AN SSC PUBLICATION

CONVERTING E-BOOKS TO OPEN FORMATS

LINUX JOURNAL

Since 1994: The Original Magazine of the Linux Co
JUNE 2005

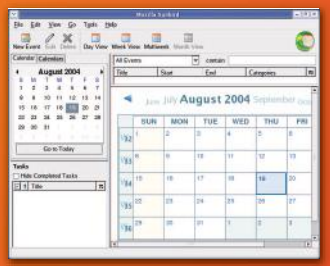
ATA over Ethernet:
Simple SAN on a budget

SUPPORTING HIGH-END HARDWARE

Hardware vendors have big demands—can the 2.6 kernel deliver?

CREATING CALENDARS FROM YOUR WEB APP

The standard that's opening up user calendars to your software



MODELING THE HUMAN BRAIN WITH PYTHON

Simple OO code for taming the Beowulf cluster

PLUS:

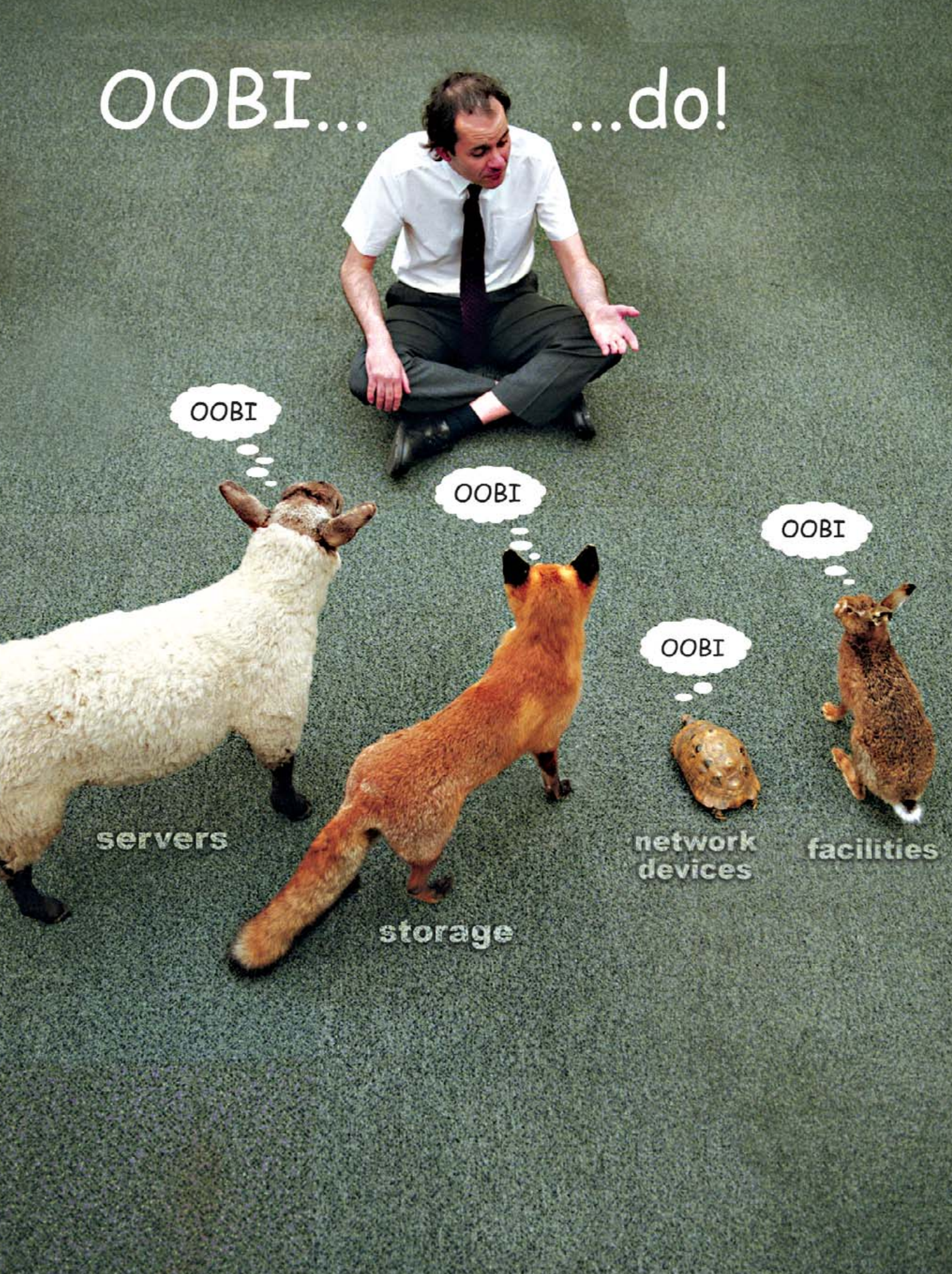
Real-life real-time: benchmarking kernel improvements

Samba goes to school

Replicating databases for reliability or performance

USA $5.00 CAN $6.50
www.linuxjournal.com
06
0 71486 03102 4

LINUX JOURNAL · INTRANET · Slony-I · Squid · libextractor · e-Books · 2.6 Kernel · FreeRADIUS · Brainlab · JUNE 2005 ISSUE 134

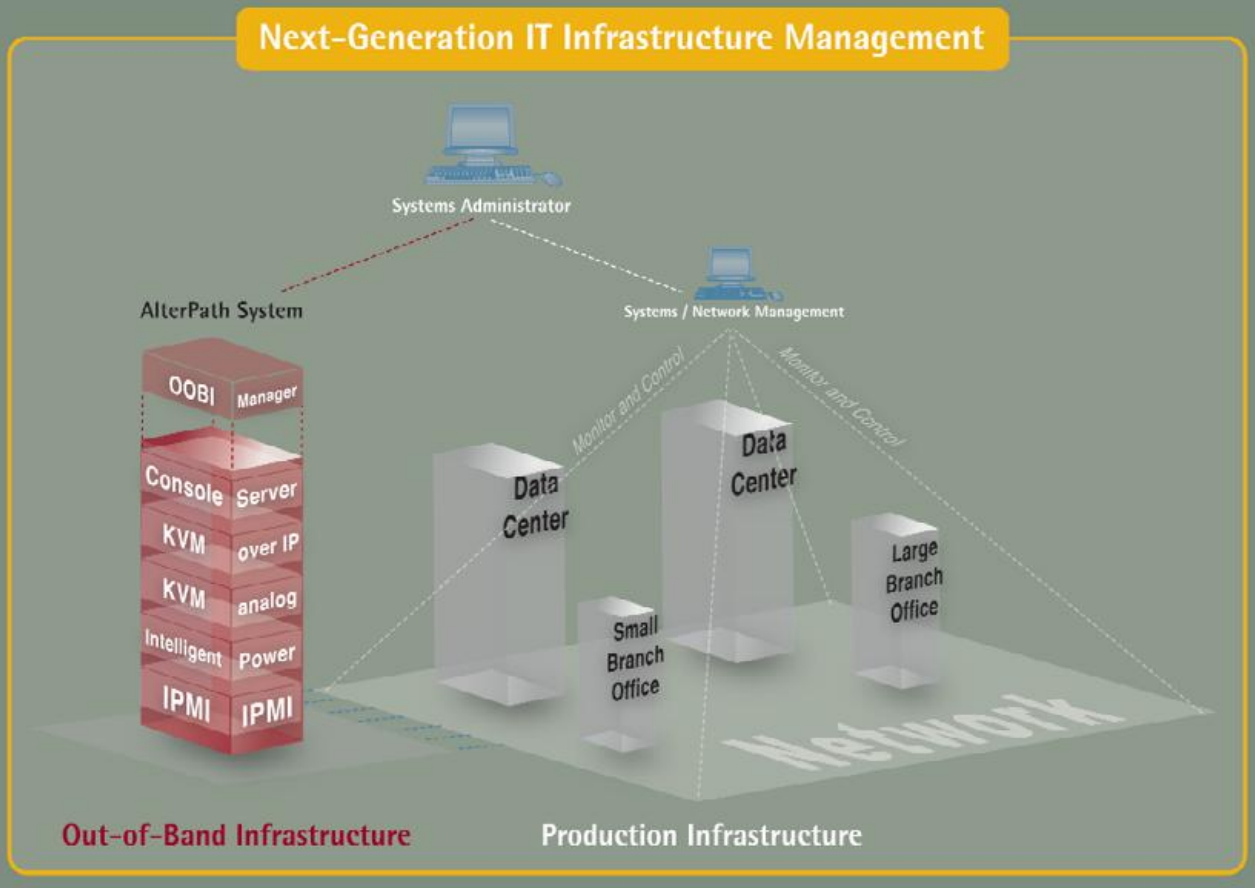

OOBI...
...do!
OOBI
OOBI
OOBI
OOBI
servers
storage
network devices
facilities

Let's talk OOBI

Next-Generation IT Infrastructure Management

OOBI, or Out-of-Band Infrastructure, is Cyclades' unique approach to enhance your IT infrastructure management capabilities. It integrates management of serial ports, KVM, KVM over IP, intelligent power distribution and IPMI devices in a secure, consolidated management solution for remote IT infrastructure administration. It complements system management tools like HP OpenView, IBM® Tivoli®, BMC PATROL® and CA Unicenter®.

An OOBI can cut costs and boost operational efficiency and productivity by minimizing the need for redundant equipment and personnel and, in most cases, eliminating the need for crash cart runs or remote visits to restore IT assets that have become disconnected from your production infrastructure.



To receive a **FREE** white paper on OOBI, Out-of-Band Infrastructure, visit us at **www.cyclades.com/oobiwp**

Over 85% of Fortune 100 choose Cyclades.

www.cyclades.com/lja

1.888.cyclades · sales@cyclades.com

cyclades

©2005 Cyclades Corporation. All rights reserved. All other trademarks and product images are property of their respective owners. Product information subject to change without notice.

Reduce Your Deployment and Support Costs

MBX is *the* leader for your server and appliance manufacturing needs

Supermicro 5013G-MB

- Intel® Pentium 4 Processor® at 3.0E GHz
- 1U Rackmount Chassis
- 512MB PC3200 DDR
- Maxtor 80GB Serial ATA Hard Drive
- Dual Onboard Gigabit NIC's
- Includes CDROM, Floppy and Video
- Lifetime toll free tech support
- 3 Year Warranty

$959 or lease for **$33**/mo.

Or Promote Your Brand

- Same Configuration as Above
- Custom Branded With Your Logo
- Worldwide Deployment and Support
- Custom Branded Packaging Available
- Configurations in 2U and 4U Available
- Custom OS and Software Install
- Custom Chassis Color Available
- No Minimum Quantity Required

$999 or lease for **$38**/mo.

MBX is the leader in custom appliances. Many premier application developers have chosen MBX as their manufacturing partner because of our experience, flexibility and accessibility. Visit our website or better yet, give us a call. Our phones are personally answered by experts ready to serve you.

MBX™
MOTHERBOARDEXPRESS

www.mbx.com
1.800.688.2347

Intel, Intel Inside, Pentium and Xeon are trademarks and registered trademark of Intel Corporation or its subsidiaries in the United States and other countries. Lease calculated for 36 months, to approved business customers. Prices and specifications subject to change without notice. Setup fee may apply to certain branding options. Motherboard Express Company. 1101 Brown Street Wauconda, IL. 60084.

COVER STORY

LINUX JOURNAL

JUNE 2005 ISSUE 134

FEATURES

INDEPTH

EMBEDDED

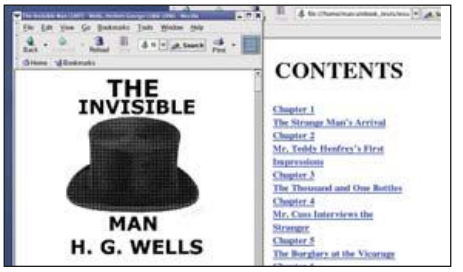
TOOLBOX

COLUMNS

REVIEWS

ON THE COVER: FUJITSU PRIMEQUEST SERVER IMAGE COURTESY OF FUJITSU.

DEPARTMENTS



Don't let your e-book collection lock you in to one device. Marco Fioretti makes order out of format chaos, on page 89.

NEXT MONTH

SYSTEM ADMINISTRATION

One of our most frequently referenced articles was December 2002's "OpenLDAP Everywhere". The authors, Craig Swanson and Matt Lung, are back with a step-by-step how-to, updated for new software versions and features, that will get your Linux and Microsoft clients all happily using the same OpenLDAP directory for everything from address books to NFS and Samba home directories.

Joshua Bentham had a typical business application development task. He needed to modify the contents of a database with forms and generate printed reports. By the way, the app should be cross-platform. His answer was Rekall, a slick tool that lets you build forms visually, create reports and add functionality in Python.

We've all had to use applications that aren't user-friendly, but when media players get to be positively user-hostile with annoying restrictions, it's time for a change. Bert Hayes helps you move your Apple iPod from the bundled software to a freedom-friendly music organizer.

Other People's Problems

Peer production is only the beginning. Today, the best software maintenance is part salesmanship.

BY DON MARTI

As long as there has been software, we've been facing the "buy or build" decision. But "build" became a last resort as packaged proprietary software offered better value. Today there's a third option, free and open-source software, or what Yochai Benkler called "commons-based peer production" in his paper "Coase's Penguin, or Linux and the Nature of the Firm".

Cooperating on software development is great, but most of the cost of software is maintenance. If you've been using Linux for a while, you probably have in-house versions of software that don't match the mainstream versions, and you're stuck maintaining it. Just as you have the "buy, build or peer-produce" decision, you have a decision to make about maintenance of code you'll need in the future. Maintain it yourself, sell a free software project on maintaining it or work with support vendors—who probably will try to sell it to a project themselves.

Except for the little bit that gets value from being secret—the formula that decides which households receive a credit-card offer, or the algorithm for making the aliens in the game attack you in a suitably compelling way—code is better and cheaper if you get someone else to maintain it for you. The ideal is to get an ongoing free software project to decide to do things your way. Glen Martin of open-source support company SpikeSource says they'll support fixes they make for customers as long as necessary, but "We don't want to continue maintaining them." That means part of the business is selling changes to project maintainers.

Red Hat's Tim Burke makes the same point on page 70. Red Hat now makes it a priority to get kernel patches into the main tree, contentious as the process can be. If you don't want to use your powers of persuasion to manipulate the software ecosystem, some vendors will tell you to drop open source, give up control and just do it their way. But somewhere in the middle, between spending all your time playing open-source politics and giving up entirely, is the approach that's working for more and more companies. You might be happy with Red Hat's kernel, but get involved in Web reporting software yourself, for example.

Free databases are taking the same steps into business-critical roles that Linux did last century. Ludovic Marcotte has a promising answer to the database clustering problem that beats switching to a proprietary database or hacking up something that just works for your application. Get started with database replication on page 52.

ATA over Ethernet (AoE) storage hit the market recently, and when we saw the new driver in the kernel, we got Ed Cashin to explain it. AoE goes with logical volume management like cookies and milk, as you'll see on page 24.

Selling projects on maintaining your code for you is such a powerful lever that we can expect to see more persuasion and sales skills included in future developer training. Whether you're buying, building or getting someone else to do it for you, enjoy the issue.

Don Marti is editor in chief of *Linux Journal*.

LINUX JOURNAL

JUNE 2005
ISSUE 134

EDITOR IN CHIEF Don Marti, ljeditor@ssc.com

EXECUTIVE EDITOR Jill Franklin, jill@ssc.com

SENIOR EDITOR Doc Searls, doc@ssc.com

SENIOR EDITOR Heather Mead, heather@ssc.com

ART DIRECTOR Garrick Antikajian, garrick@ssc.com

TECHNICAL EDITOR Michael Baxter, mab@cruzio.com

SENIOR COLUMNIST Reuven Lerner, reuven@lerner.co.il

CHEF FRANÇAIS Marcel Gagné, mggagne@salmar.com

SECURITY EDITOR Mick Bauer, mick@visi.com

CONTRIBUTING EDITORS

David A. Bandel • Greg Kroah-Hartman • Ibrahim Haddad • Robert Love • Zack Brown • Dave Phillips • Marco Fioretti • Ludovic Marcotte • Paul Barry

PROOFREADER Geri Gale

VP OF SALES AND MARKETING Carlie Fairchild, carlie@ssc.com

MARKETING MANAGER Rebecca Cassity, rebecca@ssc.com

INTERNATIONAL MARKET ANALYST James Gray, jgray@ssc.com

REGIONAL ADVERTISING SALES

NORTHERN USA: Joseph Krack, +1 866-423-7722 (toll-free)

EASTERN USA: Martin Seto, +1 905-947-8846

SOUTHERN USA: Annie Tiemann, +1 866-965-6646 (toll-free)

ADVERTISING INQUIRIES ads@ssc.com

PUBLISHER Phil Hughes, phil@ssc.com

ACCOUNTANT Candy Beauchamp, acct@ssc.com

***LINUX JOURNAL* IS PUBLISHED BY, AND IS A REGISTERED TRADE NAME OF, SSC PUBLISHING, LTD.**

PO Box 55549, Seattle, WA 98155-0549 USA • linux@ssc.com

EDITORIAL ADVISORY BOARD

Daniel Frye, Director, IBM Linux Technology Center

Jon "maddog" Hall, President, Linux International

Lawrence Lessig, Professor of Law, Stanford University

Ransom Love, Director of Strategic Relationships, Family and Church History Department, Church of Jesus Christ of Latter-day Saints

Sam Ockman, CEO, Penguin Computing

Bruce Perens

Bdale Garbee, Linux CTO, HP

Danese Cooper, Open Source Diva Intel Corporation

SUBSCRIPTIONS

E-MAIL: subs@ssc.com • URL: www.linuxjournal.com

PHONE: +1 206-297-7514 • FAX: +1 206-297-7515

TOLL-FREE: 1-888-66-LINUX • MAIL: PO Box 55549, Seattle, WA 98155-0549 USA • Please allow 4–6 weeks for processing address changes and orders • PRINTED IN USA

USPS *LINUX JOURNAL* (ISSN 1075-3583) is published monthly by SSC Publishing, Ltd., 2825 NW Market Street #208, Seattle, WA 98107. Periodicals postage paid at Seattle, Washington and at additional mailing offices. Cover price is $5 US. Subscription rate is $25/year in the United States, $32 in Canada and Mexico, $62 elsewhere. POSTMASTER: Please send address changes to *Linux Journal*, PO Box 55549, Seattle, WA 98155-0549. Subscriptions start with the next issue. Back issues, if available, may be ordered from the Linux Journal Store: store.linuxjournal.com.

LINUX is a registered trademark of Linus Torvalds.

Rackspace — Managed Hosting backed by Fanatical Support.™

Servers, data centers and bandwidth are not the key to hosting enterprise class Web sites and Web applications. At Rackspace, we believe hosting is a service, not just technology.

Fanatical Support is our philosophy, our credo. It reflects our desire to bring responsiveness and value to everything we do for our customers. You will experience Fanatical Support from the moment we answer the phone and you begin to interact with our employees.

Fanatical Support has made Rackspace the fastest-growing hosting company in the world. Call today to experience the difference with Fanatical Support at Rackspace.

LINUX JOURNAL READERS' CHOICE 2004

Thanks for honoring us with the 2004 Linux Journal Readers' Choice Award for **"Favorite Web-Hosting Service"**

rackspace™ MANAGED HOSTING

1.888.571.8976 or visit us at **www.rackspace.com**

Accepted at Fish Markets Everywhere

Here's my favorite credit card. When I use it, I frequently hear the cashier say, "Wow. Cool card!" I used to get excited thinking I'd made a Linux connection. Now I wait for the other shoe to drop, as it's usually followed by, "What's the penguin for?" But, sometimes it gives me a chance to evangelize just the same. Either way, it's nice to have a bit of fun while they're taking my money.

--
Brian Elliott Finley

That card is from linuxfund.org and helps fund free and open-source software grants and fellowships.—Ed.

Ultimate Power Saver Too?

Each year, *Linux Journal* embarks on the assembly of the Ultimate Linux Box, with the apparent goal of crafting the most powerful system possible within budget—a machine to shake the earth for miles around when switched on. This is now enough of a tradition that I wouldn't suggest tampering with it, but I wonder if some variants could be added with less coverage.

What I'm curious about is Linux systems set up with different goals in optimization. For example, what hardware exists with the lowest energy budget that is also capable of office work? The old Rebel machines came in at something like 15 watts without monitor. Can we do better? It would be instructive, but possibly less useful, to try optimizing new hardware for a similar task, but to optimize for minimum cost. Perhaps another category would be the machine that creates the least office clutter in deployment, which might well be an excuse to perform some heavy-duty case mods.

Linux is so flexible and adaptable, with so much hardware supported, it seems shameful that the only "ultimate" system is a fur-covered, fire-breathing, earth-shaking, meat-eating beast of a machine.

--
Thompson Freeman

Useless Use of For

The last trick Prentice Bisbal provides in his article ["My Favorite bash Tips and Tricks", April 2005] to list files in a directory should win him a UUOF award in the spirit of the UUOC awards. In order to list all the entries in a directory, all you have to do when `ls` doesn't work is `echo *`. And yes, I've had to use it.

--
Mike Mattice

One More Shell Tip

Prentice Bisbal asked how to show the contents of a file using only bash ["My Favorite bash Tips and Tricks", April 2005]. Here's one way: `while read; do echo "$REPLY";done < file.txt`. (The quotes around $REPLY prevent the shell from expanding any glob characters that might be in the file text.)

--
Steve Greenland

Corrections on Interrupts

The IRQ article in the April 2005 issue has a number of technical problems:

- "Any attempt to allocate an interrupt already in use, however, eventually crashes the system." Not true, as the article itself points out later.
- The prototype for interrupt handlers is wrong; it was changed in April 2003, for 2.5.69.
- "The second argument is a device identifier, using major and minor numbers...." is wrong. dev_id is simply the same pointer passed in to request_irq().
- The explanation of SA_INTERRUPT, beyond its grammatical problems, is not really correct; SA_INTERRUPT should not be used for anything anymore. SA_PROBE has never been meant for use outside of the IRQ subsystem itself, and nobody has ever passed it to request_irq().

The sample module would not compile, and in any case, the build system has changed to

the point that you cannot build a module with a simple gcc command anymore.

--
Jonathan Corbet

Considering the rapid pace of kernel development, we should not have run an article last tested on an early 2.6 kernel. It was our mistake to run it without sending it back to the author for an update.—Ed.

B. Thangaraju responds: *I was very happy to note that a person of Mr Jonathan Corbet's eminence has made his valuable suggestions on my article. The first sentence can be changed to "IRQ allocation will fail if it attempts to allocate an interrupt already in use."*

Prior to 2.5.69, interrupt handlers returned void. The prototype mentioned in the article was correct in the 2.4 kernel but in 2.6, interrupt handlers now return an irqreturn_t value.

This article was written in February 2003 and published in April 2005. I was working with the 2.4 kernel during the preparation of the article, and I tested the code with the 2.6.0-0.test2.1.29 kernel. So, some of the newer developments were not in use at the time of that writing, but the scenario, as you have rightly pointed out, has changed now.

IM Server Recommendation

First off, I'd like to say that *Linux Journal* is the absolute best Linux magazine out there in my opinion. The how-tos are intuitive, and my career has improved because of my subscriptions to this magazine. Now, I would like to see an article on jivesoftware.org's Jive Messenger Server. To me, this is where Jabber *should* be as an open-source alternative to the commercial IM servers out there. It's extremely configurable for a plethora of back-end databases, and runs best on...well, you know...Linux.

--
Anthony Moore

Get Maps from Google?

I enjoyed Charles Curley's article on GpsDrive in *Linux Journal* [April 2005]. Near the very end he suggested anyone who knows of a mapping data source let him know. You might consider looking at maps.google.com. It uses an open XML standard and API for free mapping integration. It might be worth looking at.

--

Burk Price

Easier Package Picking?

I'd really like to see Debian and Debian-based distros become easier for non-gurus to live with.

I tried two Debian-based distros, Mepis and Ubuntu. Each of them used about 1.5GB of hard drive space. Mepis used 150MB of RAM, but to be fair, it included lots of extra desktop gizmos. Ubuntu used 90MB of RAM. I also especially appreciated Ubuntu because it comes default with GNOME. Fedora 3 uses 2.5GB of hard drive space and 90MB of RAM for its home computer configuration.

Debian users will tell you that apt-get is more efficient than RPM because RPM's dependencies are other packages, while apt-get's dependencies are individual files. They'll also tout that apt-get does a better job of taking care of dependencies for you. But, guess what? With apt-get, you have to know exactly which packages you need to make a software system work.

Let's take MySQL for example. To make it work, you need the mysql-common, mysql-server and mysql-client packages. Technically, mysql-common will install without mysql-server and mysql-client. But it doesn't do you much good. With apt-get, you already have to know this. You also have to know the package name of any add-ons you might want, like graphical administration tools or Apache plugins. And yes, I was using the graphical interface to apt-get, not the command line.

With RPM, you would run into the same problem; however, Fedora's application management tool includes categories for common programs like MySQL. So I just click that I want MySQL, and Fedora selects all the necessary packages for me. I can then click details and select or de-select optional components.

Need to find something fast?

INDEX
INDEX
INDEX

With c-tree Speed.

FairCom's c-tree Plus® embedded database engine offers Superior Indexing Technology – the key to performance, data integrity, and concurrency. c-tree Plus offers direct record-oriented C and C++ APIs with an industry-standard SQL interface that allows use of any combination of APIs within the same application. Furthermore, we offer source code access for intimate programming control, unmatched portability, and developer-to-developer technical support.

Migrate from other Record-Oriented Databases!

Custom upgrade pricing is available for developers using any other record-oriented database. Btrieve®, VSAM®, C-ISAM™, and CodeBase® developers can migrate to c-tree with minimal overhead! E-mail info@faircom.com for more information.

Go to www.faircom.com/go/ljdownload for a FREE evaluation of c-tree Plus!

13 supported 64-bit platforms, now including AMD Opteron™

FairCom®

www.faircom.com

USA • Europe • Japan • Brazil

Other company and product names are registered trademarks or trademarks of their respective owners. © 2005 FairCom Corporation

MonarchComputer.com

Visit us online TODAY at
www.monarchcomputer.com

Get Half-Life 2 *FREE* by upgrading to AMD Athlon™ 64!

Visit our website to see how!

PayPal

Buy Online or by phone:

1-800-611-0875
1-8-MONARCHPC

Paypal - Visa - Mastercard - Discover - AMEX

Monarch makes it quick and easy to upgrade with FREE setup and testing on Motherboard Combos and $18.00 build fee on Barebones.

FREE INSTALLATION SETUP & TESTING BY CERTIFIED TECHS

FREE TECH SUPPORT!

AMD Motherboard Combos

Asus A8V-E Deluxe Mainboard with AMD Athlon™ 64 processor 3000+ (939) — Only $292

Abit AV8-3rd Eye Mainboard with AMD Athlon™ 64 processor 3200+ — Only $331

Tyan S2882G3NR Mainboard with AMD Opteron™ processor 244 — Only $621

Tyan S2882G3NR (Thunder K8S) with AMD Opteron™ processor 248 — Only $859

Abit AV8-3rd Eye K8T800 w/ AMD Athlon™ 64 processor 3200+ (939) — Only $334

Tyan S2882G3NR (Thunder K8S) MB w/ AMD Opteron™ processor 252 — Only $1243

Mainboard - Processors - Heatsink and Fan with Memory Options - FREE INSTALLATION AND TESTING
LASTEST BIOS loaded for easy upgrades - AMD Athlon™ XP, Athlon™ MP, Athlon™ 64, Athlon™ 64 FX, and Opteron™ Combos Available

AMD Barebone Systems

Go to www.monarchcomputer.com. select Barebones from the Menu. Choose AMD Athlon™ XP or AMD Athlon™ 64. Then configure your barebones online or call 1-8-MONARCHPC.

Lian-Li PC-V1200 Server Case
w/Wheels w/460W PS
Tyan Thunder K8W S2885ANRF
AMD Opteron™ processor 244 1.4 GHz
Starting @ $921

Antec Plusview 1000AMG
w/460W PS
Abit AV8-3rd Eye K8T800
AMD Athlon™ 64 processor 3500+
(939 - 90nm)
Starting @ $575

SliverStone SST-TJ06-B (Black)
E-ATX w/460W PS
Tyan S2882G3NR Thunder K8S
AMD Opteron™ processor 248
Starting @ $1138

Antec Performance One P160 Case
w/420W PS
Abit KV8 PRO Motherboard w/
AMD Athlon™ 64 processor 2800+ (754)
Starting @ $438

***AMD Athlon 64 and Athlon 64 FX are the ONLY Windows®-compatible 64-bit PC processor

www.monarchcomputer.com

AMD 64-bit CPUs

AMD Athlon™ 64 OEM CPUs

AMD Athlon™ 64 2800+ (754) $122.00
AMD Athlon™ 64 3000+ (754) $149.00
AMD Athlon™ 64 3200+ (754) $194.00
AMD Athlon™ 64 3400+ (754) $223.00
AMD Athlon™ 64 3500+ (939) $272.00
AMD Athlon™ 64 3700+ (754) $329.00
AMD Athlon™ 64 3800+ (939) $424.00
AMD Athlon™ 64 4000+ (939) $643.00
AMD Athlon™ 64 FX-55 (939) $828.00

AMD Athlon™ 64 CPUs Feature:
- HyperTransport™ technology
- Enhanced Virus Protection for Microsoft® Windows® XP- SP2
- Cool'n'Quiet™ technology
- AMD64 technology

AMD Opteron™ OEM CPUs

AMD Opteron™ 144 1.8GHz $173.00
AMD Opteron™ 146 2.0GHz $211.00
AMD Opteron™ 148 2.2GHz $270.00
AMD Opteron™ 150 2.4GHz $404.00

AMD Opteron™ 244 1.8GHz $203.00
AMD Opteron™ 246 2GHz $307.00
AMD Opteron™ 246 HE (55W) 2GHz $441.00
AMD Opteron™ 248 2.2GHz $441.00
AMD Opteron™ 250 2.4GHz $669.00
AMD Opteron™ 252 2.6GHz $825.00
AMD Opteron™ 844 1.8GHz $677.00
AMD Opteron™ 846 2GHz $677.00
AMD Opteron™ 846HE (55W) 2.0GHz $847.00
AMD Opteron™ 848 2.2GHz $847.00
AMD Opteron™ 850 2.4GHz $1130.00
AMD Opteron™ 852 2.6GHz $1469.00

NEW AMD OPTERON™ 252 & 852
Available at our website
www.monarchcomputer.com

Components and Upgrades
1000s of In-Stock Components

100756
Apex, FoxConn Mid-Tower
TU-150 400W PS
USB 2.0 Ports (Black)
$67.00

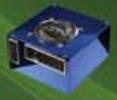
150137
Western Digital
250 GB WD2500JB
7200 RPM 8MB IDE
$151.00

140270
1 GB DDR (266) PC-2100
REG ECC Corsair
(CM72SD1024RLP-2100)
$229.00

140662
512 MB DDR (266)
PC-2100 REG-ECC
Mushkin Pro
(990742)
$136.00

190148
eVGA GeForce FX 5200
128MB DDR/PCI/S-Video
DVI/No-VGA
$62.00

150239
Western Digital
74 GB SATA 10K
Raptor (WD740GD)
$182.00



100963
Ultra X-Connect ULT31558
ATX 500 Watt Power Supply
w/2-80mm Fans (Blue)
$92.00

110407
Asus A7V8X-X KT400
AUDIO/LAN DDR 400
8X /4X AGP 333 FSB XP
$54.00

Visit www.monarchcomputer.com for products in this ad, current inventory and up-to-the-minute pricing. Call or email for special orders

Educational and Government POs Welcome.

GSA Schedule Contract GS-35F-0202P

We ship to the Continental U.S., Alaska, Hawaii, APOs, Puerto Rico, and Canada

Commercial leasing available for purchases as low as $1000.

Prices subject to change without notice. Monarch Computer not responsible for typographical errors.

AMD, the AMD Arrow logo, AMD Athlon, and combinations thereof, are trademarks of Advanced Micro Devices, Inc. All brands and product names are trademarks or registered trademarks of their respective companies."QuantiSpeed™ architecture for exceptional software performance ™. Processor architecture operates at 2.0GHz AMD model numbers are a simple, accurate representation of relative AMD processor performance on industry-standard software benchmarks. Model numbers convey relative performance among different AMD processors to help you simplify your purchase decision.

Monarch Has The *LOWEST PRICES*
Custom 64-Bit Servers, Workstations & Desktops

AMD 64 Athlon FX | AMD 64 Athlon | AMD 64 Opteron

Exlusive Athlon™ 64 FX Launch Partner

NEW! ***Monarch's EMPRO line makes buying AMD Opteron™ Workstations and Servers Easier than ever before!***

A FULL LINE OF WORKSTATIONS, RACK SERVERS, TOWER SERVERS

Choose a preconfigured System Special, upgrade any component or accessory, even customize your system from the inside out - see the dynamic, real-time pricing on your configuration, and save a quote to lock in pricing for up to 7 days!

Part:80340
Monarch Furia Deluxe Workstation Special
w/ AMD Athlon™ 64 3800+
1GB PC3200 DDR
PNY Quadro FX500
ONLY $1802.00

Part:90999
Monarch Centira Ultimate Desktop Special
w/ AMD Sempron™ 3100+
512MB DDR
ATI 9600SE AGP
ONLY $739.00

8 New System Lines,Including:

Centira™ - Starts @ **$359** (Athlon™ XP or Sempron™)

Solia™ - Starts @ **$459** (Athlon™ 64 or Sempron™)

Furia™ - Starts @ **$1085** (Athlon™ 64 or Athlon™ 64 FX)

Empro™ - Starts @ **$1494** (Opteron™)

NEW SYSTEM MODELS!

Value - The most value for the lowest price

Deluxe - The best balance of performance and price

Ultimate - The best of the best in cutting-edge technology

www.monarchcomputer.com/empro

The AMD Opteron processor— built upon forward-thinking AMD64 technology—provides flexibility with a 1–8-way scalable design.

The Official Linux Journal Web Server

Monarch Computer Systems

The AMD Athlon™ 64 processor improves security against certain types of viruses, with Enhanced Virus Protection for the Microsoft® Windows® XP SP2

GET QUICK QUOTES
Online or by phone:
www.monarchcomputer.com
1-800-611-0875
1-8-MONARCHPC
Paypal - Visa - Mastercard - Discover - AMEX

Now available with AMD Opteron™ processor 252!

Part# 80544

- Lian-Li PC-V1200 Aluminum Quiet Tower (Black) -550W Antec Power Supply)
- Tyan Thunder K8W S2885ANRF Board AGP Pro 4X 8X SATA/IEEE
- (2) AMD Opteron™ Processors 252
- 2 GB (4 pcs 512 MB) DDR (400) PC-3200 REG ECC Corsair Memory (TwinX1024RE-3200LL)
- (4) Western Digital 74 GB 10K RPM Raptor (WD740GD) 8 MB Cache SATA HDDs
- 3ware Escalade 9500S-4LP 4 Port SATA RAID Controller w/ RAID 5 Setup
- Asus 5232ASQT-BLK 52X32X52 CD-RW (Black
- Plextor PX-712A/SW-BL DVD±R/RW
- Mitsumi Floppy 7-in-1 Smart Card reader (Black)
- Creative Audigy 2 ZS 24BIT Advanced HD Sound
- PNY QuadroFX 3000 AGP 8x/4x 256MB
- Linux Fedora Core 3 A64 Installed
- 1 or 3 year warranties available

The *NEW* MONARCH ULB for 2005
Custom Opteron Workstation

As Reviewed in the Dec. 2004 issue of Linux Journal!

On SALE!
Starting @ $1276

As configured at left:
$5934

$100 Off Reviewed Price!

Browse now to monarchcomputer.com for a full list of available options

AMD, the AMD Arrow logo, and combinations thereof, and the AMD64 logo, and AMD Opteron, are trademarks of Advanced Micro Devices, Inc.

Commercial leasing available for purchases as low as $1000.

Prices subject to change without notice. Monarch Computer not responsible for typographical errors.

GSA Schedule Contract GS-35F-0202P

We ship to the Continental U.S., Alaska, Hawaii, APOs, Puerto Rico, and Canada

AMD, the AMD Arrow logo, AMD Athlon, and combinations thereof, are trademarks of Advanced Micro Devices, Inc. All brands and product names are trademarks or registered trademarks of their respective companies."QuantiSpeed™ architecture for exceptional software performance. ™ Processor architecture operates at 2.0GHz AMD model numbers are a simple, accurate representation of relative AMD processor performance on industry-standard software benchmarks. Model numbers convey relative performance among different AMD processors to help you simplify your purchase decision.

The problem isn't so bad with MySQL, but now let's talk about more complex package structures, like GNOME (or KDE). There are dozens of GNOME packages available via apt-get. Which ones do I need? I don't know. Is there one that will install all of the other necessary ones as dependencies? I don't know. Do I want any of the packages that aren't explicit dependencies? I don't know. With apt-get, I'd have to spend hours reading the descriptions of all the packages. With Fedora, I just click GNOME, and I get the important stuff and a list of the optional stuff to choose from.

My grandma could probably install KDE for Fedora. But Debian needs work. There needs to be "master" packages that install all of the required stuff for a given complex system and then prompt you to make choices about the add-on stuff.

--
R. Toby Richards

Mmm, VPN Article

My daughter, Angel Sakura, and I were reviewing a back article on Linux VPNs. She

really ate it up.

--
Patrick Betts

Why C for CGI?

I found several flaws with Clay Dowling's article "Using C for CGI Programming" [April 2005]. He seems to not realize that there is software that caches compiled PHP bytecode that can speed up execution quite a bit. An example is Turck MMCache: turck-mmcache.sourceforge.net/index_old.html.

An interesting statement: "The fairly close times of the two C versions tell us that most of the execution time is spent loading the program." Well, duh! It seems downright absurd to go through the hassle of coding CGIs in C, and then use the old fork-exec model. Why not write the applications as Apache modules? This would have sped up execution time significantly. Besides, a lot of the cross-platform issues already have been resolved in the Apache Portable Runtime.

--
Brian Akins

Who Let Marketing Edit the RSS Title?

I like your articles okay so far, but your RSS feed sucks. That is the longest damn title I ever saw, and I don't even want to hear about Linux by the time you're done blowing your own horn.

--
Anonymous

TV Watchers Rejoice

I thoroughly enjoyed Doc Searls' Linux for Suits column ("The No Party System") in the April 2005 issue of *LJ*. However, I feel that he left out one excellent example of his point. Toward the end of the article, he discusses the new Linux version of SageTV as well as the many benefits provided by ReplayTV as a result of it being based on a Linux system. I have never used SageTV nor have I owned a ReplayTV or TiVo (although I have quite a few friends who do), but I've been a dedicated user of MythTV (www.mythtv.org) for almost two years now.

From everything I've seen or read, MythTV seems to be head and shoulders better than the other options out there, including Windows Media Center Edition, SageTV, ReplayTV and TiVo, and it's only on version 0.17! Now I know that most people would normally be scared off by a version number that low, but trust me, Myth is already incredibly polished and user-friendly at this stage of the game. MythTV can do pretty much anything your TiVo or ReplayTV can, plus more. And, with the possible exception of some new hardware, depending on what you've got sitting in your basement/closet, it's completely free! There is most definitely a bit of up-front setup required to get it going in the first place, but once the system is up and running, it's a piece of cake to use.

Myth can handle everything from time-shifting television to storing and playing back your music library (in almost any format), to watching DVDs (or DVDs that you've ripped to the hard drive, effectively providing movies on demand), to weather information, to managing your digital picture galleries, to playing your favorite arcade/NES/SNES/atari games on your TV. And the best part is, if there's a feature you want that Myth doesn't already have, you can always write it yourself. The developers are always happy to include new patches and features from the user community.

If you're interested in seeing the power of Linux and the Open Source community, I'd highly suggest that you at least take a look at MythTV.

--
Brad Benson

Where's the HP Linux Laptop?

A few weeks ago, after dropping my laptop on the floor, I went shopping on the HP Web site. On the nx5000 page, HP still touted that it came with a choice of XP or SUSE 9.2, but when I went to the configuration pages (I tried all of them), there was no such choice. I e-mailed HP shopping support and thus far have received only an automated acknowledgement. A week later, I was asked to complete a survey of HP E-mail support, and I did so, noting how completely useless it was. I checked "Yes, you may contact me about my response to the survey", but they never followed up on that either. I've since given up and bought a refurbished ThinkPad, but I have to conclude that HP has quietly discontinued their Linux laptop.

--
Larry Povirk

The nx5000 is no longer manufactured. We checked with Elizabeth Phillips at HP, and she says that Linux on HP notebooks and desktops lives on. Through a "Factory Express" program, you can get Linux on any desktop or notebook. Ordering info at www.hp.com/go/factory-express.*—Ed.*

Photo of the Month

No photo qualified this month, but continue to send photos to ljeditor@ssc.com. Photo of the month gets you a one-year subscription or a one-year extension.—Ed.

We welcome your letters. Please submit "Letters to the Editor" to ljeditor@ssc.com or SSC/Editorial, PO Box 55549, Seattle, WA 98155-0549 USA.

Cyclades AlterPath™ Manager for Cyclades IPMI Management

Enjoy the magic

- Remote IT infrastructure administration
- Centralized OOBI™* management
- Remote incident resolution
- Web-based access
- Remote logging and monitoring
- Supports IPMI 1.5 and 2.0

Cyclades' AlterPath™ Manager is the first secure enterprise IPMI manager to boost operational efficiency and productivity with industry leading features:

- Vendor-independent IPMI management
- Enterprise security and authentication (SSH v2, LDAP, RADIUS, TACACS+, Kerberos)
- Centralized access, control and auditing
- Event notification and alarms
- Scalable to 5000 devices and 256 simultaneous connections

We've worked our magic. Now you can work yours.

To receive a **FREE** white paper on IPMI, visit us at www.cyclades.com/ipmiwp

*OOBI, or Out-of-Band Infrastructure, integrates management of serial ports, KVM, KVM/IP, intelligent power distribution and IPMI devices in a secure, consolidated management solution for remote IT infrastructure administration.

Over 85% of Fortune 100 choose Cyclades.

www.cyclades.com/ljb

1.888.cyclades · sales@cyclades.com

cyclades

©2005 Cyclades Corporation. All rights reserved. All other trademarks and product images are property of their respective owners. Product information subject to change without notice.

On the WEB

It's time to start the voting process for the 2005 Readers' Choice Awards. This year, we're changing the procedure to allow you to have even more input about which tools, products, publications and other Linux necessities are your favorites. Head over to the *LJ* site to read "New Procedures for 2005 Readers' Choice Awards" (www.linuxjournal.com/article/8192) and learn how you can become involved.

Thinking about teaching a class on Linux? If so, be sure to read "Designing a Course in Linux System Administration" (www.linuxjournal.com/article/8193) by Mike LeVan, a professor at Transylvania University. LeVan explains how he designed the syllabus, prepared assignments and chose the textbook. He also discusses how to integrate the philosophy behind the technology and methods of assessment.

Spring typically is a busy time for book publishers, so be sure to keep an eye on the *LJ* Web site for reviews of some of the newest Linux and OSS titles. In addition, we're running an excerpt from *Firefox & Thunderbird Garage* (www.linuxjournal.com/article/8194), written by some of Mozilla's core developers. We're also running excerpts from Chapter 7 of Arnold Robbin's *Linux Programming by Example* (www.linuxjournal.com/article/8195), a walk-through of the UNIX V7 version of ls.

diff -u

What's New in Kernel Development

The **iswraid** driver seems to be on the fast track into the 2.4 tree, apparently in spite of the fact that it adds new functionality to a stable series kernel. **Marcelo Tosatti** deferred to **Jeff Garzik**'s judgment on the issue, over strenuous objections from other developers. Jeff reasoned that without iswraid, 2.4 users would be unable to make use of their hardware, while detractors (including **Arjan van de Ven**, **Bartlomiej Zolnierkiewicz** and **Christoph Hellwig**) argued that the same could be said for all new hardware that was not yet supported. As it stands, the issue is Jeff's call to make, so we can expect iswraid in an upcoming 2.4 release.

A number of new drivers have seen the light of day. **Vojtech Pavlik** has written a driver for the **serial Elo touchscreen device**, expected to support all generations of serial Elos. Apparently this area of the kernel is just waiting to bloom, as some folks have been supporting touchscreen hardware for years as in-house company projects. A new **real-time-clock** driver for the **ST M41T00 I2C RTC chip** has been released by **Mark A. Greer** and almost immediately is slated for inclusion in the 2.6 tree. Mark also has released a driver for the I2C controller on Marvell's host bridge for PPC and MIPS systems.

