LV is written to, using the copy-on-write mechanism, the snapshot will write all original data from the original volume to the snapshot volume before it replaces the original volume with the new data. To better understand the mechanics behind the snapshot, mount the snapshot volume, so that it can be accessed like any other mounted device.

Here is a simple exercise to verify that the snapshot is functional: write to the original volume—that is, modify an existing file or add/remove a file. The original data for those files will be present on the mounted snapshot. If a new file is added/removed from the original volume, it will not be present on the snapshot. Note that the same logic applies if the snapshot data is modified. The original volume will remain unaltered:

```
$ dd if=/dev/zero of=/home/petros/test.file count=65536
65536+0 records in
65536+0 records out
33554432 bytes (34 MB) copied, 2.95349 s, 11.4 MB/s
$ ls /home/petros/
Desktop  Downloads  Music      Public     test.file
Documents  drvadm    Pictures   Templates  Videos
$ ls /mnt/VolGroup/rootsnapshot/home/petros/
Desktop  Downloads  Music      Public     Videos
Documents  drvadm    Pictures   Templates
```

Using the `lvdisplay` command, you now can observe that more space has been allocated for the snapshot volume. The value for the “Allocated to snapshot” field has increased to 0.24%:

```
$ sudo lvdisplay /dev/VolGroup/rootsnapshot
--- Logical volume ---
LV Name                /dev/VolGroup/rootsnapshot
VG Name                VolGroup
LV UUID                kAc3Iq-Gn3e-pBWs-KC9V-bFi8-0fHr-SsdRLR
LV Write Access        read/write
LV snapshot status     active destination for /dev/VolGroup/lv_root
LV Status              available
# open                 1
LV Size                5.51 GiB
Current LE             1410
COW-table size         500.00 MiB
COW-table LE           125
Allocated to snapshot  0.24%
Snapshot chunk size    4.00 KiB
Segments               1
Allocation              inherit
Read ahead sectors     auto
- currently set to    256
Block device           253:3
```

Removing a snapshot is almost as simple as creating it. First, unmount the snapshot, and then use the `lvremove` command to remove the LV from the VG:

```
$ sudo umount /mnt/VolGroup/rootsnapshot/
$ sudo lvremove /dev/VolGroup/rootsnapshot
```

In some versions of various Linux distributions, including Red Hat Enterprise Linux (also the latest beta release of RHEL 6), CentOS and even SUSE Linux, there exists a known problem when attempting to remove or deactivate logical volumes. Unable to remove the LV, the following error message will be returned: `Can't remove open logical volume "rootsnapshot".` If `dmsetup info -c rootsnapshot` is invoked on the command line, the status of the LV will be returned and it will confirm the error message. To work around this, use the `dmsetup` command followed by the `lvremove` command. Confirm that the LV has been removed with the `lvdisplay` command:

```
$ sudo dmsetup remove /dev/mapper/VolGroup-rootsnapshot-cow
$ sudo dmsetup remove /dev/mapper/VolGroup-rootsnapshot
$ sudo lvremove /dev/VolGroup/rootsnapshot
Logical volume "rootsnapshot" successfully removed
```


In the worst-case scenario, if it is found that space is running out on the snapshot, the LV always can be resized dynamically to a larger capacity.

Best Practices

In some cases, it is advised to ensure that enough storage space is allocated for the snapshot or (as discussed below) a backup directory that will contain all of the archived snapshot data for restoring purposes. To extend an existing volume group, a new PV needs to be labeled. To do so, identify the physical storage device, and using `fdisk`, `sfdisk` or `parted`, create the desired partition size. Verify the partition by reading back the partition table. Then, continue to create the PV:

```
$ sudo sfdisk -l /dev/sde
```

```
Disk /dev/sde: 261 cylinders, 255 heads, 63 sectors/track
Units = cylinders of 8225280 bytes, blocks of 1024 bytes,
        counting from 0
```

Device	Boot	Start	End	#cyls	#blocks	Id	System
/dev/sde1	0+	260	261-	2096451	83	Linux	
/dev/sde2	0	-	0	0	0	0	Empty
/dev/sde3	0	-	0	0	0	0	Empty
/dev/sde4	0	-	0	0	0	0	Empty

```
$ sudo pvcreate /dev/sde1
Physical volume "/dev/sde1" successfully created
```

Append a newly labeled PV to an existing VG with the `vgextend` command:

```
$ sudo vgextend VolGroup /dev/sde1
Volume group "VolGroup" successfully extended
```

If at some point the PV needs to be removed from a VG, use the `vgreduce` command followed by the names of the VG and the PV:

```
$ sudo vgreduce VolGroup /dev/sde1
```

If the VG is being extended for the purpose of creating a backups directory to archive routine snapshots, following the normal `lvcreate` procedure, define the name, size and VG for the desired LV. Then, format the LV with a filesystem, and for file I/O accessibility, mount it to a directory path:

```
$ sudo lvcreate --name backups --size 1G VolGroup
Logical volume "backups" created
$ sudo mke2fs -j /dev/VolGroup/backups
```

dmsetup(8)

`dmsetup(8)` is a low-level tool used to manage logical devices that use the device-mapper driver. The LVM2 user-space toolset relies heavily on the device-mapper kernel module and support library.

```
$ sudo mkdir -p /mnt/VolGroup/backups
$ sudo mount /dev/VolGroup/backups /mnt/VolGroup/backups
```

When the snapshot has been created, an archive can be made with the `tar` command, located in the newly created backups directory:

```
$ sudo tar -pczf /mnt/VolGroup/backups/rootsnapshot.tar.gz
->/mnt/VolGroup/rootsnapshot
```

In an event of failure or if older revisions of files need to be retrieved, the archived snapshot can be used to restore the original data contents. This is an extremely ideal backup strategy when running a high-availability production environment. No downtime is required. Although this backup does not necessarily need to be written to a file, using the `tar` or `dd` commands, the snapshot can be written directly to another physical storage device, including a tape drive:

```
$ sudo tar -cf /dev/st0 /mnt/VolGroup/rootsnapshot
```

Summary

LVM2 comes prepackaged with some of the more common Linux-based distributions. In some cases, it even is used as part of the default filesystem layout. Its snapshot feature is one of those lesser-known treasures that really can be used to one's advantage, ranging from personal to larger-scale environments. All it takes is a little time, a little knowledge and a plan on design, deployment and configuration. ■

Petros Koutoupis is a full-time Linux kernel, device driver and application developer for embedded and server platforms. He has been working in the data storage industry for more than six years and enjoys discussing the same technologies.

Resources

LVM HOWTO: tldp.org/HOWTO/LVM-HOWTO/snapshotintro.html

Logical Volume Manager (Wikipedia): [en.wikipedia.org/wiki/Logical_Volume_Manager_\(Linux\)](http://en.wikipedia.org/wiki/Logical_Volume_Manager_(Linux))

Snapshot (Wikipedia): [en.wikipedia.org/wiki/Snapshot_\(computer_storage\)](http://en.wikipedia.org/wiki/Snapshot_(computer_storage))

LVM2 Project Page: sourceware.org/lvm2

LVM2 Wiki: sources.redhat.com/lvm2/wiki

Known `lvremove` Bug (original no.): https://bugzilla.redhat.com/show_bug.cgi?id=577798

Known `lvremove` Bug for RHEL 6: https://bugzilla.redhat.com/show_bug.cgi?id=638711

Linux Swap Space

Swap space isn't important, is it? Swap space just slows you down—or does it? Discover some little-known facts about your operating system's virtual memory that may change the way you think about swap. TONY KAY

When it comes to system administration, one of the earliest decisions to be made is how to configure swap space. Many readers already are thinking they know what to do: throw in as much RAM as you can afford and configure little or no swap space. For many systems with a lot of RAM, this works great; however, very few people realize that Linux makes this possible by using a form of memory accounting that can lead to system instabilities that are unacceptable for production environments. In this article, I explain the fundamentals of Linux's swap system and show how to configure swap space for optimal stability and performance.

Linux is a demand-paged virtual memory system: all memory is broken up into pages—small equal-size chunks of a few kilobytes—and most of these chunks can be swapped (or “paged”) in or out of RAM as demand dictates (some pages are locked and can't be swapped). When a running process requires more RAM than is available, one or more pages of RAM that have not been used recently are “swapped out” to make RAM available. Similarly, if a running process requires access to RAM that previously has been “swapped out”, one or more pages of RAM are swapped out and the previously swapped-out RAM is swapped in. All of this happens behind the scenes without the programmer having to worry about it.

The filesystem cache, program code and shared libraries have a filesystem source, so the RAM associated with any of them can be reused for another purpose at any time. Should they be needed again, Linux can just read them back in from disk.

Program data and stack space are a different story. These are placed in anonymous pages, so named because they have no named filesystem source. Once modified, an anonymous page must remain in RAM for the duration of the program unless there is secondary storage to write it to. The secondary storage used for these modified anonymous pages is what we call swap space. Figure 1 shows a typical process' address space.

This immediately should clear up some common myths:

1. Swap space does not inherently slow down your system. In fact, not having swap space doesn't mean you won't swap pages. It merely means that Linux has fewer choices about what RAM can be reused when a demand hits. Thus, it is possible for the throughput of a system that has no swap space to be lower than that of a system that has some.
2. Swap space is used for modified anonymous pages only. Your programs, shared libraries and filesystem cache are never written there under any circumstances.

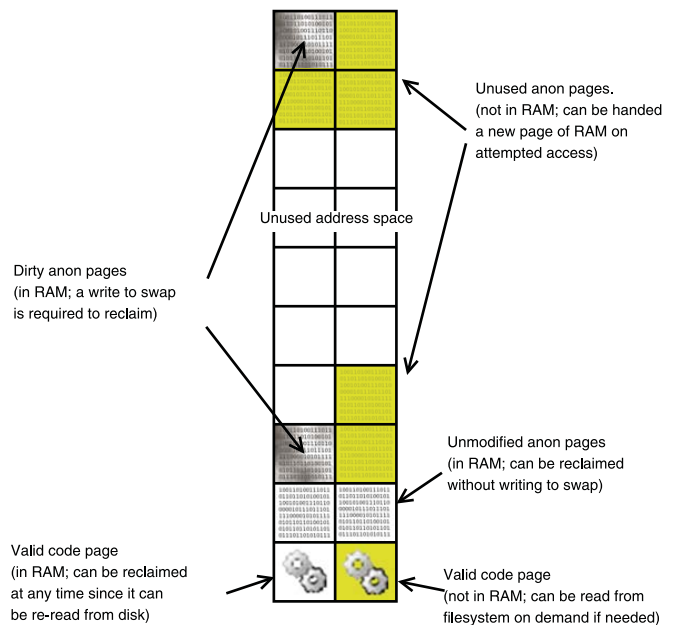


Figure 1. A typical process address space, broken into pages. Some of the pages have no valid mapping to virtual memory. Of the ones that do, many of them (shown with a yellow background) are not given RAM until the program tries to use them.

