AN SSC PUBLICATION

CONVERT RADIO SHOWS TO PODCASTS, AUTOMATICALLY

LINUX JOURNAL™

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

WI-FI, MUSIC AND MOVIES IN YOUR POCKET: ARCHOS PMA400

THE IDENTITY METASYSTEM

Big companies are talking about you behind your back. Can the Identity Gang get you into the conversation?

HELLO, RUBY: Getting started with the hot "new" Web language

Securing Wi-Fi networks with WPA

Cracking WPA-protected Wi-Fi networks

Managing remote tasks safely with SSH keys

USA $5.00 CAN $6.50
www.linuxjournal.com
09
0 71486 03102 4

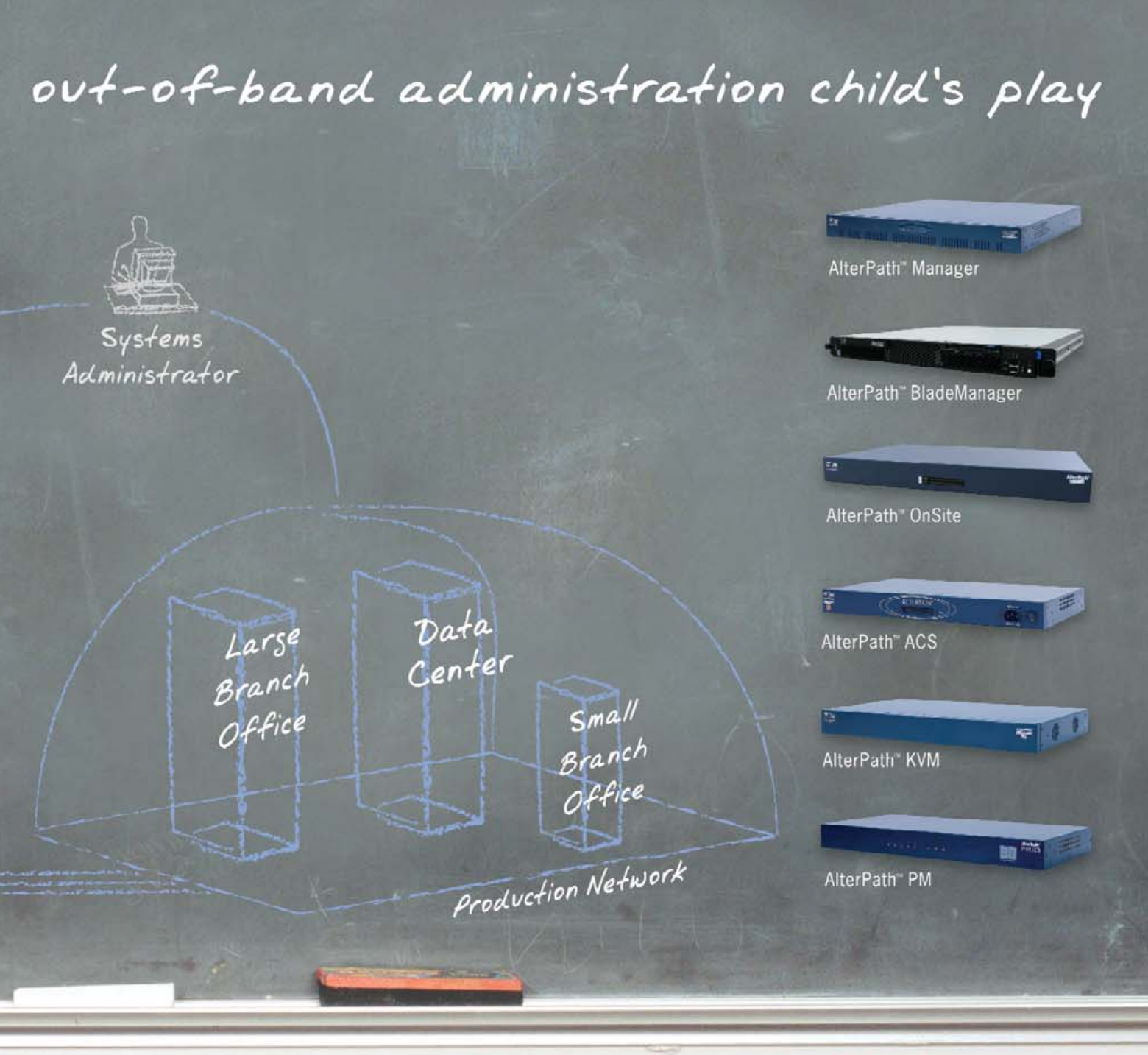
Doc Searls

LINUX JOURNAL · WIRELESS · xsupplicant · Groupware Roundup · Ruby · Identity · WPA · SSH · SEPTEMBER 2005 ISSUE 137

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

8 Opteron sockets
IWILL H8501 Barebone System

AMD 64 Opteron™

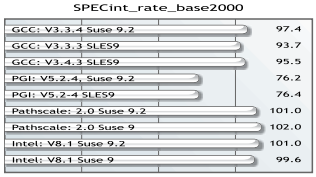
High Performance Computing
Complex Executing
DataCenter
SuperCluster

IWILL H8501 ▶

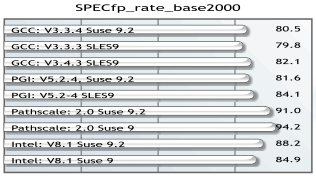
- 8x AMD Opteron Processor 940 sockets
- Supports 800 series Opteron CPUs with dual core tech.
- Up to 128GB DDR Registered ECC memory
- Support 4 Ranks memory module
- 1350W Redundant PSU 3+1
- Support IPMI server management
- Industry 19" rack-mountable 5U chassis
- 4 x Gigabit Ethernet ports, and 4 PCI-X slots
- Up to 10 hot-swap HDDs with option HDD canister
- Modularization design, I/O may vary

8-Way AMD Opteron Server Benchmark Rating
SPEC CPU2000: COMPILERS COMPARED

SPECint_rate_base2000



SPECfp_rate_base2000



Opteron Dual Core Ready

Highest Performance
Accommodate 128GB RAM
8-WAY/ 16-CORE

IWILL Other Outstanding Barebone Systems:

▼ H4203

- 4 AMD Opteron Processor 940 sockets
- Supports 8xx Opteron CPUs with dual core tech.
- Up to 64GB DDR Registered ECC memory
- Support 4 Ranks memory module
- Support IPMI server management
- Industry 19" rack-mountable 2U chassis
- 4 x Gigabit Ethernet ports via PCI-X interface
- Modularization design, I/O may vary

▼ H2B Blade Server

- 2 AMD Opteron Processor 940 sockets
- Supports 2xx Opteron CPUs with dual core tech.
- Up to 16GB DDR Registered ECC memory
- Power distribution backplane in subrack
- 8U height, 10 blades subrack
- 2 x Gigabit Ethernet ports, one PCI-X slot
- Support IPMI server management (Option)