Willy Tarreau, with blessings from Marcelo Tosatti, has started a new hot fix branch of the 2.4 tree. The -hf branch will have the same fixes that go into 2.4, but on an accelerated release schedule. New drivers and updates to existing drivers will be excluded. The -hf branch will be only for security fixes and clear bug fixes. Some might argue that before putting out a -hf branch, Marcelo might consider a slightly accelerated release schedule himself. But the situation seems to work for the developers and is in tune with Marcelo's desire to affirm 2.4's relentless drive toward stability and not to give in to any sense of urgency in the process.

Christoph Lameter has created a **scrubd** page zeroing dæmon and related kernel infrastructure. This is intended to help eke out the best possible speed from the page fault handler, by zeroing pages of memory before they are needed, rather than at the time they are requested. It's nice to pay attention to this sort of improvement, because even though it is not a new driver, changes no APIs and is not really visible to the outside world, it contributes to making Linux the snappy, sleek operating system that serves us all so well. These sorts of optimizations are the bread and butter of Linux and should be recognized along with the hot new drivers and fancy filesystems.

The **out-of-memory process killer** (OOM Killer) continues to be one of the tough nuts to crack in Linux development. **Mauricio Lin** recently released a user-space version that he claimed worked as well as the in-kernel version. There are many issues, however. A user-space tool runs the risk of being the victim of an out-of-memory condition itself, like any other program. But a kernel-side OOM killer is more difficult to tune for a particular system. Mauricio's compromise moves the ranking algorithm into user space, where it is more easily configurable, while leaving the actual killer in the kernel, where it is somewhat protected from the out-of-memory conditions it seeks to mitigate. Although it is a controversial issue because of the many complexities of any OOM handling tool, Mauricio's approach seems to be finding some support among top developers like Marcelo Tosatti. Mauricio also has been working in related areas, and he recently produced a patch to allow users to track the size of a process' physical memory usage in the /proc directory. This also has proven to be somewhat controversial, but **Andrew Morton** favors it, and others have proposed actual uses that would make it valuable in practice.

Jeff Garzik put out a reminder recently that several broken and deprecated drivers soon would be removed from the 2.6 tree. The iphase driver has been broken for years and won't even compile. The xircom_tulip_cb driver is unmaintained and doesn't cover the full spectrum of xircom 32-bit cards; the xircom_cb driver, on the other hand, works for them all and is a fine replacement. The **eepro100** driver is unmaintained and will be replaced by the e100 driver. However, users who are bumping into issues where e100 is not yet a workable replacement can relax: the issues will be resolved before eepro100 is removed.

—ZACK BROWN

Speed and endurance take many forms...

You decide...Linux ready on every board

WAN

LAN

Storage

Carriers

Custom

SBE®

Linux On Demand

flexibility on demand | 925-355-2000 | info@sbei.com | www.sbei.com

TONG

www.nongnu.org/tong

Tetris or *Pong*? *Tetris* or *Pong*? If this is the hardest decision of your working day, Owen Swerkstrom just doubled your productivity with this game that plays *Tetris* and *Pong* clones at the same time. You play *Tetris* with the keyboard and *Pong* with the mouse. What happens when the ball hits the descending block? Does the ball knock blocks off the stack, or just bounce? You'll have to play to find out because the rules for *Tetris-Pong* interaction are different every time. And if you can't get your hands trained to play the game, you always can snag Jared Burke's "Fanfare for the Common Rabbit" and other background tunes for your "happy synth songs" playlist.

—DON MARTI

Ten Years Ago in *Linux Journal*

Greg Hankins put multi-port serial boards to the test and found a Comtrol RocketPort board got the best speed score, and a Cyclades one came in best for low CPU usage. All of the competitors were EISA cards and had IRQs and I/O addresses selectable with DIP switches.

Before "commercial open source" became common, "commercial applications" meant proprietary software. A directory of commercial applications had 23 entries, including five databases and three Motif ports.

One of the classic Linux books made its first appearance. Grant Johnson reviewed the first edition of *Running Linux* by Matt Welsh and Lar Kaufman. Besides installing Slackware, the book got readers started with setting up a mail server and creating a Web site—even writing HTML.

Galacticomm took out a full-page ad for its bulletin board software product, The Major BBS. *Linux Journal* publisher Phil Hughes announced *Linux Journal*'s first Web site and offered advertisers links from an on-line ad index, or "if they don't have their own Web site, for a nominal fee we will put their Web pages on www.ssc.com." Out of the 47 ads in the issue, 42 included an e-mail address, but only 13 had a URL. (O'Reilly had e-mail, Web, telnet and Gopher contact info—show-offs.)

—DON MARTI

They Said It

A patent is merely the ticket to the license negotiation.

—STEPHEN WALLI
stephesblog.blogs.com/my_weblog/2005/02/a_patent_is_mer.html

The biggest problem is going to be rewriting the budget, having to figure out what to do with all that money that's no longer going to Microsoft.

—BOYCE WILLIAMS, FROM A THREAD ON DOC SEARLS' IT GARAGE
(garage.docsearls.com/node/550)

Don't think like a cost center, you'll get cut. Think like an entrepreneur.

—ANONYMOUS, ALSO FROM A THREAD ON DOC SEARLS' IT GARAGE
(garage.docsearls.com/node/550)

You're right not because others agree with you, but because your facts are right.

—WARREN BUFFET, www.fortune.com/fortune/fortune75

The gap between customer 0 (the alpha geek) and customer n ("pro-sumer") is narrowing.

—RAEL DORNFEST

Hack your system: It's a Good Thing.

—PEGGY ROGERS, "MS. COMPUTER", *THE MIAMI HERALD*

In fact I think every programmer should fight for attribution, no matter what company is writing the paycheck. Look at the entertainment industry. Who shows up where in the credits is a big, big deal...translating directly to job satisfaction and a way to track an individual's body of work over time. This is one of the best features of open source in my opinion.

—DANESE COOPER,
danesecooper.blogs.com/divablog/2005/03/about_attributi.html

EmperorLinux

...where Linux & laptops converge

The Meteor: 3lb Linux

- Sharp Actius MM20/MP30
- 10.4" XGA screen
- X@1024x768
- 1.6 GHz Transmeta Efficeon
- 20-40 GB hard drive
- 512-1024 MB RAM
- CDRW/DVD (MP30)
- 802.11b/g wireless
- Ethernet/USB2
- ACPI hibernate
- 1" thin

The SilverComet: 4 lb Linux

- Sony VAIO S270
- 13.3" WXGA+ screen
- X@1280x800
- 1.5-2.0 GHz Pentium-M
- 40-100 GB hard drive
- 256-1024 MB RAM
- CDRW/DVD or DVD-RW
- 802.11b/g wireless
- 10/100 ethernet
- ACPI hibernate
- USB2/FireWire

The Toucan: 5 lb Linux

- IBM ThinkPad T series
- 14.1" SXGA+/15.0" UXGA
- X@1400x1050/X@1600x1200
- ATI FireGL graphics
- 1.6-2.13 GHz Pentium-M 7xx
- 40-80 GB hard drive
- 512-2048 MB RAM
- CDRW/DVD or DVD-RW
- 802.11b/g wireless
- 10/100/1000 ethernet
- APM suspend/hibernate

The Rhino: 7 lb Linux

- Dell Latitude D810/M70
- 15.4" WUXGA screen
- X@1920x1200
- NVidia Quadro or ATI Radeon
- 1.73-2.13 GHz Pentium-M 7xx
- 30-80 GB hard drive
- 256-2048 MB RAM
- CDRW/DVD or DVD+/-RW
- 802.11b/g wireless
- 10/100/1000 ethernet
- USB2/SVideo/serial

Since 1999, EmperorLinux has provided pre-installed Linux laptop solutions to universities, corporations, and individual Linux enthusiasts. We specialize in the installation and configuration of the Linux operating system on a wide range of the finest laptop and notebook computers made by IBM, Dell, Sharp, and Sony. We offer a range of the latest Linux distributions, as well as Windows dual boot options. We customize each Linux distribution to the particular machine it will run upon and provide support for: ethernet, modem, wireless, PCMCIA, USB, FireWire, X-server, CD/DVD/CDRW, sound, power management, and more. All our systems come with one year of Linux technical support by both phone and e-mail, and full manufacturers' warranties apply. Visit www.EmperorLinux.com or call 1-888-651-6686 for details.

Custom Configurations

Linux Pre-Installed

Technical Support

www.EmperorLinux.com 1-888-651-6686

Model prices, specifications, and availability may vary. All trademarks are the property of their respective owners.

YOUR HIGH PERFORMANCE COMPUTING SOLUTION HAS ARRIVED.

VXRACK™ with the Intel® Xeon™ processor helps you simplify computing operations, accelerate performance and accomplish more in less time.

1 Choose one of the 3 convenient rack sizes

VXR-96
Rack accomodating up to
96 VXBlades/**192 Processors**
36TB of aggregated **Storage**
1.15TB of Global Memory
Power Distribution Included
Patented Architecture
Advanced Cooling System
Integrated InfiniBand Cable Mgnt.

$ 1,750.00*

VXR-128
Rack accomodating up to
128 VXBlades/**256 Processors**
48TB of aggregated **Storage**
1.5TB of Global Memory
Power Distribution Included
Patented Architecture
Advanced Cooling System
Integrated InfiniBand Cable Mgnt.

$ 2,190.00*

VXR-72
Rack accomodating up to
72 VXBlades/**144 Processors**
27TB of aggregated **Storage**
864GB of Global Memory
Power Distribution Included
Patented Architecture
Advanced Cooling System
Integrated InfiniBand Cable Mgnt.

$ 1,590.00*

CIARA TECHNOLOGIES

For more Information call
or visit us at

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, Pentium, and
*Important Information. All prices, specifications and promotional offers are subject to change without notice. Ciara cannot be responsi

VXB-7221B
Intel SE7221B Motherboard
800MHz Front Side Bus
Intel® Pentium® 4 3.2GHz
1GB DDR2 400 Memory
Single 40GB 7200RPM ATA Drive
One **PCI/Express Slot Available**
Dual 10/100/1000 Intel Lan Port
350W Power Supply

$ 985.00

VXB-7501W
Intel SE7501W Motherboard
533MHz Front Side Bus
2 x Intel® Xeon™ 3.06GHz
2GB DDR 333 ECC Reg.Mem
Single 40GB 7200RPM ATA Drive
One **PCI/X Slot Available**
Dual 10/100/1000 Intel Lan Port
350W Power Supply

$ 2,355.00

VXB-7520J
Intel SE7520J Motherboard
800MHz Front Side Bus
2 x Intel® EM64T Xeon™ 3.2GHz
2GB DDR2 400 ECC Reg.Mem
Single 40GB 7200RPM ATA Drive
One **PCI/Express Slot Available**
Dual 10/100/1000 Intel Lan Port
500W Power Supply

$ 2,950.00

2 Choose one or more type of VXBlade

3 Add, Mutiply,That's it. Easy as 1, 2, 3...

For example you choose the following: One VXR-96 with 48 Dual Intel® EM64T Xeon™ and 40 Single Intel® Pentium®4. You take 1 (VXR-96) + 48 (VXB-7520J) + 40 (VXB-7221B)...That's it

VXRACK

THE FUTURE OF CLUSTER TECHNOLOGY

CIARA TECHNOLOGIES...A GLOBAL SOLUTION PROVIDER.

Ciara Technologies is a world-class computer systems manufacturer. Ciara designs, develops, manufactures, markets, services, and supports a variety of computer systems including graphic workstations, rackmount and tower servers, networked storage and the newly acclaimed VXRACK™ Cluster Technology. The company's state of the art supercomputer cluster is based on the Intel IA32 and IA64 architectures and utilizes Linux operating systems. We are proud to be recognized by Intel as an "Intel Premier Provider". Choosing Ciara is choosing a single point of contact for all your IT requirements. All our products are built under the ISO 9001 standards and regulations. The growth of Ciara enabled the company to move its 300+ employees,in February 2003, to an ultra-modern plant of 576,000 sqft.. Ciara now has the capability of producing more than 500,000 units per year.

866-7VX-RACK (866-789-7225)
WWW.VXRACK.COM

intel inside XEON™

Pentium III Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries
le for typography errors, photographies errors, pricing errors. All pricing in US dollar. Shipping and applicable taxes are not included.

Dynamically Generated Calendars

Want to remind your Web site's users about upcoming events or get the whole company synced on a common calendar? Get started creating iCalendar files with Python. BY REUVEN M. LERNER

Last column, we looked at Sunbird, a standalone application from the Mozilla Foundation for tracking calendars. As we saw, Sunbird is able to work with calendars in the iCalendar format. These calendars may be on the local filesystem or retrieved by HTTP from a remote server. We also saw how easy Sunbird makes it to use a calendar that a remote server has made available. We simply enter the URL into a dialog box, and after waiting for Sunbird to retrieve the iCalendar file, the new events are added to our calendar display.

A variety of remote calendars already exist on the Internet in iCalendar format, and you can find and subscribe to them without too much trouble. But doing so is helpful only if you want to subscribe to a calendar that already exists or is available publicly. What if your organization wants to standardize on iCalendar for exchanging event information? How can you create and distribute iCalendar files, such that others can keep track of the events they must attend?

This month, we look at the server side of iCalendar files and create calendars designed to be retrieved by calendar applications, such as Sunbird, within an organization.

iCalendar Files

If two computers are going to exchange calendars, we obviously need to have a standard that defines how those calendars should be formatted. The protocol over which they are exchanged is not defined, although both standards and daily use seem to indicate that HTTP is the overwhelming favorite for such transactions. The format for calendar exchange, defined in RFC 2445, reflects its age. Whereas a new calendar format would undoubtedly be defined to use XML, this RFC, dated November 1998, uses a set of name-value pairs, with some primitive nesting of elements within a hierarchy. For example, here is the the iCalendar file that we examined last month, when we first looked at Sunbird:

```
BEGIN:VCALENDAR
VERSION
 :2.0
PRODID
 :-//Mozilla.org/NONSGML Mozilla Calendar V1.0//EN
BEGIN:VEVENT
UID
 :05e55cc2-1dd2-11b2-8818-f578cbb4b77d
SUMMARY
 :LJ deadline
STATUS
 :TENTATIVE
CLASS
 :PRIVATE
X-MOZILLA-ALARM-DEFAULT-LENGTH
 :0
DTSTART
 :20050211T140000
DTEND
 :20050211T150000
DTSTAMP
 :20050209T132231Z
END:VEVENT
END:VCALENDAR
```

As you can see, the file begins and ends with BEGIN:VCALENDAR and END:VCALENDAR tags. There is some calendar-wide data at the top of the file, VERSION and PRODID, but then the first and only event is defined, bracketed by BEGIN:VEVENT and END:VEVENT entries. You can imagine how a file could have many more entries than this single one.

iCalendar makes it possible for an event to recur at regular intervals. You thus could have a single VEVENT entry reminding you about the weekly Monday-afternoon meeting or reminding you to put out the trash every Tuesday and Friday morning. Each event also has a beginning and ending time, DTSTART and DTEND, allowing for different lengths.

Although it is not obvious from the above example, iCalendar also allows us to make exceptions to recurring events. So, if your Monday-afternoon meeting is not going to take place during a holiday week, you can insert an EXDATE entry. The application that displays your calendar then ignores the recurring event on that date.

Publishing iCalendar Files

Assuming that we already have an iCalendar file on our system, making it available on the Web is quite easy. Listing 1 contains a simple CGI program that I wrote in Python; it looks for an iCalendar file in a particular directory and returns the contents of that file to the requesting calendar application.

If you haven't written a CGI program in Python before, this example should demonstrate how straightforward it is. Load the CGI module for some basic CGI functionality. Then, load the cgitb, for CGI traceback, module, which allows us to put debugging information in a file, if and when a problem occurs.

We then send a text/calendar Content-type header. It's probably safe to assume that most content on the Web is sent with a Content-type of text/html (for HTML-formatted text), text/plain (for plain-text files), with many of types image/jpeg, image/png and image/gif thrown in for good measure. The iCalendar standard indicates that the appropriate Content-type to associate with calendar files is text/calendar, even if programs such as Sunbird are forgiving enough to accept the text/plain format as well. Finally, we end the program by sending the contents of the calendar file, which we read from the

Is the Man getting you down? The Man says you can't. The Man says not today, maybe tomorrow. The Man wants to follow the path well trod. But the Man knows jack. **The Penguin,** on the other hand, knows Linux. Or, at least, we at Penguin Computing®, know what you want from it. Freedom to do your own thinking. To implement things the way you want to, not the way the software wants you to. The capability to find a better way - without crashing every five minutes. Best-in-class **Scyld-driven** clusters. More power-to-the-pound BladeRunner™ cluster-in-a-box. Powerful, scalable servers. And the sort of support you'd want for your children. Or, to be precise, your company's core applications. Your business' critical project. Or your industry changing ideas. So get back up. Stick it to the Man. **Love what you do.**

www.penguincomputing.com/lj

Listing 1. static-calendar.py, a simple CGI program in Python to open an iCalendar file and send it by HTTP.

```
#!/usr/bin/python

# Grab the CGI module
import cgi

# Log any problems that we might have
import cgitb
cgitb.enable(display=0, logdir="/tmp")

# Where is our calendar file?
calendar_directory = '/usr/local/apache2/calendars/'
calendar_file = calendar_directory + 'test.ics'

# Send a content-type header to the user's browser
print "Content-type: text/calendar\n\n"

# Send the contents of the file  to the browser
calendar_filehandle = open(calendar_file, "rb")
print calendar_filehandle.read()
calendar_filehandle.close()
```

local filesystem.

If you have been doing Web programming for any length of time, this example should be raising all sorts of red flags. The idea that we would use a program to return a static file seems somewhat silly, although this does have the slight advantage of letting us hide the true location of the calendar file from outside users. There are undoubtedly better ways to accomplish this, however, including the Apache Alias directive. We could improve this program somewhat by passing the calendar's filename as a parameter, but that still would require that we have a set of statically generated files.

Creating an iCalendar

The real solution, and one that makes life more interesting, is to create the iCalendar file dynamically when the user requests it. That is, our CGI program does not return the contents of an existing iCalendar file; instead, it creates an iCalendar file programmatically, returning it to the user's calendar client program.

At first glance, this might seem to be a simple task. After all, the iCalendar file format appears to be straightforward, so maybe we can code something together ourselves. But upon closer examination, we discover that creating an iCalendar file is easier said than done, particularly if we want to include recurring events.

Given the increasing popularity of the iCalendar standard and the plethora of open-source projects, I was surprised to discover the relative lack of attention that iCalendar has received from the biggest open-source programming communities. Part of my surprise was because iCalendar has been around for several years, is used by many companies and is supported by many calendar programs, from Novell's Evolution to Lotus Notes to Microsoft Outlook. This combination usually is a recipe for several different options, in several different programming languages.

I first looked at Perl, whose CPAN archive is renowned for its many modules, including many for Internet standards of various sorts. Although several Perl modules are available that parse iCalendar files, no up-to-date module exists for building them. Net::ICal::Libical was going to be a wrapper around the C-language libical library but was last released in a pre-alpha version, several years ago. Net::ICal was part of a project called ReefKnot, which also appears to have been abandoned.

Luckily, the Danish developer Max M (see the on-line Resources) recently decided to fill this gap and wrote a Python package that makes it easy to create an iCalendar file. I downloaded and installed the package on my computer without any trouble, and I found that it is quite straightforward to create a calendar with this package. Combined with our simple CGI program from before, we should be able to create and publish a calendar without any trouble.

Creating a Dynamic Calendar

I downloaded and installed the iCalendar package from the maxm.dk site. Unlike many modern Python packages, it doesn't install automatically. You must copy it manually to your system's site-packages directory, which on my Fedora Core 3 system is located at /usr/lib/python-2.3/site-packages.

As you can see in Listing 2, I was able to use this newly installed iCalendar package to create new objects of type Calendar and Event. The first thing I had to do was import the appropriate packages into the current namespace:

```
from iCalendar import Calendar, Event
```

The Calendar and Event modules inside of the iCalendar package correspond to the entire iCalendar file and one event in that file, respectively. We thus create a single instance of the Calendar object and one Event object for each event that we might want to create.

We then can create the calendar object:

```
cal = Calendar()
cal.add('prodid',
        '-//Python iCalendar 0.9.3//mxm.dk//')
cal.add('version', '2.0')
```

The second and third lines here, in which we invoke cal.add(), allow us to add identifying data to our iCalendar file. The first of these allows us to tell the client software which program generated the iCalendar file. This is useful for debugging; if we consistently get corrupt iCalendar files from a particular software package, we can contact the author or publisher and report a bug. The second line, in which we add a version identifier, indicates which version of the iCalendar specification we are following. RFC 2445 indicates that we should give this field a value of 2.0 if we are going to follow that specification.

Now that we have created a calendar, let's create an event and give it a summary line to be displayed in the calendar program of anyone subscribing to this iCalendar file:

```
event = Event()
event.add('summary', 'ATF deadline')
```

Listing 2. dynamic-calendar.py, a program that generates a calendar in iCalendar format.

```
#!/usr/bin/python

# Grab the CGI module
import cgi
from iCalendar import Calendar, Event
from datetime import datetime
from iCalendar import UTC # timezone

# Log any problems that we might have
import cgitb
cgitb.enable(display=0, logdir="/tmp")

# Send a content-type header to the user's browser
print "Content-type: text/calendar\n\n"

# Create a calendar object
cal = Calendar()

# What product created the calendar?
cal.add('prodid',
        '-//Python iCalendar 0.9.3//mxm.dk//')

# Version 2.0 corresponds to RFC 2445
cal.add('version', '2.0')

# Create one event
event = Event()
event.add('summary', 'ATF deadline')
event.add('dtstart',
           datetime(2005,3,11,8,0,0,tzinfo=UTC()))
event.add('dtend',
           datetime(2005,3,11,10,0,0,tzinfo=UTC()))
event.add('dtstamp',
           datetime(2005,3,11,0,10,0,tzinfo=UTC()))
event['uid'] = 'ATF20050311A@lerner.co.il'

# Give this very high priority!
event.add('priority', 5)

# Add the event to the calendar
cal.add_component(event)

# Ask the calendar to render itself as an iCalendar
# file, and return that file in an HTTP response!
print cal.as_string()
```

Every event, as we have already seen in the file we examined, has three date/time fields associated with it: the starting date and time, dtstart; the ending date and time, dtend; and an indication of when this entry was added to the calendar, dtstamp. The iCalendar standard uses a strange if useful format for its dates and times, but the Event object knows how to work with those if we give it a datetime object from the standard datetime Python package. So, we can say:

```
event.add('dtstart',
          datetime(2005,3,11,14,0,0,tzinfo=UTC()))
event.add('dtend',
          datetime(2005,3,11,16,0,0,tzinfo=UTC()))
event.add('dtstamp',
          datetime(2005,3,11,0,10,0,tzinfo=UTC()))
```

Notice that the above three lines used UTC as the time zone. When the iCalendar file is displayed inside of a client Calendar application, it is shown with the user's local time zone, as opposed to UTC.

Once we have created the event, we need to give it a unique ID. When I say unique, I mean that the ID should be truly unique, across all calendars and computers in the world. This sounds trickier than it actually is. You can use a number of different strategies, including using a combination of the creation timestamp, IP address of the computer on which the event was created and a large random number. I decided to create a simple UID, but if you are creating an application to be shared across multiple computers, you probably should think about what sort of UIDs you want to create and then standardize on them:

```
event['uid'] = 'ATF20050311A@lerner.co.il'
```

Finally, we must give our event a priority, in the range of 0 through 9. An event with priority 5 is considered to be normal or average; urgent items get higher numbers and less-urgent items get lower ones:

```
event.add('priority', 5)
```

Once we have created our event, we attach it to the calendar object, which has been waiting for us to do something with it:

```
cal.add_component(event)
```

If we are so interested, we then could to add more events to the calendar. So long as each has a unique UID field, there won't be any problems.

Finally, we turn our Calendar object into an iCalendar file, using the as_string() method:

```
print cal.as_string()
```

Because print writes to standard output by default, and because CGI programs send their standard output back to the HTTP client, this has the effect of sending an iCalendar file back to whomever made the HTTP request. And because we have defined the MIME type to be of type text/calendar, the HTTP client knows to interpret this as a calendar and display it appropriately. If we look at the output ourselves, we see that it is indeed in iCalendar format:

```
BEGIN:VCALENDAR
PRODID:-//Python iCalendar 0.9.3//mxm.dk//
VERSION:2.0
BEGIN:VEVENT
DTEND:20050311T160000Z
DTSTAMP:20050311T001000Z
DTSTART:20050311T140000Z
PRIORITY:5
SUMMARY:ATF deadline
UID:ATF20050311A@lerner.co.il
END:VEVENT
END:VCALENDAR
```

Now, I must admit that this example is almost as contrived as the previous one. True, we have exploited the fact that we can generate a calendar dynamically, but this event was hard-coded into the program, making it impossible for a nonprogrammer to add, modify or delete the event. That said, we have taken an additional step toward the programmatic calculation of events and dates. The next step is to store the dates in a file or even in a relational database and to use our program to convert the information on the fly.

Conclusion

This month, we looked at the creation of a dynamic calendar using the iCalendar module for Python wrapped inside of a simple CGI program. At the same time, we saw the limitations of having a calendar whose entries need to be on disk. A better solution would be to put that event information in a relational database, which has built-in support for dates, as well as security mechanisms for user and group access. Next month, we will extend our calendar program so that it retrieves information from a database, turning PostgreSQL tables into iCalendar files.

Resources for this article: www.linuxjournal.com/article/8197.

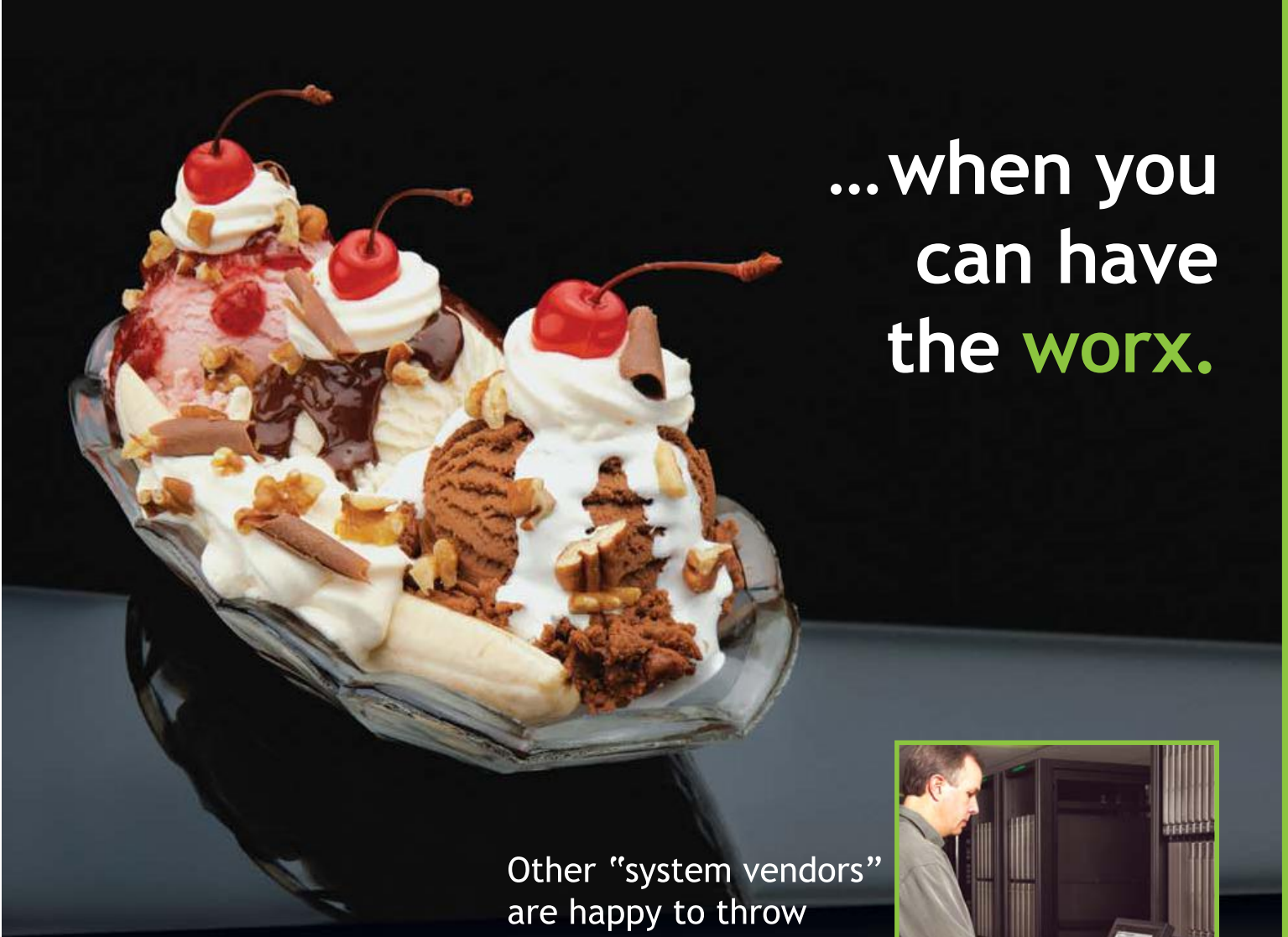
Reuven M. Lerner, a longtime Web/database consultant and developer, now is a graduate student in the Learning Sciences program at Northwestern University. His Weblog is at **altneuland.lerner.co.il**, and you can reach him at reuven@lerner.co.il.

GPSBabel

gpsbabel.sourceforge.net

If you're making maps, traveling, geocaching or otherwise using a GPS with your Linux system, don't let the crazy array of GPS data formats get you lost. Robert Lipe's command-line tool GPSBabel does for GPS data what ImageMagick does for graphics—converts what you have to what you need. Read the fine manual for options to convert data to and from Garmin, Magellan and other manufacturers' formats, along with formats that will work with Netstumbler, Google Maps and other software.

—DON MARTI



...when you can have the worx.

Other "system vendors" are happy to throw a processor in a box and call it good.

Not Linux Networx.

Our cluster systems are more than plain vanilla. That's why each Linux Networx system is engineered with our Active Cooling™ hardware technology, Total Cluster Management™ tools, Xilo™ scalable cluster storage, application optimization, and Certified Cluster Services.

The result is a high productivity computing experience unlike any other.

Linux Networx®
PROVEN

An important ingredient of any cluster system is the processing technology.

AMD 64 Opteron™

AMD Opteron™ processors provide a highly scalable architecture and support large memory addressability to deliver next-generation performance as well as a flexible upgrade path from 32- to 64-bit computing.

AMD

© Copyright 2005 Linux Networx, Inc. All rights reserved. Linux Networx and the cube logo, are registered trademarks of Linux Networx. Active Cooling, Total Cluster Management, and Xilo are trademarks of Linux Networx Inc. Linux is a registered trademark of Linus Torvalds. AMD, the AMD Arrow logo, AMD Opteron and combinations thereof, are trademarks of Advanced Micro Devices, Inc.

tech article

To learn how our cluster computing products can help your business and receive a free technical report on high productivity computing, visit www.linuxnetworx.com/theworxlj

ATA over Ethernet: Putting Hard Drives on the LAN

With ATA hard drives now cheaper than tape, this simple new storage technology enables you to build storage arrays for archives, backup or live use.

BY ED L. CASHIN

Everybody runs out of disk space at some time. Fortunately, hard drives keep getting larger and cheaper. Even so, the more disk space there is, the more we use, and soon we run out again.

Some kinds of data are huge by nature. Video, for example, always takes up a lot of space. Businesses often need to store video data, especially with digital surveillance becoming more common. Even at home, we enjoy watching and making movies on our computers.

Backup and data redundancy are essential to any business using computers. It seems no matter how much storage capacity there is, it always would be nice to have more. Even e-mail can overgrow any container we put it in, as Internet service providers know too well.

Unlimited storage becomes possible when the disks come out of the box, decoupling the storage from the computer that's using it. The principle of decoupling related components to achieve greater flexibility shows up in many domains, not only data storage. Modular source code can be used more flexibly to meet unforeseen needs, and a stereo system made from components can be used in more interesting configurations than an all-in-one stereo box can be.

The most familiar example of out-of-the-box storage probably is the storage area network (SAN). I remember when SANs started to create a buzz; it was difficult to work past the hype and find out what they really were. When I finally did, I was somewhat disappointed to find that SANs were complex, proprietary and expensive.

In supporting these SANs, though, the Linux community has made helpful changes to the kernel. The enterprise versions of 2.4 kernel releases informed the development of new features of the 2.6 kernel, and today's stable kernel has many abilities we lacked only a few years ago. It can use huge block devices, well over the old limit of two terabytes. It can support many more simultaneously connected disks. There's also support for sophisticated storage volume management. In addition, filesystems now can grow to huge sizes, even while mounted and in use.

This article describes a new way to leverage these new kernel features, taking disks out of the computer and overcoming previous limits on storage use and capacity. You can think of ATA over Ethernet (AoE) as a way to replace your IDE cable with an Ethernet network. With the storage decoupled from the computer and the flexibility of Ethernet between the two, the possibilities are limited only by your imagination and willingness to learn new things.

What Is AoE?

ATA over Ethernet is a network protocol registered with the IEEE as Ethernet protocol 0x88a2. AoE is low level, much simpler than TCP/IP or even IP. TCP/IP and IP are necessary for the reliable transmission of data over the Internet, but the computer has to work harder to handle the complexity they introduce.

Users of iSCSI have noticed this issue with TCP/IP. iSCSI is a way to send I/O over TCP/IP, so that inexpensive Ethernet equipment may be used instead of Fibre Channel equipment. Many iSCSI users have started buying TCP offload engines (TOE). These TOE cards are expensive, but they remove the burden of doing TCP/IP from the machines using iSCSI.

An interesting observation is that most of the time, iSCSI isn't actually used over the Internet. If the packets simply need to go to a machine in the rack next door, the heavyweight TCP/IP protocol seems like overkill.

So instead of offloading TCP/IP, why not dispense with it altogether? The ATA over Ethernet protocol does exactly that, taking advantage of today's smart Ethernet switches. A modern switch has flow control, maximizing throughput and limiting packet collisions. On the local area network (LAN), packet order is preserved, and each packet is checksummed for integrity by the networking hardware.

Each AoE packet carries a command for an ATA drive or the response from the ATA drive. The AoE Linux kernel driver performs AoE and makes the remote disks available as normal block devices, such as /dev/etherd/e0.0—just as the IDE driver makes the local drive at the end of your IDE cable available as /dev/hda. The driver retransmits packets when necessary, so the AoE devices look like any other disks to the rest of the kernel.

In addition to ATA commands, AoE has a simple facility for identifying available AoE devices using query config packets. That's all there is to it: ATA command packets and query config packets.

Anyone who has worked with or learned about SANs likely wonders at this point, "If all the disks are on the LAN, then how can I limit access to the disks?" That is, how can I make sure that if machine A is compromised, machine B's disks remain safe?

The answer is that AoE is not routable. You easily can determine what computers see what disks by setting up ad hoc Ethernet networks. Because AoE devices don't have IP addresses, it is trivial to create isolated Ethernet networks. Simply power up a switch and start plugging in things. In addition, many switches these days have a port-based VLAN fea-

Do you take

"the computer doesn't do that"

as a personal challenge?

So do we.

LINUX JOURNAL™

Since 1994: The Original Monthly Magazine of the Linux Community

Subscribe today at www.linuxjournal.com

Because AoE devices don't have IP addresses, it is trivial to create isolated Ethernet networks.

ture that allows a switch to be partitioned effectively into separate, isolated broadcast domains.

The AoE protocol is so lightweight that even inexpensive hardware can use it. At this time, Coraid is the only vendor of AoE hardware, but other hardware and software developers should be pleased to find that the AoE specification is only eight pages in length. This simplicity is in stark contrast to iSCSI, which is specified in hundreds of pages, including the specification of encryption features, routability, user-based access and more. Complexity comes at a price, and now we can choose whether we need the complexity or would prefer to avoid its cost.

Simple primitives can be powerful tools. It may not come as a surprise to Linux users to learn that even with the simplicity of AoE, a bewildering array of possibilities present themselves once the storage can reside on the network. Let's start with a concrete example and then discuss some of the possibilities.

Stan the Archivist

The following example is based on a true story. Stan is a fictional sysadmin working for the state government. New state legislation requires that all official documents be archived permanently. Any state resident can demand to see any official document at any time. Stan therefore needs a huge storage capacity that can grow without bounds. The performance of the storage needn't be any better than a local ATA disk, though. He wants all of the data to be retrievable easily and immediately.

Stan is comfortable with Ethernet networking and Linux system administration, so he decides to try ATA over Ethernet. He buys some equipment, paying a bit less than $6,500 US for all of the following:

- One dual-port gigabit Ethernet card to replace the old 100Mb card in his server.
- One 26-port network switch with two gigabit ports.
- One Coraid EtherDrive shelf and ten EtherDrive blades.
- Ten 400GB ATA drives.

The shelf of ten blades takes up three rack units. Each EtherDrive blade is a small computer that performs the AoE protocol to effectively put one ATA disk on the LAN. Striping data over the ten blades in the shelf results in about the throughput of a local ATA drive, so the gigabit link helps to use the throughput effectively. Although he could have put the EtherDrive blades on the same network as everyone else, he has decided to put the storage on its own network, connected to the server's second network interface, eth1, for security and performance.

Stan reads the Linux Software RAID HOWTO (see the online Resources) and decides to use a RAID 10—striping over mirrored pairs—configuration. Although this configuration doesn't result in as much usable capacity as a RAID 5 configuration, RAID 10 maximizes reliability, minimizes the CPU cost of performing RAID and has a shorter array re-initialization time if one disk should fail.

After reading the LVM HOWTO (see Resources), Stan comes up with a plan to avoid ever running out of disk space. JFS is a filesystem that can grow dynamically to large sizes, so he is going to put a JFS filesystem on a logical volume. The logical volume resides, for now, on only one physical volume. That physical volume is the RAID 10 block device. The RAID 10 is created from the EtherDrive storage blades in the Coraid shelf using Linux software RAID. Later, he can buy another full shelf, create another RAID 10, make it into a physical volume and use the new physical volume to extend the logical volume where his JFS lives.

Listing 1 shows the commands Stan uses to prepare his server for doing ATA over Ethernet. He builds the AoE driver with AOE_PARTITIONS=1, because he's using a Debian sarge system running a 2.6 kernel. Sarge doesn't support large minor device numbers yet (see the Minor Numbers sidebar), so he turns off disk partitioning support in order to be able to use more disks. Also, because of Debian bug 292070, Stan installs the latest device mapper and LVM2 userland software.

The commands for creating the filesystem and its logical volume are shown in Listing 2. Stan decides to name the volume group ben and the logical volume franklin. LVM2 now needs a couple of tweaks made to its configuration. For one, it needs a line with `types = [ "aoe", 16 ]` so that LVM recognizes AoE disks. Next, it needs `md_component_detection = 1`, so the disks inside RAID 10 are ignored when the whole RAID 10 becomes a physical volume.

I duplicated Stan's setup on a Debian sarge system with two 2.1GHz Athlon MP processors and 1GB of RAM, using an

Listing 1. The first step in building a software RAID device from several AoE drives is setting up AoE.

```
# setting up the host for AoE

# build and install the AoE driver
tar xvfz aoe-2.6-5.tar.gz
cd aoe-2.6-5
make AOE_PARTITIONS=1 install

# AoE needs no IP addresses!  :)
ifconfig eth1 up

# let the network interface come up
sleep 5

# load the ATA over Ethernet driver
modprobe aoe

# see what aoe disks are available
aoe-stat
```

Why buy more Servers when all you need is more Disks?

Linux + Disks + Ethernet = EtherDrive®

Disks go inside servers... right? If you run out of disk space you get another server... right? Well, that used to be the case, but not any more. Now you can expand the disk space on any server with EtherDrive Storage Blades.

EtherDrive Storage Blades are simple and easy to use. And the best part is, you already know how. An EtherDrive Storage Blade is a disk drive mounted on a very small server attached directly to your network. Each blade is a nanoserver, with firmware that puts the disk's storage right on your network. No IP addresses. Just disks on the network accessible by your servers.

Just Disk Drives on Ethernet

The open protocol, ATA-over-Ethernet, allows the most in flexibility and operation simplicity. Since EtherDrive blades just look like local disks, you already know how to use them. Use any file system software. Use any RAID software, or Coraid's open source RAIDBlade appliance. Use any volume management software. It's all up to you to decide how to organize your disks. And, since the protocol is open, you know everything about how it works. The protocol is simple, only 8 pages. The open source device driver means you never have to look at the protocol. But isn't it good to know you can?

Complete Control

You have complete control over the contents of the disk. EtherDrive doesn't store anything on your disks that you don't want. You can take a disk from a running system, install it on an EtherDrive Storage Blade, and mount it. That means you are always in control. No data reformatting. No captive data. Just disk drives on the network. You never have to worry about getting your data off of an EtherDrive Blade if it fails. Just mount the disk on a system or another Blade and you're back in business.

EtherDrive Storage Blades insert into a shelf of 10 slots. Using 400GB ATA disks you can have 4TB in one 3U rack space. You can add up to 4,095 shelves on a single network. That means you can have servers sharing more than 16 Petabytes.....Imagine that.

40,000 Disks on Your Servers

A system that can go from a couple of disks, all the way to 40,000 disks, in whatever increment you want. That's probably more than you'll ever need, but isn't that the idea of scalability? Since our shelves mount in simple relay racks, just like your switches, you never run out of room. Never have your data captive inside one server's chassis. Never have to fork lift obsolete systems. Never have to buy more servers when all you need is more disks.

Processing Power with Each Blade

EtherDrive Storage Blades can go fast, too. Since each blade has its own cpu, memory and Ethernet interface, they can all work independently or in unison. Striping software can read/write blades in parallel. The wider the stripe, the faster the I/O.

Each Blade isn't limited to a single server, either. Many servers can access the same group of EtherDrive Storage Blades. They can share read-only file systems or use available software like Red Hat's GFS to share read/write file systems.

Using EtherDrive Storage Blades you only add pennies to the cost of the raw storage. **Less than $0.65 per Gigabyte.**

CORAID

www.coraid.com
info@coraid.com
1-877-548-7200

Minor Device Numbers

A program that wants to use a device typically does so by opening a special file corresponding to that device. A familiar example is the /dev/hda file. An `ls -l` command shows two numbers for /dev/hda, 3 and 0. The major number is 3 and the minor number is 0. The /dev/hda1 file has a minor number of 1, and the major number is still 3.

Until kernel 2.6, the minor number was eight bits in size, limiting the possible minor numbers to 0 through 255. Nobody had that many devices, so the limitation didn't matter. Now that disks have been decoupled from servers, it does matter, and kernel 2.6 uses 20 bits for the minor device number.

Having 1,048,576 values for the minor number is a big help to systems that use many devices, but not all software has caught up. If glibc or a specific application still thinks of minor numbers as eight bits in size, you are going to have trouble using minor device numbers over 255.

To help during this transitional period, the AoE driver may be compiled without support for partitions. That way, instead of there being 16 minor numbers per disk, there's only one per disk. So even on systems that haven't caught up to the large minor device numbers of 2.6, you still can use up to 256 AoE disks.

Listing 2. Setting Up the Software RAID and the LVM Volume Group

```
# speed up RAID initialization
for f in `find /proc | grep speed`; do
        echo 100000 > $f
done

# create mirrors (mdadm will manage hot spares)
mdadm -C /dev/md1 -l 1 -n 2 \
        /dev/etherd/e0.0 /dev/etherd/e0.1
mdadm -C /dev/md2 -l 1 -n 2 \
        /dev/etherd/e0.2 /dev/etherd/e0.3
mdadm -C /dev/md3 -l 1 -n 2 \
        /dev/etherd/e0.4 /dev/etherd/e0.5
mdadm -C /dev/md4 -l 1 -n 2 -x 2 \
        /dev/etherd/e0.6 /dev/etherd/e0.7 \
        /dev/etherd/e0.8 /dev/etherd/e0.9
sleep 1

# create the stripe over the mirrors
mdadm -C /dev/md0 -l 0 -n 4 \
        /dev/md1 /dev/md2 /dev/md3 /dev/md4

# make the RAID 10 into an LVM physical volume
pvcreate /dev/md0

# create an extendable LVM volume group
vgcreate ben /dev/md0

# look at how many "physical extents" there are
vgdisplay ben | grep -i 'free.*PE'

# create a logical volume using all the space
lvcreate --extents 88349 --name franklin ben

modprobe jfs
mkfs -t jfs /dev/ben/franklin
mkdir /bf
mount /dev/ben/franklin /bf
```

Listing 3. To expand the filesystem without unmounting it, set up a second RAID 10 array, add it to the volume group and then increase the filesystem.

```
# after setting up a RAID 10 for the second shelf
# as /dev/md5, add it to the volume group
vgextend ben /dev/md5
vgdisplay ben | grep -i 'free.*PE'

# grow the logical volume and then the jfs
lvextend --extents +88349 /dev/ben/franklin
mount -o remount,resize /bf
```

Intel PRO/1000 MT Dual-Port NIC and puny 40GB drives. The network switch was a Netgear FS526T. With a RAID 10 across eight of the EtherDrive blades in the Coraid shelf, I saw a sustainable read throughput of 23.58MB/s and a write throughput of 17.45MB/s. Each measurement was taken after flushing the page cache by copying a 1GB file to /dev/null, and a sync command was included in the write times.