3. Given items 1 and 2 above, the philosophy of “minimization of swap space” is really just a concern about wasted disk space.

In some demand-paged virtual memory systems, the operating system refuses to hand out anonymous pages to programs unless there is sufficient swap space on which to store modified versions of those pages (so the RAM can be reused while the program remains active). The accounting roughly says that VM size = swap size. This provides two guarantees: that programs have access to every byte of virtual memory they allocate and that the OS always will be able to make progress because it can swap out one process' pages for another.

The problem with this is twofold. First, programs often ask for more memory than they use. The most common case is during a process fork, where an entire process is duplicated using copy-on-write anonymous pages. (Copy-on-write is a mechanism by which two processes can “share” a private writable page of RAM. Either of the processes can read the page, but the OS is required to

resolve write conflicts by giving the writer a new copy of the page so as not to conflict with the other. This prevents the kernel from having to copy data unnecessarily.) Second, being able to write all of the anonymous pages to a swap device implies you are never willing to swap out pages that came from the filesystem (that is, you're not willing to allocate a filesystem page to an anonymous page such that the filesystem page may have to be swapped in later). Such systems typically require an over-provisioning of swap space in order to work properly.

Solaris relaxed this by allowing a certain amount of RAM to be considered in the allocation accounting for anonymous pages (VM size = swap size + percentage of RAM size). This reduced the need for swap space while still maintaining stability. If there wasn't sufficient swap space for modified anonymous pages, the ones currently in RAM simply could stay there while code and filesystem cache pages were reused instead.

Linux took this one step further and relaxed the accounting rules themselves, so that it tries to track memory "in use" (the non-yellow pages in Figure 1), as opposed to memory that has been promised via allocation requests. This works reasonably well because:

1. Many anonymous pages never get used, particularly the rarely copied copy-on-write pages generated during a fork.
2. Filesystem-based pages can be swapped when memory gets low.
3. The natural slowdown due to swapping of program code and shared library pages will discourage users from starting more than the system can handle.

It's not unlike airlines handing out reservations for seats. On average, a certain percentage of customers don't show for flights. So, overcommitting on reservations ensures that they will fly with a full plane and maximize their profit.

Similarly, Linux overcommits the available virtual memory in an attempt to maximize your return on investment in RAM and disk space. Unfortunately, if the overcommitment turns out to have been a mistake, it *kills a (seemingly) random process*.

To be fair, the algorithm is careful when it *knows* it is running low on memory, but this is effective only if the growth in *VM allocation* roughly matches *VM use*. In other words, if a program allocates a lot of memory and immediately starts writing to the allocated pages, the algorithm is pretty good about keeping things in check. If a process allocates a lot of virtual memory but does not immediately use it (which is a common case with Java Virtual machines, databases and other production systems), the algorithm may hand out dramatically more virtual memory than it can back up with real resources.

Additionally, many programs can handle a refusal for more memory gracefully, for example, databases have tunable parameters that tell them how much RAM to use for specific tasks. Other programs might contain something like:

```
buffer = allocate_some_memory(10 MB)
if buffer allocation ok
```

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	MICROWAY, INC. www.microway.com	C2, C4
2NDQUADRANT LTD. www.2ndquadrant.co.uk	47	MIKRO TIK www.routerboard.com	5
ABERDEEN, LLC www.aberdeenninc.com	3	POLYWELL COMPUTERS, INC. www.polywell.com	78, 79
ARCHIE MCPHEE www.mcphree.com	79	SAINT ARNOLD BREWING COMPANY www.saintarnold.com	78
ASA COMPUTERS, INC. www.asacomputers.com	17	SCALE www.socallinuxexpo.org	49
EMAC, INC. www.emacinc.com	57	SERVERBEACH www.serverbeach.com	7
EMPERORLINUX www.emperorlinux.com	33	SHAREPOINT www.sptechcon.com	9
GENSTOR SYSTEMS, INC. www.genstor.com	59	SILICON MECHANICS www.siliconmechanics.com	25, 37
IXSYSTEMS, INC. www.ixsystems.com	C3	SIOS TECHNOLOGY CORP www.steeleye.com	41
LINODE, LLC www.linode.com	43	SOUTHWEST DRUPAL SUMMIT www.swdrupalsummit.com	31
LOGIC SUPPLY, INC. www.logicsupply.com	23	TECHNOLOGIC SYSTEMS www.embeddedx86.com	11
LULLABOT www.lullabot.com	19	UTILIKILTS www.utilikilts.com	79

ATTENTION ADVERTISERS

April 2011 Issue #204 Deadlines

Space Close: January 24; Material Close: February 1

Theme: Web Development

BONUS DISTRIBUTIONS: Cloud Connect, Posscon, Texas Linux Fest

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

```

    sort_using(buffer)
else
    do_a_slower_thing_that_uses_less_memory

```

But, Linux may tell such a program that it can have the requested memory, only to kill something in order to fulfill that commitment.

Fortunately, there is a kernel-tuning parameter that can be used to switch the memory accounting mode. This parameter is `vm.overcommit_memory`, and it indicates which algorithm is used to track available memory. The default (0), uses the heuristic method and overcommits the virtual memory system. If you want your programs to receive appropriate out-of-memory errors on allocation instead of subjecting your processes to random killings, you should set this parameter to 2.

Most Linux systems allow for tuning this parameter via the `sysctl` command (which does not survive reboot) or by placing it in a file that is applied when the system boots (typically `/etc/sysctl.conf`). To make the parameter permanent, add this to `/etc/sysctl.conf`:

```
vm.overcommit_memory=2
```

Now for the slightly harder part. With `vm.overcommit_memory` set to 2, Linux will no longer hand out anonymous pages unless it knows it has a place to store them in RAM or on swap space. So, you'll have to configure enough swap to cover it, or you won't fully utilize your RAM, because it will get reserved for things that never end up being used. The amount is the tough part. You either have to estimate the anonymous page space requirements for your system's common load, or you need to be conservative and configure a lot of it.

The classic recommendation on systems that do strict VM accounting vary, but most of them hover around a "twice the amount of RAM" figure. That number assumes your memory mostly will be filled with a bunch of small interactive programs (where their stack space is possibly their largest memory demand).

Say you're running a Web server with 500 threads, each with 8MB of stack space. That stack space alone is going to require that you have 4GB of swap space configured for the memory accountant to be happy.

Disk is cheap, so I typically start with the "twice RAM" figure. A 16GB box gets 32GB of swap. I fully expect this is overkill for my load, but disk performance considerations (lots of separate heads) mean I usually have more space than I can use anyway.

Next, I monitor system behavior. Remember, the swap space is for accounting; I don't want to see much I/O happening to it. A small amount of I/O on the swap partition(s) of a busy system is not a problem until overall throughput decreases, at which point you need more RAM or fewer programs.

Too little swap space definitely can be a problem, either because Linux denies requests for memory that can be served easily, or because idle dirty anonymous pages end up effectively locked in memory and might remain so for a very long time.

Performance indicators:

- Superb: no swap space even has been allocated (the `free` command shows 0 swap in use), and RAM usage is low. Unless you benefit from a huge filesystem cache, you may have spent

too much on RAM. Run more stuff.

- Great: swap space is allocated, but there is almost no I/O on the swap partition.
- OK: swap space is allocated, and there is some I/O on the swap partition. System throughput is still OK. For example, the CPUs are busy, and most of that is in the User or Nice categories. If you see CPU Wait, it indicates a disk bottleneck (possibly on swapping), and system time could mean the OS is frantically looking for memory to reuse (page scanning).
- Not OK (too much swapping): lots of I/O on the swap partition. CPU is spending a lot of time in Sys or Wait, and swap disk service times exceed the average disk access times.
- Not OK (too little swap space): system is running very well, but new programs refuse to start due to lack of virtual memory.

I typically have the `sysstat` package installed on my CentOS systems and configure the `/usr/lib64/sa/sa1` script to collect all system data (including disk I/O) so I can analyze it over time. My crontab entry for this is:

```
*/* * * * * root /usr/lib64/sa/sa1 -d 240 1
```

which gathers data over a four-minute time span every five minutes. You can analyze the resulting data using a utility called `kSar` (ksar.atomique.net) or at the command line. `kSar` has the advantage of making graphs that compare one metric to another.

You can check your historical usage by running `sar` against one of these data collection files. For example, to view data from the 2nd of this month:

```
# sar -f /var/log/sa/sa02
```

The most direct measure of swapping is reported from `sar -W`:

```
# sar -W -f /var/log/sa/sa02
00:00:01      pswpin/s  pswpout/s
00:10:01          0.00      0.00
...
```

The raw page numbers are a good start. If nonzero, you're doing some swapping. How much is acceptable is more difficult to judge. You'll need to examine the I/O on your swap partition. To do this, you may need to figure out the device major/minor numbers of your swap device. For example, if you swap to `/dev/sdb1`, use:

```
# ls -l /dev/hdb1
brw-r----- 1 root disk 8, 17 Dec  1 14:24 /dev/sdb1
```

to find that your device numbers are 8/17. Many versions of `sar` will report disk I/O by major/minor number. The command to look at the gathered disk data is:

```
# sar -d -f /var/log/sa/sa02
23:15:01  DEV          \
          tps      rd_sec/s  \
          wr_sec/s  avgrq-sz  avgqu-sz  \
          await     svctm     %util     \
23:15:01  dev8-17      \
          0.00     0.00      \
          0.00     0.00     0.00      \
          0.00     0.00     0.00
```

If you have a lot of disks, use `egrep` to see only the header and device:

```
# sar -d -f /var/log/sa/sa02 | egrep '(DEV|dev8-17)'
...
```

Any swapping could be bad, but if you're seeing the `avgrq-sz` (average size in sectors of the requests that were issued to the device) below 1 and an `await` time that roughly matches `svctm` (average service time in milliseconds for I/O requests that were issued to the device), you may be okay. Check other indicators of system performance, such as CPU waiting for I/O and high system CPU times.

Figures 2 and 3 show some custom graphs generated with `kSar` that make it easy to see things more clearly.

Figure 2 is a graph comparing the amount of swap space used to the amount of time the CPU spent waiting on I/O. There are spikes of I/O wait, but they don't follow any sort of pattern with the swap use. However, this system deserves more attention because there is an I/O bottleneck of some sort. It is likely that this system is underperforming, but it probably is not a swapping issue (in this case, the I/O waits were heavy database request loads).