▼ H2103

- 2 AMD Opteron Processor 940 sockets
- Supports 2xx Opteron CPUs with dual core tech.
- Up to 16GB DDR Registered ECC memory
- Support 4 Ranks memory module
- Support IPMI server management
- Industry 19" rack-mountable 1U chassis
- 2 x Gigabit Ethernet ports via PCI-E interface

▼ ZMAX-DP

- 2 AMD Opteron Processor 940 sockets
- Dual processors Small Form Factor
- Supports 2xx Opteron CPUs with dual core tech.
- Up to 4GB DDR Registered ECC memory
- 1x AGP 8X , 1x GbE, 1x PCI and 1x mini PCI slot
- 3x 3.5" HDD bays, and 1x 5.25" CD-ROM bay
- 1x IEEE1394, 8x USB 2.0 ports
- 300W Power supply

IWILL USA Corp.
9004 Research Drive
Irvine, CA 92618
Tel: +1 949 753-5488
Fax: +1 949 753-5499

Visit www.iwill.net for more information.
Or contact us: sales@iwillusa.com, oem@iwillusa.com

Iwill®

COVER STORY

FEATURES

INDEPTH

EMBEDDED

TOOLBOX

COLUMNS

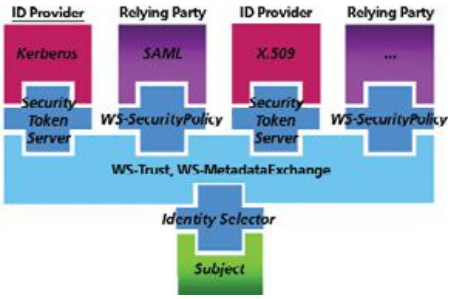
REVIEW

Cover photo: Turk's Head

LINUX JOURNAL

SEPTEMBER 2005 ISSUE 137

DEPARTMENTS



You're the green box at the bottom. The red and purple boxes at the top are companies that know something about you. Are you safe? (See page 44.)

NEXT MONTH

PERSONAL DESKTOP

Voice communication is finally discovering the low costs and versatility that come with open-source software and commodity hardware. In "Building a Call Center with LTSP and kphone", Michael George sets up an ambitious project for a cost-sensitive client using thin clients and soft phones.

Anyone who reads our "Letters to the Editor" knows that our readers are a fecund bunch—babies and children everywhere! When the little ones are ready to move beyond the stuffed penguin, have a look at proud Linux dad Paul Barry's article on his family gaming, learning and communications environment.

If 5.1 surround sound sounds a little boring, how about eight-channel, three-dimensional sound, with the speakers at the vertices of a cube? In "Dirt Cheap 3-D Spatial Audio", Eric Klein, Greg S. Schmidt, Erik B. Tomlin and Dennis G. Brown get together to make it happen, using the ALSA drivers we know and love, plus generic PC sound hardware.

Wireless on Your Own Terms

Your wireless network might need a security overhaul, but that beats the alternative.

BY DON MARTI

I booted up my new Cingular wireless phone, and the default item on the main menu is "Media Mall". How nice—they want to sell me $1.99 "ring tones" enough to make that the number one item on the menu. And even though the phone is a much better computer than my first Linux box was, the selection of apps I can install, and networks I can connect to, is locked in by Cingular.

On the other hand, when I booted up my Linux box, I got LILILILILILILILI...oh wait, let me fix that. Much better. A standard Web browser, able to connect to anybody. Able to do business with anyone's store and connect through any ISP. No, I didn't get the computer "free" with my Internet connection, but free as in cell phones is no bargain when it means being locked out of fun stuff.

Phil Salkie has a great example of why on page 48. Can't listen to your favorite Internet radio show at the time it's on? Or want to capture it to listen somewhere you don't have Net access? Time-shift it! What Phil did to make things run extra smoothly is to turn Net radio shows into RSS feeds that work as podcasts. You can use any one of a growing assortment of clients to play them on your own schedule. Before the next road trip I have to take, I'm definitely setting up a podcast client on a laptop for the car, and I'll use Phil's script to snarf some radio shows to listen to.

Speaking of devices that let you listen on the road, Dovid Kopel tried out the Archos PMA400. If you're looking for a combined PDA and music player, check out this Linux-based unit on page 74. And save yourself some bandwidth and time snarfing all those podcasts with the thorough compression tools overview from Kingsley G. Morse Jr. on page 62.

Security-wise, today's wireless networks are where the Internet was in the 1980s, before the 1988 Morris Worm helped create the network security scene as we know it. We all know that people can do bad things with all those open access points, but the bad things aren't happening. Many of you could "borrow" a departing Starbucks customer's identity to get a free Net connection, but you're not. Thank you.

But it's only a matter of time before all our nice little access points start getting clobbered by spammers, phishers and new categories of naughty people we don't even have cool words for yet. So please read our two articles on wireless security technology—John L. MacMichael's on page 56 and Matthew Gast's on page 68. Combined with Mick Bauer's series from earlier this year, this issue will get you up to speed on how to make your wireless network run smoothly.

Finally, is "identity" just a directory server that has its own fancy conference? Instead of letting big companies decide among themselves what to do with your personal information, Doc Searls ventured into the lairs of some mighty scary beasts and came back with the beginning of a plan, from a strange source. See page 40 if you dare.

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

LINUX JOURNAL

SEPTEMBER 2005
ISSUE 137

***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.

The Power of Choice...

Command the game with your next I/O move.

Modularity. Scalability. Reliability. Cost-effectiveness. These represent the solid foundations that SBE delivers to OEMs for building innovative end solutions. Partnering with SBE for networking and communications I/O solutions allows you to take advantage of proven technology and field-tested products designed to optimize performance for your unique application needs.

SBE offers a full spectrum of interface cards, ranging from T1 and T3 to Gigabit Ethernet and IPsec/SSL acceleration. These boards are available in multiple form factors, including PCI, PMC, and PTMC. Customers have the choice of buying these boards individually or bundling any of the PMC/PTMC modules with our intelligent core processing platforms to create a flexible, cost-efficient blade solution ideal for serving demanding telecom applications. Full Linux support is available on every board.

- Channelized T3
- 24-port T1/E1
- LAN/Ethernet
- Storage
- IPsec/SSL Encryption
- Blade platforms
- I/O and beyond...

SBE®

Linux On Demand

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

Painting

Well, it's rather an image. My daughter, Liv Helene, seven years old, has used Tux Paint to make a picture of a painting Tux. She has not told me what he is painting, though. My guess is that it is a kernel image.

--
Morten

See next issue for more fun Linux apps for kids.—Ed.

/var/spool/fanmail

My niece, Addison Lotspaih, really seems to like penguins. I think she was wondering "where's the Easter Penguin?" in this picture. Future Linux guru? I hope so.