The RAID 10 in this case has four stripe elements, each one a mirrored pair of drives. In general, you can estimate the throughput of a collection of EtherDrive blades easily by considering how many stripe elements there are. For RAID 10, there are half as many stripe elements as disks, because each disk is mirrored on another disk. For RAID 5, there effectively is one disk dedicated to parity data, leaving the rest of the disks as stripe elements.

The expected read throughput is the number of stripe elements times 6MB/s. That means if Stan bought two shelves initially and constructed an 18-blade RAID 10 instead of his 8-blade RAID 10, he would expect to get a little more than twice the throughput. Stan doesn't need that much throughput, though, and he wanted to start small, with a 1.6TB filesystem.

Listing 3 shows how Stan easily can expand the filesystem when he buys another shelf. The listings don't show Stan's mdadm-aoe.conf file or his startup and shutdown scripts. The mdadm configuration file tells an mdadm process running in monitor mode how to manage the hot spares, so that they're

Network backup seems insurmountable if you don't have the right solution.

Arkeia Backup and Recovery. The Right Solution.

A proven, reliable solution. Arkeia pioneered professional network backup software for Linux. Today over 100,000 networks and 4,000 customers depend on Arkeia's data protection because it supports a wide array of professional environments and architectures-including new 64-bit architectures. As a result, Arkeia delivers robust and highly reliable data protection solutions for Linux and mixed environments.

Arkeia is fast. Our innovative multi-flow and multiplexing technologies dramatically increase backup speeds for large file servers—five times faster for 10,000 files within a single directory and up to 50 times faster for 150,000 files!

Since Arkeia software processes up to 200 data streams simultaneously, you can now complete operations within required backup windows.

Arkeia is highly scalable. Our modular architecture easily scales for different network sizes, operating systems and technologies-from a simple two-client configuration to a multi-site enterprise. As a result, our software easily keeps pace with your growing storage and data protection needs.

Arkeia offers the options you want. IT managers told us the options they want most, including:

- D2D (disk-to-disk) backup
- Bare-metal disaster recovery
- NDMP support for NAS backup
- Hot backup plug-ins for open applications and databases

Plus a lot more-all at a price you'll appreciate.

Try Arkeia's backup and recovery solutions for 30 days. FREE!

The best way to prove that Arkeia is the right solution for you is to let you try it—FREE—for 30 days. We'll even include free installation tech support. Simply download the demo version at www.arkeia.com/download.

arkeia®
ENTERPRISE BACKUP SOLUTIONS

wwwarkeia.com

ready to replace any failed disk in any mirror. See spare groups in the mdadm man page.

The startup and shutdown scripts are easy to create. The startup script simply assembles each mirrored pair RAID 1, assembles each RAID 0 and starts an mdadm monitor process. The shutdown script stops the mdadm monitor, stops the RAID 0s and, finally, stops the mirrors.

Sharing Block Storage

Now that we've seen a concrete example of ATA over Ethernet in action, readers might be wondering what would happen if another host had access to the storage network. Could that second host mount the JFS filesystem and access the same data? The short answer is, "Not safely!" JFS, like ext3 and most filesystems, is designed to be used by a single host. For these single-host filesystems, filesystem corruption can result when multiple hosts mount the same block storage device. The reason is the buffer cache, which is unified with the page cache in 2.6 kernels.

Linux aggressively caches filesystem data in RAM whenever possible in order to avoid using the slower block storage, gaining a significant performance boost. You've seen this caching in action if you've ever run a `find` command twice on the same directory.

By using a cluster filesystem such as GFS, it is possible for multiple hosts on the Ethernet network to access the same block storage using ATA over Ethernet.

Some filesystems are designed to be used by multiple hosts. Cluster filesystems, as they are called, have some way of making sure that the caches on all of the hosts stay in sync with the underlying filesystem. GFS is a great open-source example. GFS uses cluster management software to keep track of whom is in the group of hosts accessing the filesystem. It uses locking to make sure that the different hosts cooperate when accessing the filesystem.

By using a cluster filesystem such as GFS, it is possible for multiple hosts on the Ethernet network to access the same block storage using ATA over Ethernet. There's no need for anything like an NFS server, because each host accesses the storage directly, distributing the I/O nicely. But there's a snag. Any time you're using a lot of disks, you're increasing the chances that one of the disks will fail. Usually you use RAID to take care of this issue by introducing some redundancy. Unfortunately, Linux software RAID is not cluster-aware. That means each host on the network cannot do RAID 10 using mdadm and have things simply work out.

Cluster software for Linux is developing at a furious pace. I believe we'll see good cluster-aware RAID within a year or two. Until then, there are a few options for clusters using AoE for shared block storage. The basic idea is to centralize the RAID functionality. You could buy a Coraid RAIDblade or two and have the cluster nodes access the storage exported by them. The RAIDblades can manage all the EtherDrive blades behind them. Or, if you're feeling adventurous, you also could do it yourself by using a Linux host that does software RAID and exports the resulting disk-failure-proofed block storage itself, by way of ATA over Ethernet. Check out the vblade program (see Resources) for an example of software that exports any storage using ATA over Ethernet.

Backup

Because ATA over Ethernet puts inexpensive hard drives on the Ethernet network, some sysadmins might be interested in using AoE in a backup plan. Often, backup strategies involve tier-two storage—storage that is not quite as fast as on-line storage but also is not as inaccessible as tape. ATA over Ethernet makes it easy to use cheap ATA drives as tier-two storage.

But with hard disks being so inexpensive and seeing that we have stable software RAID, why not use the hard disks as a backup medium? Unlike tape, this backup medium supports instant access to any archived file.

Several new backup software products are taking advantage of filesystem features for backups. By using hard links, they can perform multiple full backups with the efficiency of incremental backups. Check out the Backup PC and rsync backups links in the on-line Resources for more information.

Conclusion

Putting inexpensive disks on the local network is one of those ideas that make you think, "Why hasn't someone done this before?" Only with a simple network protocol, however, is it practical to decouple storage from servers without expensive hardware, and only on a local Ethernet network can a simple network protocol work. On a single Ethernet we don't need the complexity and overhead of a full-fledged Internet protocol such as TCP/IP.

If you're using storage on the local network and if configuring access by creating Ethernet networks is sufficient, then ATA over Ethernet is all you need. If you need features such as encryption, routability and user-based access in the storage protocol, iSCSI also may be of interest.

With ATA over Ethernet, we have a simple alternative that has been conspicuously absent from Linux storage options until now. With simplicity comes possibilities. AoE can be a building block in any storage solution, so let your imagination go, and send me your success stories.

Acknowledgements

I owe many thanks to Peter Anderson, Brantley Coile and Al Dixon for their helpful feedback. Additional thanks go to Brantley and to Sam Hopkins for developing such a great storage protocol.

Resources for this article: www.linuxjournal.com/article/8201.

Ed L. Cashin has wandered through several academic and professional Linux roles since 1997, including Web application developer, system administrator and kernel hacker. He now works at Coraid, where ATA over Ethernet was designed, and he can be reached at ecashin@coraid.com. He enjoys music and likes to listen to audio books on his way to martial arts classes.

Scale up performance. Scale down costs.

Appro lets you scale up performance while scaling down cost. The choice and the cost savings are yours.

Appro *HyperBlade* Clusters

Full-Cluster
Up to **80** nodes

Mid-Cluster
Up to **50** nodes

Mini-Cluster
Up to **17** nodes

- Flexible, modular and scalable architecture
- Blade servers fit all cabinet sizes
- Dual AMD Opteron™ processors per node
- Up to 16GB ECC DDR Memory per node
- One PCI-X slot per node
- Up to two internal HDDs
- Remote Management Options
- High-Speed Interconnect Options: Myrinet, Dolphin, Quadrics and InfiniBand™

✔ *Specially created for the Appro* Hyper*Blade servers*
✔ *Outstanding hardware and software management tool*

Appro **BladeDome** Remote Server Management

- GUI & command line interfaces
- In-band and out-of-band control
- Remote reset and power cycle
- Platform Monitoring: fan fail, over-temperature & voltage
- Failure alert e-mail notifications
- Enhaced security
- Multiple user account set-up
- BCC redundancy

AMD Opteron™ Processors
- Integrated AMD HyperTransport™ technology allows for concurrent multiple processors in a single system.
- Shorten run-time cycles and increase bandwidth for processing computing requests.
- 32 bit applications while you migrate to 64 bit computing for long-term investment protection.

APPRO
HPC Cluster Solutions

Appro has everything you need to create a network blade cluster-ready.
On-site maintenance and installations services are also available.

For more information, please visit www.appro.com
or call Appro Sales at 800.927.5464, 408.941.8100.

AMD 64 Opteron

L'Intranet Originale

Think you can't run a real on-line community in about 64k? Try a bulletin board system.

BY MARCEL GAGNÉ

That's right, it's completely nongraphical, but there is color, *mon ami*. Why? Well, François, I suppose I'm feeling a bit nostalgic. When I read that this issue's theme would be intranets, it started me thinking about the whole idea of an intranet, literally an internal network—a private little universe, if you will, for a specific set of users. Usually, we think of a business or an organization making use of this, but intranets also are perfect for hobby or user groups. When we talk about intranets, we tend to think of Web contact management systems and portals that perform these functions.

Quoi? The text-only screen? That's easy. The original intranet existed long before we all started getting on the Internet, *mon ami*, and communication was nongraphical. *Mon Dieu*, why are we still talking? Our guests are already here. Welcome, *mes amis*, make yourselves comfortable while François brings you your wine. To the wine cellar, François. Please bring back the 2003 Coastal Sauvignon Blanc. *Vite!*

I was just telling François about the original intranets, *mes amis*. Way back when I was just getting out of my teens, I started running one of these original intranets on an old Commodore 64. They were called bulletin board systems, or BBSes. In fact, I wrote and ran my own BBS all those years ago. The one I operated had one phone line, which meant only one user could dial in at a time. This was a non-networked system, but it was an intranet and, at its peak, 40 or 50 users took advantage of it. That little trip down memory lane is why I put together a menu of BBS programs.

You might think that in this heavily graphical age, no one uses or continues to work on text-style BBS programs. In truth, many BBSes still are in operation, and developers continue to work on and develop the programs.

The first item on tonight's menu is Bryan Burns' NexusChat. NexusChat, or NChat, is an excellent BBS-style program that provides different user levels, multiple rooms, private and group chats, e-mail messaging, on-line configuration, on-line help and a whole lot more. Furthermore, you don't need to be root to run NexusChat, nor do you need to be root to install it. Start by creating a directory where you would like the chat server to be installed. For instance, I created a directory called nexuschat in my home directory. The next step is to extract the source package:

```
tar -xzvf nchat-3.31.tar.gz
cd nchat-3.31
./setup.sh
```

The questions you have to answer are pretty basic, and you can accept the defaults, with a few exceptions. When asked where you would like the binaries installed, indicate the chat directory you created earlier. The base data directory, which defaults to /home/nchat/etc, now can be an etc subdirectory wherever you chose to install it. Next, you are asked for the number of ports. That's the maximum number of people who can connect to your chat server at any given time. The default here is 15. When you have answered this last question, it's time to type `make`. After a few seconds of compiling, the final step is to create the user database. By default, you should create 999 slots for possible users.

That's it; there's no install here. The final step involves moving the etc directory to its final location manually. You also need to do the same for the nchat and userdb binaries. In my case, I chose to run the server in /home/marcel/nexuschat, so I executed the following commands:

```
mv etc /home/marcel/nexuschat
mv nchat /home/marcel/nexuschat
mv userdb /home/marcel/nexuschat
```

Switch to your NexusChat directory and prime the user database with `userdb -z -s 999`. Aside from prepping the database, you need to create the 000 user with a password of root. To start the server, which runs on port 4000 by default, simply type `/path_to/nchat`. Now, from another terminal, connect to your chat server and log in as 000:

```
telnet your_server 4000
```

Figure 1. Telnet to the NexusChat port and get this login screen.

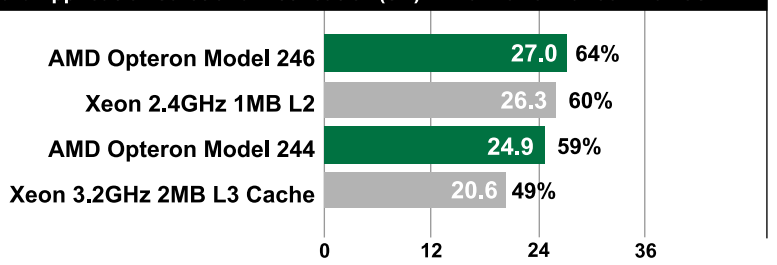
One of the first things you need to do once connected is change your password. You do that by typing `/passwd topsecret` where topsecret is the new password you choose. Once you are connected and chatting, a number of different commands are at your disposal. As with the password change command, these all begin with a slash character. To get a list of available commands, type `/?`. If, for some strange reason, you can't see what you are typing, type `/echo`.

At this point, guests also can log in. All they have to do is press Enter, and they automatically are entered as a guest. They can type `NEW` to register themselves as a user on the system, but

Receive the performance and security benefits of 64-bit computing, while getting the best 32-bit performance available anywhere.

AMD Opteron™ Processor with Direct Connect Architecture Performance

SAP Standard Application Sales and Distribution (SD) 2. Tier Performance Linux OS



AMD 64 Opteron™

ZT Optimum 1U Server A8062

- **AMD Opteron™ 244 Processor w/ 1MB L2 Cache (Upgradable to Dual Opteron™ 252 Processor)**
- **AMD-8111™ + 8131™ Chipset Server Board**
- 1GB ECC Registered DDR400 SDRAM
- 2 x Seagate® 200GB SATA 7,200RPM w/8MB Cache Hard Drive
- 2 x 1" SATA Hot-Swappable Drive Bays
- On Board 4 Channel SATA RAID Controller (Support RAID 0, 1 and 10)
- 1.44MB Floppy Drive
- 52x32x52 CD-RW & 16x DVD-ROM Combo Drive
- Dual Ports 10/100/1000 Gigabit Ethernet controllers
- 1U Rackmount Chassis W/350Watt Power Supply
- 3 Years Limited Warranty

$1,599

ZT Optimum 2U Server A8063

- **AMD Opteron™ 246 Processor w/ 1MB L2 Cache (Upgradable to Dual Opteron™ 252 Processor)**
- **AMD-8111™ + 8131™ Chipset Server Board**
- 1GB ECC Registered DDR400 SDRAM
- 2 x Seagate® 200GB SATA 7,200RPM w/8MB Cache Hard Drive
- 6 x 1" SATA Hot-Swappable Drive Bays
- On Board 4 Channel SATA RAID Controller (Support RAID 0, 1 and 10)
- 1.44MB Floppy Drive
- 52x32x52 CD-RW & 16x DVD-ROM Combo Drive
- Dual Ports 10/100/1000 Gigabit Ethernet controllers
- 2U Rackmount Chassis W/350Watt Power Supply
- 3 Years Limited Warranty

$1,699

ZT Optimum 2U Server A8064

- **AMD Opteron™ 246 Processor w/ 1MB L2 Cache (Upgradable to Dual Opteron™ 252 Processor)**
- **AMD-8111™ + 8131™ Chipset Server Board**
- 1GB ECC Registered DDR400 SDRAM
- Seagate® 73GB 10,000rpm Ultra320 SCSI Hard Drive
- 6 x 1" SCSI Hot-Swappable Drive Bays
- On Board Adaptec AIC-7902 U320 Dual Channel SCSI Controller
- 1.44MB Floppy Drive
- 52x32x52 CD-RW & 16x DVD-ROM Combo Drive
- Dual Ports 10/100/1000 Gigabit Ethernet controllers
- 2U Rackmount Chassis W/350Watt Power Supply
- 3 Years Limited Warranty

$1,899

ZT Optimum Tower Server A8065

- **AMD Opteron™ 244 Processor w/ 1MB L2 Cache (Upgradable to Dual Opteron™ 252 Processor)**
- **AMD-8111™ + 8131™ Chipset Server Board**
- 1GB ECC Registered DDR400 SDRAM
- 4 x Seagate® 300GB SATA 7,200RPM w/8MB Cache Hard Drive (Total Storage 1.2TB)
- 4 x 1" SATA Hot-Swappable Drive Bays
- 4 Channel Serial ATA Controllers (RAID 0, 1, 5, 10 , JBOD Support)
- 1.44MB Floppy Drive
- 52x32x52 CD-RW & 16x DVD-ROM Combo Drive
- Dual Ports 10/100/1000 Gigabit Ethernet controllers
- MID Tower Server Chassis w/645Watt Power Supply
- 3 Years Limited Warranty

$2,199

Please contact our system specialists for any customized systems, latest pricing and quantity orders via email to Shopper@ztgroup.com or call (866)984-7687

Quality Assured • Lifetime Tech Support • Free Shipping !

- **Your Ultimate Solution Provider**
- **Most Competitive Prices**
- **Free Gift When You Open a Business Account**
- **Onsite Service Available**
- **24x7 Lifetime Phone Support Available**
- **Reseller and Volume Pricing Available**

* Price subject to change without notice

CALL 866-ZTGROUP 9847687

Go to ztgroup.com/go/linuxjournal

ZT GROUP

Purchaser is responsible for all freight costs on all return of returns of merchandise, Full credit will not be given for incomplete or damaged returns, Absolutely no refunds for merchandise returned after 30 days, All prices and configurations are subject to change without notice and obligation, Opened software is non-refundable, All returns have to be accompanied with an RMA number and must be in re-sellable condition including all original packaging, System's picture may include some equipment and/or accessories, which are not standard features, Not responsible for errors in typography and/or photography, All rights reserved, All brands and product names, trademarks or registered trademarks are property of their respective companies,AMD, the AMD Arrow logo, AMD Athlon, Cool'n'Quiet and combinations thereof, are trademarks of Advanced Micro Devices, Inc, Windows is a registered trademark of Microsoft Corporation in the United States and/or other jurisdictions, Hyper Transport is a licensed trademark of the HyperTransport Technology Consortium, Other Product and company names used in this publication are for identification purposes only and may be trademarks of their respective companies.

the SysOp has to confirm their registration before they can log in. At this point, they can change their handles and chat with a limited set of commands. The administrator—that is, the person running the nchat program—can add permanent users or activate a self-registered user while logged in by calling up the user editor; use the `/ue username` command. You also can do this from the command line with `userdb`, the other binary that was installed. To add a user from the NexusChat directory, then, you would enter the following:

```
./userdb -a user -u -l 003 -h Francois -p 123 -t 3600
```

You are adding a user-level account (-a), there is also sysop; updating the user database (-u); creating user number 003 (-l); assigning the user a handle of Francois (-h); assigning a password of 123 (-p); and setting a session timeout of 3600 seconds (-t). If you simply type `userdb` without any options, a list of all the various options is returned.

I mentioned that the default port number was 4000. This and a few other parameters can be changed by editing the etc/nchatrc file. You likely want to change chat_name to something of your choosing, as this is the BBS' name. Some parameters, such as `ask_ansi = true`, are commented out. Also, although most terminals can handle the ANSI colors without a problem, it might be nice to offer that choice to users when they log on.

Some other interesting files are located in the etc directory. The nc_login file, for example, is what the user sees upon logging in, along with an equivalent nc_ansi_login, and nc_motd is the message of the day.

NexusChat is a lot of fun and easy to run, with minimal administrative issues. It's also quite flexible and offers simple user and chat room creation options. There's even a basic e-mail function so you can leave private messages for users that aren't currently on-line. Should you decide to try NexusChat, it's worth checking out the NexusChat Web site for a comprehensive list of its many features (see the on-line Resources).

While François refills your glasses, let's look at another example of the venerable BBS. Some programs offer more sophisticated features than NexusChat does, such as full message facilities, complex room creation—some for messaging, others just for chatting—statistical information, world clocks and calendars and more. One such BBS is Walter de Jong's bbs100.

To get bbs100 ready to use, you need to build it from source, which you can get from the bbs100 Web site (see Resources). Compiling and installing the program is fairly easy, but the steps might seem a bit strange:

```
tar -xzvf bbs100-2.1.tar.gz
cd bbs100-2.1/src
./configure --prefix=/home/bbs100
make dep
make
make install
```

In particular, notice the prefix above. It's important not to use the /usr/local default, because the BBS needs to be able to write in various directories under that prefix, and permissions may not allow it under /usr/local. I also didn't do a `make install` as root, because it isn't necessary. That said, you need to make sure your login has access to the directory in which you are trying to install. I created a /home/bbs100 directory for this particular BBS.

When you are done with the installation, switch to the installation directory, in my case /home/bbs100, and open etc/param in your favorite editor. A few settings here should be changed right away, such as the ones that include the BBS name, the port on which you want to run the program and the base directory for the installation, mostly for confirmation:

```
bbs_name                The Cellar
port_number             12345
basedir                 /home/bbs100
```

Before we move on, I suggest you take some time to become familiar with the various files in the etc directory. They include welcome screens, the message of the day, help files, system rules displayed on first login and a lot of other interesting things.

You're almost there. Because we made François the SysOp, we also need to give him a password to log in. From the directory where you installed the BBS, type `bin/mkpasswd SysOP_Name`; you then are asked for a passpharase for that user:

```
bin/mkpasswd Francois
bbs100 2.1 mkpasswd by Walter de Jong
<walter@heiho.net> (C) 2004
Enter password:
Enter it again (for verification):
OIGxutxGpuTowzw2AgMXZRkCNk
```

The last line is the SysOp's encrypted password. To let the BBS know about it, edit etc/su_passwd and enter the SysOp's name followed by a colon, followed by the encrypted passphrase:

```
Francois:OIGxutxGpuTowzw2AgMXZRkCNk
```

To start the BBS, simply type `/home/bbs100/bbs start`. Once the dæmon is running, connect from a terminal window by doing a telnet to the port you defined:

```
telnet your_system 12345
```

To change to the BBS equivalent of the superuser, or root, press the $ hot key. In this case, the superuser is known as the SysOp, or system operator. Only the person with his or her handle in the etc/su_passwd file has this hot key at his or her disposal. In all other cases, a nice calendar is displayed showing times in various worldwide locations. Once you are SysOp, you have access to a number of additional commands; simply press Ctrl-S to enter the SysOP menu. Once you are the SysOp, you have the option of configuring various system parameters, creating rooms (message as well as live chat rooms) and dealing with pesky users if need be.

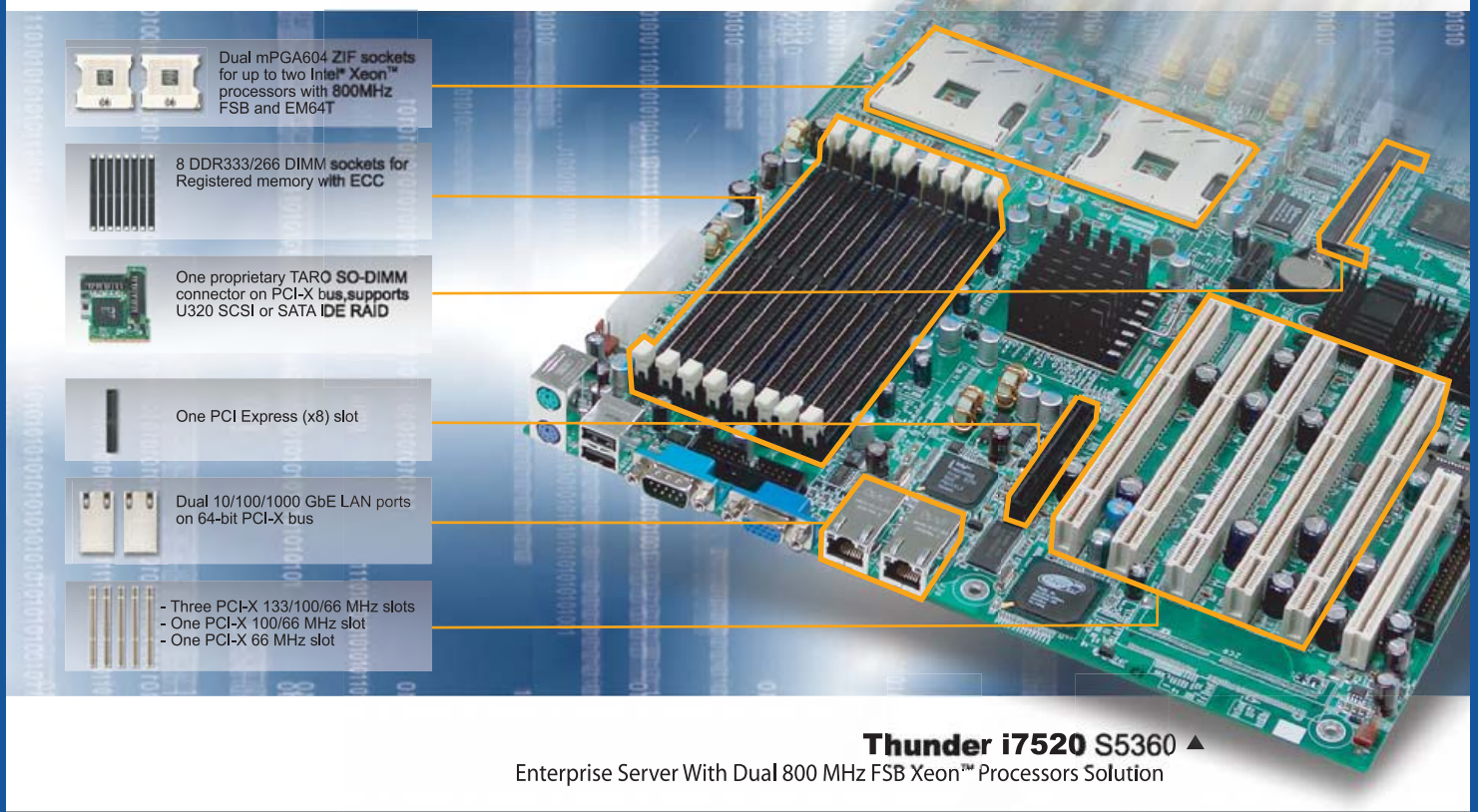
It may take some getting used to, but the BBS concept is powerful and may be a little addictive. Here's another reason to consider it. With six users on-line, my total memory usage, including the running bbs100 program, was 66,917 bytes. As you can see, *mes amis*, being smaller and simple surely had its advantages.

As we marvel at the popularity of instant messaging and cell-phone text messaging, let's remember that the roots of these technologies go back a long time. To prove my point, I'm going to end this with a little trip down memory lane. Once

TYAN

Fast, Flexible, and Feature-Rich!

PCI Express and EM64T Servers Have Arrived



Thunder i7520 S5360 ▲
Enterprise Server With Dual 800 MHz FSB Xeon™ Processors Solution

intel inside XEON

Thunder i7520 S5360

- Supports two Intel® Xeon™ processors with 800 MHz FSB and EM64T
- 8 DIMMs for DDR266/333 memory
- Three PCI-X 133/100/66 MHz slots, one PCI-X 100/66 MHz slot, one PCI-X 66 MHz slot and one 33MHz PCI slot
- One PCI Express™ x8 slot
- One proprietary SO-DIMM connector on PCI-X bus, supports U320 SCSI or SATA
- Dual GbE LAN

Tiger i7320 S5350

- Supports two Intel® Xeon™ Processor with 800MHz FSB and EM64T
- 8 DIMMs for DDR266/333 memory
- Two PCI-X 64/66 MHz slots; three 32/33 PCI 2.3 slots
- One proprietary SO-DIMM connector on PCI-X bus, supports U320 SCSI or SATA
- Dual PCI Express GbE LAN

TYAN COMPUTER CORP.

Tyan Computer USA
3288 Laurelview Court
Fremont, CA 94538 USA
Tel: +1-510-651-8868 Fax: +1-510-651-7688
Pre-Sales Tel: +1-510-651-8868 x5120
Email: marketing@tyan.com

For more information about this and other Tyan products, please contact Tyan Pre-Sales at (510) 651-8868 x5120, or contact your local Tyan system integrator/reseller.

www.tyan.com

PGI Compilers are building the 64-bit applications infrastructure.

C, C++, F77, F95 and HPF • 32-bit and 64-bit Linux
Optimized for AMD64 and IA32/EM64T • Full 64-bit support
Workstation, Server and Cluster configurations • Fast compile times
Native OpenMP • Native SMP auto-parallelization • Cache tiling
Function inlining • SSE/SSE2 Vectorization • Loop unrolling
Interprocedural optimization • Profile-feedback optimization
Large file support on 32-bit Linux • 64-bit integers and pointers
F77 pointers • Byte-swapping I/O • VAX and IBM extensions
OpenMP/MPI/threads debugging • OpenMP/MPI/threads profiling
Interoperable with g77/gcc/gdb • PDF and printed documentation
Electronic purchase, download and upgrades • Tech support
Network-floating licenses • Academic and volume discounts

Visit www.pgroup.com to download a free PGI evaluation package and see the latest tips and techniques for porting to 64-bit systems.

The Portland Group™
www.pgroup.com ++01 (503) 682-2806

The registered trademarks and marks are the property of their respective owners.

STMicroelectronics

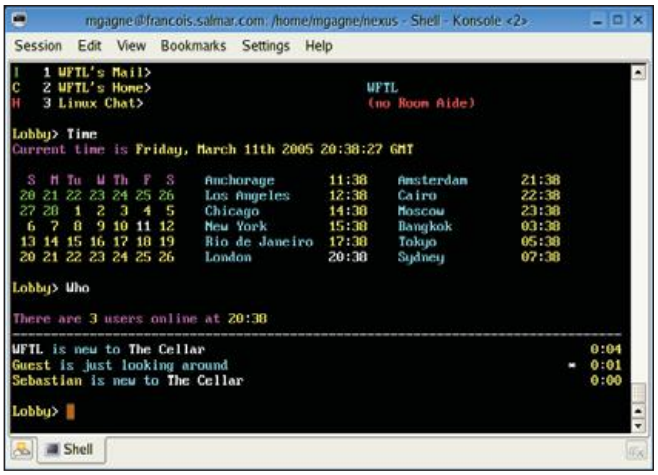


Figure 2. The bbs100 bulletin board system offers chat rooms and calendars with memory usage measured in kilobytes.

upon a time, there was a command called write and another called mesg. The mesg command allowed you to turn on your message facility like this:

```
mesg y
```

Simply stated, you were allowing others to send you messages. Now, log on to another terminal session and turn on message there as well. Let's pretend that I am logged in as marcel on one terminal and François is logged in as francois at another. He could open a chat session with me by doing this:

```
write marcel /dev/pts/16
```

He then would be able to start writing whatever he wanted, until he pressed Ctrl-D to finish the chat session. On my terminal session, I would see the following:

```
[marcel@francois marcel]$
Message from francois@francois.salmar.com on pts/14 at
19:30 ...
Hello there, Chef!
Have you decided what kind of wine we will be serving
tonight?
```

As the saying goes, *Plus ça change, plus c'est la même chose*.

It appears, *mes amis*, that closing time is once again upon us. Take your time though, and finish your conversations. In the world of text, it somehow feels easy to sit back and enjoy a glass of wine without rushing. Therefore, *mes amis*, let us all drink to one another's health. *A votre santé! Bon appétit!*

Resources for this article: www.linuxjournal.com/article/8198.

Marcel Gagné is an award-winning writer living in Mississauga, Ontario. He is the author of *Moving to the Linux Business Desktop* (ISBN 0-131-42192-1), his third book from Addison-Wesley. He also is a pilot, was a Top-40 disc jockey, writes science fiction and fantasy and folds a mean Origami T-Rex. He can be reached at mggagne@salmar.com. You can discover a lot of other things, including great WINE links, from his Web site at www.marcelgagne.com.

64-bit
LS-DYNA for
AMD Opteron

LS-DYNA is an explicit general-purpose multiphysics simulation software package used to model a wide range of complex real-world problems. It is used worldwide by automotive companies and their suppliers to analyze vehicle designs, predict the behavior of vehicles in a collision, and study occupant safety. These companies use LS-DYNA to test automotive designs to reduce the number of experimental test prototypes, saving time and expense in the design of new vehicles. Visit www.lstc.com to learn how LS-DYNA for 64-bit systems enables a new level of innovation in physics simulation.

LSTC builds 64-bit LS-DYNA for AMD Opteron processor-based systems using *PGI Compilers and Tools*.

LSTC
Livermore Software
Technology Corp.

Securing Your WLAN with WPA and FreeRADIUS, Part III

The final step in this new, more secure wireless network project includes hooking up some non-Linux clients to the new standard.

BY MICK BAUER

In the previous two Paranoid Penguin columns, I described how Wi-Fi protected access (WPA) can protect wireless LANs (WLANs) from unauthorized access and eavesdropping. I also began explaining how to use FreeRADIUS to implement WPA on your own WLAN. So far, we covered installing FreeRADIUS, creating a certificate authority (CA) and generating and signing digital certificates for WPA use. This month, I show you where to put those certificates, how to configure FreeRADIUS and how to configure your wireless access point and clients. With this information, you should be off to a good start in securing your WLAN.

A Brief Review

In case you're new to this series of articles or simply need some reminders about precisely what we're trying to achieve, let's briefly review our purpose and scope. WPA adds powerful authentication functionality to the older, cryptographically broken WEP protocol in the form of the 802.1x protocol and its subprotocols, such as EAP, PEAP and EAP-TLS. WPA also adds dynamic session key negotiation and automatic key regeneration, by way of the TKIP protocol. If your wireless client software supports WPA—that is, if it includes a WPA supplicant—and your wireless access point supports WPA, you're two-thirds of the way there already. But if you want to take full advantage of 802.1x, you need a back-end RADIUS server, which is where FreeRADIUS comes in.

In the example scenario I established last time, we're configuring a FreeRADIUS server to authenticate Windows XP wireless clients connecting to any WPA-compatible wireless access point. Our 802.1x method is EAP-TLS. EAP-TLS, you might recall, uses the TLS protocol to authenticate wireless supplicants (clients) and your access point to one another by using X.509 digital certificates.

The tasks at hand in this column are:

- To install the server and CA certificates we created last time onto our FreeRADIUS server.
- To configure FreeRADIUS to use these certificates with EAP-TLS to authenticate users for our access point.
- To configure our access point to redirect authentication to our FreeRADIUS server.
- To install the client and CA certificates we created last time onto a Windows XP client and configure it to use WPA when connecting to the WLAN.

Preparing the FreeRADIUS Server

In Part II of this WPA series, we created three X.509 digital certificates: a certificate authority certificate, called cacert.pem; one server certificate, called server_keycert.pem; and a client certificate, called client_cert.p12. The server and client files contain both a certificate and its private key, so each of these must be handled carefully. The CA certificate, however, is stored separately from its key, so you can distribute cacert.pem freely.

FreeRADIUS stores its configuration files in either /etc/raddb/ or /usr/local/etc/raddb/, depending on your distribution. This directory contains a subdirectory, certs/—this, naturally, is where you need to copy your CA certificate and your server certificate/key. Make sure that cacert.pem is owned by the user root and that its permissions are set to `-r--r--r--`. server_keycert.pem, on the other hand, should be owned by the user nobody and its permissions set to `-r--------`. Listing 1 shows the long directory listings for these two files.

As long as you're attending to file ownerships, you also should make sure that the file /var/log/radius/radius.log and the directory /var/run/radiusd/ are writable by nobody. If you compiled FreeRADIUS from source, these paths instead may be /usr/local/var/log/radius/radius.log and /usr/local/var/run/radiusd/. Both radius.log and radiusd/ may be owned by nobody.

Before we dive into FreeRADIUS' configuration files, we need to create two files that FreeRADIUS must have in order to use TLS. The first is a Diffie-Hellman parameters file, or dh file, which is used for negotiating TLS session keys. To create a dh file, change your working directory to FreeRADIUS' raddb/certs/ directory

Listing 1. Ownerships and Permissions for Certificates in raddb/certs

```
-r--r--r--  1 root   users 1294 2005-02-10 01:05 cacert.pem
-r--------  1 nobody users 1894 2005-02-10 01:00 server_keycert.pem
```

servers DIRECT™

More products, better service, guaranteed.

GO STRAIGHT TO THE SOURCE!

SERVERS DIRECT - YOUR DIRECT SOURCE FOR SCALABLE, COST-EFFECTIVE SERVER SOLUTIONS.

1U Xeon Entry Level Server
SDR-1300T

Highest performing with Dual Xeon 800MHz. Excellent with general purpose applications and provide the most power

- Intel Xeon Processor 2.8Ghz with 800FSB 1MB Cache (Dual Processor Option)
- Intel Extended Memory 64 Technology
- 1U Chassis with 420W power supply
- Supermicro server board with Intel® E7320 Chipset
- Kingston 512MB DDR400 ECC Reg. RAM (2x256MB)
- Seagate 80GB SATA 7200RPM hard drive
- 2 x 1" Hot-swap SATA drive bays
- Integrated ATI Rage XL SVGA PCI video controller
- 2x Intel® 82541GI Gigabit Ethernet Controllers
- 2x SATA Ports via 6300ESB SATA Controller RAID 0, 1 Supported

$999

2U Database Server
SDR-2101T

Highly manageable Storage server that support up to 3.2TB of SATA hot-swappable storage.

- Intel Xeon Processor 2.8Ghz with 800FSB 1MB Cache (Dual Processor Option)
- Intel Extended Memory 64 Technology
- 2U Chassis with 460W power supply
- Supermicro server board with Intel® E7520 Chipset
- Kingston 512MB DDR333 ECC Reg. RAM (2x256MB)
- Western Digital 250GB 7200RPM hot-swap SATA RAIID drive with 8MB Cache
- 8 x 1" Hot-swap SATA drive bays
- ATI RageXL 8MB Graphics
- Dual Intel® 82541GI Gigabit Ethernet Controllers
- 2x SATA Ports via 6300ESB SATA Controller RAID 0, 1 Supported

$1599

3U SCSI Storage Server
SDR-3302S

Ideal solution with high reliability Storage server in SCSI solution.

- Intel Xeon Processor 2.8Ghz with 800FSB 1MB Cache (Dual Processor Option)
- Intel Extended Memory 64 Technology
- 3U Chassis with Triple-Redundant 760W power supply
- Supermicro server board with Intel® E7520 (Lindenhurst) Chipset
- Kingston 512MB DDRII-400 ECC Reg. RAM (2x256MB)
- Seagate 36GB SCSI 10K RPM U320 SCA hard drive
- 8 x 1" hot-swap Ultra320 SCSI Drive Bays (Expandable to 15 Drives)
- ATI Rage XL SVGA PCI video controller with 8 MB of video memory
- Intel® 82546GB Dual-port Gigabit Ethernet Controllers
- Adaptec AIC-7902 Controller Dual-Channel Ultra320 SCSI Host RAID 0, 1, 10, JBOD support

$2399

4U Enterprise Server
SDR-4301T

Easily scalable storage solution with hot-swap functionality for growing business

- Intel Xeon Processor 3.0Ghz with 800FSB 1MB Cache (Dual Processor Option)
- Intel Extended Memory 64 Technology
- 4U Chassis with 460W Redundant power supply
- Supermicro server board w/Intel® E7520 Chipset
- Kingston 1024MB DDR266 ECC Reg. RAM (2x512MB)
- 3Ware 9500S-8port SATA Controller Card
- Western Digital 250GB 7200RPM hot-swap SATA RAIID drive with 8MB Cache
- 16x1" Hot-swap SATA drive bays
- ATI Rage XL SVGA PCI video controller with 8MB of video memory
- Dual Intel® 82541GI Gigabit Ethernet Controllers

$3099

5U Advanced Storage Server
SDR-5300T

Storage server with 24 hot-swap hard disk bays suitable for 9.6TB of pure data storage capacity

- Intel Xeon Processor 3.0Ghz with 800FSB 1MB Cache (Dual Processor Option)
- Intel Extended Memory 64 Technology
- 5U Chassis with 950W Triple Redundant power supply
- Supermicro server board w/Intel® E7520 Chipset
- Kingston 1024MB DDR400 ECC Reg. RAM (2x512MB)
- 3Ware 9500S-12port SATA Controller Card
- Western Digital 250GB 7200RPM hot-swap SATA RAIID drive with 8MB Cache
- 24x1" Hot-swap SATA drive bays
- ATI Rage XL SVGA PCI video controller with 8MB of video memory
- Dual Intel® 82541GI Gigabit Ethernet Controllers

$4199

5U Intel SCSI Server
SDR-5301S

Outstanding performance, excellent data protection, and advanced management for departmental servers

- Intel Xeon Processor 3.0Ghz with 800FSB 1MB Cache (Dual Processor Option)
- Intel Extended Memory 64 Technology
- Intel SC5300LX Chassis with Redundant 730W Power Supply
- Intel server board w/Intel® E7520 Chipset
- Kingston 1024MB DDR400 ECC Reg. RAM (2x512MB)
- Adaptec 2200S SCSI RAID Controller Card
- Include 6-Drive SCSI Hot-Swap Cage Kit
- 6 x Seagate 36GB SCSI 10K RPM U320 SCA hard drive
- ATI Rage XL SVGA PCI video controller with 8MB of video memory
- Dual Intel® PRO/1000 Server Network Connections

$4999

PROVEN TECHNOLOGY FROM YOUR TRUSTED SERVER SOURCE

Your business requires solid server solutions. With Servers Direct server systems based on the Intel® Xeon™ Processor, you can count on high availability, maximum efficiency and proven performance to help you meet your business reliability requirements.

1.877.727.7127 | **sales@serversdirect.com**

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

and issue this command:

```
# openssl dhparam -check -text -5 512 -out dh
```

The second file you need is a data file that contains a random bitstream that also is used in TLS operations. **Do not** simply stick the current timestamp or any other similarly nonrandom string into a file called random, as is suggested in at least one WPA procedure I've seen on the Internet. Rather, use the kernel's high-quality random number generator. From within raddb/certs, run this command:

```
# dd if=/dev/urandom of=random count=2
```

Both of these files need to be readable by the user nobody, but they should not be writable by anybody.

Configuring FreeRADIUS

We're finally ready to configure FreeRADIUS. You may be intimidated when you see the long list of files in etc/raddb, but don't be. For WPA with EAP-TLS, we need to edit only three files: radiusd.conf, eap.conf and clients.conf.

In radiusd.conf, all we need to do is set the user and group accounts that the radiusd process runs as. By default these are inherited from whatever user starts the dæmon. If you run radiusd from a startup script, this is root; however, you definitely do not want to run radiusd as root. Therefore, you should set the user and group parameters in radiusd.conf, both set to nobody, as shown in Listing 2.

Listing 2. Two Parameters to Set in radiusd.conf

```
user = nobody
group = nobody
```

Naturally you can choose different nonprivileged user and group accounts instead of nobody and nobody, but if you do so, you need to adjust the ownerships and permissions on the certificate files we tweaked earlier. Regardless, make sure your nonprivileged user's entry in /etc/password sets the user's shell to a non-shell, such as /bin/false or /bin/true—this account should not be usable for SSH, telnet or similar programs. For that matter, make sure both the user and group accounts exist in the first place, and create them if they don't.

Other parameters may be set in radiusd.conf, but these really are the only two whose default settings need to be changed. See the radiusd.conf(5) man page or Jonathan Hassell's book *RADIUS* for more information.

The next file we need to edit is eap.conf; here's where the real heavy lifting occurs. Listing 3 shows the lines you need to edit in eap.conf.

In Listing 3, I've specified a server-key passphrase with the private_key_password parameter. This actually should be empty if you created your server certificate and key with OpenSSL's -nodes option. Unfortunately, I told you to use this option in last month's column, and I'm retracting that advice now: it is poor practice to use passphrase-free X.509 keys, even when that key is stored in a clear-text configuration file such as eap.conf. Yes, if the FreeRADIUS server gets rooted—hacked into with root privileges—even a passphrase-protected certificate still can be compromised, thanks to eap.conf. But if the certificate/key file is eavesdropped in transit—when, for example, you transfer it from your CA host to your FreeRADIUS server—it is useless to the attacker if it's passphrase-protected.