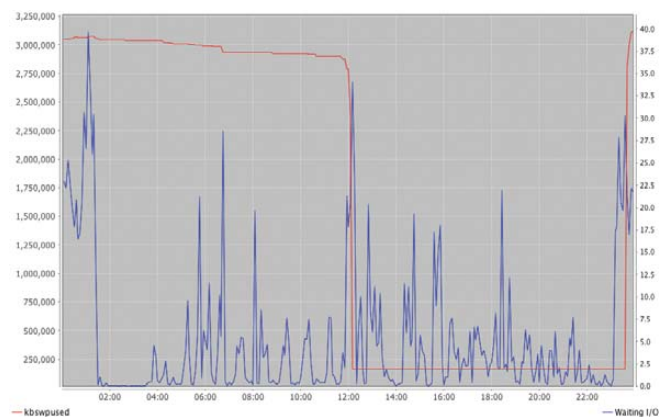


Figure 2. This graph of swap use vs. CPU I/O wait contains no real correlation between a nonzero use of swap and a performance degradation.

Figure 3 shows a graph comparing writes to the swap device vs. I/O wait (over a daytime time interval). It is a bit hard to see, but the red line for I/O is actually zero across the whole of the time span. This indicates that the I/O wait had nothing to do with swapping. Note that this does *not* mean no swapping of any sort occurred. Non-anonymous pages could have been reclaimed and reloaded.

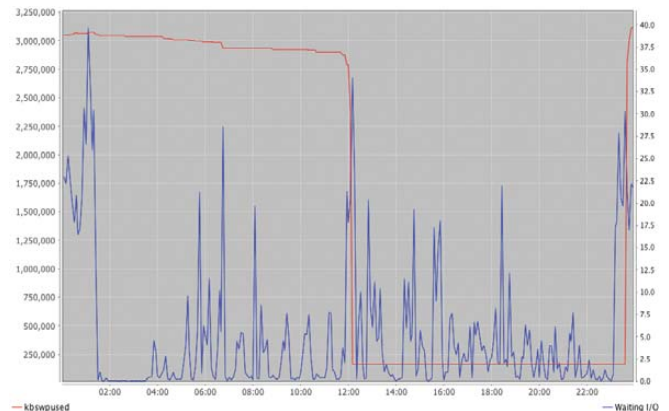


Figure 3. A Graph of I/O Writes vs. CPU I/O Wait

If you are lucky enough to have all of your data on one partition and all of your code on another, you can analyze this sort of paging by watching the reads on your "code" partition. The reads will indicate program loading at startup (which may be infrequent on your systems) and also paging due to memory pressure. If the reads correspond to large I/O waits, you may benefit from more RAM.

The final parameter I want to talk about can affect the responsiveness of certain loads. Have you ever left your system idle for a few hours, only to return to find that it takes a long time to start responding again? Here is probably what is happening: while you were gone, a cron job (like `updatedb`) scanned the filesystem, and because you weren't using the computer, all of the code pages for your "inactive" programs were thrown out in favor of disk cache! There is a kernel parameter called `vm.swappiness` that will affect the tendency of the system to throw out code pages instead of others.

The default value of `vm.swappiness` is 60 on most systems. Changing this parameter to 0 will prevent the kernel from throwing out code pages for filesystem caching and will prevent your desktop from becoming zombie-like due to an unmonitored filesystem scan.

The default value is fine for production systems, because any services that are heavily used will be kept in memory due to their activity. This allows the file cache to expand over anything that is truly unused and generally will work better. It may be advantageous to tune this if you are pretty certain that the majority of memory for programs should be kept (for example, you run a database that does most of its own caching). It might also help to increase this parameter to 100 on systems that have a set of very active programs.

As with any performance parameter, it helps to have some benchmarks that simulate your real workload so you can measure the effects.

Tuning these parameters will lead to a virtual memory system that provides a more stable foundation, particularly for production systems running programs that consume large amounts of memory. ■

Tony Kay has been working with UNIX and Linux systems for more than 20 years. He lives in Bend, Oregon, where he manages large Linux systems and develops mobile applications.

Clonezilla: Build, Clone, Repeat

Finally, open-source cloning that delivers. JERAMIAH BOWLING

The practice of cloning is a time-tested standard operating procedure in many IT shops. It can assist in migrations and building test/lab environments, and it enhances system recovery processes. Due to the tremendous utility of cloning, there also is a healthy market of commercial cloning tools, each with their own unique abilities. Unfortunately, most of these commercial tools offer limited support for Linux. Enter Clonezilla, an open-source suite of cloning tools developed by Taiwan's National Center for High-Performance Computing (NCHC) software lab under the GPL that works impeccably with Linux distributions. Clonezilla boasts high-performance cloning, simplified deployment and support for a multitude of system types (Windows, Linux and VMware), all of which make it a worthy competitor to its commercial counterparts.

In this article, I describe two common cloning scenarios where Clonezilla performs serviceably in place of commercial cloning suites. In the first scenario, I install Clonezilla Server Edition (SE) and Diskless Remote Boot Linux (DRBL) on a Linux server for use as a central imaging solution in the enterprise. In the second scenario, I focus on Clonezilla's disaster recovery (DR) abilities using the Live version to create a custom recovery solution for a critical system.

For the first scenario, I have chosen to deploy DRBL and Clonezilla on CentOS 5.5 server, although both programs work on a multitude of distributions. Prior to selecting server hardware, you need to make two decisions. The first is how many networks you want to use with DRBL. It is recommended to run DRBL services on a nonproduction network segment. As a result, DRBL is best run on a server with more than one NIC, specifically where one NIC is connected to your production network and a second NIC is connected to a dedicated cloning network. Clonezilla is scalable, so you can have multiple NICs in a server dedicated to different imaging segments. This is a common network

```

root@ImageServer:~
File Edit View Terminal Tabs Help
[1] 100
*****
How many DRBL clients (PC for students) connected to DRBL server's ethernet network interface eth
1 ?
Please enter the number:
[2] 12
*****
The final number in the last set of digits in the client's IP address is "111".
We will set the IP address for the clients connected to DRBL server's ethernet network interface
eth1 as: 192.168.162.100 - 192.168.162.111
Accept ? [Y/n] y
*****
OK! Let's continue...
*****
The Layout for your DRBL environment:
*****
      NIC      NIC IP      Clients
-----
DRBL SERVER
  +-- [eth0] 10.11.12.65 +-- to WAN
  +-- [eth1] 192.168.162.130 +-- to clients group 1 [ 12 clients, their IP
                               from 192.168.162.100 - 192.168.162.111]
*****
Total clients: 12
*****
Press Enter to continue...

```

Figure 1. drblpush Script Displaying the Network Design

design for test labs and classroom environments. You can use a single NIC with an alias for your imaging network, but you'll be running two IPs on one card, which is not recommended. Let's keep deployment simple by using two NICs in the server, with one plugged in to the production and the second plugged in to the imaging network.

After deciding on your network configuration, you need to decide where you want to store the cloned images. If you want to use your DRBL/Clonezilla server for storage, you may need to plan for a larger hard drive or storage array to accommodate the size of your image library. You also may have an existing large-capacity high-speed storage network, like a SAN, that performs better over the network when cloning multiple images at once. In that case, you can use remote mountpoints with your networked storage to house the images. I've kept it simple here and used the directly attached

hard drive in my server, as it has enough room for everything.

With the hardware in place, install CentOS using an installation image from one of the distribution's mirrors. Select whatever install options you require, but it is recommended that you disable any firewalls and SELinux. Once the CentOS installation is complete, download and install the DRBL rpm package from the project's SourceForge site. When the rpm completes installation, open a terminal console and run the following command to finalize the DRBL install and create the PXE image client files:

```
/opt/drbl/sbin/drblsrv -i
```

After entering the above command, you are presented with a series of prompts. For our purposes, you can accept all of the defaults, but make sure to read them to understand what is occurring

during the DRBL installation. Because I used the i386 version of the CentOS distribution, I had to upgrade to the i686 version before DRBL would begin the installation. When this process is complete, enter the following command to configure the DRBL server:

```
/opt/drbl/sbin/drblpush -i
```

This launches the DRBL configuration script in interactive mode. As with the `drblsrv` script, you can accept most of the default options. When prompted to configure the network, assign `eth0` as the public Internet (WAN) connection. After that, any other active NICs found by the script are placed on their own DRBL segment. You then are prompted for the initial number of the IP address you want to use. This is the starting number your DRBL clients will receive in the last octet of their IP address from DHCP. At the next prompt, enter the number of DRBL clients you want to use with this server. This number will increment the IP addresses used on the server. For example, if you entered 100 as the first number and 12 as the second, you can use up to 12 client machines assigned with the IP addresses 192.168.162.100–192.168.162.111 with your imaging server. When all prompts have been answered, you are presented with a visual representation of your DRBL network environment (Figure 1). Press Enter to continue. When prompted for the DRBL and Clonezilla modes, select the defaults of Full DRBL and Full Clonezilla mode. It then will ask you for the directory to store images. This directory can be local or remote storage. Based on our earlier decision, leave the default as a local folder named `/home/partimag`. After these prompts, you can accept the rest of the defaults. When you have answered all the prompts, the script will create instances of the DRBL client (based on your distro) for each IP address allotted from the configuration. DRBL configuration is now complete.

Next, run the following command from a console to launch the Clonezilla setup script:

```
/opt/drbl/sbin/dcs
```

Select All Clients from the first screen.

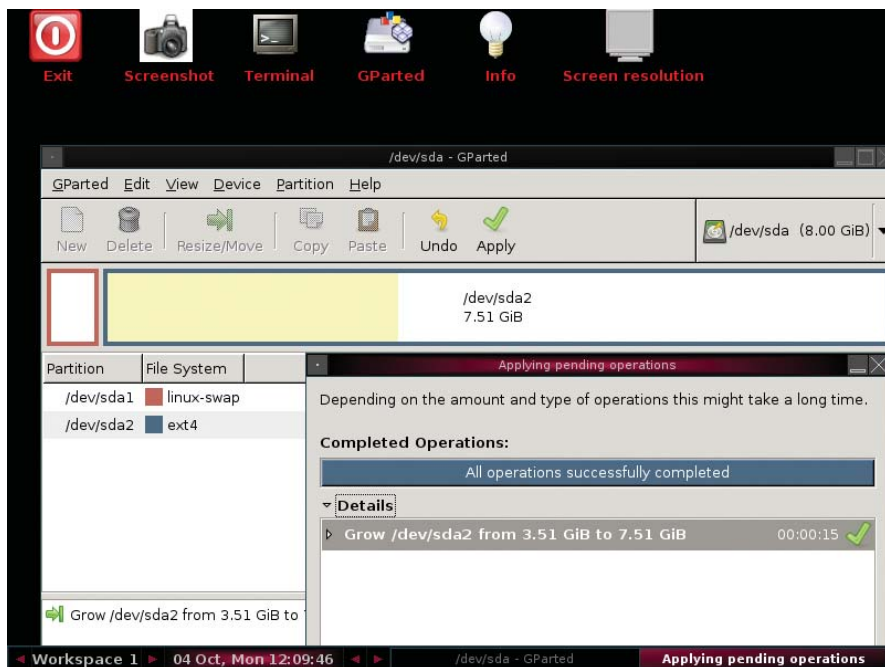


Figure 2. GParted in Action

Set the mode to Clonezilla-start and then Beginner mode. On the resulting screen, choose select-in-client Save/Restore in the Client. This gives you the flexibility to select either operation at the client. You can accept the rest of the defaults until the script exits. You may want to experiment with these options further, as they can help streamline the cloning process. To make a configuration change to your DRBL or Clonezilla server, simply rerun the `drblsrv`, `drblpush` or `dcs` script again. If you make a change that involves the PXE setup, it always is a good idea to blow away any existing PXE image profiles (a prompt in the `drblpush` script) to make sure your new settings override any previous ones.