I wanted to let everyone at *Linux Journal* know, I think you're doing a wonderful job! I look forward to reading your magazine each and every month.

--
Mark Lotspaih

"Wrong Sed Fred"

Thanks to Larry Richardson for an enlightening overview of the sed command [July 2005]. I've been using it for years, but did not know about its pattern matching (//command) capabilities.

My friend, Samueluel, tells me there's a small bug in the `s/Sam/Samuel/` code example, though.

--
Jeremy

```
echo "How's this, Samuel?" | \
sed s/Sam\\b/Samuel/
```

—Ed.

Historic Linux Distribution

After trying out several different Linux distributions, my baby Molly decides Caldera Network Desktop version 1.0 is just right. Should I be worried that she might become a lawyer instead of a hacker?

--
Terry

That was the first distribution to bundle a pre-Mozilla Netscape browser.—Ed.

Nice Shirts

Baby Tux is in good hands. The oldest is saying, "We're protecting Baby Tux."

--
Wendy Cho

LJ Cover Okay for US Government Offices

Anyone that thinks the May 2005 cover is "smut" is a complete idiot and doesn't deserve to read your excellent publication. I work in a Federal Government installation, and since the woman on the cover is clothed, I don't see any sexual harassment potential.

--
Tony Heaton

Another Cover Photo Fan

I just got the latest *LJ* and was shocked at reading two of the letters to the editor concerning

the May 2005 cover. I work at a software company that displayed the issue on the magazine rack for a full month. Not one person mentioned the cover image. Don't listen to the backward, foolish idiots who would have us stay in 1920. Good job, *LJ*. Don't stick to men and/or hardware on the cover. Keep branching out.

--
Dave Wiard

Millions for iSCSI, Not One Cent for *LJ*

The article on ATA over Ethernet [June 2005] has got to be the perfect example of the kind of crap that stops me from buying the magazine again. Great eye-catching title. Excellent introduction to a SAN alternative (we use a Hitachi that cost several million so I like to see what the alternatives are).

Halfway through the article I find out that the setup requires a vendor-specific piece of hardware that I'm sure not so coincidently is the advertisement facing page 26 of the article. If you're going to have multipage advertisements disguised as articles, then state it up front. I buy this magazine for Linux solutions (like iSCSI target and host on Linux) primarily. I don't mind seeing vendor stuff, but don't dress it up as something else. It wastes my time.

--
Henry Scott

AoE is an open standard protocol that any vendor is free to implement. Check out sourceforge.net/projects/aoetools *for a free server and recent Linux releases for the client code.—Ed.*

GB: the New MB?

In the third paragraph of the July 2005 "diff -u" section, it states: "Of course, because SquashFS is a compressed filesystem, this really amounts to about 8MB of actual data." This should probably be 8GB of data instead of 8MB.

--
Dan Eisenhut

Yay, Monarch

I was reading your July 2005 issue, and I saw an ad for Monarch Computer. I wanted to drop you a note to tell you that I have been buying from them for five years now. My company buys all of our systems from them, I have never heard of a company that bends over backwards for their customers like Monarch does. They have over-nighted me parts at no charge, along with a number of other amazing things.

--

Scott Adams

Linux Now on Tour

It is sad to see the ignorance and prudishness in some of letters published in the June 2005 issue with regards to the cover of the May 2005 issue.

Belly dancing has been a form of improvisational dance from the Middle East, for both women and men, dating back thousands of years. The dance is not inherently sexual in nature, and while the image on the cover does display a fair amount of belly, it is far less than what one would see at a swimming pool or at the beach.

I found the cover to be very tasteful and artistic. The article was quite interesting and already has proven useful in helping to spread Linux to several of my friends who are about to start touring with their (non-belly dancing) shows. Lastly, it was very nice to see a cover that explodes the male-computer-geek stereotype so effectively. My wife even picked up my copy just to read the cover article. Keep up the good work.

--

Chris Poupart

Everyone's Playing *Tux Racer*

Greetings from Mexico. This is my niece Regina, enjoying *Tux Racer* on my Debian box.

--

Leonardo Ibarra Moran

Bad Advertiser, Bad, Bad

I would like to register my protest against the publishing of the advertisement that appeared on page 7 of the July 2005 issue.

I'm not a member of any identifiable minority—obese, ethnic, physically or mentally handicapped—but I strongly disapprove of the use of photos such as the one in the advertisement. I hope that I don't need to explain why.

For several years I have been a regular buyer of *Linux Journal* and a proud owner of two of Marcel Gagné's books. Any remote idea of ever buying a Carinet server has now been banished from my mind, forever.

May I suggest that you publish some sort of apology in the next issue?

--

Alan Eastgate

Everyone's Favorite Magazine

I wanted to send you a picture for your next issue. We get *LJ* sent to our office every month, and between the developers and sysadmins it is a race to crack into the magazine first; however, we all lost this time. The newest addition to our development team got to it first. Keep up the good work *LJ*.

--

Travis

Reply to July "Vendor Troubles" Letter

Monarch apologizes for the inconvenience to the valued customer [Letters, July 2005]. We take pride in taking care of our customers. Upon receiving his contact info, we contacted him immediately and are pleased to say he has been taken care of to his satisfaction.

We checked our Web/voice-mail logs and found no trace of receiving a message related to this customer's problem, which is why there was not an immediate response to the customer's inquiry. Monarch handles over 250k customers a year, more than 700 per day, and has the highest rating for customer service based on volume at www.resellerratings.com; unfortunately, this one got through the cracks. We thank *Linux Journal* for bringing this to our attention.

--

Trey Harris, Monarch Computer, Founder

Tux over My Hammy

I have been an avid *LJ* reader for a couple of years now and always look forward to receiving my monthly dose of Linux news updates from you all. Here is a picture of a 72 processor Linux cluster install at Florida International University. On a funny note, I won the stuffed Tux out of one of those prize machines during a 4 AM run to Denny's on the last day of the install.

--

Jason Grimm

GIMP and nvu on Ubuntu

This photo is of my 11-year-old daughter Julia Meadows, who is happily using Ubuntu Linux. She enjoys working with the nvu Web authoring tool and is having fun learning GIMP. She is also holding her own personal penguin, from an art class project at school.

