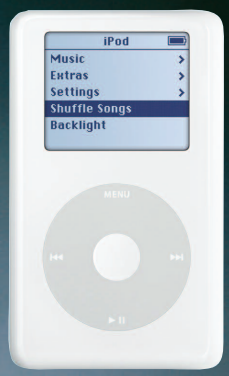
AN SSC PUBLICATION

TUNING UP APPLICATIONS FOR TODAY'S FASTEST HARDWARE

LINUX JOURNAL

Since 1994: The Original Magazine of the Linux Community
JULY 2005

MANAGING AN APPLE iPOD FROM A LINUX SYSTEM



Richard Naylor
CityLink
Wellington, New Zealand

CITY-WIDE ETHERNET

IP peering with cheap hardware puts Wellington a step ahead

BUSINESS APPLICATIONS WITH REKALL

Design cross-platform forms and reports with point-and-click ease

Saving screen space with tabbed terminals

LDAP directory for Linux and Microsoft clients

Fast drop-in firewall: a smart Ethernet bridge

Plugging Java applications in to the desktop

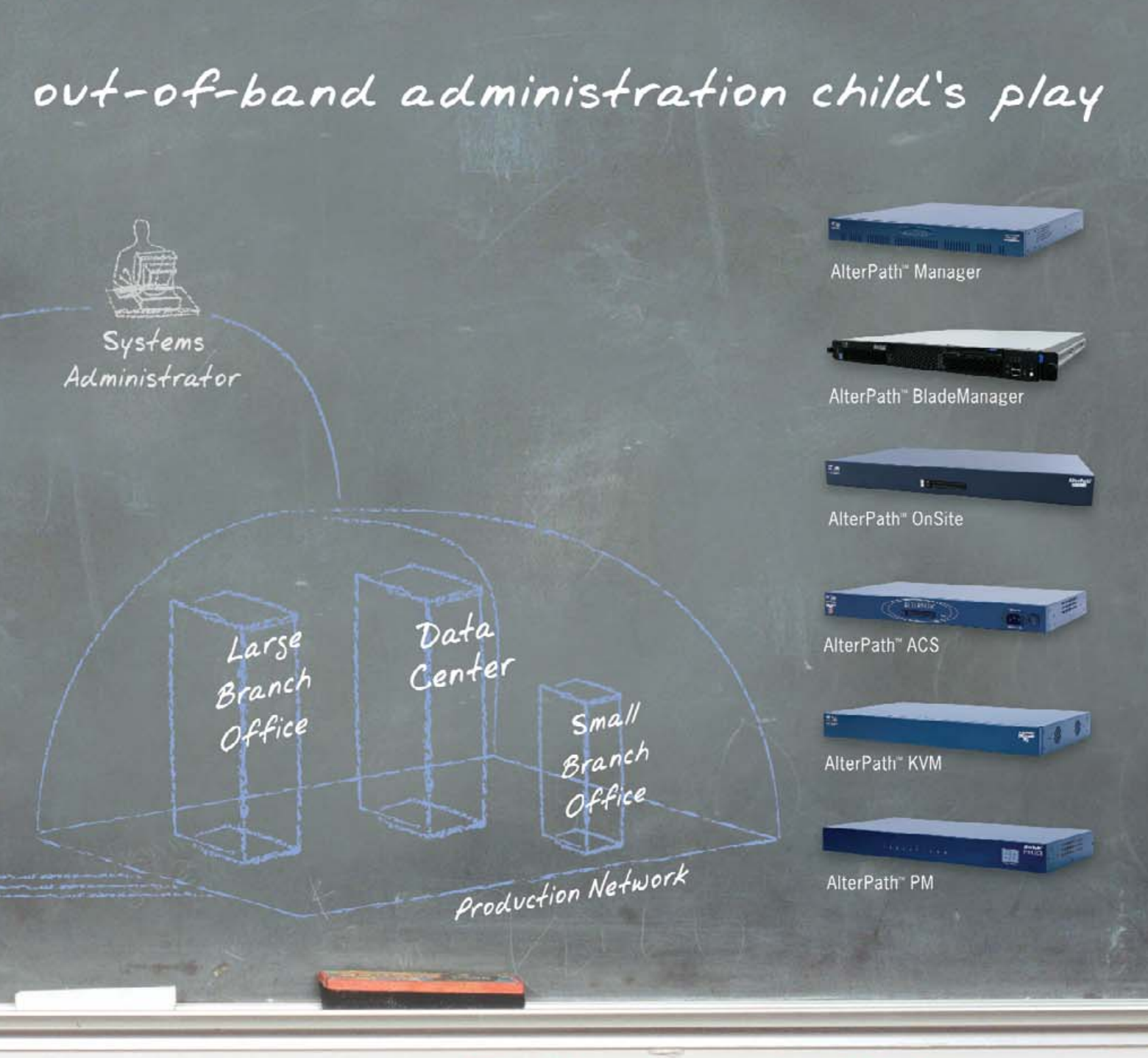
USA $5.00 CAN $6.50
www.linuxjournal.com
07
0 71486 03102 4

LINUX JOURNAL · SYSTEM ADMINISTRATION · RTLinux · Rekall · OpenLDAP · sed · PerfSuite · iPod · JULY 2005 ISSUE 135

Cyclades AlterPath™ System makes

A+

For a **FREE** copy of Enterprise Management Associates' white paper **Increased IT Operations Effectiveness With Cyclades AlterPath™ System for the Out-of-Band Infrastructure,** please visit us at www.cyclades.com/ema



The Next-Generation IT Infrastructure

Cyclades AlterPath™ System is the industry's most comprehensive Out-of-Band Infrastructure (OOBI) system. The AlterPath System allows remote data center administration, eliminating the need for most time-consuming, remedial site visits. When fully deployed in your data center, Cyclades AlterPath System lowers the risks associated with outages, improves productivity and operational efficiency, and cuts costs.

Each component of the AlterPath System is designed to seamlessly integrate into the enterprise, able to scale in any direction. Whether you need serial console management of networking equipment, KVM for access to Windows® servers, branch management, IPMI or HP iLO for service processor management or advanced power management, the AlterPath System delivers. Cyclades brings it all together, making OOBI administration seem like child's play.

Over 85% of Fortune 100 choose Cyclades.

www.cyclades.com/lja

1.888.cyclades • sales@cyclades.com

cyclades

Reclaim lost time

Levanta Intrepid ™

The world's first Linux management appliance

Plug the Levanta Intrepid ™ into your network and perform the most important Linux management tasks in a fraction of the time you spend now. And gain power and flexibility that you've never had before:

- **Fast & Portable:** Provision servers or workstations practically anywhere, anytime – in minutes. Swap them around, mix it up.
- **Flexible:** Supports commodity hardware, blades, virtual machines, and even mainframes.
- **Out of the Box:** Includes pre-defined templates for servers, workstations, & software stacks. Or create your own.
- **Total Control:** Track any file changes, by any means, at any time. And undo them at will.
- **Disaster Recovery:** Bring dead machines quickly back to life, even if they're unbootable.

Based upon technology that's already been proven in Fortune 500 enterprise data centers. Now available in a box, priced for smaller environments. **Just plug it in and go.**

30-Day
Money-Back Guarantee

Pre-order online by 7/31/05
Get $500 Off
Enter PROMO CODE: LJ0705

LEVANTA®
www.levanta.com
1.877.LEVANTA

LINUX JOURNAL
JULY 2005 ISSUE 135

COVER STORY

FEATURES

INDEPTH

EMBEDDED

TOOLBOX

COLUMN

DEPARTMENTS

ON THE COVER:
RICHARD NAYLOR PHOTO COURTESY OF CITYLINK, WELLINGTON, NEW ZEALAND

Thanks to customer-to-customer peering, users of Wellington's fiber-optic network pay only a flat rate (page 48).

NEXT MONTH

ULTIMATE LINUX BOX

A custom BIOS doesn't only speed up boot times. You get a failover BIOS to rescue you in case of boot problems, the ability to write BIOS code in C and more. Ron Minnich gets into the code.

Your computer's sound hardware probably has more functionality than you think. Find out what all those channels in the mixer application really do, as Dave Phillips explains the new sound standard, ALSA.

Finally, it's about that time again. We join up with a Linux hardware vendor to put together the Ultimate Linux Box. This time, we make a performance measurement that readers have been asking for, and we get a low number, which we'll reveal next issue.

Win-Lose Situations

Can't we all just get along? The answer is no, we can't, so we'd better at least be polite about it.

BY DON MARTI

We go to press at what we hope is the end of a nasty brouhaha over kernel development, with Linus Torvalds finally forced to abandon BitKeeper, his favorite source code management system, and develop an alternative. Samba guru Andrew Tridgell developed a free tool to pull data from BitKeeper repositories, BitKeeper's Larry McVoy responded by pulling the free-of-charge version that kernel developers had been using, and a lot of people's carpal tunnels took a beating in the resulting—well let's be nice and call it a discussion.

Much as we like to find win-win situations where we can, people have different goals. Are you writing a really good operating system kernel, are you building a proprietary software business or do you want to keep up with what's going on without having to accept BitKeeper's non-compete clause? The Linux business is all grown up in a lot of ways, moving billions of dollars in hardware, software and services, but now that we are in the IT marketplace, it's time to be more honest with ourselves about conflict.

Some of us are always going to sound like software freedom zealots, and some are always going to sound like greedy swindlers. The answer isn't to flame the other side with the "if you'd just compromise on my issue, it would be good for Linux" argument. Understand you're in conflict and sell your alternative as well as you can. Here's where *Linux Journal* comes down on a side that has to be in opposition to some of the other participants in the market—but we're not going to say it's for the good of "Linux", or start flaming when people won't act against their own interests.

In the long run, we say it's worth a lot of late nights, hot coffee and risking getting flamed on a support list to get your organization's directory service away from a proprietary choice and onto one of the freedom-friendly, standardized alternatives. Although you might be able to move some applications to Linux sooner if you just plug Linux in to your existing proprietary directory, that is the road to lock-in. Letting readers get locked in is bad for us because we're here to help everyone do new, innovative projects on all kinds of systems, not just whatever is in the directory vendor's interest to support.

When Craig Swanson and Matt Lung proposed the now-famous "OpenLDAP Everywhere" in 2002, it was as a piece on making your whole business run on Linux. That's a big subject, so Craig and Matt decided to narrow the focus. Strangely, they still managed to cover the essentials for getting your whole company running right. A lot has changed since 2002, so Craig and Matt are back on page 40 with a new, updated version that covers new software versions and lessons learned.

It's not all controversy this month, though. The best tools don't force you into hard choices. Joshua Bentham has an intro to a cross-platform, easy-to-use way to develop database apps on page 54. Enjoy the issue.

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

LINUX JOURNAL

JULY 2005
ISSUE 135

***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

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.

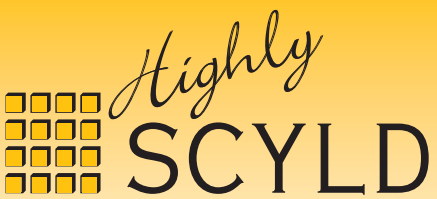
SCYLD

Pronunciation: ***skild*** (That's a hard "sc" as in "scalability," not a sibilant "sc" as in "sci-fi")
Function: *proper noun*
Etymology: *Scyld*, from Middle English *skilled*, to be exceptionally talented, trained, or abled

1: the original pioneer of Linux clustering software **2**: home of the industry leading Scyld Beowulf™ software **3**: the end of the nightmare of do-it-yourself Linux clustering **4**: how's this for some turn-key, worry-free features **a**: commercial-grade solution *<as in no integrating, testing and re-testing>* **b**: elegantly simple *<as in wickedly easy to use and highly scalable>* **c**: unified process space *<as in an SMP-like experience>* **d**: and get this: it runs out of the box; we repeat *<out of the box>* **5**: software sophisticated enough to manage the most compute-intensive applications and propel the most promising IT careers.

synonyms: elegance, simplicity, power

antonyms: labor intensive, SMP, Unix, Windows



www.scyld.com

Boo, Belly Dance Cover

I just received my May 2005 issue of the *Linux Journal* magazine. I am extremely upset with the cover picture selected for this issue. When I subscribed to this journal, I never dreamed my name would be associated with a "pictured" magazine like this. I was extremely embarrassed for my secretary to have had to see this when distributing my mail. I cannot display this on the public table for my other technicians to use as a reference in the office. Your company has presented a very volatile Sexual Harassment Potential in my Federal Government Office.

--

David Jerkins

A Black Wrapper for *LJ*?

I am shocked and appalled at the picture on the cover of this most recent issue [May 2005]. It really should have come in one of those black wrappers. If I want to see that sort of smut, I would subscribe to *Dr. Dobb's Journal*!

--

Rudy Pawul

Yay, Belly Dance Cover

I want to thank you for your interesting feature article, "Belly Dance and Free Software" by Dawn Devine and Michael Baxter [May 2005]. Although it's geared specifically toward performers, there's much that's useful for any business or anyone who has events to promote.

I would particularly like to commend you for your cover photo, because it explodes so many misconceptions at once! Too often, the computer world looks like a boys' clubhouse with a "NO GIRLS" sign on the door. To show a woman who's a professional in the arts, rather than another high-tech or high-business field, is even rarer. Being an artist and Middle Eastern dancer myself, as well as a Linux-using computer geek, I'm delighted!

--

Marty Hale-Evans

Photo of the Month: Talos the Rally Robot

This photo is of the robot Talos and his minders, Richard Gardiner, Aristotelis Papadimitriou, James O'Hea, Sang Hun Lee and Matthew Gray. Talos is one of four entries in the Cybernetics MEng Robot Rally Cybernetics ng. challenge project (see www.cyber.reading.ac.uk/robot_rally) and the only one to use Linux as its operating system. The challenge project is loosely based on the DARPA Grand Challenge. On March 17th, the four robots raced across the Reading University campus. Although none of the robots completed the course (just like their DARPA challenge counterparts) all managed to get past the first leg. Everything now depends on the final event in June.

--

William Harwin

Photo of the Month gets you a one-year extension for your subscription. Photos to ljeditor@ssc.com.—Ed.

Hello, My Name Is...

When my son was born in 1999, I named him Linus in honour of Linus Torvalds. I have included here a picture of me and my son Linus, who will be a future volunteer in the Linux community.

--

Jack Barbosa

Penguin Shopping Zone

I'm a Brazilian guy that has used Linux since 1998, when I worked at Conectiva, I designed (technically) the project to implant more than 20,000 workstations and 1,000 servers in our public network, here in São Paulo. This public network was on the schools and high schools to offer Internet access to the people who don't have it. This project was called Linux Na Escola (Linux in School), and until today, is the biggest project in this segment on earth. Workstations are all in production, in more than 500 labs educating thousands of children.

When I went to this supermarket to buy some beers and bodka, I saw penguins. I said to my cousin, "Look, Tux brings his family here to buy some fish, and beers too!"

--
Thiago Sobral

Starting Off Right

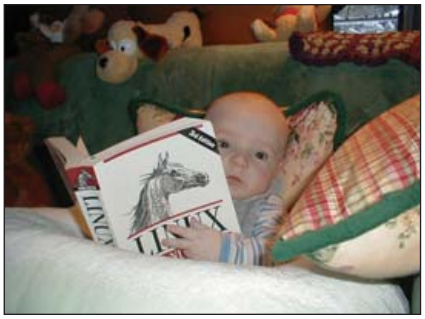
I wanted to send along a picture of my son with his baby tux shirt. Can't get them started on Linux too early!

--
W

You might want to point the new Linux users you know to www.tuxmagazine.com. *It's a free download, and we hear they have an article on TuxPaint coming up.—Ed.*

Hey, Mom, Block Port 137

Here is a picture of my four-month-old son, Griffin, looking up information on iptables



one snowbound day in February. He was very helpful. I have not let him get a hold of my magazine yet. Somehow I do not think it would survive.

--
Laura Goepfert

Adding Ham Position Data to GPSDrive?

I read with great interest in the April 2005 issue of *LJ* Charles Curley's article "Finding Your Way with GPSDrive". With a little additional code, GPSDrive could be easily modified with code for Automatic Position Reporting System (APRS) use in emergency services. With WiFi support, some radio data traffic that are non-priority, or very large files, it could lessen the load on congested ham radio frequencies. WiFi would be excellent to backhaul VoIP traffic, pictures and video and other such things to be shared with emergency services personnel. The possibilities are endless.

I noticed that one can import USGS maps, or street maps, but does this software support importing mapsets from DeLorme? USGS maps leave much to be desired, as some of the maps date back to before the 1960s. Of course, the topo does not change, but roads are sometimes nonexistant on these maps. Topo USA I found to be much better, and also covers current roads.

Regarding Chris McAvoy's reply to Jason Shelton's letter [*LJ*, April 2005], the MAX232 chip from Maxim is a very versatile chip. The kit for the MAX232 may be inexpensive, but "samples" of this chip and others are free. Maxim has a very extensive line of chips for use in the computer, electronics and ham radio hobbies.

LJ is my only connection to the Linux world until 2019. I am in Federal Prison for a victimless crime. Glad to see that there are other Tux fans in a microcosm society that sometimes has no justice or intelligent life. Please keep up the good work with *LJ*. I am glad to see the radio geek articles, along with articles covering other hobbies. How about some

MonarchComputer.com

AMD Solution Provider PLATINUM

Visit our website or Call 1-800-611-0875

PayPal

FREE! - Get Half-Life 2 by upgrading to AMD Athlon™ 64! Get 4 months FREE Napster music service with AMD Sempron™

Purchase any AMD Athlon™ 64 or Athlon™ 64 FX combo, barebone, or system from monarchcomputer.com and get Half-Life 2. Get a 4 month subscription to Napster™, including 12 FREE song downloads by upgrading to AMD Sempron™! Visit our website for full details!

napster. Get 4 months of Napster® FREE with qualified purchase.

FREE INSTALLATION SETUP & TESTING BY CERTIFIED TECHS

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

AMD Motherboard Combos

Asus K8V-SE Deluxe Mainboard with AMD Sempron™ processor 2600+ (754) — Only $199

Asus A8V-E Deluxe Mainboard with AMD Athlon™ 64 processor 3000+ (939) — Only $269

Tyan S2882G3NR Mainboard with AMD Opteron™ processor 244 — Only $589

Tyan S2882-DGNR (Thunder K8SD) with AMD Opteron™ processor 265 (Dual Core) — Only $1239

WinFast NF4K8MC-ERS nForce 4 Mainboard w/ AMD Athlon™ 64 processor 3200+ (939) — Only $279

Asus K8N-DL nForce4 Pro MB w/ AMD Opteron™ processor 275 (Dual Core) — Only $1639

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 Sempron™, AMD Athlon™ 64 or AMD Opteron™. Then configure your barebones online or call 1-8-MONARCHPC.

Lian-Li PC-V1200 Server Case (Black) w/Wheels w/460W PS
Asus K8N-DL nForce4 Motherboard
AMD Opteron™ processor 244 1.4 GHz
Starting @ $869

Thermalrock Ocean Dome Full Tower E-ATX Case w/Window w/460W PS
Gigabyte GA-K8NF-9 nForce 4
AMD Athlon™ 64 processor 3500+ (939 - 90nm)
Starting @ $609

Thermalrock Dragon Full Tower E-ATX Case w/Window (Black) w/460W PS
Tyan S2882-DG3NR Thunder K8SD
AMD Opteron™ processor 270 (Dual Core)
Starting @ $1669

Antec Plusview 1000AMG Case w/300W PS
MSI K8M Neo-V Motherboard w/
AMD Sempron™ processor 2600+ (754)
Starting @ $269

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

www.monarchcomputer.com

AMD Price Drop!

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 3700+ (939) $329.00
- AMD Athlon™ 64 3800+ (939) $424.00
- AMD Athlon™ 64 4000+ (939) $643.00
- AMD Athlon™ 64 FX-55 (939) $848.00

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™ 142 1.6GHz $158.00
- 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™ 242 1.6GHz $157.00
- AMD Opteron™ 244 1.8GHz $203.00
- AMD Opteron™ 246 2GHz $307.00
- AMD Opteron™ 246 HE 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™ 846 2GHz $677.00
- AMD Opteron™ 846HE 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

AMD Opteron™ Dual Core CPUs

- AMD Opteron™ 265 1.8GHz $851.00
- AMD Opteron™ 270 2.0GHz $1051.00
- AMD Opteron™ 275 2.2GHz $1299.00
- AMD Opteron™ 865 1.8GHz $1514.00
- AMD Opteron™ 870 2.0GHz $2149.00
- AMD Opteron™ 875 2.2GHz $2649.00

NEW Rev E-SSE3 CPUs Available at 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

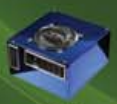
150137
Western Digital 250 GB WD2500JB 7200 RPM 8MB IDE
$130.00

140270
1 GB DDR (266) PC-2100 REG ECC Corsair (CM72SD1024RLP-2100)
$125.00

140662
512 MB DDR (266) PC-2100 REG-ECC Mushkin Pro (990742)
$109.00

190528
ATI (Connect3D) Radeon 9250 SE 128MB DDR/8x-AGP/TV-Out (Retail Box)
$39.00

150239
Western Digital 74 GB SATA 10K Raptor (WD740GD)
$189.00



100963
Ultra X-Connect ULT31558 ATX 500 Watt Power Supply w/2-80mm Fans (Blue)
$69.00

110192
Asus K8N-DL nForce4 Audio/GB-LAN/USB/IEEE PCI-E/SATA/DDR/E-ATX Opteron
$339.00

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.

Monarch Has The *LOWEST PRICES*
Custom 64-Bit Servers, Workstations & Desktops
Available with AMD Dual-Core Technology!

AMD Sempron™ | 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!

"I bought the dream system with top components that were hard to find elsewhere. The guys at Monarch did a perfect job of building it, down to the smallest details. Wiring was perfect, position of hard drives within the chassis also perfect, etc. Even the packaging was superb. And every communication I've had with them was also perfect. Ordering from Monarch was a good move."

User: canbbb - ResellerRatings.com

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

"BOTTOM LINE: MUST BUY"

"What's not to like? Monarch provides top parts, excellent customer service, and has earned the highest-level solutions provider status recognized by AMD and other key component vendors."

Jason Perlow
Linux Magazine
April 2005

LINUX MAGAZINE EMPEROR CLASS

For more information on the Empro™ Line Visit www.monarchcomputer.com/empro

LINUX FRIENDLY!
CUSTOM AND PREBUILT PCS - Linux Preinstalled!
Fedora - RedHat - Mandrake - SuSe

LOOK FOR THIS SYMBOL ON LINUX COMPATIBLE MONARCH SYSTEMS!

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

The Official Linux Journal Web Server
Monarch Computer Systems™

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

Monarch Empro™ Enterprise 4P/8-core Rack Server

Based on Dual-Core AMD Opteron™ processors

Also Available in Yellow Orange or Blue!

SELECTED COMPONENTS:
Monarch Enterprise 3U Rack Mount Server System (case and motherboard)
Up to 64 GB DDR SDRAM, 16 DIMM Slots, supports DDR-400, 7 PCI-X Slots
Dual Redundant (n+1) Hot-swap PS
Features 4 AMD Opteron™ Processors 875 (2.2 GHz)
8GB (8 pcs 2GB) DDR (333) PC-2700 Corsair
2 x Seagate Cheetah 73.4 GB 10k RPM SCSI HDDs w/Hot Plug Bays
Slim DVD Combo Drive
Floppy Drive
Industry Standard Upgradable
1 or 3 year warranties available
24/7 on-site service available

Compare Sun Microsystems Sun Fire V40z Server Extra Large (Dual Core) Config priced at *$38,995.00
Get 2 Monarchs for the Price of 1 Sun!

Starting @ ONLY $20789!

Monarch Empro™ Custom Rack Server (2U configuration)

1 Terabyte Shown Here - Upgradable to 2 Terabytes!

Part #: 80313
SELECTED COMPONENTS:
AIC RMC2K2-9I-XPSS 2U Rack Mount w/460W Power Supply
Tyan S2882G3NR (Thunder K8S)
2 x AMD Opteron™ 248 2.2GHz
2 GB (2 pcs 1GB) DDR (400) PC-3200 Corsair
4x Western Digital 250GB 8MB Cache 7200 RPM SATA (WD2500JD)
3Ware Escalade 9500S-8 8 Port SATA
Mitsumi 24X Slim CD-ROM (Black)
Sony MPF820 1.44 Slim Floppy Drive (Black)
Industry Standard Upgradable
1 or 3 year warranties available
24/7 on-site service available

This Configuration Starting @ ONLY $3589!

Monarch Empro™ Custom Rack Server (3U configuration)

1 Terabyte Shown Here - Upgradable to 3 Terabytes!

Part #: 80313
SELECTED COMPONENTS:
AIC RMC3K2-0-XPSS 3U Rack Mount w/600W Power Supply
Tyan S2882G3NR (Thunder K8S)
2 x AMD Opteron™ 248 2.2GHz
2 GB (2 pcs 1GB) DDR (400) PC-3200 Corsair
4x Western Digital 250GB 8MB Cache 7200 RPM SATA (WD2500JD)
3Ware Escalade 9500S-12 12 Port SATA
Mitsumi 24X Slim CD-ROM (Black)
Sony MPF820 1.44 Slim Floppy Drive (Black)
Industry Standard Upgradable
1 or 3 year warranties available
24/7 on-site service available

This Configuration Starting @ ONLY $4059!

Monarch Empro™ 1U Value Server Rack Special (IDE & SATA)

Great Server for a Great Price! One of our Best Selling Configurations!

Part #: 90344
SELECTED COMPONENTS:
AIC RMC1L2-6I-XP 1U Rack Mount Server w/400W EPS Power Supply
Tyan S2882G3NR (Thunder K8S) AMD 8131 Onboard Video, GB-LAN, USB, SATA w/RAID, REG DDR, E-ATX
2 x AMD Opteron 240 1.4GHz 1MB 64/32 Bit
1 GB (2 pcs 512) DDR (400) PC-3200 REG Corsair (TwinX1024R-3200C2PT)
Western Digital 80GB, 8MB Cache, 7200 RPM, SATA (WD800JD)
Samsung H492A 16x52X32X52 DVD/CD-RW W/Nero Software (Black)
Mitsumi 1.44MB 3.5" Floppy Drive (Black)

This Configuration Starting @ ONLY $1589!

Commercial leasing available for purchases as low as $1000.

Prices subject to change without notice. Monarch 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

embedded Linux while SCUBA diving?

--

William StormBringer Smith

Yes, DeLorme format is one of many supported by GPSBabel (gpsbabel.sourceforge.net). We'll look for an article on integrating APRS and GPSDrive.—Ed.

Vendor Troubles

I've read many things about Monarch Computers in your magazine (which I greatly enjoy). However, my experience with Monarch is less than acceptable.

It took a month to receive our new dual Xeon server, and it didn't work on arrival. They do not seem to respond to e-mail or telephone queries about orders or rma's. The only responses I get are autogenerated. The voice mail box is full and will not accept a message.

--

Jack Adwon

We sent a copy of this letter to Monarch, but did not get a response.—Ed.

Moodle Success Story

I just received the first issue of my *Linux Journal* subscription and I am pleased with every page. I am a newbie (but learning quickly) and I was surprised to see a letter from an inmate who finds it difficult to meet other Linux enthusiasts in prison. I am also incarcerated and know how hard it is to learn about Linux since most of the people around here either use Windows or don't know anything at all about computers. I don't have access to the Internet until 2008, but fortunately I've been able to acquire an excellent job at the Vo-tech here. I set up a LAMP platform on a Dell Poweredge 4400, a Fedora workstation for development and 25 Win2k workstations; and I owe *Linux Journal* a debt of gratitude.

Another inmate here introduced me to *Linux Journal* and I read an article about Moodle [December 2004]. We are now happily using this program, which makes my job easier and provides me with more time to learn about Linux. I've seen a few advertisements for Linux certification programs (such as Linux Professional Institute, Red Hat and so on) and I wonder which certifications are the most marketable and recognized by employers. I just want to focus on getting the best Linux education I possibly can. Thank you for your wonderful magazine.

PS. I really liked the "Linux on a Small Satellite" article [April 2005]. I am interested in Amateur Near Space flight using helium-filled weather balloons, and I've been looking for examples of how Linux can be used for these projects. All of the spacecraft I've seen so far only use Basic Stamps and PBasic code. If you can do a piece on near-space sometime, that would be excellent.

--

Aaron Kirby

Father and Son Reading

Here's a picture of my youngest, enjoying the latest *Linux Journal*. Thanks for a great magazine!

--

Keith Blackwell

Thanks for Shell Tips

LJ is the most educational computer magazine I have ever known. I learn something new with every issue, often several new somethings. The article on Bash in the April 2005 issue was very enlightening, and I have used some of the techniques described in shell scripts used in the Computer Repair class for user account maintenance on the Linux-based server for the LAN operated in the class.

Over the past few months I have seen letters written by other incarcerated persons and want to express my thanks that *LJ* recognizes that while some Linux enthusiasts may be in prison, the mistakes individuals like myself have made in the past do not nullify our abilities to spread the ideals of Linux or our abilities to make meaningful contributions to Linux and free software.

Please keep up the excellent work and while I didn't get to be part of your first ten years, I hope to be a part of the next ten, twenty, thirty years.

--

Frank O. Robinson

Breakfast with a Penguin

Here's my four-month-old daughter Amelia Danielle playing with an old-school Caldera Tux! Can't start 'em early enough, eh?

--

Scott Friedman

Check Mount Options for Live CD

Thanks for the great piece on building live CDs [April 2005]. The recipe for mounting and chrooting the decompressed Knoppix tree will work if you do it on a filesystem mounted the usual "defaults" way. But I tried it in my test machine running off a Knoppix CD and it didn't work.

Knoppix had mounted my hard drive partition (/mnt/hda5) with no dev options, so no devices worked in the chrooted environment. I needed to do:

```
mount -o remount,dev /mnt/hda5
```

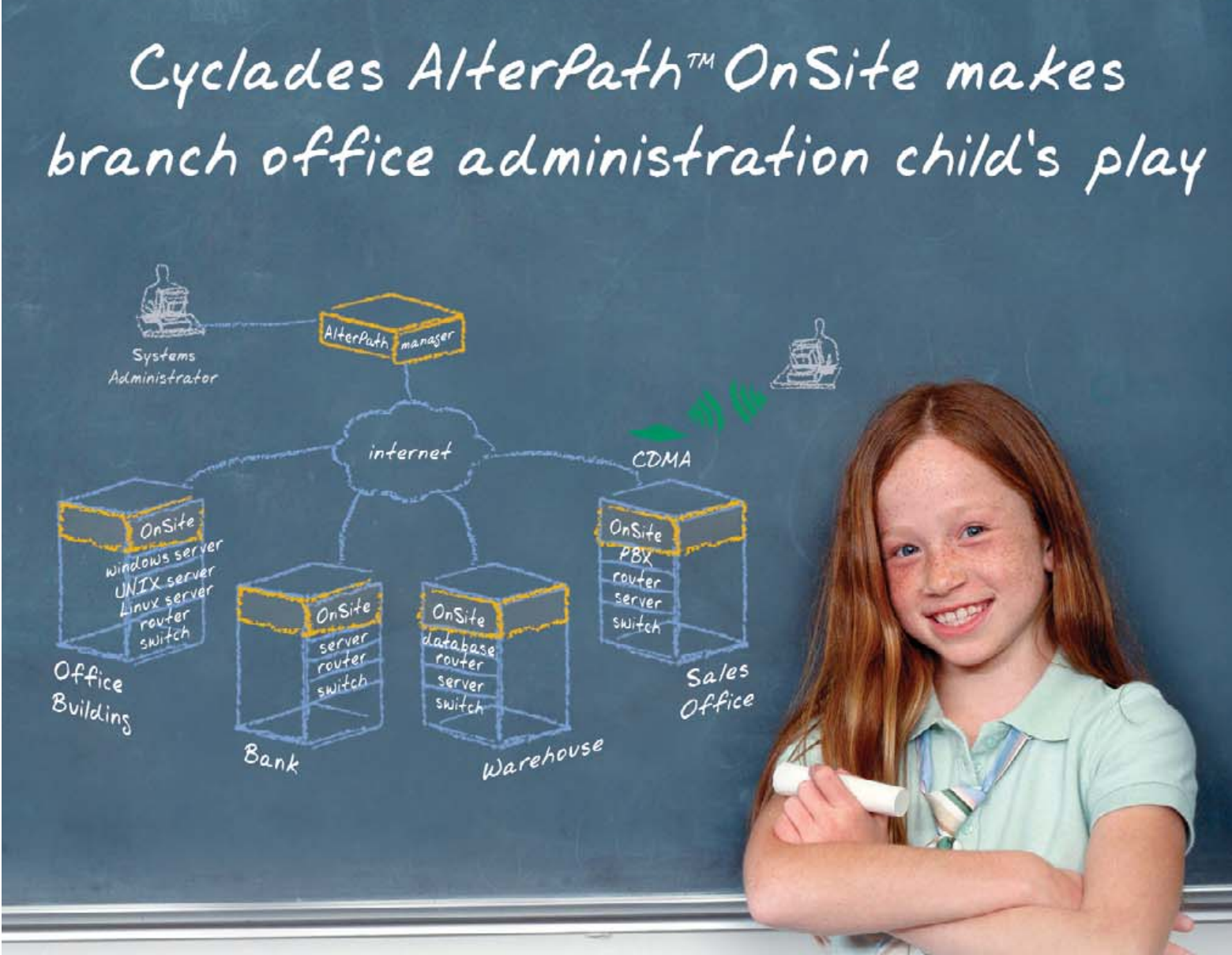
just before giving the chroot command, and everything worked.

--

Cameron Splitzer

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™ OnSite makes branch office administration child's play



AlterPath™ OnSite

A+

For a **FREE** copy of Enterprise Management Associates' white paper **Increased IT Operations Effectiveness With Cyclades AlterPath™ System for the Out-of-Band Infrastructure,** please visit us at www.cyclades.com/ema

The Next-Generation IT Infrastructure

Cyclades AlterPath™ OnSite is the most comprehensive remote site and branch office administration appliance available. This small, inexpensive solution for controlling network equipment, servers and other IT infrastructure devices can

- Access, diagnose and restore remote IT devices quickly
- Download software to multiple devices automatically and simultaneously
- Configure user information, system settings and operating parameters
- Send alerts of intrusions, equipment failures and alarms

The AlterPath OnSite combines the functionality of both serial console and KVM over IP, allowing IT administrators to manage multiple servers and network devices through a single appliance. Cyclades brings it all together making remote site and branch office administration seem like child's play.

Over 85% of Fortune 100 choose Cyclades.

www.cyclades.com/ljb

1.888.cyclades · sales@cyclades.com

cyclades

On the WEB

We're counting on our readers even more this year to participate in the Readers' Choice Awards, and voting will happen in two stages. By the time you read this, "First-Round Voting in 2005 Readers' Choice Awards" (www.linuxjournal.com/article/8266) will be underway on the *Linux Journal* Web site. The final ballot will be based on the results of that initial ballot. Final voting will take place in July, and the winners will be announced in the November 2005 issue of *Linux Journal*. As the name says, these are the *LJ* Readers' Choice Awards, so get on over to the Web site, read the nomination list and send us your votes!

For complete information, details and dates regarding the 2005 Readers' Choice Awards, read "New Procedures for 2005 Readers' Choice Awards" (www.linuxjournal.com/article/8192).