With the server in place, let's move on to creating and cloning the base desktop image. Many things must be taken into account when building the base image. First, partition/use only the minimum amount of storage space necessary to fit on the smallest drive to which you plan to clone. Clonezilla supports restoring a cloned image to a larger drive, but it will not restore to a smaller drive than the one found in the image. If your target disk is larger than the image being restored, you can use free partition tools like GParted or

PartedMagic to expand the restored OS to use some or all of the free space on the local disk. GParted is nice because it can be added to a PXE server or run from a live CD (Figure 2).

The second caveat is to use like-to-like hardware between the base image and the target machines as much as possible. Differences in hardware from the base to target machines can cause a multitude of issues when cloning. Some items, such as memory and hard drives, are not critical, but switching system components, such as motherboards and processors, can cause a host of issues. For most people, this is not a problem, as many administrators already work highly standardized environments.

Third and last, build your OS on a pristine machine or "bare metal". If experience has taught me anything, it is to start with a clean slate. After installing the desktop OS on your base image, install only those programs absolutely necessary to your deployment, and avoid installing any antivirus, backup agents or desktop management clients before cloning. Anything that requires a unique identifier should be left off the base image. After installing all required software, make sure the machine shuts down cleanly. For my base image, I have built a vanilla installation

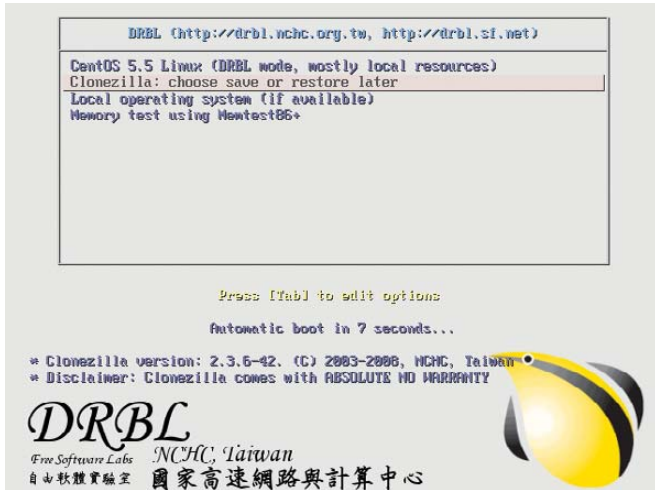


Figure 3. The GRUB screen pushed by the DRBL server via PXE.

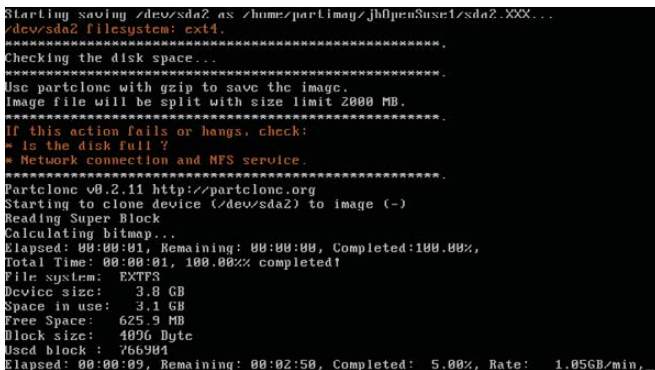


Figure 4. Capturing the Disk to Image

of openSUSE with OpenOffice.org for redeployment on my target machines.

After preparing the OS on your base machine and powering it down, you need to make it talk to the DRBL server. You can accomplish this by using the Pre-Boot eXecution Environment (PXE, pronounced “pixie”). PXE provides a shell for a thin OS that runs in local memory and provides direct access to local data outside the installed OS—a prerequisite for most cloning programs. If not already enabled on your target hardware, PXE can be enabled on most motherboards in the BIOS. Once enabled, connect the base machine to the DRBL network segment, and boot from it as a source by either setting it in the boot order or using a boot menu (usually accessed by pressing a function key prior to the bootloader screen).

Upon booting from PXE, the client should pick up an address from the DRBL

(Figure 4). When the image capture has been completed, power down the base machine.

With your image stored on the server, you can simultaneously clone up to 12 end systems. Restoring a cloned image to a new machine via PXE follows the same procedure as capturing it, with the exception that you need to select the restore-disk option, and if you store multiple images on the server, you need to specify which image

server (also running DHCP) and present you with a GRUB-loader screen (Figure 3). Leave the default selection of “Clonezilla: choose save or restore later” to load Clonezilla into memory via TFTP. Once Clonezilla has loaded, select the following options to capture your base image on the server: device-image, Beginner mode and savedisk. Provide a name for the image and choose the local disk as the source. When prompted, confirm you want to run the script and capture the image. Clonezilla will launch the appropriate cloning program (partclone, partimage, ntfsclone, dd) and begin capturing the disk to image

to restore. Depending on the OS contained in your image, further steps may be needed after the restore. At a bare minimum, you will want to change each newly cloned machine’s hostname. If you are restoring a Microsoft OS, you need to change the Security Identifier (SID) so as not to cause problems in your domain. You can avoid this issue with Windows XP clients and higher by using the sysprep utility pre-clone.

As long as everything went smoothly, you now have a working DRBL and Clonezilla SE server that you can use to mass clone devices on your network as often as needed.

For the second scenario, let’s use the Live version of Clonezilla to create a recovery CD/DVD for a critical system. This is handy for near-line backups and/or off-site storage, but it also can double as a method for mass cloning systems if you lack the network infrastructure to use DRBL and PXE. If you are an OEM builder, you can use Clonezilla in this way to create recovery CDs for inclusion with your systems, like the big PC manufacturers have done for years. Recovery discs also can supplement change management processes—for example, if you have a server that requires high availability that is in need of major software upgrade. By capturing the server in its current state prior to the upgrade, you have a viable rollback plan should the upgrade go awry.

The recovery disc creation process includes two parts: taking a backup image of the machine and then creating

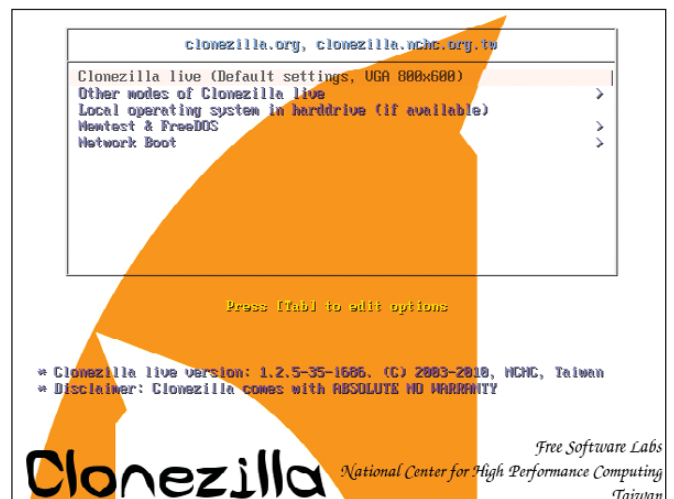


Figure 5. The Clonezilla Live Boot Screen

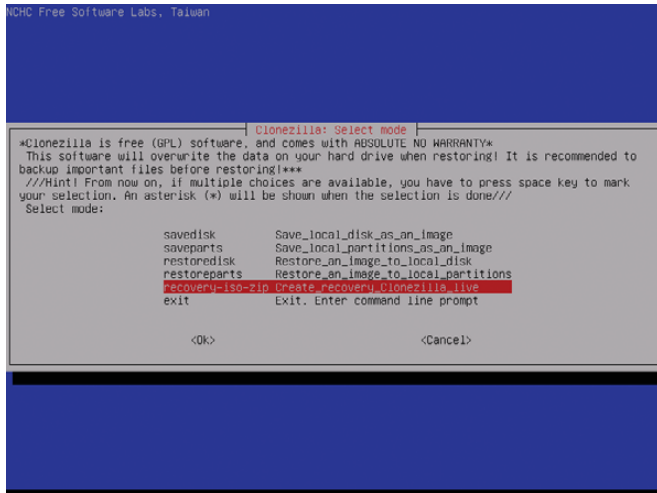


Figure 6. Selecting the Recovery Mode Option in Clonezilla Live

the CD itself. Many of the same caveats apply to capturing an image using the Live version of Clonezilla as they did in the earlier scenario with two major differences. One, the system disk or partition to be restored must fit on a single CD or DVD along with all of the files needed to create the disc. Using a dual-layer DVD means you tap out at approximately 8.54GB. This is not a huge problem because, as mentioned earlier, you can expand the restored partition using GParted or other tools post-clone.

Two, unlike a base system that may be restored multiple times over, you will want to configure this host as much as possible to minimize recovery time. If you suffer total server loss due to a catastrophic event and are forced to use a recovery disc, and you do not want to spend additional time editing configurations. This works well for static servers that don't change much or servers with highly unique and, therefore, time-consuming configurations. You may need to shift this approach for dynamic servers like database servers, but you still can use a recovery disc to reduce recovery time. In this case, build the server, install and configure the database engine (for example, MySQL) without any databases and capture it. Then, if you need to go back to the recovery image, the only additional step needed for full recovery is to restore the database onto the newly restored OS. For my test scenario, I used a DNS server

running on CentOS 5.5 as my critical system.

To get started, download the Clonezilla Live .iso file from SourceForge and burn it to a CD. Place the Clonezilla Live CD in your system and boot from it (Figure 5). At the boot menu, accept the default option for Clonezilla Live.