Listing 3. Changes in eap.conf

```
eap {
  # There are several generic EAP parameters you can
  # set here, but the important one for our purposes
  # is default_eap_type:

  default_eap_type = tls

  # Next come parameters for specific EAP types. Since
  # we're going to use EAP-TLS, the tls{} section is
  # the one we care about:

  tls {
    # The following parameters tell radiusd where to
    # find its certs and keys, plus dh & random files:

    private_key_password = keYpasSphraSE_GOES_h3r3
    private_key_file = ${raddbdir}/certs/bt_keycert.pem
    certificate_file = ${raddbdir}/certs/bt_keycert.pem
    CA_file = ${raddbdir}/certs/cacert.pem
    dh_file = ${raddbdir}/certs/dh
    random_file = ${raddbdir}/certs/random
    }
  }
```

Either way, make sure that eap.conf is owned and readable only by root and not by the unprivileged user account you configured in radiusd.conf. This may seem paradoxical—doesn't nobody need to be able to read configuration files? But, if you start radiusd as root, it reads its configuration files, including radiusd.conf, eap.conf and clients.conf, before demoting itself to nobody.

Finally, you need to create an entry for your access point in clients.conf. Listing 4 shows such an entry.

Listing 4. Access Point Entry in clients.conf

```
client 10.1.2.3/32 {
       secret          = 1sUpErpASSw0rD
       shortname       = wiremonkeys_AP
}
```

In Listing 4, the client statement specifies the access point's IP address. Its secret parameter specifies a string that your access point uses as an encryption key for all queries it sends to your FreeRADIUS server. shortname simply is an alias for your access point to be used in log entries and so on.

You now can start radiusd by using the rc.radiusd script, for example, `rc.radiusd start`. You also could restart it with `rc.radiusd restart`. If radiusd starts without errors, you're ready to go.

Linux Lunacy '05

CRUISE THE SOUTHWESTERN CARIBBEAN

October 2 – 9, 2005

Speakers* *Andrew Dunstan, Jon "maddog" Hall, Andrew Morton, Andy Lester, Ken Pugh, Doc Searls, Ted Ts'o, and Larry Wall*

Pricing*

Conference fee: $995

Cruise/Cabin fee:

- *Inside cabin, $699*
- *Outside cabin, $799*
- *Outside w/balcony, $899*
- *Mini-suite, $999*
- *Full Suite, $1499*

Seminars: *Risk Management, Firewall Basics, Setting Up iptables, Intrusion Detection, Wireless Mayhem, Introduction to PostgreSQL, PostgreSQL and Database Basics, PostgreSQL: Advanced Topics, New Developments in ext3 Filesystem, The Linux Boot Process, Introduction to the Linux Kernel, Recovering From Hard Drive Disk Disasters, An Introduction to Voice- and Video-Over-IP, Linux Kernel Disk I/O, Linux Kernel Memory Reclaim, Linux Kernel Development*

For general information: ***http://www.geekcruises.com/top/ll05_top.htm***

**Cruise/Cabin fees are subject to change (book early to lock in these rates) and are per person based on double occupancy. Port charges and taxes, est'd to be $192, are add'l.*

SPONSORED BY: LINUX JOURNAL

geekcruises.com
EDUCATION THAT TAKES YOU PLACES

EDUCATION THAT TAKES YOU PLACES

www.geekcruises.com

Configuring the Access Point

The next step is the easiest part of this entire process: configure your wireless access point to use WPA and to point to your FreeRADIUS server. This requires only two pieces of information, the RADIUS secret you entered in your FreeRADIUS server's clients.conf file and the IP address of your FreeRADIUS server.

How you present those two pieces of information to your access point depends on your particular hardware and software. My own access point is an Actiontec DSL router with WLAN functionality. From its Web interface I clicked Setup→Advanced Setup→Wireless Settings and set Security to WPA. I then configured it to use 802.1x rather than a pre-shared key. I also provided it with a Server IP Address of 10.1.2.3, my FreeRADIUS server's IP and a Secret of 1sUpErpASSw0rD, as shown in Listing 4. I left the value for Port to its default of 1812.

Speaking of which, if your access point and RADIUS server are separated by a firewall, you need to allow the access point to reach the RADIUS server on UDP ports 1812 and 1813. Doing so also allows the RADIUS server to send packets back from those ports.

After you configure your wireless network profile, your Windows system should connect automatically to your access point and negotiate a WPA connection.

Configuring Windows XP Clients

And that brings us to configuring a Windows XP wireless client to use your newly WPA-enabled access point. This being a Linux magazine, I'm not going to describe this process in painstaking detail—for that you can see section 4.3 of Ken Roser's HOWTO, listed in the on-line Resources. In summary, you need to:

1. Run the command `mmc` from Start→Run....
2. In Microsoft Management Console, select File→Add/Remove Snap-in, add the Certificates snap-in and set it to manage certificates for My user account and, on the next screen, only for the Local computer.
3. Copy your CA (cacert.pem) certificate to your Windows system's hard drive, for example, to C:\cacert.pem.
4. From within MMC, expand Console Root and Certificates - Current User and right-click on Trusted Root Certification Authorities. In the pop-up menu, select All Tasks→Import. Tell the subsequent wizard to import the file C:\cacert.pem and to store it in Trusted Root Certification Authorities.
5. Copy your client certificate/key file to your Windows system, for example, to C:\client_cert.p12.
6. From within MMC→Console Root→Certificates, expand Personal and right-click on Certificates. In the pop-up menu, select All Tasks→Import. Tell the subsequent wizard to import the file C:\client_cert.p12.
7. The certificate-import wizard then prompts you for the certificate's passphrase. In the same dialog, it offers the option to enable strong private key protection. Unfortunately, enabling this breaks WPA, so be sure to leave this option unchecked. Also, leave the option to mark this key as exportable unchecked—you're better off backing up the password-protected file you just imported rather than allowing the imported nonprotected version to be exportable.
8. In the subsequent screen, let the wizard automatically select the certificate store.

Now your Windows XP system is ready to go—all that remains is to create a wireless network profile. This, however, varies depending on your wireless card's drivers and which Windows XP Service Pack you're running. On my Windows XP SP1 system, using a Centrino chipset and XP's native WPA supplicant, I created a wireless network profile specifying my WLAN's SSID. I set Network Authentication to WPA, Data encryption to TKIP and EAP type to Smart Card or other Certificate. Windows automatically determined which client certificate I used—this is because we took pains to create a client certificate that references Windows XP's extended attributes (see my previous column).

After you configure your wireless network profile, your Windows system should connect automatically to your access point and negotiate a WPA connection. If this succeeds, Network Connections should show a status of Authentication succeeded for your Wireless Network Connection entry.

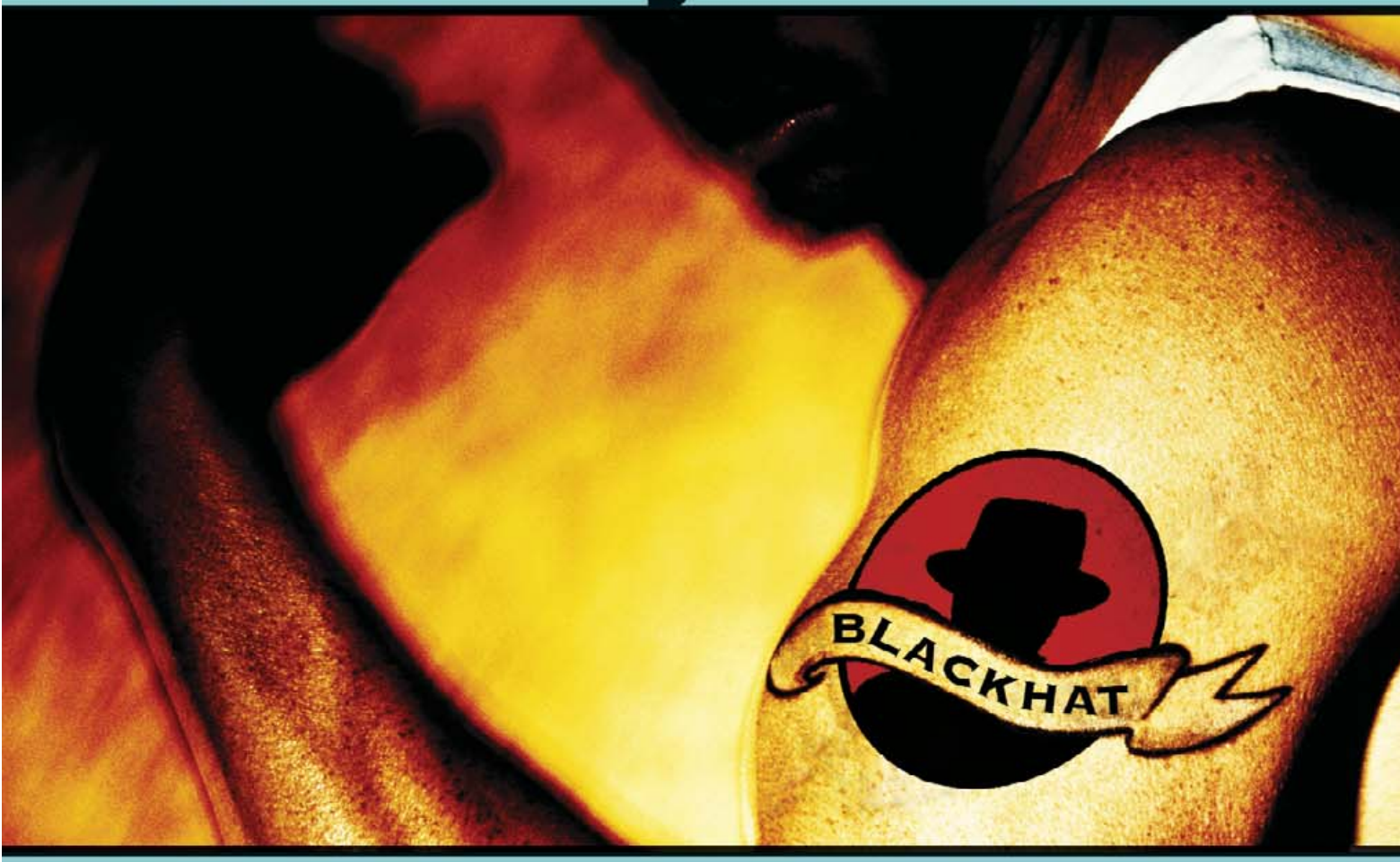
Conclusion

I hope you've gotten this far successfully and are off to a good start with WPA. WPA isn't perfect—the world needs WPA supplicants that can handle passphrase-protected client certificates without storing passphrases in clear text. But, wireless networking is, it seems, finally headed in a secure direction.

Resources for this article: www.linuxjournal.com/article/8200.

Mick Bauer, CISSP, is *Linux Journal*'s security editor and an IS security consultant in Minneapolis, Minnesota. O'Reilly & Associates recently released the second edition of his book *Linux Server Security* (January 2005). Mick also composes industrial polka music but has the good taste seldom to perform it.

Increase Your Security Muscle



Strengthen your defenses. Train your mind. Learn the threats of tomorrow, today. Be challenged by the experts who are doing innovative work. Meet and network with thousands of your peers from all corners of the world at the Black Hat Briefings USA 2005– the only technical security event to offer you the best of all worlds.

Black Hat®

Briefings & Training USA 2005

July 23-28, 2005 • Caesars Palace Las Vegas

Training: 4 days, 24 topics • Briefings: 2 days, 10 tracks, 60 speakers

www.blackhat.com
for updates and to register.

sponsors

diamond: BindView

platinum: Ernst & Young, netIQ

gold: Qualys, Configuresoft, Citadel

silver: IOActive, SAINT, watchfire, GuardedNet, Arbor, SurfControl, Foundstone, radware, nCircle, nuFORENSICS, ArcSight, Granite Edge

Real-Time and Performance Improvements in the 2.6 Linux Kernel

Work on improving the responsiveness and real-time performance of the Linux kernel holds even more promise for the future.

BY WILLIAM VON HAGEN

The Linux kernel, the core of any Linux distribution, constantly is evolving to incorporate new technologies and to improve performance, scalability and usability. Every new kernel release adds support for new hardware, but major version upgrades of the kernel, such as the 2.6 Linux kernel, go beyond incremental improvements by introducing fundamental changes in kernel internals. Many of the changes to the internals of the 2.6 Linux kernel have a significant impact on the overall performance of Linux systems across the board, independent of hardware improvements. The 2.6 kernel provides substantial improvements in system responsiveness, a significant reduction in process- and thread-related kernel overhead and a commensurate reduction in the time between when a task is scheduled and when it begins execution.

Released in late 2003, the 2.6 kernel now is the core of Linux distributions from almost every major Linux vendor in the enterprise, desktop and embedded arenas. Kernel and system performance are critical to focused markets such as embedded computing, where high-priority tasks often must execute and complete in real time, without being interrupted by the system. However, system performance and throughput in general equally are important to the increasing adoption of Linux on the desktop and the continuing success of Linux in the enterprise server market.

This article discusses the nature of real-time and system parameters that affect performance and highlights the core improvements in performance and responsiveness provided by the 2.6 kernel. Performance and responsiveness remain active development areas, and this article discusses several current approaches to improving Linux system performance and responsiveness as well as to achieving real-time behavior. Kernel and task execution performance for various Linux kernels and projects is illustrated by graphed benchmark results that show the behavior of different kernel versions under equivalent loads.

Latency, Preemptibility and Performance

Higher performance often can be realized by using more and better hardware resources, such as faster processors, larger amounts of memory and so on. Although this may be an adequate solution in the data center, it certainly is not the right approach for many environments. Embedded Linux projects, in particular, are sensitive to the cost of the underlying hardware. Similarly, throwing faster hardware and additional memory at performance and execution problems only masks the problems until software requirements grow to exceed the current resources, at which time the problems resurface.

It therefore is important to achieve high performance in Linux systems through improvements to the core operating system, in a hardware-agnostic fashion. This article focuses on such intrinsic Linux performance measurements.

A real-time system is one in which the correctness of the system depends not only on performing a desired function but also on meeting a set of associated timing constraints. There are two basic classes of real-time systems, soft and hard. Hard real-time systems are those in which critical tasks must execute within a specific time frame or the entire system fails. A classic example of this is a computer-controlled automotive ignition system—if your cylinders don't fire at exactly the right times, your car isn't going to work. Soft real-time systems are those in which timing deadlines can be missed without necessarily causing system failure; the system can recover from a temporary lack of responsiveness.

In both of these cases, a real-time operating system executes high-priority tasks first, within known, predictable time frames. This means that the operating system cannot impose undue overhead on task scheduling, execution and management. If the overhead of tasks increases substantially as the number of tasks grows, overall system performance degrades as additional time is required for task scheduling, switching and rescheduling. Predictability, or determinism, therefore is a key concept in a real-time operating system. If you cannot predict the overall performance of a system at any given time, you cannot guarantee that tasks will start or resume with predictable latencies when you need them or that they will finish within a mandatory time frame.

The 2.6 Linux kernel introduced a new task scheduler whose execution time is not affected by the number of tasks being scheduled. This is known as an O(1) scheduler in big-O notation, where O stands for order and the number in parentheses gives the upper bound on worst-case performance based on the number of elements involved in the algorithm. O(N) would mean that the efficiency of the algorithm is depen-

dent on the number of items involved, and O(1) means that the behavior of the algorithm and therefore the scheduler, in this case, is the same in every case and is independent of the number of items scheduled.

The time between the point at which the system is asked to execute a task and the time when that task actually begins execution is known as scheduling latency. Task execution obviously is dependent on the priority of a given task, but assuming equal priorities, the amount of time that an operating system requires in order to schedule and begin executing a task is determined both by the overhead of the system's task scheduler and by what else the system is doing. When you schedule a task to be executed by putting it on the system's run queue, the system checks to see if the priority of that task is higher than that of the task currently running. If so, the kernel interrupts the current task and switches context to the new task. Interrupting a current task within the kernel and switching to a new task is known as kernel preemption.

Unfortunately, the kernel cannot always be preempted. An operating system kernel often requires exclusive access to resources and internal data structures in order to maintain their consistency. In older versions of the Linux kernel, guaranteeing exclusive access to resources often was done through spin-locks. This meant the kernel would enter a tight loop until a specific resource was available or while it was being accessed, increasing the latency of any other task while the kernel did its work.

The granularity of kernel preemption has been improving steadily in the last few major kernel versions. For example, the GPL 2.4 Linux kernel from TimeSys, an embedded Linux and tools vendor, provided both an earlier low-latency scheduler and a fully preemptible kernel. During the 2.4 Linux kernel series, Robert Love of Novell/Ximian fame released a well-known kernel patch that enabled higher preemption and that could be applied to the standard Linux kernel source. Other patches, such as a low-latency patch from Ingo Molnar, a core Linux kernel contributor since 1995, further extended the capabilities of this patch by reducing latency throughout the kernel. A key concept for the TimeSys products and these patches was to replace spin-locks with mutexes (mutual exclusion mechanisms) whenever possible. These provide the resource security and integrity required by the kernel without causing the kernel to block and wait. The core concepts pioneered by these patches now are integral parts of the 2.6 Linux kernel.

Approaches to Real-Time under Linux

Three projects for real-time support under Linux currently are active: the dual-kernel approach used by the RTAI Project and by products from embedded Linux vendors, such as FSMLabs; a real-time Linux project hosted by MontaVista, an embedded Linux vendor; and freely available preemptibility and real-time work being done by Ingo Molnar and others, which is discussed openly on the Linux Kernel mailing list and which the MontaVista project depends upon. In addition to these core kernel projects, other supporting projects, such as robust mutexes and high-resolution timers, add specific enhancements that contribute to a complete solution for real-time applications under Linux.

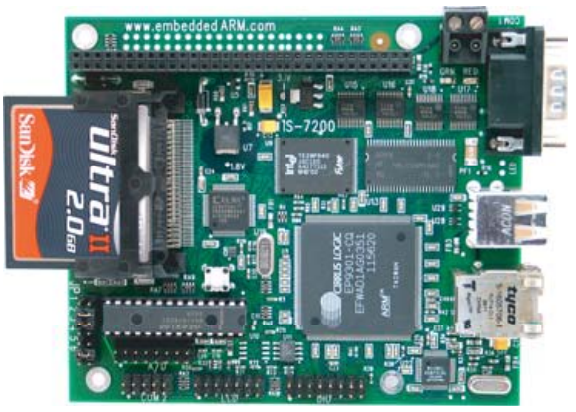
The dual-kernel approach to real time is an interesting approach to real-time applications under Linux. In this approach, the system actually runs a small real-time kernel that is not Linux, but which runs Linux as its lowest-priority process. Real-time applications specifically written for the non-Linux kernel using an associated real-time application interface execute within that kernel at a higher priority than Linux or any Linux application, but they can exchange data with Linux applications. Although this is a technically interesting approach to running real-time applications while using a Linux system, it avoids the question of general Linux kernel preemption and performance improvements. Therefore, it is not all that interesting from a core Linux development perspective.

MontaVista's project to further real-

$119 qty 100
- 200 MHz ARM9
- 10/100 Ethernet
- PC/104 bus

TS-7200 ARM9 Single Board Computer

Linux 2.4



Shown with optional Compact Flash

$149 qty 1
- 32 MB SDRAM (64 MB optional)
- 8 MB Flash (16 MB optional)
- Compact Flash
- 10/100 Ethernet
- 2 USB ports
- 20 Digital I/O
- 2 Serial Ports

Options:
- RS-485
- 8 ch 12-bit A/D
- USB WiFi

- Boots Debian stable from Compact Flash
- Boots TS-Linux from on-board Flash
- Call for custom designs

NEW
TS-7250 SBC $149
32 MB Flash
(128 MB Flash optional)

Technologic SYSTEMS
(480)-837-5200
www.embeddedARM.com

time Linux leverages much of the existing work being done by Ingo Molnar and other Linux kernel contributors, but it includes some additional prototype patches available only on the MontaVista Web site. The current patches available there are for a release candidate for the 2.6.9 Linux kernel (rc4). Therefore, they did not apply cleanly against official drops of the Linux kernel, which is moving toward 2.6.11 at the time of this writing. As such, the results from this project could not be included in this article.

The real-time, scheduling and preemptibility work being done by Ingo Molnar, the author of the O(1) Linux scheduler, and others has a significant amount of momentum, enhances the core Linux kernel and provides up-to-date patches designed to improve system scheduling, minimize latency and further increase preemptibility.

These patches have an enthusiastic following in the Linux community and include contributions from developers at many different groups and organizations, including Raytheon, embedded Linux vendors such as TimeSys and from the Linux audio community. These patches provide capabilities such as heightening system responsiveness and minimizing the impact of interrupts by dividing interrupt handling into two parts, an immediate hardware response and a schedulable interrupt processing component. As the name suggests, interrupts are requests that require immediate system attention. Schedulable interrupt handling minimizes the impact of interrupts on general system responsiveness and performance.

The illustrations in the next section focus on comparing benchmark results from various vanilla Linux kernels against those obtained by applying the real-time, scheduling and preemptibility patches done by Ingo Molnar and others. These patches are up to date and provide complete, core Linux kernel enhancements that can provide direct benefits to Linux users who want to incorporate them into their projects and products.

The Sample Benchmark

In 2002, the *Linux Journal* Web site published an article titled “Realfeel Test of the Preemptible Kernel Patch”, written by Andrew Webber. This article used an open benchmark called Realfeel, written by Mark Hahn, to compare preemption and responsiveness between the standard Linux 2.4 kernel and a kernel against which Robert Love’s preemption patch had been applied. Realfeel issues periodic interrupts and compares the time needed for the computer to respond to these interrupts and the projected optimal response time of the system. The time between the expected response and the actual response is a measurement of jitter. Jitter is a commonly used method for measuring system response and estimating latency.

This article uses the same benchmark application as Webber’s article but imposes substantially more load on the system when measuring results. This is a technique commonly applied when benchmarking real-time operating systems, because even non-real-time operating systems may exhibit low latencies in unloaded or lightly loaded situations. The graphics in the next sections also present the results differently to make it easier to visualize and compare the differences between latency on various Linux kernels.

Benchmark Results

The results in this section were compiled using a medium-strength Pentium-class system with a single 1.7GHz AMD Athlon processor and 512MB of system memory. The system was running the GNOME desktop environment and the system processes associated with the Fedora Core 3 Linux distribution, with up-to-date patches as of Feb 10, 2004. The system kernels tested were a vanilla 2.6.10 Linux kernel, the 2.6.10-1.760_FC3 kernel available as a Fedora Core 3 update, a vanilla 2.6.11-rc3 kernel and a 2.6.11-rc3 kernel with Ingo Molnar’s current real-time and preemption patch. All of these kernels were compiled against the same kernel configuration file, modulo new configuration options introduced in the newer kernel sources.

In multiprocessing operating systems such as Linux, the system never is dormant. System processes such as the scheduler always are running. If you are using a graphical user interface (GUI), interfaces such as KDE, GNOME or standard X Window system window managers always are waiting for input events and so on. In order to examine true preemptibility and real-time performance, additional load was imposed on the system by starting various processes while each set of benchmark results was being collected. As mentioned previously, the system was running GNOME with four xterms open—one to run the Realfeel benchmark, another to run a script that constantly ran recursive find and ls processes on the system’s root partition and two in which 2.6.x Linux kernels, with separate source directories, were being compiled from a clean state.

Figure 1 shows a plot of the results of the Realfeel benchmark run on a stock Fedora Core system for a period of one minute. The system was running kernel version 2.6.10-1.760_FC3, which is a 2.6.10 kernel with various patches and enhancements applied by Red Hat. Each dot in the figure represents the jitter between an interrupt request and its handling. The X axis is the sample time in 1/60 of a second. Negative jitter numbers are displayed when the system responded to the interrupt faster than the projected standard time. As you can see from the figure, a fair number of these interrupt requests were handled exactly as expected, resulting in a visibly dark line along the 0 value of the Y axis.

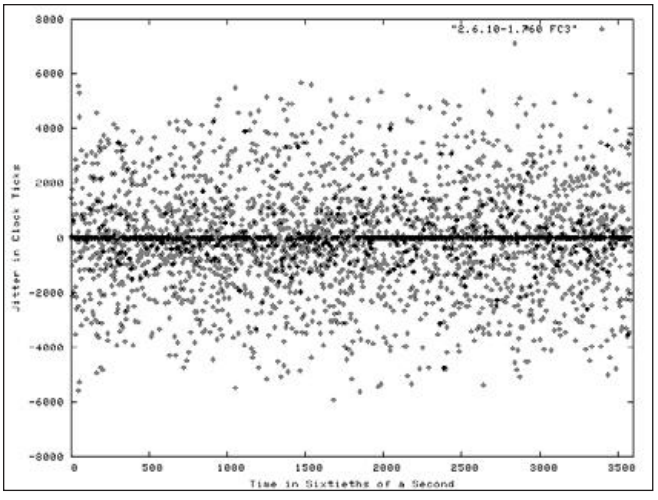


Figure 1. Jitter Results on a Stock Fedora Core Kernel

Figure 2 shows a plot of the results of the Realfeel benchmark run on the same system with a vanilla 2.6.11rc3 kernel, which is release candidate 3 of the upcoming 2.6.11 kernel. These results also were collected over a period of one minute. As you can see from these results, the 2.6.11-rc3 kernel provides improved results from the FC3 kernel, with many more instances where the jitter between an interrupt request and its handling was zero.

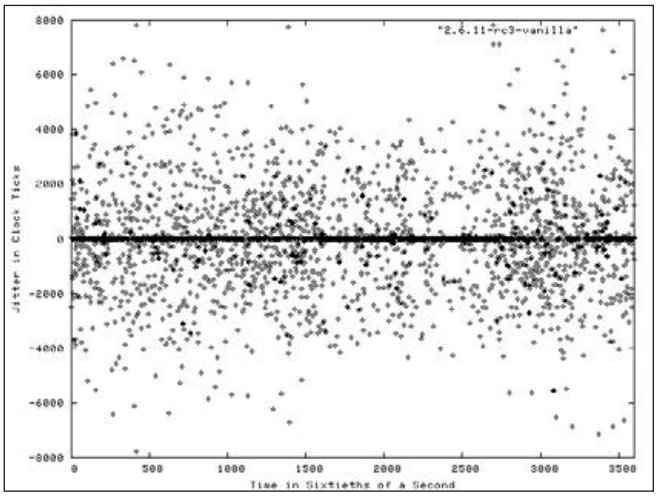


Figure 2. Jitter Results on a Vanilla 2.6.11-rc3 Kernel

Figure 3 shows a plot of the results of the Realfeel benchmark run on the same system with a 2.6.11rc3 kernel to which Ingo Molnar's real-time/preemption patches have been applied. These results also were collected over a period of one minute, with the same load generators as before. As you can see from these results, the real-time/preemption patch provides impressively better jitter results, with relatively few departures from handling interrupts within the expected period of time. On the target system, these improvements translate into a much more responsive system, on which expectations about program execution are much more predictable than they are when running the vanilla FC3 or stock 2.6.11-rc3 kernels.

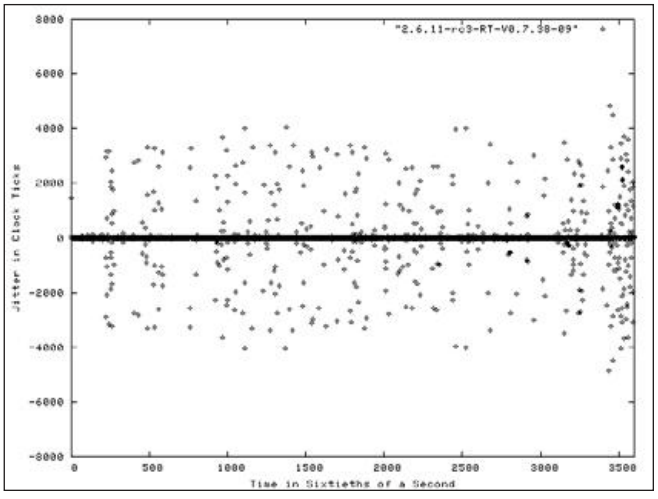


Figure 3. Jitter Results on a 2.6.11-rc3 Kernel with Real-Time/Preemption Patches

Summary

The improved scheduling, SMP and scalability improvements in the 2.6 Linux kernel provide higher-performance Linux systems than ever before, enabling them to make better use of system resources and more predictably execute kernel and user tasks as requested by the system. Further improvements are available but currently are available only by patching your system manually or by obtaining a Linux distribution from a vendor such as TimeSys, which already incorporates and tests these high-performance patches.

The very existence of GNU/Linux as a free, open-source kernel and robust execution environment is something of a marvel. The contributions of individuals and, more recently, corporations to improving its performance will lead to an even brighter future. These and other improvements to Linux argue for and help guarantee the adoption of Linux as the preferred operating system for embedded, server and desktop applications.

Resources for this article: www.linuxjournal.com/article/8199.

William von Hagen is a senior product manager at TimeSys Corporation, a leading embedded Linux and Tools vendor. He was written many books and articles on a variety of Linux and general computing topics.

We've got problems with your name on them.

At Google, we process the world's information and make it accessible to the world's population. As you might imagine, this task poses considerable challenges. Maybe you can help.

We're looking for experienced software engineers with superb design and implementation skills and expertise in the following areas:

- high-performance distributed systems
- operating systems
- data mining
- information retrieval
- machine learning
- and/or related areas

If you have a proven track record based on cutting-edge research and/or large-scale systems development in these areas, we have brain-bursting projects with your name on them in Mountain View, Santa Monica, New York, Bangalore, Hyderabad, Zurich and Tokyo.

Ready for the challenge of a lifetime? Visit us at http://www.google.com/lj for information. EOE

Google™

Schooling IT

"All happy families are alike; each unhappy family is unhappy in its own way. All was confusion in the Oblonskys' house. The wife had found out that the husband was having an affair with their former French governess, and had announced to the husband that she could not live in the same house with him."—Leo Tolstoy, *Anna Karenina*

BY DOC SEARLS

Stories are about problems. That's what makes them stories. They don't start with "happily ever after". Properly equipped with interesting causes for unhappiness, they tease us toward a resolution that arrives after dozens or hundreds of pages. That's how the Oblonsky family made great literature.

The Saugus Union School District is no Oblonsky family. It's too happy. Sure, they had problems or they wouldn't have migrated to Linux. But they did it fast and with hardly a hitch. Not great material for Tolstoy, but perhaps a useful example for similar organizations planning the same move.

Being both an educational and (after the migration) an open-source institution, Saugus Union is eager to share those lessons with their communities. So, after I asked in a recent column for migration stories, the first person to respond was Jim Klein, Director of Information Services and Technology at Saugus Union. And here I am, playing Tolstoy for the School District. That's a little lesson in PR for the rest of y'all.

The Saugus Union School District is a good-sized public school system, containing a total of 15 schools and office sites serving 11,000 students in the southern California towns of Saugus, Santa Clarita, Canyon Country and Valencia. Although the district is regarded as an exemplary public school system, it's also bucking for leadership as an exemplar of resourceful and independent IT deployment and operations. That's why the top item on its Web site is "Open Source Migration", a series of essays explaining the project and passing along wisdom for other schools.

Old-timers can guess what the district was migrating away from when Jim Klein talks about moving from one NOS—network operating system—to another. The NOS label was invented by Novell back in the 1980s. It was a positioning statement, against Microsoft's personal operating systems.

Jim writes:

> When we first decided to use Novell solutions for our primary NOS, it was really a no-brainer. Microsoft's Windows NT was the only real alternative (sorry to those of you who were LANtastic fans), and it didn't scale well for our 13 (at the time) locations (I won't even go into the reliability issue, because I'm sure most of us remember the days of weekly, scheduled reboots). Over the years, we have continued to upgrade and stay current with Novell solutions, all the while giggling as we read of the pain and suffering in Redmond's world.

They kept up with what was happening in Redmond, of course, because they used Microsoft Windows on plenty of desktops, even if they kept it off the servers. Also, Jim adds, "Let's face it, Novell wasn't winning any popularity contests." This is when they were learning about what happens when you're stuck inside a vendor's slowly depopulating silo.

Jim adds:

> Then a funny thing happened—Novell acquired SUSE in January 2004 and announced shortly thereafter that it would be moving all of its services to Linux. We had taken only a casual glance at Linux up until that point and were seriously considering Apple's Mac OS X server as a possible migration option for some of our services. With Novell throwing its weight behind Linux, especially as an enterprise server platform (instead of an application-specific server, as Linux is so often relegated to in the media), we decided to take a more serious look.

Because they wanted what they were accustomed to getting from Novell—training, a choice of applications, documentation and support—they quickly narrowed their choices to SUSE and Red Hat. Jim continues:

> Because of our Novell background, our first choice was to look at SUSE. Novell was more than happy to provide us with CDs, and although we knew little of SUSE short of vague references, we went forward with our evaluation. After running the installer several times (before we got it to take), we looked at the basic functionality of the system. We really didn't like the "jello-like" interface very much and had many issues getting some of the most basic functions to work. So it was off to the bookstore.
>
> We knew from our research that SUSE was the number-two Linux distribution on the market, so we were quite surprised to find zero, that's right, zero books on SUSE Linux. The best we could find were vague references in more generalized Linux documentation. Red Hat documentation, on the other hand, was in abundance and on a variety of topics of interest. So we bought a Red Hat book, which had a free Fedora DVD in it—Red Hat: 1, SUSE: 0. Fedora installed on the first try, and with the help of some good documentation, we were able to get basic services working—Red Hat: 2, SUSE: 0. We explored more advanced functionality, both desktop and server-oriented, and found that most Web resources were, once again, Red Hat-oriented. We were able to get Fedora to do just about anything we wanted—Red Hat: 3, SUSE: 0.
>
> But we hadn't given up on SUSE yet. Armed with a laptop, loaded with both SUSE and Fedora, we headed off to Novell's Brainshare 2004 conference in early April. Here we talked to everyone about every topic of concern. We gleaned all we could about Linux in the enterprise, spoke to techs about our concerns, looked at Novell's solutions and so on. We spoke to HP about our servers, explaining our concern over Linux compatibility with our older machines. They recommended Red Hat.

> We looked at Novell Nterprise Linux Services and discovered nothing unique about the implementations, other than that they were standard open-source apps installed in strange locations. We heard promises of real training programs somewhere down the road and that documentation would be coming soon. By the end of the conference, Novell had convinced us of two things: 1) Linux is, in fact, ready for the enterprise, and 2) that we didn't need them anymore. (Okay, that's a little harsh—we are still using Novell GroupWise—on our Red Hat servers.)

The next step was what Jim calls "trial by fire": installing Linux on all the staff laptops and running "solutions for everything we do on a day-to-day basis". After a month of "self-induced pain and frustration", they were well conditioned for off-site RHCE (Red Hat Certified Engineer) "boot camp" training. They also accumulated piles of books and other documentation and set to work evaluating open-source replacements for the applications they had been running on NetWare. Jim adds, "Our goals rapidly evolved from potentially using Linux for some services to definitely using it for several services to 'can we use it for everything?' to 'wow, I think we can use it for everything we do.'"

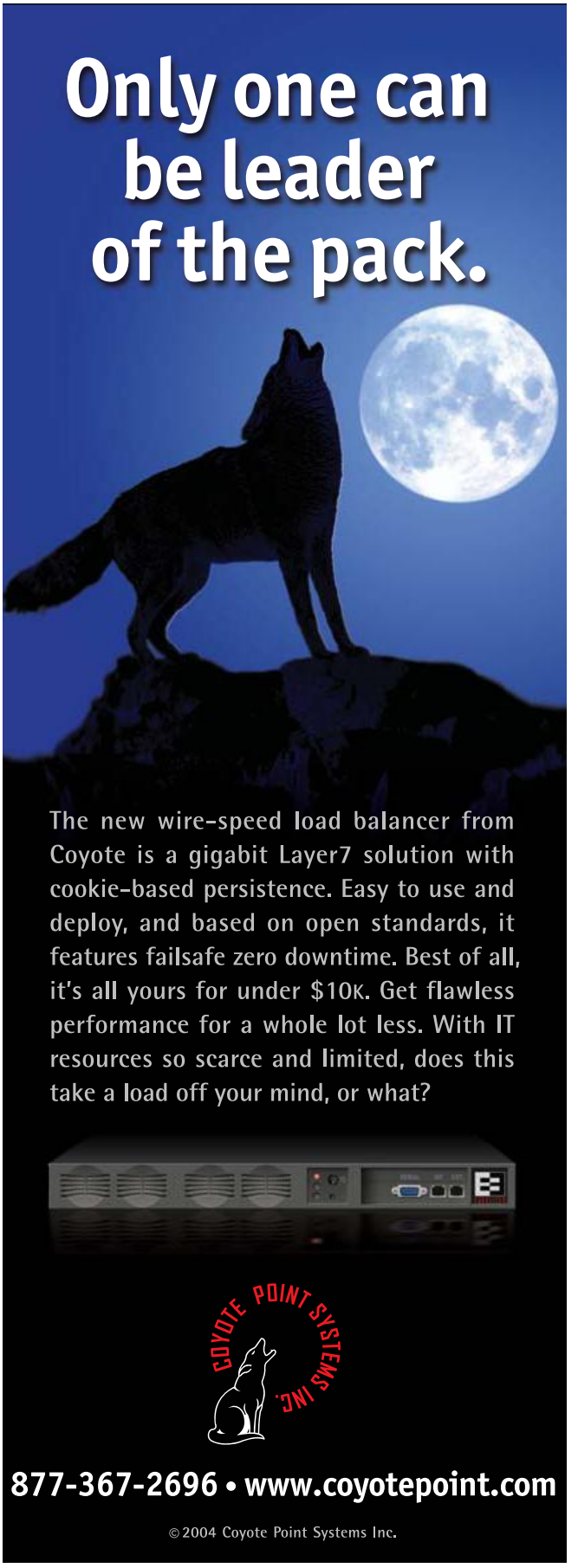
Jim's advice: "...it is important to establish, well in advance, which services you need to provide, and what solution will provide said services. In some cases, options may be a little sparse, while in others, myriad. In either case, good documentation and research are critical to any implementation."

Jim's use of the term services may seem innocuous, but it originates in Novell's intentional shift of the network paradigm in the 1980s. Before that shift, every network was a silo of proprietary offerings standing on a platform of "pipes and protocols" with names like DECnet, WangNet, OmniNet, Sytek, 3Com, Ungermann-Bass, Corvus and IBM's Token Ring. With NetWare, Novell provided the first network operating system that would run on anybody's pipes and protocols and also on anybody's hardware. As a platform, NetWare hosted a variety of network services, starting with file and print. Craig Burton, who led Novell's NOS strategy, called the new paradigm the "network services model". Services included file, print, management, messaging and directory, among others, eventually including Web. This is the conceptual model by which we still understand networks today. It's also one in which Linux makes a great deal of sense—and why NetWare isn't too hard to replace.

The main services Jim and his crew wanted to support—directory, file, print, Web, messaging (e-mail), DNS/DHCP and backup—had Novell offerings that easily were replaced by OpenLDAP, Samba, Netatalk, Apache, BIND 9, dhcpd, Squid and Bacula ("dumb name, great solution", Jim writes). The only remaining exception was Novell GroupWise 6.5, which lives on as a proprietary application running on Linux.

They deployed gradually, starting with nonessential edge servers and working their way to core servers and services:

> We updated a Web server at the district office first and gradually added services to it for testing purposes. Then, we updated the Web, proxy and DHCP servers at two school sites. We added Samba to the servers so that Webmasters could update their sites. Then we convinced an administrator to let us install Linux on 30 laptops in a wireless cart. We learned a great deal by starting small and building up to more and more services,



> and the laptops taught us how to "script" the installation and rapidly deploy through the use of Red Hat's Kickstart utility. Finally, it was summer, and it was time for the bold step—full migration of 14 sites totaling 42 servers in six weeks.

They deployed everything at the server end, including automated backups for multiple PC platforms, in four weeks. Then they went out to the mass of clients throughout the school district:

> When the office staff returned and were given their passwords (we had to change them as we are now on a completely different authentication system), they went right to work. We proceeded busily to remove the Novell software (except GroupWise) and join the new Windows domains (on the Samba servers) on our 3,000 or so Windows machines in our school classrooms and to update aliases and so forth on about 1,000 Macs....
>
> When all that was said and done, we were pleasantly surprised by how smoothly the transition went. While our 800 or so users (and 11,000 students) may know that we are running Linux, it is relatively transparent to them. The Linux servers offer no indication that they are running Linux. To the Windows machines, they look like Windows servers. The Macs think they are Apple servers. Everything just works. Sure, we were in a continual state of tweaking for a while, which was understandable under the circumstances, but we did not (and have not) had a single "show-stopper" of a problem.

The dollar savings weren't small, especially for a school system. Nearly $54,000 US in licensing fees to Novell, plus $50–$200 per desktop workstation. Less measurable but even more gratifying are the ongoing time and hassle savings:

> We are now able to install software, even if it has a GUI installer, remotely, which has saved us a tremendous amount of time. Software management and configuration is not only consistent, but accessible and easily modified, as opposed to being hidden away somewhere in an obscure directory object, registry entry or other mysterious location. In addition, the myriad of management and configuration tools that were required to manage the servers has been reduced, for all intents and purposes, to one. And, thanks to the Red Hat Network, we now know, in an instant, the status of all of our machines and what patches are needed and are able to schedule automated updates district-wide at the click of a mouse.
>
> Perhaps the most interesting benefit we have enjoyed has been our newfound ability to modify solutions to meet our needs....We have, on numerous occasions, changed the way a script works or added functionality to a software package. For example, we use the idealx-smbldap Perl scripts to add, modify and delete Samba accounts from the LDAP directory. These scripts, however, did not offer the ability to add such attributes as a user's first name or title, which we needed for some of the Web applications we are using. So, with absolutely no Perl experience (although reasonable scripting/programming experience), we were able to add this functionality to the scripts and enjoy the new functionality immediately.

I was surprised that they deployed first on laptops, which are notoriously less "white-box-like" than desktops. Sleep, for example, has always been an issue.

Jim said:

> We used HP NX5000s mostly, quite a long time before they started shipping SUSE on them, however. We also used NC4000s and NC6000s. We put Fedora Core on all of them, and do our installs via Kickstart. The big benefit of Fedora is that we can host a local yum repository and mirror Fedora updates (as well as other sites), which makes it easy (and fast) to distribute software and updates, through Red Hat's up2date. We don't like SUSE very much, because of the way it litters all the files all over the filesystem. It adds an extra step when you are trying to find help, as you first have to figure out what SUSE did with all of the pieces.
>
> Sleep still doesn't work right. There are some nice kernel patches to make them hibernate, but they are a bit of work to install. We couldn't get built-in 2.6 hibernate functions to work either. This is, by far, our biggest headache with laptops. We have two batteries in all of ours, though, so we can keep them running for the day with relative ease.
>
> On the other hand, the laptops running Linux are working great. And we've had no problems getting users to adjust. In fact, the only instruction we've offered is, "The little red hat in the start bar is the same as the Start button on Windows", and "Firefox is your Internet browser." They've been fine with all the rest. In fact, even trainers we've brought in from outside have had no problem adjusting to the machines and completing their tasks.

Craig Burton says "There are always two kinds of problems, technical and political. And the technical problems are usually easiest to solve." Jim told me, "The biggest help we got from Novell was political, as they added credibility to open source through their name and industry recognition." But, he added, "We encountered no political problems (and) almost no resistance because we came in fully informed, with all the right answers."

I asked where he went for help during the migration. Jim replied, "Actually, Red Hat and the Web were our sources. RHCE boot camp got me up on the enterprise side of things and the Web worked for everything else. I was surprised at how much help I got from SourceForge forums and the like—even from the programmers themselves. I put my techs through Linux Professional Institute boot camp. One will attend RHCE in the Spring."

I told Jim I often hear that, at large companies, migration is a trade of licensing costs for personnel time. Was this also the case here? "I suppose first year, you could say that", he said. "If I consider cost in terms of our salaries and the amount of time we put into learning and doing, training fees and support fees, you could say we broke even. But then, we consider learning and research part of our job description. Outside of salaries and time, actual cash outlays were only $6,700, and savings are $50K+ per year, so I'd say we came out ahead." Today, the district is running Red Hat Enterprise Linux 3 AS on 31 servers, and Fedora Core 1 on 11 older servers that don't meet the minimum hardware requirements for the Enterprise product.

What were the licensing fees for exactly, I asked. Jim replied, "We were a Novell shop, so it's almost all Novell fees. Generally, it's $3 a kid for Novell ed licenses—we have 11,000

students. The rest would be Veritas Backup Exec maintenance, Surf Control and so on."

When I asked about remaining problem areas, for higher-level application migration, he said:

> The problem with that move is compatibility with some of the multiuser educational software we use. Quarter Mile Math, Follett Library Automation, Renaissance's Accelerated Reader, Scholastic's Reading Counts and Orchard Software don't have Linux clients. We have pretty healthy investments there. We have experimented with Follett under Wine, and found that we can make the classroom portion work, but have not, as yet, looked at the others.