Back in November 2004, Michael Boerner reviewed two of gumstix's tiny little SBCs, built around the Intel XScale PXA255 chip with Linux onboard. Since then, gumstix's product line has expanded, and Michael is following its progress in a two-part review to be featured on the *Linux Journal* Web site. He tells us, "My last review focused on the waysmall computer (WS200-bt) and its components, as well as using it and its different elements, such as Bluetooth." This set of reviews focuses on the revised unit, with the addition of Ethernet, the new audio, breakout and external Flash storage modules and actually using the units for embedded applications. In "Testing and Building with the New gumstix SBCs, Part 1" (www.linuxjournal.com/article/8268), Boerner looks at these gumstix components: gumstix connex, a 10-100baseT wired etherstix and a waysmall board, which converts the gumstix unit into a waysmall computer.

diff -u

What's New in Kernel Development

Linux kernel development is in the process of creating a new push for stability. **Greg Kroah-Hartman** and **Chris Wright** have volunteered to maintain a tightly controlled **stable** tree. The stable trees of the old days would alternate with development trees in cycles measured in years. This new stable tree will exist concurrently with the development tree and will consist solely of important bug fixes. So the 2.6.11 kernel release from **Linus Torvalds** has been followed by 2.6.11.1, 2.6.11.2 and additional stabilizing releases from Greg and Chris. Even after 2.6.12 is released, the 2.6.11.z tree may continue to stabilize, even as 2.6.12.1 comes out, and so on. Unlike previous stable series, whose maintainers had wide latitude to choose which patches to accept, there are strict rules over what can go into this new stable branch. Even the method of considering patches, and the time between patch submissions and patch acceptance or rejection, is tightly regulated. The stable tree has been dubbed by Linus, the "sucker" tree, because he felt no one in his or her right mind would take on the burden of maintaining it. Chris and Greg have risen to the challenge, and the process is itself still undergoing changes in all aspects. But it does appear that stability is once again a serious target of Linux development.

The **SysFS** filesystem recently felt a pang of uncertainty. One of the driving forces behind its development always has been to replace the chaos and historical baggage of ProcFS with something clean and sane. The developers hoped that, given a fresh start, the old mistakes could be avoided. Recently, however, kernel folks realized that one of the SysFS directories had been put in the wrong place: /sys/block, it was felt, should really have been /sys/class/block instead. Too late! A great mass of user code already had come to rely on the existing directory location. Greg Kroah-Hartman reluctantly had to admit that the SysFS inconsistency could not be repaired. The first spot of age has appeared on the pristine face of SysFS.

SquashFS continues to try for kernel inclusion and continues to come up short, and its principal developer, **Phillip Lougher**, grows more and more frustrated. One obstacle appears to be cultural: the kernel developers would like to hear convincing reasons in favor of inclusion; while Phillip has been, as he's said, more concerned with coding the filesystem than selling it. Other obstacles are more technical. Currently, for example, SquashFS has a file size limit of 4GB. Of course, because SquashFS is a compressed filesystem, this really amounts to about 8MB of actual data. Also, it recently was pointed out that readdir() does not return either the ./ or ../ directories for SquashFS, as it does for virtually all standard filesystems. These and other problems continue to thwart efforts to get SquashFS into the standard kernel.

FUSE (Filesystem in USErspace) seems to be on the brink of getting into the main kernel tree, after spending quite a while in **Andrew Morton**'s -mm branch. FUSE has had a rough life, with Linus Torvalds saying for a long time that he thought a user-space filesystem was just inherently a bad idea that never would go anywhere. But, FUSE apparently is turning into the little engine that could and even its detractors are having to step aside. Andrew is turning into a strong proponent and seems ready at last to push it along to Linus.

Several projects have changed hands recently, or their maintainers have become officially recognized for the first time. **Pete Zaitcev** has been listed as maintainer for both the **USB block driver** and the **Yamaha PCI sound driver**. **Herbert Xu** has replaced **James Morris** as co-maintainer of the kernel **crypto API**. And **Gerd Knorr** has stepped down as the **Video4Linux** maintainer, leaving that project currently unmaintained.

—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

Crimson Fields

crimson.seul.org

This 2-D turn-based game offers many strategy choices, no icky blood and guts and lots of challenging maps. You control an army of tanks, helicopters, artillery and other units, and each map gives you a different set of units to command and different victory conditions. The project Web site has a tutorial on creating your own maps too.

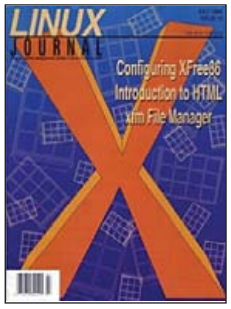
The interface is slick and the code is rock solid. Play "hot seat" with two players taking turns, by e-mail or against the computer.

The AI is merciless, even where its own units are concerned. Sometimes you can hold a defensive line against a crowd of computer units and pick off the ones in the middle using artillery. The AI doesn't seem smart enough to fall back to a better position.

Crimson Fields is based on the popular Simple DirectMedia Layer (SDL), a library that you probably already have on your system to support other games.

—DON MARTI

Ten Years Ago in *Linux Journal:* July 1995



Kids today have it pretty easy. Ten years ago, we needed half a magazine merely to get the basics going. Greg Lehsy's tutorial on configuring XFree86 included a helpful "How to Fry Your Monitor" section, covering things *not* to do. Configuring wasn't just a matter of editing a tweaky config file—it was a matter of editing a tweaky config file that, if you got it wrong, would destroy your hardware.

We also reviewed two proprietary X server packages with graphical configuration utilities. Both Metro-X and Accelerated-X came up and ran without editing any text.

Dean Oisboid surveyed games for Linux, including the classic *Adventure* and the BSD games. He ran into a little trouble with Id Software's *Doom*: "I just wanted to play *Doom* with sound. Having to recompile the kernel just to get sound had to be insane, but it seemed that this simple yearning for *Doom* had to develop into a learning experience." One quick compilation tutorial later, "It worked!"

Advertisers offered everything from full systems to mouse pads and T-shirts. A Burgess Shale of distribution ads included Yggdrasil Plug & Play Linux; Slackware; a Pacific HiTech set with four distributions including Debian; Caldera Network Desktop, which was still based on Red Hat; SoftCraft Linux; LinuxWare from Trans-Ameritech; and S.u.S.E., still with all four periods. PromoX advertised a 100MHz Pentium system with 16MB of RAM and a 540MB hard drive for $2,500.

—DON MARTI

They Said It

"Everybody in Mali uses Linux." That is no doubt a bit of an exaggeration, but it's a phrase that you'd hear only in a flat world.

—TOM FRIEDMAN, IN *THE WORLD IS FLAT: A BRIEF HISTORY OF THE 21ST CENTURY*

A telemedicine centre was installed in Segou and Sikasso, regional capitals of Mali, to improve the ability of these medical centres to both communicate and provide quality medical information. These systems are also being used for "tele-radiology", where these centres can scan an x-ray, then transmit it to specialists in Europe or the United States for a second opinion.

—GEEKCORPS MALI, mali.geekcorps.org/article.php3?id_article=52

The capability of computers keeps growing and the number of applications running keeps increasing. The people building the interface keep growing the complexity of that. It's not for lack of effort but the software people are losing ground.

—GORDON MOORE, blogs.zdnet.com/BTL/index.php?p=1264

You know, Richard [Stallman] is in many ways a walking advertisement for the advantages of not compromising—when you have a long-term goal whose achievement requires dogged long-term effort over decades.

—RICK MOEN, linuxmafia.com/pipermail/conspire/2005-April/000996.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

Databases and Calendars

Building with the iCalendar standard, it's time to extract schedule information from a database and build calendars on the fly. BY REUVEN M. LERNER

Last month, we continued our look at the iCalendar standard, which makes it possible for programs to exchange calendar and appointment information. As we saw, an iCalendar file contains one or more events and tasks. If we make the file available via an HTTP server such as Apache, we can distribute it to anyone with an iCalendar-compatible program, such as Mozilla's Sunbird. As we saw last month, we can go one step further than this, generating an iCalendar file dynamically, using a CGI program.

Although the programs I presented and discussed last month might be useful in a limited context, it should be clear to any Web developer that keeping the date and time information inside of a program would be foolish, to say the least.

One of the best ways to keep track of such data is in a relational database such as PostgreSQL. A relational database allows you to ensure that the data you have entered is valid and provides you with fast, flexible access to some or all of the data it contains. Moreover, by storing the calendar information inside of a database, you can create multiple versions of the same calendar file, using the same source.

This month, we look at a simple example of a Web-based program that takes calendar information from a relational database and uses it to generate an iCalendar data file, which then can be imported into iCalendar-compliant programs, such as Mozilla's Sunbird.

Defining the Table

If we are going to store our calendar information in a relational database, we need to define at least one table. This is because everything in a relational database—often including configuration and status information—is stored in a two-dimensional table, in which the columns define individual fields, and each row contains one record. For example, here is how we might define a simple table of events in PostgreSQL:

```
CREATE TABLE Events (
  event_id         SERIAL     NOT NULL,
  event_summary    TEXT       NOT NULL
    CHECK (event_summary <> ''),
  event_location   TEXT       NOT NULL
    CHECK (event_location <> ''),
  event_start      TIMESTAMP  NOT NULL,
  event_end        TIMESTAMP  NOT NULL,
  event_timestamp  TIMESTAMP  NOT NULL
    DEFAULT NOW(),
  PRIMARY KEY(event_id)
);
```

The above table contains six columns. The first, event_id, is defined to be of type SERIAL. If we don't explicitly provide a value for event_id when adding a row to the table, PostgreSQL retrieves a new integer value automatically, up to a maximum of 2^{31}. PostgreSQL allows you to set a larger ceiling to allow the sequence to wrap around to 1, or both; see the documentation for more details.

The event_id column uniquely identifies rows in our table, and we tell the database this by marking it as a PRIMARY KEY. This not only tells other database programmers which column will be used for retrieving records, but it ensures that values are unique and that the column is indexed as well.

Another automatically populated column is event_timestamp. From the definition, it might appear as though we can (and will) set event_timestamp to an explicit value, with the current time providing a default as necessary. But whenever I define a column in this way, it implies that I never expect to set a value explicitly for this column. Rather, I am interested in letting PostgreSQL set the column's value with the current date and time.

Notice how the event_summary and event_location columns are both defined to be of type TEXT (that is, infinite-length text fields), while event_start, event_end and event_timestamp are all of type TIMESTAMP, the SQL-standard way of saying date and time.

All of the columns in this table are defined to be NOT NULL, meaning that they may not be assigned SQL's undefined value of NULL. NULL is distinct from true and false, which can make it a bit tricky for newcomers to understand. However, if you think of NULL as representing an unknown or undefined value, it might become clearer. As useful as NULLs can be in distinguishing between false and unknown values, it's usually a good idea to cut down on them as much as possible. Indeed, the advice that I have long heard, and repeated to others, is that you should define columns to be NOT NULL by default, opening them up to NULL values as the situation requires.

Finally, notice how our two text columns (event_summary and event_location) are defined both as NOT NULL and with an integrity check that ensures we enter something other than an empty string. Whether this combination of constraints is

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

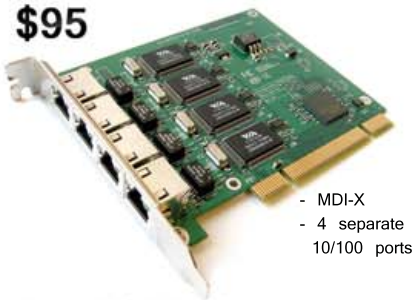
- Nstreme wireless polling protocol
  - no decrease in speed over long distances (as seen with the 802.11 ack packet bottleneck)
  - polling improves speed and eliminates contention for access to the wireless bandwidth
  - access point control over Nstreme clients tx data to optimize use of the wireless 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**

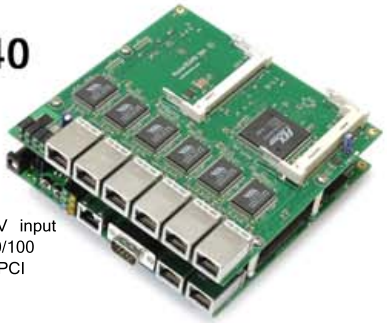
Eight ports Four ports

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

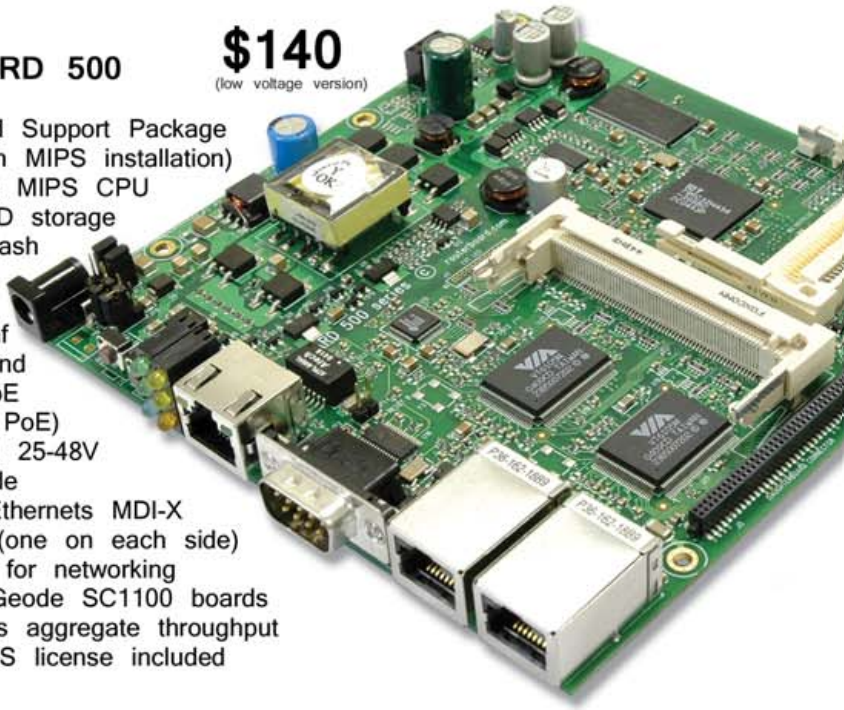
RouterBOARD 500 & RouterBOARD 564

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

appropriate depends on your database needs. You might want to think about whether you want to allow NULL values at all and also if you want to allow empty strings for the summary and location columns.

Although this simple definition is meant to serve as an example, consider how much better it would be if we were to have a separate Locations table, with a location_id and location_name, and then replace the textual event_location column with a location_id. This would have the advantage of standardizing location names, which would lead to fewer inconsistencies. It also would allow us to search for all of the events taking place in a particular location.

Once we are done defining the table, we add some indexes to our table. Each index ensures that data will be retrieved from the table more quickly than usual, at the expense of additional time for each INSERT. Here are the definitions:

```
CREATE INDEX event_location_idx
  ON Events(event_location);
CREATE INDEX event_start_idx
  ON Events(event_start);
CREATE INDEX event_end_idx
  ON Events(event_end);
```

Inserting New Data

Now that we have a defined table and indexes, we can start to populate our database table with some events. We can, as always, INSERT new events into our table with the following syntax:

```
INSERT INTO Events
  (event_summary, event_location,
   event_start, event_end)
VALUES
  ('Ides of March', 'Everywhere',
 '2005-March-15 00:00', '2005-March-15 23:59:59')
```

As you can see, the above INSERT statement names only four of the six columns defined in Events. When we check our new row, we find the following:

```
atf=# select * from events;
-[ RECORD 1 ]---+--------------------------
event_id        | 1
event_summary   | Ides of March
event_location  | Everywhere
event_start     | 2005-03-15 00:00:00
event_end       | 2005-03-15 23:59:59
event_timestamp | 2005-04-04 01:20:15.575032
```

As you can see, event_id (which we defined to be of type SERIAL) has automatically received a value of 1. Furthermore, event_timestamp has been set with the date and time at which we executed the query.

It's easy to imagine how we could invoke this INSERT statement with a Web-based program using CGI or a more advanced system, such as mod_perl or Zope. Indeed, we really don't have to think much about how the data has arrived in the database, particularly if we have set appropriate constraints on our data. We can assume that whatever resides in the database

Listing 1. db-calendar.py

```
#!/usr/bin/python

# Grab the CGI module
import cgi
import psycopg
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
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 the database connection
db_connection =
  psycopg.connect('dbname=atf user=reuven')
db_cursor = db_connection.cursor()

db_cursor.execute
  ('''SELECT event_id, event_summary, event_location,
             event_start, event_end, event_timestamp
                 FROM Events
             ORDER BY event_start''')

result_rows = db_cursor.fetchall()

for row in result_rows:
    # Create one event
    event = Event()

    # Set the event ID
    event['uid'] = str(row[0]) + 'id@ATF'

    # Set the description and location
    event.add('summary', row[1])
    event.add('location', row[2])

    # Transform the dates appropriately
    event.add('dtstart', datetime(tzinfo=UTC(),
              *row[3].tuple()[0:5]))
    event.add('dtend', datetime(tzinfo=UTC(),
              *row[4].tuple()[0:5]))
    event.add('dtstamp', datetime(tzinfo=UTC(),
              *row[5].tuple()[0:5]))

    # 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()
```

is reliable, and that the server has rejected any entries that would violate our rules.

Creating a Dynamic iCalendar File

Now that we have some activities in our database table, we can retrieve them into a CGI program. That program then produces output in iCalendar format, allowing iCalendar clients to retrieve its data. Listing 1 contains the program, which is a modified version of last month's dynamic-calendar.py program. As I mentioned last month, I wrote this program in Python in no small part because of the relative dearth of modules to create iCalendar-format files. Fortunately, there is such a module for Python, and I have taken advantage of that fact in this program.

As you can see in Listing 1, the program is fairly straightforward. After importing a number of modules, we create a calendar object and insert the iCalendar-mandated fields indicating the source of the calendar.

We then connect to a PostgreSQL server, which is presumed to be on the local computer. Although several database adaptors exist in Python for PostgreSQL access, I have long used psycopg, which is both fast and stable. To connect to PostgreSQL with psycopg, we use the following syntax:

```
db_connection = psycopg.connect
                    ('dbname=atf user=reuven')
```

The above indicates that the database name is atf and the user name is reuven. You also might need to specify the server and a password as additional arguments, especially if you are working on a production system.

Once we have connected to the database, we get a cursor, which allows us to submit queries and get their results:

```
db_cursor = db_connection.cursor()
```

With a cursor in hand, we now can send our SQL query to the database, using Python's triple-quote functionality to make our SQL more readable. Now we retrieve our results. If we were expecting to retrieve dozens or hundreds of rows, we probably would want to get them one at a time, or perhaps in batches. But I know that this calendar will contain only a few events, so I use the fetchall() method to

get them in one large sequence:

```
result_rows = db_crsor.fetchall()
```

Each element of result_rows is a row from our PostgreSQL database. We thus iterate (in a for loop) over the rows, retrieving the different elements that appear.

For the most part, this is pretty straightforward. However, things get a bit tricky when we are working with dates and times—important elements of any calendar of events! The problem is that psycopg uses the open-source mxDateTime module from eGenix.com, which makes working with dates extremely easy. But mxm's iCalendar module uses Python's datetime module, which is different. We thus need to retrieve each of the dates (for the event's starting time, ending time and stamp), turn them from an instance of mxDateTime into a datetime-compatible tuple, use that tuple to create an instance of datetime and then pass that to event.add, using the three calls starting with:

```
event.add('dtstart', datetime(tzinfo=UTC(),
event.add('dtend', datetime(tzinfo=UTC(),
          *row[3].tuple()[0:5]))
```

The second argument to datetime() in the above three rows of code does exactly what we said. It retrieves one column from the returned row and turns it into a tuple. We then take a slice of the sequence (with Python's convenient [0:5] notation) to grab a subset of the items returned by tuple().

But we can't pass datetime() a sequence; rather, it is expecting a number of individual elements. In other words, datetime() wants several numbers, not a reference or pointer to a list of numbers. We turn the tuple into its individual elements with Python's * operator. Finally, sharp-eyed readers will notice that we have passed the tzinfo argument before the individual elements of the tuple; this is because Python requires that we pass named arguments before the * operator.

What More Can We Do?

Sure enough, the result of invoking db-calendar.py is a fully iCalendar-compliant file, suitable for importing into Sunbird or any other calendar program. Moreover, simply by modifying the contents of our Events database table, we can ensure that everyone who subscribes to our calendar gets the latest version.

We can go one step further than this, modifying db-calendar.py such that it includes only certain events in its result. For example, perhaps the calendar needs to contain only events in the future; there is no need to clutter someone's calendar (and bandwidth) with events from the past. By adding a simple WHERE clause to our SQL query, we easily can remove all of the events from the past.

More intriguing is the possibility of supporting different groups and access levels to a calendar. HTTP supports authentication with user names and passwords, and although Sunbird doesn't support such protections at the present time, I would expect it (and other programs) to do so in the future. Given that a CGI program easily can determine the user name of the person making an authenticated HTTP request, it's not too far-fetched to say that db-calendar.py could produce different output for different users, depending on a set of assigned permissions or roles.

Finally, although we have focused on iCalendar-format output for the last few months, there isn't any reason why we can turn only the contents of the database into an iCalendar file. Indeed, it's quite possible that we would want to display our events database in plain-old HTML, as well as in iCalendar. Once again, it's easy to see how we could do that using HTML tables—demonstrating once again that relational databases make it easy to display a set of data in a number of different ways.

Conclusion

This month, we have seen how to use a database to store event information that eventually will be transformed into an iCalendar-compliant file. Using a database makes us not only more confident that stored data is valid, but it allows us to create dynamically generated files quickly and easily that are suitable for use in programs that use the iCalendar format.

Resources for this article: www.linuxjournal.com/article/8263.

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.

Linux as an Ethernet Bridge

It passes packets like a bridge and filters like a firewall. Put extra security and versatility in front of any server or device, without reconfiguring it.

BY JIM ROBINSON

Have you ever been asked to secure a router over which you did not have administrative control? What about when you are on a network you don't own but want to secure the segment you are using? A request similar to this one is what brought me to the wonderful world of Bridge, the Linux Ethernet bridging project.

According to the Bridge Web site:

> Ethernet bridging is a way to connect networks together to form a larger network. The standard for bridging is ANSI/IEEE 802.1d. A bridge is a way to connect two separate network segments together in a protocol-independent way. Packets are forwarded based on Ethernet address, rather than IP address (like a router). Since forwarding is done at Layer 2, all protocols can go transparently through a bridge.

The code currently is maintained by Stephen Hemminger for both the Linux 2.4 and 2.6 kernels. Most modern distributions using the 2.6 series kernel have the bridging code built in. For the purposes of this article, we are using Fedora Core 3, which is built on the 2.6 kernel. If you're stuck with the 2.4 kernel, don't despair. Kernel patches are available on the Bridge site (see the on-line Resources), so you can play too.

The firewall component of the bridging firewall is achieved by using another related project called ebtables. The ebtables program is a filtering layer for a bridging firewall. The filtering connects into the Link Layer Ethernet frame field. In addition to filtering, you also may manipulate the Ethernet MAC addresses. The ebtables code also allows iptables rules to function in bridging mode, giving you both IP- and MAC-level filters for your firewall.

What Is a Bridge?

A bridge is a device that links two or more network segments that use the same network technologies. The topologies may differ, though, so you can go from fiber to copper, but the technologies must remain the same. In its most simple form, think of a Linux hub. Add as many ports to the box as you want, and they all become part of the single hub device. What comes in one port goes out all of the other ports in the hub fabric, unless you state otherwise in the rules. Once your hub is up, you can use iptables and ebtables to filter traffic as you would any other Linux forwarding system.

Getting Started

We start out simply by attempting to achieve connectivity between a simple two-NIC machine. When we are finished, this Linux box should act as a standard hub, passing traffic from one port to another as needed. When we plug one NIC in to our regular network jack and a laptop in to the second NIC, we will be able to use the network from the laptop as if we were connected directly.

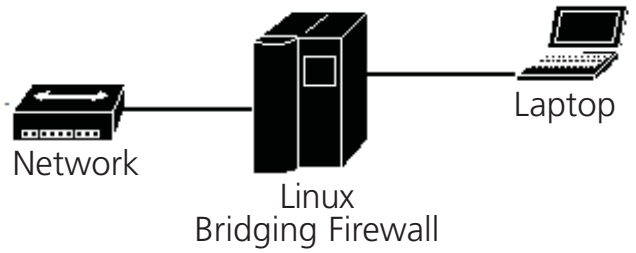


Figure 1. In this simple network, the Linux system acts like an Ethernet hub, passing all traffic.

We want this bridge to be transparent to any device plugged in to it. Interestingly enough, beyond the ability to connect remotely to the bridge to maintain it and check logs, there is no requirement to give the bridge an IP address. Of course, in today's connected world it makes sense to assign an IP address and we do so here.

I started with an old box that has been waiting for a project such as this. It's an AMD K6-450 with 256MB of RAM. It has a single 15GB IDE hard drive and a single 3Com 10/100MB Ethernet card. I also had a spare 3Com 10/100MB Ethernet card that works well with Linux, so it is added as the second interface. I am going to run only the bridge software, some simple firewall rules and perhaps Snort for intrusion detection. The traffic volumes are low and I don't expect massive amounts of Snort data, so 256MB of RAM should suffice. If you're going to be passing gigabit traffic and want to sniff live, ramp up the specs of the machine considerably.

Now install Fedora Core 3, selecting the extras you feel are needed. If you work in high-security environments, I recommend keeping your software options to the bare minimum. You always can grab extras later with YUM if you forget something. For now, simply get a working Linux install going and make sure that it finds your network cards. You need the kernel source and usual compile utilities to make the ebtables code, so add those in. Remember to stay secure and remove any software you don't need once you place the device into production. Once the install completes, reboot and log in as root.

Now you are ready to create a virtual network device. You can call it whatever you want; I went with br0—the first bridge device:

```
#> brctl addbr br0
```

Run `ifconfig`. Do you see your network interfaces (Listing 1)? In Listing 1, you can see that we have two network cards with no IP addresses bound to them. If you have IP addresses

The Straight Talk People℠
SINCE 1991

ABERDEEN

YOU CAN BUY THESE NOW OR WAIT TILL DELL FREEZES OVER

ABERDEEN STONEHAVEN A141

1U Dual Opteron™ 4 SATA/SCSI

High performance dual server for top-of-the-line processing power with ultra-dense storage capacity.

- Dual AMD Opteron™ Processors w/HyperTransport and 1MB Cache
- AMD 8000 Series Chipset w/64-bit Support
- Up to 16GB DDR-400 Reg. ECC Memory
- Up to 4 x 400GB (1.6TB) Hot-Swap SATA or 4 x 300GB (1.2TB) Hot-Swap SCSI Drives
- 400W AC Power Supply w/PFC
- **5-Year Limited Warranty**

Starting at **$1,895**

ABERDEEN STONEHAVEN A261

2U Dual Opteron™ 6 SATA/SCSI

The highest performing 2U server available for the money. "Staggering ... Powerhouse Performance ... Highest Webbench numbers we've seen to date" – *PC Magazine, December 27, 2004.*

- Dual AMD Opteron™ Processors w/HyperTransport and 1MB Cache
- AMD 8000 Series Chipset w/64-bit Support
- Up to 16GB DDR-400 Reg. ECC Memory
- Up to 6 x 400GB (2.4TB) Hot-Swap SATA or 6 x 300GB (1.8TB) Hot-Swap SCSI Drives
- 460W Hot-Swap Redundant Power Supply
- **5-Year Limited Warranty**

Starting at **$2,875**

ABERDEEN STONEHAVEN A381

3U Dual Opteron™ 8 SATA/SCSI

Gargantuan storage beast with a capacity of up to 3.2TB, room for a dual-height tape drive, at an incomparable cost/TB ratio.

- Dual AMD Opteron™ Processors w/HyperTransport and 1MB Cache
- AMD 8000 Series Chipset w/64-bit Support
- Up to 16GB DDR-400 Reg. ECC Memory
- Up to 8 x 400GB (3.2TB) Hot-Swap SATA or 8 x 300GB (2.4TB) Hot-Swap SCSI Drives
- 760W Hot-Swap Redundant Power Supply
- **5-Year Limited Warranty**

Starting at **$2,975**

ABERDEEN STONEHAVEN A124

1U Quad Opteron™ HPC

64-bit HPC environment workhorse server/cluster node. Superior cooling with plenty of power to handle any project.

- Quad AMD Opteron™ 800 Series Processors
- AMD 8000 Series Chipset w/64-bit Support
- Up to 32GB DDR-400 Reg. ECC Memory
- Up to 2 x 300GB (600GB) SCSI Hard Drives
- 500W Power Supply
- Ultra Cool with Superb Air Flow
- **5-Year Limited Warranty**

Quads Starting at **$8,265**

ABERDEEN STONEHAVEN A234

2U Quad Opteron™ 3 SATA/SCSI

Robust 64-bit server ideal for the HPC environment as a high performance server. Able to provide all the power and I/O for large databases and memory intensive projects.

- Quad AMD Opteron™ Processors w/HyperTransport and 1MB Cache
- AMD 8000 Series Chipset w/64-bit Support
- Up to 32GB DDR-400 Reg. ECC Memory
- Up to 3 x 400GB (1.2TB) Hot-Swap SATA or 3 x 300GB (900GB) Hot-Swap SCSI Drives
- 700W Power Supply
- Ultra Cool with Superb Air Flow
- **5-Year Limited Warranty**

Quads Starting at **$8,125**

ABERDEEN STONEHAVEN A484

4U Dual Opteron™ 8 SATA/SCSI

Best of both worlds, all-inclusive server with enterprise-class 64-bit HPC Quad power along with maximum storage capacity.

- Quad AMD Opteron™ Processors w/HyperTransport and 1MB Cache
- AMD 8000 Series Chipset w/64-bit Support
- Up to 32GB DDR-400 Reg. ECC Memory
- Up to 8 x 400GB (3.2TB) Hot-Swap SATA or 8 x 300GB (2.4TB) Hot-Swap SCSI Drives
- 950W 3+1 Hot Swap Redundant Power Supply
- Ultra Cool with Superb Air Flow
- **5-Year Limited Warranty**

Quads Starting at **$9,625**

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

Listing 1. Before configuring the network, check that both Ethernet interfaces are up.

```
#> ifconfig

eth0      Link encap:Ethernet  HWaddr 00:CC:D0:99:EB:26
          inet6 addr: fe80::2b0:d0ff:fe99:eb26/64 Scope:Link
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:86208855 errors:0 dropped:0 overruns:63 frame:0
          TX packets:77098217 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3871506445 (3692.1 Mb)  TX bytes:266311184 (253.9 Mb)
          Interrupt:5 Base address:0xec00

eth1      Link encap:Ethernet  HWaddr 00:CC:03:D8:3A:1A
          inet6 addr: fe80::201:3ff:fed8:3a1a/64 Scope:Link
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:77087614 errors:0 dropped:0 overruns:0 frame:0
          TX packets:85110321 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:264995582 (252.7 Mb)  TX bytes:3672580334 (3502.4 Mb)
          Interrupt:9 Base address:0xec80
```

assigned to the interface, remove them for simplicity's sake. On Fedora, edit the file /etc/sysconfig/networking-scripts/ifcfg-X, where X is the card identifier. On my system, the two interfaces are eth0 and eth1. Delete or comment out the lines that relate to the IP address. It is important to make sure the cards are on at boot time. Listing 2 shows a basic configuration that should work. Don't forget to reinitialize networking once you've completed the above, using `service network reload`.

Listing 2. Two Simple Config Files for Network Cards with No IP Addresses

```
/etc/sysconfig/networking-scripts/ifcfg-eth0:

DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static


/etc/sysconfig/networking-scripts/ifcfg-eth1:

DEVICE=eth1
ONBOOT=yes
BOOTPROTO=static
```

Next, tell the system what devices belong to this group, as shown below. Also, give the command that actually initializes the virtual device, as shown in the last line:

```
#> brctl addif br0 eth0
#> brctl addif br0 eth1
#> ip link set br0 up
```

In its most basic form, your Linux box now is acting like a hub. For the keen ones, you can plug in the Ethernet adapters and begin to play. The box itself, however, currently is passing traffic blindly and does not have an IP address assigned to it. I like to be able to connect remotely to my devices after I install them, so I am going to add an IP address and some routing information to the virtual device br0.

To add an IP address to the bridge interface, issue:

```
#> ip addr add 10.1.1.18/16 brd + dev br0
```

I had to state both the subnet mask (/16) and which bridge device it should be assigned to. This becomes important if you have more than one virtual device on the machine. I have only the one, but the syntax requires it. If you named your bridge device something else, you need to state that explicitly here.

The last thing to do before you can play with your bridge remotely is to configure the routing:

```
#> route add default gw 10.1.1.1 dev br0
```

The usual routing rules and commands apply, and for all intents and purposes you can use the device (br0) as you would any other Linux network interface.

Testing

Now that we have everything in place, let's test it out. First, let's confirm that all of our configurations have taken hold:

```
#> brctl show
bridge name  bridge id           STP enabled  interfaces
br0          8000.0030843e5aa2   no           eth0
                                              eth1
```

As you can see above, we have a single bridge device called br0 that uses interfaces eth0 and eth1. This confirms that we should be in business.

Installation

Now it's time to do the physical setup. Connect one network card to your network switch as you would normally do for any other computer. You should see link lights on both ends of the link. Connect a desktop or laptop to the other interface on your Linux box using a crossover cable. Wait for the link lights to come on, count to ten and ping another node on your network from your desktop or laptop. You should be able to use the network on the other side of the Linux hub as if it were attached directly.

Surviving a Reboot

How you set up your install to survive a reboot is your choice. A simple way is to add all of the commands we have used to /etc/rc.local, which is processed at the end of startup. Enter the commands used above to this file, and your bridge is functional after startup.

Appro *Xtreme*Blade Solution

Clusters. High performance, high availability.

The Most Powerful InfiniBand™ Enabled Blade Solution

Fits up to 12 Blades in a 7U Height Subrack

Analyst Firm IDC examines the Appro *Xtreme*Blade based on AMD Opteron™ processors to address HPC and Enterprise needs. In addition, IDC provides good information on technologies that are experiencing growth in adoption such as blade servers. Get this IDC White Paper at http://www.appro.com.

- **Scalable** - 2-way and/or 4-way configuration
- **Flexible** - Ability to mix blade configurations in one rack cabinet
- **Connected** - Integrated Infiniband™ and Gigabit switches
- **Powerful** - AMD Opteron™ based server with leading 32-bit performance with 64-bit capabilities
- **Reliable** - Hot swappable blades, redundant switches, power supplies and cooling fans
- **Smart** - BladeDome II – Centralized remote blade server and system management
- **Balanced Architecture** - Memory, I/O and communication bandwidth match CPU bandwidth

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

The ability of Linux to slide transparently into existing network infrastructure opens a world of new and improved services that the penguin can provide.

Firewalling

As with any Linux install that passes or forwards traffic, you have the ability to filter the stream of information as it passes by. A bridging firewall is no different. There are many ways to create and maintain firewall configurations. Below, I explain how to use the most basic firewall type: deny all, pass some. We want to deny everything passing this firewall unless we specifically state that something is allowed.

This firewall configuration requires you to download and install the ebtables user-space tools available from the ebtables Web site (see Resources). At the time of this writing, the latest release was v2.0.6. Grab a copy of this from one of the many mirrors. Do the usual extract and install dance without the initial configure step:

```
#> tar -xzf ebtables-v2.0.6.tar.gz
#> cd ebtables-v2.0.6
#> make
#> install
```

If all goes well, you should have the ebtables command set at your fingertips. Test this by typing `ebtables` at the prompt; you should see something similar to this:

```
#> ebtables -V
ebtables v2.0.6 (November 2003)
```

Let's start by making sure iptables is set to accept. Remember we're on Fedora Core 3, so we simply can tell the service to quit, which does the same thing:

```
#> service iptables stop
#> chkconfig --level 35 iptables off
```

You can do something similar by issuing the flush command. List your available chains and then flush each of them in turn:

```
#> iptables -L
#> iptables -F INPUT
#> iptables -F OUTPUT
#> iptables -F FORWARD
#> iptables -F RH-Firewall-1-INPUT
```

Now we want to stop all traffic from all areas of our network from passing through the firewall. The following rules are specific to the network we're working with for this example; you need to amend the subnets or hosts to reflect your specific requirements:

```
/sbin/ebtables -A FORWARD -p IPv4 \
--ip-source 10.2.0.0/16 -j DROP
/sbin/ebtables -A FORWARD -p IPv4 \
--ip-source 10.7.0.0/16 -j DROP
/sbin/ebtables -A FORWARD -p IPv4 \
--ip-source 10.4.0.0/16 -j DROP
/sbin/ebtables -A FORWARD -p IPv4 \
--ip-source 10.5.0.0/16 -j DROP
/sbin/ebtables -A FORWARD -p IPv4 \
--ip-source 10.6.0.0/16 -j DROP
/sbin/ebtables -A FORWARD -p IPv4 \
--ip-source 10.1.0.0/16 -j DROP
```

Those of you familiar with iptables should notice that the syntax above is similar. We tell the ebtables program that when FORWARDING using the IPv4 protocol to DROP any packets sourced from the 10.1.0.0/16 subnet. We then tell it to repeat for the rest of the subnets.