When prompted,

accept the same Clonezilla options previously selected for capturing an image (Start Clonezilla, device-image and so on), and when prompted for a mountpoint for /home/partimag, select somewhere other than the local disk you want to clone. I used a USB hard drive, so I selected the local_dev option. Leave the defaults on the option "Select the top directory to store the image file" to store it on the root of your chosen

storage device. Complete the selections using the same options chosen earlier for capturing an image. When the clone process is finished, return to the console by pressing Enter and selecting "(3) Start over".

With your image captured, you now can build the recovery disc. Select the same options as above to start Clonezilla (device-image, local-dev, mount USB as /home/partimag), but then select Beginner Mode and "recovery-iso-zip Create_recovery_Clonezilla_Live" (Figure 6). Select the image that was just captured as the image to restore, and confirm that the destination device to be restored matches the mount label of the critical system (that is, sda, hda1). Leave the language and keymap options the defaults. Select .iso as the recovery file type. If prompted that the target .iso file is too large to fit on a CD, answer yes to create the file anyway. Clonezilla will start building the .iso file in the root of the USB drive and include the captured image in the .iso image (Figure 7). At the end of the process, power off the machine using the prompts. Burn the resulting .iso from the USB hard drive to a CD or DVD depending on the size.

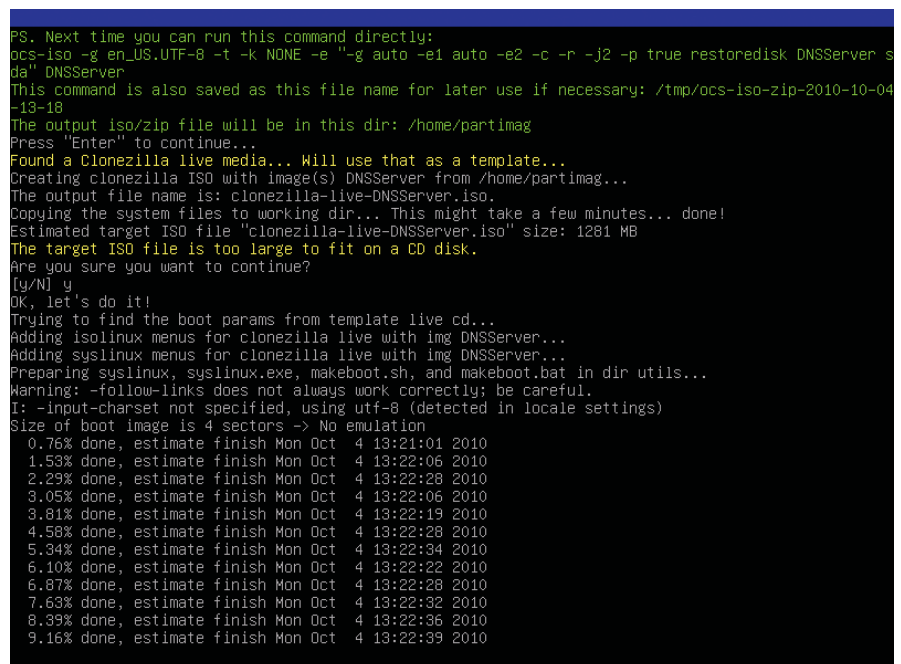


Figure 7. Creating the .iso File

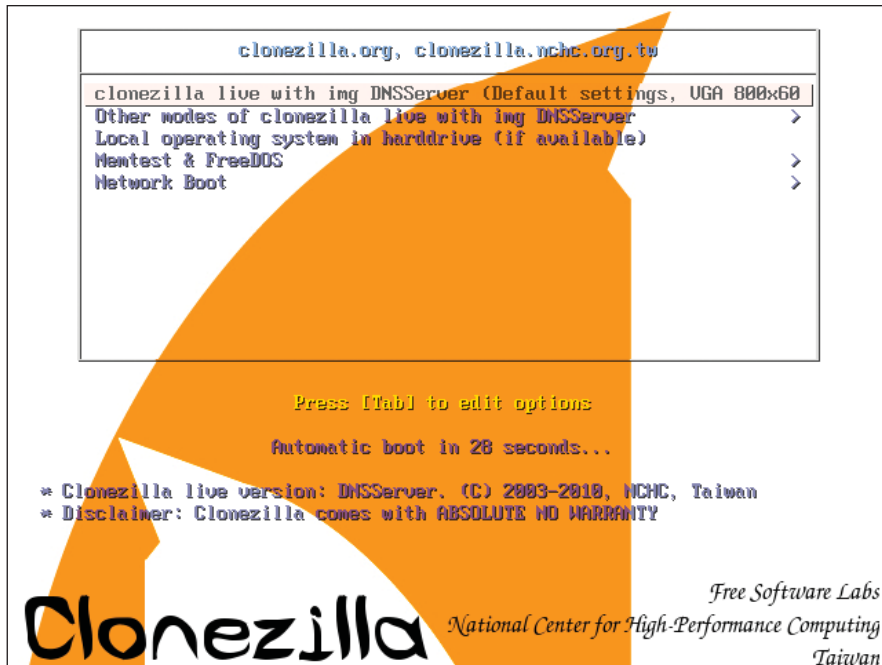


Figure 8. The Recovery Disc Boot Screen (with Image Name)

```
Setting the TERM as linux
*****
Clonezilla image dir: /home/partimag
*****
Shutting down the Logical Volume Manager
. No volume groups found
. No volume groups found
Finished Shutting down the Logical Volume Manager
*****
Activating the partition info in /proc... done!
Getting /dev/sda1 info...
Getting /dev/sda2 info...
*****
The following step is to restore an image to the hard disk/partition(s) on this machine: "/home/part
imag/DNSServer" -> "sda sda2"
WARNING!!! WARNING!!! WARNING!!!
WARNING! THE EXISTING DATA IN THIS HARDDISK/PARTITION(S) WILL BE OVERRITTEN! ALL EXISTING DATA WILL
BE LOST:
*****
Machine: VMWare Virtual Platform
sda (8590MB_VMWare_Virtual_S_No_disk_serial_no)
sda2 (8064MB_ext4(In_VMWare_Virtual_S_No_disk_serial_no)
*****
Are you sure you want to continue? ?
[y/n] _
```

Figure 9. Restoring the Recovery Image

To test the recovery disc, boot from it and select the default option from the bootloader screen, which will default to “Clonezilla Live” with the name of your img file (Figure 8). The recovery disc then will bypass the usual Clonezilla selection screens, immediately launch

the appropriate cloning program and ask for confirmation to overwrite your drive with the image on the disc (Figure 9). If all went well, you now have a good copy of your system to use in case of emergency.

In this article, I’ve walked through the

two most popular uses for Clonezilla, but they are by no means the only ones. One emerging trend is using Clonezilla to capture images of physical systems for redeployment on virtual servers—a process known as physical-to-virtual conversion (P2V). Others have combined Clonezilla Live with the popular System Rescue CD and built a powerful rescue CD called Clonezilla-SysRescCD. There are a multitude of uses for the suite and its components, but for all its utility, Clonezilla does lack some of the nicer features found in the commercial cloning suites. Those missing features include the ability to clone a running system, take differential (delta) images and restore to dissimilar hardware or “bare-metal” restores. Still, those features come at a premium and are not always easy to implement. In the end, Clonezilla is a common-sense tool that does its job well for all types of systems—not just those that don’t end in *nix. ■

Jeremiah Bowling has been a systems administrator and network engineer for more than ten years. He works for a regional accounting and auditing firm in Hunt Valley, Maryland, and holds numerous industry certifications, including the CISSP. Your comments are welcome at jb50c@yahoo.com.

Resources

CentOS: www.centos.org

DRBL: drbl.sourceforge.net

Clonezilla SE and Live: clonezilla.org

GPated: gpated.sourceforge.net

GPated Live:
gpated.sourceforge.net/livecd.php

PartedMagic: partedmagic.com

Clonezilla-SysRescCD:
clonezilla-sysresccd.hellug.gr

Migrate to a Virtual Linux Environment with Clonezilla: www.ibm.com/developerworks/linux/library/l-clonezilla

Managing KVM Deployments with Virt-Manager

The Virtual Machine Manager tools (aka virt-manager and libvirt) make it simple to set up and manage multiple KVM-based virtual machines. MICHAEL J. HAMMEL

KVM is the kernel-based virtual machine, a mainline kernel feature since versuib 2.6.20 for running virtualized guest systems on top of a host system. In less technical terms, this means you can run other operating systems as just another window on your desktop. But, before you can get to the desktop, you need to become familiar with creating and managing virtual machines using the Virtual Machine Manager suite of tools.

Each virtual machine instance, referred to as a VM, has its own virtualized hardware. This includes a chunk of memory and one or more of the available CPUs, network and graphics hardware, virtual disks and a mouse. The Virtual Machine Manager tools, often referred to collectively by the name of the primary GUI application virt-manager, make it possible to bring up the virtual hardware quickly with the virtual CD-ROM device “connected to” an ISO file in order to install an operating system. At work, we run a variety of Linux distributions and

The Virtual Machine Manager tools, often referred to collectively by the name of the primary GUI application virt-manager, make it possible to bring up the virtual hardware quickly with the virtual CD-ROM device “connected to” an ISO file in order to install an operating system.

a couple Windows versions. Each operating system instance is installed with a base configuration that is common to all copies of that instance. This is known as a base image. From these, we make copies, known as clones, in which specific applications are installed. This result of cloning and custom application configuration is what the cloud computing world refers to as virtual appliances.

KVM is one of a number of options for running virtual machines. Xen, VirtualBox and VMware are alternatives with different requirements for setup and use. KVM was chosen for use at my office, because of the simplicity of setup on Fedora systems (the standard Linux development platform at work) and because of the ease of creating and using virtual machines across the internal network. The process discussed here is applicable specifically to using KVM-based virtual machines.

Since KVM is part of the kernel, most modern Linux distributions offer KVM kernels. However, the user-level tools that utilize KVM, such as virt-manager and libvirt, are what most users will be interested in. Check your distribution for availability of ready-to-install packages or grab the source packages from the URLs provided at the end of this article.

Getting Started: Host Hardware Setup

In virtualization parlance, the host system is the real hardware and its operating system that will be the platform on which you will run guest systems. Each guest system is the KVM-based virtual hardware and the OS installed on it. Before you can use a KVM-enabled kernel on your host system, you need to make sure your hardware supports virtualization. To check whether the processor supports virtualization, search for the following CPU flags:

```
egrep "vmx|svm" /proc/cpuinfo
```

Intel processors with VM support will have the vmx flag in the output of this command. AMD processors that support virtual machines use the svm flag. This command will print one line for each CPU core with VM support.

The next step is to configure the host operating system. The first thing to do is install the user-space tools associated with virt-manager:

```
sudo yum install kvm qemu libvirt \
    libvirt-python python-virtinst \
    bridge-utils virt-manager
```

This may install additional packages, but these are the ones critical to using KVM and virt-manager. These package names are specific to Fedora. Check your distribution for proper package names.