--

John Meadows

LETTERS CONTINUED ON PAGE 76

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 w/ AMD Sempron™ processor 2600+ (754 - 64 bit) — Only $199

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

Tyan S2882UG3NR Mainboard with AMD Opteron™ processor 244 — Only $459

Tyan S2882GNR-D (Thunder K8S) with AMD Opteron™ processor 265 (Dual Core) — Only $1219

WinFast NF4K8MC-ERS nForce 4 Mainboard w/ AMD Athlon™ 64 processor 3200+ (939) — Only $269

Asus K8N-DL nForce4 Pro MB w/ AMD Opteron™ processor 275 (Dual Core) — Only $1539

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-V1000B Plus Quiet Tower (Black) w/Wheels w/460W PS
Asus K8N-DL nForce4 Motherboard
AMD Opteron™ processor 244 1.4 GHz
Starting @ $699

Thermaltake Armor Full Tower
E-ATX Case w/Window w/460W PS
Gigabyte GA-K8NF-9 nForce 4
AMD Athlon™ 64 processor 3500+ (939 - 90nm)
Starting @ $589

Antec Titan 550 Tower
E-ATX Case (Black) w/550W PS
Tyan S2882-DG3NR Thunder K8SD
AMD Opteron™ processor 270 (Dual Core)
Starting @ $1599

Antec Plusview 1000AMG Case w/300W PS
MSI K8M Neo-V Motherboard w/
AMD Sempron™ processor 2600+ (754 - 64 bit)
Starting @ $209

***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 Retail Box CPUs
AMD Athlon™ 64 3000+ (754) $146.00
AMD Athlon™ 64 3200+ (754) $190.00
AMD Athlon™ 64 3500+ (939) $219.00
AMD Athlon™ 64 3700+ (939) $267.00
AMD Athlon™ 64 3800+ (939) $323.00
AMD Athlon™ 64 4000+ (939) $368.00
AMD Athlon™ 64 FX-55 (939) $811.00
AMD Athlon™ 64 FX-57 (939) $1062.00

AMD Athlon™ 64 X2 Retail CPUs
AMD Athlon™ 64 X2 4200+ $557.00
AMD Athlon™ 64 X2 4400+ $603.00
AMD Athlon™ 64 X2 4600+ $824.00
AMD Athlon™ 64 X2 4800+ $1035.00

AMD Opteron™ OEM CPUs
AMD Opteron™ 144 1.8GHz $158.00
AMD Opteron™ 146 2.0GHz $173.00
AMD Opteron™ 148 2.2GHz $211.00
AMD Opteron™ 150 2.4GHz $270.00
AMD Opteron™ 242 1.6GHz $404.00
AMD Opteron™ 244 1.8GHz $158.00
AMD Opteron™ 246 2.0GHz $203.00
AMD Opteron™ 248 2.2GHz $307.00
AMD Opteron™ 250 2.4GHz $441.00
AMD Opteron™ 252 2.6GHz $669.00
AMD Opteron™ 846 2.0GHz $677.00
AMD Opteron™ 848 2.2GHz $677.00
AMD Opteron™ 850 2.4GHz $847.00
AMD Opteron™ 852 2.6GHz $1130.00

AMD Opteron™ Dual Core Retail Box CPUs
AMD Opteron™ 265 1.8GHz $825.00
AMD Opteron™ 270 2.0GHz $1019.00
AMD Opteron™ 275 2.2GHz $1260.00
AMD Opteron™ 865 1.8GHz $1469.00
AMD Opteron™ 870 2.0GHz $2085.00
AMD Opteron™ 875 2.2GHz $2570.00

The AMD Athlon™ 64 X2 dual-core processor provides the ame level of system features customers have grown to expect with the AMD Athlon™ 64 product family: HyperTransport™ technology - Enhanced Virus Protection for Microsoft® Windows® XP- SP2 - Cool'n'Quiet™ technology

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
$109.00

140270
1 GB DDR (266) PC-2100 REG ECC Corsair (CM72SD1024RLP-2100)
$118.00

140662
512 MB DDR (266) PC-2100 REG-ECC Mushkin Pro (990742)
$85.00

190528
ATI (Connect3D) Radeon 9250 SE 128MB DDR/8x-AGP/TV-Out (Retail Box)
$36.00

150239
Western Digital 74 GB SATA 10K Raptor (WD740GD)
$175.00

100122
Enermax EG565P-FMA REV.2.0 ATX 2.0 w/SLI Support 535W Power Supply
$89.00

110192
Asus K8N-DL nForce4 Audio/GB-LAN/USB/IEEE PCI-E/SATA/DDR/E-ATX Opteron
$284.00

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 64 Athlon X2 | AMD Sempron | AMD 64 Athlon FX | AMD 64 Athlon | AMD 64 Opteron

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"

LINUX MAGAZINE EMPEROR CLASS

"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

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!

Part #: 80601
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)
16GB (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 $18,439!

Monarch Empro™ Custom Rack Server (2U configuration)

Part #: 80313
SELECTED COMPONENTS:
AIC RMC2K2-9I-XPSS 2U Rack Mount w/460W Power Supply
Tyan S2882G3NR-D (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

1 Terabyte Shown Here - Upgradable to 2 Terabytes!

This Configuration Starting @ ONLY $3,389!

Monarch Empro™ Custom Rack Server (3U configuration)

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

1 Terabyte Shown Here - Upgradable to 3 Terabytes!

This Configuration Starting @ ONLY $3,819!

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

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)

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

This Configuration Starting @ ONLY $1,589!

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

On the WEB

» Matthew Gast, author of this month's article on xsupplicant, spent part of his summer co-teaching a class in Australia with Chris Hessing, lead developer of the xsupplicant Project. Lucky for us, the *LJ* Web site got an "Interview with Chris Hessing, Lead Developer of xsupplicant" article out of the deal (www.linuxjournal.com/article/8388). Read what Chris had to say about developing one of the two major 802.1x client packages for Linux, including some of the challenges of developing on Linux, keeping up with the relevant standards and why it is difficult to build WPA(2)/802.11i support for Linux.

» In an upcoming article, Michael George will be covering "Building a Call Center with LTSP and KPhone", including some of the problems he ran into when installing and configuring KPhone, such as a configuration option that doesn't seem to change anything and icons not being found. Because of these factors, "Building kphone was a bit more complicated", and our Editor in Chief asked Michael to follow up with the KPhone developers by submitting a bug report. In his *LJ* Web site article "Filing a KPhone Bug Report" (www.linuxjournal.com/article/8389), Michael updates us on what, if any, progress has been made.

diff -u

What's New in Kernel Development

The **git** phenomenon continues. Self-hosting and fast as lightning at three days old, git and its favored set of wrapper scripts **Cogito** have continued to improve and have already had a huge impact on kernel development. Projects left and right are migrating from BitKeeper to git. **Net driver** development and **libata** development have switched. **JFS** and **NTFS** have switched. And the **stable** w.x.y.z kernel tree, maintained by **Greg Kroah-Hartman** and **Chris Wright**, has also converted recently to git. Some kernel hackers find git to be such an improvement over BitKeeper that they are able to produce much more work than they had before, to the point that **Linus Torvalds** is having to rethink the way he handles patches in order to accommodate them. BitKeeper documentation has been removed from the kernel sources, and mailing lists such as bk-commits-head that originally were intended to receive announcements of new BitKeeper changesets, receive git kernel patch notices instead.