The next step is to allow the device behind the firewall itself. If you do not allow its IP address to pass through, nothing works. Also, if you assign an IP address to the firewall itself, don't forget to allow it as well:

```
/sbin/ebtables -I FORWARD 1 -p IPv4 \
--ip-source 10.1.1.5 -j ACCEPT
/sbin/ebtables -I FORWARD 1 -p IPv4 \
--ip-source 10.1.1.18 -j ACCEPT
```

Here, I add the devices on my network that are allowed to access my laptop:

```
/sbin/ebtables -I FORWARD 1 -p IPv4 \
--ip-source 10.1.10.30 -j ACCEPT
/sbin/ebtables -I FORWARD 1 -p IPv4 \
--ip-source 10.1.10.19 -j ACCEPT
/sbin/ebtables -I FORWARD 1 -p IPv4 \
--ip-source 10.1.10.87 -j ACCEPT
```

To test this, I simply go to a machine listed in the ACCEPT rules above and see if I can ping my laptop at 10.1.1.5. Now move to a node not listed above—no pings for you!

Real-World Implementation

Recently, I was called to a customer's site to secure a financial server. The request was simple: we need a firewall in front of this system but we cannot change its IP address. With two NICs and a Linux OS, I was able to have a working firewall up and running in a few minutes. Installation also was a breeze. I simply used a crossover cable that connected the firewall to the server and a regular cable from the other network card on the firewall to the network jack. That was it. No redesign was necessary of any part of the existing IP scheme; it truly was plug and play. Once a few rules were in place to drop all packets unless they were from the IP addresses and ports listed as acceptable, the project was completed.

One of the beautiful aspects of Linux is its ability to run many services on one system. Take the above example. I

quickly firewalled a sensitive server, but that was not the end of the project. With all the extra time and money we saved using Linux, we were able to load Snort on the firewall. With a quick hack to the sniffer's config file—/etc/snort.conf in our case—we told Snort to listen to interface br0, and snort immediately began to do its stuff on the bridging interface.

This is where the true power of the bridging code can be felt. Ever had a segment of the network running slow but you don't know why? Next time, load a Linux box with Snort and any other sleuthing software you like and get the bridge up and running. Find your trusty crossover cable and head out to the site. Because the bridge acts like a hub, you simply can insert your Linux box at any point in the network. As long as you have the physical connections, you can drop your box in and begin to sniff live in a matter of seconds. The latest project we have been working on included transparent Squid cache servers that are truly transparent requiring zero reconfiguration to the IP scheme, clients or browsers. Simply insert the Squid box in front of the router and redirect all port 80 traffic to the box itself and you're done.

The ability of Linux to slide transparently into existing network infrastructure opens a world of new and improved services that the penguin can provide. With the ability to place dissimilar networking devices into one virtual entity, you can use a single device to firewall and monitor any aspect of your network. Your only limitation is the speed of your hardware and its number of available slots.

Acknowledgements

The author sends kudos to the guys and girls that wrote the Bridging code and to the authors of the ebtables command set for producing stable, usable tools and releasing them under the GPL.

Resources for this article: www.linuxjournal.com/article/8261.

Jim Robinson is President of Linux Solutions Provider, Inc., a consulting company based in Macon, Georgia. He enjoys being a husband and a dad, playing guitar and, of course, using Linux.

Tricked-Out Terminals

Command-line users get some GUI toys with these windows that save screen space, add backgrounds and even start multiple terminals with one command.

BY MARCEL GAGNÉ

Please, François, do not cry. What is making you so sad? The restaurant looks fantastic, *mon ami*. Everything is ready. What could possibly be so bad? *Quoi?* You were reading the menu and saw that we would be featuring terminal programs and that made you sad? Ah, *Mon Dieu* François, terminal programs have nothing to do with death; they are windows to the shell, much like the GNOME terminal, xterm or the KDE Konsole.

Our guests are arriving, François. Welcome, everyone, to *Chez Marcel*, where fine Linux fare meets fine wine. Please sit and make yourselves comfortable. François, please go to the wine cellar and bring back the 1999 Côte du Rhône. *Vite!*

While putting together the menu for this Systems Administration-themed issue, I was somewhat torn. There are, after all, so many different aspects to this topic. Then, I asked myself what systems administration tool I use most and came up with an interesting answer—it's a command shell run inside some kind of terminal emulator. Although this may seem extremely basic, the terminal program has come a long way since the days of the venerable xterm.

For those of you too young to remember, we call them terminal programs or emulators as opposed to shell windows, because they were software desktop equivalents of the terminals that at one time were connected physically to the mainframe system. But I digress, and François has returned with the wine.

I've used quite a number of terminal programs over the years. Originally, it was a plain xterm, still available on most Linux systems. Dressing it up meant changing the font color or background but not much else. For instance, to start a plain xterm with a steel-blue background and red text, I would do the following:

```
xterm -background "SteelBlue" -foreground "Red"
```

In time, I discovered another terminal program called rxvt that also should be included on most systems. It gave me a little more control over configuration options. It not only looked better than my old xterm, but I could assign an XPM bitmap image for a background, which made it look pretty cool at the time:

```
rxvt -pixmap /usr/share/themes/BrushedMetal/gtk/bg.xpm -fg black
```

Then came the first terminal emulator I really fell in love with, one I still enjoy today. It's called Eterm, and although it was designed as an xterm replacement for the Enlightenment window manager, it should work with whatever you have, be it GNOME, XFCE, KDE or anything else. Check your distribution CDs for this one, as it may not be installed already on your system. You also can get the latest Eterm from the Eterm Web site (see the on-line Resources).

Besides looking great, Eterm offers some cool features. For example, Eterm can have a variety of background images, and in fact, it comes with several. Some of these are tiled pixmaps, and others are full background images (Figure 1). Simply click Background on the Eterm menu bar, then Pixmap and select from the tiled or scaled pixmaps available.

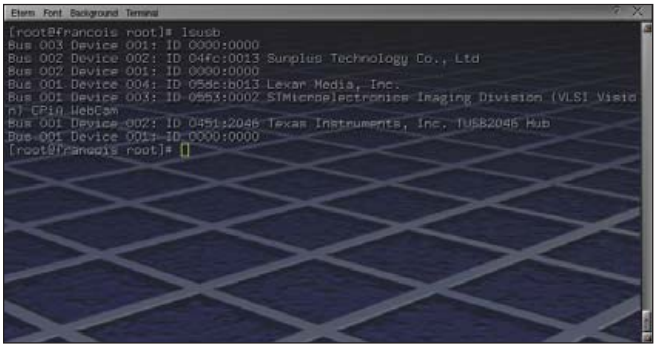


Figure 1. Eterm does transparency, backgrounds, themes and more.

You also can change the font size, brightness, contrast, scrollbar style and much more. You even can create and customize your own menu. If you are happy with the settings you have modified, click Eterm on the menu bar and select Save User Settings.

Eterm is themeable as well. Click the Themes link on the Eterm Web site, and you will see several themes that can be downloaded and installed on your system. These all are tarred and gzipped bundles. To install them, create a .Eterm directory in your $HOME directory and then create a themes directory below that. All you have to do now is extract the theme into that directory and it becomes available. To start your next Eterm with the appropriate theme, name it on the command line like this:

```
Eterm -t theme_name
```

One of my favorite features of Eterm, however, is the ability to make it transparent. I find this particularly nice because it lets me keep an eye on system logs as they scroll across my desktop and my current favorite wallpaper. All you have to do is click Background on the menu bar and select Toggle Transparency. This option doesn't work on KDE, but it does work on GNOME, Window Maker and others. To view the Apache logs on my Web server on my transparent Eterm, I would run the following command:

```
Eterm -O -e sudo tail -f /var/log/httpd/access_log
```

This doesn't have anything to do with terminal emulators per se, but in the above example, you may need to add yourself

to the /etc/sudoers file in order for this to work, unless, of course, you run the Eterm as root. In my case, I added a line in the file right below the definitions for root's privileges:

```
# User privilege specification
root    ALL=(ALL) ALL
marcel  ALL=(ALL) ALL
```

These days, there's another feature in terminal programs that has me excited, and that's tabs. There is no need to open three or four terminal applications, which can take up a fair amount of real estate. Take a look at the KDE Konsole program, for instance, or the GNOME terminal. Both provide not only basic shell access, but they also allow you to run multiple shells in tabbed sessions. To use tabs on your GNOME terminal, press Shift-Ctrl-T and you are presented with a new session (Figure 2).

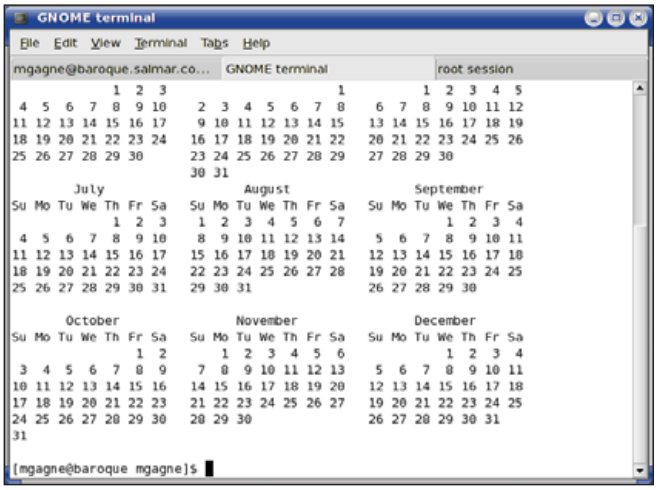


Figure 2. A GNOME terminal with tabs saves screen real estate.

The GNOME terminal also supports transparency, so if I want to start up a script that lets my logs scroll across my wallpaper *du jour*, I can do that as well. With the GNOME terminal, this is done with profiles. To create a profile, click Edit on the GNOME terminal menu bar and select Profiles. Click New and give your profile a name, for example, seethrough. When the Editing Profile dialog appears, click on the Effects tab and click the Transparent background radio button (Figure 3).

Use the slider to adjust the level of transparency and then close the profile dialog. Now, start the GNOME terminal like this:

```
gnome-terminal --window-with-profile=seethrough \
-e "sudo tail -f /var/log/messages"
```

If you like the results, use that command in a shell script in order to call it by a name that makes sense to you.

In the KDE world, there's Konsole, which also handles tabs nicely. To start a tabbed Konsole session, click Session on the menu bar and select New Shell. Then, customize the tabs by double-clicking on them and giving them a name. When you are running multiple shells—a compile here, a monitor session there and a root shell in yet another tab—naming your tabs

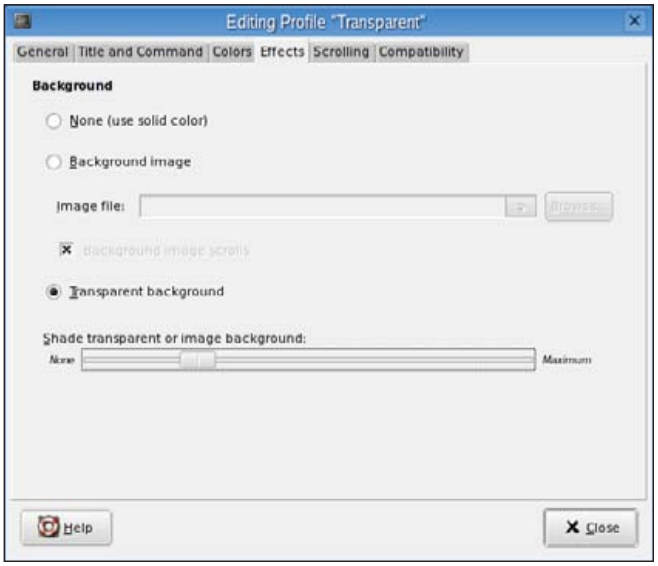


Figure 3. To create a transparent GNOME terminal, you first need to create a profile.

makes a lot of sense. Also notice that when you click the Session menu, several options are available, including starting a session in a bookmarked location (Figure 4).

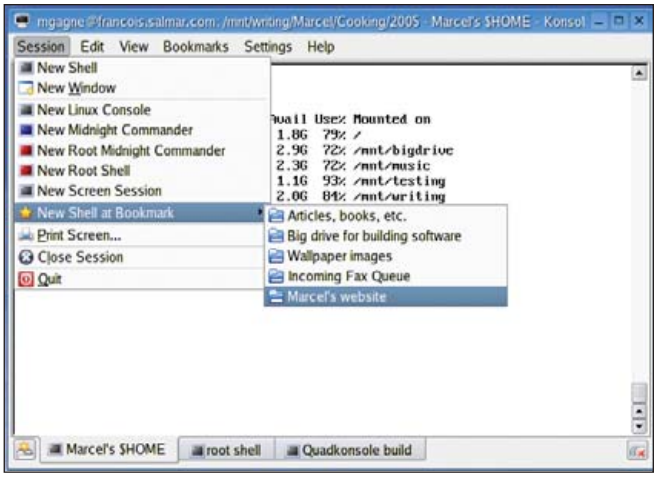


Figure 4. The amazing Konsole—tabs, bookmarks and more!

Bookmarks? As you do your job as administrator, you find yourself going to the same directories over and over again. Sure, you can type quickly and get where you want to go, but Konsole extends its capabilities by letting you assign bookmarks. When you are in that familiar directory seven levels deep, simply click Bookmarks on the menu bar and then select Add Bookmark. You can assign a name to the location you want, and open a shell in that location with a single click.

And, of course, I can't move on without mentioning transparency. To enable transparency in Konsole, click Settings on the menu bar, then Schema and select one of the Konsole transparencies from there. While you are in the Settings menu, you see a ton of configuration options for the Konsole terminal program. Change the font, go to full-screen mode and change the encoding, change the keyboard layout or set an alarm to inform you when changes take place in one of your tabbed sessions. You can explore the Konsole features on your own, but I will leave you with a nice script to start a clean, transparent Konsole to track your log file:

```
konsole --schema Transparent.schema --nomenubar \
  --notabbar --noframe \
  -e sudo tail -f /var/log/messages
```

The final selection on tonight's menu is something so great, I'm frankly amazed I hadn't seen anything like it before. I decided it was great, because I started using it regularly as soon as I discovered it. Using KDE kparts, Simon Perreault created a program called Quadkonsole. As the name implies, it starts up four Konsole programs in a grid (Figure 5). There's no need to line up your Konsoles side by side or click from one shell tab to the next. To make things even nicer, the Quadkonsole creates only one task in the task bar, leaving your Kicker panel uncluttered. That feature is going to be particularly useful as I tell you more about this great terminal program.

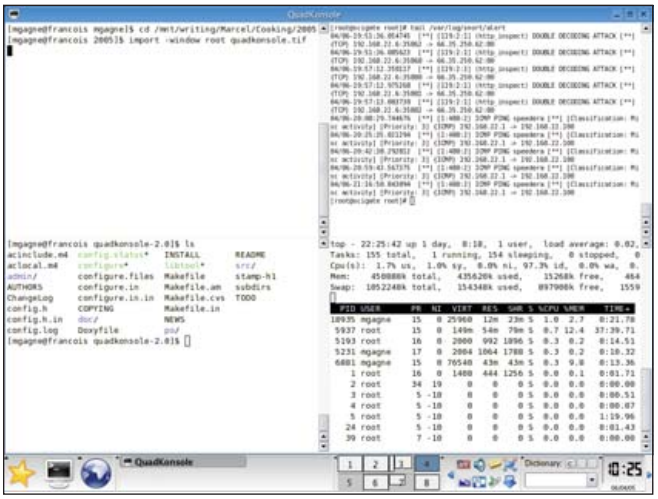


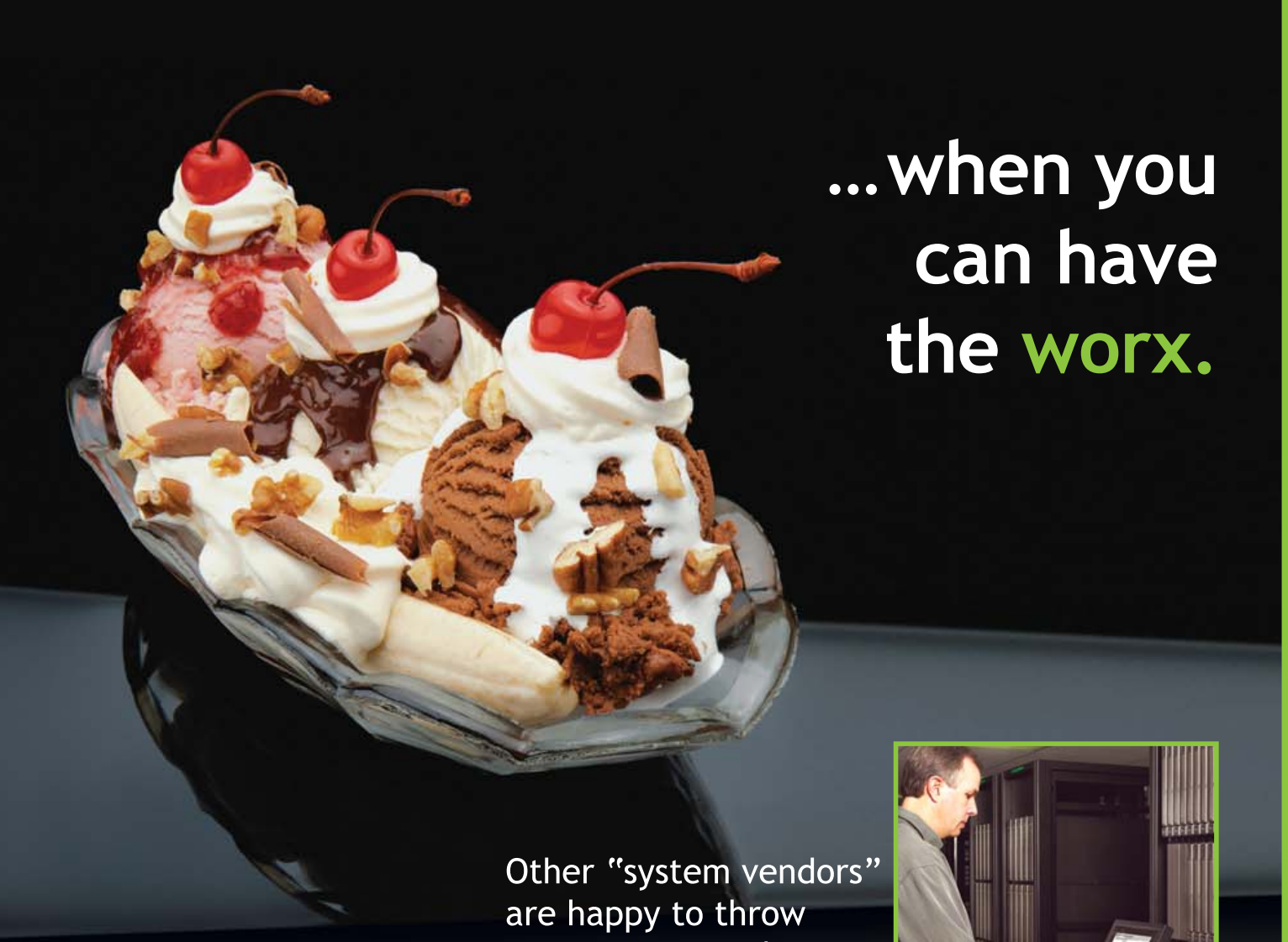
Figure 5. Quadkonsole starts up four Konsoles in a perfect grid.

Each running Konsole can be modified at will by right-clicking on it, choosing Settings and then selecting the changes you are interested in making from the menu. For instance, you may want one to be transparent, another white on black with smaller fonts and so on. After all, each one is a Konsole and can be modified accordingly.

If you are feeling particularly distracted, you can tell Quadkonsole to start with more than four Konsoles. Simply specify the number of rows and columns, as follows:

```
quadkonsole --rows 4 --columns 4
```

This command starts up Quadkonsole with 16 individual terminal sessions (Figure 6). That's why having only one instance in your Kicker panel is such a nice feature. This also might be a good time to start playing with the font size if you want the information to fit. In one of my sessions, I have top, the real-time process and resource monitor, running with a super-tiny font. Of course, now I need to run a desktop magni-



...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

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

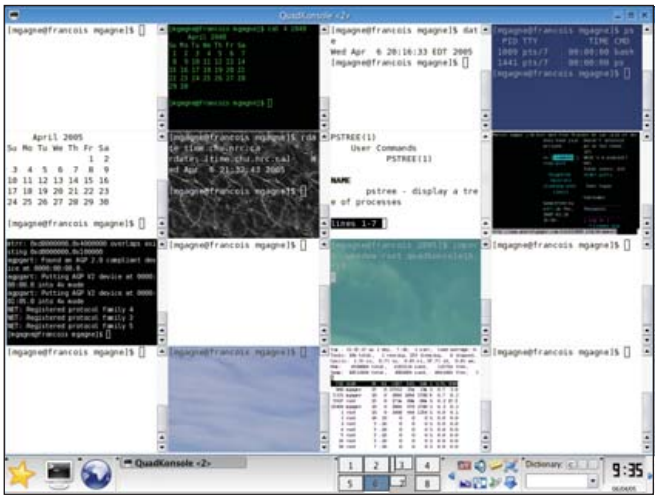


Figure 6. When too much information is just right, Quadkonsole answers the call.

fier such as kmag in order to see it.

By default, the focus follows the mouse pointer, so if you happen to nudge the mouse accidentally while typing, strange and wonderful things may occur. To override that default, use the `--clickfocus` command option to start your Quadkonsole program. After doing so, you have to click in each window in order to select a particular Konsole.

I happen to prefer the click-to-focus option, but I don't always like having to reach for the mouse each time. Luckily, Quadkonsole has that covered as well. You can navigate from one Konsole instance to another by using the Shift-Ctrl-arrow key combination. This allows you to move up, down, left or right from one session to another.

Incroyable! It would seem, *mes amis*, that closing time has once again arrived. If only someone would write an application that could transform our one instance of leisure time into four or 16. Too much wonderful wine to savor, too many programs to explore and certainly not enough time. Do not worry, *mes amis*, I am more than willing to spend a little more time, and François would be happy to refill your glasses while you chat. Sit back and enjoy that Côte du Rhône. Raise your glasses, *mes amis*, and let us all drink to one another's health. *A votre santé! Bon appétit!*

Resources for this article: www.linuxjournal.com/article/8259.

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 makes regular television appearances as Call for Help's Linux guy. Marcel 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 by e-mail at mggagne@salmar.com. You can discover a lot of other things (including great Wine links) from his Web site at www.marcelgagne.com.

Flexibility to power the enterprise.

From mail servers to databases, ZT servers powered by the 64-bit Intel® Xeon™ Processor can run the full range of 32-bit applications and offer extended flexibility for your 64-bit needs. So you can create powerful, all-purpose IT infrastructure that enhances business agility – and the bottom line.

New Powerful 64-bit Server Line

Friendly Server Specialists
Complete Solution Provider
Flexibility and Variety: Customize platforms with leading-edge products
We Build the Future !

ZT Optimum 1U Server X6715

- **Intel® Pentium® 4 Processor 630 with HT Technology - 3 GHz, 2MB L2 Cache, 800MHz FSB**
- Intel® E7221 Chipset Server Board
- 1GB DDR2 533MHz SDRAM (Up to 4GB)
- 2 x Seagate® 250GB SATA 7,200RPM Cache Hard Drive
- 4 x 1" SATA Hot-Swappable Drive Bays
- Slim 24x CD-ROM & 1.44MB Floppy Drive
- RAID 0, 1, 10 Support
- 2 x 10/100/1000 Gigabit Network Controllers
- 1U Rackmount Chassis w/350W Power Supply
- 3-Year Limited Warranty

$1,649

ZT Optimum 2U Server X9466

- **2 - Intel® Xeon™ Processor 3 GHz - 1MB L2 Cache, 800MHz FSB**
- Intel® E7320 Chipset Server Board
- 1GB ECC Registered DDR333 SDRAM (Up to 8GB)
- 2 x Seagate® 250GB SATA 7,200RPM Cache Hard Drive
- 6 x 1" Hot-Swap Serial ATA Drive Bays
- Slim 24x CD-ROM & 1.44MB Floppy Drive
- RAID 0, 1, 10 Support
- Intel® 10/100/1000 Gigabit Network Controller
- 2U Rackmount Chassis w/550W Power Supply
- 3-Year Limited Warranty

$2,299

ZT Optimum 3U Server X9467

- **2 - Intel® Xeon™ Processor 3 GHz - 1MB L2 Cache, 800MHz FSB**
- Intel® E7520 Chipset Server Board
- 1GB ECC Registered DDR-II 400 SDRAM (Up to 16GB)
- 8 x Seagate® 250GB SATA 7,200RPM Hard Drive (3.2TB Storage Max)
- 8 x 1" Hot-Swap Serial ATA Drive Bays
- 52x32x52 CD-RW & 1.44MB Floppy Drive
- 8 Port Serial ATA Controller (RAID 0, 1, 5, 10 , 50 Support)
- 2 x Broadcom 10/100/1000 Gigabit Network Controllers
- 3U Rackmount Chassis w/550W Power Supply
- 3-Year Limited Warranty

$4,299

1. OEM Computer Manufacturer

- 3 Year Warranty with Lifetime Tech Support
- Onsite Service Available.
- Phone Orders Receive Free Air Shipping. Promo Code # ZTL05

New Accounts Receive Free Gift

- Personal Attention (Dedicated Technical Sales Team)
- Reseller and volume pricing available.
- Call now to customize using the latest technology !

Find out how ZT Insider Program can help maximize your Business Solution

Go to **ztgroup.com/go/linuxjournal**

Call **866- ZTGROUP** (984-7687)

Real-Time Control of Magnetic Bearings Using RTLinux

RTLinux meets one of the engineering world's most demanding control requirements, without requiring separate controller hardware and costing substantially less than comparable proprietary systems.

BY HARLAND ALPAUGH

One of the most demanding applications of real-time control is the active magnetic bearing (AMB). Suspending a shaft rotating faster than 10,000 RPM in a magnetic field with a gap of .015 inches—roughly the size of a grain of table salt—requires precise, reliable control of the magnetic field. Magnetic suspensions inherently are unstable. Think of a trapeze artist balancing a long pole on her chin, and you can appreciate the difficulty that a control system encounters when stabilizing a magnetic bearing. Magnetic bearings provide several advantages, however, which justify their use. The major advantage is magnetic bearings eliminate physical contact between the shaft and the support, minimizing friction and eliminating the wear inherent to conventional roller bearings.

A recent application under development in Japan is an implantable heart pump. Because magnetic bearings require no maintenance, this is an ideal situation in which to use them. For the same reason, satellite wheels also are a logical application.

The Test Setup

The experimental test rig, shown in Figure 1, consists of two eight-pole laminated stator assemblies, with individual windings on each of the poles. The bearing assembly includes inductive gap sensors at the centerline of each pair of diametrically opposed poles. These bearings support a two-foot-long shaft driven by a brushless DC motor. Figure 2 shows a schematic of a single axis. The controller uses the signal from the gap sensor to adjust the current from the power amplifiers driving the magnetic coils to keep the rotating shaft centered in the gap. The original controller was an analog circuit, which was replaced by the digital controller. The capability of running either the analog or digital controller is retained. The digital controller is implemented on an Intel Pentium III PC with a multichannel data acquisition board and a multichannel analog output board. The PC is configured as a dual-boot system, and the user selects plain Linux or RTLinux at startup.

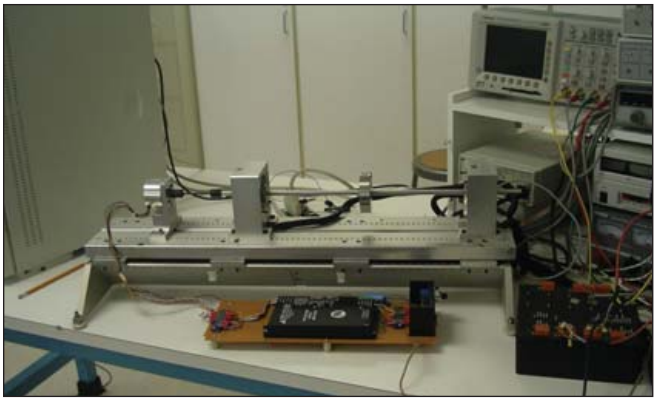


Figure 1. The Experimental Magnetic Bearing Test Rig

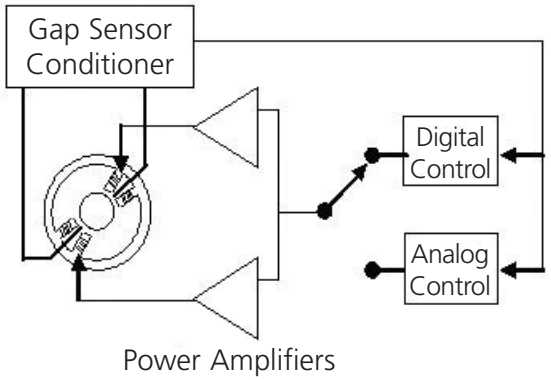


Figure 2. Schematic of a Single Axis of the Magnetic Bearing

A real-time OS must ensure that a specific task executes at a fixed rate, regardless of the many system-level demands that burden the OS. To meet this requirement, two organizations, FSMLabs and RTAI, have developed special-purpose kernels that run Linux as a low-priority task within a real-time OS. This substantially reduces the timing from the hundreds of milliseconds on desktop systems to the microsecond range. It also allows the user to control precisely the timing of critical control processes.

For this magnetic bearing project, I selected the free RTLinux implementation from FSMLabs. RTLinux, developed by Michael Barabanov and Victor Yodaiken in 1996, currently is marketed by FSMLabs, a private company located in New Mexico. FSMLabs provides two versions of RTLinux, including RTLinux/Free, which I used for this project. FSMLabs holds a software patent on RTLinux, but the patent's license allows it to be used in projects licensed under the GNU GPL.

Conceptually, RTLinux splits the OS into user space and a real-time kernel. You may think of these as two separate cities, walled off from each other and able to communicate only by

guarded pathways, such as real-time, first-in-first-out devices (RT-FIFOs). User space is the familiar Linux system with all its friendly utilities, such as the vi editor, the GCC compiler and the shutdown command. The real-time kernel is the Spartan environment that relentlessly executes the real-time task regardless of the activities in user space.

Real-time programs are coded as kernel modules and do not use the main{} program construct of user-space C programs. The module requires two functions: init_module, which is called when starting, and cleanup_module, which is called when turning off the real-time module. The init_module creates the entry point for the real-time module and allocates the RT-FIFOs used to communicate with user space. To start the real-time module, use the insmod command. Once the real-time module starts, it can be stopped only by issuing the rmmod command or by pulling the plug on the processor. As a new user of RTLinux, I was quite unnerved to discover that despite issuing the shutdown command, the controller continued to run.

A Control Theory Palimpsest

Control theory, central to all modern technologies from the automobile to the jetliner, is an extensive field in which graduate students have toiled for many decades. I cannot cover this extensive body of theory here, but I can explain the essentials of the digital control for the magnetic bearing. First, the quantity to be controlled is instrumented and measured. In this case, the quantity is the gap between the rotating bearing and the magnetic poles of the bearing. This gap is converted to a voltage with signal conditioners and input to an analog/digital input (AI) board. In my setup, four separate gap sensor signals control the rotating shaft. All four gap signal voltages are sampled simultaneously.

The gap is controlled by the current traveling through the magnets, which are driven by eight power amplifiers. The power amplifiers are controlled by the voltage from a separate digital/analog output (AO) board. The AO board receives a digital input and converts it to a voltage that is held constant until the next signal. This sample-and-hold operation is fundamental to all digital control systems. In the control loop, the AO board receives the processed signals from the AI board after numerical processing. In an ideal digital controller, both AI and AO operations occur simultaneously at precise constant intervals. Although impossible to achieve this ideal, you must ensure that the code within the control algorithm runs efficiently. In my control program this occurs at 10kHz.

The numerical operations within the control program include the history of the input, x, and the output, y, of the controller for several previous steps. These are stored in memory and shifted one increment each time the control loop executes. The history is incorporated in a difference equation:

$$y(n)=A*y(n-1)+B*y(n-2)+...+C*x(n) + D*x(n-1) +...$$

where y(n) is the output of the controller for the current time step, y(n-1) is the output of the controller in the previous time step, y(n-2) is the output two steps in the past, y(n-3) is three steps in the past and so forth to the depth demanded by the sophistication of the control algorithm. Similarly, x(n) is the input voltage for the current time step, and x(n-1) is the input for the previous step. A, B, C, D and the rest are constant coefficients determined by the particular control law implementation. Controllers are either single-input-single-output (SISO) or multiple-input-multiple-output (MIMO). In my magnetic bearing test setup, y is the voltage driving the power amplifier, and x is the signal from the gap sensor. I use the three previous values in my magnetic bearing difference equation.

Digital Control Implementation

The digital controller is implemented on an Intel Pentium III PC operating at 1GHz with a six-slot PCI bus. The system was procured as a customized desktop personal computer with Red Hat Linux version 7.2 installed. In the laboratory the PC is not networked. I installed version 3.1 of RTLinux from a tar

Listing 1. Real-Time Code Skeleton to Test A/D and D/A Conversion

```
#include <stdlib.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <rtl.h>
#include <time.h>
#include <rtl_fifo.h>

// #includes for DAQ cards here

#define PERIODIC_FREQ_HZ         10000.0
#define FRAME_PERIOD_NS
  ((hrtime_t)((1.0/PERIODIC_FREQ_HZ)*1000000000.0))

pthread_t periodic_thread;
int AI_board_handle = 0;
int AO_board_handle = 1;

void *Periodic_entry_point(void)
{
     static double volts[MAX_RESULT_QTY+1];
     static u16 adc_data[MAX_RESULT_QTY];
     static u16 dac_data0;
pthread_make_periodic_np(pthread_self(),
                       gethrtime(),FRAME_PERIOD_NS);
pthread_setfp_np(pthread_self(),1);

// Initialize DAQ boards here

while (1)
{
    pd_ain_get_samples(AI_board_handle,
                                MAX_RESULT_QTY,
                                adc_data, &samples);

  for (i = 0; i < (CL_SIZE+1); i++)
   {volts[i+1]=
     ((float)(adc_data[i]^0x8000)*.00030518)-10.0;}
    //  output to AO board
        dac_data0 = (volts[1]+10.0)*3276.6;
        ret = pd_ao32_writex(AO_board_handle, 0,
                                  dac_data0,0,0);
      pd_ain_sw_cl_start(AI_board_handle);
   // multiplies for timing test
         for (i = 0; i < 5000; i++) z=x*y;
         pthread_wait_np();
     }
}

int init_module(void)
{
   pthread_attr_t attrib;
   sched_param.sched_priority
     sched_get_priority_max(SCHED_FIFO);
   pthread_attr_setschedparam(&attrib,&sched_param);
// create the thread
   pthread_create(&periodic_thread, &attrib,
                      Periodic_entry_point,NULL);
   pthread_wakeup_np(periodic_thread);
}

void cleanup_module(void)
{
     pthread_delete_np(periodic_thread);
}
```

archive downloaded from FSMLabs.

The three possible approaches for selecting the digital acquisition and control (DAC) boards are to write the required board driver software, to obtain a driver from an open-source project such as Comedi and to use vendor-supplied driver software. The first and second options require a high level of sophistication and expertise with using Linux and data acquisition programming. The second option reflects the open-source nature of the Linux system, but the selection of vendors is limited and the latest products often are unavailable. The third option, although it requires the least expertise, places the user at the mercy of the board vendor. The vendors supplying and supporting the necessary drivers are limited and quite often use the same sources as the second option. In the end, I chose the third option and purchased two PCI bus multichannel DAC boards from United Electronics, Inc. These came with the required RTLinux drivers.

Before implementing the digital control law, I performed tests to characterize the digital system behavior. These tests are various program codes that evaluate conversion and timing interactions of the digital boards. For my primary functional test, I designed and coded a C language module that reads the analog data on the analog input board, converts it to floating-point variables, converts it back to a digital variable and then outputs the signal by way of the analog output board.