After installing the required software, a bridge network device needs to be added to the host system. Using a bridge device allows the VM to function like any other system on your network, using its own IP address. On the host system, the bridge will be assigned an IP address and the default “eth” device will be attached to that. Using a bridged network is a little easier if you disable NetworkManager:

```
# chkconfig NetworkManager off
```

```
# chkconfig network on
# service NetworkManager stop
# service network start
```

NetworkManager is ideal for mobile platforms, but as of Fedora 12, it's not suitable for use with shared networks, such as those used with a host and guest virtualized systems. Although you can run KVM on mobile systems, this article assumes the use of a desktop or server that is unlikely to need to change network configurations often.

Note that the above commands must be run as root. On Fedora, configuration scripts for each network device are located under `/etc/sysconfig/network-scripts`. Assuming there is at least one network card in your system, there will be a configuration file named `ifcfg-eth0` in that directory. Open it and add the following line:

```
BRIDGE=br0
```

Then, comment out the following lines by prefixing each line with the `#` symbol:

```
#BOOTPROTO=none
#DNS1=...
#GATEWAY=...
#IPADDR=...
#NETMASK=...
```

To add the bridge device, create a file called `ifcfg-br0` in that directory, and add the following lines to it:

```
DEVICE=br0
TYPE=Bridge
BOOTPROTO=static
DNS1=...
GATEWAY=...
IPADDR=...
NETMASK=...
ONBOOT=yes
```

The lines with `...` can, in most cases, be the lines that you commented out of `ifcfg-eth0`. Note that the `TYPE` entry is case-sensitive.

Be sure to assign proper values for `DNS`, `GATEWAY`, `IPADDR` and `NETMASK`. This example assumes a static IP address, but if the device will use DHCP, change `BOOTPROTO` and leave these entries out:

```
DEVICE=br0
TYPE=Bridge
BOOTPROTO=dhcp
ONBOOT=yes
```

Note that the host also should have a large amount of disk space for the VM image files. The size of these can vary, but in this article, we'll create 12GB files to support complete Linux distributions on 8GB with 4GB for local user directories. Smaller installation partitions could be used for the operating

system, and user directories can be NFS-mounted if desired. Also, it is possible to place the VM image files on NFS-mounted directories, although this can have serious performance impacts when running the virtual machine.

At this point, the host network can be restarted with the new networking configuration:

```
# service network start
```

The host now is ready for use with virtual machines.

virt-manager and Friends

Before diving into creating virtual machines, it's important to take a look at the related tools and files. Virtual Machine Manager is actually a suite of tools for working with virtual machines:

1. `virt-install`: command-line tool used to install software into a VM image.
2. `virt-clone`: command-line tool used to clone a VM image.
3. `virt-manager`: GUI for starting, stopping and using VM images.
4. `virt-image`: command-line tool used to create a VM-based on an XML description.

The first three of these will be used to create, clone and use virtual machines. The latter is useful for advanced users but is beyond the scope of this article.

The `virt-manager` tools are Python programs that depend on the `libvirt` library to manage virtual machines and `QEMU` to run the virtual machines. Both `libvirt` and `QEMU` offer sophisticated features for a wide variety of tasks. Fortunately, you don't need to know anything about `QEMU` to get started using a VM, and you need to know only very basic



Figure 1. The `virt-manager` wizard can create images for the local host.

information about libvirt.

The virt-manager application uses VNC to connect to remote libvirt daemons and view remote virtual machines on a local display. This means you can launch virt-manager on your system and connect to a VM running on another system across the network. Such connections will require password authentication in most cases, depending on how libvirt is configured. This article takes the simple (and highly insecure) path of using root SSH access to access remote hosts. This works for my company, because the virtual machines are on an insulated networks. This also works if you're behind a well-configured firewall at home, but to be safe, consider further research into the secure (TLS) transport options of libvirt.

Note that virt-manager provides a wizard for creating new virtual machines on the localhost using local or remote media installations (Figure 1). However, this article focuses on the underlying tools virt-install and virt-clone. These tools offer far more power and flexibility when creating virtual machines.

Installing Base Images

With the software installed and the host network configured, it's time to create a base image. The base image is an installation of an operating system into a VM image file. These files can take multiple formats, but the recommended format is qcow2:

```
sudo virt-install --connect qemu:///system \
-n <vm-name> \
-r 512 \
--vcpus=1 \
-f ~/<vm-name>.qcow2 \
-s 12 \
-c <distro-install-image>.iso \
--vnc \
--noautoconsole \
--accelerate \
--os-type linux \
--os-variant generic26 \
--network=bridge:br0
```

Replace <vm-name> with a symbolic name for the virtual machine. At work, we use the distribution name, version and CPU size, such as "fedora11-64". The <distro-install-image> is the name of the ISO image file used to install a Linux distribution.

The man page for virt-install covers the available options in detail. This particular command attaches to the local hypervisor as root (--connect) and sets up a guest virtual machine with 512MB of memory and the maximum number of CPUs it can use (-r, --vcpus). It will create a virtual machine image in the file ~/<vm-name>.qcow2 that is 12GB (-f, -s) and boot the installation media <distro-install-image>.iso. virt-install will start a VNC console on the guest and make it available for use via the host (--vnc), but no connection to it is started by default (--noautoconsole). A connection to it will be made later using virt-manager. The guest machine will run using kernel acceleration if available (--accelerate).

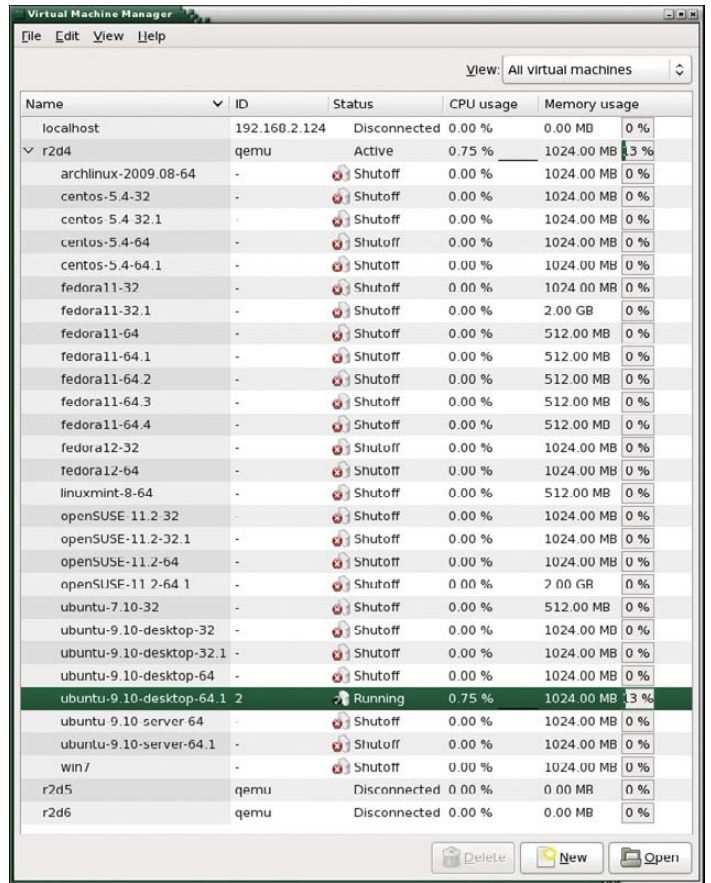


Figure 2. The virt-manager window shows hosts with VNC servers. Connecting to the localhost is automatic.

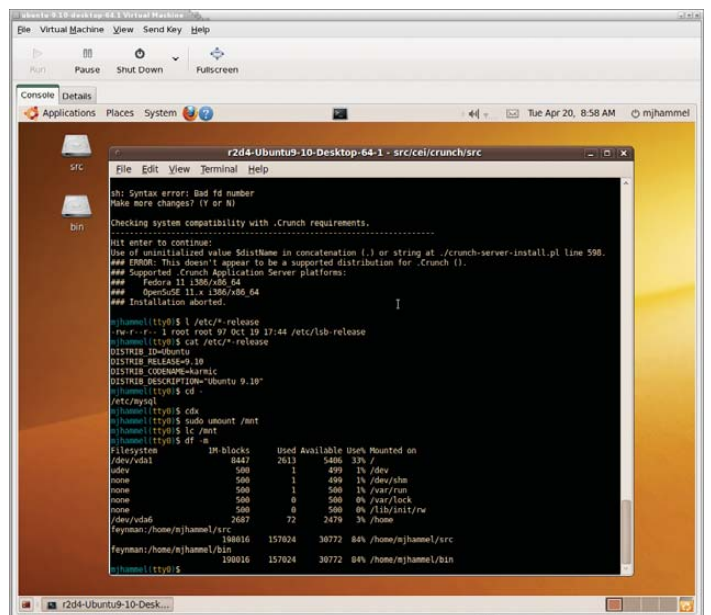


Figure 3. Each OS will set the default size of the console. The console window can be scaled up, but not down, using the View→Scale Display menu.

Clones, however, are useful if you plan on having multiple, independent configurations using the same base OS installation.

The guest will be created using optimized ACPI, APIC, mouse and other hardware configurations (`--os-type`, `--os-variant`) and use the bridged network device (`--network`). Note that the `os_type` has specific valid values. Check the man page for details.

This command will exit immediately while it starts up the VM in the background. To interact with the operating system installation, start the `virt-manager` application. This program will recognize VNC servers on the local network and list them (Figure 2). Double-clicking on one will open a connection to that host and list the available guest systems. Double-clicking on a guest will open a window to the guest (Figure 3).

With the VNC window open, the installation proceeds just as it would when installing on bare metal. Click inside the VM window to activate the guest VM mouse, then hold down `Ctrl-Alt` at the same time to return the mouse to the host desktop. In many cases, you won't need to capture the mouse pointer this way. Instead, just type with the host mouse pointer over the VM console window and keystrokes are passed to the guest VM.

Once the installation completes, a reboot typically is required. A reboot of the VM, that is—a reboot in this VM instance shuts down only the virtual machine, not the host. You must use the Run button in the menu bar of the VNC window to start the VM again manually. After rebooting, be sure to install any updates for the Linux distribution.

libvirt Configurations

Using `virt-install` to create a VM image does two things. It creates an image file, and it creates a configuration (Listing 1) to launch it for libvirt. The configuration file is an XML file found under `/etc/libvirt/qemu`, which should be accessible only by the root user.

If edits are done manually to this file, libvirt should be restarted:

```
sudo service libvirtd restart
```

However, it probably is better not to edit the file and to use the `virsh` command for libvirt to make updates to a VM configuration. If the amount of memory or number of CPUs to use for a VM needs to be lowered, the `virt-manager` Details tab needs to be used rather than using `virsh`. Be sure to exit any instances of `virt-manager` before restarting libvirt.