I asked about adoption prospects at the desktop level. "Several site administrators have expressed an interest in Linux desktops as an avenue for acquiring more machines for the same money, that is, to pay less Microsoft tax", Jim said. "Most of the immediate impact has been an increased awareness of what's out there in open source. They use the Linux laptops for training and learn that they can use the same applications on their existing machines for free as well. Right now we have multiple sites experimenting with open source on Windows and Mac OS X, via OpenOffice.org, The Gimp and so on."

As for commercial educational software vendors, Jim adds:

> We've seen a fair amount of interest. For example, Follett server already runs on Linux, and we helped Quarter Mile get its Java-based server to run on Linux as well. I believe Scholastic is using a Java-based client now, which would require minimal tweaking. Better support will probably require pressure from a few good-sized districts. As we see upgrades coming, we try to force the issue a bit.

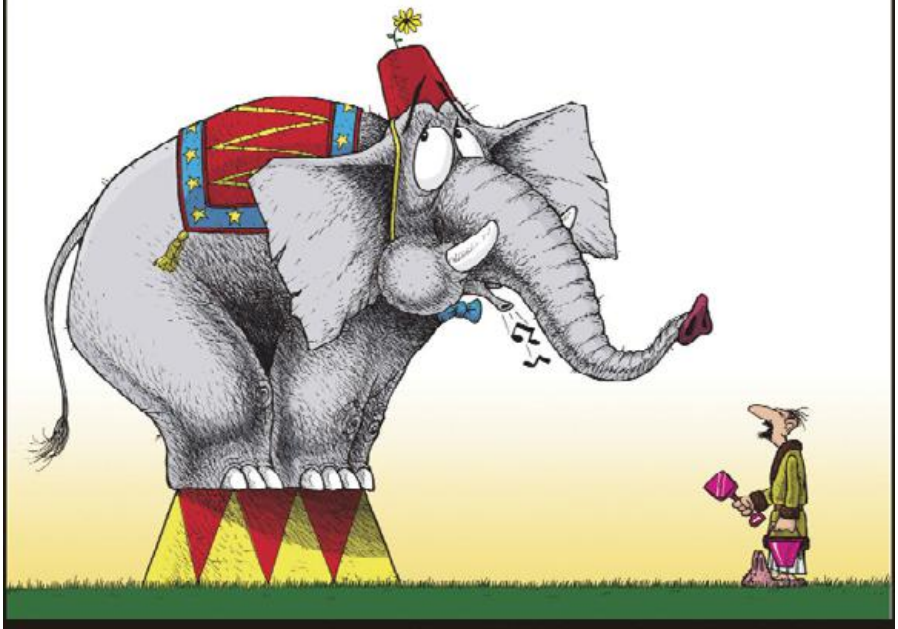
Finally, I asked him if his experience offered lessons for business enterprises. He replied:

> I think the biggest thing is that Linux can be done successfully, on a multi-site enterprise scale, and that Linux truly is enterprise-ready. Most of what they hear from the Microsoft camp is simply inaccurate or incomplete analysis. We've already recouped our costs, and more, and are thrilled with performance, reliability and security. Add the fact that "patch management" doesn't have to take up an entire salary, and you'll find that there's more time for innovating and less required for maintaining. I've rebooted my servers once since last September, and it was because I wanted them to reboot, not because they needed to or did it spontaneously on their own.

If you want to know more, I'm sure Jim will keep reports current at the Saugus Union School District Web site (www.saugus.k12.ca.us). The story might not be worthy of Tolstoy, but it might be worth a lot for the thousands of other school systems and mid-sized enterprises planning similar moves.

Doc Searls is Senior Editor of *Linux Journal*.

ONCE AGAIN,
HEAP PROBLEMS HAD
SPOILED CODY'S DAY



Characters and Images ©2004 Brad Fitzpatrick, ActiveEdge. All Rights Reserved.

Debugging heap allocation problems can be a real chore, but TotalView now has built-in memory features that track memory usage for all processes and can even stop execution at the point that a memory problem occurs. And it's all integrated, so there's no need to interrupt your debug session to invoke an external memory tool. Etnus TotalView is also the best threads debugger available and offers superior C++ support. So, don't forget to download a free fully functional trial of TotalView today.

Try TotalView FREE at www.etnus.com

TotalView, the Most Advanced Debugger on Linux and UNIX

ETNUS
TOTALVIEW

Database Replication with Slony-I

Whether you need multiple instances of your database for high availability, backup or for a no-downtime migration to a new version, this versatile tool will keep all of them in sync.

BY LUDOVIC MARCOTTE

Database management systems have been a crucial component of infrastructures for many years now. PostgreSQL is an advanced, object-relational database management system that is frequently used to provide such services. Although this database management system has proven to be stable for many years, the two available open-source replication solutions, rserv and ERServer, had serious limitations and needed replacement.

Fortunately, such a replacement recently became available. Slony-I is a trigger-based master to multiple slaves replication system for PostgreSQL being developed by Jan Wieck. This enterprise-level replication solution works asynchronously and offers all key features required by data centers. Among the key Slony-I usage scenarios are:

- Database replication from the head office to various branches to reduce bandwidth usage or speed up database requests.
- Database replication to offer load balancing in all instances. This can be particularly useful for report generators or dynamic Web sites.
- Database replication to offer high availability of database services.
- Hot backup using a standby server or upgrades to a new release of PostgreSQL.

This article walks you through the steps required to install Slony-I and replicate a simple database located on the same machine. It also describes how Slony-I can be combined with high-availability solutions to provide automatic failover.

Installing Slony-I

To install Slony-I and replicate a simple database, first install PostgreSQL from source. Slony-I supports PostgreSQL 7.3.2 or higher; 7.4.x and 8.0 need the location of the PostgreSQL source tree when being compiled. If you prefer using PostgreSQL packages from your favorite distribution, simply rebuild them from the package sources and keep the package build location intact so it can be used when compiling Slony-I. That said, obtain the latest Slony-I release, which is 1.0.5, compile and install it. To do so, proceed with the following commands:

```
% tar -zxvf slony1-1.0.5.tar.gz
% cd slony1-1.0.5
% ./configure \
--with-pgsourcetree=/usr/src/redhat/BUILD/postgresql-7.4.5
% make install
```

In this example, we tell the Slony-I's configure script to look in /usr/src/redhat/BUILD/postgresql-7.4.5/ for the location of the PostgreSQL sources, the directory used when building the PostgreSQL 7.4.5 RPMs on Red Hat Enterprise Linux. The last command compiles Slony-I and installs the following files:

- $postgresql_bindir/slonik: the administration and configuration script utility of Slony-I. slonik is a simple tool, usually embedded in shell scripts, used to modify Slony-I replication systems. It supports its own format-free command language described in detail in the Slonik Command Summary document.
- $postgresql_bindir/slon: the main replication engine. This multithreaded engine makes use of information from the replication schema to communicate with other engines, creating the distributed replication system.
- $postgresql_libdir/slony1_funcs.so: the C functions and triggers.
- $postgresql_libdir/xxid.so: additional datatype to store transaction IDs safely.
- $postgresql_datadir/slony1_base.sql: replication schema.
- $postgresql_datadir/slony1_base.v73.sql.
- $postgresql_datadir/slony1_base.v74.sql.
- $postgresql_datadir/slony1_funcs.sql: replication functions.
- $postgresql_datadir/slony1_funcs.v73.sql.
- $postgresql_datadir/slony1_funcs.v74.sql.

- $postgresql_datadir/xxid.v73.sql: a script used to load the additional datatype previously defined.

Generally, $postgresql_bindir points to /usr/bin/, $postgresql_libdir to /usr/lib/pgsql/ and $postgresql_datadir to /usr/share/pgsql/. Use the `pg_config --configure` command to display the parameters used when PostgreSQL was built to find the various locations for your own installation. Those files are all that is needed to offer a complete replication engine for PostgreSQL.

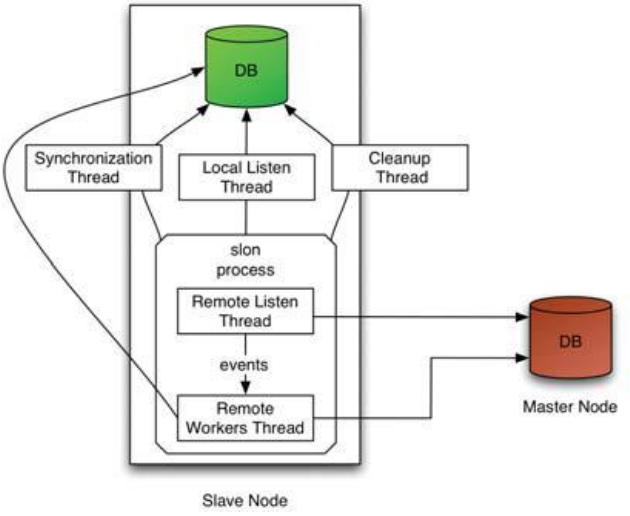


Figure 1. How the Slony-I replication engines work for a master with a slave database.

As you can see in Figure 1, Slony-I's main replication engine, slon, makes use of many threads. The synchronization thread verifies at a configurable interval if there has been replicable database activity, generating SYNC events if such activity happens. The local listen thread listens for new configuration events and modifies the cluster configuration and the in-memory configuration of the slon process accordingly.

As its name implies, the cleanup thread performs maintenance on the Slony-I schema, like removing old events or vacuuming the tables. The remote listen thread connects to the remote node's database to receive events from its event provider. When it receives events or confirmations, it selects the corresponding information and feeds the internal message queue of the remote workers thread. The replication data is combined into groups of transactions. The remote workers thread, one per remote node, does the actual data replication, events storing and generation of confirmations. At any moment, the slave knows exactly what groups of transactions it has consumed.

Replicating a Small Database

We first create the database we will replicate. This database contains a single table and sequence. Let's create a user contactuser, the contactdb database and activate the plpgsql programming language to this newly created PostgreSQL database by proceeding with the following commands:

```
% su - postgres
```


ASA COMPUTERS
www.asacomputers.com
1-800-REAL-PCS

Hardware Systems For The Open Source Community–Since 1989
(Linux, FreeBSD, NetBSD, OpenBSD, Solaris, MS, etc.)

The AMD Opteron™ processors deliver high-performance, scalable server solutions for the most advanced applications. Run both 32- and 64-bit applications simultaneously

AMD Opteron™ Value Server—$795
- 1U 14.3" Deep
- AMD Opteron™ 240
- 512MB RAM *Max 8GB*
- 40GB IDE HDD
- 2x 10/100/1000 NIC
- Options: CD, FD or 2nd HD, RAID

Front I/O Dual AMD Opteron™ Cluster Node—$1,850
- 1U Dual AMD Opteron™ Capable Font I/O
- Single 240 AMD Opteron™
- 1GB RAM *Max RAM 16GB*
- 80GB HDD
- Dual PCI Expansion Slot

8 Hot Swap Bays in 2U AMD Opteron™—$1,950
- 1 of 2 AMD Opteron™ 240
- 512MB RAM *Max 16GB*
- 3x80GB IDE RAID # 5
- 2xGigE, CD+FD
- Options: SATA/SCSI, Redundant PS

No Frills AMD Opteron™ Storage Server—$12,050
- 6TB+ IDE/SATA Storage in 5U
- Dual AMD Opteron™ 240
- 512MB RAM
- 6TB IDE Storage
- Dual GigE, CD
- Options: SATA HDD, DVD+RW etc.

Your Custom Appliance Solution
Let us know your needs, we will get you a solution
"Your Logo Here"

Custom Server, Storage, Cluster, etc. Solutions
Please contact us for all type of SCSI to SCSI, Fibre to SATA, SAN Storage Solutions and other hardware needs.

AMD 64 Opteron

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

Prices and availability subject to change without notice. Not responsible for typographical errors. All brand names and logos are trademark of their respective companies.

```
% createuser --pwprompt contactuser
Enter password for user "contactuser": (specify a
password)
Enter it again:
Shall the new user be allowed to create databases?
(y/ n) y
Shall the new user be allowed to create more new
users? (y/ n) n

% createdb -O contactuser contactdb
% createlang -U postgres -h localhost plpgsql \
contactdb
```

Then, we create the sequence and the table in the database we will replicate and insert some information in the table:

```
% psql -U contactuser contactdb

contactdb=> create sequence contact_seq start with 1;

contactdb=> create table contact (
 cid          int4 primary key,
 name         varchar(50),
 address      varchar(255),
 phonenumber  varchar(15)
);

contactdb=> insert into contact (cid, name, address,
phonenumber) values ((select nextval('contact_seq')),
'Joe', '1 Foo Street', '(592) 471-8271');
contactdb=> insert into contact (cid, name, address,
phonenumber) values ((select nextval('contact_seq')),
'Robert', '4 Bar Roard', '(515) 821-3831');
contactdb=> \q
```

For the sake of simplicity, let's create a second database on the same system in which we will replicate the information from the contactdb database. Proceed with the following commands to create the database, add plpgsql programming language support and import the schema without any data from the contactdb database:

```
% su - postgres
% createdb -O contactuser contactdb_slave
% createlang -U postgres -h localhost plpgsql \
contactdb_slave
% pg_dump -s -U postgres -h localhost contactdb | \
psql -U postgres -h localhost contactdb_slave
```

Once the databases are created, we are ready to create our database cluster containing a master and a single slave. Create the Slonik cluster_setup.sh script and execute it. Listing 1 shows the content of the cluster_setup.sh script.

The first slonik command (cluster name) of Listing 1 defines the namespace where all Slony-I-specific functions, procedures, tables and sequences are defined. In Slony-I, a node is a collection of a database and a slon process, and a cluster is a collection of nodes, connected using paths between each other. Then, the connection information for node 1 and 2 is specified, and the first node is initialized (init cluster). Once completed, the script creates a new set to replicate, which is essentially a collection containing the public.contact table and the public.contact_seq sequence. After the creation of the set, the script adds the contact table to it and the contact_seq sequence. The store node command is used to initialize the second node (id = 2) and add it to the cluster (sql_cluster). Once completed, the scripts define how the replication system of node 2 connects to node 1 and how node 1 connects to node 2. Finally, the script tells both nodes to listen for events (store listen) for every other node in the system.

Once the script has been executed, start the slon replication processes. A slon process is needed on the master and slave nodes. For our example, we start the two required processes on the same system. The slon processes must always be running in order for the replication to take place. If for some reason they must be stopped, simply restarting allows them to continue where they left off. To start the replication engines, proceed

Listing 1. cluster_setup.sh

```
#!/bin/sh

CLUSTER=sql_cluster
DB1=contactdb
DB2=contactdb_slave
H1=localhost
H2=localhost
U=postgres

slonik <<_EOF_

cluster name = $CLUSTER;

node 1 admin conninfo = 'dbname=$DB1 host=$H1 user=$U';
node 2 admin conninfo = 'dbname=$DB2 host=$H2 user=$U';

init cluster (id = 1, comment = 'Node 1');

create set (id = 1, origin = 1,
        comment = 'contact table');

set add table (set id = 1, origin = 1, id = 1,
         full qualified name = 'public.contact',
         comment = 'Table contact');

set add sequence (set id = 1, origin = 1, id = 2,
         full qualified name = 'public.contact_seq',
         comment = 'Sequence contact_seq');

store node (id = 2, comment = 'Node 2');
store path (server = 1, client = 2,
       conninfo = 'dbname=$DB1 host=$H1 user=$U');

store path (server = 2, client = 1,
            conninfo = 'dbname=$DB2 host=$H2 user=$U');

store listen (origin = 1, provider = 1, receiver = 2);
store listen (origin = 2, provider = 2, receiver = 1);
```

with the following commands:

```
% slon sql_cluster "dbname=contactdb user=postgres" &
% slon sql_cluster "dbname=contactdb_slave user=postgres" &
```

Next, we need to subscribe to the newly created set. Subscribing to the set causes the second node, the subscriber, to start replicating the information of the contact table and contact_seq sequence from the first node. Listing 2 shows the content of the subscription script.

Listing 2. subscribe.sh

```
#!/bin/sh

CLUSTER=sql_cluster
DB1=contactdb
DB2=contactdb_slave
H1=localhost
H2=localhost
U=postgres

slonik <<_EOF_

cluster name = $CLUSTER;

node 1 admin conninfo = 'dbname=$DB1 host=$H1 user=$U';
node 2 admin conninfo = 'dbname=$DB2 host=$H2 user=$U';

subscribe set (id = 1, provider = 1, receiver = 2, forward = yes);
```

Much like Listing 1, subscribe.sh starts by defining the cluster namespace and the connection information for the two nodes. Once completed, the subscribe set command causes the first node to start replicating the set containing a single table and sequence to the second node using the slon processes.

Once the subscribe.sh script has been executed, connect to the contactdb_slave database and examine the content of the contact table. At any moment, you should see that the information was replicated correctly:

```
% psql -U contactuser contactdb_slave
contactdb_slave=> select * from contact;
 cid |  name  |    address     |  phonenumber
-----+--------+----------------+----------------
   1 | Joe    | 1 Foo Street   | (592) 471-8271
   2 | Robert | 4 Bar Roard    | (515) 821-3831
```

Now, connect to the /contactdb/ database and insert a row:

```
% psql -U contact contactdb
contactdb=> begin; insert into contact (cid, name,
 address, phonenumber) values
 ((select nextval('contact_seq')), 'William',
 '81 Zot Street', '(918) 817-6381'); commit;
```

If you examine the content of the contact table of the

ASA COMPUTERS

Scale your performance, not your costs.

From mail servers to departmental databases, the ASA Servers based on the Intel® Xeon™ processor can speed up your applications. And deliver big-budget performance at a small-budget price.

Hardware Systems For The Open Source Community–Since 1989

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

6TB + in 5U—$12,050
Intel 7501, Dual Intel® Xeon™ 2.4GHz
512 MB DDR ECC RAM *Max: 8GB*
6TB + IDE Storage
Dual Gigabit LAN, CD+FD, VGA
Options: SATA Drives, Firewire, DVD+RW, CD+RW, 64 Bit OS Configurations, etc.

1U Dual Itanium IDE—$4,889
Dual Intel® Itanium® 2 1.4 Ghz
2 GB ECC DDR
1 of 4 x 40 GB HDD
Dual Gigabit LAN
Based on Supermicro 6113M-i

14" Deep Appliance Server—$865
Intel® Xeon™ 2.4 Ghz Processor
40 GB Hard Drive, One GigE
Options: CD, FD, 2nd HD, Your Logo on Bezel
Call for Low Cost Options.

1U Dual Xeon™ EM64T Superserver—$1,925
SuperMicro 6014H-82 Barebones
1 of 2 Intel® Xeon™ 2.8 GHz 800 FSB
1 GB DDR II-400 RAM *Max: 16GB*
36 GB 10K RPM SCSI *Max: 4 HS HDD*
CD+FD, Dual GigE, VGA, RAILS
Options: RAID, etc.

Your Custom Appliance Solution
Let us know your needs, we will get you a solution

ASA Colocation
$50 per month for 1U Rack - 20 GB/month

ASA Colocation Special
First month of colocation free.*

Storage Solutions
IDE, SCSI, Fiber RAID solutions
TB storage options
3Ware, Promise, Adaptec, JMR, Kingston/Storcase solutions

Clusters
Rackmount and Desktop nodes
HP, Intel, 3Com, Cisco switches
KVM or Cyclades Terminal Server
APC or Generic racks

All systems installed and tested with user's choice of Linux distribution (free). ASA Colocation—$50 per month

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

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

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

contactdb_slave database once more, you will notice that the row was replicated. Now, delete a row from the /contactdb/ database:

```
contactdb=> begin; delete from contact
 where cid = 2; commit;
```

Again, by examining the content of the contact table of the contactdb_slave database, you will notice that the row was removed from the slave node correctly.

Instead of comparing the information for contactdb and contactdb_slave manually, we easily can automate this process with a simple script, as shown in Listing 3. Such a script could be executed regularly to ensure that all nodes are in sync, notifying the administrator if that is no longer the case.

Listing 3. compare.sh

```
#!/bin/sh

CLUSTER=sql_cluster
DB1=contactdb
DB2=contactdb_slave
H1=localhost
H2=localhost
U=postgres

echo -n "Comparing the databases..."
psql -U $U -h $H1 $DB1 >dump.tmp.1.$$ <<_EOF_
         select 'contact'::text, cid, name, address,
           phonenumber from contact order by cid;
_EOF_
psql -U $U -h $H2 $DB2 >dump.tmp.2.$$ <<_EOF_
         select 'contact'::text, cid, name, address,
           phonenumber from contact order by cid;
_EOF_

if diff dump.tmp.1.$$ dump.tmp.2.$$ >dump.diff ; then
         echo -e "\nSuccess! Databases are identical."
         rm dump.diff
else
         echo -e "\nFAILED - see dump.diff."
fi
rm dump.tmp.?.$$
```

Although replicating a database on the same system isn't of much use, this example shows how easy it is to do. If you want to experiment with a replication system on nodes located on separate computers, you simply would modify the DB2, H1 and H2 environment variables from Listing 1 to 3. Normally, DB2 would be set to the same value as DB1, so an application always refers to the same database name. The host environment variables would need to be set to the fully qualified domain name of the two nodes. You also would need to make sure that the slon processes are running on both computers. Finally, it is good practice to synchronize the clocks of all nodes using ntpd or something similar.

Later, if you want to add more tables or sequences to the initial replication set, you can create a new set and use the merge set slonik command. Alternatively, you can use the set move table and set move sequence commands to split the set. Refer to the Slonik Command Summary for more information on this.

Failing Over

In case of a failure from the master node, due to an operating system crash or hardware problem, for example, Slony-I does not provide any automatic capability to promote a slave node to become a master. This is problematic because human intervention is required to promote a node, and applications demanding highly available database services should not depend on this. Luckily, plenty of solutions are available that can be combined with Slony-I to offer automatic failover capabilities. The Linux-HA Heartbeat program is one of them.

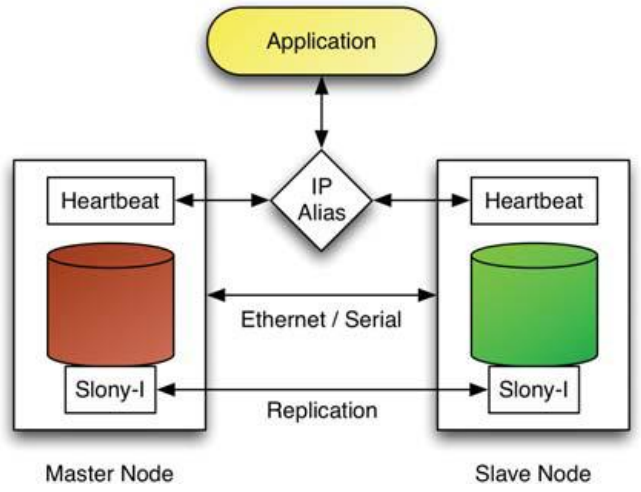


Figure 2. Heartbeat switches the IP alias to the slave node in case the master fails.

Consider Figure 2, which shows a master and slave node connected together using an Ethernet and serial link. In this configuration, the Heartbeat is used to monitor the node's availability through those two links. The application makes use of the database services by connecting to PostgreSQL through an IP alias, which is activated on the master node by the Heartbeat. If the Heartbeat detects that the master node has failed, it brings the IP alias up on the slave node and executes the slonik script to promote the slave as the new master.

The script is relatively simple. Listing 4 shows the content of the script that would be used to promote a slave node, running on slave.example.com, so it starts offering all the database services that master.example.com offered.

From Listing 4, the failover Slonik command is used to indicate that the node with id = 1, the node running on master.example.com, has failed, and that the node with id = 2 will take over all sets from the failed node. The second command, drop node, is used to remove the node with id = 1 from the replication system completely. Eventually, you might want to bring back the failed node in the cluster. To do this, you must configure it as a slave and let Slony-I replicate any missing information. Eventually, you can proceed with a switchback to the initial master node by locking the set (lock set), waiting for all events to complete (wait

Listing 4. promote.sh

```
#!/bin/bash

CLUSTER=sql_cluster
H1=master.example.com
H2=slave.example.com
U=postgres

DB1=contactdb
DB2=contactdb

su - postgres -c slonik <<_EOF_
cluster name = $CLUSTER;

node 1 admin conninfo = 'dbname=$DB1 host=$H1 user=$U';
node 2 admin conninfo = 'dbname=$DB2 host=$H2 user=$U';

failover (id = 1, backup node = 2);
drop node (id = 1, event node = 2);
```

for event), moving the set to a new origin (move set) and waiting for a confirmation that the last command has completed. Refer to the Slonik Command Summary for more information on those commands.

Conclusion

Replicating databases using Slony-I is relatively simple. Combined with the Linux-HA Heartbeat, this allows you to offer high availability of your database services. Although the combination of Slony-I and Linux HA-Heartbeat is an attractive solution, it is important to note that this is not a substitute for good hardware for your database servers.

Even with its small limitations, like not being able to propagate schema changes or replicate large objects, Slony-I is a great alternative to both rserv and ERServer and is now, in fact, the preferred solution for replicating PostgreSQL databases. Slony-II even supports synchronous multimaster replication and is already on the design table.

To conclude, I would like to thank Jan Wieck, the author of Slony-I, for reviewing this article.

Resources for this article: www.linuxjournal.com/article/8202.

Ludovic Marcotte (ludovic@sophos.ca) holds a Bachelor's degree in Computer Science from the University of Montréal. He is currently a software architect for Inverse, Inc., an IT consulting company located in downtown Montréal.

Hurricane Electric Internet Services...Speed and Reliability That Sets You Apart from the Competition!

Flat Rate Gigabit Ethernet

1,000 Mbps of IP

$13,000/month*

Full 100 Mbps Port

Full Duplex

$2,000/month

Colocation Full Cabinet

Holds up to 42 1U servers

$400/month

Order Today!

email sales@he.net or call 510.580.4190

* Available at PAIX in Palo Alto, CA; Equinix in Ashburn, VA; Equinix in Chicago, IL; Equinix in Dallas, TX; Equinix in Los Angeles, CA; Equinix in San Jose, CA; Telehouse in New York, NY; Telehouse in London, UK; NIKHEF in Amsterdam, NL; Hurricane in Fremont, CA and Hurricane in San Jose, CA

Modeling the Brain with NCS and Brainlab

Beowulf Linux clusters and Python toolkits team up to help scientists understand the human brain.

BY RICH DREWES

Computer scientists have been studying artificial neural networks (ANNs) since the 1950s. Although ANNs were inspired by real biological networks like those in your brain, typical ANNs do not model a number of aspects of biology that may turn out to be important. Real neurons, for example, communicate by sending out little spikes of voltage called action potentials (APs). ANNs, however, do not model the timing of these individual APs. Instead, ANNs typically assume that APs are repetitive, and they model only the rate of that repetition. For a while, most researchers believed that modeling the spike rate was enough to capture the interesting behavior of the network. But what if some of the computational power of a biological neural network was derived from the precise timing of the individual APs? Regular ANNs could never model such a possibility.

NCS: the NeoCortical Simulator

In 1999, the thought that ANNs were overlooking the reality of individual APs convinced Phil Goodman at the University of Nevada, Reno, to change his focus from ANNs to more realistic spiking neural network models. He started by looking for a program that would allow him to conduct experiments on large networks of spiking neurons. At the time, a couple of excellent open-source research software packages existed that were capable of simulating a few spiking neurons realistically; GENESIS and NEURON were two of the most popular. But these programs were not designed to work with the networks of thousands of spiking neurons that he was envisioning. Goodman believed that with low-cost Linux clustering technology, it should be possible to construct a parallel program that was realistic enough to model the spiking and cellular membrane channel behavior of neurons, while also being efficient enough to allow the construction of large networks of these neurons for study. Goodman launched the NeoCortical Simulator (NCS) Project to create such a program. Starting with a prototype program that Goodman wrote in the proprietary MATLAB environment, a student working with computer science Professor Sushil Louis wrote the first parallel version of NCS in C using the MPI parallel library package.

When I joined the research group in 2002, NCS already was undergoing a major rewrite by another student, James Frye, who was working with CS Professor Frederick C. Harris, Jr. This time, the goal was to take the system from prototype to streamlined and reliable production software system. I helped with this effort, implementing a number of optimizations that greatly improved performance.

I also set up the first version control for the NCS source code, using the then-new open-source Subversion system. At the time, Subversion still was an alpha project. Nevertheless, I was sold on several features of the system, including the automatic bundling of an entire set of files into a single release. After working with Subversion a bit, the old workhorse CVS seemed cumbersome in comparison. Subversion was evolving quickly then. More than once after a system software upgrade, though, I had to spend hours trying to rebuild a Subversion executable with a certain combination of component library versions that would restore access to our version history. The Subversion user mailing list always was helpful during these recovery efforts. Eager to take advantage of the new features, I willingly paid the price for choosing alpha software. Fortunately, that trade-off is no longer necessary. Subversion now is stable and flexible, and I would not hesitate to choose it for any new project.

As the NCS software matured, our cluster expanded, thanks to several grants from the US Office of Naval Research. The initial Beowulf cluster of 30 dual-processor Pentium III machines grew with the addition of 34 dual-processor Pentium 4s. It grew again recently with the addition of 40 dual-processor Opterons. Linux has been the OS for the cluster from the start, running the Rocks cluster Linux release. The compute nodes are equipped with a full 4GB of system memory to hold the large number of synapse structures in the brain models. Memory capacity was a major motivation for moving to the 64-bit Opterons. Administrative network traffic moves on a 100MB and, later, 1GB Ethernet connection, while a specialized low-latency Myrinet network efficiently passes the millions of AP spike messages that occur in a typical neural network simulation.

Designing Brain Models

With NCS now capable of simulating networks of thousands of spiking neurons and many millions of synapses, students began to use it for actual research. NCS could be quite hard to use effectively in practice, however, as I discovered when I began my own first large-scale simulation experiments. Much of the difficulty in using NCS stemmed from the fact that NCS takes

a plain-text file as input. This input file defines the characteristics of the neural network, including neuron and dendrite compartments, synapses, ion channels and more. For a large neural network model, this text file often grows to thousands or even hundreds of thousands of lines.

Although this plain-text file approach allows a great deal of flexibility in model definition, it quickly becomes apparent to anyone doing serious work with NCS that it is not practical to create network models by directly editing the input file in a text editor. If the model contains more than a handful of neural structures, hand-editing is tedious and prone to error. So every student eventually ends up implementing some sort of special-purpose macro processor to help construct the input file by repeatedly emitting text chunks with variable substitutions based on a loop or other control structure. Several of these preprocessors were built in the proprietary MATLAB language, because MATLAB also is useful for the post-simulation data analysis and is a popular tool in our lab. Each of these macro processors was implemented hurriedly with one specific network model in mind. No solution was general enough to be used by the next student, therefore, causing a great deal of redundant effort.

I searched for a more general solution, both for my own work and to prevent future students from facing these familiar hurdles as they started to use NCS for large experiments. No templated preprocessing approach seemed up to the task. After a bit of experimentation, I concluded that the best way of specifying a brain model was directly as a program—not as a templated text file that would be parsed by a program, but actually as a program itself.

To understand the problem, consider that our brain models often contain hundreds or thousands of structures called cortical columns, each made up of a hundred or more neurons. These columns have complex, often variable internal structures, and these columns themselves are interconnected by synapses in complex ways. We might want to adjust the patterns of some or all of these connections from run to run. For example, we might want to connect a column to all neighbor columns that lie within a certain distance range, with a certain probability that is a function of the distance. Even this relatively simple connection pattern can't be expressed conveniently in the NCS input file, which permits only a plain list of objects and connections.

But, by storing the brain model itself as a small script that constructs the connections, we could have a model in only a few lines of code instead of thousands of lines of text. This code easily could be modified later for variations of the experiment. All the powerful looping and control constructs, math capabilities and even object orientation of the scripting language could be available directly to the brain modeler. Behind the scenes, the script automatically could convert the script representation of the model into the NCS text input file for actual simulation. No brain modeler ever would be bound by a restrictive parsed template structure again. I gave the generalized script-based modeling environment that I planned to develop the name Brainlab and set to work picking a suitable scripting language for the project.

Brainlab

My first thought for a scripting language was MATLAB, given its prominence in our lab. But repeated licensing server failures during critical periods had soured me on MATLAB. I considered Octave, an excellent open-source MATLAB work-alike that employed the same powerful vector processing approach. I generally liked what I saw and even ported a few MATLAB applications to work in Octave in a pinch. I was pleased to find that the conversions were relatively painless, complicated only by MATLAB's loose language specification. But I found Octave's syntax awkward, which was no surprise because it largely was inherited from MATLAB. My previous Tcl/Tk experiences had been positive, but there didn't seem to be much of a scientific community using it. I had done a few projects in Perl over the years, but I found it hard to read and easy to forget.

Then I started working with Python on a few small projects. Python's clean syntax, powerful and well-designed object-oriented capabilities and large user community with extensive libraries and scientific toolkits made it a joy to use. Reading Python code was so easy and natural that I could leave a project for a few months and pick it up again, with barely any delay figuring out where I was when I left off. So I created the first version of Brainlab using Python.

In Brainlab, a brain model starts as a Python object of the class BRAIN:

```
from brainlab import *
brain=BRAIN()
```

This brain object initially contains a default library of cell types, synapse types, ion channel types and other types of objects used to build brain models. For example, the built-in ion channel types are stored in a field in the BRAIN class named chantypes. This field actually is a Python dictionary indexed by the name of the channel. It can be viewed simply by printing out the corresponding Python dictionary:

```
print brain.chantypes
```

A new channel type named ahp-3, based on the standard type named ahp-2, could be created, modified and then viewed like this:

```
nc=brain.Copy(brain.chantypes, 'ahp-2', 'ahp-3')
nc.parms['STRENGTH']="0.4 0.04"
print brain.chantypes['ahp-3']
```

To build a real network, the brain must contain some instances of these structures and not only type profiles. In NCS, every cell belongs to a structure called a cortical column. We can create an instance of a simple column and add it to our brain object like this:

```
c1=brain.Standard1CellColumn()
brain.AddColumn(c1)
```

This column object comes with a set of default ion channel instances and other structures that we easily can adjust if necessary. Most often we have a group of columns that we want to create and interconnect. The following example creates a two-dimensional grid of columns in a loop and then connects the

columns randomly:

```
cols={}
size=10
# create the columns and store them in cols{}
for i in range(size):
    for j in range(size):
        c=brain.Standard1CellColumn()
        brain.AddColumn(c)
        cols[i,j]=c
# now connect each column to another random column
# (using a default synapse)
for i in range(size):
    for j in range(size):
        ti=randint(0, size-1)
        tj=randint(0, size-1)
        fc=cols[i,j]; tc=cols[ti,tj]
        brain.AddConnect(fc, tc)
```

Our brain won't do much unless it gets some stimulus. Therefore, we can define a set of randomly spaced stimulus spikes in a Python list and apply it to the first row of our column grid like this:

```
t=0.0
stim=[]
for s in range(20):
    t+=random()*10.0
    stims.append(t)
for i in range(size):
    brain.AddStim(stim, cols[i,0])
```

Simulating the Models

So far, our brain model exists only as a Python object. In order to run it in an NCS simulation, we have to convert it to the text input file that NCS demands. Brainlab takes care of this conversion; simply printing the brain object creates the corresponding NCS input text for that model. The command `print brain` prints more than 3,000 lines of NCS input file text, even for the relatively simple example shown here. More complicated models result in even longer input files for NCS, but the program version of the model remains quite compact.

By changing only a few parameters in the script, we can create a radically different text NCS input file. The experimenter can save this text to a file and then invoke the NCS simulator on that file from the command line. Better yet, he or she can simulate the model directly within the Brainlab environment without even bothering to look at the intermediate text, like this: `brain.Run(nprocs=16)`.

The Run() method invokes the brain model on the Beowulf cluster using the indicated number of processor nodes. Most often, an experiment is not simply a single simulation of an individual brain model. Real experiments almost always consist of dozens or hundreds of simulation runs of related brain models, with slightly different parameters or stimuli for each run. This is where Brainlab really shines: creating a model, simulating it, adjusting the model and then simulating it again and again, all in one integrated environment. If we wanted to run an experiment ten times, varying the synapse conduction strength with each run and with a different job number each run so that we could examine all the reports later, we might do something like this:

```
for r in range(10):  # r is run number
    s=brain.syntypes['C.strong']
    s.parms['MAX_CONDUCT']=.01+.005*r
    brain.parms['JOB']='testbrain%d'%r
    brain.Run(nprocs=16)
```

Toolkits for Data Analysis and Search

The numarray extension package for Python provides for efficient manipulation and statistical analysis of the large NCS datasets that result from a simulation. For graphs and charts of results, the excellent matplotlib package produces publication-quality output through a simple yet powerful MATLAB-like interface (Figure 1). Brainlab also provides a number of convenient interfaces for these packages, making it easier to do the operations commonly needed for neuroscience research. Brainlab also provides interactive examination of

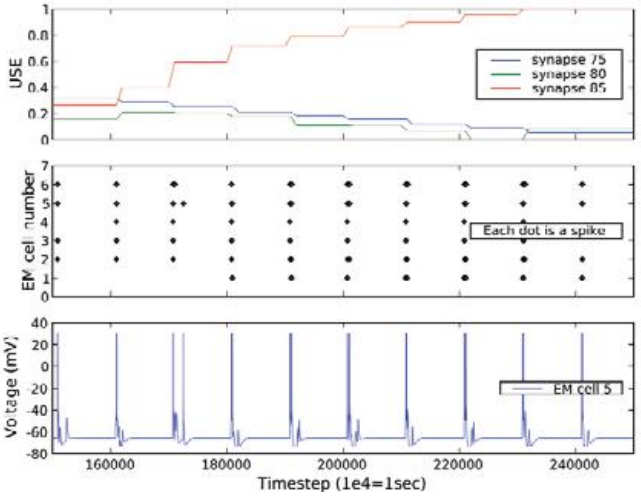


Figure 1. Creating publication-ready charts is easy using the matplotlib package.

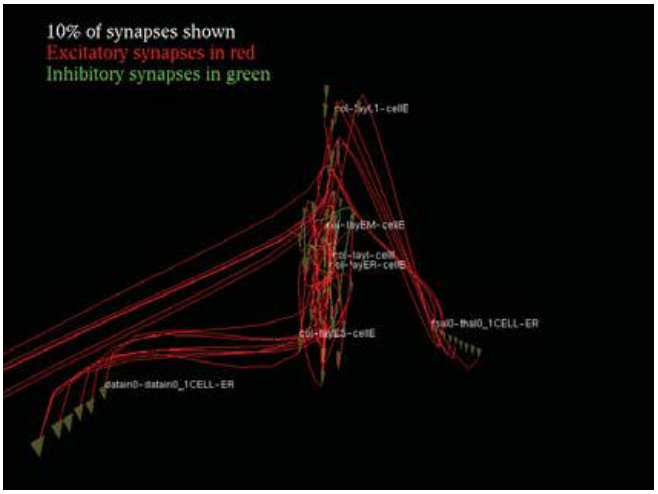


Figure 2. For interactive experimentation with 3-D views, Brainlab offers an OpenGL interface.

3-D views of the network models using the Python OpenGL binding (Figure 2).

Quite often, some experimentation with a number of network parameters is required in order to find a balanced brain model. For example, if a synaptic strength is too high or too low, the model may not function realistically. We have seen how Brainlab could help a modeler do a search for a good model by repeatedly running the same model with a varying parameter. But an even more powerful technique than that simple search is to use another inspiration from biology, evolution, to do a genetic search on the values of a whole set of parameters. I have used Brainlab to do this sort of multiparameter search with a genetic algorithm (GA) module of my own design and also with the standard GA module of the Scientific Python package, SciPy.

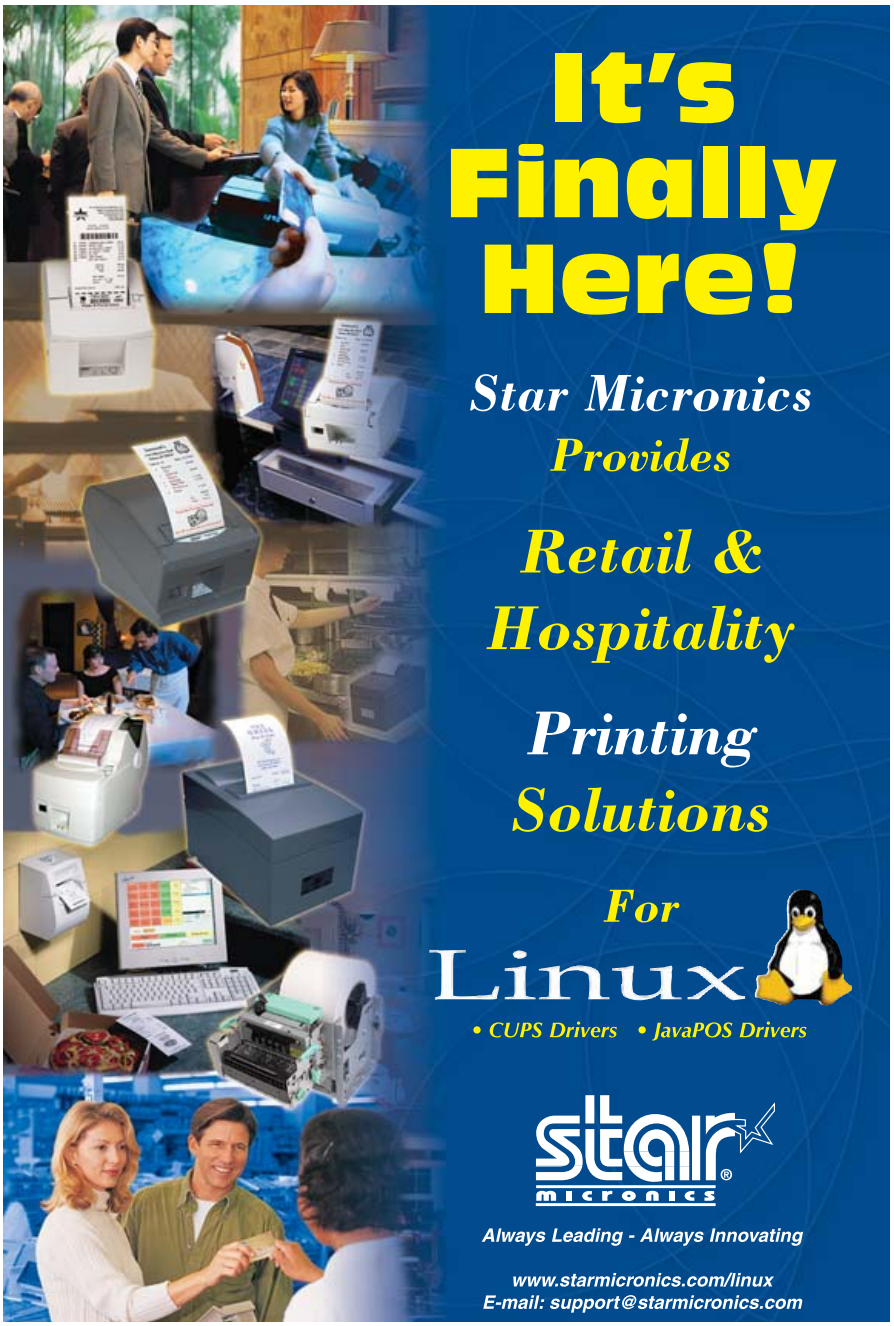
Conclusion

Brainlab has made my complex experiments practical, perhaps even possible. At this point I can't imagine doing them any other way. In fact, if NCS were to be reimplemented from scratch, I would suggest a significant design change: the elimination of the intermediate NCS input text file format. This file format is just complex enough to require a parser and the associated implementation complexity, documentation burden and slowdown in the loading of brain models. At the same time, it is not nearly expressive enough to be usable directly for any but the simplest brain models. Instead, a scripting environment such as Python/Brainlab could be integrated directly into NCS, and the scripts could create structures in memory that are accessed directly from the NCS simulation engine. The resulting system would be extremely powerful and efficient, and the overall documentation burden would be reduced. This general approach should be applicable to many different problems in other areas of model building research.

This summer, NCS is going to be installed on a new 4,000-processor IBM BlueGene cluster at our sister lab, the Laboratory of Neural Microcircuitry of the Brain Mind Institute at the EPFL in Switzerland, in collaboration with lab director Henry Markram. Early tests show that we can achieve a nearly linear speedup in NCS performance with increasing cluster size, due to efficient programming and the highly parallel nature of synaptic connections in the brain. We hope that other researchers around the world will find NCS and Brainlab useful in the effort to model and understand the human brain.

Resources for this article: www.linuxjournal.com/article/8203.