Listing 1 shows the skeleton of the C program for the primary functional test. At the heart of the real-time control program is the RTLinux function, pthread_wait_np, which suspends execution of the currently running real-time thread until the start of the next period. This thread is marked for execution using pthread_make_periodic_np. The thread gives up control until the next time period. The default arithmetic in RTLinux is integer. My control application requires floating point, which is turned on by pthread_setfp_np. A comparison of the input and output is recorded on a Tektronix two-channel digital storage oscilloscope. Figure 3 shows a typical record of system performance on this test. The main loop in the software is set at 10kHz in this plot; the analog input is a 1,000Hz sawtooth. The output shows the step waveform characteristic of sample-and-hold operation.

A more intuitive way to test the operation of the system is to hook the analog output to an audio amplifier and speaker and input a sine wave from an external signal generator. I did this, and the tone was clean and steady. The original analog controller is a simple lead-lag filter on each of the four bearing axes, which is duplicated with the difference equation:

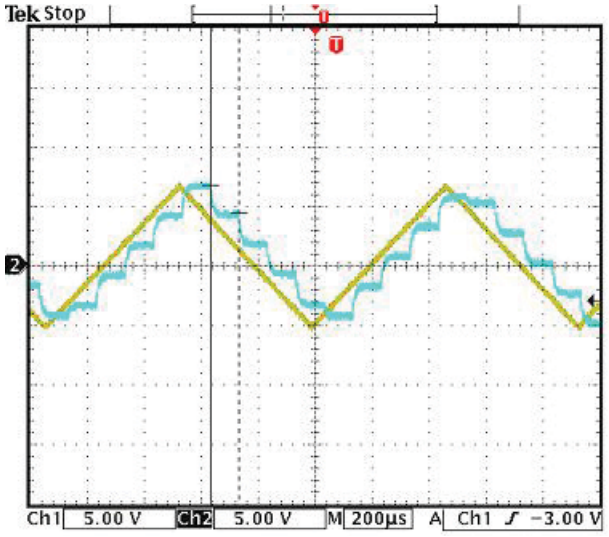


Figure 3. Oscilloscope Trace Showing Both the Test Signal (Yellow) and Analog Output (Blue)

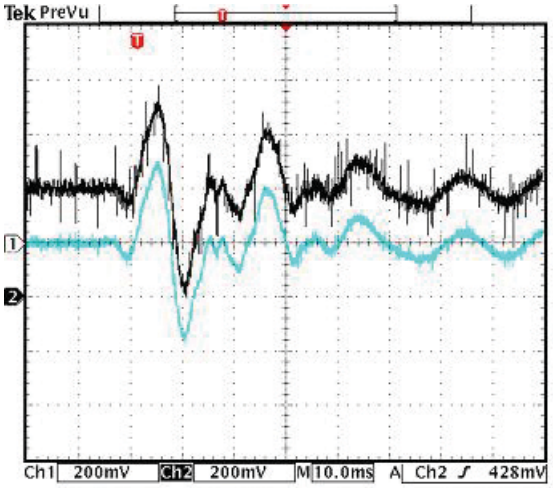


Figure 4. Oscilloscope Trace Comparing Digital (Upper) to Analog (Lower) Controller Response to a Mechanical Impulse

$$[\, y(n) = 0.7467*y(n-1) + 4.6380*x(n) - 4.5189*x(n-1)]$$

Figure 4 shows the digital and analog controller responses to a mechanical impulse on the shaft. The responses are virtually identical. In this figure, the digital controller response is shown at the top, and the analog controller response is at the bottom. The time shown in the figure is 100 milliseconds. The digital loop is operating at 10kHz in a MIMO configuration, which has five input channels and eight output channels.

Advanced Experiments

Alternate control laws are implemented easily in C and experimentally verified. One of the more robust of these is shown in the difference equation:

$$y(n) = 1.4934*y(n-1) - 0.5576*y(n-2) + 0.5795*x(n) + .01487*x(n-1) - 0.5646*x(n-2)$$

The rotor has spun up to 11,000 RPM successfully, with the AMB under full digital control, passing through a critical speed at 2,700 RPM. In virtually all rotating machinery, from the humblest hair dryer to the modern passenger jet engine, critical speeds occur at distinct RPMs. At these critical speeds, the vibration of the rotating shaft grows large and places high loads on the bearings and other components. These present extreme tests for the bearings.

To change the coefficients of the control law while the rig is operating, I used RT-FIFOs. These are first-in-first-out files for communicating between Linux user space and RTLinux threads. Because RT-FIFOs are unidirectional, I created two separate files for two-way communication with the control module. Function rtf_create(fifo_id_no, fifo_length) allocates a buffer of the specified size for the specified FIFO (/dev/rtf0, /dev/rtf1,.../dev/rtf64). It must be called from init_module(). Function rtf_destroy deallocates the FIFO at the completion of execution. It can be called from init_module( ) or clean-up_module(). This allows me to change the control law on the fly by changing the difference equation coefficients while the real-time module is running. Using the function rtf_get(fifo_id,&variables,sizeof(variables)) within the real-time thread reads the coefficients in a nonblocking mode. The user-space code for sending the coefficients to the real-time module is:

```
ctl = open("/dev/rtf1",O_WRONLY);
write(ctl,&coeffs,sizeof(coeffs));
ctl = close(ctl);
```

Listing 2. Code Snippet at the Heart of Control Module

```
while (1)
{
// Read in new coefficients on control FIFO
  rtf_get(CONTROL_FIFO_ID,&coeffs,sizeof(coeffs));
  rtf_flush(CONTROL_FIFO_ID);

// This code places coeffs[] in A, B, C, D, E ...

   pd_ain_get_samples(AI_board_handle,NO_CHANNELS,
                              adc_data,  &samples);
  for (i = 0;i < (NO_CHANNELS);i++)
   volts[i+1]=
   ((float)(adc_data[i]^0x8000)*.00030518)-10.000;

// Difference equation
        x0[0]=volts[1];
        x1[0]=volts[2];
        x2[0]=volts[3];
        x3[0]=volts[4];
        y0[0] =d1*x0[0]+stuff0;
        y1[0] =d2*x1[0]+stuff1;
        y2[0] =d1*x2[0]+stuff2;
        y3[0] =d2*x3[0]+stuff3;
        y4[0] = -y0[0];
        y5[0] = -y1[0];
        y6[0] = -y2[0];
        y7[0] = -y3[0];

//  output to AO board
        dac_data[0] = (y0[0]+10.0)*3276.6;
        dac_data[1] = (y1[0]+10.0)*3276.6;
        dac_data[2] = (y2[0]+10.0)*3276.6;
        dac_data[3] = (y3[0]+10.0)*3276.6;
        dac_data[4] = (y4[0]+10.0)*3276.6;
        dac_data[5] = (y5[0]+10.0)*3276.6;
        dac_data[6] = (y6[0]+10.0)*3276.6;
        dac_data[7] = (y7[0]+10.0)*3276.6;
  for (j = 0;j < 8 ;j++)
    pd_ao32_write(AO_board_handle,j,dac_data[j]);

// Perform shift operations
     for (i = 3; i > 0; i--)
          {
           x0[i]=x0[i-1];
          x1[i]=x1[i-1];
          x2[i]=x2[i-1];
          x3[i]=x3[i-1];
           y0[i]=y0[i-1];
           y1[i]=y1[i-1];
           y2[i]=y2[i-1];
           y3[i]=y3[i-1];
           }

// Setup difference equations
           stuff0 = A*y0[1]+B*y0[2]+C*y0[3]
                      +E*x0[1]+F*x0[2]+G*x0[3];
           stuff1 = A*y1[1]+B*y1[2]+C*y0[3]
                      +E*x1[1]+F*x1[2]+G*x1[3];
           stuff2 = A*y2[1]+B*y2[2]+C*y0[3]
                      +E*x2[1]+F*x2[2]+G*x2[3];
           stuff3 = A*y3[1]+B*y3[2]+C*y0[3]
                      +E*x3[1]+F*x3[2]+G*x3[3];

// end this thread until next periodic call
           pthread_wait_np();
           pd_ain_sw_cl_start(AI_board_handle);
// multiplies for time test
            for (i = 0; i < 5000; i++) qq=a*b;
  }
```

This code is embedded in an NCURSES interface, which allows me to change the coefficients with manual entries as the rig is rotating. The NCURSES Programming How-To by Pradeep Padala (see the on-line Resources) is an excellent resource for this work.

In a similar way, I can access the data stream in the control program and send it to user space. The appropriate function in the real-time module is rtf_put(framerate_rtfifo_id, volts, offset). In user space, send the output to a file with `cat /dev/rtf0 > file`. A simple C program to convert the file to a readable form must be written.

Conclusion

RTLinux is used to control a working rotor test rig at Tufts University. The controller is realized on a conventional Pentium III personal computer using the RTLinux extension of the Linux operating system. The control algorithm is implemented in C. Various control laws can be implemented and tried on an actual experiment.

An additional advantage is the elimination of a target computer, since the real-time OS operates on the same processor as the host computer. Most applications developed as digital control systems launch as a startup executable on a proprietary real-time target computer. The approach presented here differs; it does not target a RT controller based on a proprietary development system. It uses a Linux software environment developed for applications in control and data acquisition requiring hard real-time (deterministic) execution.

Acknowledgements

I wish to acknowledge the support of Professors Fred Nelson and Denis Fermental at Tufts for supporting this work. This work was partially funded by C. S. Draper Lab. of Cambridge, Massachusetts.

Resources for this article: www.linuxjournal.com/article/8260.

Harland Alpaugh (justanoldpaddler@verizon.net) is working on his PhD in Mechanical Engineering at Tufts University. He enjoys whitewater canoeing and often can be found on a stream in New England when the ice has disappeared.

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**

OpenLDAP Everywhere Revisited

Samba 3 offers new capabilites for a unified directory for all clients. Get mail, file sharing and more all working together with the latest software.

BY CRAIG SWANSON AND MATT LUNG

Many readers have used our December 2002 article, "OpenLDAP Everywhere", to achieve unified login company-wide. Since then, OpenLDAP and Linux have progressed. Here, we demonstrate the use of OpenLDAP as the core directory service for a mixed environment. The LDAP server provides a shared e-mail directory, login for Linux and Microsoft Windows clients, automount of home directories and file sharing for all clients. A simple mixed environment used in the examples in this article is shown in (Figure 1).

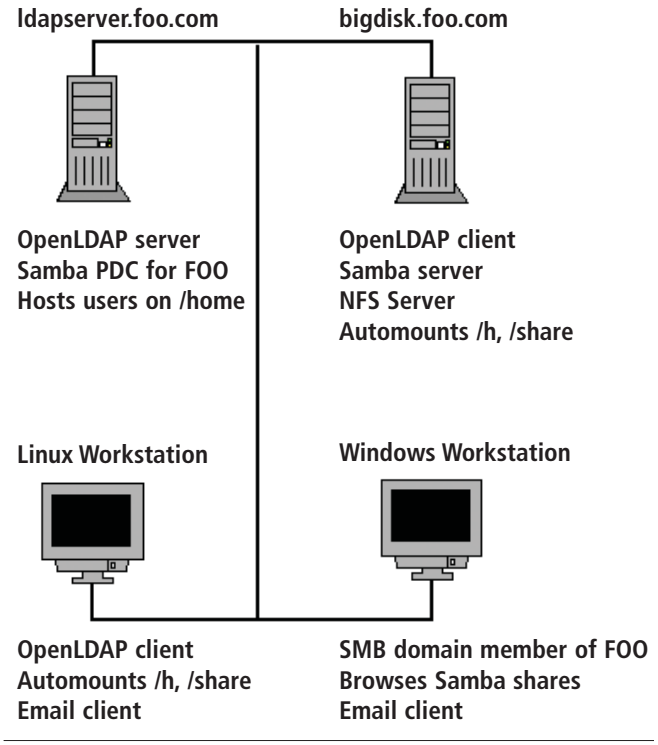


Figure 1. In the mixed environment, both Linux and Windows clients use a common LDAP infrastructure.

LDAP Server Installation and Configuration

The LDAP server we discuss was installed using RPM binary packages and openldap-2.2.13-2 on Fedora Core 3. The nss_ldap package also is required. For the most recent source from openldap.org, see the on-line Resources. Edit the server configuration file, /etc/openldap/slapd.conf, as shown in Listing 1. Lines beginning with whitespace are interpreted as a continuation of the previous line, so it's not necessary to use a back slash at the end of a long line.

The LDAP schema defines object classes and attributes that make up the directory entries. Red Hat's autofs schema fits our needs and was packaged with the RPM installation. If you find that you need to add an objectClass or an attribute to your directory, see the OpenLDAP admin guide.

We use the default database type ldbm. Our example uses the LDAP domain component. So, foo.com becomes dc=foo,dc=com.

The Manager has full write access to LDAP entries. Create the manager's password using /usr/sbin/slappasswd. Paste the encrypted password into the rootpw entry in slapd.conf.

The index lines enhance performance for attributes queried often. Access control restricts access to the userPassword entry. The user and manager may modify the entry. For all other entries, the manager has write access, and everyone else is granted read access.

Create the Directory Structure

Each entry in the directory is identified uniquely with a distinguished name (dn). The dn for foo.com is dn: dc=foo, dc=com. The organizationalUnit (ou) provides a method for grouping entries. The directory structure is shown in Listing 2.

We create the top-level entries in LDAP Interchange Format (LDIF) and save them to top.ldif, as shown in Listing 3.

Add the top-level entries to the directory with ldapadd:

```
ldapadd -x -D 'cn=manager,dc=foo,dc=com' \
-W -f top.ldif
```

Then, test your work with an ldapsearch command that

Listing 1. The slapd.conf file includes important settings for running LDAP securely.

```
# slapd.conf
# schemas to use
include      /etc/openldap/schema/core.schema
include      /etc/openldap/schema/cosine.schema
include      /etc/openldap/schema/inetorgperson.schema
include      /etc/openldap/schema/nis.schema
include      /etc/openldap/schema/samba.schema
include      /etc/openldap/schema/redhat/autofs.schema

# database definition
database    ldbm
suffix      "dc=foo,dc=com"
rootdn      "cn=Manager,dc=foo,dc=com"
# Cleartext passwords, especially for the rootdn,
# should be avoided.    Use strong authentication.
#rootpw      secret
rootpw      {SSHA}xxxxxxxxxxxxxxxxxxxx
directory   /var/lib/ldap

# Indices to maintain for this database
index       objectClass,uid,uidNumber,gidNumber,
            memberUid eq
index       cn,mail,surname,givenname eq,subinitial
index       sambaSID eq
index       sambaPrimaryGroupSID eq
index       sambaDomainName eq

# Users can authenticate and change their password
access to attrs=userPassword,sambaNTPassword,
       sambaLMPassword
       by dn="cn=Manager,dc=foo,dc=com" write
       by self write
       by anonymous auth
       by * none
# All other attributes are readable to everybody
access to *
       by self write
       by dn="cn=Manager,dc=foo,dc=com" write
       by * read
```

Listing 2. LDAP distinguished names are organized into a tree of organizational units.

```
+ dc=foo,dc=com
|- ou=People                    Persons
|  |- ou=contacts,ou=people     Email contacts
|- ou=Groups                    System groups
|- ou=auto.master               Automount master map
|- ou=auto.home                 Automount map
|- ou=auto.misc                 Automount map
|- ou=Computers                 Samba domain members
|- cn=NextFreeUnixId            Samba Next Free ID
|- SambaDomainName              Samba domain info
object class
```

Listing 3. Create the top of the LDAP tree, top.ldif, manually in the simple key: value LDIF format.

```
dn: dc=foo,dc=com
objectClass: dcObject
objectClass: organization
o: Foo Company
dc: foo

dn:ou=People,dc=foo,dc=com
objectClass: organizationalUnit
ou: People

dn:ou=Groups,dc=foo,dc=com
objectClass: organizationalUnit
ou: Groups

dn:ou=contacts,ou=people,dc=foo,dc=com
associatedDomain: foo.com
ou: contacts
ou: people
objectClass: organizationalUnit
objectClass: domainRelatedObject
```

retrieves all entries:

```
ldapsearch -x -b 'dc=foo,dc=com'
```

Share E-Mail Contacts

At this point, we have enough structure in LDAP to put it to real use. We start by sharing our e-mail contacts. To simplify the process, you may be able to export your e-mail address book in LDIF format. For example, in Mozilla Thunderbird, you can export in LDIF from the Tools menu on the Address Book window. You do need to process the resulting file so it looks like our contacts example below. We suggest using Perl for the task.

Contacts are identified uniquely by their e-mail addresses. Here is the dn for a contact:

```
dn: uid=someone@somewhere.com,ou=contacts,
➥ou=people,dc=foo,dc=com.
```

With all of the attributes, the full entry for a contact looks like:

```
dn: uid=someone@somewhere.com,ou=contacts,
➥ou=people,dc=foo,dc=com
mail: someone@somewhere.com
uid: someone@somewhere.com
givenName: Someone
sn: Youknow
cn: Someone Youknow
objectClass: person
objectClass: top
objectClass: inetOrgPerson
```

Separate each contact entry with a blank line and save it to a file called contacts.ldif. Add the contacts to the directory with ldapadd:

```
ldapadd -x -D 'cn=manager,dc=foo,dc=com' \
-W -f contacts.ldif
```

Then, test with an ldapsearch command, as shown above.

Configure E-Mail Clients

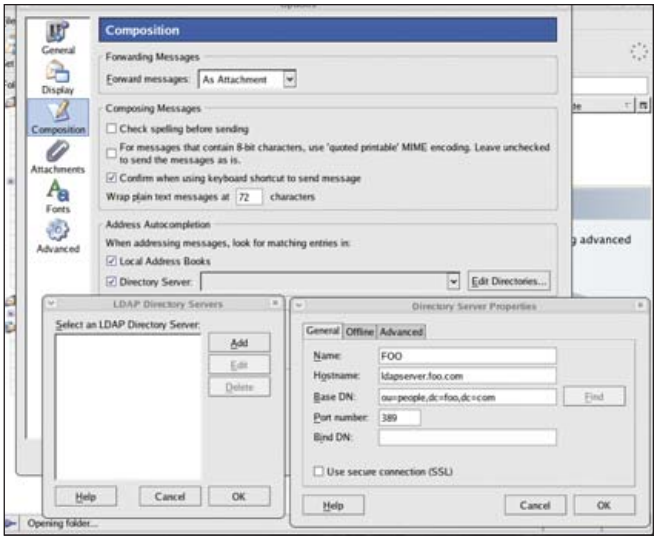


Figure 2. To use the company address book, fill in the information on your server in Thunderbird's Directory Server Properties.

Next, we configure Mozilla Thunderbird to use the new LDAP server (Figure 2). From the Tools menu in Thunderbird, select Options. In the Composition tab, select Directory Server, Edit Directories and then Add. Fill in the Directory Server Properties with:

```
Name: FOO
Server: ldapserver.foo.com
base DN: ou=people,dc=foo,dc=com
```

In the Advanced tab, increase the number of results returned to fit your directory size. For foo.com, we selected 1,000 results.

Test your settings by composing a message to one of your contacts in your LDAP directory. The address should auto-complete as you type. Another test is to search the LDAP directory from within the Thunderbird Mail Address Book. Search in the FOO address book for "Name or Email contains: *". That should return all of the contacts entries.

Unified Linux Login with LDAP

By storing user account information in LDAP, you can use the same user name and password at any Linux console. To start, you must decide which user names should be entered in LDAP. Table 1 shows our user scheme for UID/GIDs.

This user scheme allows for 9,000 LDAP unified login entries, while also allowing local users and groups that do not interfere with LDAP UIDs and GIDs. The user scheme also allows for the accounts required by the Samba Primary Domain Controller.

Table 1. User Scheme for UID/GIDs

Type of account	UID
System accounts	UID < 500
Samba special accounts	499 < UID < 1,000
Unified login accounts	999 < UID < 10,000
Local users and groups, not in LDAP	> 10,000

Create LDAP User Login Entries

A user login entry is identified by the login name as uid. Login users are members of ou=people, resulting in this dn:

```
dn: uid=gomerp,ou=people,dc=foo,dc=com
```

The full entry contains attributes that are needed to control account access, as shown in Listing 4. The full entry also includes attributes needed by the Samba configuration that is discussed below.

OpenLDAP ships with migration utilities that can extract the user account information; see /usr/share/openldap/migration. To convert an existing /etc/passwd file to LDIF, start by checking migrate_common.ph. Edit the file to include your domain name, default base and enable extended schema:

```
# Default DNS domain
$DEFAULT_MAIL_DOMAIN = "foo.com";

# Default base
$DEFAULT_BASE = "dc=foo,dc=com";

# turn this on to support more general object classes
# such as person.
$EXTENDED_SCHEMA = 1;
```

Extract the user account information from /etc/passwd:

```
/usr/share/openldap/migration/migrate_passwd.pl \
/etc/passwd > people.ldif
```

Review the resulting LDIF file. You should remove entries for system accounts such as root and for local users' private groups that do not need to appear in LDAP.

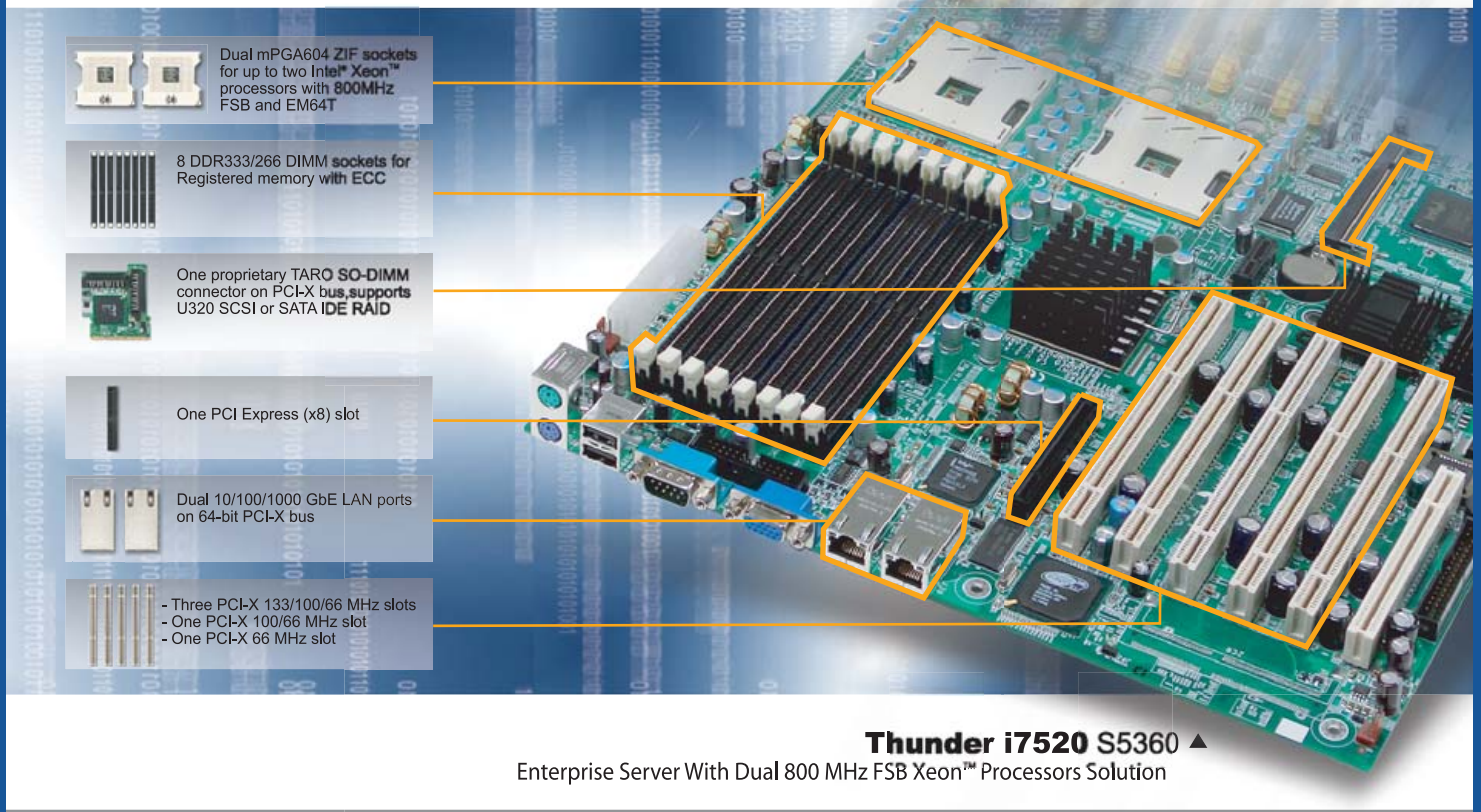
Add the user entries to LDAP and test with an ldapsearch command, as discussed above:

```
ldapadd -x -D 'cn=manager,dc=foo,dc=com' -W \
-f people.ldif
```

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

Listing 4. A user login entry contains the password information needed to log in, along with Samba configuration.

```
dn:uid=gomerp,ou=People,dc=foo,dc=com
uid: gomerp
cn: Gomer Pyle
sn: Pyle
givenname: Gomer
mail: gomer.pyle@foo.com
objectClass: top
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
objectClass: sambaSAMAccount
uidNumber: 5000
homeDirectory: /h/gomerp
loginShell: /bin/bash
description: Gomer Pyle
displayName: Gomer Pyle
gecos: Gomer Pyle
gidNumber: 513
userPassword: {SSHA}xxxxxxxxxxxxxxxxxxxxxxxx
sambaLogonTime: 0
sambaLogoffTime: 2147483647
sambaKickoffTime: 2147483647
sambaPwdCanChange: 0
sambaSID: S-1-5-21-1400792368-3813960858-1703501993-11000
sambaPrimaryGroupSID: S-1-5-21-1400792368-3813960858-1703501993-513
sambaLogonScript: gomerp.cmd
sambaHomeDrive: H:
sambaHomePath: \\LDAPSERVER\gomerp
sambaLMPassword: XXXXXXXXX
sambaAcctFlags: [U]
sambaNTPassword: XXXXXXXXX
sambaPwdLastSet: 1097240543
sambaPwdMustChange: 1105016543
```

Because the login users belong to ou=people, you now may look up their e-mail addresses within your e-mail client.

Create Group Entries

You need to make a group entry for each group to be shared between multiple Linux computers. Each user also needs a group entry for the user private group. A group entry is identified by cn, and each group belongs to ou=Groups. For example:

```
dn: cn=gomerp,ou=Groups,dc=foo,dc=com
```

A user private group would look like this:

```
dn: cn=gomerp,ou=Groups,dc=foo,dc=com
objectclass: posixGroup
objectclass: top
cn: gomerp
userPassword: {crypt}x
gidNumber: 5223
```

A shared group would look like:

```
dn: cn=web_dev,ou=Groups,dc=foo,dc=com
objectclass: posixGroup
objectclass: top
cn: web_dev
gidNumber: 5019
memberUid: gomerp
memberUid: goober
memberUid: barneyf
```

Extract the group information from /etc/group:

```
/usr/share/openldap/migration/migrate_passwd.pl \
/etc/group > group.ldif
```

Review the resulting LDIF file. You should remove entries for system groups and for local system users that do not need to appear in LDAP.

Add the group entries to LDAP and test with an ldapsearch command:

```
ldapadd -x -D 'cn=manager,dc=foo,dc=com' -W \
-f group.ldif
```

Configure Automount to Share Home Directories and NFS Shares

With unified login, users have a single home directory that is shared by way of the Network File System (NFS). We host our home directories from ldapserver.foo.com and share /home, but the file server and OpenLDAP do not need to run on the same machine. Details on NFS are outside the scope of this article, but here is the line from /etc/exports that works to export home directories:

```
/home *.foo.com(rw)
```

Linux LDAP clients mount the user's home directory at login, using automount and NFS. The LDAP use of automount is a replacement for NIS (network information service) automount maps. Replace the automount maps for auto.master, auto.home and auto.misc. To do so, we create a new organizational unit for auto.master:

```
dn: ou=auto.master,dc=foo,dc=com
objectClass: top
objectClass: automountMap
ou: auto.master
```

An auto.master entry is identified by cn. The automountInformation attribute instructs automount to look for the map in LDAP:

```
dn: cn=/h,ou=auto.master,dc=foo,dc=com
objectClass: automount
automountInformation: ldap:ou=auto.home,dc=foo,dc=com
cn: /h
```

While we're at it, let's create an auto.master entry for other NFS shared directories:

```
dn: cn=/share,ou=auto.master,dc=foo,dc=com
objectClass: automount
automountInformation: ldap:ou=auto.misc,dc=foo,dc=com
cn: /share
```

Create the automount entries in LDIF format, save as auto.master.ldif and add the entries to LDAP:

```
ldapadd -x -D 'cn=manager,dc=foo,dc=com' -W -f auto.master.ldif
```

Next, we create a new organizational unit for auto.home:

```
dn:ou=auto.home,dc=foo,dc=com
objectClass: top
objectClass: automountMap
ou: auto.home
```

A home directory entry is identified by cn:

```
dn: cn=gomerp,ou=auto.home,dc=foo,dc=com
objectClass: automount
automountInformation: ldapserver.foo.com:/home/gomerp
cn: gomerp
```

Create auto.home entries for each user in ldif format, save as auto.home.ldif and add the entries to LDAP:

```
ldapadd -x -D 'cn=manager,dc=foo,dc=com' -W \
-f auto.home.ldif
```

When automounted from a Linux LDAP client, your home directory, ldapserver.foo.com:/home/gomerp, is mounted on /h/gomerp. Other NFS shares may be entered in LDAP and automounted as needed. The auto.misc organizational unit holds these automount maps, which have the form ou=auto.misc.

We've already created an auto.master entry for /share, as shown above. Now, we create the ou=auto.misc entry:

```
dn:ou=auto.misc,dc=foo,dc=com
ou: auto.misc
objectClass: top
objectClass: automountMap
```

Create entries for the NFS shares under ou=auto.misc:

```
dn:cn=redhat,ou=auto.misc,dc=foo,dc=com
objectClass: automount
automountInformation: bigdisk.foo.com:/pub/redhat
cn: redhat

dn:cn=engineering,ou=auto.misc,dc=foo,dc=com
objectClass: automount
automountInformation: bigdisk.foo.com:/data/engineering
cn: engineering
```

Save the entries as auto.misc.ldif and add the entries to LDAP:

```
ldapadd -x -D 'cn=manager,dc=foo,dc=com' -W -f auto.misc.ldif
```

When automounted from a Linux LDAP client, your shared directory bigdisk.foo.com:/data/engineering is mounted on /share/engineering.

Configure the Linux LDAP Client

To begin configuring the Linux LDAP client, you need to install the name switch service package, nss_ldap. The Red Hat tool /usr/bin/authconfig is handy for configuring the client. Select Use LDAP and fill in the fields so that they read Server: ldapserver.foo.com and Base DN: dc=foo,dc=com. Authconfig writes to these files: /etc/ldap.conf, /etc/openldap/ldap.conf and /etc/nsswitch.conf.

Verify that /etc/nsswitch.conf has the following entries:

```
passwd:     files ldap
shadow:     files ldap
group:      files ldap
automount:  files ldap
```

Verify that /etc/ldap.conf has these entries:

```
host ldapserver.foo.com
base dc=foo,dc=com
```

Verify that /etc/openldap/ldap.conf has these entries:

```
HOST ldapserver.foo.com
BASE dc=foo,dc=com
```

Final Linux Server Configuration

The user's password and group entries must be removed from the password and group files on the NFS server where home directories live. Create backups and then edit /etc/passwd, /etc/shadow, /etc/group and /etc/gshadow to remove the LDAP real-people entries. In our case, /etc/passwd should have no accounts left with a UID from 1,000 to 9,999.

To test, log in to a Linux LDAP client using an LDAP user name. You should see the appropriate login shell and home directory for the user. To test auto.misc shares, you must access the share by name, for example:

```
cd /share/redhat
```

Automount only mounts NFS shares as they are used, so the directory /share/redhat is not visible until it has been accessed.

Achieve Unified Login with Samba and LDAP

The main purpose of using Samba and LDAP together is to achieve unified login for Microsoft Windows clients. What this means to your organization is a user will be able to log on to your network from any workstation and have access to all shared folders, files and printers.

The first step to unified login starts by configuring Samba as a primary domain controller (PDC). The full configuration details on how to set up Samba as your PDC are outside the

Listing 5. Excerpts from a Samba smb.conf file configured to work with the OpenLDAP directory.

```
[global]

...
  obey pam restrictions = No
  ldap passwd sync = Yes
  ldap passwd sync = Yes
...
  passdb backend = ldapsam:ldap://ldapserver.foo.com/
  ldap admin dn = cn=Manager,dc=foo,dc=com
  ldap suffix = dc=foo,dc=com
  ldap group suffix = ou=Groups
  ldap user suffix = ou=People
  ldap machine suffix = ou=Computers
  ldap idmap suffix = ou=People
  ldap ssl = no
  add user script = \
    /usr/local/sbin/smbldap-useradd -m "%u"
  ldap delete dn = Yes
  delete user script = \
    /usr/local/sbin/smbldap-userdel "%u"
  add machine script = \
    /usr/local/sbin/smbldap-useradd -w "%u"
  add group script = \
    /usr/local/sbin/smbldap-groupadd -p "%g"
  delete group script = \
    /usr/local/sbin/smbldap-groupdel "%g"
  add user to group script = \
    /usr/local/sbin/smbldap-groupmod -m "%u" "% g"
  delete user from group script = \
    /usr/local/sbin/smbldap-groupmod -x "%u" "%g"
  set primary group script = \
    /usr/local/sbin/smbldap-usermod -g "%g" "%u "
```

scope of this article. Please visit the Idealx Web site for a great HOWTO (see Resources). The folks at Idealx have made great contributions to the Samba Project, and you should become familiar with their tools if you plan on using Samba.

Assuming you already have experience with setting up Samba domain controllers, this Samba configuration file should get you up and running with our directory example in this article (Listing 5). The full file is available from the *Linux Journal* FTP site (see Resources).

The remaining piece of the puzzle involves setting up LDAP to take advantage of Samba's advancements made in the past couple of years. This should be similar to the LDAP setup above, but with updated features added in for Samba. With the new Samba 3 version, we now are able to store all Samba account information inside the LDAP directory. This is beneficial because now all the information is stored in a centralized location.

Samba LDAP Setup

One difference in the LDAP/Samba combination setup is the additional accounts and LDAP entries that need to be populated for the two to work together. Several well-known Windows domain user accounts and domain group accounts are required for your unified login server to function. Special LDAP OU entries also are required for the server to store domain account information. Fortunately, a script called smbldap-populate is available that does all of this for you. This script is part of the Idealx smbldap-tools package that can aid you in setting up both your PDC and your Samba Enabled LDAP directory. Listing 6 is sample output of what you should see when you run the smbldap-populate script.