The base image can be copied to an NFS directory along with the XML configuration file, so that other hosts can make use of it. For another host to use it, copy the XML file to the new host's `/etc/libvirt/qemu` directory, and edit it to point to the NFS mountpoint. Then, restart the libvirt on the new host.

Creating a Clone

The base image is just the core image for creating your VM appliances. Each appliance is a clone of the base image with additional

Listing 1. Sample XML Configuration File for a VM

```
<domain type='kvm'>
  <name>ubuntu-9.04-64</name>
  <uuid>19a049b8-83a4-2ed1-116d-33db85a5da17</uuid>
  <memory>1048576</memory>
  <currentMemory>1048576</currentMemory>
  <vcpu>2</vcpu>
  <os>
    <type arch='x86_64' machine='pc'>hvm</type>
    <boot dev='hd' />
  </os>
  <features>
    <acpi />
    <apic />
    <paef />
  </features>
  <clock offset='utc' />
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
  <devices>
    <emulator>/usr/bin/qemu-kvm</emulator>
    <disk type='file' device='disk'>
      <source
        file='/home/baseimage/ubuntu-9.04-64.qcow2' />
      <target dev='hda' bus='ide' />
    </disk>
    <disk type='file' device='cdrom'>
      <target dev='hdc' bus='ide' />
      <readonly />
    </disk>
    <interface type='bridge'>
      <mac address='54:52:00:42:df:25' />
      <source bridge='br0' />
    </interface>
    <serial type='pty'>
      <target port='0' />
    </serial>
    <console type='pty'>
      <target port='0' />
    </console>
    <input type='mouse' bus='ps2' />
    <graphics type='vnc' port='-1' autoport='yes' />
  </devices>
</domain>
```

applications installed. Base images are also known as “backing stores”, because the base image is used for reads, and the clone image is used when writes occur. In this way, the clone gets updated while the base image remains pristine.

Creating a clone requires that the host libvirt knows about the base image, which means the XML configuration for the base image is in the libvirt configuration directories and libvirt has been restarted. Cloning is performed by the `virt-clone` command:

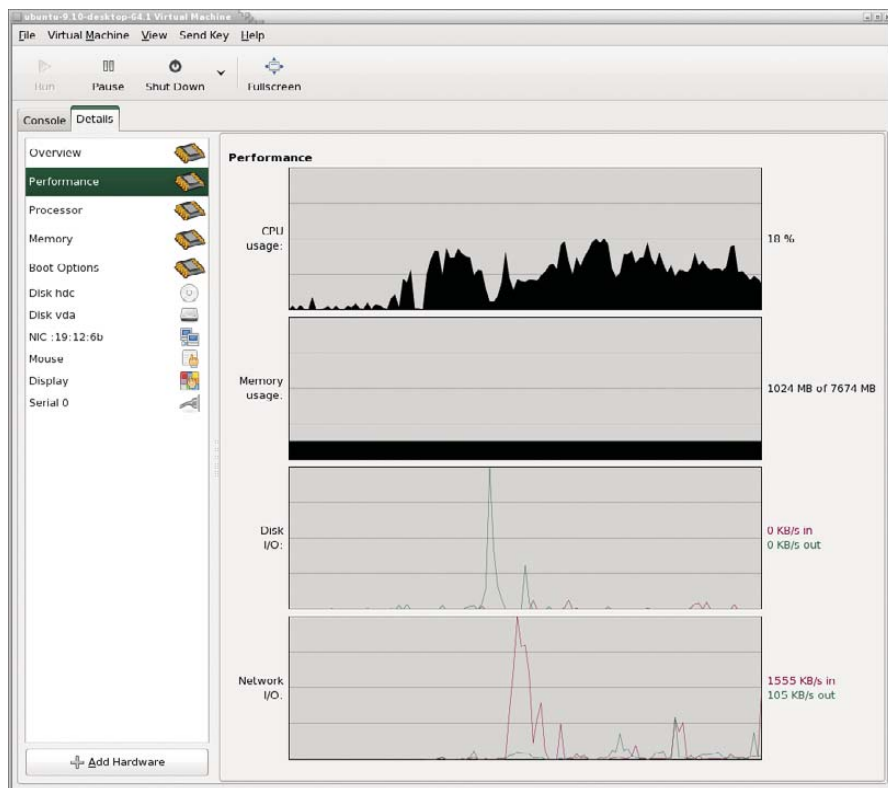


Figure 4. Processor and memory usage can be adjusted dynamically up to the configured maximum for the guest.

```
sudo virt-clone -o <base-image-name> \
-n <clone-image-name> \
-f <path-to-base-image>
```

As with `virt-install`, `virt-clone` options are described in detail in the man page. Using the previous `virt-install` example and the naming scheme described earlier, a clone of a Fedora base image would look more like this:

```
sudo virt-clone -o fedora11-64 \
-n fedora11-64.1 \
-f /home/cloneimages/fedora11-64.1
```

In this example, the clone image name is the same as the base image with an identifying suffix, and the clone image file has the same name as the image and will be created in `/home/cloneimages`. The image name is used to identify the guest VM to `libvirt` and is saved as the “name” element in the XML configuration file for the VM. The image filename is the name of the file where the image is located. This is stored in the “source file” element of the XML configuration file.

Cloning an image can be a system-intensive operation. Although a quad-core CPU with 8GB of memory might handle this just fine, a single-core system may get bogged down if other operations are in progress. Cloning at home may be something to leave for overnight processing.

Clones are not required for working with a VM. You can just as

easily work directly with the base image. Clones, however, are useful if you plan on having multiple, independent configurations using the same base OS installation. In other words, clones are not typically necessary at home, but they may be required when used with clusters of servers.

Summary

`libvirt` provides many features to do more complex things with virtual machines, such as storage, networking and USB/PCI options. All of those features should be accessed using the `virsh` command-line tool. `virt-manager` provides GUI-based wizards for creating new virtual machines and allows dynamic reconfiguration of the boot device while displaying interactive performance graphs (Figure 4). For home users, the best feature of `virt-manager` is its ability to show multiple VM consoles at the same time.

Other distributions will work with KVM, although the creation of the host bridged network is likely to be different. Advanced users may want to investigate the use of Logical Volume Managers (LVMs) instead of flat-file images to improve performance. ■

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

Virtual Machine Manager: virt-manager.et.redhat.com

`libvirt`: libvirt.org

QEMU: wiki.qemu.org

KVM: www.linux-kvm.org

KVM on Fedora 11: www.howtoforge.com/virtualization-with-kvm-on-a-fedora-11-server

Shared Networks on Fedora: fedoraproject.org/wiki/Features/Shared_Network_Interface

`libvirt` Networking: wiki.libvirt.org/page/Networking#Fedora.2FRHEL_Bridging

SAINT ARNOLD



TOURS
EVERY
SATURDAY
AT 1PM

TEXAS'
OLDEST
CRAFT
BREWERY

WWW.SAINTARNOLD.COM

Polywell Industrial Mini-PCs

Cost Effective Embedded PC For Appliances



ITX-10A



ITX-30A with PCI Riser



ITX-30G with NVIDIA® ION™ Graphics
Barebone system **\$199**



NVIDIA® ION™



ITX-50 Series



ITX-40A with Slim Optical Bay



**ITX-1000C with 4LAN
and WiFi Option**



VESA / Wallmount option



Full Height Riser or Low-Profile Add-on Slot
up to 8 x 2.5" or 4 x 3.5" Hard Drives

Over 250 Mini-ITX Models Available:

- NVIDIA® GeForce 8200/8100 with AMD® Athlon/Phenom Processor
- NVIDIA® GeForce 9300/9100M/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

888.765.9686

linuxsales@polywell.com

polywell.com/us/Lx

■ 23 Years of Customer Satisfaction

■ 5-Year Warranty, Industry's Longest

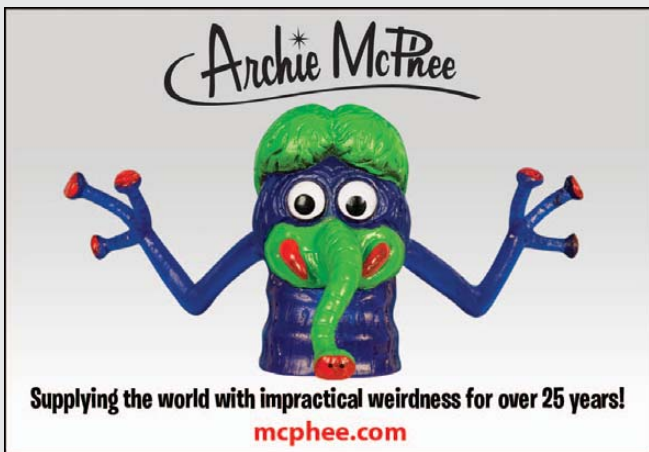
■ First Class Customer Service

Polywell Computers, Inc






1461 San Mateo Ave. South San Francisco, CA 94080 650.583.7222 Fax: 650.583.1974

NVIDIA, ION, nForce, GeForce and combinations thereof are trademarks of NVIDIA Corporation. Other names are for informational purposes only and may be trademarks of their respective owners.





*** STAY CONNECTED**

-  flickr.com/groups/linuxjournal/pool/
-  twitter.com/linuxjournal
-  linuxjournal.com/rss_feeds
-  facebook.com/linuxjournal
-  identi.ca/linuxjournal

Netdisk 9015N



Polywell Storage Servers

More Choices, Excellent Service, Great Prices!

Quiet Storage NAS/SAN/iSCSI

- 10TB \$1,799** - Dual Gigabit LAN
- 20TB \$2,999** - RAID-5, 6, 0, 1, 10
- 30TB \$4,999** - Hot Swap, Hot Spare
- Linux, Windows, Mac
- E-mail Notification
- Tower or Rackmount

Silent Eco Green PC The Best Terminal PC

Intel® Low Voltage Processor
Energy efficient, Quiet Platform.
starts at \$199



Mini-ITX LD-001

High Performance 5048A



**5U-48Bay 96TB
Storage Server**

4U45JB / 4U36A / 4U24A
RAID-6, NAS/iSCSI/SAN Storage
Mix SAS / SATA, 4 Giga / 10Gbit LAN



4U-24Bay 48TB \$8,999
4U-36Bay 72TB \$12,500
4U-45Bay 90TB JBOD

2008A / 2012A / 2012B
RAID-6, NAS/iSCSI/SAN Storage
Mix SAS / SATA, 4 GigaLAN



2008A 16TB \$2,699
2012A 24TB \$3,999
2012B 24TB \$4,899

1U945GCL2



Mini-1U Server \$499

Intel® Dual-Core Processor, 2 x 500G RAID
Dual GigaLAN, 4GB DDR2 RAM

- Over 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/Storage



Polywell Computers, Inc 1461 San Mateo Ave. South San Francisco, CA 94080 650.583.7222 Fax: 650.583.1974

Intel is trademarks of Intel Corporation. Other names are for informational purposes only and may be trademarks of their respective owners.