Rich Drewes (drewes@interstice.com) is a PhD candidate in Biomedical Engineering at the University of Nevada, Reno.



Squid-Based Traffic Control and Management System

When Web traffic became a major use of the organization's network, this university put in a control system to track and limit access, using the open-source Squid caching system. BY TAGIR K. BAKIROV AND VLADIMIR G. KOZLOV

Internet access is one of the major and most demanded services in the computer network of any organization. Olifer and Olifer, in *Computer Networks: Principles, Technologies and Protocols* write that during the past 10–15 years, the 80/20 split between internal and outgoing traffic has turned over, and the split is now 80% outgoing (see the on-line Resources). The speed of access, the number of services and the volume of available content increase permanently. And the actuality of the Internet user access control task grows up. This problem is quite old, but now some of its aspects are changing. In this article, we consider the variants of its modern solution in the example of the computer network at Bashkir State Pedagogical University (BSPU).

First, we proposed some initial requirements for the Internet access control and management system:

- User account support and management.
- User traffic accounting and control.
- Three types of user traffic limitation: per month, per week and per day.
- Support for mobile users—people who use different computers each time they access the Internet, such as students.
- Daily and weekly statistics and Web and e-mail system condition reports.
- Web-based statistics and system management.

Apparently, these requirements do not specify the system implementation stage in any way and hence do not limit our "fantasy" in this aspect. Therefore, we have done a general consideration of the problem and how to solve it. In the rest of this article, we discuss the ideas and reasoning that led us to our final decision.

Common Analysis of the Problem

Let us revisit the Internet access process itself, with the example of the most popular World Wide Web (WWW) service:

1. The user runs the browser and enters the required URL.
2. The browser establishes the connection either directly with the WWW server via the gateway, which makes the network address translation or other network packet manipulations, or with the proxy server, which analyzes the client request thoroughly and looks through its cache for the required information. If there is no such information or if it is outdated, the proxy server connects with the WWW server in its own name.
3. The obtained information is returned to the client.
4. The browser ends the connection or enters the keep-alive state.

Figure 1 shows the scheme of Internet user access organization.

The main elements of the scheme are the user; client software, including browser and operating system; workstation and other client hardware; network equipment; and the gateway (or proxy server). Other user authorization servers, such as Microsoft Windows domain controllers, OpenLDAP or NIS also may exist in the network.

As Figure 1 shows, the relation between the users and the workstations can be of the one-to-one or the many-to-many type. For instance, members of the university staff are mostly equipped with their own computers.

The main aspects of the problem are user traffic accounting, user authentication, user access control and management and reporting.

These aspects are quite independent of one another and each of them has several ways of implementation. The functions of authentication, traffic accounting and access control may be assigned to any element of the scheme above. And, the

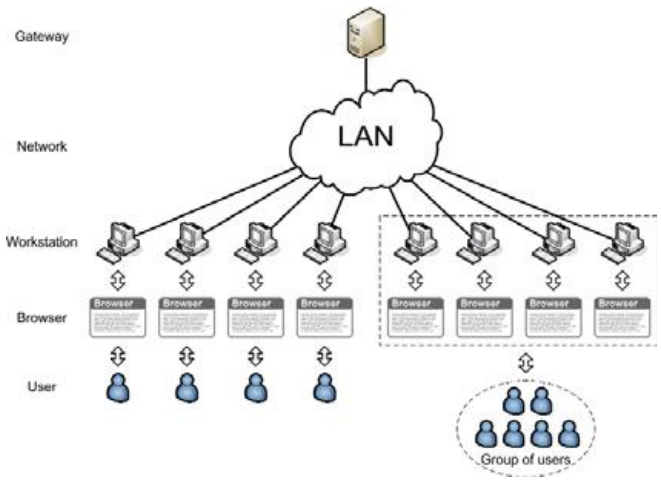


Figure 1. Internet User Access Organization

best solution will concentrate all of the functions in the single module or in the single access scheme element.

Access control can be implemented on the client side or on the server side. Client-side access control requires using the special client software, which also can authenticate the users. And, there are two ways of server-side access control implementation: firewall and proxy server. Firewall access control has the problem of user authentication. The network packets include only the IP addresses, which are not bound to user names. In the case of using a firewall, this problem has two solutions: use of VPN, which has its own user authentication mechanism and dynamic user-to-IP assignment control. This is possible with some external tools.

The simpler solution, however, is the use of the proxy server, which supports user authentication using the browser. There are three methods of browser authentication:

- Basic authentication—a simple and widely distributed scheme, which is supported by the majority of Internet browsers and proxy servers. Its main disadvantage is that the user password is sent over the network with no encryption.

- Digest authentication is a more reliable scheme, which uses password hashes for security. Its main imperfection is the lack of special software support.

- NTLM authentication is specific for the Microsoft product network infrastructure. Nevertheless, this authentication scheme is acceptable and, furthermore, desirable in many computer networks, including Windows workstations, which are prevalent in Russia as far as we know. The main advantage here is the possibility of the integration of the proxy authentication scheme with Windows and Samba domain controllers.

The task analysis and some of the ideas above led us to the development of two systems:

1. VPN using PPTP based on the firewall internal features. Historically, the VPN server used FreeBSD, hence, we used the ipfw firewall interface and mpd ported application as a PPTP server. Traffic control is made using the free, distributable NetAMS system.

2. Squid-based Internet user access control and management system.

The first system was developed by Vladimir Kozlov and is used to connect the university staff members, who use dedicated computers for Internet access. Its main disadvantage is the requirement of a client-side VPN setup. This is a considerable obstacle in the case when the computer network is distributed and the users are not familiar enough with computers.

The second system was developed by Tagir Bakirov and is used to connect the majority of university users, who have no constant computer for Internet access. The complexity of the development was the main drawback of this solution. Next, we discuss the implementation of the second solution in detail.

Squid-Based Internet User Access Control and Management System

Before we start, we should mention that the file paths here are always relative to the Squid source base catalog, which, in our case, is /usr/local/src/squid-2.5STABLE7/. The detailed information of getting, compiling and using Squid can be obtained from the Squid site.

Let us now consider some characteristics of Squid, taken from the *Squid Programming Guide*.

Squid is a single-process proxy server. Every client HTTP request is handled by the main process. Its execution progresses as a sequence of callback functions. The callback function is executed when I/O is ready to occur or some other event has happened. As a callback function completes, it registers the next callback function for the subsequent I/O.

At the core of Squid are the select(2) or the poll(2) system calls, which work by waiting for I/O events on a set of file descriptors. Squid uses them to process I/O on all open file descriptors. comm_select() is the function that issues the select() system call. It scans the entire fd_table[] array looking for handler functions. For each ready descriptor, the handler is called. Handler functions are registered with the commSetSelect() function. The close handlers normally are called from comm_close(). The job of the close handlers is to deallocate data structures associated with the file descriptor. For this reason, comm_close() normally must be the last function in a sequence.

An interesting Squid feature is the client per-IP address database support. The corresponding code is in the file src/client_db.c. The main idea is the hash-indexed table, client_table, consisting of the pointers to ClientInfo structures. These structures contain different information on the HTTP client and ICCP proxy server connections, for example, the request, traffic and time counters. The following is the respective code from the file src/structs.h:

```
struct _ClientInfo {
    /* must be first */
    hash_link hash;
    struct in_addr addr;
    struct {
```

```
        int result_hist[LOG_TYPE_MAX];
        int n_requests;
        kb_t kbytes_in;
        kb_t kbytes_out;
        kb_t hit_kbytes_out;
    } Http, Icp;
    struct {
        time_t time;
        int n_req;
        int n_denied;
    } cutoff;
    /* number of current established connections */
    int n_established;
    time_t last_seen;
};
```

Here are some important global and local functions for managing the client table:

- clientdbInit()—global function that initializes the client table.
- clientdbUpdate()—global function that updates the record in the table or adds a new record when needed.
- clientdbFreeMemory()—global function that deletes the table and releases the allocated memory.
- clientdbAdd()—local function that is called by the function clientdbUpdate() and adds the record into the table and schedules the garbage records collecting procedure.
- clientdbFreeItem()—local function that is called by the function clientdbFreeMemory() and removes the single record from the table.
- clientdbSheduledGC(), clientdbGC() and clientdbStartGC()—local functions that implement the garbage records collection procedure.

By parallelizing the requirements to the developed system and the possibilities of the existing client database, we can say that some key basic features already are implemented, except the client per-user name indexing. The other significant shortcoming of the existing client statistic database is that the information is refreshed after the client already has received the entire requested content.

In our development, we implemented another parallel and independent client per-user database using the code from the src/client_db.c file with some modifications. User statistics are kept in structure ClientInfo_sb. The following is the corresponding code from the file src/structs.h:

```
#ifdef SB_INCLUDE
#define SB_CLIENT_NAME_MAX_LENGTH 16
struct _ClientInfo_sb {
    /* must be the first */
    hash_link hash;
    char *name;
    unsigned int GID;
    struct {
        long value;
        char type;
        long cur;
        time_t lu;
    } lmt;
    /* HTTP Request Counter */
    int Counter;
};
#endif
```

The client database is managed by the following global and local functions, quite similar to those listed previously:

- clientdbInit_sb()—global function that initializes the client table.
- clientdbUpdate_sb()—global function that updates the record in the table, disconnects the client when the limit is exceeded or adds the new record when needed by calling the function clientdbAdd_sb().
- clientdbEstablished_sb()—global function that counts the number of client requests and periodically flushes the appropriate record into the file, disconnects the client when the limit is exceeded and adds the new record when needed by calling the function clientdbAdd_sb().
- clientdbFreeMemory_sb()—global function that deletes the table and releases the allocated memory.
- clientdbAdd_sb()—local function that is called by the function clientdbUpdate_sb() and adds the record into the table and schedules the garbage records collecting procedure.
- clientdbFlushItem_sb()—local function that is called by the functions clientdbEstablished_sb() and clientdbFreeItem_sb() and flushes the particular record into the file.
- clientdbFreeItem_sb()—local function that is called by the function clientdbFreeMemory_sb() and removes the single record from the table.
- clientdbSheduledGC_sb(), clientdbGC_sb() and clientdbStartGC_sb()—local functions that implement the garbage records collecting procedure.

The client database initialization and release are implemented similarly to the original table in the file src/main.c. The main peculiarity of our code is the calls of the functions clientdbUpdate_sb() and clientdbEstablished_sb() in the client-side routines in the file src/client_side.c:

- call of the function clientdbUpdate_sb() from the auxiliary function clientWriteComplete(), which is responsible for sending the portions of data to the client.
- call of the function clientdbEstablished_sb() from the function clientReadRequest(), which processes the client request.

Listing 1 shows the corresponding fragments of the functions clientWriteComplete() and clientReadRequest() from the

www.LinuxJournal.com/ArchiveCD

The 1994-2003 Archive CD,
back issues, and more!

Listing 1. Fragments of the Functions clientWriteComplete() and clientReadRequest() from the src/client_side.c File

```
static void
clientWriteComplete(int fd,
                    char *bufnotused,
                    size_t size,
                    int errflag,
                    void *data)
{
    clientHttpRequest *http = data;

...

    if (size > 0)
    {
        kb_incr(&statCounter.client_http.kbytes_out,
                size);
/*-Here comes the SB section----------------------*/
#ifdef SB_INCLUDE
    if (http->request->auth_user_request)
    {
         if ( authenticateUserRequestUsername(
              http->request->auth_user_request) )
             if (!clientdbUpdate_sb(
                  authenticateUserRequestUsername(
                      http->request->auth_user_request),
                      size) )
             {
                 comm_close(fd);
                 return;
             }
    }
#endif
/*-----------------------------------------------*/
    if (isTcpHit(http->log_type))
        kb_incr(
            &statCounter.client_http.hit_kbytes_out,
            size);
    }

...

}

...

static void
clientReadRequest(int fd, void *data)
{
    ConnStateData *conn = data;
    int parser_return_code = 0;
    request_t *request = NULL;
    int size;
    void *p;
    method_t method;
    clientHttpRequest *http = NULL;
    clientHttpRequest **H = NULL;
    char *prefix = NULL;
    ErrorState *err = NULL;
    fde *F = &fd_table[fd];
    int len = conn->in.size - conn->in.offset - 1;

...

    /* Process request body if any */
    if (conn->in.offset > 0 &&
        conn->body.callback != NULL)
    {
        clientProcessBody(conn);
    }
    /* Process next request */
    while (conn->in.offset > 0 &&
           conn->body.size_left == 0)
    {
        int nrequests;
        size_t req_line_sz;

...

        /* Process request */
        http = parseHttpRequest(conn,
            &method,
            &parser_return_code,
            &prefix,
            &req_line_sz);
        if (!http)
            safe_free(prefix);
        if (http) {

...

            if (request->method == METHOD_CONNECT)
                {
                /* Stop reading requests... */
                    commSetSelect(fd,
                    COMM_SELECT_READ,
                    NULL,
                    NULL,
                    0);
                    clientAccessCheck(http);
/*-Here comes the SB section----------------------*/
#ifdef SB_INCLUDE
                if(http->request->auth_user_request)
                {
                    if (
                    authenticateUserRequestUsername(
                    http->request->auth_user_request
                    )!=NULL)
```

```
                    {
                        if(!clientdbCount_sb(
                           authenticateUserRequestUsername(
                           http->request->
                                auth_user_request)))
                            {
                                comm_close(fd);
                                return;
                            }
                        }
                    }
#endif
/*-----------------------------------------------*/
                break;
                } else {
                clientAccessCheck(http);
/*-Here comes the SB section----------------------*/

#ifdef SB_INCLUDE
                if(http->request->auth_user_request)
                {
                    if (
                        authenticateUserRequestUsername(
                        http->request->auth_user_request
                        )!=NULL)
                    {
                        if(!clientdbCount_sb(
                        authenticateUserRequestUsername(
                        http->request->auth_user_request)))
                        {
                            comm_close(fd);
                            return;
                        }
                    }
                }
#endif
/*-----------------------------------------------*/

/* while offset > 0 && body.size_left == 0 */
                    continue;
                }
            } else if (parser_return_code == 0) {

...

/* while offset > 0 && conn->body.size_left == 0 */
        }

...

}
```

file src/client_side.c.

Thus, the mechanism is quite simple. Figure 2 shows the simple client request processing diagram from the point of view of our system. Each client request contains the user authentication information, including the user name. The function clientdbUpdate_sb() searches for the ClientInfo_sb record, which corresponds to the user name obtained from the request. In the case of the absence of such a record, it adds the new ClientInfo_sb record using the information from the authority files. If users exceed their limit, they are disconnected immediately with the function comm_close(). The call of the function clientdbEstablished_sb() is also used to control the number of client requests and to save current user information into the authority files every SB_MAX_COUNT requests. The authority files are called passwd and group analogously to the UNIX files. The passwd file contains the user information, and the group file contains the user group information. Here are the descriptive samples:

```
`passwd':
#<name>:<full name>:<group id>:
#<current limit value>:<last limit update time>

tagir:Tagir Bakirov:1:6567561:12346237467

`group':
#<name>:<full name>:<group id>:
#<group limit value>:<group limit type>

users:BSPU users:1:10000000:D
```

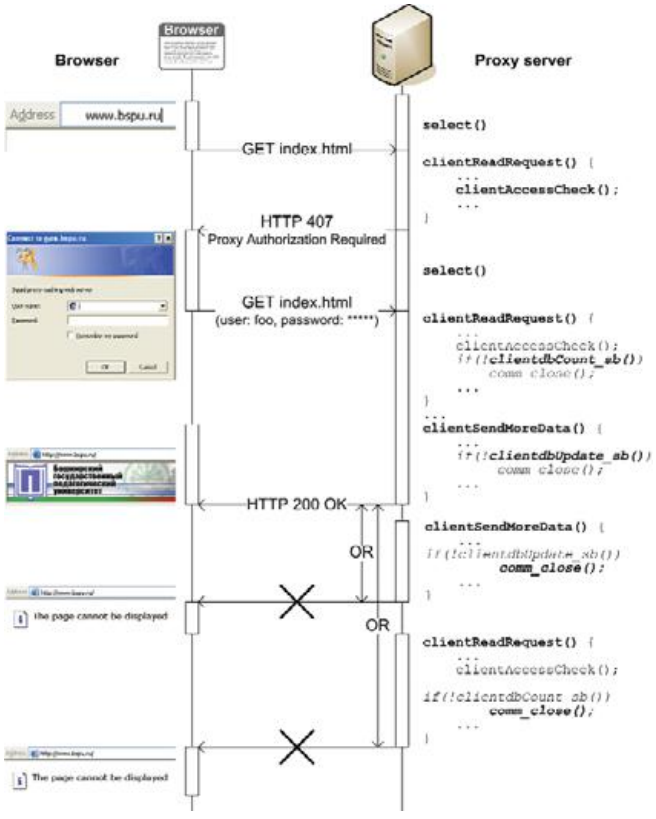


Figure 2. Simple Client Request Processing Diagram

There are three types of limit: D (daily), W (weekly) and M (monthly). The passwd and group filenames and paths can be set in the Squid configuration file squid.conf. This was implemented by modifying the structure of the squid.conf template file and the structure of the Squid configuration structure.

Here are the other slight changes in the Squid source code:

- Global functions definition in the file src/protos.h.
- ClientInfo_sb structure type definition in the file src/typedefs.h.
- ClientInfo_sb structure identifier declaration in the structure list in the file src/enums.h.
- ClientInfo_sb structure initialization in the memory allocation procedure memInit() in the file src/mem.c.

All of these changes are made analogously to the code, maintaining the original client per-IP database. We hope everything was done right.

Looking through our modifications, you may have noticed that all the code is put into the conditional compilation blocks (#ifdef SB_INCLUDE ... #endif). The variable SB_INCLUDE is declared when the parameter --enable-sbclientdb is included into the command line of the Squid configure script. This was made by recompiling the configure.in script with autoconf after putting in some slight modifications.

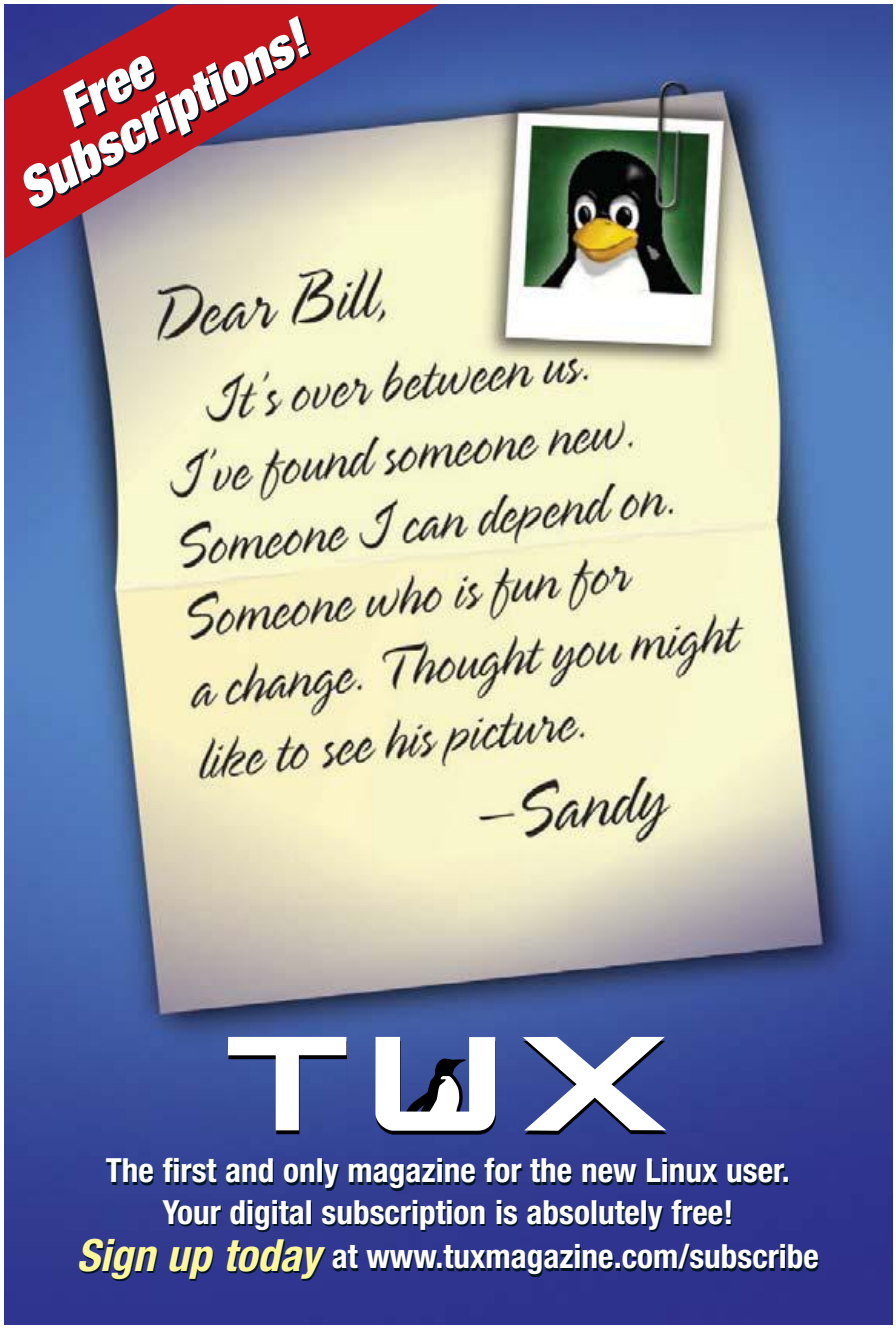
Conclusion

In this article, we considered the state of the art in the Internet access control problem. We proposed several methods for its solution and considered the variant based on the Squid proxy server, which has been implemented in the LAN of BSPU. Our solution is not the panacea and possibly has several drawbacks, but it is rather simple, flexible and absolutely free.

We also should say that our Web programmer, Elmir Mirdiev, is now finishing the implementation of a small PHP-based Web site designed for system management and user statistics reporting. The user-detailed statistics are generated from the Squid logs using the Sarg system.

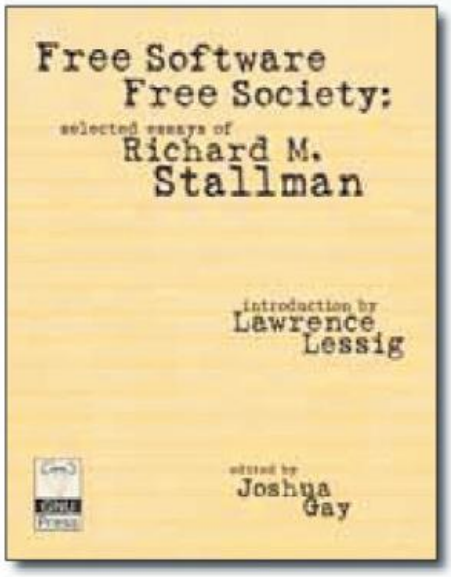
Other information can be obtained from the source code of the system. You can get the whole modified source code of Squid version 2.5STABLE7 tarball on our site or only the patch file. We will be glad to answer your questions by e-mail.

Resources for this article: www.linuxjournal.com/article/8205.



Tagir K. Bakirov (batk@mail.ru) is a system administrator at BSPU and a first-year postgraduate student of Ufa State Aviation Technical University. His main interests are information security, multi-agent systems and other IT. His hobbies include sporting activities, books, music and foreign languages.

Vladimir G. Kozlov (admin@bspu.ru), doctor of pedagogical science, assistant professor, is the senior system administrator and lecturer of several IT disciplines at BSPU. His main interests are *NIX networking, IT and electronics. His hobbies include ham radio (UA9WBZ), family and sports.



Free Software Free Society:

selected essays of Richard M. Stallman

Introduction by
Lawrence Lessig
Edited by Joshua Gay

224 Pages
Cover Price: $24.95
ISBN 1-882114-98-1
Hardcover with Dust Jacket

Richard Stallman is the prophet of the free software movement. He understood the dangers of software patents years ago. Now that this has become a crucial issue in the world, buy this book and read what he said.

Tim Berners-Lee, inventor of the World Wide Web

The intersection of ethics, law, business, and computer software is the subject of this collection of essays and speeches by MacArthur Foundation Grant winner Richard M. Stallman. This collection includes historical writings such as *The GNU Manifesto*, which defined and launched the activist Free Software Movement, along with new writings on hot topics in copyright, patent law, and the controversial issue of "trusted computing."

Stallman takes a critical look at common abuses of copyright law and patents when applied to computer software programs, and how these abuses damage our entire society and remove our existing freedoms. He also discusses the social aspects of software and how free software can create community and social justice. He argues that for creativity to flourish, software must be free of inappropriate and overly-broad legal constraints.

Over the past twenty years his arguments and actions have changed the course of software history; this new book is sure to impact the future of software and legal policies in the years to come.

About the Author:

Creator of the Free Software Movement—a progressive worldwide movement to make software code freely available so as to improve its development and accessibility—Richard M. Stallman is an internationally recognized computer scientist, author, and speaker. He has received a host of scientific awards, ranging from a 1990 MacArthur Foundation "Genius Grant" Fellowship in Computer Science, to his election to the American National Academy of Engineering in 2002.

By his hugely successful efforts to establish the idea of "Free Software", Stallman has made a massive contribution to the human condition. His contribution combines elements that have technical, social, political, and economic consequences.

Gerald Jay Sussman
Matsushita Professor of Electrical Engineering, MIT

For the first time this book collects the writing and lectures of Richard Stallman in a manner that will make their subtlety and power clear. The essays span a wide range, from copyright to the history of the free software movement. They include many arguments not well known, and...will serve as a resource for those who seek to understand the thought of this most powerful man . . . .

Lawrence Lessig
Stanford University Law School professor
and expert on cyberlaw

Richard is the leading force of the free software movement. This book is very important to spread the key concepts of free software world-wide, so everyone can understand it. Free software gives people freedom to use their creativity.

Masayuki Ida
Professor, Graduate School of International Management,
Aoyama Gakuin University

Order directly from our website and use code LQIO to receive 30% off this book!

FREE SOFTWARE FOUNDATION
59 Temple Place Suite 330
Boston' MA 02111-1307
tel: 617-542-5942
fax: 617-542-2652

order.fsf.org
press@fsf.org

Constructing Red Hat Enterprise Linux 4

How do you put together a stable Linux distribution better and faster? Get adamant about pushing your changes upstream, maintain a community test distribution and bring developers from partner companies on-site. BY TIM BURKE

Wow, time sure flies when you are having fun! Seems like only yesterday I was sitting here writing "Constructing Red Hat Enterprise Linux v.3" (see the on-line Resources). Hard to believe that 16 months have flown by so quickly, resulting in the launch of Red Hat Enterprise Linux v.4 in February 2005. The last article on v.3 provided a behind-the-scenes glimpse of the challenges we face here at Red Hat in order to deliver a robust enterprise-caliber Linux distribution. Although we still face many of the same challenges with the new release, there were many changes in how we conduct business. In this article, I cover the new challenges we faced and how we adapted to address them.

Out of practical necessity, I cover only a small fraction of the hundreds of features and issues we address in a new Red Hat release. Also for this reason, I am unable to identify all of the literally hundreds of contributors, both internal and external. Allow me to apologize up front to my Red Hat friends who escape mention here (it's not that you too aren't awesome).

The Stakes Get Higher

Truly the most remarkable trend in the computing industry is the dramatic rise in Linux adoption. Seemingly, every day, there are media alerts, on-line articles, notifications from our peers in local Linux User Groups (LUGs) and sales announcements reporting large new user communities migrating to Red Hat Enterprise Linux. For example:

- Entire country governments, government agencies and departments.
- Public school systems, from grade schools to universities.
- Huge corporations increasingly are making Red Hat Enterprise Linux their primary software development platform and engineering design workstations.
- Call centers and desktops.
- Scientific research, public and private.
- Telco and increasing usage in embedded appliances.

It is an immensely gratifying phenomenon to have the work you do benefit a huge and swiftly climbing user community. The collective user base of both Red Hat Enterprise Linux and the Fedora community version is well above a million users. In fact, due to the proliferation of our software, it is impossible to derive exact numbers to characterize the popularity. Given this scope, all our developers have a strong sense that their contributions truly have impact. There is a betterment of humanity aspect that is inherent with the spread of open-source software.

Given the great diversity of our user base, it becomes increasingly challenging to meet its needs with a finite set of internal developers and testers. In order to keep pace with the growing user base, we needed to find a better way to scale our effectiveness. To accomplish this, we had to look no further than the open-source model that is the core of Red Hat's philosophy. That is, to involve a broader community of participants in an inclusive "early and often" approach. This was the genesis of Fedora.

Fedora

Fedora is one of the main differences in the Red Hat Enterprise Linux v.4 development as compared to Red Hat Enterprise Linux v.3. There are several objectives of the Fedora Project, including:

- Providing a freely downloadable Linux distribution for interested contributors. By aggregating the latest available versions of a great diversity of packages, Fedora is an ideal incubator for new technology.
- Providing a forum for external contribution and participation.
- Forming a proving ground for new technologies that later may appear in an upcoming Red Hat Enterprise Linux release.

The experiences gleaned from Fedora are invaluable in the productisation of Red Hat Enterprise Linux. The Fedora com-

munity consists of tens of thousands of users. This volume is larger than the Red Hat Enterprise Linux beta-testing audience. Through the experiences of Fedora, we are able to get a solid understanding of which package revisions and new technologies are mature enough for inclusion in Red Hat Enterprise Linux. The Fedora community members were involved actively in many aspects of development.

A perfect example of community involvement in Fedora development consisted of an external contributor developing an awesome Java application that output diagrams illustrating where time was spent in the boot process. This highlighted slow-starting system services. One such offending service identified by this application subsequently had its starting time corrected to take half a second rather than 20 seconds.

Portions of Fedora are even developed and maintained entirely outside of Red Hat. A key example of this is the yum package delivery and update technology. This shows how Fedora is free to grow in many dimensions, unrestricted from Red Hat's agenda.

For those who demand the latest bleeding-edge technology, Fedora is a perfect, free distribution. For those who demand a more stable supported product, Red Hat Enterprise Linux is the right choice. The Fedora Project has moved ahead in the new technology curve from Red Hat Enterprise Linux v.4. In this manner, it forms a glimpse of promising new features that may appear in future Red Hat Enterprise Linux releases.

The success of the Fedora Project truly has been win-win. Community contributors and users receive a free vehicle to mature open-source technology. Enterprise customers benefit from an increasingly feature-rich and mature product after completion of the stabilization phase.

Red Hat Enterprise Linux v.4 Requirements Planning

With this increasingly diverse user base comes a corresponding large set of requirements. Example requirements include bug-fix requests, software feature addition and hardware enablement. By far, our biggest challenge is to strive to prioritize customer bugs and feature requests to identify the set that yields broadest general usefulness.

In the initial planning phases of Red Hat Enterprise Linux v.4, we carefully reviewed more than 500 feature requests. This was accomplished in numerous marathon sessions of feature reviews interspersed with countless hours of follow-up scoping of the viability and developer time required to deliver. Below are some of the main themes we tried to focus on in Red Hat Enterprise Linux v.4:

- Security.
- 2.6 kernel.
- Storage management.
- Ease of use, particularly in the desktop.

Highlights of each of these main themes appear in upcoming sections.

On-Site Partners

In addition to an increased user base since the introduction of Red Hat Enterprise Linux v.3, we also have fostered closer working relationships with a growing set of hardware and software partners. We recognize that the operating system itself is only one layer in an overall solution stack that end customers need in order to make Linux practical for them in solving their computing needs. For this reason, we work closely with our partners in terms of identifying our priorities, aligning schedules and addressing issues critical in enabling their hardware and software.

Our hardware and software partners increasingly are seeing value in working closely with Red Hat. Historically, it has been highly challenging for us to accommodate the insatiable and diverse requirements from our partners. As much as we would like to satisfy everyone, ultimately we do have a finite staff and time frame in which to do this work. In response, we have invited many of our partners to join us inside Red Hat to work alongside our developers to augment our staff to achieve mutually beneficial objectives. For example, we currently have multiple on-site staff members from IBM, Intel, SGI, HP, Fujitsu and NEC. Here are some of the benefits:

- Increased delivery of feature enhancements and bug fixes.
- Better communication at the engineering level.
- Faster turnaround time to address problems. When it comes to the short time windows involved in new platform support, these efficiencies have yielded support that otherwise would

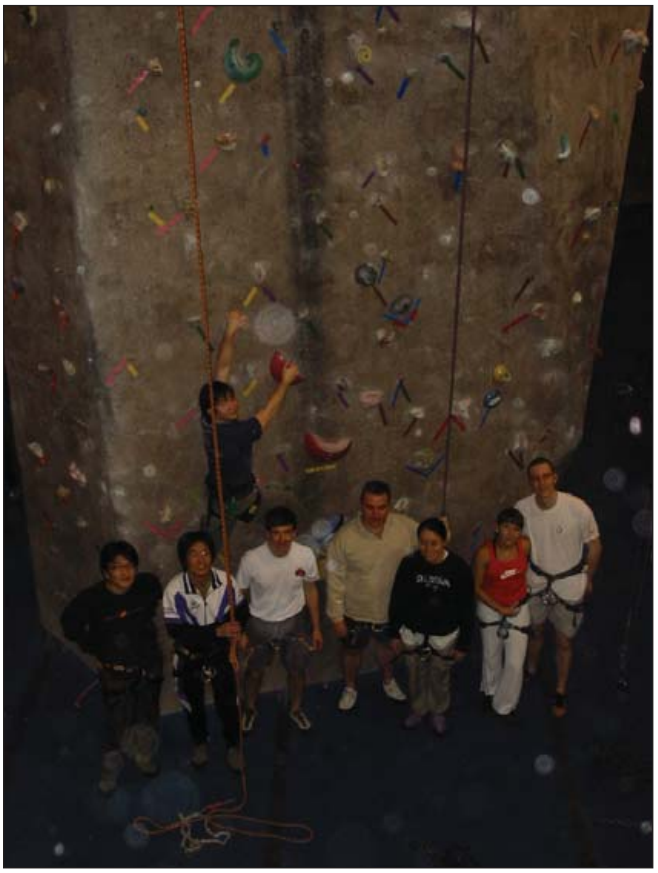


Figure 1. The Red Hat rock stars out for a night of climbing.

have been deferred to the next update cycle.

- Partners get an inside view into how the Open Source community functions and how to become effective community participants.

- Fostering friendships from people around the world.

The on-site partner contribution benefits the product set beyond the parochial interests of the sponsoring company. For example, although the SGI team's primary mission was support of their large CPU count Altix platform, a side effect was overall improvement in scalability in generic layers, which benefits all architectures. Another example is the work the Fujitsu team accomplished by adding diskdump support. Other hardware partners have augmented this support in Red Hat Enterprise Linux to yield improved problem analysis capability by our collective support organizations.

Numerous on-site partners are here from Japan. We invited them to join us at Boulder Morty's indoor rock climbing gym. It's amazing how much trust it fosters to be hung 40 feet up on a rope with your new-found friends. Given that English isn't their primary language, I often wonder how much of the introductory rock climbing instruction they understood before we gave them the "Go!" thumbs up. Figure 1 shows the Red Hat and partner crew out for our weekly climbing session.

Security

One of the major themes of Red Hat Enterprise Linux v.4 was security. Security considerations prevail throughout the entire distribution. For example:

- Increased compile time checking for buffer overflows, stack overflows, bounds checking, initialization and correctness checks have been added to the compiler. We have defensively incorporated these checks into our internal build processes. Having core GCC compiler developers on staff enables them to provide such constructive recommendations for defensive programming.

- Increased kernel and runtime loader provisions to prevent execution of malicious code and blocking of common stack overflow techniques. This has resulted in Red Hat Enterprise Linux v.4 not being vulnerable to a large class of exploits (see Resources).

- Participation and monitoring of several industry consortiums whose missions are to share security exploit information and work on common resolutions.

SELinux

SELinux refers to Security Enhanced Linux. Details of SELinux have been presented in prior *Linux Journal* articles (see Resources).

At its core, SELinux consists of a set of low-level primitives that provide fine-grained access control. Prior to the advent of SELinux, the Linux security model had been a rather all-or-nothing approach, in that the two common cases were general unprivileged user applications and privileged applications. The privileged applications typically consisted of system services such as bind, Apache, MySQL, Postgres, ntpd, syslogd, snmpd and squid. The historical downside to having all-powerful system services is that if they were compromised by a virus attack or other exploit, the entire system could then become compromised.

SELinux provides a means of tightly restricting the capabilities of user applications and system services to a strict need-to-know authorization. For example, it sets access control on the Apache Web server (httpd) to limit the set of files and directories it is able to modify. Additionally, Apache is strictly limited to what other applications it is capable of executing. In this manner, if Apache is attacked, the set of damage that can occur is well contained. In fact, SELinux is so well contained that one of Red Hat's developers, Russell Coker, has set up a Fedora system where he provides the root password and invites people to see if they can inflict damage to the system.

What is most monumental about Red Hat Enterprise Linux v.4's SELinux implementation is that it is the first widely adopted commercial operating system to provide such fine-grained security integrated in the newest release. Historically, it has been the case that such fully featured secure operating systems have been relegated to obscure forks of mainstream products, which typically have lagged a year or two behind the respective new releases.

The implementation of SELinux got its tentacles into virtually all areas of the distribution. This included:

- Implementation of policies for the core system services.

- Providing default policies for all RPM packages we provide.

- Installer and system management utilities to enable end users to define access domains of their own.

- Kernel support throughout a range of subsystems.

There were many challenges in the implementation of SELinux. On the kernel front, the core SELinux primitives were highly at risk of being accepted into the upstream 2.6 Linux kernel. James Morris valiantly completed the implementation and garnered the required upstream consensus. On the user-level package front, the introduction of SELinux required a specific or default policy to be constructed for each package. Naturally, this at times was a bumpy process as we sorted out which files should be writable and other details.

Minor implementation glitches would wreak havoc across the entire distribution. However, it also resulted in SELinux being the initial scapegoat for virtually all problems. Dan Walsh was a true workhorse in pouring through this onslaught of issues.

2.6 Kernel

"Upstream, Upstream, Upstream"—this became the mantra among our kernel team throughout the entire duration of Red Hat Enterprise Linux v.4 construction. The reason for this is that every change in which Red Hat's kernel diverges from the upstream Linux community kernel.org becomes a liability for the following reasons:

- Peer review—all patches incorporated upstream undergo a

rigorous peer review process.

- Testing—there are thousands of users worldwide from hundreds of companies who routinely access upstream kernels.
- Maintenance burden—the closer we are to upstream kernels, the more efficient we can be about pulling fixes back into the maintenance streams for shipping products.
- Next release—getting fixes and features into upstream means that we don't have to re-add the feature manually into future releases.

These principles are core to the value of true community open-source development. As testament to Red Hat's active participation in the upstream Linux Kernel community, through the course of 2.6 development more patches were accepted from Red Hat kernel developers than from any other company. During the past year, more than 4,100 patches from Red Hat employees were integrated into the upstream 2.6 kernel. In contrast, other companies boast that their offering contains the most patches on top of the community kernel. An interesting statistic is that currently, more than 80% of all kernel patches originate from kernel developers employed explicitly to do such development. The kernel has become mostly a professional employment endeavor, not a hobbyist project.

Red Hat's developers were highly active in upstream 2.6 development. Some of the areas of involvement included:

- Filesystem.
- Virtual Memory (VM) management.
- SELinux and other security features.
- Networking.
- IDE and USB.
- Serial ATA.
- Logical Volume Manager (LVM).
- Graphics.
- Hardware and driver support.

Arjan van de Ven and Dave Jones, Red Hat Enterprise Linux v.4 kernel pool maintainers, integrated kernel contributions from our collective internal kernel development team.

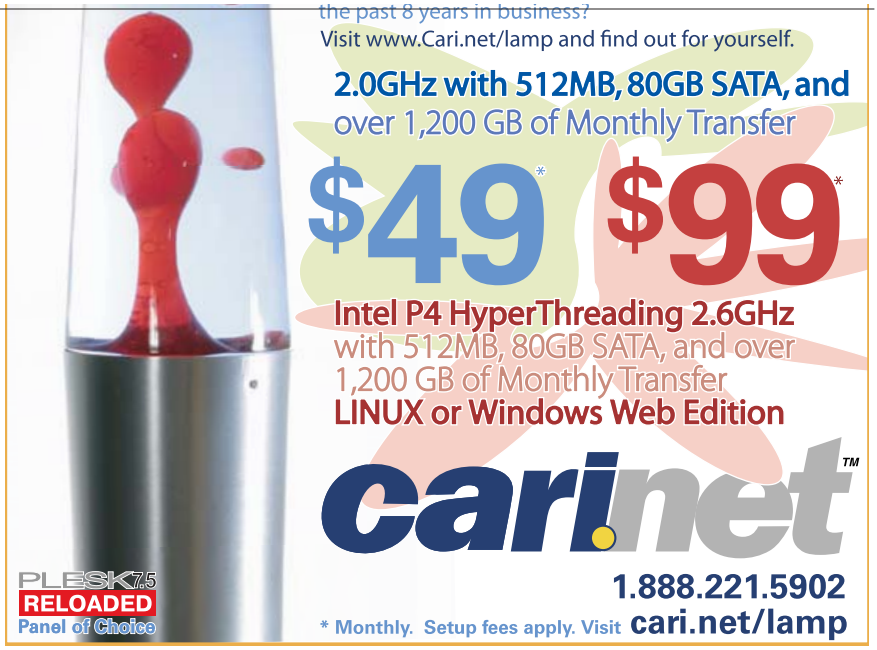
They frequently rebased our trees against the latest upstream kernels as well as integrated additional bug fixes, performance tunings, hardware platform support and feature additions. This is truly a monumental effort given that we simultaneously support seven different architectures: x86, x86_64—AMD64 and Intel(r) EM64T, Itanium2, IBM Power (31- and 64-bit), mainframe in 31- and 64-bit variants from a single codebase.

Initially, it was painful for Arjan to be beating everyone over the head to ensure that all patches were accepted upstream prior to incorporating them into our pool. Through his vigilance, the entire team became conditioned to working upstream first. In the short term, it involves more effort on the part of the developer to work both internal to Red Hat as well as upstream. However, in the long term, as described above, the benefits are considerable.

Storage Management

A large class of new Linux deployments consists of proprietary UNIX migrations. These users represent a set of enterprise customers who have high expectations (a euphemism for highly demanding). Traditional functionality gaps in Linux consist of robust software volume management capabilities. In response to these needs, over the course of Red Hat Enterprise Linux v.4, Red Hat acquired a strong team of storage-centric experts when Red Hat purchased Sistina. In this manner, Red Hat now employs the major upstream developers of the Logical Volume Manager (LVM) technology.

Overall ease of use has been improved in the installer, where it now enables the user to create LVM vol-



umes. Through the use of a graphical interface in Disk Druid, usage of LVM is much more approachable to the end user. Another example of ease-of-use improvements are the capabilities to grow both LVM volumes and ext3 filesystems that are on-line. This obviates the need to unmount the filesystem, back up, grow the volume, reformat the filesystem and restore the data.

We also wanted to take open-source storage management to the next level to provide a cluster filesystem. The industry trends have been toward distributed computing among large sets of commodity computers. Although that yields cost savings in hardware, it increases costs of managing storage and filesystems among a distributed pool of servers. To address this need, Red Hat has augmented the LVM layer to operate in a clustered environment by layering a robust cluster filesystem called GFS.

In keeping with Red Hat's core values of being an open-source player, the complete source code base for LVM and GFS is now freely available to the Linux community at large. Ongoing development has rekindled industry-wide contributions. Cluster Suite is the name of the productised version of GFS and LVM, which is layered on top of Red Hat Enterprise Linux.

Desktop

One of Red Hat's largest areas of increased investment is in what we refer to as the desktop space. Under the guidance of Havoc Pennington, we have formed an extensive close-knit team of developers. The primary mantra of the desktop team has been ease of use. If you look closely at the new adoptions of Linux you will see an increasing trend of usage in less computer-savvy scenarios. Examples include kiosks, call centers, government agencies and earlier grade-school levels.

The desktop team worked with our application developers to identify the most useful application set. Although there are more than 80,000 projects on Sourceforge.net, for example, it is impractical to include all of them in our distribution. One of our main roles as a system integrator is selecting and organizing the most useful applications. In v.4 we have reorganized how the applications are listed in the menus so as to be grouped logically by function.

Inside the walls of Red Hat, we take the open-source model to an extreme, where product decisions are debated by anyone who has a nearby soapbox. Given that everyone is a self-proclaimed authority on "what the users want" and what "usability" means, this provided ample fodder for highly emotionally charged debates. This all came to a head in the selection of the default browser. The main contenders were Firefox and Epiphany. The on-line e-mail debates raged on. In the end, Havoc pulled all interested parties together for a raucous conference call to hash things out. The result was the selection of Firefox. Given the huge amount of attention that Firefox has been garnering, both in the media and practical deployments, we think we made the right choice.