Listing 6. The smbldap-populate tool automatically adds the accounts required to make your OpenLDAP server work with Samba.

```
[root]# smbldap-populate
Using builtin directory structure
adding new entry: dc=foo,dc=com
adding new entry: ou=Users,dc=foo,dc=com
adding new entry: ou=Groups,dc=foo,dc=com
adding new entry: ou=Computers,dc=foo,dc=com
adding new entry: ou=Idmap,dc=foo,dc=org
adding new entry: cn=NextFreeUnixId,dc=foo,dc=org
adding new entry: uid=Administrator,ou=Users,dc=foo,dc=com
adding new entry: uid=nobody,ou=Users,dc=foo,dc=com
adding new entry: cn=Domain Admins,ou=Groups,dc=foo,dc=com
adding new entry: cn=Domain Users,ou=Groups,dc=foo,dc=com
adding new entry: cn=Domain Guests,ou=Groups,dc=foo,dc=com
adding new entry: cn=Print Operators,ou=Groups,dc=foo,dc=com
adding new entry: cn=Backup Operators,ou=Groups,dc=foo,dc=com
adding new entry: cn=Replicator,ou=Groups,dc=foo,dc=com
adding new entry: cn=Domain Computers,ou=Groups,dc=foo,dc=com
```

If you examine the sample output of this populate script, you should notice that it has added several new users, groups and OUs to the directory. For example, the script adds the well-known Domain Admins and Domain Users groups to the directory. The NT-based versions of Microsoft Windows all are preconfigured with specific user and group entries. Each one of those has a relative identifier (RID) associated with it. What this means to LDAP is the corresponding LDAP user or group entry must be assigned to the respective RID of the Windows user or group. Using the smbldap-populate script takes care of

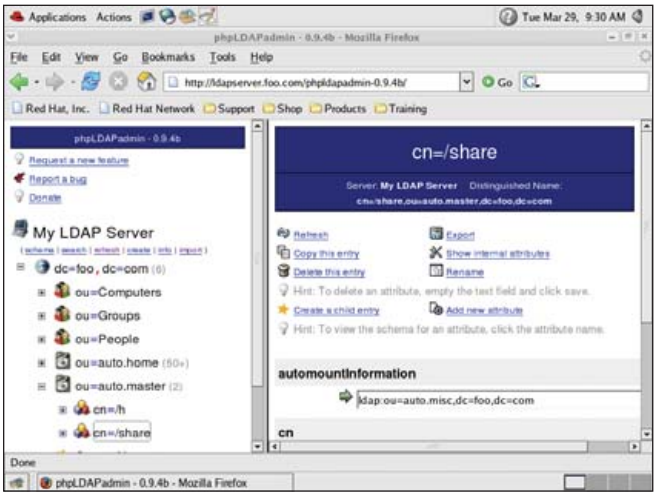


Figure 3. Get a Web view of your directory with phpLDAPadmin.

making the relation for you. The well-known user and groups RIDs that are required are:

```
Name           RID
-----------------
Domain Admins 512
Domain Users  513
Domain Guests 514
```

Aside from the new user and group entries, several new OU entries can provide further domain functionality. The first of these is ou=Computers, which is used to store all machine accounts for member servers and workstations on the domain. Second, the ou=Idmap is used if Samba is being used as a domain member server of a Windows server controlled domain. The last new entry is ou=NextFreeUnixId. This entry is used to define the next UID and GID available for creating new users and groups.

Managing Your Directory

After your LDAP directory is populated and Samba is set up correctly, you are ready to start adding users and groups to populate your directory. The Idealx command-line utilities can take care of this job nicely for you. Some PHP-based directory managers are available that can be useful here as well. Consider using phpLDAPadmin and/or the LDAP Account Manager (LAM) to take on this task. Both are helpful, providing a graphical view of your directory. Each also provides the ability to view and edit LDAP entries in a user-friendly graphical environment (Figure 3). The LDAP browser, which is Java-based, is another option for viewing and editing your directory.

Since the December 2002 article, we have seen much improvement in Samba with the 3.x releases. Moving to the new version should mean greater control over accounts and improved group mapping functionality, thus giving you greater control over your domain.

Maintenance

We strongly suggest that you use simple authentication and security layer (SASL) and transport layer security (TLS) to secure your new LDAP directory. See Resources for details.

Congratulations! Your LDAP server is up and running with shared e-mail contacts, unified login and shared file storage that is accessible from any client.

Resources for this article: www.linuxjournal.com/article/8267.

Craig Swanson (craig.swanson@slssolutions.net) designs networks and offers Linux consulting at SLS Solutions. He also develops Linux software at Midwest Tool & Die. Craig has used Linux since 1993.

Matt Lung (matt.lung@slssolutions.net) provides network and computer systems consulting at SLS Solutions. He also works at Midwest Tool & Die as a Network Engineer.

WIX: a Distributed Internet Exchange

A city-wide Ethernet opens up a world of possibilities, from letting companies share data across sites cheaply to saving ISPs from crushing demand when a local site hosts a Webcast all the users want.

BY RICHARD HULSE

Wellington, the capital city of New Zealand, has one of the oldest and possibly largest distributed Internet exchanges in the world. It is built on top of the Citylink public LAN infrastructure. In this article, we look at how it all started and the part Linux plays in its day-to-day operation.

History

Back in the 1980s, Richard Naylor, then IT manager for the Wellington City Council, was stuck with a common but difficult problem—an over-utilized VAX cluster at one data centre and an under-utilized one at another centre across town. He came up with an idea that was unique for the time: run a fibre optic cable between the two council buildings and share processing resources that way. The idea was so new that a jointer had to be flown down from Auckland to do the splices on the now-ancient slotted core cable.

Naylor set up a 10Mbps Ethernet connection using DECBridges, and the network itself was running DECnet on a 10-base5 and a little 10-base2. Terminals made up most of the load at the time, as PCs didn't network. The overall idea worked, and the system was upgraded to 10-baseT in 1989, with IP being added.

In the early 1990s, Naylor's idea caught the attention of then Mayor Fran Wilde, who was intrigued by what Naylor and colleague Charles Bagnall had been up to in what was called "their spare time". Mayor Wilde had attended a local secondary-school production and noticed that it was being streamed live on the Internet by an off-duty Naylor.

After talking with Naylor and Bagnall, Wilde came to understand what was possible. Their design and the use of fibre could be used to provide a broadband infrastructure for the city, and that plan became a key part of a 25-year strategy for Wellington City.

Soon after, 17 investors, including the council, came up with $5,000 each, and three drums of fibre optic cable were run from one end of Wellington City to the other. The cable was run along overhead trolley bus electric support cables during November and December 1996. The primary aim was to provide an infrastructure to enable greater growth within the local business community.

In the first few years, Citylink expanded at a rate of 100% each year—doubling the number of connected buildings. At one point, the team connected 50 new buildings in ten weeks.

In 1997, Naylor left the council and helped set up Citylink as a separate company. The first customers were government departments and financial businesses, as central government is located in Wellington. Later customers included publishers and IT companies. ISPs were there from the start too, with many using the fibre infrastructure as a way of providing genuine broadband to city customers at a low cost. I should note here that Citylink does not consider domestic 1Mb and 2Mb connections to be broadband; it prefers to start at 10Mb. Citylink can provide 10Mb, 100Mb or 1,000Mb connections anywhere on the fibre infrastructure.

Nearly ten years after the initial cable was positioned, around 50 kilometres (30 miles) of cable now exist within the central business district of Wellington. More than 300 buildings are connected.

The Infrastructure

The Citylink fibre network originally was interconnected with a bunch of oddball switches and hubs from various vendors. Over time, these have been phased out and replaced with Cisco switches, mainly 35xx and 29xx, on a gigabit core.

Citylink now offers two major services, dark fibre and Ethernet. Dark fibre services allow point-to-point connectivity between buildings. Each customer has sole use of his or her own strand of fibre and can connect whatever gear is required on either end. Dark fibre runs up to 1Gb/s at present. The Ethernet services are the most widely used and allow customers to connect to a city-wide "shared Ethernet".

INL, a newspaper publisher, was the first Citylink customer. The company used dark fibre to link its corporate office to its Wellington office, and it used Ethernet to link back to the Wellington City Council offices. From there, a microwave link to Victoria University of Wellington provided access to the public Internet. At that point, Citylink was involved in providing only basic layer 2, Ethernet infrastructure.

Routing 101

The Internet exchange runs on top of the fibre Ethernet infrastructure. Before we look at this in detail, let's briefly run through a few routing basics. A router on a network receives packets of data, each with its own destination address. The router checks its internal list of addresses—the

routing table—to see if there is a route or path to that destination. If there is, the packet is sent to the specified address. This might be the final destination, or it could be a gateway—another router that repeats the process and sends the packet on once again.

A traditional Internet exchange has participants connecting to a single location, and each participant has a router. One side of the router is connected to all of the other routers by way of a common Ethernet backbone. The other side is connected to the participant's own network infrastructure (Figure 1).

Figure 1. One set of route servers—IPv4 (Soekris 4801) on the left and IPv6 (Soekris 4501) on the right.

Each participant has a list of IP addresses that represents the networks it can access. Each router has its own IP address on the common backbone. These addresses often are private to the outside world and act as gateways to each participant's network.

As a point of interest, the first Citylink routers in general use were based on standard PC components with LS/120 floppy or disk-on-chip boot mechanisms running the Linux Router Project (LRP). These were deployed only for customers using wireless access points.

Enter BGP

Border Gateway Protocol (BGP) is fundamental to the operation of the Internet, because it automates the sharing of routes by a process called advertising. Adjacent routers establish a BGP session and advertise their networks to the other routers. It is a bit like one router saying to another, "Here is my IP address. If you have traffic with any of the following destination addresses, send it to me." BGP doesn't stop there, though. Routers also share information they have gathered from other routers. They say, "I also know how to get traffic to some other networks. Send that to me as well."

Typical Internet exchanges are different from the rest of the Internet and use Internal Gateway Protocol (IGP) instead. This protocol does not pass on information to other routers—it advertises routes only within the networks to which it is connected. The router says, "You can send me traffic for these addresses, but you can't pass that address information on to any other routers."

The Citylink Exchange

In 1998, Citylink started work on deploying a BGP/4-based Internet exchange on top of the public Ethernet. One of the key design criteria was a low cost of entry. So the Citylink team took a step back and looked at the exchange problem from a logical point of view: what were the core requirements of an exchange, and

Figure 2. The Citylink fibre (yellow cables) is converted to copper (blue) for distribution to clients.

Figure 3. A typical fibre switching point, as found in many PABX rooms around Wellington.

how could they use the existing fibre infrastructure to create an environment that would encourage as many people as possible to join? Exchanging traffic via an exchange is called peering.

A key component of Citylink's approach involved placing routers at a customer's point of connection to the fibre infrastructure rather than in a central location. Not a lot of options were available for BGP/4-capable routers, though—you pretty much bought a Cisco or a Cisco (Figure 2).

Because the exchange was distributed, Citylink needed a low-cost mechanism to allow people to peer, and so they started using Zebra on Linux as an alternative routing platform to Cisco.

Because of the limited space in clients' cabinets and the need to be able to buy the same hardware over a reasonable period of time, Citylink now has standardized on two systems. Boards from Soekris Engineering (4501 and 4801) are used for routers up to 30Mb/s, while Nexcom P4 GigE and VIA 100Mb boards are used for 100Mb/s connections and upwards to 300–500Mb/s on a good day. The Linux Embedded Appliance Firewall (LEAF) forms the basis of all software used on Citylink routing hardware.

Customers can use the Citylink routers or whatever suits their needs and budgets. A wide variety are in use—from Dlink and Xyzel at the low end to Juniper and Cisco boxes at the high end. A lot of people also use their own Linux and *BSD boxes. Using PC-based routers does limit the options for network interfaces, but this is not an issue for Citylink, which uses only Ethernet.

In addition to shorter and faster network paths, no charges are levied for traffic sent over Citylink by way of the exchange.

In addition to devolving hardware to client's premises, the routing tables for the whole exchange are managed by way of route servers, the operation of which I explain later in this article.

The new exchange was dubbed WIX, the Wellington Internet Exchange. There is a fixed monthly charge for connection, and traffic over the fibre network is free. Because the network often runs right past a potential user's door, it is easy for anyone to connect and peer. And this is exactly what has happened—even end users who could never peer under a traditional model now can. For all exchange users, access to the global Internet still necessitates the purchase of "transit"—access to the global Internet routing table—from at least one ISP.

The open peering approach has made a huge impact on traffic flows and latency. When the Internet first started in New Zealand, the country's single exchange point, at Waikato University, was also the international gateway for the whole country. Two businesses in Wellington wanting to exchange data would send it to their respective ISPs, who sent it upstream to Waikato. The path time for this typically ran 50–200ms. When ISPs began to exchange data directly in Wellington, this rate dropped to 20ms. End-user peering reduced this by another factor of ten, to 2ms.

Figure 4. Cables coming off overhead trolley bus supports to a junction point on a building veranda. The ability to exchange huge files at no cost helps local businesses save money and offer new services.

In addition to shorter and faster network paths, no charges are levied for traffic sent over Citylink by way of the exchange. ISPs never see the traffic, because the exchange directs it through the shortest path to the requesting party.

A number of printing companies peer at WIX and exchange huge publishing files at no cost. One local newspaper runs an FTP server on WIX to which pre-press companies can upload files. Some graphics houses also run media stores for their clients, and these can browsed at no cost, as though they were part of the client's own LAN. These are only a few of the ways that the fibre infrastructure and WIX have helped local businesses to lower costs, expand and innovate.

The distributed exchange environment has been likened to a market square—anyone can trade his or her wares with no cost for participation. Not everyone chooses to peer openly though, and the exchange supports all types of participation. A number of ISPs exchange traffic only with established customers and ignore any other traffic. Some simply use WIX as a neutral point to exchange data with one other party.

About 130 entities are peering at WIX, with close to 1,000 using the Citylink fibre for private purposes or public Internet connectivity. Encouraging peering comes with a downside though—a lot of participants means a lot of routing tables need to be managed.

DON'T BE SQUARE! GET CUBED!

L Series Laptop - LS1240
Light &Thin Performance
Starting at $1,448.14

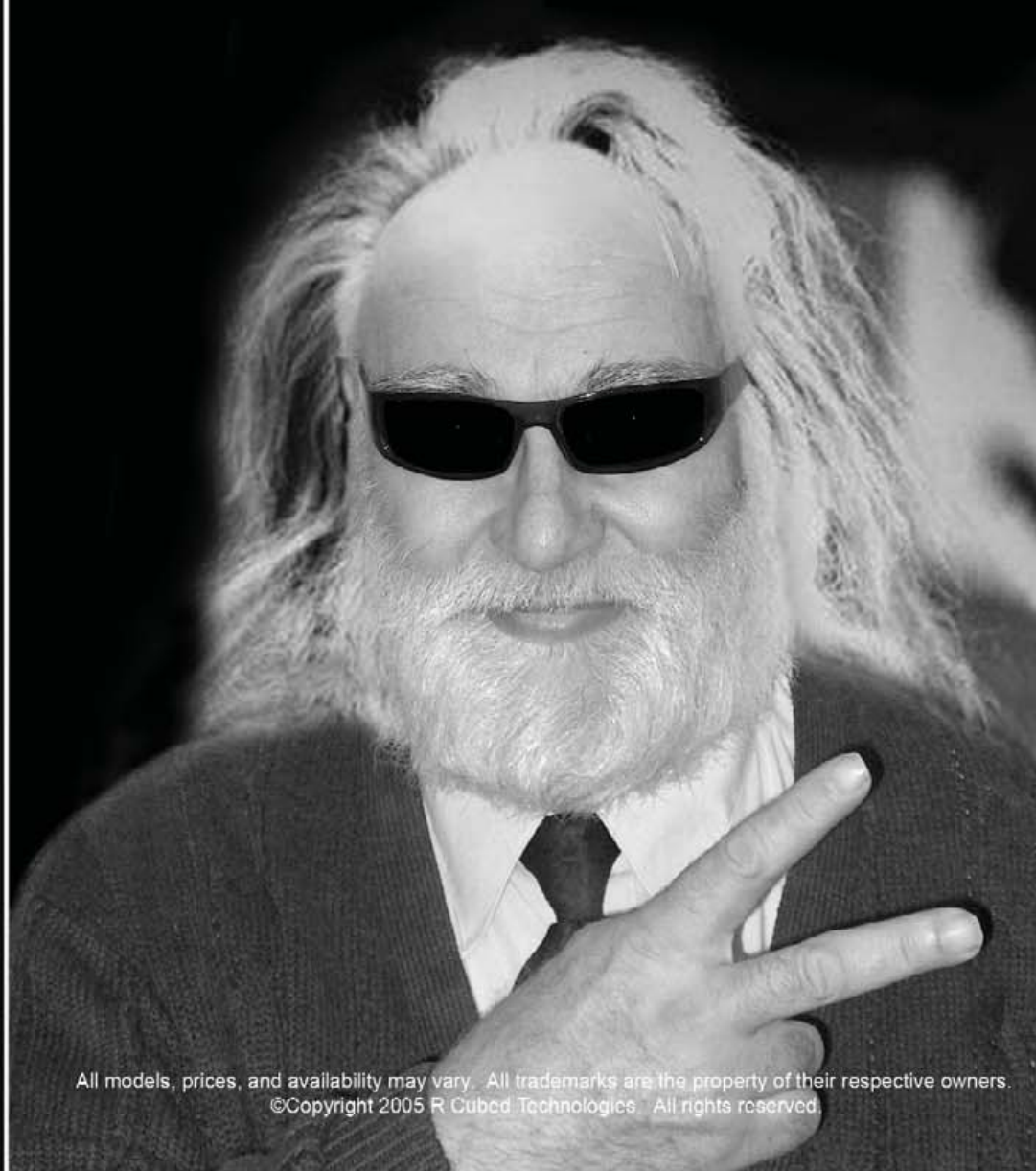
G Series Laptop - GW1550
Essential Technology on a Budget
Starting at $1,302.54

X Series Laptop - XW1550
Extreme Technology & Performance
Starting at $1,608.04

R Cubed Technologies has provided pre-installed Linux laptops without OS tax since 2003. We customize the Fedora Linux distribution for each laptops configuration providing support for: PCMCIA, USB, FireWire, X, CD/DVD/CDRW/DVDRW, Sound, Power Management, Ethernet, Modem, Wireless, and more. Our laptops are equipped Intel Centrino Mobile Technology. We also offer Windows dual boot options. All of our laptops come with a one year parts and labor warranty. Visit us online at www.shoprcubed.com or call 309.34.CUBED for details

R3 Technologies™
Working Hard to Bring Technologies to Life.

309.34.CUBED
www.shoprcubed.com



The Many Uses of Linux

In order to make route changes simpler for the hundreds of WIX users, Citylink has deployed two route servers. Rather than having to peer with every router on WIX, the preference is to peer with only the route servers, dramatically simplifying route table maintenance for everyone. Each client router, then, has to maintain only two BGP sessions—one to each of the two route servers rather than to hundreds of routers.

The route servers (Figure 1) don't carry any traffic at all. They simply reflect routes from one peer to all of the rest. I was surprised to find that the servers are based on the same small footprint diskless 266MHz Pentium board made by Soekris Engineering that is used for the routers. LEAF is installed on CompactFlash for reliability and fast booting. The Quagga Routing Software Suite is used to provide BGP services, and the kernel handles packet routing.

Two routers are used for redundancy, for IPv4, and a second set provides services for IPv6. Citylink maintains its own route registry and uses the Routing Policy Specification Language (RPSL) to manage the IPv4 routers. A set of shell and Perl scripts has been developed that use RTCONFIG to construct configuration files for the Quagga software. All this gives Citylink tight control over what peers can announce to the servers and ensures that a replacement can be deployed quickly if required. The whole process is managed with Revision Control System (RCS) to allow backups to be made and to ensure consistency.

Participants on WIX may advertise only those address spaces that are within their own individual network boundaries. The route servers re-advertise what they learn and filter out addresses that should never be routed, which are called Bogons. Examples of these are the loopback address, 127.0.0.1/8; other addresses allocated for private networks, such as 172.16.0.0/12; and non-assigned addresses.

At present, the IPv6 routers are maintained manually. When RPSL for IPv6 is standardised, however, and the amount of work required increases sufficiently, scripts will be used.

Anycast but Not Anywhere

One interesting technique deployed on the WIX and its sister exchange, APE, in Auckland, is anycast routing. A good example of anycast routing is the recent addition of a mirror of a root nameserver at WIX. Because of the way BGP and anycast works, a query to the root server goes to the nearest mirror automatically. If an ISP peers at WIX, it can get a 2ms ping time. International paths are over 200ms, so this is a huge improvement.

Local media companies also use anycasting to provide content on the exchange at a low marginal cost to ISPs. Rather than having to bring content requested by their customers across expensive international circuits, the ISPs can get it locally.

Anycast also can be used to limit the distribution of the traffic to only local networks. One example of this is *The Return of the King* premiere parade, which was Webcast using anycast routing from downtown Wellington. Over five hours, about 12 terabytes of data was requested by New Zealand customers, and this content was provided at no cost to ISPs. A mirror of the stream also was provided from a server in the USA for international viewers. Another example is the provision of software mirrors such as one for Debian, the distribution used by Citylink.

Figure 5. A CafeNet access point. The antenna is the small white rod on the right side of the box.

One of Citylink's biggest innovations is the provision of wireless Internet connections in cafés and some business premises in Wellington. The first access point was installed in June 2002, and the service was launched officially in November of that year. Currently, more than 200 access points are in operation (Figure 5).

One good example of how this Wi-Fi technology can benefit the community is the recent installation of a wireless access point at the Mary Potter Hospice for terminally ill patients. Two laptops on mobile trolleys are used to allow patients to stay in touch with their families or simply to read material online in their own time.

Summary

Citylink also operates the Auckland Peering Exchange (APE), which has about 40 peering participants. A recent addition was the Palmerston North Internet Exchange (PNIX), which, although it has only one participant right now, serves as a place for content providers to mirror servers. Other exchanges are planned for other parts of New Zealand in the near future.

Citylink has found that Linux readily is adaptable to whatever the company needs it to do. Having "intelligent" routers running Linux has meant that deeper firewalls can be deployed; it also has given staff access to better debugging tools. For example, TCP Dump can be used to examine traffic through a router in real time if required.

The use of Linux and other open-source software has been a key enabler in creating an affordable public Ethernet and a low cost-of-entry distributed Internet exchange. It will continue to allow the number of exchanges to grow and to fuel innovation and collaboration in other centres around New Zealand.

Resources for this article: www.linuxjournal.com/article/8265.

Richard Hulse is a broadcaster based in Wellington, New Zealand. He currently is working on a number of IT projects that involve using Linux to bridge between disparate systems.

KEYNOTE ADDRESS
"Computer Security in the Real World"
by Butler W. Lampson, *Microsoft and MIT*

A six-day tutorial and refereed technical program for security professionals, system and network administrators, and researchers

Cutting-Edge Security Developments from Industry Experts

14th USENIX SECURITY SYMPOSIUM

Baltimore, MD USA • July 31–August 5, 2005

NEW IN 2005! 3 DAYS OF IN-DEPTH TUTORIALS, INCLUDING:

- Network Security Monitoring with Open Source Tools, by Richard Bejtlich
- Organizing a Cybersecurity Exercise, by Ron Dodge and Dan Ragsdale

TECHNICAL PROGRAM

22 refereed papers and 2 panels on the best new research, including RFID device analysis, managing secure networks, defenses, and building secure systems

INVITED TALKS BY EXPERTS, INCLUDING:

- Avi Rubin, Johns Hopkins University — "Electronic Voting in the U.S.: An Update"
- Ben Shneiderman, University of Maryland — "Human-Computer Interaction Opportunities for Improving Security"
- Pradeep Khosla, Carnegie Mellon University — "Cybersecurity: Opportunity and Challenges"

USENIX
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

Register online by July 16, 2005, and Save!
www.usenix.org/sec05/lj

Easy Database Development Using Rekall

Rekall allows you to develop database applications quickly that run under Linux or Microsoft Windows. Find out how to use this free development tool for your next project. BY JOSHUA BENTHAM

The issue of cross-platform programming will become bigger over the next few years, as companies—particularly small- and medium-sized businesses—see the advantages of migrating to Linux and take the smart road by migrating a few machines at a time. Being able to develop vertical-market applications quickly that run on the multiple platforms a client has will become a competitive advantage for those consultants who are able and smart enough to do so. With Rekall and PostgreSQL, that ability is available here and now.

Last year, I was approached by my homeowners' association about replacing an old Microsoft Access application. I had completed assignments like this in the past using various toolkits, such as a combination of Glade, C++ and Sybase ASE, and then later, Apache, PHP and PostgreSQL. This time I was faced with a number of restrictions: the software had to work under Microsoft Windows as well as Linux; the software had to have a local thick client; and the software had to produce reports.

The database back end was a no-brainer; I used PostgreSQL. Having access to PostgreSQL version 8 under both Linux and Windows is not only a testament to the power of great free software, but it also provides an extremely fertile ground upon which to grow cross-platform applications that are robust and scalable. With a suitable front end, it does not matter if your server is Linux or Windows or how many of your desktops are Linux or Windows. This is great for companies pondering a switch to desktop Linux for some or all of their employees.

For the front end, I wanted a development environment that would allow me to design forms, reports and the databases to which they connect quickly. Cross-platform operation was a must, because the association was interested in migrating to Linux for some of their employees, while keeping Windows on the desktops of those who used proprietary Windows-only applications.

During my search, I ran across several tools that purported to be like Microsoft Access, only better. All of these tools allow access to multiple databases as well as ODBC sources, and they are scriptable using some form of BASIC, JavaScript or Python. These tools included Kexi, OpenOffice.org Base, Kylix, Knoda, Rekall and Glom.

Of those products, the only two that claimed to be production-ready were Kylix and Rekall. So, I investigated them further. After finding out that Kylix does not have built-in support for producing reports, I concentrated my research on Rekall.

Rekall Revealed

Rekall is developed by the British company Series One Consulting. Mike Richardson, the primary consultant at Series One, began writing Rekall in 2001 due to his frustration at the lack of database development tools under Linux. Mike was joined by John Dean, who took charge of Windows and Macintosh development and wrote drivers for Oracle and DB2.

Rekall was distributed commercially by TheKompany as one of its cross-platform development applications. In late 2003, the distribution contract ended, and Series One decided to distribute Rekall themselves. If you decide to check out Rekall, be sure to get it from either the TotalRekall or RekallRevealed Web sites, as these are the versions actively supported.

Rekall is available under two licenses, the GPL and a proprietary license. This is due in part to the fact that it is built using Qt, and at the time it was first developed, Trolltech did not offer the Windows version of Qt under the GPL. Therefore, you can download the Linux version of Rekall from the RekallRevealed Web site as source code and compile it yourself. Or, you can pay £25 (roughly $45 US) for a download of the Linux, Windows and Macintosh binary installation packages. Additionally, Series One offers database drivers for ODBC, Oracle and DB2, as well as runtime packages, at additional cost.

The next version of Rekall, 2.4, is scheduled to be delivered in several different versions. The GPL version always will be available in source form. Additionally, Series One plans to offer a Professional version that includes two additional features:

- Encryption allows you to distribute your applications without worrying about someone copying your source code. The encryption provided by the Professional version of Rekall will allow you to distribute your application and secure it on a per-client basis.

- Web application creation allows you to take any Rekall application and make a LAMP-based Web application out of

it. This can be seen on the RekallRevealed Web site.

Drawbacks of Rekall

In my application, I found only a couple of drawbacks to using Rekall. First, there is no straightforward way to create menu bars. Second, the applications produced by Rekall are not encrypted.

In a Rekall application, there is a standard menu bar and toolbar with commands that allow the end user to execute queries and complete other tasks. It's fairly feature-complete, so if you want to limit what your users can do, you have to turn the menu and toolbars off completely. If you don't mind hand-editing XML files, there is a way that you can limit which buttons and menus show up or even create your own. However, this is not a supported use, so it would be helpful if the authors would include this in the next version.

All of the code, as well as the XML describing an application's forms, is stored as text either in the filesystem or in the database. If you're developing a potentially lucrative application, it might be best to wait until encryption and Web application creation is complete. Or, investigate technologies such as FreeNX that allow you to deliver the application as a service. Purchasing a runtime library to distribute with your application would allow you to restrict the things your end user could do, as the runtime libraries don't include the development tools. But, doing so would not prevent savvy users from downloading the full version of Rekall and taking the code from your application to use in their own or editing those files to add their own functionality. When the Professional version of Rekall is released, you will be able to scramble your applications on a per-client basis with private key encryption.

Of course, having everything in plain-text XML also is a lifesaver. Recently, I realized that I had a series of component groups, otherwise known as blocks, that were misconfigured. Instead of having to cut and paste or redo the layout of 11 fields in each block, I simply modified the XML defining the blocks to point to the tables instead of the queries, and it worked flawlessly.

Getting Rekall

After developing for a while on the Linux version of Rekall I downloaded as source, I tested my application on the version I purchased as a Windows install. My application looked identical in both environments. In fact, porting to Windows was a breeze: I simply performed a backup of the PostgreSQL database on my Linux box and copied that file, as well as the Rekall project file, to the Windows box. There, I restored the backup file to the Windows version of PostgreSQL 8.0.2 and ran the project file in the Windows version of Rekall.

The Windows install is reasonably sized at about 7MB. You have to be careful, though, to install the right version of Python before you install Rekall. For Rekall 2.2.3, the current version, you need any Python version in the 2.3 line.

Compiling under Linux was fairly straightforward. Under CentOS 3, the compile produced no errors and installed cleanly. After upgrading to CentOS 4, I received compile errors during the install, but the application did install cleanly. The compile process took about an hour on my relatively old 900MHz Athlon-equipped CPU with 512MB of RAM.

The first time you run Rekall, it presents you with a series of dialog boxes to configure certain behaviors of the development environment. Among these behaviors are the verification of record updates and deletes as well as the layout of the development environment. After this occurs the first time, it doesn't happen again unless the user's Rekall configuration file is deleted.

An Example Application

To demonstrate the ease of development in Rekall, I put together a small application that tracks philosophers and their writings. This example provides a good examination of both the ease of development and the Python scripting capabilities of Rekall. The end goal was an application similar to what you see in Figure 1—a small data-entry window with two tabs, a report and a status bar at the bottom describing the current philosopher.

Figure 1. The Completed Philosopher Application

Before you begin, you need to configure PostgreSQL to work well with Rekall. Configure PostgreSQL to support connections from local applications, as is shown in the Configuring PostgreSQL sidebar. After this is done, add a user philosophy_major with a password and create the philosophers database. This all must be done before the initial connection by Rekall, because Rekall does not have the capability to add users or databases. After a reload of the PostgreSQL server, it is ready for you to connect.

Now, we need to create the project. Open a terminal window and make a directory for the project. Then, fire up Rekall by typing `rekall` on the command line. After installation, the rekall executable should be under the /usr/bin directory.

Starting a project is straightforward. After clicking on the New button, you are presented with a project creation wizard. On the first screen, you enter the directory where your information should be stored, as well as the name of the project, the Database Name. In the next dialog window, tell it where to store the forms, reports and other XML structures. These items can be stored either in the database, in a special Rekall Objects table or as files in the filesystem. After you choose that, select which database platform you're using. You can have multiple data sources, each running a different driver. This particular dialog controls the main database.

The next two dialogs allow you to enter the database host and port as well as the user name/password pair you want to use to connect. Then, upon a successful connection, you specify which database you want to use.

After you select the database, a screen similar to the one shown in Figure 2 appears. Now you can start building your

application. The first step in application building is to create the tables. To do so, click on the Tables item in the Objects tree. Then, select the correct server and double-click on Create new table. The table builder, as shown in Figure 3, then appears.

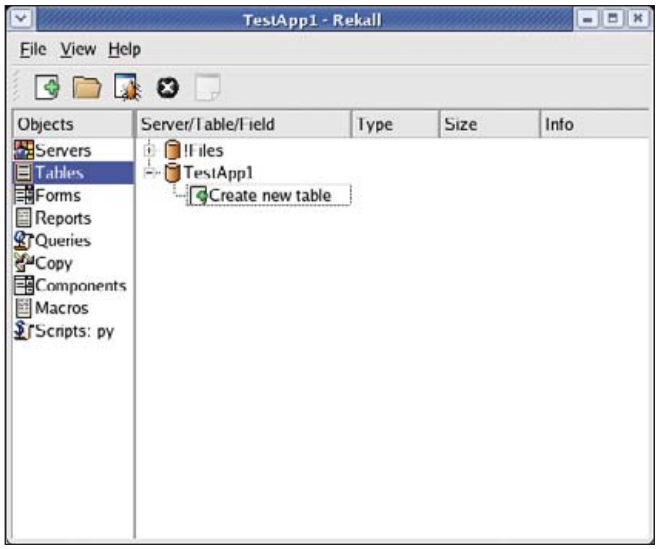


Figure 2. From the main project window, you can drill down to set up tables and create forms and reports.

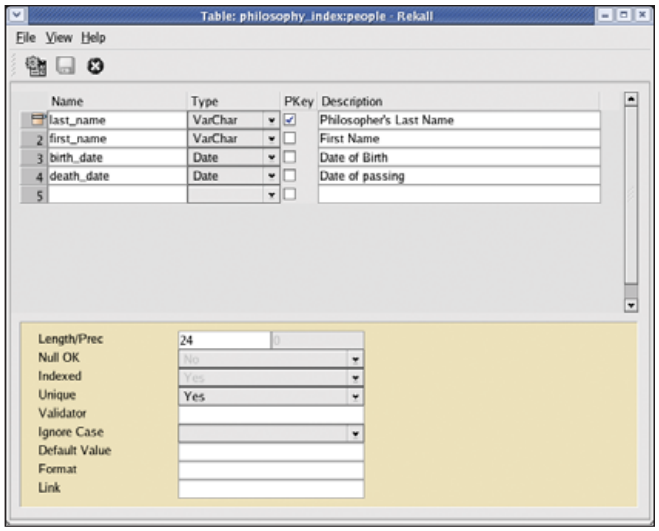


Figure 3. Create a new table using the Table window.

Rekall can't use tables that already exist. It would be an interesting exercise to write a utility to convert database schemas to Rekall definition files. Right-clicking your project name listed in the Tables tree reveals a menu that allows you to import a table definition, and this is contained in an XML file that could be built from SQL table definitions.

Next, it's time to build some forms from the database schemas you've created. Fortunately, Rekall provides some easy tools with which to do so. When you click on the Forms item in the Objects tree and expand the item named after your project, you can create a form or create a form with a wizard. Even when creating complex applications that use things such as tabbed pages, it's helpful first to create each page as a separate form, copy those objects and then paste them into the appropriate blocks on your form.

In Figure 4, you can see how I've used the wizard to create a form based on the philosophers table. You have a large amount of control over your forms, even when you use the wizard. Not only does it ask you for a table and the fields to use, you also can specify multiple records per page, field formats and the tools that should be added automatically to the page. In the form in Figure 4, the wizard added the buttons you see along the bottom of the screen, as well as the small navigation tool at the very bottom. Called a Nav. Tool, this widget allows you to navigate through the database on a record-by-record basis.

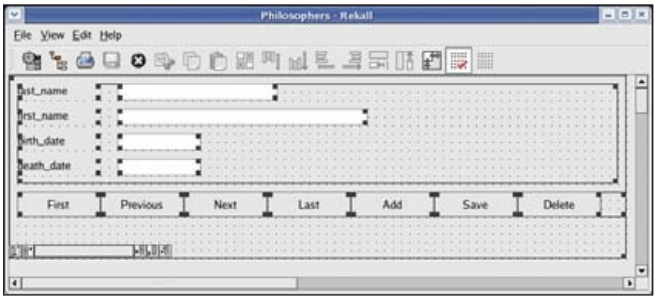


Figure 4. The form wizard does most of the work for you, but you have a lot of flexibility in designing the form.

After creating the forms for Philosophers and their Publications, as seen in Figure 5, I need to combine them into one window. This is where it is advantageous to create a form by hand, so that I can add components at will. Before I do that, however, we need to examine the concept of blocks.

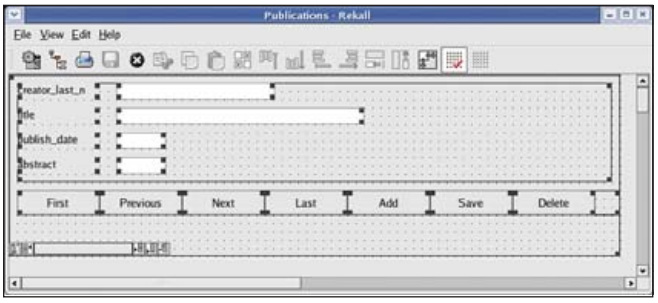


Figure 5. The publications form will be part of a notebook window, with the main Philosopher form shown in Figure 4.

In a Rekall form, data is represented in an entity called a block. There are several types of blocks, the simplest being a table block. Other types of blocks include query blocks and SQL blocks, which display data retrieved from a query. A form can have any number of blocks. These blocks can exist side by side or in any configuration on the same visual plane. Alternatively, they can exist as pages in a notebook by way of a tabbed-page control. This is what I exploit to get to my one-window goal.

To create my notebook, I create a new form without using

the wizard. After I select that command, I get the Form Attribute window shown in Figure 6. As you can see, I've configured a basic window. It's not stretchable, there is no status bar and the top-level block type is null— it's called a menu block in the selection dialog. Setting the top-level block type to menu only or null is what allows us to place blocks and controls arbitrarily on the form.