What Does Linux Want?

Is it more than just to be fruitful and multiply? DOC SEARLS

Linux is a family of operating systems, all based on the Linux kernel. Each Linux OS is called a distribution, or a distro. There is a plethora of these. DistroWatch lists just the top 100 of a number so large that it invites satire:

Somewhere in California—At 8:30 PDT with the release of Snoopy Linux 2.1 and Goober Linux 1.0, the number of Linux distributions finally surpassed the number of actual Linux users.

“We’ve been expecting it for some time”, Merrill Lynch technology analyst Tom Shayes said, “but this is a little sooner than most expected. We’ve seen explosive growth in the number of Linux distributions; in fact, my nephew just put out LittleLinux Chart Tommy Linux 1.1 last week.”

Except, that’s not a goof on DistroWatch. It’s a goof that ran on April 3, 2000—almost 11 years ago—in BBspot (www.bbspot.com/News/2000/4/linux_distros.html).

So Plethorization (www.linuxjournal.com/content/linux-and-plethorization) has been around for a long time. It is also a virtue of Linux, and of open-source code in general. It reproduces easily by copying (literally), and by adaptation. Distributions that evolve do so by adapting to purposes and improving. Those that don’t die off. Thus, Darwin’s *The Origin of Species* is as relevant to code as it is to life.

Species and code are both experimental. Both have genetics. Both differentiate by forking. Both are generative. Both are brutal meritocracies. Improvements that work are what survive and thrive. As Eric S. Raymond explains in *The Magic Cauldron*, the Linux marketplace is not about what gets sold, but about what gets used.

Linux is a family of code genera and species that grows and differentiates as hackers create and improve code for many different purposes. The result is lots of leaders but no winners, because neither evolution nor

plethorization are zero-sum games. The number of Linux distros did not go down when Red Hat was the top distro at the turn of the millennium, nor when Ubuntu moved to the front of the pack during the next decade. Instead, variants emerged.

The Wikipedia entry for Fedora currently lists 19 variants of that distro alone. The Fedora Wiki says, “There are roughly over a hundred distributions based on Fedora.” A DistroWatch search (in October 2010) brings up 38. Ubuntu itself is a child of Debian. Other Debian children include BackTrack, Knoppix, Linspire, MEPIS and Xandros. Grandchildren through Ubuntu and Knoppix alone run into many dozens.

Perhaps the only popular open-source codebase to “win” the market-share battle is the Apache HTTP server. According to Netcraft (news.netcraft.com/archives/2009/08/31/august_2009_web_server_survey.html), Apache has been the top Web server since May 1996 and has hosted the majority of the world’s Web sites for most of the duration. But, while Apache still hosts more than two-thirds of the busiest domains, its share of the Web server market has declined since peaking several years ago (as did its main competitor, Microsoft’s IIS). Since then, other Web servers also have become popular, even though Apache still leads the pack. But, that sports talk is zero-sum stuff. The fact is, the HTTP server market is a growing pile, not a sliced-up pie. So, although Apache still leads in percentage share, everybody whose numbers grow gets to keep playing. No winners required.

We do, however, see converging as well as diverging development vectors. Although divergence seeks out many problems to solve, convergence zeros in on a single problem. Lately, we have been learning that convergence needs more than a distro, just as an architect needs more than a sturdy set of building materials. This is why we now have Android and MeeGo, two Linux-based operating systems for mobile phones. Neither is listed among Linux distros by Wikipedia or by DistroWatch—because they’re not distros. They’re a layer up from that.

With Android, Google wanted to provide a complete platform for makers of mobile hardware and software. With MeeGo, Nokia and Intel wanted to do the same. All three know Linux well. It’s under most of Google’s infrastructure and on top of Intel’s hardware. Nokia’s early Linux experience was with Maemo, which didn’t get much traction, but which also taught a lesson: you need more. We already see where Android is going. In the next year, we’ll see if MeeGo does the same. In both, we’ll get to see what Linux wants them to do.

That last sentence was informed by Kevin Kelly’s brilliant new book, *What Technology Wants*. The evolution of technology, Kevin says, resembles that of biology by combining three forces. The first two are *structural* and *historical*. The third in biology is *functional*. In technology, however, the third is *intentional*. In biology, the functional is *adaptive*: “the relentless engine of optimization and creative innovation that continually solves the problems of survival”. But technology is interested in more than survival. The intentional is *open*. That is, it is about “human free will and choice”. Thus, “It is the third leg, the collective choice of free-willed individuals, that provides the character of the technium” (Kelly’s term for the sum of technology).

As “the collective choice of free-willed individuals”, it’s hard to find a better example than Linux. What Linux wants, as a collective choice, is made clear by the distinction in kernel development between “kernel space” and “user space”, and how the former exists and improves to support the latter. The kernel’s purpose is not to make anybody rich, or to make and break companies. (Although it runs in the credits of many who have done those things.) The purpose of Linux is to be useful. That’s what Linux wants. That’s what everybody who uses it gets. And, that’s why the population of Linux uses and users only goes up. ■

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.

iX-Triton TwinBlade Servers: The Easy-to-Manage, Greener Way to Serve

► AFFORDABLE ► ECONOMICAL ► SAVINGS

The new **Triton TwinBlade Server** is the most technologically advanced blade server system in the industry, and the ideal solution for power-efficiency, density, and ease of management.



The **Triton TwinBlade Server** supports up to 120 DP servers with 240 Intel® Xeon® 5600/5500 series processors per 42U rack, achieving an unmatched 0.35U per DP node. Up to two 4x QDR (40 Gbps) Infiniband switches, 10GbE switches or pass-through modules give the TwinBlade the bandwidth to support the most demanding applications.

With **N+1 redundant, high efficiency (94%) 2500W power supplies**, the TwinBlade is the Greenest, most energy-efficient blade server in the industry. The energy saved by the iX-Triton TwinBlade Server will keep the environment cleaner and greener, while leaving the green in your bank account.

Server management is also simple with the Triton Twin Blade Server.

Remote access is available through SOL (Serial Over Lan), KVM, and KVM over IP technologies. A separate controller processor allows all of the Triton's remote management and monitoring to function regardless of system failures, offering true Lights Out Management.

Using the **Triton's management system, administrators can remotely control TwinBlades**, power supplies, cooling fans, and networking switches. Users may control the power remotely to reboot and reset the Triton TwinBlade Center and individual Twin Blades, and may also monitor temperatures, power status, fan speeds, and voltage.

For more information on the **iX-Triton TwinBlade**, or to request a quote, visit: <http://www.iXsystems.com/tritontwinblade>

Key features:

- Up to 10 dual-node TwinBlades in a 7U Chassis, 6 Chassis per 42U rack
- Remotely manage and monitor TwinBlades, power supplies, cooling fans, and networking switches
- Virtual Media Over Lan (Virtual USB, Floppy/CD, and Drive Redirection)
- Integrated IPMI 2.0 w/ remote KVM over LAN/IP
- Supports one hot-plug management module providing remote KVM and IPMI 2.0 functionalities
- Up to four N+1 redundant, hot-swap 2500W power supplies
- Up to 16 cooling fans

Each of the TwinBlade's two nodes features:

- Intel® Xeon® processor 5600/5500 series, with QPI up to 6.4 GT/s
- Intel® 5500 Chipset
- Up to 128GB DDR3 1333/ 1066/ 800MHz ECC Registered DIMM / 32GB Unbuffered DIMM
- Intel® 82576 Dual-Port Gigabit Ethernet
- 2 x 2.5" Hot-Plug SATA Drive Trays
- Integrated Matrox G200eW Graphics
- Mellanox ConnectX QDR InfiniBand 40Gbps or 10GbE support (Optional)



Call iXsystems toll free or visit our website today!
+1-800-820-BSDi | www.iXsystems.com



Cut Execution Time by >50% with WhisperStation-GPU

Delivered ready to run new GPU-enabled applications:

Design

3ds Max
Bunkspeed
Shot
Adobe CS5

Simulation

ANSYS Mechanical
Autodesk Moldflow
Mathematica

MATLAB
ACUSIM AcuSolve
Tech-X GPULib

BioTech

AMBER
GROMACS
NAMD, VMD
TeraChem

Integrating the latest CPUs with NVIDIA Tesla Fermi GPUs, Microway's WhisperStation-GPU delivers 2x-100x the performance of standard workstations. Providing explosive performance, yet quiet, it's custom designed for the power hungry applications you use. Take advantage of existing GPU applications or enable high performance with CUDA C/C++, PGI CUDA FORTRAN, or OpenCL compute kernels.

- ▶ Up to Four Tesla Fermi GPUs, each with: 448 cores, 6 GB GDDR5, 1 TFLOP single and 515 GFLOP double precision performance
- ▶ Up to 24 cores with the newest Intel and AMD Processors, 128 GB memory, 80 PLUS® certified power supply, and eight hard drives
- ▶ Nvidia Quadro for state of the art professional graphics and visualization
- ▶ Ultra-quiet fans, strategically placed baffles, and internal sound-proofing
- ▶ New: Microway CL-IDE™ for OpenCL programming on CPUs and GPUs



WhisperStation with 4 Tesla Fermi GPUs

Microway's Latest Servers for Dense Clustering

- ▶ 4P, 1U nodes with 48 CPU cores, 512 GB and QDR InfiniBand
- ▶ 2P, 1U nodes with 24 CPU cores, 2 Tesla GPUs and QDR InfiniBand
- ▶ 2U Twin² with 4 Hot-Swap MBs, each with 2 Processors + 256 GB
- ▶ 1U S2050 servers with 4 Tesla Fermi GPUs

Microway Puts YOU on the Cutting Edge

Design your next custom configuration with techs who speak HPC. Rely on our integration expertise for complete and thorough testing of your workstations, turnkey clusters and servers. Whether you need Linux or Windows, CUDA or OpenCL, we've been resolving the complicated issues – so you don't have to – since 1982.

Configure your next WhisperStation or Cluster today!

microway.com/quickquote or call 508-746-7341

Sign up for technical newsletters and special GPU promotions at microway.com/newsletter



OctoPuter™ 4U Server with up to 8 GPUs and 144 GB memory

1U Node with 2 Tesla Fermi GPUs

2U Twin² Node with 4 Hot-Swap Motherboards
Each with 2 CPUs and 256 GB



GSA Schedule
Contract Number:
GS-35F-0431N

Microway
Technology you can count on™