However, git is not for everyone. When asked, **Andrew Morton** said he did not intend to use git for his -mm kernel tree, because his set of patching scripts are still sufficient for his needs. Also, **Matt Mackall** has been working on his own fast-as-lightning version-control system, **Mercurial**. This is also an excellent tool and shows itself to be the equal of git in a number of ways, especially speed. In fact, as Linus has pointed out, the two are actually quite similar in their underlying behaviors. Clearly, both of these tools represent an entirely new look at version control, and projects like arch that had previously been in the lead are finding themselves struggling to catch up.

Markus Klotzbuecher has produced an interesting new virtual filesystem called **mini_fo** (fanout overlay). It allows users to write to files on read-only filesystems, by creating a writable area elsewhere and layering the user's changes from that area on top of the read-only data. To the user, the effect is transparent. What had been read-only, now appears writable, although in fact the read-only data is never actually modified. The mini_fo tool is intended to allow software upgrades on embedded systems, but other uses have already been found and more undoubtedly will appear.

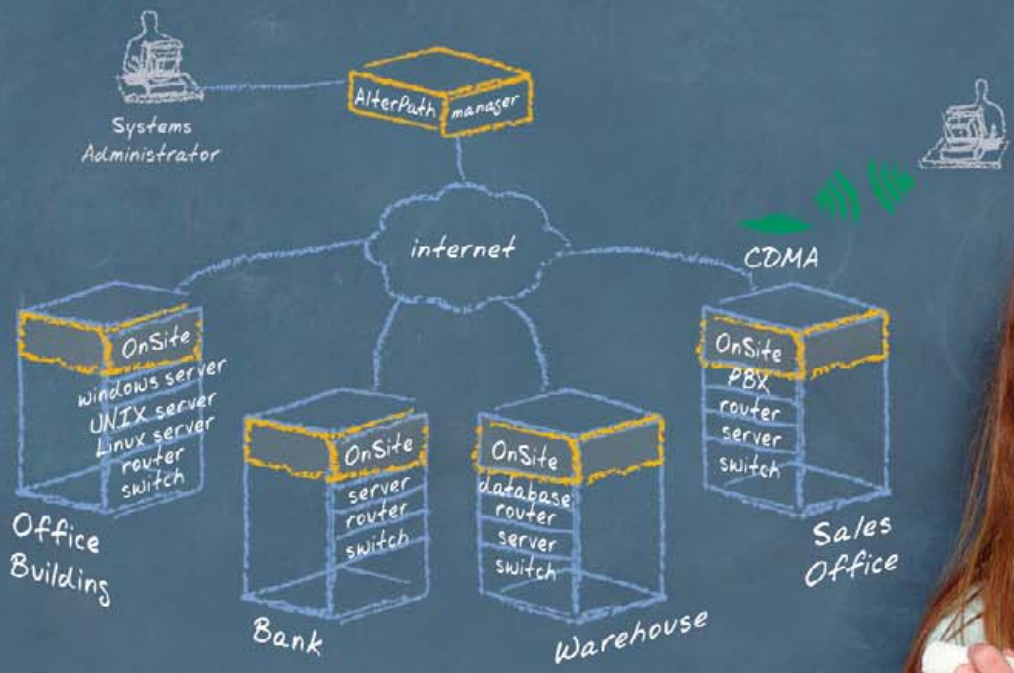
Alan Cox and **Bartlomiej Zolnierkiewicz**, two big-time **IDE** developers, are having trouble pooling their efforts. While Bartlomiej is the current IDE maintainer, Alan did much of the initial work to convert the old IDE code into a maintainable state from the unutterable corpse-strewn nightmare it had been for years. Although Alan has been somewhat out of the Linux picture lately, he returned to check on the progress of IDE and didn't like some of the changes Bartlomiej had made. There does seem to be some bad blood between them, as there seems to be between any developers who have at one time or another worked on IDE. Bartlomiej invited Alan to fork the code and do things differently if he pleased.

It's sad that IDE is still able to inspire enmity among developers. For this we should blame the IDE disk industry itself, which has so warped and destroyed any possible standard, creating exception upon exception upon exception, compounding all of it with trade secrets and proprietary documentation, that anyone would have to be insane even to attempt to maintain the IDE kernel code. That folks like Bartlomiej and Alan, and those that came before them, like **Mark Lord** and **Andre Hedrick**, have done so is a tribute to their generous natures. Without the IDE code, most of us would not find Linux nearly as useful.

Benjamin LaHaise recently tried to simplify and make more maintainable the implementation of **semaphore locking** across kernel architectures. The current code is complex and difficult to read, with many architecture-specific details. These nuances have grown with the number of supported architectures, and the natural inclination is to create a generic semaphore system that compiles and works on all architectures uniformly. However, semaphores run deep, and the need for blinding speed in that code is difficult to compromise on. Given that any slowdown would have a noticeable impact on kernel performance, it's likely that any attempt at code unification will meet strong resistance by the maintainers of the various architectures. This is in fact how Benjamin's work was received. And so, while some improvements certainly may be made, it seems unlikely that the semaphore code will ever become truly generic and simple. Speed is just too strong an incentive.

—ZACK BROWN

Cyclades AlterPath™ OnSite makes branch office administration child's play



AlterPath™ OnSite

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.

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

Over 85% of Fortune 100 choose Cyclades.

www.cyclades.com/ljb

1.888.cyclades · sales@cyclades.com

cyclades

GUADEC 2005

GNOME users and developers alike descended on Stuttgart, home of Porsche in southern Germany, for the sixth annual GUADEC—the GNOME User and Developer European Conference—to discuss the future of GNOME development, hear about business and government migration to Linux desktops, marvel at the latest cool applications and cunning hacks, and witness Nokia's awesome announcement of its GNOME-powered 770 Internet Tablet and developer's program.

Keynote speakers at the May 29–31, 2005 conference included Miguel de Icaza, GNOME founder and Vice President at Novell; Mark Shuttleworth, Canonical founder and one-time cosmonaut; and Nathan Wilson, Software Lead at DreamWorks Animation Studios. Miguel gave, as usual, a rousing talk on the future of GNOME, Mono and Linux as a whole, and Nathan discussed DreamWorks' transition to a GNOME-based Linux desktop. *Madagascar*, DreamWorks' next animated feature film, was made with GNOME!

Quality speakers abounded too, including Owen Taylor, Gtk maintainer; Keith Packard, X supreme commander; and Jon Trowbridge, Beagle maintainer. Anna Dirks and Pete Goodall, both of Novell, presented results from their in-depth Windows Migration Study. Glynn Foster of Sun gave a moving introduction to the GNOME Project. They even let me speak, against better judgment, on optimal GNOME programming techniques.

The most valuable part of the conference, however, was the informal hallway chatter and the widely attended hackfests, where GNOME hackers could meet (many for the first time), debate hot topics and then sit down and hack out elegant solutions.