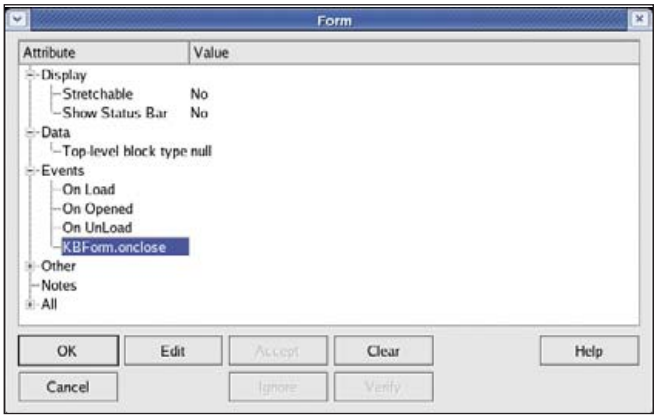


Figure 6. Creating a Rekall Form with a Null Top Block

After going through each attribute by clicking on it, making a selection and pressing the Accept button at the bottom of the dialog, I am presented with a Block attribute window that is much like the Form Attribute page. Because this is simply a menu block, I safely can accept many of the default values, as most values are appropriate only for table or query blocks. That action nets me a blank page, which I can resize appropriately and add the tabber control.

On each page of the tabber control, I simply create a table block corresponding to the correct table. Then, I copy all of the contents from the block contained in the Philosophers form to the corresponding block in the Philosophers tabber page, and I repeat the action for the Publications page. After doing so, I dress up the fields with their correct names, modify the size of the Abstract field on the Publications page and my notebook is complete.

Listing 1. The On Display Function for the Philosophers Block

```
def eventFunc (block, row) :
    someMainForm = block.getForm();
    currBlock = block;
    dataLabel = someMainForm.getNamedCtrl("current_philosopher");
    dataLabel.setText(currBlock.getNamedCtrl("last_name").getValue());
```

Listing 2. The On Display Function for the Publications Block

```
def eventFunc (block, row) :
    mainForm = block.getForm();
    currBlock = block;
    dataLabel = mainForm.getNamedCtrl("current_philosopher");
    currBlock.setUserFilter(dataLabel.getValue());
```

CONFIGURING POSTGRESQL

As it is configured after a default installation, PostgreSQL 8.0.2 authenticates its users by checking their Linux identities. To create a more secure application, you should change this to password authentication. The following steps describe how to do so.

First, modify the password of the database user postgres so that you can log in when passwords are required:

1. At a command prompt, type `su` and enter your root password.
2. Then, type `su postgres.`
3. Now, start the psql monitor by typing `psql template1.`
4. We modify the password by typing `alter user postgres with password 'pgUser89'` or some other suitable password.
5. Exit the monitor by typing `\q` and pressing Enter.

Second, modify the pg_hba.conf file so that the database accepts md5 passwords for all connections. By default, it's configured to authenticate based on the identity of the current Linux account. On a default installation, this file is found under /var/lib/pgsql/data. This file has lines that look like this:

```
# "local" is for Unix domain socket connections only
local   all     all                             trust
# IPv4 local connections:
host    all     all     127.0.0.1/32            md5
# IPv6 local connections:
host    all     all     ::1/128                 md5
```

To enable passwords, change the `trust` option on the line for local to md5 and save the file. Then, restart PostgreSQL. On a Red Hat-like system, this can be done by issuing the command `/sbin/service postgresql reload.`

After this is done, users and databases can be created by using PostgreSQL's built-in tools or by using third-party tools such as PgAdminIII. The PostgreSQL Web site always is the best resource for more information on these topics.

Now, I need to find a way for the Philosopher page to set the key that is used to look up the publications. I do this through the use of the status label at the bottom of the main form, underneath the tabber page control. This label's text value is set whenever a database lookup is performed or a new entry is created. The label is set when the code, shown in Listing 1, is run by way of a callback set by the On Display event for the Philosophers block. When the Publications block is shown, the On Display event is called for that block, which sets a user filter on the data shown in that block. The user filter code can be seen in Listing 2.

Finally, I need some way of listing the philosophers I have in my database. Figure 7 shows the report functionality of Rekall.

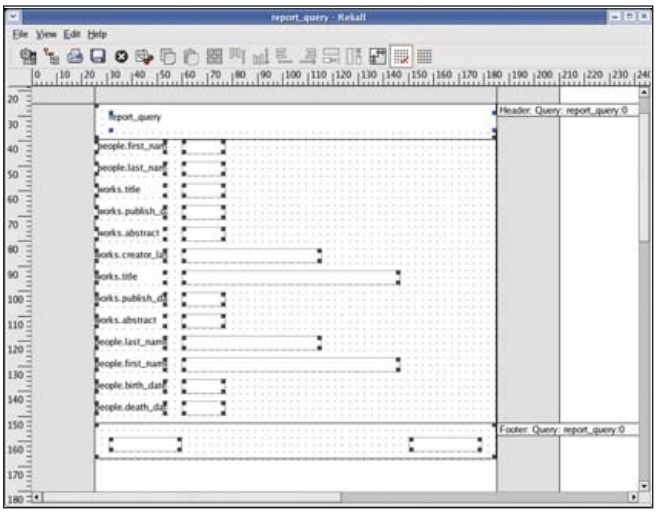


Figure 7. The report design window builds report layouts as easily as designing a form.

Rekall also offers a number of other components, such as reports, queries and data copiers. Each component can be created with the same ease and offers the same versatility as the forms.

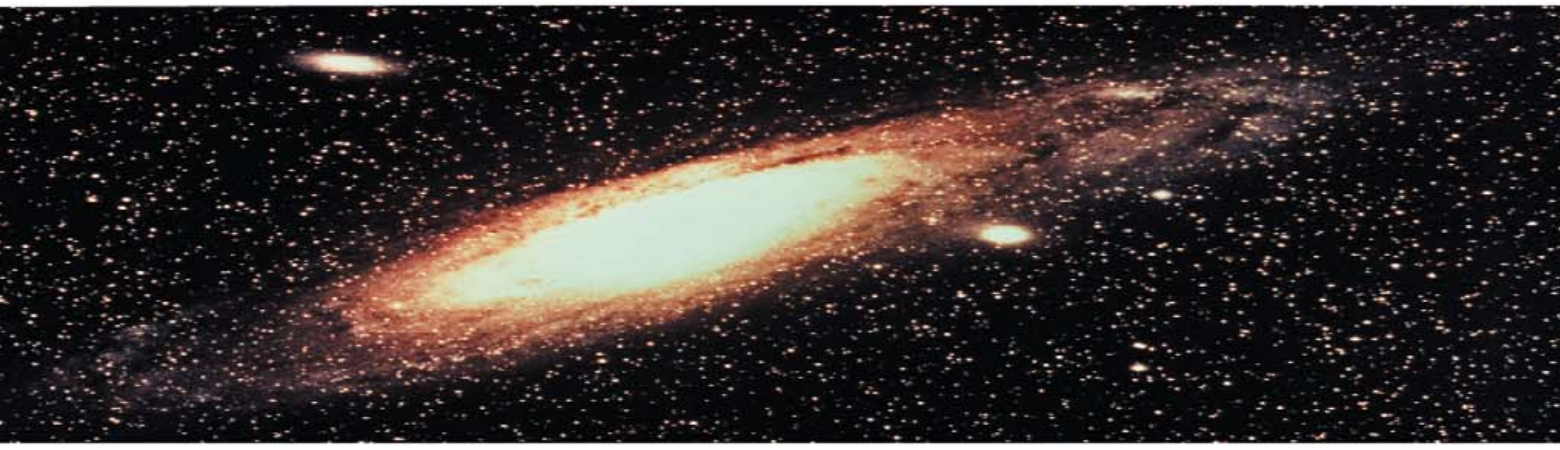
Conclusion

As I have demonstrated, Rekall and PostgreSQL offer the ability to complete all kinds of database programming tasks quickly under Linux while providing the cross-platform capability that many consultants need. As companies migrate to Linux for their desktops, products such as Rekall will come into far greater demand.

Resources for this article: www.linuxjournal.com/article/8271.

Joshua Bentham will graduate soon from Capital University in Bexley, Ohio, with a BA in Philosophy. He has been using Linux since kernel version 1.2.8. His Weblog is at **www.globalherald.net/jb**, and he can be reached at jb42@globalherald.net.

MORE SPACE. LESS MONEY.



UNLIMITED AFFORDABLE NETWORK STORAGE

Everybody needs more space. And they need to spend less money. What if you can both have more space and spend less money?

What if you could put 7½ terabytes in only 3 rack units? What if that 7½ terabytes cost less than $10,000? Including the SATA disk drives. Imagine if you could glue it all together with a RAID appliance into one system. What if you could add as much storage as you wanted, one shelf at a time, and never have to 'fork-lift' anything?

Coraid's new SATA EtherDrive Storage allows you to do just that. Using industry standard SATA disk drives, EtherDrive Storage connects disks directly to your Ethernet network. Each disk appears as a local drive to any Linux, FreeBSD or Solaris system using our open ATA-over-Ethernet (AoE) protocol .Since the disks just appear as local drives you already know how to use them.

The EtherDrive® SATA Storage Shelf is a 3U rack mount network appliance that contains 15 SATA drive slots. Its triple redundant power supply protects you from your most likely failure. Its dual Gb Ethernet interfaces allow your data to go fast; 200MB per second. And at a very affordable price. List price for the EtherDrive Storage Shelf, without disks, is only $3,995.

Our companion product, the RAIDBlade RAID controller, allows an virtually unlimited number of Storage Shelves to be combined into a set of logical AoE storage devices.

Now you can have unlimited storage at a very affordable price. For complete information, visit our website at www.coraid.com, or call, toll-free, 1-877-548-7200. And we'll show how we've made network storage so affordable, you can have all the space you want.

www.coraid.com
info@coraid.com
1.877.548.7200

CORAID

WhisperStation

Microway introduced its new WhisperStation workstation, designed for use in demanding design applications or small- to medium-sized cluster environments. WhisperStation features dual AMD Opteron or Intel Xeon EM64T processors, an NVIDIA FX1300 PCI Express Graphics engine, 2GB of memory, ultra-quiet fans and power supply and a Viewsonic 20" LCD monitor. WhisperStation is available with a 64-bit install of Red Hat, SUSE or Gentoo, or Microsoft Windows. It can be custom-configured to exact user specifications, including large hard disks, RAID and specialized applications.

CONTACT Microway Incorporated, Plymouth Industrial Park, 12 Richards Road, Plymouth, Massachusetts 02360, 508-746-7341, www.microway.com.

Comet12 Notebook

Tadpole Computer's Comet12 is a Federal Information Processing Standards (FIPS) 140-2 compliant wireless mobile notebook designed for government users. Based on Tadpole's Comet line of wireless Sun Ray ultra-thin clients and Fortress Technologies' family of secure wireless gateways, the Comet12 provides secure encrypted wireless networking for federal network and communications systems that handle sensitive information. The technology behind Comet12 enables user privacy, access control, device and user authentication and data link layer integrity to guard against denial-of-service attacks. The Comet12 has a 12.1" TFT-LCD XGA display, measures 10.4" × 8.7" × 0.9" and weighs about three pounds.

CONTACT Tadpole Computer, 20450 Stevens Creek Boulevard, 3rd Floor, Cupertino, California 95014, 408-973-9944, www.tadpolecomputer.com.

The Orion PMC or PMI

Curtiss-Wright Controls Embedded Computing introduced a new dual-channel video compression/decompression board, the Orion, available in both PMC and PCI form factors for use in VME, CompactPCI and desktop PCI systems. The Orion features dual on-board JPEG 2000 engines to support full-frame encoding of standard 625-line PAL or 525-line NTSC composite video. In input mode, Orion accepts up to ten single-ended or four differential analog PAL or NTSC video inputs, two of which can be selected for simultaneous JPEG 2000 compression. In output mode, Orion receives one or two JPEG 2000 data streams through the 64-bit, 66MHz PCI bus, decompresses the data streams and outputs the resulting one or two independent PAL or NTSC video output signals by way of the 20-way MDR sockets located on the board's front panel. As part of a video capture and recording system, the resulting compressed video streams can be stored locally or distributed over a network to a remote display. Software support for Orion includes drivers for PowerPC-based Linux, with options available for other platforms. A low-level driver and comprehensive board support library provide a set of C functions for the card that can be ported to many OSes and host processor platforms. The Orion is available in either PMC or PCI form factors.

CONTACT Curtiss-Wright Controls, 3120 Northwest Boulevard, Gastonia, North Carolina 28052-1167, 704-869-4600, www.cwcembedded.com.

NetTracker Lite

Sane Solutions, developers of the NetTracker line of Web analytics software, has released NetTracker Lite, a free-of-charge version of NetTracker that provides owners of small and low-traffic Web sites with robust Web site traffic analysis software at no cost. Features of NetTracker Lite include deep, detailed reporting that allows users to see all Web site traffic data; true drill-down functionality, so users can access important details from any report; and advanced filtering capabilities, such as where traffic is referred from at a certain time. NetTracker Lite's user interface provides a calendar display from which users can select the time frame they want to view. Reporting options include sharing Web browser-based reports on-line, exporting data to various document formats and e-mailing reports to others. NetTracker Lite supports Linux, FreeBSD, Mac and Windows platforms and can be downloaded for free at www.nettrackerlite.com.

CONTACT Sane Solutions, LLC, 35 Belver Avenue, North Kingstown, Rhode Island 02852, 800-407-3570, www.sane.com.

PRIMEQUEST Servers

Fujitsu Computer Systems announced the new PRIMEQUEST server line, based on the Intel Itanium 2 processor. The PRIMEQUEST server line combines data center-class fault immunity and high system scalability for industry-standard environments running Red Hat Enterprise Linux, Novell/SUSE Linux Enterprise Server and Windows Server 2003 for Itanium-based systems. The PRIMEQUEST server architecture provides platform fault immunity, with up to eight highly available, independent and hardware-isolated partitions, each one being a fully independent server within the system. In addition, the chipset on the PRIMEQUEST server offers system mirror mode and flexible I/O capabilities to ensure high availability. PRIMEQUEST servers also feature redundant management networks hosting SSL protocols, integrated Gbit switching and hub connections, integrated SCSI hard drives and integrated KVM/USB for simplified partition management.

CONTACT Fujitsu Computer Systems, 1250 East Arques Avenue, MS122, Sunnyvale, California 94085, 800-831-3183, us.fujitsu.com/computers.

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.

HP WORLD 2005 Conference & Expo

August 14 – 18
Moscone Center
San Francisco

The industry's largest independent HP-centric technical training event.

KEYNOTE SPEAKERS

Ann Livermore
Executive Vice President, Technology Solutions Group, HP

Kris Hagerman
Executive Vice President Storage and Server Management Group, VERITAS Software

Patrick J. Martin
Chairman, President, and CEO, StorageTek

Chuck Rozwat
Executive Vice President, Server Technologies, Oracle

Premier Sponsors

hp invent · VERITAS · ORACLE · StorageTek

Gold Sponsors

BROADCOM. Connecting everything · ca Computer Associates

Join us for a week of in-depth technical training that will sharpen your Linux skills and provide you with the tools you need to deploy Linux as a mainstream enterprise solution.

At HP World you'll learn Linux support and administration, database deployment under Linux, Linux clustering, Linux virtualization technologies and securing Linux.

Plenary Speakers

Learn about trends and initiatives in the open-source movement from **Bruce Perens**, the noted leader of the Free Software and Open Source Community and creator of the Open Source Definition, and **Bdale Garbee**, Chief Technology Officer, Linux Enterprise Storage and Solutions for HP.

Linux Certification

Join us for on-site certification including LPI Exams (Level 1 and Level 2)

Some of the tutorials and break-out sessions available are:

- Linux for UNIX Administrators
- Practical Linux Server Security for the New Linux Administrator
- Harnessing the Power of Open Source in Your Software Architecture
- Experiences with F/OSS Business Software
- Validation Issues for the Use of Linux in FDA-Regulated Environments
- Virtualization Technologies in Linux: A Survey
- OpenIB InfiniBand Performance
- Building Clustered Linux Systems
- Open Source Security Technology
- Linux Server Deployments in the Enterprise
- Eclipse and Linux: An IDE, an Application Platform, and More

For more information go to www.hpworld.com/sessions

Sign-up for your Interex membership and save $200!

SPECIAL DISCOUNT:
15% Discount for Linux Journal Readers
Use Discount Code: LJAD15

Produced by interex shared knowledge • shared power — The International Association of HP Computing Professionals

Register today at www.hpworld.com

Measuring and Improving Application Performance with PerfSuite

Get a realistic view of how your program runs on real hardware, so you can find small changes that make a big performance difference.

BY RICK KUFRIN

At some point, all developers of software applications, whether targeted to Linux or not, are likely to spend at least a small amount of time focusing on the performance of their applications. The reason is simple: many potential benefits can be gained from tuning software for improved performance. For example, in the scientific and engineering arenas, performance gains can make the difference between running smaller-scale simulations rather than larger and potentially more accurate models that would improve the scientific quality of the results. Applications that are more user-oriented also stand to benefit from improvements that result in faster responsiveness to the user and an improved overall user experience.

Although microprocessor improvements over the past decade or so have made clock speeds well in excess of the gigahertz range commonplace, most developers are aware that a tenfold increase in processor frequency does not guarantee a tenfold reduction in the runtime of your application. Additionally, for those developing software for distribution to others, attention to performance and responsiveness can pay big dividends when you consider that your end user may be running your application on a mid-1990s era 100MHz Pentium processor.

This article is an introduction to a set of open-source software tools called PerfSuite that can help you to understand and possibly improve the performance of your application under Linux. PerfSuite consists of several related tools and libraries targeted at several different activities useful in performance-oriented analysis.

The development of PerfSuite was motivated by my own experiences in working with not only applications that I had developed, but a number of large supercomputer-class applications in both academic and corporate settings. After having worked with several research groups, I realized that developers often take advantage of only a limited set of tools that may be available to them. They typically rely on traditional time-based statistical profiling techniques such as gprof.

Of course, gprof-style profiles are invaluable and should be the mainstay of any developer's performance toolbox. However, the microprocessors of today, such as those on which you probably are using Linux, offer advanced features that can provide alternative insights into characteristics that directly affect the performance of your software. In particular, nearly all microprocessors in common use today incorporate hardware-based performance measurement support in their designs. This support can provide an alternative viewpoint of your software's performance. While time-based profiles tell you *where* your software spends its time, hardware performance measurements can help you understand *what* the processor is doing and how effectively the processor is being utilized. Hardware measurements also pinpoint particular reasons why the CPU is stalling rather than accomplishing useful work.

Hardware Performance Counter Basics

The first time I encountered the term hardware performance counters, it was in the context of having access to multi-million-dollar supercomputers where every CPU cycle is critical and research teams spend substantial amounts of time tweaking their codes in order to extract maximum performance from the system. Often, software is tailored explicitly for each type of computer on which it is to be run. Research teams sometimes pore over the numbers generated by these performance counters to measure the exact performance of their applications and to ferret out places where they might gain additional speedup. Needless to say, this all sounded exotic to me. But the purpose and function of the counters turned out to be simple: they are extra logic added to the CPU that track low-level operations or events within the processor accurately and with minimal overhead.

For example, even if you're not an expert in computer architecture, you probably already know that nearly all processors in common use are cache-based machines. Caches, which offer much higher-speed access to data and instructions than what is possible with main memory, are based on the principles of temporal and spatial locality. Put another way, cache designs hope to take advantage of many applications' tendency to reuse blocks of data not long after first use (temporal locality) and also to access data items near those already used (spatial locality). If your application follows these patterns, you have a much greater chance of achieving high performance on a cache-based processor. If not, your performance may be disappointing. If you're interested in improving a poorly performing application,

your next task is to try to determine why the processor is stalling instead of completing useful work. This is where performance counters may help.

It takes a little research to learn which performance counters are available to you on a particular processor. Each CPU has a different set of available performance counters, usually with different names. In fact, different models in the same processor family can differ substantially in the specific performance counters available. In general, the counters measure similar types of things. For example, they can record the absolute number of cache misses, the number of instructions issued, the number of floating-point instructions executed and the number of vector, such as SSE or MMX, instructions. The best reference for available counters on your processor are the vendor's technical reference on the processor, often available on the Web.

Another complication is kernel-level support is needed to access the performance counters. Although the Itanium (IA-64) kernel provides this support through the perfmon driver in the official kernel (authored by Stephane Eranian of HP Research), the standard x86 Linux tree currently does not.

Fortunately, efforts are underway to address these issues. The first is the development of a performance monitoring driver for the x86 kernel called perfctr. This is a very stable kernel patch developed by Mikael Pettersson of Uppsala University in Sweden. The perfctr kernel patch is becoming more widely adopted by the community and continually is improved and maintained. The second is an effort from the Innovative Computing Laboratory at the University of Tennessee-Knoxville called PAPI (Performance Application Programming Interface). PAPI defines a standard set of cross-platform performance monitoring events and a standard API that allows measurement using hardware counters in a portable way. The PAPI Project provides implementations for the library on several current processors and operating systems, including Intel/AMD x86 processors, Itanium systems and, most recently, AMD's x86-64 CPUs. On Linux, PAPI uses the perfmon and perfctr drivers as appropriate. Refer to the on-line Resources for references where you can learn much more about perfctr, perfmon and PAPI.

PerfSuite, which is discussed in the remainder of this article, builds upon PAPI, perfmon and perfctr to provide developers with an even higher-level user interface as well as additional functionality. A main focus of PerfSuite is ease of use. Based on my experiences in working with developers interested in performance analysis, it became clear that an ideal solution would require little or no extra work from users who simply want to know how well an application is performing on a computer. They want to know this without having to learn many details about how to configure or access the performance data at a low level.

Using Performance Counters to Measure Application Characteristics

Let's say that you do have an application that isn't cache-friendly—what might happen? In the worst-case scenario, rather than loading a line of data into the cache and operating on the data

contained in that line repeatedly, it may use only one piece of data and then be done with it. The next piece of data you need may require another cache line to be loaded and so forth. Each of these cache loads are relatively expensive and can result in reduced performance, because the processor is waiting primarily for the data it needs to become available. Each time the next piece of data is required, the processor attempts to load it from data already resident in the cache. If it's not found, a cache miss occurs and a corresponding hardware event is signaled. The higher the ratio of cache misses to hits, the more likely it is that the overall performance of the software degrades.

Listing 1 shows a basic but concrete example of how this might occur. The listing shows a loop that initializes each element of a matrix using the sum of the corresponding element of another matrix and a vector. Because the C language stores data in row-major order, the loop as written does not access neighboring data elements in the two matrices. Fortunately, this problem has a simple solution: interchange the nested loops so the matrices are processed on a row-by-row basis. This pattern of array access also is referred to as stride-one access. Many optimizing compilers perform this type of loop-interchange optimization automatically, depending on the optimization level you select.

Listing 1. Loop from a Program with Cache-Unfriendly Behavior

```
for (j = 0; j < COLS; j++)
     for (i = 0; i < ROWS; i++)
          a[i][j] = b[i][j] + c[i]
```

Test cases containing these two versions of the loop were compiled with a recent release of Intel's ICC compiler, run on a Pentium III computer and timed. The result of this simple change sped up the loop by a factor of ten. Not unexpectedly, the overall level 2 cache miss count decreased considerably for the optimized version of the loop (212,665,026 versus 25,287,572—see the next section for more information).

Often, it's useful to combine the raw hardware performance counts into a derived metric that can provide a normalized view of performance. For example, one of the most widely used metrics for performance measurement describes the average number of cycles required to complete an instruction (CPI). By counting the total number of cycles and instructions retired (both of which are available as hardware events), we easily can obtain this metric. Similarly, we might be interested in knowing, on average, how often a piece of data was reused once it was resident in the cache. By counting the appropriate cache-related events and combining them into a single metric, we can obtain an approximation of this information as well.

PerfSuite's hardware performance counter tools and libraries provide easy access to both the raw measurement data as well as a large number of derived metrics that you can use to learn about and hopefully improve the performance of your application. In its most basic use, PerfSuite requires nothing more than a slight modification to the command you execute to run your program. If your executable is in the file myprog, then instead of running myprog directly, you instead would enter `psrun myprog`. If all goes well, the output of psrun is an XML document that contains a standard set of hardware events along with additional information about the CPU. You can translate this XML document into a comprehensive performance report with the command `psprocess`, supplying it with the name of the XML file.

PerfSuite Basics

The current release of PerfSuite includes the following four tools for accessing and working with performance data:

- psrun: a utility for hardware performance event counting and profiling of single-threaded, POSIX threads-based and MPI applications.
- psprocess: a utility that assists with a number of common tasks related to pre- and post-processing of performance measurements.
- psinv: a utility that provides access to information about the characteristics of a machine (for example, processor type, cache information and available performance counters).
- psconfig: a graphical tool for easy creation and management of PerfSuite configuration files.

This section demonstrates the two commands psrun and psprocess. Visit the PerfSuite Web site for more information about and examples of the use of psinv and psconfig.

The easiest way to learn to use the basic PerfSuite tools is try them out on your own programs. Here is a sequence of commands you might enter to run the simple cache example discussed earlier with performance measurement enabled. Also shown are the current contents of the directory after each run with psrun to show that XML documents are created:

```
1% ls
badcache
goodcache

2% psrun badcache

3% ls
badcache
goodcache
psrun.22865.xml

4% psrun goodcache

5% ls
badcache
goodcache
psrun.22865.xml
psrun.22932.xml

6% psprocess psrun.22865.xml
7% psprocess psrun.22932.xml
```

FREE SOFTWARE
FOUNDATION

be FREE

Join us at member.fsf.org

Listing 2. psprocess Output from the Cache-Unfriendly Version of the Loop

```
PerfSuite Hardware Performance Summary Report

Version                  : 1.0
Created                  : Thu Feb 19 22:43:01 2004
Generator                : psprocess 0.2
XML Source               : psrun.22865.xml

Processor and System Information
================================================
Node CPUs                : 2
Vendor                   : Intel
Family                   : Pentium Pro (P6)
CPU Revision             : 6
Clock (MHz)              : 997.173
Memory (MB)              : 1510.82
Pagesize (KB)            : 4

Cache Information
================================================
Cache levels             : 2
--------------------------------
Level 1
Type                     : instruction
Size (KB)                : 16
Linesize (B)             : 32
Assoc                    : 4
Type                     : data
Size (KB)                : 16
Linesize (B)             : 32
Assoc                    : 4
--------------------------------
Level 2
Type                     : unified
Size (KB)                : 256
Linesize (B)             : 32
Assoc                    : 8

Index Description                        Counter Value
================================================
 1 Conditional branch instructions........ 52663367
 2 Branch instructions.................... 52650952
 3 Conditional branch ins mispredicted..... 112009
 4 Conditional branch instructions taken.. 52610596
 5 Branch target address cache misses..... 31020
 6 Requests for excl acc to clean cache line.. 1165
 7 Requests for cache line invalidation.......... 0
 8 Requests for cache line intervention..... 32801
 9 Requests for excl acc to shared cache ln.. 26537
10 Floating point multiply instructions.......... 0
11 Floating point divide instructions............ 0
12 Floating point instructions........... 208155552
13 Hardware interrupts..................... 22134
14 Total cycles....................... 21407855039
15 Instructions issued.................. 2010041200
16 Instructions completed............... 624104056
17 Vector/SIMD instructions..................... 0
18 Level 1 data cache accesses........... 678945043
19 Level 1 data cache misses............. 244760094
20 Level 1 instruction cache accesses.. 21332388384
21 Level 1 instruction cache misses.......... 22546
22 Level 1 instruction cache reads..... 21309322857
23 Level 1 load misses................... 244318153
24 Level 1 store misses...................... 9852
25 Level 1 cache misses.................. 243826788
26 Level 2 data cache reads.............. 243745402
27 Level 2 data cache writes................. 10317
28 Level 2 instruction cache accesses........ 24335
29 Level 2 instruction cache reads........... 21362
30 Level 2 cache misses.................. 212665026
31 Cycles stalled on any resource...... 21057880641
32 Instruction TLB misses...................... 64

Statistics
================================================
Counting domain.............................. user
Multiplexed.................................... yes
Graduated floating point ins. per cycle...... 0.010
Vector ins. per cycle........................ 0.000
Floating point ins per graduated ins ........ 0.334
Vector ins per graduated ins ................ 0.000
Floating point ins per L1 data cache access.. 0.307
Graduated ins per cycle...................... 0.029
Issued ins per cycle......................... 0.094
Graduated ins per issued ins................. 0.310
Issued ins per L1 ins cache miss......... 89152.896
Graduated ins per L1 ins cache miss...... 27681.365
Level 1 ins cache miss ratio................. 0.000
Level 1 data cache access per graduated ins.. 1.088
% floating point ins of all graduated ins... 33.353
% cycles stalled on any resource............. 98.365
Level 1 ins cache misses per issued ins...... 0.000
Level 1 cache read miss ratio (instruction).. 0.000
Level 1 cache miss ratio (data).............. 0.361
Level 1 cache miss ratio (instruction)....... 0.000
Bandwidth used to level 1 cache (MB/s)..... 363.437
Bandwidth used to level 2 cache (MB/s)..... 316.988
MFLIPS (cycles).............................. 9.696
MFLIPS (wall clock).......................... 9.530
MVOPS (cycles)............................... 0.000
MVOPS (wall clock)........................... 0.000
MIPS (cycles)................................ 29.071
MIPS (wall clock)............................ 28.572
CPU time (seconds)........................... 21.469
Wall clock time (seconds).................... 21.843
% CPU utilization............................ 98.285
```

Listing 3. Part of the psprocess output from the optimized version of the loop. The Processor and System Information and Cache Information sections are the same.

```
Index Description                          Counter Value
=================================================
 1 Conditional branch instructions........ 49627213
 2 Branch instructions.................... 49971420
 3 Conditional branch ins mispredicted..... 97630
 4 Conditional branch ins taken........... 49089592
 5 Branch target address cache misses..... 3816
 6 Requests for excl access to clean cache ln.  820
 7 Requests for cache line invalidation.... 0
 8 Requests for cache line intervention..... 2796
 9 Requests for excl access to shared cache ln. 494
10 Floating point multiply instructions..... 0
11 Floating point divide instructions....... 0
12 Floating point instructions.......... 189564951
13 Hardware interrupts..................... 2577
14 Total cycles....................... 2471179766
15 Instructions issued.................. 513936102
16 Instructions completed............... 509580537
17 Vector/SIMD instructions.................. 0
18 Level 1 data cache accesses.......... 372965600
19 Level 1 data cache misses............. 23010188
20 Level 1 instruction cache accesses... 2769671237
21 Level 1 instruction cache misses........ 2369
22 Level 1 instruction cache reads...... 2746595553
23 Level 1 load misses.................. 25980065
24 Level 1 store misses..................... 995
25 Level 1 cache misses................. 25772544
26 Level 2 data cache reads............. .25617201
27 Level 2 data cache writes................ 935
28 Level 2 instruction cache accesses..... 2405
29 Level 2 instruction cache reads........ 2652
30 Level 2 cache misses................. 25287572
31 Cycles stalled on any resource..... 2199590592
32 Instruction TLB misses.................... 0

Statistics
=================================================
Counting domain.......................... user
Multiplexed.............................. yes
Graduated floating point ins per cycle..... 0.077
Vector ins per cycle.....................0.000
Floating point ins per graduated ins....... 0.372
Vector ins per graduated ins............... 0.000
Floating point ins per L1 data cache access. 0.508
Graduated ins per cycle...................0.206
Issued ins per cycle......................0.208
Graduated ins per issued ins............... 0.992
Issued ins per L1 ins cache miss..... 216942.213
Graduated ins per L1 ins cache miss.... 215103.646
Level 1 ins cache miss ratio............... 0.000
Level 1 data cache access per graduated ins. 0.732
% floating point ins of all graduated ins.. 37.200
% cycles stalled on any resource.......... 89.010
Level 1 ins cache misses per issued ins..... 0.000
Level 1 cache read miss ratio (instruction). 0.000
Level 1 cache miss ratio (data)............ 0.062
Level 1 cache miss ratio (instruction)..... 0.000
Bandwidth used to level 1 cache (MB/s).... 332.792
Bandwidth used to level 2 cache (MB/s).... 326.530
MFLIPS (cycles)............................ 76.493
MFLIPS (wall clock)........................ 66.787
MVOPS (cycles)............................. 0.000
MVOPS (wall clock)......................... 0.000
MIPS (cycles).............................. 205.626
MIPS (wall clock)......................... 179.533
CPU time (seconds)......................... 2.478
Wall clock time (seconds).................. 2.838
% CPU utilization.......................... 87.310
```

Listings 2 and 3 show the output of the psprocess command for the unoptimized and optimized versions of the test program; these listings have been edited slightly to fit in the available space. As you can see, a substantial amount of information is gathered during the course of the measurement and the report includes not only the raw event counts measured using PAPI, but also a series of metrics that can be derived from the counts.

Customizing Your Performance Analysis

psrun determines the performance events to be measured by consulting a configuration file you can supply, which is an XML document that describes the measurements to be taken. If you don't supply a configuration file, a default is used (the output shown in Listings 2 and 3 used the default). As an XML document, the configuration file is straightforward to modify and read. For example, if you wanted to obtain the raw events required to calculate the CPI metric discussed earlier, you'd need to ask psrun to measure the total number of graduated instructions and the total number of cycles. These events are predefined in PAPI and are called PAPI_TOT_INS and PAPI_TOT_CYC, respectively. Listing 4 shows a

Listing 4. An Example PerfSuite XML Configuration Document

```
<?xml version="1.0" encoding="UTF-8" ?>
<ps_hwpc_eventlist class="PAPI">
  <!-- ================================================
       Configuration file to measure graduated instructions
       and total cycles.
       ================================================ -->
  <ps_hwpc_event type="preset" name="PAPI_TOT_INS" />
  <ps_hwpc_event type="preset" name="PAPI_TOT_CYC" />
</ps_hwpc_eventlist>
```

Listing 5. A source code profile generated by psprocess based on level 2 cache misses (output has been truncated to fit in available space).

```
PerfSuite Hardware Performance Summary Report


Profile Information
==================================================
Class           : PAPI
Event           : PAPI_L2_TCM (Level 2 cache misses)
Period          : 10000
Samples         : 16132
Domain          : user
Run Time        : 319.72 (seconds)
Min Self %      : (all)

Module Summary
--------------------------------------------------
Samples   Self %  Total %  Module

  16131   99.99%   99.99%  /home/nobody/solver/sol
      1    0.01%  100.00%  /lib/libc-2.2.4.so

File Summary
--------------------------------------------------
Samples   Self %  Total %  File
   5093   31.57%   31.57%  matxvec2d_blk3.f
   5015   31.09%   62.66%  cg3_blk.f
   4162   25.80%   88.46%  pc_jac2d_blk3.f
   1407    8.72%   97.18%  dot_prod2d_blk3.f
    429    2.66%   99.84%  add_exchange2d_blk3.f
     20    0.12%   99.96%  glibc-2.2.4/csu/init.c
      4    0.02%   99.99%  main3.f
      1    0.01%   99.99%  linuxthreads/weaks.c
      1    0.01%  100.00%  cs_jac2d_blk3.f

Function Summary
--------------------------------------------------
Samples   Self %  Total %  Function

   5093   31.57%   31.57%  matxvec2d_blk3
   5015   31.09%   62.66%  cg3_blk
   4162   25.80%   88.46%  pc_jac2d_blk3
   1407    8.72%   97.18%  dot_prod2d_blk3
    429    2.66%   99.84%  add_exchange2d_blk3
     20    0.12%   99.96%  ?
      4    0.02%   99.99%  main3
      1    0.01%   99.99%  __pthread_return_0
      1    0.01%  100.00%  cs_jac2d_blk3

File:Line Summary
--------------------------------------------------
Samples   Self %  Total %  File:Line

   5089   31.55%   31.55%  matxvec2d_blk3.f:19
   4125   25.57%   57.12%  pc_jac2d_blk3.f:20
   2763   17.13%   74.24%  cg3_blk.f:206
   1346    8.34%   82.59%  cg3_blk.f:346
    576    3.57%   86.16%  dot_prod2d_blk3.f:24
    524    3.25%   89.41%  cg3_blk.f:278
    489    3.03%   92.44%  dot_prod2d_blk3.f:23
    332    2.06%   94.50%  dot_prod2d_blk3.f:25
    197    1.22%   95.72%  cg3_blk.f:279
    176    1.09%   96.81%  add_exchange2d_blk3.f:29
     99    0.61%   97.42%  add_exchange2d_blk3.f:50
     71    0.44%   97.86%  add_exchange2d_blk3.f:30
     71    0.44%   98.30%  add_exchange2d_blk3.f:51
     55    0.34%   98.64%  cg3_blk.f:55
     38    0.24%   98.88%  cg3_blk.f:207
     34    0.21%   99.09%  cg3_blk.f:218
     31    0.19%   99.28%  pc_jac2d_blk3.f:27
     24    0.15%   99.43%  cg3_blk.f:139
     20    0.12%   99.55%  init.c:0
      8    0.05%   99.60%  dot_prod2d_blk3.f:22
      5    0.03%   99.63%  add_exchange2d_blk3.f:44
      4    0.02%   99.66%  matxvec2d_blk3.f:17
      4    0.02%   99.68%  cg3_blk.f:140
      3    0.02%   99.70%  cg3_blk.f:347
      3    0.02%   99.72%  cg3_blk.f:268
      3    0.02%   99.74%  cg3_blk.f:280
      3    0.02%   99.76%  pc_jac2d_blk3.f:18
      3    0.02%   99.78%  cg3_blk:/home/nobody/solver/cg3_blk.f:174
```

PerfSuite XML configuration file that could be used to measure these events. To use this configuration file with psrun, all you need to do is supply the option -c, along with the name of your custom configuration and run as usual.