These debates are a core part of being at Red Hat. They become so volatile because the crew sincerely cares about what they are doing. Most people here feel part of something bigger than a small company. The high level of energy, creativity and enthusiasm found at Red Hat make it extremely challenging to be a manager. Sometimes it seems like I'm a referee to a crew of prize fighters, who in addition to sparring with each other, often share a punch to the head with me too. Perhaps I should have strived to find a more constructive example. It's really not combative here, just highly stimulating and challenging. After living in this world for 3.5 years now, I can't imagine what it's like to work at a place that would be "just a job".

One of the key usability technologies that our developers (including Havoc Pennington and John Palmieri) were involved with is D-BUS (see Resources). D-BUS is a communication and event mechanism that enables a range of desktop applications to complement each other in a coordinated manner. For example, the insertion of a CD results in the launching of a corresponding application depending on media format type. Similarly, D-BUS is used for USB device hot plug, for example, to initiate configuration and startup of network services or mounting filesystems from USB pen drives.

Ease of use was further enhanced through the bundled collection of third-party proprietary applications. This is done for the convenience of the end user, so that it doesn't become an egg hunt for them to find commonly used applications. This resulted in the bundling of RealPlayer, Helix Player, Adobe Acrobat Reader, Citrix, Macromedia Flash and a Java runtime environment (JRE).

Worldwide Development

In April 2004, Red Hat conducted a global company meeting in Raleigh, North Carolina. The entire company was invited. One of the strongest impressions I took from this meeting was how truly worldwide Red Hat is. It seemed as though there were as many non-US team members as US members. In addition to the US, development is conducted in Australia, Canada, Germany, Czech Republic, UK, Japan, India and Brazil.

Not all development is conducted within the offices of Red Hat. Through the worldwide legions of contributors to Fedora we invite broader participation. We actively contribute and draw from a great diversity of community open-source projects. Again, this substantially broadens the circle of participation. In many ways, this inclusive process makes Red Hat feel like a trusted steward of the community, forming a distribution representing the best and brightest technology. This is a privilege we do not take for granted as we know it needs to be continuously earned every day. This makes both Red Hat Enterprise Linux and Fedora truly distributions "by the people, for the people".

Red Hat Enterprise Linux v.4 is supported in 15 different languages. These translations are all performed as an integral part of the development cycle. Consequently, the translation process doesn't lag the release or introduce forks in the development trees. We have a team of "translation elves" located in Australia who magically do their work at an opposite phase of the clock from headquarters. This results in a nearly real-time translation that tracks development changes. Additionally, there are many contributors to Fedora who are actively involved in internationalization activities.

Hear Yourself Think Again!

WhisperStation™

Originally designed for a group of power hungry, demanding engineers in the automotive industry, WhisperStation™ incorporates dual 64-bit AMD Opteron™ or Intel® EM64T™ processors, ultra-quiet fans and power supplies, plus internal sound-proofing that produce a powerful, but silent, computational platform. The WhisperStation™ comes standard with 2 GB high speed memory, an NVIDIA FX1300 PCI Express graphics adapter, and 20" LCD display. It can be configured to your exact specifications with either Linux or Windows, and specialized applications including Mercury's AmiraMOL,™ PathScale's EKO Compiler Suite or the Intel Performance Tools. RAID is also available. WhisperStation™ will also make a system administrator very happy, when used as a master node for a Microway cluster! Visit www.microway.com for more technical information.

Experience the "Sound of Silence".
Call our tech sales team at 508-746-7341 and design your WhisperStation™ today.

AMD 64 Opteron™

intel inside XEON

Microway®
Technology you can count on℠

Figure 2. The Red Hat Crew from the Westford, Massachusetts Office

Lessons Learned

There are several ways in which Red Hat has improved upon our development methodology over the course of Red Hat Enterprise Linux v.4's construction. Interestingly, the main theme of these improvements has been to stick to core proven Linux open-source development practices. Although we did subscribe to these practices previously, we paid increased focus this time around to the following:

- Upstream—doing all our development in an open community manner. We don't sit on our technology for competitive advantage, only to spring it on the world as late as possible.
- Customer/user involvement—through a combination of Fedora and increased "early and often" releasing of beta versions through the development cycle, we are able to get huge volumes of invaluable feedback (both good and bad).
- Partner involvement—on-site partner developers have augmented our ability to address features, bugs and incremental testing.
- Avoiding feature creep—putting a clamp on the introduction of late-breaking features in order to allow stabilization.

We are all extremely grateful for the steady guiding influences of Donald Fischer who did an outstanding job as overall product manager and release manager. He was at once a diplomat, innovator, bookkeeper and go-to guy. Hats off to "the Donald".

What's Next?

Red Hat is truly a restless place to be. It seems that no sooner have we shipped one release, than we are already behind on the next one. This is due to the fact that in addition to new release development, we also support prior releases for a seven-year interval. So, for example, here's the list of releases concurrently in development now:

- Fedora Core 4 (FC4).
- Red Hat Enterprise Linux v.2.1 Update 7.
- Red Hat Enterprise Linux v.3 Update 5.
- Red Hat Enterprise Linux v.4 Update 1.
- Red Hat Enterprise Linux v.5.
- Numerous new technologies in pre-release stages, targeted at various upstream and internal release delivery vehicles.

Never a dull moment, and we wouldn't have it any other way!

Resources for this article: www.linuxjournal.com/article/8204.

Tim Burke is the director of Kernel Development at Red Hat. This team is responsible for the core kernel portion of Red Hat Enterprise Linux and Fedora. Prior to becoming a manager, Tim earned an honest living developing Linux high-available cluster solutions and UNIX kernel technology. When not juggling bugs, features and schedules, he enjoys running, rock climbing, bicycling and paintball.

Sun microsystems

Java™

INNOVATE
with the Power of Java™

June 27–30, 2005

JavaOne℠ Pavilion: June 27–29, 2005
Moscone Center, San Francisco, CA

Connect with the full power of Java™ technology at the JavaOne℠ conference.

In-depth EDUCATION
Evolve your skills in hundreds of expert-led technical sessions.

Real-world INNOVATION
Evaluate proven tools and technologies in the JavaOne℠ Pavilion.

Global COMMUNITY
Celebrate the tenth anniversary of Java technology.

Visionary INSIGHT
Hear what the future holds from industry leaders.

Experience a week unlike any other

EXPERIENCE THE 10TH ANNUAL JAVAONE℠ CONFERENCE.

Core Platform | Core Enterprise | Desktop | Web Tier | Tools | Mobility and Devices | Cool Stuff

Save $200!

REGISTER
by May 27, 2005 at java.sun.com/javaone/sf

JavaOne℠
Sun's 2005 Worldwide Java Developer Conference℠

Copyright © 2005 Sun Microsystems, Inc. All rights reserved. Sun, Sun Microsystems, the Sun logo, Java, the Java Coffee Cup logo, JavaOne, the JavaOne logo, Java Developer Conference, Java Community Process, JCP, 100% Pure Java, J2EE, J2ME, J2SE, Jini, Solaris, "Write Once, Run Anywhere," and all Java-based marks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Big Drives?

I am running Red Hat 9.0, Fedora 1 and Debian 3.0r4. I have contacted Intel about running 160GB hard drives. They replied, "The OS is what determines what size the hard drive can be." And they quoted Windows 2000 and Windows XP, so I thought maybe the BIOS was involved. What is your take on this mater, and where can I find references on the subject?

--

Georg Robertson, grobertson29@earthlink.net

The machine's BIOS actually defines certain limits for hard disks, from the old Int 13 specification for a DOS (yes, Disk Operating System) capacity limit of around 8GB to the most modern BIOS and drive hardware capabilities of 32-bit sector numbers that allow a theoretical capacity limit of more than 2TB and with it a whole new challenge for software. Of course, the OS disk drivers, bootloader, filesystem and probably other features, such as software RAID, determine the actual available capacity of a disk drive or set of disk drives.

--

Felipe Barousse Boué, fbarousse@piensa.com

I often can get Linux working on strange drive geometries that give Windows fits, because the kernel can be told what to do with them manually. There is an excellent guide on just this topic, and I suggest you start there: www.tldp.org/HOWTO/Large-Disk-HOWTO.html.

--

Chad Robinson, chad@lucubration.com

Using a Mobile Phone with a USB Cable?

I am able to connect to GPRS mobile devices, including the Motorola V66 and Timeport, by using a serial cable. But the latest GPRS mobiles come only with USB data cables. I tried but was unable to connect one to a Linux system; I was told the PC could not find the modem. Can you tell me how to connect it or suggest suitable drivers for it?

--

kimaya@vsnl.com

These devices almost invariably are still serial but include a USB-to-serial-device chip to provide the USB interface. There are two forms of these conversion chips. One, such as the FTDI chipset, is designed to create a virtual serial port through the USB interface. These products usually already are supported under Linux, and if not, it typically is only a matter of time before this happens.

The second type is proprietary and relies on custom software drivers that communicate to the remote chipset. These tend to make portability more difficult, because manufacturers still generally release these drivers only for Windows, and without the driver you cannot communicate with the device. Fortunately, there are fewer of these, but because they can be less expensive than virtual serial port chipsets, some manufacturers will continue to use them. Your best bet is simply to avoid these types of products by monitoring newsgroups, forums and other information sources for Linux user success stories before purchasing them.

--

Chad Robinson, chad@lucubration.com

Plenty of GPRS phones can be used with Linux; the following Web resources provide a lot of useful information about GPRS phones and their uses. In conjunction with a Linux system, take a look at kotinetti.suomi.net/mcfrisk/linux_gprs.html, users.tkk.fi/~kehannin/bluetooth/bluetooth.html *and* markus.wernig.net/en/it/usb-serial-handy-ppp.phtml.

I also recommend that you consider using a Bluetooth wireless interface to link your Linux box, with the proper adapter and your phone, which hopefully has Bluetooth capacity.

--

Felipe Barousse Boué, fbarousse@piensa.com

Tuxmobil.org maintains a list of compatibility reports and how-to documents on connecting using specific mobile phone models.

--

Don Marti, dmarti@ssc.com

Error from MySQL Client

I am trying to use the GUI MySQL client with Fedora Core 3, but it is failing, returning this:

```
[anupam@localhost mysqlgui-1.7.5-1-linux-static]$ ./mysqlgui

mysqlgui: dynamic-link.h:57: elf_get_dynamic_info:
Assertion `! "bad dynamic tag"' failed.

Aborted
```

Any ideas what is wrong?

--

Anupam De, anupam@sail-steel.com.

Did you download mysqlgui in binary form as opposed to text or ascii? If you transferred text or ascii, your file may have been corrupted. Alternatively, try downloading the statically compiled version of the mysqlgui software package instead of the semi-static binary. You will get rid of some dependencies, as the slightly larger executable includes everything required.

--

Felipe Barousse Boué, fbarousse@piensa.com

Setting IRQs for Serial Ports

I have Win4Lin running on SUSE 9.2 and am having a hard time changing the IRQ on com port 2. I need Windows for an energy management program and must call out to check several building systems. Linux has the IRQ set at 10, but I need to have it set at 4. Can you tell me how to change the IRQ?

--

John Langston, jdl.28@cox.net

You should be able to change the IRQ in your BIOS settings. If that doesn't work, use the setserial program on Linux to change this value.

--

Greg Kroah-Hartman, greg@kroah.com

Do a `man setserial` *to learn your command options. Be aware that if your physical serial ports do have fixed IRQ and/or memory*

addresses, you may run into conflicts when playing with setserial and/or with other devices.

--

Felipe Barousse Boué, fbarousse@piensa.com

GigaDrive Doesn't Work

I recently purchased a Linksys GigaDrive on eBay. The unit seems to power up and such, but I cannot access or run any of the applications. I am thinking maybe the drive has been formatted or replaced and I need to reload the Linux software and apps. Do you have any advice on how to do this, other than to send it to Linksys? I am A+ certified, but I don't have much Linux experience. I was thinking that if I could obtain a restore CD, I may be able to rebuild it—is that true? Of course, if I can do that, I need to find such a restore CD. Any suggestions or advice?

--

Randy Warner, warn4421@bellsouth.net

*There is a page on how to load the GigaDrive's "firmware" on the Linksys site: (*www.linksys.com/support/support.asp?spid=17*).*

If that doesn't work, and you have access to an identical hard drive from a working GigaDrive, you could make a bit-for-bit copy by hooking the working drive up to a Linux box as master and the nonworking drive as slave on the secondary IDE interface and doing:

```
dd if=/dev/hdc  of=/dev/hdd
```

--

Don Marti, dmarti@ssc.com

Backing Up a Dual-Boot System

I currently use Microsoft Windows XP Pro with the intent of migrating to Linux after I get used to running it and administering it. The current backup software I use is Norton Ghost from System Works 2004.

I tried installing Fedora Core 1, as it came free with a book I bought. Installation went without a hitch, and I liked what I saw and used. But, when I boot back to Windows to use Ghost, Ghost gives me this error message:

```
Back-up Failure. Not space in the MBR.
```

I said, "forget Norton, I'll do my backups with Linux." But I haven't the faintest idea what to use on Linux. Any suggestions?

--

Lev Ranara, pinoy_techie@yahoo.com

Backups under Linux are usually straightforward. Unlike Windows, there is no special system data (registry or system configuration) that cannot be copied through traditional means. A straight file copy, in fact, usually is sufficient for a "complete" backup, unless a database server is running. In this case, it may need to be shut down during the backup.

Complex solutions abound and allow managed, catalog-style backups and restores of individual files. These are available as free software (such as Amanda and Bacula), from traditional vendors of Windows backup software (VERITAS, CA and so on), as well as from some vendors specifically focused on Linux (such as BRU). However, since you're using Ghost, it sounds like you're not really doing file-based backup anyway. The simplest solution thus would be a compressed tar archive. Restoring the entire system then is a simple matter of partitioning and formatting the drive, extracting the archive and reinstalling the boot loader.

If that's true, start with tar and see if it suits your purposes. A command such as:

```
tar -jlcvf /tmp/mybackup.tgz /bin /boot /dev /etc \
```

often suits the most basic needs. Then, simply copy /tmp/mybackup.tgz onto CD, tape or another server. You also can tar directly to tape.

--

Chad Robinson, chad@lucubration.com

My best experiences in the Linux backup world come from using the good old tar command, the compression utilities such as zip and bzip, and some scripts I have written for each specific backup need. It's reliable, portable, straightforward and free—freedom and money-wise. For more information, see www.linux-backup.net *for everything related to Linux and backups. The book* Unix Backup and Recovery *also deals with the subject; it was reviewed on* LJ *at* www.linuxjournal.com/article/3839.

Also, try installing FC3 as FC1 is now deprecated. FC3 has a lot of nice features such as drag and drop to burn CDs, which may be useful for backups.

--

Felipe Barousse Boué, fbarousse@piensa.com

Client Connects, but TFTP Fails

I'm trying to get my TFTP server running properly, and I'm not having any luck figuring out the problem. Here's the scoop. I'm running Fedora Core 3 on a PIII machine. I've installed the latest tftpd server from rpmfind.net, and have configured xinetd/in.tfptd properly (I think). Using a tftp client on another Linux machine, I can connect to my tftp server, but the read requests go unanswered. The client times out after several retries. In /var/log/xinetd, I see the following entries for each read request sent by the client:

```
05/3/16@14:11:14: FAIL: tftp address from=153.90.196.30
05/3/16@14:11:14: START: tftp pid=20184 from=153.90.196.30
05/3/16@14:11:14: EXIT: tftp pid=20184 duration=0(sec)
```

Here is what I've done to configure the server. I created a user tftp with home dir of /tftpboot and ran /sbin/nologin. I added an entry to /etc/hosts.allow of `in.tftpd:ALL`. I created a directory /tftpboot with correct permissions and ownership. I then created the file /etc/xinetd.d/tftp with the following contents:

```
service tftp
{
  disable = no
  socket_type     = dgram
  protocol        = udp
  wait            = yes
  user            = root
```

```
    server          = /usr/sbin/in.tftpd
    server_args     = -s /tftpboot -u tftp
    per_source      = 11
    cps             = 100 2
    flags           = IPv4
    #only_from      = 153.90.196.30
}
```

I've tried this with only_from both commented and uncommented. I've also made sure that the firewall trusts UDP and TCP on port 69. I verified that the contents of /etc/xinetd.conf are correct, and I verified that tfptd is running via chkconfig. I also verified that port 69 is available via netstat. I've tried running in.tftpd in standalone mode (server_args = -l).

I've been working on this problem for three days and am getting nowhere. I'm something of a newbie to Linux, but I have asked more experienced folks for insight to no avail and have spent hours trying to find instances of this problem on the Internet, also to no avail. So, I'm hoping you folks can point me in the right direction.

--

Todd Trotter, ishamt@esus.cs.montana.edu

It seems as though you have done almost everything correctly. Some issues come to mind though. First, change the user to nobody on the file /etc/xinetd.d/tftp; otherwise, the in.tftpd dæmon runs as root, which is not safe.

Second, make sure the lines:

```
tftp            69/tcp
tftp            69/udp
```

are not commented out in the /etc/services file. Also, I suggest checking the file /etc/hosts.deny to see if you are blocking requests for the in.tftpd dæmon, for all services or for requests from a specific IP (client machine).

For testing purposes only, make sure this file is empty, reload xinetd (service xinetd reload) and try again. Also, for testing only, turn off your firewall (service iptables stop) and test again. Test and make your setup work locally by issuing `tftp localhost` *before testing remotely. Hope this helps.*

--

Felipe Barousse Boué, fbarousse@piensa.com

Is Garbage Collection the Answer?

I learned about garbage collection (GC) from your journal. I do have a problem. Let me explain the situation that exists. Initially, the project occupies 192MB of RAM in Linux. It was allowed to run continuously. Then, after 12 hours, we noticed it was using 335MB. What is the solution for this problem? Is it due to garbage? Will the BDW garbage collector provide a solution? The project includes char pointers, and it doesn't include any malloc functions.

Will BDW GC work only if we include malloc, calloc or realloc functions? Can I have a program that runs along with my project and releases free memory?

--

Mythily J., mattuvar@yahoo.co.in

The answer to the last question is no. Unless you do really hairy and hard-to-debug things, only your program can free memory that it allocated.

The others are really good questions, and the only way to know for sure is to try it with your code. Even though you may not be using the malloc family of functions, you might be making library calls that allocate memory and then omitting some of the calls required to free it.

The good news is that you can build a version of your program that uses GC for all memory management, including memory allocated in library code, by "hooking" it in to malloc. See Listing 1 in this article: www.linuxjournal.com/article/6679 *for an example.*

--

Don Marti, dmarti@ssc.com

Runlevel Editing

In the April 2005 Best of Technical Support, in "Old Red Hat", Timothy Hamlin suggests changing the /etc/inittab entry from:

```
x:5:respawn:/etc/X11/prefdm -nodaemon
```

to:

```
x:3:respawn:/etc/X11/prefdm -nodaemon
```

to suppress the X graphical login. I think he made an error here. His reply will launch X at runlevel 3. Instead change:

```
id:5:initdefault:
```

to:

```
id:3:initdefault:
```

to change the default runlevel.

Also, in "Tweaking inodes and Block Sizes", Don Marti points out that Red Hat 9 is no longer supported and that this might be an issue for an older 486 system. The bigger issue is the amount of RAM Red Hat requires for the install. I'm not sure if it will install with 32MB of RAM. It definitely won't with 16MB, which is what my old 486 laptop had.

--

Roland Roberts, roland@astrofoto.org

Either inittab change will work. The second has the advantage of preserving the "runlevel 5 is GUI login" tradition that Red Hat users are used to. The Fedora release notes at fedora.redhat.com/docs/release-notes/fc3/x86 *list a Pentium as the minimum processor and 64MB as minimum memory for a text install. (See the last letter for an alternate approach.)*

--

Don Marti, dmarti@ssc.com

What about Fedora Legacy?

In the April 2005 Best of Technical Support, Don Marti writes that "Neither Red Hat 9 nor Red Hat 6.2 is still supported, which means

no more security updates." Although Red Hat has dropped support for Red Hat 9, the community-based Fedora-Legacy Project (www.fedoralegacy.org) is working to provide security updates for Red Hat 9 as well as Red Hat 7.3 and Fedora Core 1 and (soon) 2. Mr Marti does the project a disservice by ignoring its efforts.

--
John Dalbec, jdalbec@cboss.com

At the time we went to press, Fedora Legacy was not actively releasing security updates.

--
Don Marti, dmarti@ssc.com

Really, Fedora on a Pentium?

The Best of Technical Support column in the April 2005 issue of LJ contains some incorrect and incomplete statements in response to a user who wants to use Red Hat 9 on 486 computers. Don Marti writes, "[Red Hat's] successor, Fedora, requires a Pentium or better...No matter what you install, this class of machines will be too slow for a modern desktop." The RULE Project (www.rule-project.org) proves this wrong. One year ago, I ran Red Hat 9 on a Pentium I laptop with 32MB of RAM. Thanks to it, I used KOffice to make a presentation and Firefox for home banking: www.rule-project.org/article.php3?id_article=55 (see the linked screenshot).

Less than one month ago, we announced a version of our installer for Fedora Core 3: www.rule-project.org/breve.php3?id_breve=19.

Now, it certainly is true that full-fledged KDE, GNOME or OpenOffice.org installations under any desktop can be painfully slow, even on much newer computers. It is equally true that video editing or 3-D gaming requires state-of-the-art hardware. But, if by modern desktop, one means modern SOHO functionality—IMAP, digital signatures, HTML4/CSS support, CUPS, IM, Bayesian spam filtering, regardless of eye candy—there is no need to spend money. All it takes is a project such as RULE and efforts made on things such as mini-KDE. In any case, it is possible to run a modern, mainstream distro on slow hardware, with a bit of care and the right approach to the problem.

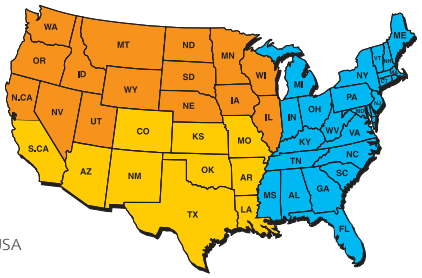
--
Marco Fioretti, mfioretti@mclink.it

Many on-line help resources are available on the *Linux Journal* Web pages. Sunsite mirror sites, FAQs and HOWTOs can all be found at **www.linuxjournal.com**.

Answers published in Best of Technical Support are provided by a team of Linux experts. If you would like to submit a question for consideration for use in this column, please fill out the Web form at **www.linuxjournal.com/lj-issues/techsup.html** or send e-mail with the subject line "BTS" to bts@ssc.com.

Please be sure to include your distribution, kernel version, any details that seem relevant and a full description of the problem.

LINUX JOURNAL



PO Box 55549
Seattle, WA 98155-0549 USA
www.linuxjournal.com

ADVERTISING SERVICES

VP OF SALES AND MARKETING
Carlie Fairchild, carlie@ssc.com
+1 206-782-7733 x110,
+1 206-782-7191 FAX

FOR GENERAL AD INQUIRIES
e-mail ads@ssc.com
or see www.linuxjournal.com/advertising

Please direct international advertising inquiries to VP of Sales and Marketing, Carlie Fairchild.

REGIONAL ADVERTISING SALES

NORTHERN USA
Joseph Krack, joseph@ssc.com
866-423-7722 (toll-free),
866-423-7722 FAX

SOUTHERN USA
Annie Tiemann, annie@ssc.com
866-965-6646 (toll-free),
866-422-2027 FAX

EASTERN USA AND CANADA
Martin Seto, mseto@ssc.com
+1 905-947-8846,
+1 905-947-8849 FAX

Advertiser	Page #
APPRO HPC SOLUTIONS appro.com	31
ARKEIA CORPORATION www.arkeia.com	29
ASA COMPUTERS www.asacomputers.com	53, 55
BLACK HAT BRIEFINGS (CONFEX PARTNERS LTD) www.blackhat.com	43
CARI.NET www.complexdrive.com	73
CIARA TECHNOLOGY www.ciara-tech.com	16, 17
CORAID, INC. www.coraid.com	27
COYOTE POINT www.coyotepoint.com	49
CYCLADES CORPORATION www.cyclades.com	C2, 1, 11
EMAC, INC. www.emacinc.com	87
EMPERORLINUX www.emperorlinux.com	15
ETNUS www.etnus.com	51
FAIRCOM CORPORATION www.faircom.com	7
FREE SOFTWARE FOUNDATION www.gnupress.org	69
GEEK CRUISES www.geekcruises.com	41
GOOGLE www.google.com/lj	47
HURRICANE ELECTRIC www.he.net	57
IRON SYSTEMS www.ironsystems.com	85
JAVA ONE java.sun.com/javaone/sf/pavilion/index.jsp	77
LPI www.lpi.org	83
LINUX JOURNAL www.linuxjournal.com	25, 65
LINUX NETWORX www.linuxnetworx.com/theworxlj	21, 23
LINUXCERTIFIED, INC. www.linuxcertified.com	91
MBX www.mbx.com	2
MICROWAY, INC. www.microway.com	C4, 75
MIKRO TIK www.routerboard.com	C3
MONARCH COMPUTERS www.monarchcomputer.com	8, 9
PENGUIN COMPUTING www.penguincomputing.com	19
THE PORTLAND GROUP www.pgroup.com	36, 37
RACKSPACE MANAGED HOSTING www.rackspace.com	5
SBE, INC. www.sbei.com	13
SERVERS DIRECT www.serversdirect.com	39
STAR MICRONICS www.starmicronics.com	61
TECHNOLOGIC SYSTEMS www.embeddedx86.com	45
TUX MAGAZINE www.tuxmagazine.com	68
TYAN COMPUTER USA www.tyan.com	35
ZT GROUP INTERNATIONAL www.ztgroup.com	33

SUSE Linux Professional 9.3

Novell released SUSE Linux Professional 9.3, which includes a complete Linux OS, more than 3,000 open-source packages and hundreds of open-source applications, productivity software and home networking capabilities. Designed for both Linux newcomers and longtime users, SUSE Pro 9.3 offers many new features, including an OS built on kernel version 2.6.11, KDE 3.4 and GNOME 2.10, Firefox 1.0, OpenOffice.org 2.0, F-Spot photo organizer, The GIMP 2.2, Mono 1.1.4, KDevelop 3.2, Eclipse 3.0.1 and improved VoIP support. SUSE Pro 9.3 also offers improved mobility support for Wi-Fi connections and Bluetooth devices, PDA and phone synchronization; iPod compatability; an integrated firewall, spam blocker and virus scanner; and Novell Evolution 2.0 and Kontact 3.4. Also included in version 9.3 are the XEN virtualization environment and intuitive search engines, plus support for AMD Athlon 64 and Intel Extended Memory 64 Technology.

CONTACT Novell Enterprises, 404 Wyman Street, Suite 500, Waltham, Massachusetts 02451, 781-464-8000, www.novell.com.

SMGateway

SMGateway is an open-source e-mail/security application from Fortress Systems, Ltd. SMGateway offers all of the functionality provided by MailScanner and SpamAssassin along with extensions and enhancements to provide a Web-based interface for users and administrators. These added features allow administrators to install, control and configure e-mail gateway operations, while allowing users to set their own spam preferences. It is designed to provide all e-mail gateway, Web access, SQL database, LDAP directory and monitoring applications on a single server. SMGateway features three levels of authentication; connectors to Microsoft Active Directory, POP- or IMAP-enabled directory service; an SQL configuration database; LDAP configuration data storage; and DCC, Pyzor and Razor2.

SMGateway is free for customers to download, and Fortress Systems provides three levels of support options.

CONTACT Fortress Systems, Ltd., 3807 Fulton Street NW, Washington, DC 20007, 202-338-1670, www.fsl.com.

OPTion

OPTion is a virtual thin client for the Linux workstation desktop. Compatible with GNOME and KDE, it provides a single application to connect to all major free and commercially available terminal server environments. All client sessions are configured and managed centrally, and all configured client sessions are presented and executed from within a central launcher. Client sessions include standard XDMCP, full screen and/or within a desktop window; secure direct X; secure X login, full screen and/or within a desktop window; RDP, full screen and/or within a desktop window; xRDP with integrated Ericom seamless applications for WTS 2000/2003 and a cost-free RemoteView terminal server agent; ICA with server and application browser; Ericom PowerTerm Emulator suite; NoMachine NX

Client, supporting NX Server 1.3 and 1.4; and native Tarantella. Supported Linux distributions include MandrakeLinux, Fedora, Novell/SUSE and Xandros.

CONTACT SmartFLeX Technology, Inc., 623 Selvaggio Drive, Suite 220, Nazareth, Pennsylvania 18064, 610-746-2390, www.smartflextech.com.

ConvertX SDK

Plextor Corporation announced the availability of a free Linux software developers kit (SDK) for ConvertX video-capture devices. The SDK can be used to develop for Plextor ConvertX PVRs, which offer real-time hardware-based MPEG-1, MPEG-2, MPEG-4 and Motion JPEG video encoding in a USB 2.0 video peripheral. The Linux SDK supports the Video for Linux 2 (V4L2) and Advanced Linux Sound Architecture (ALSA) specifications. It also supports deprecated Open Sound System (OSS) applications by way of the OSS compatibility layer provided by ALSA. The new driver, which requires the Linux 2.6 kernel, includes sample code that can be reused in open-source or proprietary applications to help developers get started.

CONTACT Plextor America, 48383 Fremont Boulevard, Suite 120, Fremont, California 94538, 510-440-2000, www.plextor.com.

ARCOS 4.0

Plus Three, LP, released ARCOS 4.0, an application built on Linux, Apache, MySQL and Perl and designed to be used by fundraising organizations. Standard features and uses of ARCOS are constituent relationship management, e-mail and link tracking, event management, social software and an on-line activism center. New features include improved real-time report generation from databases, an enterprise-class redundancy backup system, and a larger and faster user database. ARCOS' e-mail publishing feature allows users to organize and distribute e-mail lists based on a variety of factors stored in the database. The Web publishing tools offer customizable contributor pages and tell-a-friend pages. In addition, the e-mail and Web publishing tools are integrated to allow users to process up to two million messages an hour.

CONTACT Plus Three, LP, 180 Varick Street, Suite #1126, New York, New York 10014, 212-206-7819, www.plusthree.com.

Please send information about releases of Linux-related products to Heather Mead at newproducts@ssc.com or New Products c/o *Linux Journal*, PO Box 55549, Seattle, WA 98155-0549. Submissions are edited for length and content.

Why is LPI the Global Standard in Linux Certification?

Trusted.

All Linux Professional Institute certification programs are created using extensive community input, combined with rigorous psychometric scrutiny and professional delivery. We test the whole continuum of important Linux skills - we don't just focus on small, subjective tasks. LPI exams are not simply an afterthought used to help sell something else. LPI is a non-profit group that does not sell software, training or books. Our programs and policies are designed to meet educational requirements, not marketing.

Accessible.

LPI exams are available in seven languages, at more than 7,000 locations, in more than 100 countries. You take LPI exams when you want, where you want. In addition, special exam lab events around the world make our program even more affordable. And because we don't make exclusive partnerships, LPI is supported by a broad range of testing centers, book publishers and innovative suppliers of preparation materials.

Independent.

You switched to Linux to get away from single-vendor dependence. So why trade one form of vendor lock-in for another? LPI's program follows the LSB specification, so people who pass our tests can work on all major distributions. Because of its strong grass-roots base and corporate support both inside and outside the world of open source, LPI goes beyond "vendor-neutral" to truly address community needs.

LPI is IT certification done RIGHT!

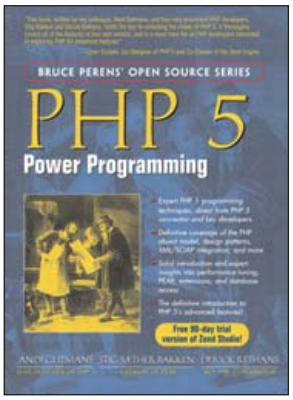
For more information, please contact us at Info@lpi.org or visit us at www.lpi.org.

Linux Professional Institute

PHP 5 Power Programming

by Andi Gutmans, Stig Bakken and Derick Rethans

Prentice Hall PTR, 2004 | ISBN: 0-131-47149-X | $39.99 US



PHP, arguably the world's best Web-scripting language, recently received a significant overhaul. Version 5 expands the object model of the language, adds support for new MySQL 4.x features and speeds up execution.

However, PHP 4 scripts may not work in PHP 5 without some rewriting. *PHP 5 Power Programming* is an excellent book for PHP 4 developers in need of a PHP 5 introduction. It's also a good book for anyone proficient in another programming language, such as Java, Perl or Python, who now wants to get started with PHP.

The book is co-authored by Andi Gutmans, Stig Bakken and Derick Rethans, three key contributors to the PHP language. They bring an intimate knowledge of the language to the book and provide anecdotal evidence as to why PHP has developed in the manner it has. Their writing style is clear, focused and enjoyable.

For PHP developers looking for a PHP 5 transition guide, this book works perfectly. The authors are candid about what they've broken in the transition from PHP 4 to PHP 5. It doesn't stop there, either; coverage of the new PHP 5 object model is excellent. Some PHP developers may not understand the usefulness of new OO concepts introduced in PHP 5, so the authors included a chapter on applying OO design patterns to PHP.

PHP and MySQL go together like peanut butter and jelly. The improved MySQL libraries for PHP further cement this relationship. PHP 5 introduces native support for SQLite, a powerful database option for PHP developers without access to another database.

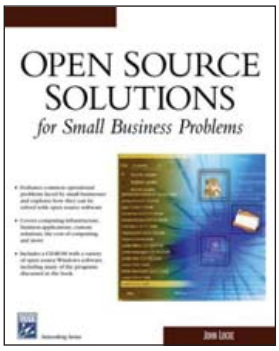
This book belongs on the desk of anyone considering a move to PHP 5. It serves as a road map for upgrading to the latest incarnation of PHP and as a reference for anyone who wants to expand his or her PHP object-oriented design skills. My copy already has a dozen or so sticky notes marking important sections and twice as many dog-eared pages. It has been an invaluable resource in my exploration of PHP 5.

—CHRIS MCAVOY

Open Source Solutions for Small Business Problems

by John Locke

Charles River Media, 2004 | ISBN: 1-58450-320-3 | $39.95 US



Working for a number of small businesses, I have seen firsthand how Linux and open-source software can be used to solve specific problems. It is great to see a good book detailing open-source solutions for small businesses. John Locke takes an excellent approach to this subject by addressing both the business manager who must decide what solutions to implement and the IT administrator who must implement those solutions.

Locke covers all of the software you need for your small business, including e-mail, customer relationship management, finance and disaster recovery. Each chapter provides valuable background information aimed at helping the nontechnical reader understand both the problem and the solution, as well as the details necessary for an intermediate Linux or Microsoft Windows administrator to implement the solution. Locke wisely chooses software that has the features you need, as well as strong community support. He recommends Postfix for e-mail because of its security, performance and feature set. He also recommends RetrieverCRM for customer relationship management and SQL-Ledger for financial management. Most of the solutions Locke presents will run on Windows as well as Linux, for an easy transition into the open-source world.

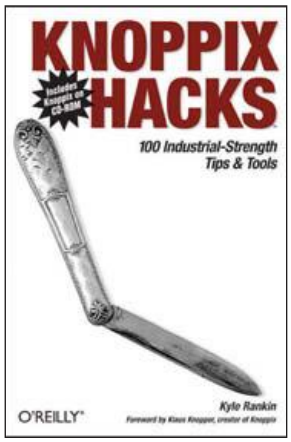
Although Locke provides good instructions on how to implement these solutions, there is not enough room in his book to provide all of the details you may need. For this reason he provides many references at the end of each chapter, pointing you to books, articles and Web sites that can provide the details you need. Written for a beginning to intermediate user, Locke does a great job of keeping the chapters simple and easy to follow.

—STEPHEN HAYWOOD

Knoppix Hacks: 100 Industrial-Strength Tips & Tools

by Kyle Rankin

O'Reilly & Associates, 2004 | ISBN: 0-596-00787-6 | $29.95 US



As distribution innovations go, Knoppix is a revelation. A bootable CD that provides a completely self-contained and fully functional desktop? All that and it leaves my hard drive untouched? What Klaus Knopper has wrought with Knoppix is all of that and much more. So much more, in fact, that even an experienced Knoppix user may not have discovered everything the compressed CD offers.

I received my first Knoppix CD, the German edition, from Volker Lendecke. Although my limited German language facility made sampling that CD a challenge, I marveled nonetheless as each application launched.

Today, because of the power and flexibility of Knoppix, like many other people, I burn multiple copies of each new release: one for my own use and the rest to give away. Just as giving away Knoppix CDs fits neatly into my advocacy agenda, *Knoppix Hacks* by Kyle Rankin fits into the O'Reilly catalog as another excellent book. Rankin formally documents what makes Knoppix and its derivatives such important tools for systems professionals.

This well-written book offers a broad range of descriptions and advice about using the capabilities of Knoppix. These are presented as a steady progression of logically grouped hacks. This book is a pleasure to read cover to cover, but it is as easy to use for an individual hack too.

The range of the hacks presented is as impressive as the contents of Knoppix itself, including: boot-time cheat codes, desktop applications, different network-related tools, software RAID and troubleshooting. Steps to remaster Knoppix to create custom derivatives also are discussed. There is no unimportant filler.

The majority of the hacks presented is not completely accessible to the beginner, but adding the required content to do so would so encumber this book such that it would cease to be useful for the experienced user, who is clearly the target for this book.

If you have not experienced Knoppix and cannot download it easily for yourself, then by all means let Kyle Rankin be your Knoppix-sharing friend. Read *Knoppix Hacks* and explore the included Knoppix CD for yourself. If you already have experienced Knoppix, you should find enough useful hacks among the 100 presented in this book to warrant its purchase.

—JEFFREY BIANCHINE

Ultra Dense, Powerful, Reliable...
Datacenter Management Simplified!
15" Deep, 2-Xeon/Opteron or P4 (w/RAID) options

Customized Solutions for...
Linux, BSD, W2K

High Performance Networking Solutions
- Data Center Management
- Application Clustering
- Network and Storage Engines

Rackmount Server Products
- **1U Starting at $499**: C3-1GHz, LAN, 256MB, 20GB IDE
- 2U with 16 Blades, Fast Deployment & more...

iron SYSTEMS™
Iron Systems, Inc.
2330 Kruse Drive, San Jose, CA
www.ironsystems.com

CALL: 1-800-921-IRON

Reading File Metadata with extract and libextractor

Don't just guess about a file's characteristics in a search. Use specific extractor plugins to build an accurate database of files.

BY CHRISTIAN GROTHOFF

Modern file formats have provisions to annotate the contents of the file with descriptive information. This development is driven by the need to find a better way to organize data than merely by using filenames. The problem with such metadata is it is not stored in a standardized manner across different file formats. This makes it difficult for format-agnostic tools, such as file managers or file-sharing applications, to make use of the information. It also results in a plethora of format-specific tools used to extract the metadata, such as AVInfo, id3edit, jpeginfo and Vocoditor.

In this article, the libextractor library and the extract tool are introduced. The goal of the libextractor Project is to provide a uniform interface for obtaining metadata from different file formats. libextractor currently is used by evidence, the file manager for the forthcoming version of Enlightenment, as well as for GNUnet, an anonymous, censorship-resistant peer-to-peer file-sharing system. The extract tool is a command-line interface to the library. libextractor is licensed under the GNU General Public License.

libextractor shares some similarities with the popular **file** tool, which uses the first bytes in a file to guess the MIME type. libextractor differs from file in that it tries to obtain much more information than the MIME type. Depending on the file format, libextractor can obtain additional information, including the name of the software used to create the file, the author, descriptions, album titles, image dimensions or the duration of a movie.

libextractor achieves this information by using specific parser code for many popular formats. The list currently includes MP3, Ogg, Real Media, MPEG, RIFF (avi), GIF, JPEG, PNG, TIFF, HTML, PDF, PostScript, Zip, OpenOffice.org, StarOffice, Microsoft Office, tar, DVI, man, Deb, elf, RPM, asf, as well as generic methods such as MIME-type detection. Many other formats exist, and among the more popular formats only a few proprietary formats are not supported.

Integrating support for new formats is easy, because libextractor uses plugins to gather data. libextractor plugins are shared libraries that typically provide code to parse one particular format. At the end of this article, we demonstrate how to integrate support for new formats into the library. libextractor gathers the metadata obtained from various plugins and provides clients with a list of pairs, consisting of a classification and a character sequence. The classification is used to organize the metadata into categories such as title, creator, subject and description.

Installing libextractor and Using extract

The simplest way to install libextractor is to use one of the binary packages available for many distributions. Under Debian, the extract tool is in a separate package, extract. Headers required to compile other applications against libextractor are contained in libextractor0-devel. If you want to compile libextractor from source, you need an unusual amount of memory: 256MB of system memory is roughly the minimum, as GCC uses about 200MB to compile one of the plugins. Otherwise, compiling by hand follows the usual sequence of steps, as shown in Listing 1.

After installing libextractor, the extract tool can be used to obtain metadata from documents. By default, the extract tool uses a canonical set of plugins, which consists of all file-format-specific plugins supported by the current version of libextractor, together with the MIME-type detection plugin. Example output for the *Linux Journal* Web site is shown in Listing 2.

If you are a user of BibTeX, the option -b is likely to come in handy to create BibTeX entries automatically from documents that have been equipped properly with metadata, as shown in Listing 3.

Another interesting option is -B LANG. This option loads one of the language-specific but format-agnostic plugins. These plugins attempt to find plain text in a document by matching

Listing 1. Compiling libextractor requires about 200MB of memory.

```
$ wget http://ovmj.org/libextractor/
➥download/libextractor-0.4.1.tar.gz
$ tar xvfz libextractor-0.4.1.tar.gz
$ cd libextractor-0.4.1
$ ./configure --prefix=/usr/local
$ make
# make install
```

Listing 2. Extracting metadata from HTML.

```
$ wget -q http://www.linuxjournal.com/
$ extract index.html
description - The Monthly Magazine of the Linux Community
keywords - linux, linux journal, magazine
```

Listing 3. Creating BibTeX entries can be trivial if the documents come with plenty of metadata.

```
$ wget -q http://www.copyright.gov/legislation/dmca.pdf
$ extract -b ~/dmca.pdf
% BiBTeX file
@misc{ unite2001the_d,
     title = "The Digital Millennium Copyright Act
     of 1998",
     author = "United States Copyright Office - jmf",
     note = "digital millennium copyright act
     circumvention technological protection management
     information online service provider liability
     limitation computer maintenance competition
     repair ephemeral recording webcasting distance
     education study vessel hull",
     year = "2001",
     month = "10",
     key = "Copyright Office Summary of the DMCA",
     pages = "18"
}
```

Listing 4. libextractor can sometimes obtain useful information even if the format is unknown.

```
$ wget -q http://www.bayern.de/HDBG/polges.doc
$ extract -B de polges.doc | head -n 4
unknown - FEE Politische Geschichte Bayerns
Herausgegeben vom Haus der Geschichte als Heft
der zur Geschichte und Kultur Redaktion Manfred
Bearbeitung Otto Copyright Haus der Geschichte
München Gestaltung fürs Internet Rudolf Inhalt im.
unknown - und das Deutsche Reich.
unknown - und seine.
unknown - Henker im Zeitalter von Reformation und Gegenreformation.
```

strings in the document against a dictionary. If the need for 200MB of memory to compile libextractor seems mysterious, the answer lies in these plugins. In order to perform a fast dictionary search, a bloomfilter is created that allows fast probabilistic matching; GCC finds the resulting data structure a bit hard to swallow.

The option -B is useful for formats that currently are undocumented or unsupported. The printable plugins typically print the entire text of the document in order. Listing 4 shows the output of extract run on a Microsoft Word document.

This is a rather precise description of the text for a German speaker. The supported languages at the moment are Danish (da), German (de), English (en), Spanish (es), Italian (it) and Norwegian (no). Supporting other languages merely is a question of adding free dictionaries in an appropriate character set. Further options are described in the extract man page; see `man 1 extract`.

Using libextractor in Your Projects

Listing 5 shows the code of a minimalistic program that uses libextractor. Compiling minimal.c requires passing the option -lextractor to GCC. The EXTRACTOR_KeywordList is a simple linked list containing a keyword and a keyword type. For details and additional functions for loading plugins and manipulating the keyword list, see the libextractor man page,

Integrating support for new formats is easy, because libextractor uses plugins to gather data.