GUADEC veterans such as Nat Friedman remarked that "this may be the best GUADEC ever." While I was without reference—this was my first, but hopefully not my last GUADEC—it was certainly time well spent, in a beautiful city with some smart folks.

—ROBERT LOVE

They Said It

...how are you arranging your conclusions so that your current experiences fit into your scheme of complacency?

—CARLOS CASTENEDA, *The Power of Silence*

The greatest mistake we can make is to be continually fearing that we will make one.

—ELBERT HUBBARD,
www.downtheavenue.com/2005/05/random_reflecti.html

The lack of any concrete numbers at all shows the typical academic hand-wavy "our asymptotic is good, we don't need to worry about reality" approach. Good asymptotics are one thing, but constant multipliers can be killer, and it's necessary to work out constant multipliers for all potentially problematic constants, not just the easy ones like CPU.

—BRAM COHEN, www.livejournal.com/users/bramcohen/20140.html

He is no fool, who gives what he cannot keep to gain what he cannot lose.

—JIM ELLIOT, www.tonywoodlief.com

The true lesson of the Internet is in the end-to-end argument. It gives us a real working model of how individual efforts can composite into a valuable whole.

—BOB FRANKSTON,
www.frankston.com/public/writing.asp?name=DIYConnectivity

A Report from the Linux Audio Conference 2005

Linux audio developers and musicians came together for the third annual Linux Audio Conference, held April 21–24 at the Zentrum für Kunst und Medientechnologie (ZKM, the Center for Art and Media Technology) in Karlsruhe, Germany. This event included topic presentations on sound and music software, such as hard-disk recording, sound synthesis languages, music composition, softsynths, MIDI technologies and audio-optimized Linux distributions, along with two concerts and two more music programs. The entire conference was broadcast and recorded in audio and video formats.

Ivica Bukvic stressed the need for a coordinating effort to ally developers with manufacturers, vendors and possible sponsors more closely, while Christoph Eckert focused on the need for greater concern for usability issues. The conference also included an update from Maarije Baalman regarding her WONDER software and a report from Georg Bonn on free software for music composition.

A presentation on FireWire as an audio interface created quite a stir among developers, particularly since no manufacturer has yet disclosed driver code for any available FireWire audio interface. Track 2 covered software, including the MusE audio/MIDI sequencer, the ZynAddSubFX synthesizer and the Ardour digital audio workstation.

Linux sound and music applications have become a real choice for musicians and sound researchers, and the music at both concerts put the issue to rest. Participants and guests also enjoyed Linux Sound Night, during which developers of Linux music performance software got to show us what their programs could do. The Sound Night shaded imperceptibly into Linux Chill Night, ending an exhilarating day with more music, dancing, drinks and conversation.

Standout presentations included the FireWire and Ardour demos already mentioned, as well as Julian Claasen's presentation on his text-based studio. The Ogg Vorbis and Ogg Theora audio and video streams were set up and managed by conference stalwart Joern Nettingsmeier and newcomer Erik Rzewnicki. The A/V streams were themselves a demonstration of the power of free and open-source software. I am happy to announce that LAC2006 will be held again at ZKM.

The motley crew (photo by Frank Neumann).

Resources for this article:
www.linuxjournal.com/article/8410.

—DAVE PHILLIPS

Reduce Your Deployment and Support Costs

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

Supermicro 5013G-MB

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

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

Or Promote Your Brand

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

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

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

MBX™
MOTHERBOARDEXPRESS

www.mbx.com
1.800.688.2347

Getting Started with Ruby

What's behind all the Ruby hype? Reuven walks us through a couple of examples to let the code speak for itself. BY REUVEN M. LERNER

About ten years ago, back when I was working in New York, friends of mine showed me something that knocked my socks off—a program that actually ran inside of the Web browser, without any need for pressing submit. It was sleek, fun to use and seemed like a major paradigm shift. We all were excited about what this new "Java" language and its applets would mean for Web development. Although we didn't quite know where or how it would end, we talked about nothing else for some time.

In the decade since then, many different technologies have been hyped as "the next best thing" or "the tool you need to make better Web sites". Indeed, we constantly are bombarded with claims of newer, better, faster and cheaper ways to develop software. Some of these promises have panned out, but a trade-off usually is associated with them. For example, developing Web applications in Zope is indeed quite easy—once you get over the learning curve. Web services are fine, until you start to deal with complex data structures across different platforms.

You can imagine my surprise, then, when I began to see another "best new method" coming over the horizon—but this one was touted by people I respect, who normally don't give in to hype so quickly. I'm speaking, of course, about "Ruby on Rails", an object-oriented system for creating and deploying Web applications. For several months now, I have been reading about how wonderful Rails is and how it makes Web development utterly simple.

I had been meaning to try Ruby as a language for some time, and the growth of Rails has given me an opportunity to do so. This month, we take an initial look at Ruby, examining simple ways to create Web applications with the basic Ruby language and libraries. In my next article, we will look at Rails and see how it stacks up against other, more established frameworks.

What Is Ruby?

Ruby is an open-source programming language originally developed by Japanese programmer Yukihiro Matsumoto, also known as Matz. Ruby first was released in late 1995, making it older than many people might think. It took some time for people outside of Japan to discover and work with Ruby, in part because of the lack of documentation. The first edition of Dave Thomas' book, *Programming Ruby* (see the on-line Resources) provided a solid introduction to the language, as well as a reference guide to its class libraries, giving it a needed PR boost. The second version of the "Pickaxe book", as it is known, now is available.

Ruby was designed to be an "object-oriented scripting language", and it indeed feels like a cross between Perl and Smalltalk. It assumes that you understand object-oriented programming and probably is not a good first language for someone to learn. But if you are familiar with both objects and Perl, then you quickly can learn to do many things with Ruby.

Here is a simple "Hello, world" program in Ruby:

```
#!/usr/bin/env ruby

print "Hello, world\n"
```

The first line ensures that we run the Ruby interpreter, regardless of where it might be in our path. The second line, as you might expect, prints `"Hello, world"` followed by a newline character. Like Python and unlike Perl, no semicolon is required at the end of a line of Ruby code.

Now that we have created a simple command-line program, it's time to create an equivalent CGI program. CGI programs are portable across all types of Web servers. Although not particularly fast or smart, they are easy to write and a good way to dip our toes into the Web development side of a language.

In the case of Ruby, the easiest CGI program would be similar to the above code. After all, the CGI specification tells us that anything written to standard output is sent to the user's Web browser. So long as we send a Content-type header before our text, we can make it a CGI program with almost no effort:

```
#!/usr/bin/env ruby

# HTTP response headers, including double newline
print "Content-type: text/plain\n\n"

# Contents
print "Hello, world\n"
```

Sure enough, naming the above program hello.rb, putting it in my Web server's cgi-bin directory and pointing my Web browser to http://localhost/cgi-bin/hello.rb produces the `"Hello, world"` message in my browser.