The measurements described so far have been in aggregate counting mode, where the total count of one or more performance events are measured and reported over the total runtime of your application. PerfSuite provides an additional way of looking at your application's performance. Let's say you are interested in finding out where in your application all the level 2 cache misses occur so that you can focus your optimization work there. In other words, you'd like a profile similar to gprof's time-based profile, but instead have it be based on level 2 cache misses. This can be done rather easily with psrun by specifying a configuration file tailored for profiling rather than aggregate counting. The PerfSuite distribution includes a number of similar alternative configuration files that you can tailor as needed. Here's an example of how you would ask for a profiling experiment rather than the default total count of events:

```
8% psrun  -c \
/usr/local/share/perfsuite/xml/pshwpc/profile.xml \
solver

9% psprocess -e solver psrun.24135.xml
```

In profiling mode, the psprocess tool also needs the name of your executable (solver) to do its work. This is required in order to extract the symbol information in the executable so the program address can be mapped to source code lines.

Listing 5 shows an example of a profiling run of psrun obtained in this way. Not only is the application (solver) analyzed, but it also lists shared libraries used with the application that consumed CPU time. The combination of overall performance counting and profiling can be a powerful tool for learning about bottlenecks that may exist in your software and can help you to isolate quickly those areas of your application most in need of attention.

Summary

This article has touched only the surface of the techniques available to you when using hardware performance counters to measure and improve the performance of your applications. Hopefully, you now have an idea of what hardware performance counters are and how they can help you gain insight into performance bottlenecks. If you would like to get started using PerfSuite or other tools and supporting software mentioned in this article, visit the on-line Resources.

Many different ways exist in which applications can be tuned for higher performance. In fact, the most effective way is not loop-level improvements or tweaking but fundamental changes to the algorithms used in your application that are more computationally efficient. Ideally, your software will use efficient algorithms further tuned to make effective use of your CPU. PerfSuite and other similar tools can go a long way toward making this process easier for you.

Acknowledgements

I would like to thank Professor Danesh Tafti of the Mechanical Engineering Department at Virginia Tech for providing the program used for the psrun profiling example in Listing 5. This is a computational kernel extracted from a computational fluid dynamics application named GenIDLEST that Tafti and his research team use, maintain and develop. I also would like to express my thanks to all the PAPI team members of the Innovative Computing Laboratory at the University of Tennessee-Knoxville for their support and encouragement during the development of PerfSuite.

Resources for this article: www.linuxjournal.com/article/8264.

Rick Kufrin currently is a senior member of the technical staff at the University of Illinois' National Center for Supercomputing Applications. He is the originator and technical lead for the PerfSuite software project described in this article and is available for consultation on the use of PerfSuite and other technologies for software improvement.

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℠

Developing GNOME Applications with Java

Design your application's GUI look in XML, write the code in Java and plug the whole thing in to the GNOME desktop. BY MIKE PETULLO

The original announcement of the GNOME Desktop Project in 1997 stated the following intention, "to use GTK/Scheme bindings for coding small utilities and applications". Since then, the GNOME development platform has provided tools to develop using several alternatives to C. C++, Java, Perl and Python all are supported by the official GNOME distribution. In addition, the Mono Project provides tools necessary for developing GNOME applications using the C# programming language. All of these options are becoming quite popular. The GNOME interfaces for many of the system configuration tools for the Fedora Project, for example, are written in Python, and many new applications are being written in C#. This article describes how to create GNOME applications using the free Java compiler from the GNU Compiler Collection. Although this article focuses on Java, the techniques described revolve around the GLADE User Interface Builder and may be used with any of the bindings supported by the GNOME Project.

The GNU Compiler for the Java Programming Language (gcc-java) is a Java development environment distributed under the GNU General Public License. Because gcc-java is free software, it is developed independently of Sun Microsystems' Java efforts. As a result of this, gcc-java does not yet implement 100% of the Java standard. For example, support for the Abstract Window Toolkit (AWT) is not yet complete. Despite its current shortcomings, gcc-java shows great promise as the foundation of a completely free Java stack, and it already can be used to build many real-world applications; see the on-line Resources for examples.

Unlike many Java compilers, gcc-java can produce both Java bytecode and a native, platform-specific executable. In the latter case, the executable is linked against gcc-java's libgcj. libgcj is a library containing the core Java class libraries and a garbage collector. In addition, libgcj contains a bytecode interpreter so natively compiled Java applications can interact with Java bytecode libraries.

The simple Java source code in Listing 1 can be compiled into

Listing 1. HelloWorld.java

```
public class HelloWorld {
        public static void main(String[] args) {
                System.out.println("Hello, World!");
        }
}
```

Listing 2. ExampleAWT.java Fragment

```
public class ExampleAWT extends Frame {
 ExampleAWT() {
  super("AWT");

  Label msgLabel = new Label("Quit?");
  Button yesButton = new Button("Yes");
  Button noButton = new Button("No");

  Panel buttonbox = new Panel();
  buttonbox.setLayout(new FlowLayout());
  buttonbox.add(yesButton);
  buttonbox.add(noButton);

  Panel msgbox = new Panel();
  msgbox.setLayout(new FlowLayout());
  msgbox.add(msgLabel);

  add(msgbox, BorderLayout.NORTH);
  add(buttonbox, BorderLayout.SOUTH);

  yesButton.addActionListener(new ActionListener() {
   public void actionPerformed(ActionEvent e) {
    System.exit(0);
  }
  });

  noButton.addActionListener(new ActionListener() {
   public void actionPerformed(ActionEvent e) {
    System.exit(1);
   }
  });

  addWindowListener(new WindowAdapter() {
   public void windowClosing(WindowEvent e) {
     System.exit(0);
   }
  });
 }

 public static void main(String[] args) {
  ExampleAWT frame = new ExampleAWT();
  frame.pack();
  frame.setVisible(true);
 }
}
```

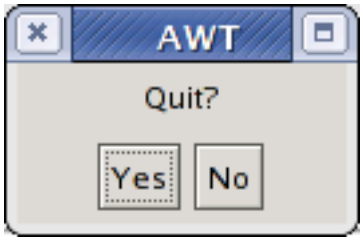


Figure 1. An AWT Application

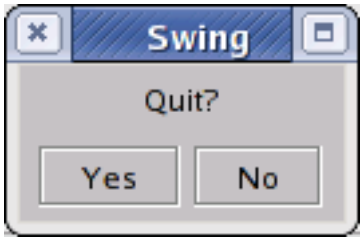


Figure 2. A Swing application—both AWT and Swing were written so that one application would behave in a similar manner on any platform.

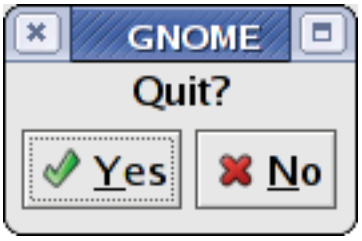


Figure 3. A Java GNOME Application

Java bytecode with `gcj -C HelloWorld.java` and interpreted using `gij HelloWorld`. The same source code can be compiled into a native executable using `gcj --main=HelloWorld -o HelloWorld HelloWorld.java` and executed using `./HelloWorld`. This article avoids including import and other trivial statements in Java code listings; see the on-line Resources for the full source files.

Sun provides two class hierarchies for developing Java applications with graphical user interfaces. The first, the Abstract Window Toolkit, has been distributed with Java since version 1.0. A picture of a gcc-java-compiled AWT application is shown in Figure 1. The corresponding source code is provided in Listing 2 and can be compiled with:

```
gcj --main=ExampleAWT -o ExampleAWT ExampleAWT.java
```

The second system, Swing, made its debut in Java 1.2. Figure 2 is a picture of the gcc-java-compiled Swing application shown in Listing 3. Listing 3 can be compiled with `gcj --main=ExampleSwing -o ExampleSwing ExampleSwing.java`. AWT uses the native GUI components in the host operating system to draw itself. Swing gives the user finer control over the look and feel of components,

Listing 3. ExampleSwing.java Fragment

```
public class ExampleSwing {
 public static void main(String[] args) {
  JFrame win = new JFrame("Swing");

  JLabel msgLabel = new JLabel("Quit?");
  JButton yesButton = new JButton("Yes");
  JButton noButton = new JButton("No");

  win.getContentPane().setLayout (new BorderLayout());

  JPanel buttonbox = new JPanel();
  buttonbox.setLayout(new FlowLayout());
  buttonbox.add(yesButton);
  buttonbox.add(noButton);

  win.getContentPane().add(msgLabel, "Center");
  win.getContentPane().add(buttonbox, "South");

  yesButton.addActionListener(new ActionListener() {
   public void actionPerformed(ActionEvent e) {
    System.exit(0);
   }
  });

  noButton.addActionListener(new ActionListener() {
   public void actionPerformed(ActionEvent e) {
    System.exit(1);
   }
  });

  win.pack();
  win.show();
 }
}
```

and most of the work is performed by Java.

IBM sponsors the Eclipse Project, an effort to produce an open-source development environment. One of the fruits of this project is the Standard Widget Toolkit, an alternative to AWT and Swing. SWT is a peer-based, operating system-independent interface that uses the host operating system's interface for rendering common components. Components not supported by an operating system are implemented in Java. On Linux, the libswt-gtk2 package provides a GTK peer for SWT. Peers also exist for other platforms, including Solaris and Windows. SWT code can run on any platform that has an SWT peer. An example SWT application is shown in Listing 4, which can be compiled against the GTK SWT peer with a variation of the following:

```
gcj --CLASSPATH=/usr/lib/libswt-gtk2.jar -lswt-gtk2
-o ExampleSWT --main=ExampleSWT ExampleSWT.java
```

See Resources for more information about the Standard Widget Toolkit.

With three existing Java GUI toolkits, one might ask why another alternative is necessary. GNOME's Java bindings are unique because they are tied directly to GNOME. An application written with GNOME's Java offerings looks and behaves exactly as if it had been written using GNOME's C libraries. It integrates seamlessly into the GNOME desktop and provides the same capabilities as any other GNOME application. The

Listing 4. ExampleSWT.java Fragment

```
public class ExampleSWT {
 public static void main(String[] args) {
  Display display = new Display();
  Shell shell = new Shell(display);
  shell.setLayout(new FillLayout(SWT.VERTICAL));

  Composite msgbox = new Composite(shell,
                                   SWT.NO_TRIM);
  RowLayout msglayout = new RowLayout();
  msglayout.justify = true;
  msgbox.setLayout(msglayout);

  Label label = new Label(msgbox, SWT.NO_TRIM);
  label.setText("Quit?");

  Composite buttonbox = new Composite(shell,
                                      SWT.NO_TRIM);
  RowLayout buttonlayout = new RowLayout();
  buttonlayout.justify = true;
  buttonlayout.pack = true;
  buttonbox.setLayout(buttonlayout);

  Button yesButton = new Button(buttonbox,
                                SWT.PUSH);
  yesButton.setText("Yes");
  Button noButton = new Button(buttonbox,
                               SWT.PUSH);
  noButton.setText("No");

  yesButton.addSelectionListener(
                              new SelectionAdapter() {
   public void widgetSelected(
                               SelectionEvent event) {
    System.exit(0);
   }
  });

  noButton.addSelectionListener(
                              new SelectionAdapter() {
   public void widgetSelected(
                               SelectionEvent event) {
    System.exit(1);
   }
  });

  shell.pack();
  shell.open();

  while (! shell.isDisposed()) {
   if (! display.readAndDispatch()) display.sleep();
  }
 }
}
```

Listing 5. ExampleGNOME.java Fragment

```
public class ExampleGNOME {
 private LibGlade libglade;
 private static final String GLADE_FILE =

"ExampleGNOME.glade";

 public ExampleGNOME () throws IOException {
  libglade = new LibGlade(GLADE_FILE, this);
 }

 public void on_noButton_released(GtkEvent event) {
  Gtk.mainQuit();
  System.exit(1);
 }

 public void on_yesButton_released(GtkEvent event) {
  Gtk.mainQuit();
  System.exit(0);
 }

 public static void main(String args[]) {
  ExampleGNOME gui;

  Gtk.init(args);
  try {
   gui = new ExampleGNOME();
  } catch (IOException e) {
   System.err.println(e);
   System.exit(1);
  }
 Gtk.main();
 }
}
```

reason for this is GNOME's Java bindings use the Java Native Interface to delegate work directly to GNOME's C libraries.

Currently, GNOME's Java bindings consist of four libraries—libgconf-java, libglade-java, libgnome-java and libgtk-java. libgtk-java and libgnome-java provide the GUI components of the bindings. libglade-java allows Java applications to read graphical user interface descriptions created by GLADE. Investigating libgconf-java, the Java interface to the GConf configuration system, is left as an exercise for the reader.

libgtk-java and libgnome-java are similar to SWT and AWT because host code implements their graphical components. However, the GNOME libraries are quite different from AWT, Swing and SWT—GNOME libraries make no claim of platform-independence. GNOME applications written in Java run only in a GNOME environment. Any platform independence is a result of the entire GNOME environment itself being platform-independent.

A gcc-java-compiled GNOME application is captured in Figure 3. Listing 5 shows the GNOME application's source code and can be compiled with:

```
gcj --CLASSPATH=/usr/share/java/gtk2.4.jar:\
/usr/share/java/gnome2.8.jar:\
/usr/share/java/glade2.8.jar \
-lgtkjar2.4 -lgnomejar2.8 -lgladejar2.8 \
-o ExampleGNOME --main=ExampleGNOME \
ExampleGNOME.java
```

At first glance, Listing 5 may look a little sparse compared to the others. ExampleGNOME's user interface is defined in ExampleGNOME.glade; as a result, there is not much GUI code in the application itself. Instead, libglade-java reads ExampleGNOME.glade and creates the application's GUI components automatically. The GUI code is tied back to our code by event callback methods. Two of these callbacks, whose names and corresponding signals are defined in ExampleGNOME.glade, are on_noButton_released and on_yesButton_released. Listing 6 contains the contents of a portion of ExampleGNOME.glade.

The GLADE system provides a User Interface Builder that makes creating definitions such as ExampleGNOME.glade simple. Figure 4 shows an example GLADE User Interface Builder session. Listing 8 contains some of the interface

Listing 6. ExampleGNOME.glade Fragment

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE glade-interface SYSTEM
 "http://glade.gnome.org/glade-2.0.dtd">

<glade-interface>
<requires lib="gnome"/>

<widget class="GtkWindow" id="ExampleGNOME">
 <property name="visible">True</property>
 <property name="title" translatable="yes">
  GNOME</property>
 <property name="type">
  GTK_WINDOW_TOPLEVEL</property>
 <property name="window_position">
  GTK_WIN_POS_NONE</property>
 <property name="modal">False</property>
 <property name="resizable">True</property>
 <property name="destroy_with_parent">
  False</property>
 <property name="decorated">True</property>
 <property name="skip_taskbar_hint">
  False</property>
 <property name="skip_pager_hint">False</property>
 <property name="type_hint">
  GDK_WINDOW_TYPE_HINT_NORMAL</property>
 <property name="gravity">
  GDK_GRAVITY_NORTH_WEST</property>

 <child>
  <widget class="GtkVBox" id="vbox1">
   <property name="visible">True</property>
   <property name="homogeneous">False</property>
   <property name="spacing">0</property>

...

    <child>
     <widget class="GtkHBox" id="hbox1">
      <property name="visible">True</property>
      <property name="homogeneous">False</property>
      <property name="spacing">0</property>

      <child>
       <widget class="GtkButton" id="yesButton">
        <property name="visible"> True</property>
        <property name="can_focus">True</property>
        <property name="label">gtk-yes</property>
        <property name="use_stock">True</property>
        <property name="relief">
         GTK_RELIEF_NORMAL</property>
        <property name="focus_on_click">
         True</property>
        <signal name="released"
         handler="on_yesButton_released"
         last_modification_time=
         "Sun, 21 Nov 2004 19:10:01 GMT"/>
       </widget>
       <packing>
        <property name="padding">0</property>
        <property name="expand">True</property>
        <property name="fill">False</property>
       </packing>
      </child>

...

     </child>
    </widget>
   </child>
</widget>

</glade-interface>
```

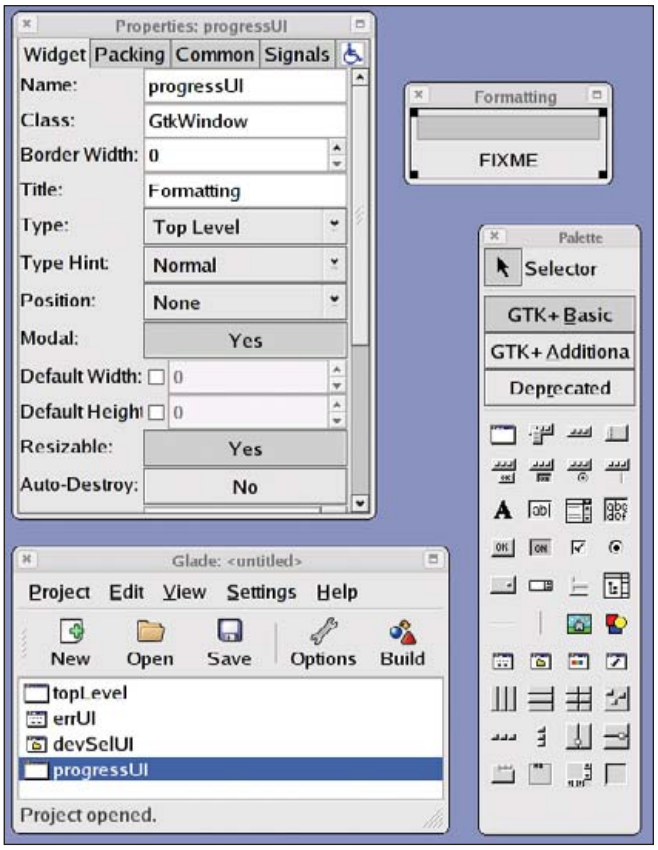


Figure 4. Designing a user interface in GLADE keeps code and layout separate.

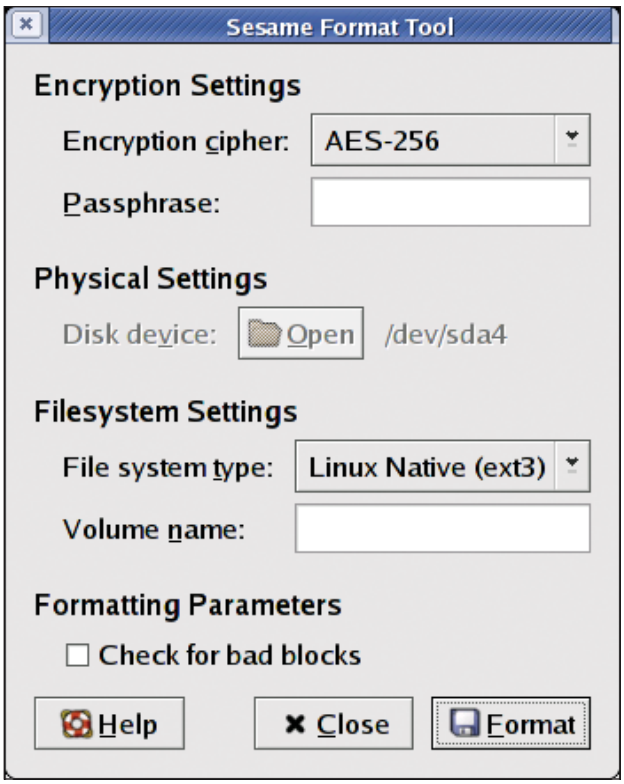


Figure 5. GnomeSesameFormat gives you an easy-to-use front end for setting up and using encrypted disk partitions.

Listing 7. GnomeSesameFormat.java Fragment

```
public class GnomeSesameFormat {

...

private void init() throws IOException {
 glade = new LibGlade(System.getProperty("GLADE_FILE"), this);

 // Default values.
 isDryRun = false;
 cipher = new
 AES256();
 fs = new Ext3();
 passphrase = null;
 volName = null;

 // References to various windows used by
 // application.
 topLevel = (Window) glade.getWidget("topLevel");
 devSelUI = (FileSelection) glade.getWidget("devSelUI");
 errUI = (Window) glade.getWidget("errUI");
 progressUI = (Window) glade.getWidget("progressUI");
}

...

public GnomeSesameFormat() throws IOException {
 init();
 device = null;

 Label l = (Label)
                  glade.getWidget("displayedDevice");
 l.setText("none selected");
}

...

public void onFormatButtonClicked(GtkEvent event) {
 Entry entry;

 entry = (Entry)
 glade.getWidget("entryPassphrase");
 passphrase = entry.getText();

 entry = (Entry) glade.getWidget("entryVolumeName");
 volName = entry.getText();

 if (topLevelInputOk ()) {
```

description being edited. Essentially, GLADE allows you to create a user interface component, name the component so it can be referenced by the corresponding program, provide method names for component signal handlers and define various properties for the component.

Designing the GUI using GLADE and allowing libglade-java to do the heavy lifting significantly reduces the work of an application developer.

```
  ProgressBar p = (ProgressBar) glade.getWidget
                                ("progressBarFormat");
  Label l = (Label) glade.getWidget("labelFormat");
  ProgressBarUpdater pU = new ProgressBarUpdater(p);
  topLevel.setSensitive(false);
  progressUI.show();

  if (! isDryRun) {
   l.setText("Formatting " + device);
   pU.start();
   execSesameFormat();
   pU.stopReq();
   try {
    pU.join();
   }
   catch (java.lang.InterruptedException e) {}
  } else {
   l.setText("[Simulated] Formatting " + device);
   pU.start();
   try {
    Thread.sleep(1000);
   } catch (java.lang.InterruptedException e) {}
  pU.stopReq();
  try {
   pU.join();
  } catch (java.lang.InterruptedException e) {}
 }

 progressUI.hide(); topLevel.setSensitive(true);
}

...

private void error(String msg) {
 Label l = (Label) glade.getWidget("labelErr");
 l.setText(msg);
 topLevel.setSensitive(false);
 errUI.show();
}

public void onErrOkButtonClicked(GtkEvent event) {
 errUI.hide();
 topLevel.setSensitive(true);
}

...
```

```
public static void main(String args[]) {
 GnomeSesameFormat gui = null;

 Gtk.init(args);

 LongOpt[] longOpt = new LongOpt[2];
 longOpt[0] = new LongOpt("help",
                     LongOpt.NO_ARGUMENT, null, 'h');
 longOpt[1] = new LongOpt("dry-run",
                     LongOpt.NO_ARGUMENT, null, 'd');

 Getopt g = new Getopt("gnome-sesame-format", args,
                       "hd", longOpt);

 int c;
 boolean optDryRun = false;
 while ((c = g.getopt()) != -1)
  switch (c) {
   case 'h':
    printUsage(0, null, null);
   case 'd':
    optDryRun = true;
    break;
   default:
    printUsage(1, null, null);
  }

 try {
  int i = g.getOptind();
  if (i == 1)
   gui = new GnomeSesameFormat(args[i]);
  else if (i > 1)
   printUsage(1, null, null);
  else
   gui = new GnomeSesameFormat();
   gui.setDryRun(optDryRun);

  Gtk.main();
 } catch (Exception e) {
  System.err.println(e);
  System.exit(1);
 }
}

...
```

Listing 7 displays some of the corresponding Java source code for GnomeSesameFormat. Listing 8 contains a portion of GnomeSesameFormat's interface definition.

GnomeSesameFormat is a simple application I developed, and most of its work is done by executing an external program called sesame-format. sesame-format formats a disk to contain an encrypted filesystem. GnomeSesameFormat simply provides a GUI wrapper for this command-line tool. GnomeSesameFormat can be executed with its --dry-run option to facilitate testing and experimenting. As of this writing, it's probably a bad idea to format a disk using this tool. A screenshot of GnomeSesameFormat is shown in Figure 5.

The GnomeSesameFormat application is implemented in a single class, GnomeSesameFormat. The GnomeSesameFormat class' main function initializes the GTK libraries using the Gtk.init method, creates a new GnomeSesameFormat instance

Listing 8. GnomeSesameFormat.glade Fragment

```
...

<widget class="GtkButton" id="buttonFormat">

...

 <signal name="clicked"
  handler="onFormatButtonClicked"
  last_modification_time=
   "Wed, 02 Feb 2005 19:16:35 GMT"/>

...

<widget class="GtkDialog" id="errUI">
  <child internal-child="vbox">
    <widget class="GtkVBox" id="vboxErr">

...

      <child internal-child="action_area">
        <widget class="GtkHButtonBox"
         id="hboxErrOk">

...

          <child>
            <widget class="GtkButton"
             id="buttonErrOk">

...

              <signal name="clicked"
               handler="onErrOkButtonClicked"
               last_modification_time=
                "Wed, 02 Feb 2005 19:22:48 GMT"/>

...

            </widget>
          </child>
        </widget>
      </child>
      <child>
        <widget class="GtkHBox" id="hboxStop">

...

          <child>
            <widget class="GtkImage" id="imageStop">

...

            </widget>
          </child>
          <child>
            <widget class="GtkLabel" id="labelErr">

...

            </widget>
          </child>
        </widget>
      </child>
    </widget>
  </child>
</widget>
```

and releases control to the GTK event loop by calling Gtk.main.

The interesting work begins in the GnomeSesameFormat class' constructor. In the constructor, a LibGlade object is instantiated. It reads a GLADE user interface description and instantiates its corresponding objects. A reference to these objects can be retrieved by name using the LibGlade object's getWidget method. Once we have a reference to an interface component, we can use them as if we created them ourselves. The GnomeSesameFormat class also contains the signal handling methods referenced in GnomeSesameFormat.glade.

In developing GnomeSesameFormat, I used the four steps presented above. For example, a button was defined using GLADE as part of the application's GUI (step 1). The button was named buttonFormat (step 2). Again using GLADE, a method name of onButtonFormatClicked was designated to handle the button's clicked symbol (step 3). Finally, the onButtonFormatClicked method was implemented in GnomeSesameFormat's Java source code (step 4).

In order to manipulate components further, libglade can provide a reference to an individual component. A LibGlade object's getWidget method provides this capability. To illustrate this, we can investigate GnomeSesameFormat's errUI component. The errUI component is a Window that displays error messages for the user. The errUI window was defined in GLADE (step 1) and named (step 2). Because we know the name of errUI, we can get a reference to it by calling getWidget(errUI). Once we receive a reference to the component, any GTK method may be invoked. GnomeSesameFormat uses errUI's show and hide methods.

The GNOME Project provides the ability to develop applications in C, C++, Java, Python and Perl. In addition, external projects such as Mono provide even more diversity. When used with several of these alternatives, the GLADE User Interface Builder makes it possible to write applications quickly with a graphical user interface for the GNOME platform. Once the graphical components are defined, an application shell and signal handlers all are that remain to be implemented. This implementation can be done using any programming language.

Resources for this article: www.linuxjournal.com/article/8274.

Mike Petullo currently is working at WMS Gaming and pursuing a Master's degree at DePaul University. He has been tinkering with Linux since 1997 and welcomes your comments sent to lj@flyn.org. Thank you to Noah Alcantara for helping to review this article.

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

Text Manipulation with sed

Save yourself tedious editing sessions with this classic tool that does changes in bulk.

BY LARRY RICHARDSON

Sed is a filter that can process text from standard input, and writes its results to standard output. The input can be redirected from a file, and the output can also be redirected to a file using your shell's redirection capabilities. sed is a pocketknife of UNIX. There are hundreds of uses for it, and you would really miss it if you lost it.

sed can append lines, remove lines, change lines, rearrange lines, substitute text strings and more. Using sed, you can write simple scripts that can become very powerful text manipulating commands.

sed can use regular expressions to define what processing will occur on the lines of text and which lines it is processing. If you have never seen or used regular expressions before, you may want to become somewhat familiar with the basic syntax of regular expressions. We use a few regular expressions to make sed do some simple text processing.

Ways to Run sed

sed can be run on the command line as follows:

```
cat sample.txt | sed -e '1,15d'
```

You can cat the file sample.txt and use the pipe to redirect its output (the lines of text) into the sed command. The -e option to sed tells it to use the next item as the sed command. The d command tells sed to delete lines 1 though 15 of the input stream, which in this case are the lines read from sample.txt. The rest of the file (if any) appears on standard output, in your terminal window, unless redirected elsewhere.

Also, you can simply specify the input file as a command-line argument, so the above sed command can also be written as:

```
sed -e '1,15d' sample.txt
```

You can also tell sed to read commands from a script file by using the [-f script-file] option.

sed Command Format

A sed command has this format:

```
[pattern1][,pattern2][!] command [args]
```

pattern1 and pattern2 are optional line ranges. Some commands don't use the patterns, some commands use only one and some can use both to specify a range of lines that the sed command can operate on, as we did in our simple example above.

pattern1 and pattern2 can be numbers, in which case they are treated like line numbers. They can also be regular expressions delimited by slashes (/pattern/). When using regular expression patterns, all lines that match the expression will be filtered through the sed command.

If no pattern is specified, then the sed command will operate on every line of input.

The ! causes sed to operate on every line not included in the pattern range. You can change our example above to be:

```
cat sample.txt | sed -e '1,15!d'
```

This command will delete all lines except lines 1 through 15.

A Few sed Commands

Here are a few basic sed examples. These can all be run right from the command line. Testing and debugging your sed commands individually on the command line before integrating them into a larger script will save you a lot of time that would otherwise be spent debugging the commands from within a running script.

Let's say that you have a file that lists customers called customer.txt. For our examples, it contains simple lines of text, like this:

```
Sam Jones
Brenda Jones
Carl Simon
Liz Smith
```

Let's use some sed commands to manipulate this file. For example, if you want to remove lines containing Carl Simon and update your customer file, you can:

```
cat customer.txt | \
sed -e '/Carl Simon/d' > customer.txt
```

The pattern /Carl Simon/ is used by sed as a regular expression, and matches every line that has that pattern somewhere on the line. The d command deletes every line that matches the pattern. So, any lines containing Carl Simon are removed from the file.

If you want to perform some type of text substitution on a text file, the s command is probably what you are looking for. It will substitute one text string for another. We tend to use this a lot in our scripts. For example, if Sam Jones calls up and tells you that you should have him listed as Samuel Jones, you can use this command to make the change:

```
cat customer.txt | \
sed -e 's/Sam Jones/Samuel Jones/' > customer.txt
```

The s command in sed has three slashes that follow the s. The text between the first and second slash is the pattern that you want to match. The text between the second and third slash contains the pattern that you will substitute for the first pattern. If you wanted all instances of Sam to be Samuel (not just Sam Jones), you could re-write this example as follows:

```
cat customer.txt | \
sed -e 's/Sam/Samuel/' > customer.txt
```

The commands for append (a), replace (c) and insert (i) typically need to have the sed commands specified in a separate script file. For example, say that you wanted to append the line After Brenda right after the line that contains the text Brenda. You can use the sed command to append the text there. However, you will need to put the sed commands in a separate script file, so fire up your favorite editor and create the following sed command file:

```
#
# sed command file (# are comment lines)
#
# append the line 'After Brenda'
# in this customer file
#
/Brenda/a\
After Brenda
```

Save this script file as sed1.cmd. Then, to run sed using this script file, use this syntax:

```
sed -f sed1.cmd customer.txt
```

You should see the contents of your customer file, with the additional line added after the line Brenda Jones. The pattern /Brenda/ (in the sed command file) determined where in the output our appended line appeared.

The difference between the append command and the insert

command is where the text is added. For the append command, the text is added after the line containing the match. For the insert command, the text is added before the line that contains the match.

Three Great Regular Expressions to Know

For those who have never used regular expressions, here are three regular expressions that you will find to be very useful when combined with sed:

1. To match the start of a line, use the ^ character.

2. To match the end of a line, use the $ character.

3. To match any number of characters in a regular expression, use the two characters .*. The . matches any single character, and the * makes it match any number of characters (including none at all).

Practical Examples

Filter out empty lines from a file:

```
sed -e '/^$/d' your_file.txt
```

Add the computer named mycomputer to the end of every line in /etc/exports.:

```
cat /etc/exports |  \
sed -e 's/$/ mycomputer/' > /etc/exports
```

Add the computer named comp2 only to the directories beginning with /data/ in /etc/exports:

```
cat /etc/exports | \
sed -e '/^\/data\//s/$/ comp2/' > /etc/exports
```

See how the forward slashes used in the directory name have to be escaped using back slashes? Without the back slashes, sed will interpret the forward slashes in the directory specifier as the delimiters in the sed command itself. However, the back slashes can make the sed command difficult to read and follow.

Remove the first word on each line (including any leading spaces and a trailing space):

```
cat test3.txt | sed -e 's/^ *[^ ]* //'
```

More regular expression matching is used in this example. Here's what it is doing.

The initial ^ * is used to match any number of spaces at the beginning of the line. The [^]* then matches any number of characters that are not spaces (note that the ^ inside the brace reverses the match on the space), so it matches a single word. The trailing space at the end will match the space found at the end of the first word. The empty replace pattern removes the text.

Remove the last word on each line:

```
cat test3.txt | sed -e 's/^\(.*\) .*/\1/'
```

This command introduces the concept of hold buffers. Hold buffers are used to keep parts of the matched text and to insert that text into the result. The pattern that matches the text between the parentheses is recalled in the substitution pattern by the \1. If an additional set of parentheses were in the match pattern, they would be addressed in the substitution pattern as \2, and so on for more sets of parentheses. Up to nine hold buffers can be specified. In this example, the pattern contained within the parentheses matches from the start of the line up to the last space (the space after the parentheses) in the line.

To remove leading { and trailing }, or } from each line:

```
sed -e 's/^.*{\(.*\)},*/\1/' table.txt
```

I'll leave it to the reader to dig into this regular expression to see how it operates. Keep this in mind—the more comfortable you get with regular expressions and hold buffers, the more powerful the sed command will become.

Conclusion

There are many other commands that sed recognizes. However, even with these basic commands, you can successfully manipulate text files from within your own shell scripts or right from the command line.

Larry Richardson develops meteorological workstation software for 3SI. He has developed software for UNIX and Windows using C and C++ for more than 13 years. Now living in Georgia with his wife and son, he enjoys playing bass in his spare time.

Designing and Implementing a Domain-Specific Language

"Like everything metaphysical the harmony between thought and reality is to be found in the grammar of the language."
—Ludwig Wittgenstein **BY RYAN PAUL**

In *Star Trek V: The Final Frontier*, Scotty reminds a junior engineer to use "the right tool, for the right job". This sage advice is applicable to computer users as well as starship engineers. GNU/Linux users have a particularly impressive assortment of tools at their disposal, many of which feature unique syntaxes that facilitate concise expression of complex operations.

Good tools will reflect the specific needs of the individual problems they are designed to solve. Consider the highly effective text processing utilities awk and sed. With simple commands, users can perform efficient search-and-replace operations on streams or filter complex data. How much C code would it take to do the same things? Even with a concise general-purpose language like Python, the tasks still will require more typing than the equivalent command-line tools do. Utilities like awk and sed are effective because they interface well with other command-line utilities, and they leverage the power of domain-specific languages (DSLs), syntaxes specialized for a particular group of related tasks.

Despite the vast number of powerful applications developed for GNU/Linux, the right tool isn't always available for any given job. What should a resourceful user do when options are limited? In most cases, it is possible to combine existing tools, possibly creating a new tool in the process. Sometimes, a new tool needs to be made from scratch with a general-purpose programming language. Developers can add value to a new tool and increase its productive potential by implementing a domain-specific language for it.

Development time is an investment, and many programmers endeavor to maximize the return on that investment by writing reusable code libraries. Tools developed with a specialized code library generally expose only a limited subset of the library's features. Developers can provide more extensive access to library functionality by constructing a domain-specific language that can act as an interface. A well-built DSL allows users to employ an intuitive and self-documenting syntax to construct a multitude of highly specialized tools rapidly.

Pitfalls

Implementation of a DSL can be tricky business. Code that parses and validates specialized syntax is difficult to write and maintain, especially if the DSL supports sophisticated control structures. Tools written with DSLs are notoriously difficult to debug, and there will be no IDE available for your new language unless you make one.

One of the most compelling arguments against corporate use of DSLs is the so-called Tower of Babel effect. When a number of developers all construct their own individual DSLs, the sheer number of disparate syntaxes can create a tremendous amount of confusion.

When developers perpetually increase the scope of a DSL's target domain, they risk under-specialization. When the target domain grows to an unmanageable proportion, the DSL will transmogrify into a personal Perl implementation, and it will cease to fulfill the needs adequately of the individual tasks associated with the actual domain.

Implementation

Meta-programming is the art of writing code that generates or manipulates code. It is the basis for language implementation, and there are many ways to do it. Meta-programming is either static or dynamic, depending on the type system of the implementation language. Static meta-programming typically is done with a preprocessor, and dynamic meta-programming typically is done with macros that are evaluated at runtime.

A number of excellent open-source language development platforms are available for GNU/Linux. One of the most impressive static meta-programming utilities is Camlp4, an extensible preprocessor for Inria's Ocaml programming language. Camlp4 facilitates rapid development of efficient, type-safe DSLs. Of the available dynamic meta-programming platforms, the best is Logix, an extremely versatile language design system implemented for and with Python.

Looking at Logix

LiveLogix is a consulting and development firm with big plans and innovative ideas. Logix, available under the GPL, is their first major release and the vanguard of their LiveLogix Application Platform, an assortment of versatile and dynamic development tools currently in the early stages of development. Inspired by the dynamism of Python, the syntactic grace of Haskell and the mutability of Lisp, Logix is a unique fusion of features and flexibility.

Logix developers do not build complete formal grammars, they incrementally define the individual operators that make up a language. It is then possible to combine these operators to form expressions, which the Logix processor can parse and convert into Python bytecode. Logix DSLs optionally can leverage powerful Python language features like control structures, object orientation

and list processing. Seamless Python integration and access to the tremendous number of useful libraries and modules available to Python further increase the power and value of Logix.

Logix developers build their programs with either the standard or base Logix dialects. The syntax of the base dialect is like normal Python syntax with a few additional features for language extension. The standard dialect includes a wide variety of excellent syntactic enhancements and unique features.

Experienced Python developers quickly adapt to standard Logix idioms. The documentation contains an excellent introduction for Python programmers that fully explores the syntactic divergences. Many of the substantial differences relate to Logix's special treatment of expressions. All statements return values, so it is possible to write code like this:

```
x = if 10 * 2 == 20: "yes it is!" else: "no"
```

A function call is written as a series of expressions:

```
min 2 6
```

In the standard dialect, parentheses distinguish individual expressions just as they do in algebra. Parentheses are not a part of the actual call nomenclature. The standard Logix expression:

```
min 2 6 (min 10 15)
```

is the same as the Python expression:

```
min(2, 6, min(10, 15))
```

Functions that do not require arguments are the exception. They are still called with trailing parentheses just as they are in Python:

```
function()
```

Language extensions are written with defop statements. It is possible to add new prefix, postfix and infix operators as well as special mixfix operators like C's conditional expressions. It also is possible to add new keywords.

Operator definitions consist of an associativity specification, a binding value, the operator syntax and the implementation. Associativity is specified with a single letter, either l for left or r for right. If associativity is not specified, Logix automatically makes the new operator left associative. The binding value specifies operator precedence. The binding value syntax is one of the only things I really don't like about Logix. Even in languages with static syntax, operator precedence tends to confuse me. It's much more difficult to keep track of in a structurally dynamic language. Fortunately, precedence and associativity won't be all that important in simple tool languages.

The operator syntax consists of variables and constants. Constants are enclosed in quotations, and variables are specified by type: expr (expression), symbol, term, token, block and freetext. The operator implementation can be either a macro or a function. A function is evaluated at runtime, whereas macros perform code replacement at compile time.

Let's have a look at an example from the Logix documentation:

```
defop 50 expr "isa" expr func ob typ:
  isinstance ob typ
```

This describes an isa operator. The new isa operator consists of an expression, followed by the constant isa, followed by an expression. The func keyword indicates that the implementation is a function, and the two symbols that follow are the names of the variables. Each variable in the implementation is associated with a variable in the syntax definition. In this case, the first expr is ob, and the second expr is typ. The code within the func block is evaluated when the operator is called.

In the following line of code:

```
"test" isa str
```

the string test is the first expression passed to the isa operator, and the type str is the second expression. The operator then passes the arguments to the isinstance function and returns the result, which in this case, is the boolean True.

A Language Is Born

Now that we have worked through the basics, let's try a more complex example. Imagine a company with a veritable fleet of network-enabled printers featuring telnet-accessible administrative interfaces. This hypothetical company maintains a record of the current configuration for all its printers in a text file. When people want to change the configuration of a particular printer, they record the change in the text document, and then they connect to the printer and make the change. The company could design a simple DSL that treats the configuration record as a program. So, when someone wants to change the configuration of a printer, this user simply could change the document and run it. When run, the text document would connect to all the printers and repopulate the configuration data.

First, let's have a look at the document:

```
default:
  syslog_facility:local3
  idle_timeout:120
  old_idle_mode:off

accounting printers:
  - 10 hp5mo1
      syslog_facility:local2
  - 28 lpt9
  - 29 lpt10
  - 48 lpt6

developer printers:
  - 26 lpt4
  - 27 lpt7

marketing printers:
  - 62 hpcolor5:
      old_idle_mode:on
  - 154 lpt11
```

```
for department in
  [accounting, developer, marketing]:
  for printer in department:
    print ("Configuring %s..."%printer.host)
    printer.transmit()

print "Finished!"
```

When you design your own DSL, you must consider the implications of the syntax you select. If you want to add more features, will you be able to? Inexperienced DSL developers monopolize common meta-characters in order to make the syntax as concise as possible. In the long run, that makes it harder to learn, harder to use and harder to extend.

The default block contains the default configuration options that will be set on all printers. Each of the printers blocks contains a description of all the printers in a single department. Each individual printer definition contains the end of the printer's IP address and the associated hostname. A printer definition optionally can be followed by a block that contains configuration options specific to that printer. Our DSL turns each printer block into a list of Printer objects and assigns that list to a variable bearing the name of the department. It then will be possible to manipulate these lists with code written in the standard Logix dialect.

Now, let's have a look at the implementation:

```
setlang logix.stdlang

from telnetlib import Telnet

class TelnetDebug:
  def write self txt: print "dbg:%s"%txt

class Printer:
  def __init__ self ip host data:
    self.ip = ip
    self.host = host

    self.data = Printer.default.copy()
    self.data.update data

  def transmit self:
    #tn = Telnet "192.168.0.%s"%self.ip
    tn = TelnetDebug()

    tn.write "printer_password"
    tn.write ("host %s"%self.host)

    for x,y in self.data.items():
      tn.write ("%s %s"%(x,y))

deflang printerdef:

  defop 50 expr ":" expr macro n v:
    str n, str v

  defop 0 "-" token expr [":" block]/-
    macro ip v *b:
      ["host":str v, "ip":str ip, "block":b]
```

```
deflang printlang(logix.stdlang):

  defop 0 expr "printers:" block@printerdef
    macro n *v:
      `\n = [\@.Printer p/ip p/host (dict p/block)
         for p in \v]

  defop 0 "default:" block@printerdef macro *b:
    `\@.Printer.default = dict \b
```

The implementation starts with a setlang directive that tells the interpreter to use the standard Logix dialect. Next, we define the Printer class. Every printer defined in a printers block eventually becomes an instance of the Printer class. The Printer class contains no code specific to the DSL and easily can be used in another project. The Printer initialization method takes three arguments: the last part of the printer IP address, the printer hostname and a dict that associates option names with option values. The init method also copies the default printer options from a class variable into an instance variable called data and updates it with the printer-specific options passed into the instance via the data argument.

Now we get to the good part, the language definition. In Logix, the deflang statement is used to start a new language block. Each language block contains a sequence of operator definitions. The first language block describes the syntax we will use in the individual printers blocks and the default block. The printerdef language's first operator is the colon, an infix operator that is used to parse individual options. The first expr is the option name, and the second expr is the option value. The colon operator implementation is a macro that converts the expressions into strings and puts them in a tuple.

The second operator in the printerdef language is the hyphen operator, a mixfix operator that is used to define individual printers. This one is a bit more complicated. The operator starts with a literal hyphen, which is followed by a variable token, an expression and an optional block. A token is a single value, in this case a number. A block, as one might guess, is a block of content that is parsed using Python's indentation rules.

In the definition, the literal colon and the block are enclosed in braces and followed by a /-. The braces group syntactic elements, and the /- following the group indicates that it is optional. This makes it possible to omit the block for printers that don't need to specify their own configuration options. The implementation is a macro that takes three arguments. The token is the IP address suffix, the expr is the printer hostname and the block contains the printer options. The asterisk in front of the b indicates that the variable is a sequence. If you don't specify that the block variable is a sequence, blocks with more than one line will not be parsable. The implementation returns a dict containing the hostname, the IP suffix and the block. The block contains options, which get transformed into tuples, so in the implementation, the b variable is a sequence of tuples.

The second language in the implementation contains the

primary syntax for our DSL. After the language name, you can see a reference to the standard Logix dialect enclosed in parentheses. Like classes, Logix languages support inheritance. The stdlang reference within the parentheses indicates that our printlang inherits all the operators of stdlang. Developers now can use standard Logix syntax in addition to the specialized operators defined within the printlang. That is how the for loop at the end of the printer configuration program is possible.

The printers operator starts with an expression, followed by the literal `printers:` and then a block. In this definition, the block is immediately followed by `@printerdef`, which tells the interpreter that the contents of the block should be parsed by the printerdef language. The printers implementation is a macro with two operators: the name of the group and the block, which is a sequence of dicts that contain printer definitions.

The back-tick at the beginning of the implementation macro replaces the escaped variables with their values and converts the expression into code data. We want to be able to make a variable that uses a name provided by the user. For instance, we want to assign the value of the first printers block to the variable accounting. If the implementation wasn't quoted, it would try to assign the value to the variable n, rather than creating a new variable that uses the name provided by the value. Quoting is like Python's exec function:

```
n = 'test'
```

is like unquoted content, whereas:

```
exec("%s = 'test'"%n)
```

is like quoted content.

In Logix, the forward slash represents an escaped variable. Escaped variables are replaced with their values the same way that %s is replaced with the value of n in the sample exec expression. The escaped @ represents the current module, so `\@.Printer` is a reference to the Printer class. The list comprehension builds a Printer instance for each printer definition. Logix provides special syntax for dictionary access:

```
some_dict/key
```

is transformed into:

```
some_dict["key"]
```

So, the interpreter acquires the IP, the host and the option block from the Printer definition dict and passes them as arguments to the Printer constructor.

The default operator takes its block and assigns it to the default Printer class variable.

That's all there is to it. Now you can build the right tool for any job! With a good language development platform at your command, the only limitation is your imagination.

What the Future Holds

Did this tantalizing taste of Logix intrigue you? I asked Logix creator Tom Locke to shed some light on the Logix future. We soon can expect to see a faster, more effective Logix. The next release will feature an efficient new parser, written entirely in C. Eventually, Tom plans to port Logix to a more suitable language platform like Mono. He wants a versatile runtime engine that emphasizes security and offers a wide variety of featureful libraries.

Logix is currently available under the GPL. Future releases also will offer a less-restrictive license that will enable developers to distribute original and modified works in both source and binary form.

Resources for this article: www.linuxjournal.com/article/8209.

Ryan Paul is a system administrator, a freelance writer and an ardent proponent of open-source technology. He welcomes your questions and comments. Ryan can be contacted at segphault@sbcglobal.net.

Using an iPod in Linux

What can you do with an Apple iPod and Linux that you can't do with Apple's iTunes? Plenty.

BY BERT HAYES

The market for portable MP3 players has exploded in the last few years, and Apple's iPod is considered by many to be the gold standard to which other players are compared. Despite the fact that Apple does not offer a flavor of its iTunes music application for Linux, the iPod still is a good MP3 player for Linux users everywhere.

As I write this, I've used a 20GB fourth-generation iPod with click-wheel and the iPod Shuffle; the software described in this article, however, should work with all iPods. To use the slick GUI of GTKPod, you will have better luck with a newer Linux distro. If you're a fan of the command line, though, you should have no problems getting GNUpod to work on almost anything that runs Perl.

With GTKPod, the open-source answer to iTunes, you can rock and roll all night with your iPod and Linux. GTKPod is the slick GUI that allows you to transfer MP3, WAV and M4A non-DRM-infected AAC files from your PC to your iPod. You can add files one at a time or you can add an entire directory at one time. You can create and edit playlists, normalize the volume on single tracks or multiple tracks at a time. You also can edit the ID3 tag of files on the iPod or files on your computer. GTKPod is even smart enough to know not to upload files that already exist on your iPod. And, naturally, GTKPod lets you delete songs from your iPod should you start running out of space.

GTKPod is built on GTK2. Originally developed for use with The GIMP, GTK now is used widely in many applications and is the set of tools used to build the GNOME desktop. As sure as GTK is slick software, it also is complex software with a raft of dependencies, interdependencies and various requirements so byzantine that compiling the stuff yourself from source code is asking for a trip down the rabbit hole. Lucky for us all, everything you need to use GTK applications for such as GTKPod should be on your computer already if you're running a recent distribution of Linux.

If you're already using your iPod, you probably have everything you need to use it in Linux: the iPod itself, a PC with FireWire or USB 2.0 support and a relatively recent Linux distribution. You'll also need a copy of GTKPod (see the on-line Resources). On the GTKPod site, you should find a lot of documentation and links to the source code, as well as GTKPod packages for SUSE, Debian, Gentoo, Mandrake and other popular Linux distros. More RPM packages can be found by searching rpm.pbone.net or rpmfind.net.

Because the iPod simply is a fancy removable hard drive, to use it with GTKPod in Linux, your kernel needs to support fancy removable hard drives. This kind of support has been in the kernel since 2.4, and it has been refined in the current 2.6

FAT32 ON A NEW IPOD

If your iPod is just out of the box, you probably have to configure it before you can use it with Linux. Configure sounds nice and easy, doesn't it? In truth, you are going to reformat it. Scared? Don't be.

Most new iPods are shipped with their hard drives formatted with the Mac OS filesystem HFS+. To use an iPod that just came out of its box with Linux, your kernel needs to have HFS+ support as well as support for Mac-style partitions. Most modern distros do not include this support automatically and require a kernel recompile to enable it. Also, it's been reported that the HFS+ kernel support still is a bit buggy. If you're not in the mood to recompile your kernel simply to use your new toy, there is an easier way.

Because most versions of Microsoft Windows turn up their noses at reading/writing to the Mac filesystem, the first thing the iTunes software does when installing on a Windows system is configure the iPod. iTunes says configure but what it's really doing is reformatting the iPod's internal hard drive to the FAT32 Windows filesystem. So if you've just pulled your iPod out of its box, and you've got a Windows machine handy, you can use the iTunes software install process to format your iPod. The iPod Shuffle comes from the factory preformatted with the FAT32 filesystem, so you should be able to use it right out of the box.

Reformatting your iPod also can be done using the Restore iPod feature in the Windows iPod software. But be warned that this step wipes the iPod completely, causing you to lose all songs and files you had. If you've been using your iPod on a Mac and now want to use it with Linux without reformatting it, check out the on-line Resources for links to tips from other Mac users who have done exactly this. If you refuse to use Windows or Mac OS or if you don't have a Windows machine handy, the Resources also contains links on how to reformat the iPod using only Linux. It is possible, but it does require HFS+ and Mac partition support in your kernel and the use of the GNUpod tools to lay down the basic directory structure on the iPod's cleaned slate.

kernel. Almost all current distributions use the 2.6 kernel, and removable hard drive support usually is compiled in the kernel or is available as loadable kernel modules. Therefore, you shouldn't need to do anything to your system to make it see the iPod.

You can connect the iPod to your Linux box through a FireWire connection or a USB 2.0 connection. USB 1.1 also works, but we don't recommend it because the connection is so much slower. Another drawback of using USB 1.1 is the iPod can't charge while it's plugged in to your PC.

Installing GTKPod should be no problem whether you're compiling the source code or installing a precompiled binary package. GTKPod requires the libid3tag library, and the libmp4v2 package is required if you plan on using AAC files (see Resources). The source code for GTKPod compiles easily with the standard:

```
./configure
make
make install
```

commands. A precompiled binary for your platform should install even easier than that and provides tighter integration with your existing desktop.

Once the iPod is formatted with FAT32 and GTKPod is installed, plug the supplied FireWire or USB cable in to your PC, and plug the other end in to the iPod. If Linux sees the iPod, you should see the iPod's screen flash the message "Do Not Disconnect"; on the iPod Shuffle, the status light blinks orange. If your iPod doesn't do this, try unplugging it and plugging it back in. I'm using FireWire instead of USB on my 20GB iPod, because the majority of on-line resources provides directions for using it. Also, if you have only the one FireWire device active, it makes things a little easier when disconnecting the iPod. The iPod Shuffle connects as easily and predictably as a USB thumbdrive. See the Troubleshooting section of this article if you have problems getting Linux to see your iPod.

If you're using a disk-based iPod that's already FAT32-formatted (see sidebar), there are two partitions on the drive. One is about 40MB and holds the iPod's firmware operating system, and the second is a huge partition that holds all of your music. If you're using an iPod Shuffle, you have only one partition, and it's already FAT32-formatted from the factory.

Every computer is going to be slightly different, so I recommend tailing your /var/log/messages file, or the analogous file for your Linux distribution, to watch what's happening behind the scenes when you first connect your iPod. The log file should indicate that your system has recognized the iPod and assigned it the next available SCSI drive letter, /dev/sda if you're running nothing but IDE drives. Some systems use the fstab-sync utility to edit automatically the /etc/fstab file to include a mount point for the iPod. Others require that a specific entry in your /etc/fstab file already exists. If you're tailing the /var/log/messages file and you don't see something like this fly by:

```
fstab-sync[4284]: added mount point /mnt/ipod for
/dev/sda2
```

then it's a good idea to add a line manually to your /etc/fstab

file that indicates this:

```
/dev/sda2 /mnt/ipod vfat rw,user 0 0
```

Remember that for the disk-based iPods, you want to mount only the large music-holding partition (sda2) and not the smaller partition that holds the OS (sda1). For the iPod Shuffle, which has one partition, an entry like this should do the trick:

```
/dev/sda1 /mnt/ipod vfat rw,user 0 0
```

If you haven't done so already, make the mount point /mnt/ipod by running `mkdir /mnt/ipod` as root.

By default, GTKPod looks for your iPod to be mounted at /mnt/ipod, but you can change this value. The easiest way to mount your iPod is to let GTKPod do it for you. To do this, start up GTKPod and select Edit then Edit Preferences. In the Input/Output tab, check the box that says Handle mounting/umounting of iPod drive. Start up GTKPod again, and your iPod automatically is mounted for you.

GTKPod has many options and features, and the best way to get your feet wet is to put some music on the iPod. Chances are, you've already got buckets of MP3 files on your computer. GTKPod handles managing files only on the iPod. To create MP3s, you need to use a separate program, such as Grip. To listen to the MP3s, GKTPod relies on an external program such as XMMS; you can specify your favorite MP3 player in GTKPod's preferences.

Once you've got some MP3s to transfer, you can use the Add Files button to add single files or the Add Dirs button to add an entire directory at one time. Figure 1 shows the Beastie Boys' album *Hello Nasty* being added to my freshly formatted iPod. When you click the OK button to add a directory, GTKPod then processes the files, adds them to your local iTunes database (in ~/.gtkpod/ by default) and lists them by artist, title, album or genre in the lower windowpane. The next step in the process is to click the Sync button. This step does the actual transfer of files from your PC to the iPod; it also syncs your local iTunes database on the PC to the one on your iPod. It takes a minute or two to accomplish, depending on how much music you're moving over. When your files have been moved over successfully, GTKPod says "iPod Database Saved" in the lower left-hand corner.

Don't remove that iPod yet, though. If the iPod still says "Do Not Disconnect" or if your Shuffle's status light still is blinking orange, then leave it connected. Consider that your iPod has been mounted as a removable hard drive. You wouldn't unplug a spinning hard drive from your system, would you? You first need to make sure the iPod is unmounted. If you're using the automatic mounting within GTKPod, simply exit GTKPod and the iPod is unmounted.

At this point, your iPod still should be flashing the "Do Not Disconnect" warning. It's safe to unplug your iPod only when this message is gone. The way to remove this message is to unload the kernel module that handles removable hard drives. If you're using FireWire, the module is sbp2, which can be removed by giving the command `modprobe -r sbp2` in a root terminal. Only when the "Do Not Disconnect" message is gone and you see the normal iPod menu should you disconnect it from your PC.

Figure 1. GTKPod organizes your music by all the usual attributes and lets you create playlists.

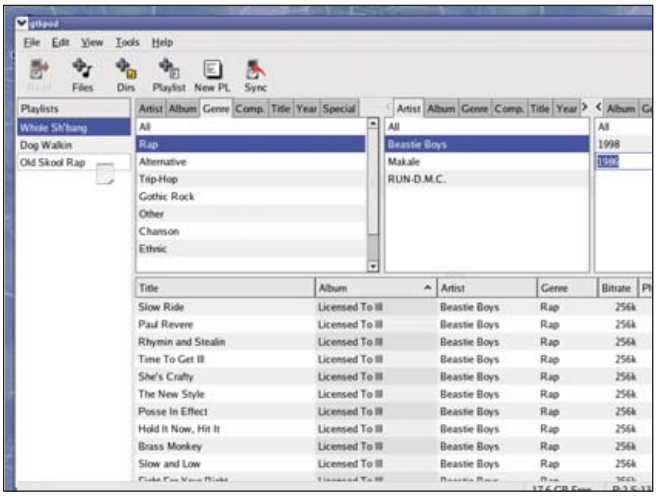


Figure 2. Building a playlist is as easy as drag and drop.

With GTKPod, you have the ability to execute scripts automatically every time you start or stop the program. The files ~/.gtkpod/gtkpod.in and ~/.gtkpod/gtkpod.out—or /etc/gtkpod.in and /etc/gtkpod.out if these files are not in your home directory—are read and run when starting up or exiting GTKPod. In the case of startup, the script is run before your iPod is mounted. So, if you need to load kernel modules or otherwise massage your system before using your iPod in Linux, this is the place to do it. Likewise, putting scripts in a gtkpod.out file can make removing your iPod a snap.

Once your iPod is removed, you should be able to play all of the tracks you transferred. If things didn't work out as planned, check the troubleshooting section of this article or the on-line Resources for help. If things did work out, you're going to want to add some more music. And once you've added some more music, you'll want to explore some of GTKPod's features for managing your tunes.

To manage the files on your iPod, use the Read button in GTKPod to read the contents of the iTunes database on your

iPod. You now should see your recently added tunes. The far left pane shows which playlist is selected. A playlist is exactly what it sounds like it is—a list of songs grouped to be played together. The tabbed windowpanes show the music on your iPod as listed in its iTunes database and displayed according to the selected tabs. By default, there are two panes, but you can edit your preferences to add more. I've added one pane for a total of three so I can sort through my music with a finer-toothed comb.

Creating playlists gives you the ability to be your own DJ and play the mixes you want when you want to hear them. I don't know about you, but nothing brightens my day quite like that old-school East Coast rap. So I'm going to create a playlist for those dark days in the office when I need more of a pick-me-up than a cup of coffee can provide. I click on New PL to create my Old Skool Rap playlist and then use GTKPod to sort through my music to select the proper songs (Figure 2). With the additional sort windowpane, I can sort by genre, artist and then year. I want to hear the Beastie Boys, but only the old stuff, so I click on 1986 and drag and drop it onto my new playlist, and 13 files are copied. Now I move down to RUN-DMC and do the same. By now, I realize that I don't have enough rap ripped, so I need to fire up Grip and get to work.

If you already are using another application to generate playlists, you still can use those playlist files in GTKPod. GTKPod should have no trouble adding preexisting .m3u or .pls files as playlists. Simply click New PL, name your playlist and then click Add File and find your playlist file to add. GTKPod also lets you export an existing playlist to m3u format.

One of my favorite features of GTKPod is the ability to edit the ID3 tag of MP3s on the iPod as well as those on your PC. The ID3 tag is the portion of the MP3 file that contains metadata such as the artist's name, album name, song name and the year the album was released. You can name an MP3 file absolutely anything in the world, and GTKPod and your iPod still will list it based on the ID3 tag. On the other hand, if your MP3 file lacks the ID3 tag for some reason, it is not listed correctly and shows up under a blank heading in GTKPod or under the All heading on your iPod. To fix this, simply click on the section you want to edit in GTKPod and type away.

If you click on an individual song in the lower windowpane, you edit the ID3 tag for that song only. If you want to edit the same field for an entire group of songs, click and edit that field in one of the sort windowpanes and the changes are reflected for all songs in the bottom windowpane. You also can use the Multi-Edit function to edit ID3 tags for several files at once. This feature is optional and must be enabled within your preferences, but it allows you to select several songs at once—highlighting them using the Shift key as in Windows—and have the edit for one field, for example, artist, apply to every selected song.

Another great feature of GTKPod that you won't find in Apple's iTunes software is the ability to export songs, copying them from the iPod back to your computer. Of course, this could be done simply by mounting the iPod as an external hard drive and rooting around until you find the exact songs to copy off, but it's much easier to use GTKPod to sort and select the songs to copy. Look under the File menu, then Copy Tracks from iPod. The Delete Completely From iPod option under the Edit menu works as advertised, freeing up precious megabytes from your iPod should you need more space for this week's favorite flavor of music.

One feature notably lacking from GTKPod is the ability to manage music purchased from the iTunes music store. This iTMS music is compressed using the AAC format and then laced with a bit of DRM technology to limit what you can do with it. If you've purchased a lot of music from iTunes and you want to manage it with GTKPod, you've got two options. The first is to use iTunes to burn the music to a CD, then re-rip it using a tool such as Grip. This works, but unless you burn and rip an entire album at a time, the CDDB database doesn't know what to make of your freshly burned CD, and you wind up adding all of the ID3 tags manually. The second option is to use a program such as Hymn to strip all of the DRM ugliness from the music that you purchased. Be warned that because Hymn circumvents DRM, it may not be legal to use if vendor lock-in laws such as the US' Digital Millennium Copyright Act exist in your country.

If you're not much for GUIs and you prefer the simple elegance of a shell prompt, consider GNUpod, a collection of Perl scripts that makes managing the music on your iPod easy. GNUpod's tools do everything from creating the directory

IPOD + LINUX = IPODLINUX

The iPodLinux Project takes Linux and puts it on your iPod. It is a port of the uClinux kernel that has been customized to run on the iPod hardware, and it includes Podzilla as a GUI OS. iPodLinux boasts the ability to dual-boot between Podzilla and the regular iPod OS, so if you feel like trying something completely different, give it a shot.

structure that holds music on the iPod, to adding or deleting music and managing playlists. It does all this from the command line by passing arguments to various Perl scripts, such as this one:

```
gnupod_addsong.pl -m /mnt/ipod /tunes/rappers_delight.mp3
```

GNUpod installs in a snap. The on-line documentation provided by the GNU folks is comprehensive and walks you through everything from getting FireWire working to re-formatting your iPod. At the time of this writing, GNUpod supports all flavors of iPods including the Shuffle. The current version of GTKPod (0.87) does not support the Shuffle, but chances are it will by the time you read this.

Troubleshooting

If you do run into trouble, relax. I had a little trouble myself getting everything to run smoothly. Be warned that some of the drivers still are a little flaky. The modules used when running the iPod over FireWire are sbp2 and ohci1394. One of my test systems was a Red Hat Fedora Core 3 system, running the 2.6.10-1.741_FC3 kernel downloaded and installed with up2date. Note: the stock kernel with this distro has a known bug that makes using your iPod with USB a nightmare. Most of the time, there were no problems, although the sbp2 module would hang when I tried to unload it by running `modprobe -r sbp2` as root. Once, I had to remove the ohci1394 and reload the ohci1394 driver for my system to see the iPod. Of course, your mileage may vary, based on your kernel version or the chipset of the FireWire card in your system.

If no amount of plugging/unplugging the iPod or loading/unloading the modules gets Linux to see the iPod, don't panic. This happened to me, and the one sure-fire way I found of getting my system to see the iPod was to boot the machine with the iPod plugged in. It's not exactly elegant, but it is effective. Watch your distribution's release notes, and consider helping to diagnose any bugs you experience.

Resources for this article: www.linuxjournal.com/article/8210.

Bert Hayes has been a Linux user and administrator since the dark days of the 2.0 kernel. He is an RHCE and a co-author of *Snort for Dummies*. His hobbies include cycling and restoring an air-cooled VW bus.

User-Modifiable Politics

This information freedom group helped elect five of the six candidates it supported. Here's how you can help get Congress interested in users' media rights. **BY REN BUCHOLZ**

Balance is the foundation of American copyright and patent law, but you wouldn't know from looking at its current trajectory. For the last 100 years, protections for rights-holders have expanded steadily, usually at the expense of users' rights. Laws such as the Digital Millennium Copyright Act (DMCA) allow programmers to be arrested for writing the wrong code. Other countries have issued travel advisories to their programmers—and half-serious invitations to ours—because American copyright and patent laws have made coding hazardous. Even worse, the US actively is exporting these policies by way of free trade agreements and pressuring countries such as Brazil that embrace free software. How did our information policy become so one-sided? More importantly, what can we do to fix it?

Part of the answer lies in connecting the "free culture" movement—including free/open-source software advocates and copyright reformers—to traditional politics. In an effort to establish those links, David Alpert and I founded IPac, a nonpartisan political action committee that promotes the public's interest regarding information policy issues. We all are volunteers, and we have taken a broad, transparent approach to politics, relying on the "bundling" of many small donations and blogging our interactions with candidates. In the last federal election, five of IPac's six candidates won their races.

IPac is necessary because traditional programmers' rights groups cannot deal directly with the electoral process. Existing organizations already take this community's message to the US Congress in the form of lobbying and activism, but there still is a huge gap in two important areas: money and votes. Tax-exempt charities such as the Electronic Frontier Foundation (disclaimer: I work there too), Public Knowledge and the Free Software Foundation do great work, but they are barred by campaign finance laws from participating in elections. Because IPac is not tax-exempt, we can buy advertising, make campaign contributions and mobilize voters to influence elections directly.

This kind of engagement is vital, because currently the public interest simply does not figure into the legislative calculus of setting information policy. In fact, most of the last decade's copyright and patent legislation, including the DMCA, was considered so noncontroversial that it was passed by voice vote, a procedure reserved for bills without real opposition.

Meanwhile, industries that want stronger restrictions have been supporting legislators generously. According to the Center for Responsive Politics, the television, movie and music industries gave almost $19 million to legislators during the last fiscal year. That's a lot of cash, but we don't have to match it in order to make an impact. The conventional wisdom in Washington, DC, is that a candidate's stance on copyright reform has no bearing on how many votes he or she receives on election day. Most congresspeople don't think that these issues are contentious, so they don't think about them at all. IPac intends to change this by associating political costs and benefits with taking a strong stance on information policy. In many instances, a handful of sympathetic legislators can derail a harmful bill, so even a relatively small number of allies would constitute a huge change in the policy landscape.

It is also worth mentioning that Hollywood's $19 million is spread thin. On average, the representatives supported by the content industry received only $15,000 during the last election. If your area's open-source users' groups passed the hat at their meetings, we almost could start "Bearded Sysadmins for Truth" and tackle a local campaign head-on. If a small fraction of those meeting attendees visited their representatives in person to talk about the importance of free software and copyright balance, Hollywood would start to get nervous.

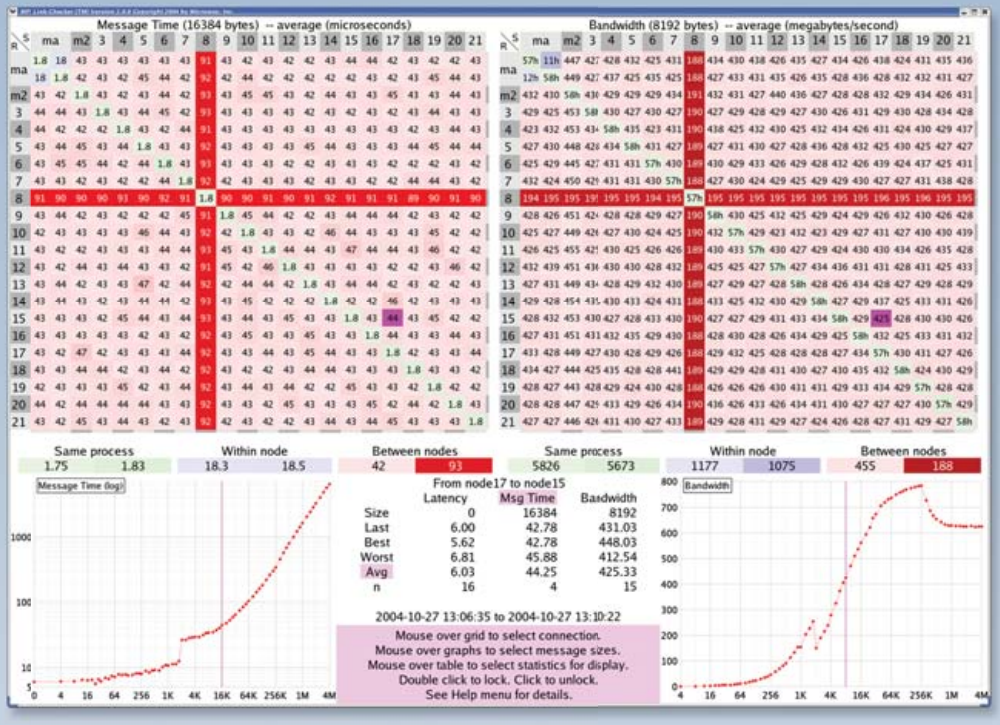
For the few legislators who already advocate balanced policy, this kind of interaction can reinforce their position and help them stay in office. For others, seeing constituent engagement—and donations—can encourage them to reexamine their policy choices. When we visited Rep. Rick Boucher, an IPac 2004 candidate, earlier this year, he argued that much of Congress simply is waiting to be convinced that these issues matter to the public. The public that I belong to cares deeply about topics such as the right to program, patent reform and balanced copyright. IPac can help amplify our voice.

IPac, like open-source development, works best when an engaged community contributes resources to achieve a common goal. The first real test of this experiment in user-modifiable politics will be the 2006 midterm election. We already are researching candidates to support, and eventually we will start the process of putting them in office. If enough of us move, I have no doubt that the balance will start to shift in our favor.

Resources for this article: www.linuxjournal.com/article/8270.

Ren Bucholz leads a double life as the co-founder of IPac and the Activism Coordinator for the Electronic Frontier Foundation.

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™