Listing 5. minimal.c shows the most important libextractor functions in concert.

```
#include <extractor.h>
int main(int argc, char * argv[]) {
  EXTRACTOR_ExtractorList * plugins;
  EXTRACTOR_KeywordList   * md_list;
  plugins = EXTRACTOR_loadDefaultLibraries();
  md_list = EXTRACTOR_getKeywords(plugins, argv[1]);
  EXTRACTOR_printKeywords(stdout, md_list);
  EXTRACTOR_freeKeywords(md_list);
  EXTRACTOR_removeAll(plugins); /* unload plugins */
}
```

`man 3 libextractor`. Java programmers should know that a Java class that uses JNI to communicate with libextractor also is available.

Writing Plugins

The most complicated thing about writing a new plugin for libextractor is writing the actual parser for a specific format. Nevertheless, the basic pattern is always the same. The plugin library must be called libextractor_XXX.so, where XXX denotes the file format of the plugin. The library must export a method libextractor_XXX_extract, with the following signature shown in Listing 6.

Listing 6. Signature of the function that each libextractor plugin must export.

```
struct EXTRACTOR_Keywords *
libextractor_XXX_extract
   (char * filename,
    char * data,
    size_t size,
    struct EXTRACTOR_Keywords * prev);
```

The argument filename specifies the name of the file being processed. data is a pointer to the typically mmapped contents of the file, and size is the file size. Most plugins do not make use of the filename and simply parse data directly, starting by verifying that the header of the data matches the specific format.

prev is the list of keywords extracted so far by other plugins for the file. The function is expected to return an updated list of keywords. If the format does not match the expectations of the plugin, prev is returned. Most plugins use a function such as addKeyword (Listing 7) to extend the list.

A typical use of addKeyword is to add the MIME type

Listing 7. The plugins return the metadata using a simple linked list.

```
static void addKeyword
   (struct EXTRACTOR_Keywords ** list,
    char * keyword,
    EXTRACTOR_KeywordType type)
{
  EXTRACTOR_KeywordList * next;
  next = malloc(sizeof(EXTRACTOR_KeywordList));
  next->next = *list;
  next->keyword = keyword;
  next->keywordType = type;
  *list = next;
}
```

Listing 8. jpegextractor.c adds the MIME type to the list after parsing the file header.

```
if ( (data[0] != 0xFF) || (data[1] != 0xD8) )
  return prev; /* not a JPEG */
addKeyword(&prev,
           strdup("image/jpeg"),
           EXTRACTOR_MIMETYPE);
/* ... more parsing code here ... */
return prev;
```

once the file format has been established. For example, the JPEG-extractor (Listing 8) checks the first bytes of the JPEG header and then either aborts or claims the file to be a JPEG. The strdup in the code is important, because the string will be deallocated later, typically in EXTRACTOR_freeKeywords(). A list of supported keyword classifications, in the example EXTRACTOR_MIMETYPE can be found in the extractor.h header file.

Conclusion

libextractor is a simple extensible C library for obtaining metadata from documents. Its plugin architecture and broad support for formats set it apart from format-specific tools. The design is limited by the fact that libextractor cannot be used to update metadata, which more specialized tools typically support.

Resources for this article: www.linuxjournal.com/article/8207.

Christian Grothoff graduated from the University of Wuppertal in 2000 with a degree in mathematics. He currently is a PhD student in computer science at Purdue University, studying static program analysis and secure peer-to-peer networking. A Linux user since 1995, he has contributed to various free software projects and now is the maintainer of GNUnet and a member of the core team for libextractor. His home page can be found at grothoff.org/christian.

Converting e-Books to Open Formats

E-books are a disappointing flurry of vendor-specific formats. Get them converted to HTML to view on your choice of device. BY MARCO FIORETTI

Books in digital format, also known as e-books, can be read on devices lacking the power and screen space to afford a regular Web browser. Several publishers, not to mention projects such as Project Gutenberg, have provided thousands of new and classic titles in digital format. The problem is both the hardware—be it generic PDAs or dedicated devices—and the whole e-book publishing industry are much more fragmented than are PCs and Web browsers. Therefore, it is probable that the e-book you recently bought will not be readable ten years from now—nor tomorrow, should you decide to use a laptop or change PDAs. To help combat this fragmentation, this article discusses some existing command-line tools that can convert the most popular e-book formats to ASCII or HTML.

Practically no tools exist now to export e-book formats to PDF or OpenDocument, the new OASIS standard used in OpenOffice.org, but this is not necessarily a big deal. Once text is in ASCII or HTML format, it easily can be moved to plain-text or PDF format by using a text browser such as w3m or programs such as html2ps. If you go this route for conversion, you are able to do it today, and because it's an open format, 20 years from now too.

PalmDoc

On PalmOS, the original and most common e-book format is PalmDoc, also called AportisDoc or simply Doc, even though it has nothing to do with Microsoft Word's .doc format. Doc, recognizable by the extensions .pdb (Palm Database) or .prc (Palm Resource Code), basically is a PalmPilot database composed of records strung together. This standard has spun off several variants, including MobiPocket, which adds embedded HTML markup tags to the basic format.

Each Palm e-book is divided into three sections: the header, a series of text records and a series of bookmark records. Normally, the header is 16 bytes wide. Some Doc readers may extend the width at run time to hold additional custom information. By default, the header contains data such as the total length of the uncompressed text, the position currently viewed in the document and an array of two-byte unsigned integers giving the uncompressed size of each text record. Usually, the maximum size for this kind of record is 4,096 bytes, and each one of them is compressed individually.

The bookmark records are composed of a 16-byte name and a 4-byte offset from the beginning of text. Because bookmarks are optional, many Doc e-books don't contain them, and most Doc readers support alternative—that is, non-portable—methods to specify them. Other reader-specific extensions might include category, version numbers and links between e-books. Almost always, this information is stored outside the .pdb or .rc file. Therefore, you should not expect to preserve this kind of data when converting your e-books.

Pyrite Publisher, formerly Doc Toolkit, is a set of content conversion tools for the Palm platform. Currently, only some text formats can be converted, but functionality can be extended to support new ones by way of Python plugins. Pyrite Publisher can download the documents to convert directly from the Web; it also can download set bookmarks directly to the output database. The package, which requires Python 2.1 or greater, can be used from the command line or through a wxWindows-based GUI. The software is available for Linux and Windows in both source and binary format. Should you choose the latter option, remember that compiled versions expect Python to be in /usr. The Linux version can install converted files straight to the PDA using JPilot or pilot-link.

Pyrite installed and ran flawlessly on Fedora Core 2. Unlike the other command-line converters presented below, however, Pyrite can save only in ASCII format, not in HTML. The name of the executable is pyrpub. The exact command for converting .pdb files uses this syntax:

```
pyrpub -P TextOutput -o don_quixote.txt \
Don_Quixote.pdb
```

Pyrite can be enough if all you want to do is quickly index a digital library. On the other hand, it is almost trivial to reformat the result to make it more readable in a browser. The snippet of Perl code in Listing 1, albeit ugly, was all it took to produce the version of *Don Quixote* shown in Figure 1.

The script loads the whole ASCII text previously generated with Publisher, and every time it finds two new lines in a row, it replaces them with HTML paragraph markers. The result then is printed to standard output and properly formatted as basic HTML. To change justification, fonts and colors, you simply need to paste your favourite stylesheet right after the `<html><body>` line.

OpenOffice.org 2.0, expected to be released in spring 2005, will be able to save text in .pdb format. If it also is able to read such files, its mass conversion feature (File→AutoPilot→Document Converter) would solve the problem nicely. I have tried to do this with the 1.9.m65 preview, but all I got was a `General input/output error` pop-up message. Hopefully, this functionality will be added to future versions.

The P5 Perl Package

Pyrite Publisher is designed mainly to go from normal HTML or text files to the Palm platform, not the other way around. The procedure discussed above is not really scalable to scenarios such as converting a great quantity of Palm e-books to customized HTML, with hyperlinks and metadata included. In such cases, the best solution might be a Perl script combining the standard XML

Listing 1. A simple Perl script converts Pyrite's extracted text to HTML.

```
#! /usr/bin/perl

undef $/;

$TEXT  = <>;
$TEXT =~ s/\n\n/<p>/gm;

print <<END_HTML;
<html><body>
$TEXT
</body></html>
END_HTML
```

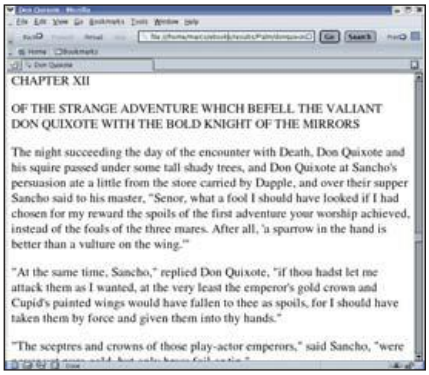


Figure 1. A PalmDoc file converted to HTML for viewing in a browser.

or HTML modules for this language with the P5-Palm bundle; these are available from the Comprehensive Perl Archive Network (see the on-line Resources). The P5-Palm set of modules includes classes for reading, processing and writing the .pdb and .prc database files used by PalmOS devices.

Rocket Ebook and MobiPocket

RocketBook e-books have several interesting characteristics, including support for compressed HTML files and indexes containing a summary of paragraph formatting and the position of the anchor names. These and many more details on .rb file internals are explained in the RB format page listed in the on-line Resources. Rbmake Rocket Ebook and Mobipocket files can be disassembled with a set of command-line tools called Rbmake. Its home page offers source code, binary packages, a mailing list and contact information to report bugs. To use rbmake, you need libxml2, version 2.3.1 or higher; the pcre (Perl-Compatible Regular Expressions) library; and zlib, to handle compression. To compile from source—at least on Fedora Core 2—it also is necessary to install the pcre-devel package separately.

The Rbmake Library

A nice feature of Rbmake is the source code is structured in a modular manner. An entire library of object-oriented C routines can be compiled and linked independently from the rest of the package from any other program dealing with .rb files. In this way, should you want to write your own super-customized Rocket Ebook converter or simply index all of your e-books into a database, you would need to use only the piece that actually knows how to read and write the .rb format, the RbFile class. This chunk of code opens the file, returns a list of the sections composing the book and uncompresses on the fly only the ones actually required by the main program. Should you need them, the library also includes functions to match and replace parts of the content through Perl-compatible regular expressions.

The Rbmake tools should compile quickly and without problems on any modern GNU/Linux distribution. Exhaustive HTML documentation also is included in the source tarball. The binary file able to generate HTML files is called rbburst. It extracts all the components—text, images and an info file—present in the original .rb container. Figure 2 shows, in two separate Mozilla Windows, the cover page and the table of contents of the file generated by rbburst when run on *The Invisible Man* by H. G. Wells.

Microsoft Reader

Microsoft's Reader files, recognizable by the .lit extension, have many of the characteristics of traditional books, including pagination, highlighting and notes. They also support keyword searching and hyperlinks, but they are locked in to one reader platform.

The tool for converting these files is called, simply, Convert Lit. Running the program with the -help option lists, according to UNIX tradition, all the available command-line options. This program has three modes of operation: explosion, downconversion and inscribing. Explosion is the one needed to convert an existing .lit file to an OEBPS-compliant package. OEBPS (Open eBook Publication Structure) is covered later in the article.

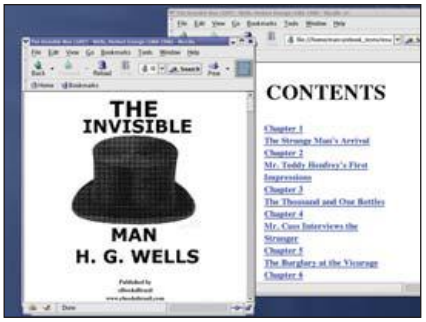


Figure 2. Rbmake extracts all the components of a RocketBook file, including text and images.

Figure 3 shows a version of Shakespeare's *A Midsummer's Night Dream* obtained by using explosion from the Convert Lit program.

Downconversion is the opposite process; it generates a .lit file for use by a Microsoft Reader-compliant device. Inscribing is when the downconversion attaches a user-defined label to the .lit file. The exact syntax is explained on the program's home page (see Resources).

We already mentioned that Convert Lit creates an OEBPS package made of different files. Here is the complete list for the example above: Contents.htm, copyright.html, ~cov0024.htm, cover.jpg, MidSummerNightDream.opf, MobMids.html, PCcover.jpg, PCthumb.jpg, stylesheet.css and thumb.jpg. HTML, CSS and JPG files were to be expected, but what is the .opf file? It is an XML container describing the structure and several portions of the original book's metadata. The extension OPF stands for open electronic book package format. The OPF file contains references to the other pieces of the e-book, as well as descriptions of their attributes. To have a clearer idea of its role, a short excerpt of MidSummerNightDream.opf is shown in Listing 2.

The practical consequence of this is

Figure 3. Convert Lit creates a readable HTML file with a hyperlinked table of contents.

Listing 2. OPF is an XML-based format for book attributes.

```
<dc:Title>A Midsummer-Night's Dream</dc:Title>
<dc:Creator role="aut"
    file-as="Shakespeare, William, 1564-1616">
        William Shakespeare, 1564-1616
</dc:Creator>
<dc:Description>fiction, poetry</dc:Description>
```

Convert Lit could be useful even if you wanted to leave all of your collection in a proprietary format. You still could run the program on all your .lit e-books and delete everything but the .opf files. Then, any quick script or full-blown XML parsing utility could scan them and index everything into the database of your choice.

Convert Lit also removes digital rights management (DRM) infections from e-book files using the older DRM1 version. And if you have Microsoft Reader e-books, you likely have a Microsoft Windows system and a licensed copy of Microsoft Reader. According to the Convert Lit Web site, you can build and run Convert Lit on Windows to first convert new DRM5 e-books to DRM1, using the Windows DRM library.

Mass Conversion

In general, we have discussed only command-line processing in this article. If, however, you have a whole collection of e-books in different formats, you can convert them all at one time with a simple shell script. As we already have shown, once the text is in ASCII or HTML format, the sky is the limit. You can add one or two lines to the loop to index with `glimpse` or `ht::dig`, print everything in one single PostScript book and much more.

OEBPS

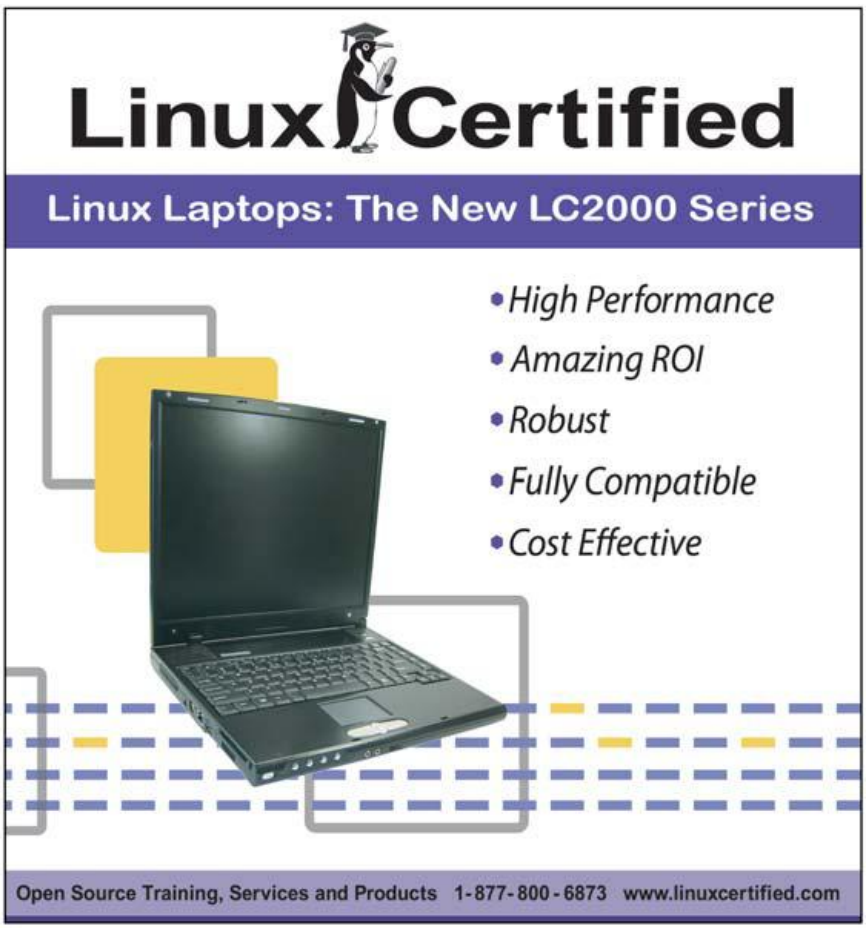
A solution for putting e-books, at least the ones you will be able to get in the near future, into an open format is in the works. It is the Open eBook Publication Structure (OEBPS). Its goal is to provide an XML-based specification, based on existing open standards, for providing content to multiple e-book platforms. OEBPS, which has reached version 1.2, is maintained by the Open eBook Forum, a group of more than 85 organizations—hardware and software companies, publishers, authors and users—involved in electronic publishing. OEBPS itself does not directly address DRM. However, an OeBF Rights and Rules Working Group is studying these issues "to provide the electronic publishing community with a consistent and mutually supporting set of specifications". Time will tell what will come from this.

In any case, the open standards on which OEBPS is built already are well established. Besides XML, Unicode, XHTML and selected parts of the CSS1 and CSS2 specifications are represented. Unicode is a family of encodings that enables computers to handle without ambiguity tens of thousands of characters. XHTML is the reformulation of HTML 4 as XML. In a nutshell, OEBPS could be described as nothing more than an e-book optimized extension of XHTML—something that won't go away when some company goes out of business. Graphics can be in PNG or JPEG formats. Metadata, including author, title, ISBN and so on, will be managed through the Dublin Core vocabulary.

OEBPS has the potential to preserve all your e-books and make sure that the ones you download or buy will not vanish if any hardware or software company goes the way of the dodo. However, DRM schemes applied on top of these "open" e-books still could lock your content in to one vendor. As long as you can obtain OEBPS e-books without DRM, OEBPS is the best way to guarantee that even if all current e-book hardware disappeared, your collection would remain usable.

Resources for this article: www.linuxjournal.com/article/8208.

Marco Fioretti is a hardware systems engineer interested in free software both as an EDA platform and, as the current leader of the RULE Project, as an efficient desktop. Marco lives with his family in Rome, Italy.



One-Click Release Management

How a large development project on a tight release schedule and a tight budget can use open source to tackle the problems of version control and release management. BY JAKE DAVIS

Say you have a large piece of software, a complicated Web site or a whole bunch of little ones. You also have a gaggle of coders and a farm of machines on which to deploy the end product. Worst of all, the client insists on a short turnaround time for critical changes. Proprietary products that may provide you with a systematic, unified development, testing and deployment process typically are expensive and offer limited deployment options. They often require new hardware resources and software licenses simply to support installation of the system itself. Such a solution can be difficult to sell to managers who are concerned about cost and to developers who are concerned about learning a new and complicated process.

However, managing the development process from end to end on a tight schedule without such a unified approach can lead to serious inefficiencies, schedule slippage and, in general, big headaches. If you're the administrator of such a project, chances are you're spending a lot of time dealing with the management of code releases. On the other hand, you already may be using an expensive piece of proprietary software that solves all of your problems today, but the higher-ups are balking at the ever-increasing license renewal fees. You need to present them with an alternative. It's also possible that you release code only once a year and have more free time than you know what to do with, but you happen to be an automation junkie. If any of these scenarios seem familiar to you, read on.

The Solution

Various open-source products can be adapted to minimize costs and developer frustration while taming your out-of-control release process by serving as the glue between existing toolsets. Maybe you even can start making it home in time to play a round or two of *Scrabble* before bedtime.

I base the examples presented in this article on a few assumptions that hopefully are common or generic enough that the principles can be extrapolated easily to fit with the particulars of a real environment. Our developers probably already use a bug-tracking system (BTS), such as Bugzilla, ClearQuest or Mantis, or an in-house database solution to track change requests and bugs. They also may be using a version control system (VCS), such as Arch, CVS or Subversion, to manage the actual code changes called for in various BTS entries.

If they're not using a BTS and a VCS for a large project, these developers probably have either superhuman organization skills or a high level of tolerance for emotional trauma. Which BTS and VCS we use is not essential to this discussion, and any exploration of the pros and cons between one system and another requires much more text than I am allotted here. In short, they all should support the building blocks needed for the type of process we'd like to employ. Namely, most any BTS can:

1. Assign a unique ID to all issues or bugs in its database.

2. Allow you to use the unique ID to track the state of an issue and store and retrieve a listing of any source files it affects.

Any VCS worth its salt (sorry VSS fans) can:

1. Allow some form of branching and merging of a central code hierarchy.

2. Allow a command-line client process to connect over a secure network connection in order to perform updates.

We use a Subversion (SVN) repository with the SVN+SSH access method enabled as our VCS and a generic MySQL database table as the BTS. We use Python, which tends to be quite readable even for the novice programmer, as our scripting language of choice. Chances are your distribution has packages for all of these products readily available; configuring them will be left as an exercise for the reader. The target machines are generic Web servers, all of which support SSH connections as well as the VCS client tools.

Here's the 10,000-foot overview of the example end-to-end process we are likely to be starting out with:

1. An issue is generated in the BTS and is assigned an ID of 001 and an initial status of "new". It includes, or will include, a listing of file paths that represent new or changed files within the VCS repository and is assigned to the appropriate developer.

2. The assignee developer makes changes to his local copy of the source code, checks these changes into the VCS repository and updates the status of BTS ID# 001 to "in testing".

3. The testing server is updated with the new changes.

4. A QA tester charged with reviewing all BTS items with a status of "in testing" verifies that the changes to the code are what is desired and updates the status of BTS ID 001 to

"ready for production".

5. A release manager then packages all changes affected by BTS ID# 001 into a release and updates the status of BTS ID# 001 to "in production".

6. The live server is updated with the changes.

For the most part, we're managing to fix bugs and add new features to the code base without bugging the system administrator for much, aside from the occasional password reset or RAM upgrade. But steps 3 and 6 require us somehow to get the code out of the VCS and onto a live system. We could cut and paste files from the VCS into a folder on our hard drive, zip it up, send it to the sysadmin and ask him to unzip it on the live system. Or, we could take advantage of the structure of our VCS and its utilities to do the work for us and completely avoid having a conversation with the administrator, whose time tends to be a hot commodity.

The Nuts and Bolts

If we structured our VCS to encompass a branching scheme that mirrors our various statuses in the BTS, we likely would end up with a BRANCH to which developers add new, untested changes and a TRUNK that includes only code that is "in production", although it easily could be the other way around. It then becomes a relatively simple matter of using the branch merging capabilities of the VCS to move "ready for production" code from the testing BRANCH to the stable TRUNK. Because no development changes happen on our TRUNK, merging from BRANCH to TRUNK is not likely to cause any conflicts. Managing the last step of moving the code from the VCS to the live system becomes even easier, because updating simply is a matter of using the VCS client utility to pull down all changes that occurred on the TRUNK of the repository.

So now all the pieces are there to allow quick and accurate code deployment, but we still need to ask our sysadmin to run the VCS client tools on the live system. We further can minimize our demands on the sysadmin's time, however, if he or she is willing to give our release manager an SSH login with permission to run the VCS client on the live system.

Expanding the Model to Enable Automated Releases

Once we've got the infrastructure in place to support performing content updates by way of our VCS, the next logical step is to remove further the need for manual intervention at release time. It now is possible for us to create a script that can use the VCS client tools to pull code updates to a live system. This method increases its usefulness as the number of machines we need to update increases. If our script has access to a list of all the target machines that need to be updated, we can hit them all in one fell swoop.

This piece of the puzzle, like the example, can be a simple script that the release manager runs from the command line of his workstation. Or, it can be a fancy Web-based GUI that a team of release managers can use to update any number of machines from any Web browser with a mouse click. In either case, it is useful to create a user ID on the client machines that has permissions to connect back to the VCS system without being prompted for login information. This may require configuring the user account on the client machines with SSH keys that allow it to connect back to the VCS server.

With this script in place on the client machines, we can update client copies of VCS files from a central location over an encrypted SSH connection.

Listing 1. vcs_update.py

```
#!/usr/bin/env python

import os, sys

clientList = ['host1', 'host2', 'host3']
sandbox = "/usr/local/www"

def updateClient(client, sandbox):
    # ssh to client machines and update sandbox
    command_line = "ssh %s svn update %s"%(client,
                                           sandbox)
    output = os.popen4(command_line)[1].readlines()
    for line in output:
        print line

if __name__=="__main__":
    for client in clientList:
        updateCLient(client, sandbox)
```

Spreading the Love

Now we have a reasonably efficient process that piggybacks almost seamlessly onto a process that our developers were, for the most part, already using. It also allows content updates with the click of a button. So what's stopping us from scripting the updates to the testing servers so that they happen automatically at regular intervals, allowing developers the chance to see their changes show up on a live test system without asking for an update? All we need to do is run the client script on the testing servers as a cron job.

Also, as long as we're asking crazy questions, why not take advantage of the power of our BTS' database back end to drive the whole process and really cut down on process management bottlenecks? To do so, our script generates a list of files that need to be merged between branches by running a query for all IDs with a status of "ready for production". The script uses the resulting lists as input for functions that perform the merge commands and update the BTS ID statuses to "in production" automatically.

Let's look at our amended 10,000-foot overview now that we've got all the bells and whistles incorporated:

1. An issue is generated in the BTS and assigned to the appropriate developer.

2. The assignee developer makes changes to his local copy of the source code, checks these changes into the TEST branch of the VCS repository and updates the status in the BTS.

Listing 2. bts_merge.py

```
#!/usr/bin/env python

import os, MySQLdb

TRUNK_WC = "/path/to/working_copy_of_trunk/"
TRUNK_URL = "svn+ssh://vcs-server/project/trunk/"
BRANCH_URL = "svn+ssh://vcs-server/project/branch/"

def initDB():
    # connect to database, return connection cursor
    connection = MySQLdb.connect(host='dbhost',
                                 db='dbname',
                                 user='user',
                                 passwd='password')
    cursor = connection.cursor()
    return connection, cursor

def listUpdatedFiles(cursor):
    # return updated file paths and BTS ids.
    cursor.execute("""SELECT changedfiles
                   FROM BugTable
                   WHERE status =
                   'ready for production'""")
    fileList = cursor.fetchall()
    cursor.execute("""SELECT bugID
                   FROM BugTable
                   WHERE status =
              -     'ready_for_production'""")
    idList = cursor.fetchall()
    return fileList, idList

def mergeUpdatedFiles(fileList):
    # merge branch changes into the trunk.
    for fileName in fileList:
        cmd = 'svn merge %s/%s %s/%s'%(BRANCH_URL,
                                       fileName,
                                       TRUNK_URL,
                                       fileName)
        for line in os.popen4(cmd)[1].readlines():
            print line

def updateBTSStatus(idList, cursor):
    # update BTS ids to 'in_production' status.
    for ID in idList:
        cursor.execute("""UPDATE BugTable
                       SET status = 'in_production'
                       WHERE bugID = %s""" % ID)

def stopDB(connection, cursor):
    # close the database connection
    cursor.close()
    connection.close()

if __name__=="__main__":
    os.chdir(TRUNK_WC)
    connection, cursor = initDB()
    fileList, idList = listUpdatedFiles(cursor)
    mergeUpdatedFiles(fileList)
    updateBTSStatus(idList, cursor)
    stopDB(connection, cursor)
```

3. The testing server content is updated automatically by a cron job.

4. A QA tester verifies that the changes to the code are correct and updates the status in the BTS.

5. A release manager presses a button to launch our merge script, which merges all changes into the stable TRUNK and updates the BTS.

6. One last click by the release manager, and the production systems are updated to the latest code by way of our VCS client script.

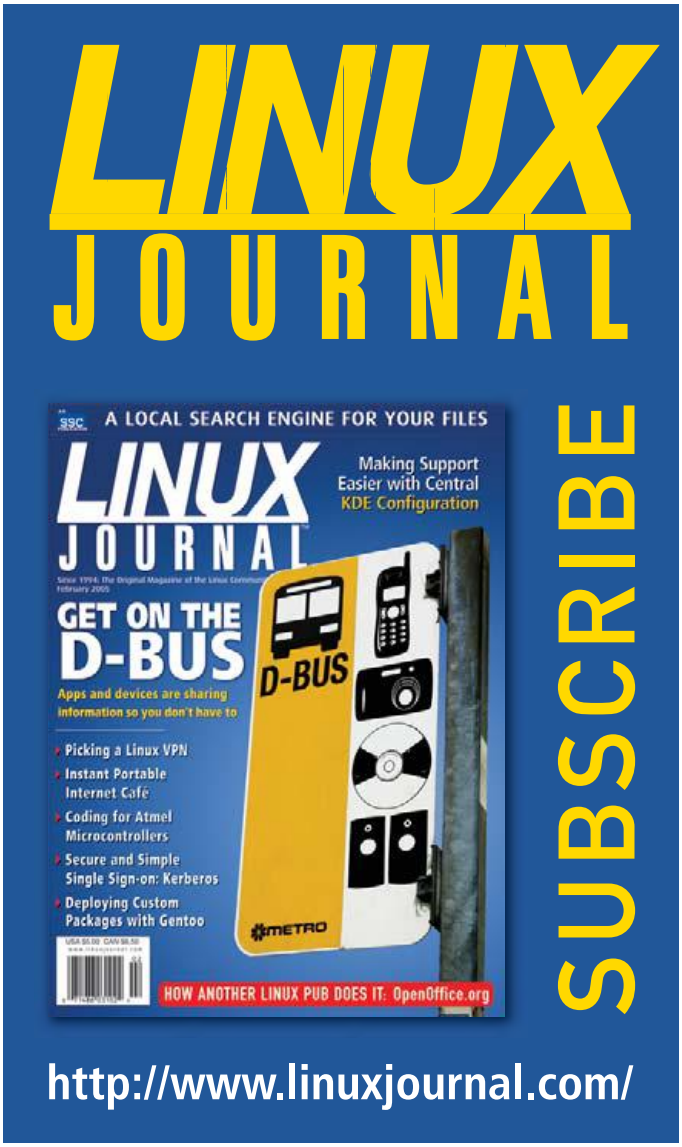
Steps 5 and 6 easily could be combined too, thereby halving the amount of work our release manager needs to perform.

Chances are at some point we'll want to add a staging branch to our VCS repository and enable our content update system to pull updates from this intermediate branch onto a staging server. QA then could see all the changes on a live system before the client does. Or, the client could be given access in order to provide final approval. Once staging has been given the thumbs up, moving updates to a production system is as easy as performing the already automated steps of merging from the staging branch to the stable TRUNK and running the content update script against the production servers.

Although these examples represent something of an oversimplification of the issues involved—for example, we haven't addressed the potential need for database structure updates—we have covered some core concepts that can be expanded on to build a truly functional, tailor-made system. In fact, we well may be approaching development process nirvana, and we still haven't spent dollar one on software licenses. Rather, we've simply written a few basic scripts to glue together our bug-tracking and version control systems. As a result, management now has more money in the reserve fund and fewer heart palpitations. Our sysadmins have more time to devote to removing spyware from desktops. Best of all, we've made it home for that round of *Scrabble* with time to spare. That's the power of open source for you.

Resources for this article: www.linuxjournal.com/article/8141.

Jake Davis (jake@imapenguin.com), IT consultant and self-described penguin, is cofounder of Imapenguin, LLC (**www.imapenguin.com**) an employer of waddling, flightless birds.



LOOK 8] Plug n' Play * Easy Admin
Audio Streaming Appliance, LIVE
GO TO wISPdirect.com
CALL 877-881-1954
SMALL AD,
BIG PRODUCT!

Why I Don't Worry about SCO, and Never Did

Lawyers can't shut down Linux now. Too many important people need it. **BY CHRIS DIBONA**

By the time this article goes to print and arrives in your mailbox, the SCO case will mean even less than it does when I'm writing this, and that's saying something. The latest headline associated with SCO is their potential delisting from the Nasdaq for failing to file paperwork. This is the public-company equivalent of being sent to bed without supper or being expelled from school. It isn't good, and it's very irresponsible.

By the time this magazine hits print they'll have sent in their homework, late, for a lesser grade, or they'll have retreated from the public markets and the expensive and revealing Sarbanes-Oxley scrutiny that comes with having a ticker symbol. Either way, they'll be even less of a threat to free software, but I have to say, I wasn't worried, not for one minute.

I wasn't worried about their legal position that they owned parts of the Linux kernel.

I wasn't worried about their complaints against friend of Linux, IBM.

I wasn't worried about the future of the Linux kernel, Linus himself, his wife, his kids, Alan Cox, Andrew Morton or, for that matter, the people in industry that SCO subpoenaed in pursuit of their action against IBM.

Why wasn't I worried? The time to sue Linux and many prominent open-source software projects has passed, and in that passing, we have a blueprint on how to avoid consequential litigation for future important free software projects. The reason I don't worry about people suing Linux, and the reason I wasn't worried when SCO did it, is because Linux has become too important to too many people for it to be vulnerable to that kind of attack.

The time to kill Linux was when it was a project with ten developers who lived on university stipends, not when it has thousands of connected developers and $14 billion in Linux-related sales (IDC's number for the year 2003, if you believe analysts). It was vulnerable when it was still a university project, not now when uncountable school districts are using it to reduce their dependence on the punitive cost structures of proprietary software. It was vulnerable when it was in use in a few countries by a few dozen users, not now when it is used by a few dozen countries to ensure their software sovereignty. In short, it was vulnerable when it meant nothing to a few, not now when it is central to the Information Age economies.

And if that hyperbole didn't make you turn the page and drool over an ad for some sexy cluster gear, here is what we learn from Linux and Litigation.

First, if you want to destroy a free software project's chances of success, start when it is young, by making a competing product so good there will be no user need for the free software.

Second, if you are running a large project, assemble your friends around you, the way Linux has, so you can absorb the hits that come with success. Surrounding Linux is a vast array of industry organizations, corporate users, nations and end users whose livelihoods are tied tightly to Linux. This doesn't mean that Linux doesn't get sued, it simply means the people doing the suing find themselves horribly alone.

Third, put off any sort of foundation or corporatization until you are ready to defeat the slings and arrows that come with the success of your project. The Samba team has played this card very well. If some large software company were to go after Samba under the rubric of patent infringement, who could it go after that would slow the use of Samba? Samba Project leaders Andrew Tridgell and Jeremy Allison would be protected by the same companies who find Samba vital to their survival. And this wouldn't stop people from using Samba one bit. Sometimes in the pages of *LJ* we talk about Microsoft as if it were filled with fools. But it's not so foolish as to sue its end users the way SCO has. The day for effectively suing Samba also has passed.

And finally, when people sue Linux or Samba, help if you can, but the best start is to keep using free software and help keep it vital to yourself, your workplace and your government. Through this need, this necessity, we enshroud our software with a warm blanket of security that the parasites will fail to penetrate.

So, in the end, this article is really about Asterisk, the free Voice-over-IP software that developed at telco hardware maker Digium. Asterisk represents a project that is on the verge of being too important to be vulnerable. If I were looking for an open-source company to back, I'd say (and have said) Digium. If you haven't tried it, you really should. It is remarkable stuff. Like Linux, you can't believe software can be this good and useful. And if it's good and useful, it will be important to enough people that legal threats from failed companies just won't matter.

Chris DiBona is the Open Source Program Manager for Mountain View, California-based Google, Inc. Before joining Google, Mr DiBona was an editor/author for the popular on-line Web site slashdot.org. He writes for a great number of publications, speaks internationally on software development and digital rights issues and co-edited the award-winning essay compilation *Open Sources*, whose sequel is planned for a July 2005 release.

routerOS: routerBOARD.com

User Management

- support more than 3000 PPPoE or Hotspot clients
  - full radius support for user parameters
  - tx/rx speed, address, filter rules
  - supports radius real time modification of parameters while users are online
- Peer to peer control (P2P)
  - burst time
  - per client P2P tx/rx rules
  - P2P pool
  - complete blocking of P2P

Wireless AP and Backbone

- Wireless monitoring
  - Frequency scanning with detailed report
  - Raw wireless packet sniffer
    - streaming option to Ethereal analyzer
    - option to save to a file format supported by Ethereal
  - Snooper packet inspection
    - analyzes all raw frames received for wireless parameters
    - monitor a single channel or all channels

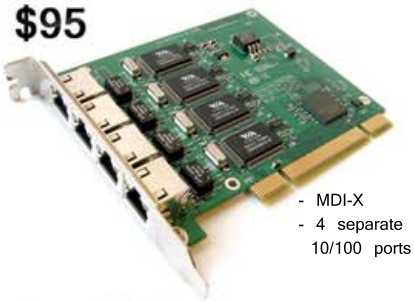
- Nstreme wireless polling protocol
  - no decrease in speed over long distances (as seen with the 802.11 ack packet bottleneck)
  - polling improves speed and eliminates contention for access to the wireless bandwidth
  - access point control over Nstreme clients tx data to optimize use of the wireless medium
  - radius support for the access control list including bandwidth settings for wireless clients

- Full 802.11a/b/g support

The above is a brief description of a few features, for more information and a fully featured 24 hour demo go to:

www.mikrotik.com

$95



- MDI-X
- 4 separate 10/100 ports

RouterBOARD 44

For the Router Builder !

- rackmount servers and routers
- up to 24 Ethernet ports in a PC
- no more straight/cross cable problems
- server quality VIA VT6105 chips

$195

RouterBOARD 230

No feature left behind !

Integrated router with various interfaces. Use as an AP on a tower with up to 500ft PoE. Includes IDE/CF, miniPCI, USB, PCMCIA, UART, PCI, GPIO, LCD controller, Linux SDK, and more.

$120 **$65**

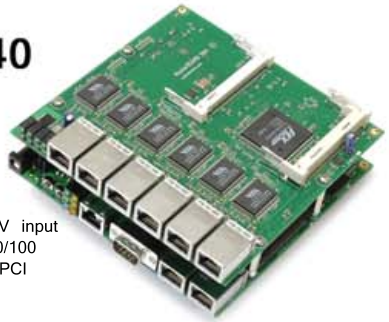
Eight ports Four ports

RouterBOARD 11/14/18

Multi radio tower !

MiniPCI to PCI adapters for multi radio system. Tested with sixteen radios in one Router/AP.

$240



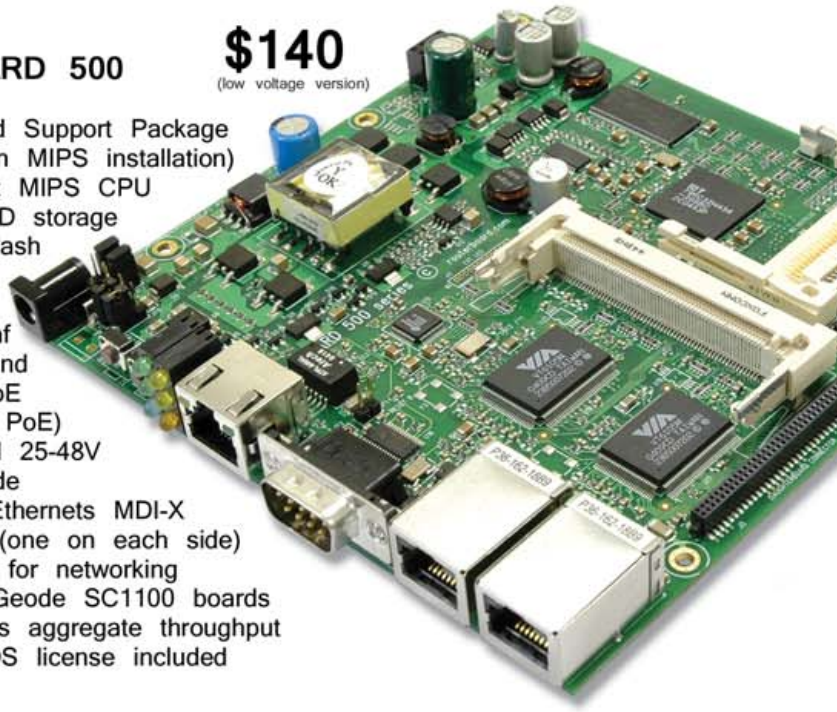
- PoE
- 10-56V input
- 9x 10/100
- 6x mPCI

RouterBOARD 500 & RouterBOARD 564

The Wireless Switchboard !

For a complete multi-radio tower system, the RouterBOARD 500 can carry a daughterboard (RouterBOARD 564) which adds six ethernets and four miniPCI.

RouterBOARD 500 **$140** (low voltage version)

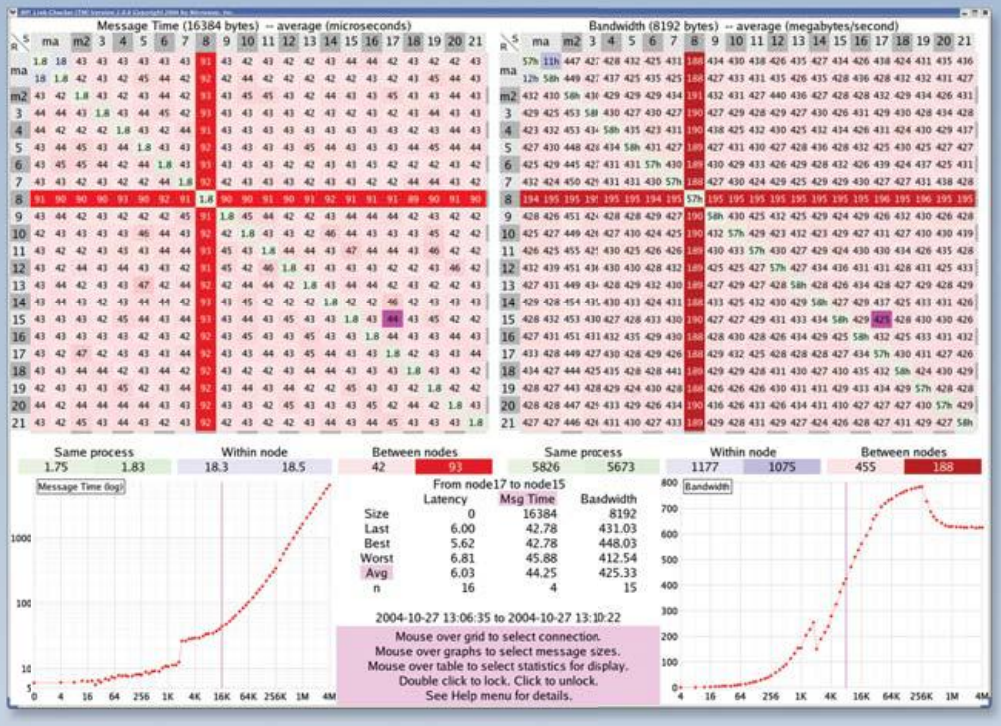


- Linux Board Support Package (full Debian MIPS installation)
- 266-400Mhz MIPS CPU
- 64MB NAND storage
- Compact Flash
- 32MB DDR
- Low power
- PoE 802.3af standard and passive PoE (also 12V PoE)
- 10-24V and 25-48V power mode
- 3 10/100 Ethernets MDI-X
- 2 miniPCI (one on each side)
- 2-3x faster for networking than the Geode SC1100 boards
- 200-300MB/s aggregate throughput
- L3 RouterOS license included

contact sales@routerboard.com or go to **www.routerboard.com**

all copyrights and logos belong to their respective owners

X Marks the Slow Node!



Microway® Quad Opteron™ Cluster with 36 Opteron 852s, redundant power and 45 hard drives in CoolRak™ cabinet.

MPI Link-Checker™ to the Rescue!

A single slow node or intermittent link can cut the speed of MPI applications by half. Whether you use GigE, Myrinet, Quadrix, InfiniBand or InfiniPath HTX, there is only one choice for monitoring and debugging your cluster of SMP nodes:

Microway's MPI Link-Checker™

Our unique diagnostic tool uses an end-to-end stress test to find problems with cables, processors, BIOS's, PCI buses, NIC's, switches, and even MPI itself! The newest release provides ancillary data on inter-process and intra-CPU latency which can vary by a factor of 10 between MPI versions. MPI Link-Checker is also useful for porting applications to new hardware. It provides instant details on how latency and bandwidth vary with packet size. It is available now for a free 30 day evaluation!

Wondering what's wrong with your cluster, or need help designing your next one? Call our HPC staff at 508-746-7341. Visit microway.com to learn about new low latency interconnects including the PathScale InfiniPath HTX Adapter, which delivers unmatched MPI latency of under 1.5 microseconds.

Microway has been an innovator in HPC since 1982. We have thousands of happy customers. Isn't it time you became one?

Call us first at 508-746-7341 for quotes and benchmarking services. Find technical information, testimonials, and newsletter at www.microway.com.

Excellence in Cluster Technology
64-bit Turnkey Solution
WINNER
ClusterWorld

Microway®
23 Years of Expertise Built In

PathScale™
Accelerating Cluster Performance™