Using the Ruby Library

The CGI object in the included Ruby library provides methods that understand Web functionality, from HTML formatting to cookies and parameters. For example, here is a new version of our "Hello, world" program written to use the built-in functionality:

```
#!/usr/bin/env ruby
# *-ruby-*-

require 'cgi'
```

EmperorLinux

...where Linux & laptops converge

You choose your laptop... from a wide selection of top tier laptops manufactured by IBM/Lenovo, Dell, Sharp, and Sony. They come in all sizes from two pound ultra-portables to eight pound desktop replacements; get exactly as much Linux laptop as you need. Need help deciding? Our experts will help you select a Linux laptop to meet your needs.

The Meteor: 3lb Linux

- Sharp Actius MM20/MP30
- 10.4" XGA screen
- 1.6 GHz Transmeta Efficeon
- 20-40 GB hard drive
- 512-1024 MB RAM
- CDRW/DVD (MP30)
- 802.11b/g wireless
- ACPI hibernate
- 1" thin
- Ask about the 3D Molecule

The SilverComet: 4 lb Linux

- Sony VAIO S380
- 13.3" WXGA+ screen
- X@1280x800
- 1.6-2.13 GHz Pentium-M
- 40-100 GB hard drive
- 256 1024 MB RAM
- CDRW/DVD or DVD-RW
- 802.11b/g wireless
- ACPI hibernate
- Ask about the 17" Gazelle

You choose your distribution... from among the most popular Linux distributions available. We'll install the distribution you select, then we'll install our custom, laptop-specific kernel and configure your distribution for full hardware support, including: X at the native resolution, wireless ethernet, power management, 3-D graphics, optical drives, and more.

The Toucan: 5 lb Linux

- IBM/Lenovo 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
- APM suspend/hibernate
- Ask about the 3 lb Raven X41

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
- 30-100 GB hard drive (7200 rpm)
- 256-2048 MB RAM
- CDRW/DVD or DVD±RW
- 802.11a/b/g wireless, GigE
- Ask about the tiny Koala X1

Let EmperorLinux do the rest. 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/Lenovo, Dell, Sharp, and Sony. We offer a range of the latest Linux distributions, as well as Windows dual boot options. All systems come with one year of Linux technical support by both phone and email, and full manufacturers' warranties apply.

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

```
# Create an instance of CGI, with HTML 4 output
cgi = CGI.new("html4")

# Send the following to the CGI object's output
cgi.out {
  cgi.html {

# Produce a header
cgi.head { cgi.title { "This is a title" }
} +

# Produce a body
cgi.body {
  cgi.h1 { "This is a headline" } +
    cgi.p { "Hello, world." }
}
  }
}
```

As you can see, the code now looks substantially different, even though the output largely is the same. What we have done is switched from explicit print statements to methods invoked on our CGI object, as well as added a title and a headline.

When we create our CGI object with CGI.new, we can pass an argument indicating the level of HTML compliance we want to have. Unless you have a good reason to do otherwise, aiming for the highest level of compliance, namely HTML4, is a good idea.

Notice how the output, beginning with cgi.out, functions as a set of code blocks, each of which is expected to return a text string. Thus, cgi.h1 and cgi.p are combined—using the + operator, as in Python or Java—and are fed to cgi.body. cgi.head and cgi.body are joined as well and fed to cgi.html. The fact that this hierarchy mimics the eventual document output format makes it easy to understand and use this functionality.

CGI programs are more interesting when they handle parameters from the user. We can get parameters with the CGI.params method:

```
#!/usr/bin/env ruby
# *-ruby-*-

require 'cgi'

# Create an instance of CGI
cgi = CGI.new("html4")

# Get our first name
firstname = cgi.params['firstname']
if (firstname.empty?)
  firstname = '(No firstname)'
end

# Get our last name
lastname = cgi.params['lastname']
if (lastname.empty?)
  lastname = '(No lastname)'
end

# Send some output to the end user
cgi.out {

  cgi.html {

# Produce a header
cgi.head { cgi.title { "This is a title" }
} +

# Produce a body
cgi.body {
  cgi.h1 { "This is a headline" } +
    cgi.p { "Hello, #{firstname} #{lastname}." }
}
  }
}
```

There are two basic differences between this code and its predecessor. To begin with, we now are defining two variables, firstname and lastname, which we then print for the user. The variables are defined based on the parameter values passed to the program, either by way of the URL in a GET request or in the body of the request for POST. We use the empty? method on both firstname and lastname to check whether they are empty and then assign a default value to them if that is the case. Finally, we use Ruby's #{expression} syntax within double-quoted strings to display the user's first and last names.

WEBrick

The above are what we might expect from simple CGI programs—easy to write, easy to work with and slow to execute. If our programs get any more complicated, we have to deal with new issues that we might prefer to ignore, such as personalization.

Luckily, Ruby comes with its own HTTP server, known as WEBrick, that is similar in some ways to AOLserver or mod_perl. There is also mod_ruby, if you are interested in a more direct equivalent to mod_perl, that runs under Apache. To start a basic HTTP server on port 8000, looking at the same static documents as Apache, use the following code:

```
#!/usr/bin/env ruby
# *-ruby-*-

require 'webrick'
include WEBrick

# Create an HTTP server
s = HTTPServer.new(
  :Port            => 8000,
  :DocumentRoot    => "/usr/local/apache/htdocs/"
)

# When the server gets a control-C, kill it
trap("INT"){ s.shutdown }

# Start the server
s.start
```

There are several things to note here. First, there isn't much

Where Open Minds Meet...

LINUXWORLD® CONFERENCE & EXPO

October 5-6 • 2005 • Olympia 2 • London

The UK's leading event for Linux and Open Source in business

Come along to LinuxWorld Expo and tackle IT business issues, gain real-time Linux and open source solutions, meet key suppliers, ask technical points and get answers, discover 'how-to', see new technologies, source products and network with the entire community; experts, colleagues and suppliers;

Register NOW at www.linuxworldexpo.co.uk for FREE entry into the exhibition, featuring;

- The Great Linux Debate
- OSC/Open Source Academy
- FREE Product Briefings and Demonstrations
- FREE Showcases of Technology
- FREE Case Study Presentations
- FREE Open Forum Europe Advice Centre
- .org village
- Internet Café with Wireless Connection
- Pre-registered visitors can take the LPI exam for ONLY £25 – Saving £100

CONFERENCE PROGRAMMES

Technical and Enterprise:
Two streams, two days of informative and in-depth sessions covering key issues and topics for technical and enterprise.

Linux in Enterprise:
Practical applications, benefits and analysis of Linux and open source in business:

SPEAKERS INCLUDE:

Bill Weinberg, *OSDL*
Kevin Carmony, *CEO, Linspire Inc*
Glenn McKnight, *Linux Professional Institute*
Andrew Eddie, *Project Director, Mambo Open Source Project and Senior Systems Integrator, Toowoomba City Council, Australia*

Linux for the Technical team:
What's here now; and what's coming next.

SPEAKERS INCLUDE:

Rasmus Lerdorf, *Creator of PHP*
Jeremy Allison, *Samba Co-developer*
Paul Everitt, *Zope Europe Association*
Larry Wall, *Creator of Perl*

Delegate places are limited, so book today and also benefit from:

- FREE LPI Examination – Saving £125
- GUARANTEED seating at The Great Linux Debate
- FREE Entry to the exhibition

View the full conference programme and book your place TODAY! - **www.linuxworldexpo.co.uk**

Register NOW at www.linuxworldexpo.co.uk

INTERNATIONAL MEDIA PARTNER: LINUX MAGAZINE
MEDIA PARTNER: LINUXUSER & DEVELOPER
PLATINUM SPONSORS: ORACLE, Novell
OWNED BY: IDG WORLD EXPO
ORGANISED BY: turret GROUP International Business Publications & Exhibitions

In spirit and terminology, there is a fair amount of overlap between WEBrick servlets and Java servlets.

code. You indicate what port WEBrick should listen to, tell it where files are located and then start it up.

Before we start the server, we have to make sure it is possible to stop it easily. To do that, we invoke trap, indicating that we want to trap SIGINT (that is, Ctrl-C) and that s.shutdown should be invoked upon receiving that signal.

If you put the above program in a file named server.rb and execute it, you should have a fully functional HTTP server running on your system. Creating a Web server has never been simpler.

Ruby Servlets

Of course, no one runs WEBrick instead of Apache for its speed or to serve static documents. Rather, WEBrick shines when you want to create custom behaviors. In spirit and terminology, there is a fair amount of overlap between WEBrick servlets and Java servlets. The basic idea is the same: define a new class and then attach an instance of that class to a particular URL. For example, if we want to create a servlet that prints the time of day, we can create the following:

```
#!/sw/bin/ruby
require 'webrick'
include WEBrick

# --------------------------------------------
# Define a new class
class CurrentTimeServlet
  < WEBrick::HTTPServlet::AbstractServlet

  def do_GET(request, response)
    response['Content-Type'] = 'text/plain'
    response.status = 200
    response.body = Time.now.to_s + "\n"
  end
end

# --------------------------------------------
# Create an HTTP server
s = HTTPServer.new(
  :Port            => 8000,
  :DocumentRoot    => "/usr/local/apache/htdocs/"
)

s.mount("/time", CurrentTimeServlet)

# When the server gets a control-C, kill it
trap("INT"){ s.shutdown }

# Start the server
s.start
```

Our one file contains both the class definition for CurrentTimeServlet and the commands for starting WEBrick. This is not the most elegant style for creating a servlet, and you typically want to put each servlet in its own file. That said, Ruby makes it easy and convenient to define or redefine classes and methods wherever it might be best to do so. This is one of those features in Ruby that reminds me of Perl: the language gives you a great deal of flexibility when writing your code but expects you to be responsible enough to avoid making a mess of it.

We define our servlet, CurrentTimeServlet, to be a subclass of WEBrick::HTTPServlet::AbstractServlet, making it a simple servlet indeed. We then define the do_GET method along with the do_POST method, if you so desire, which gets both a request and a response object. If you have written Java servlets, this should look familiar to you. We set the content type of the response, the status code (200) for the response and even the body of the response with a few simple lines of code. And that's it; our servlet has been defined and is ready to go. All that is left to do is connect the servlet to a URL:

```
s.mount("/time", CurrentTimeServlet)
```

If we want, we can pass parameters to the servlet when we initialize it. Anything beyond the first two parameters to s.mount is sent:

```
s.mount("/time", CurrentTimeServlet, 'a parameter')
```

Conclusion

Is it amazing that we can do this much in so few lines of code? Perhaps—although similar functionality certainly exists in other languages. For example, Perl programmers can download HTTP::Server::Simple from CPAN and do many of the same things. And if I really was interested in modifying the behavior of an HTTP server to do interesting things, I probably would think of using mod_perl or AOLserver first, for reasons of performance and flexibility.

That said, WEBrick is extremely easy to get running and for creating custom HTTP-based behaviors. I can imagine using it to handle Web services, for example, because of the flexibility that Ruby brings to the table, or to test applications written in Rails.

And, although people are using Ruby and WEBrick for plain-vanilla Web development, most of the excitement seems to be over the specific Rails framework, rather than Ruby or WEBrick themselves. In my next article, we will start to explore Rails—how to install it, how to develop applications with it and how it stacks up against other open-source application frameworks.

Resources for this article: www.linuxjournal.com/article/8397.

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.

Turn Control Freaks Into Remote Control Freaks.

Take charge. Win the battle and take control, right from your comfy chair. It's easy to conquer the challenges of managing serial devices in the data center with the CCM serial console manager.* When used with DSView® 3, AVWorks®, or industry-standard SSH/Telnet client software, you can remotely control servers, network gear, telco and power devices from a single interface. With proactive alerts and offline buffering you can remotely diagnose failed devices and reduce downtime without setting foot in the data center. Visit us at www.avocent.com/serialcontrol. And start looking for a new chair.

Avocent®
The Power of Being There®

** Recliner recommended, but not included.*

Sleeping in the Kernel

The old sleep_on() function won't work reliably in an age of SMP systems and hyperthreaded processors. Here's how to make a process sleep in a safe, cross-platform way. BY KEDAR SOVANI

In Linux kernel programming, there are numerous occasions when processes wait until something occurs or when sleeping processes need to be woken up to get some work done. There are different ways to achieve these things.

All of the discussion in this article refers to kernel mode execution. A reference to a process means execution in kernel space in the context of that process.

Some kernel code examples have been reformatted to fit this print format. Line numbers refer to lines in the original file.

The schedule() Function

In Linux, the ready-to-run processes are maintained on a run queue. A ready-to-run process has the state TASK_RUNNING. Once the timeslice of a running process is over, the Linux scheduler picks up another appropriate process from the run queue and allocates CPU power to that process.

A process also voluntarily can relinquish the CPU. The schedule() function could be used by a process to indicate voluntarily to the scheduler that it can schedule some other process on the processor.

Once the process is scheduled back again, execution begins from the point where the process had stopped—that is, execution begins from the call to the schedule() function.

At times, processes want to wait until a certain event occurs, such as a device to initialise, I/O to complete or a timer to expire. In such a case, the process is said to sleep on that event. A process can go to sleep using the schedule() function. The following code puts the executing process to sleep:

```
sleeping_task = current;
set_current_state(TASK_INTERRUPTIBLE);
schedule();
func1();
/* The rest of the code */
```

Now, let's take a look at what is happening in there. In the first statement, we store a reference to this process' task structure. current, which really is a macro, gives a pointer to the executing process' task_structure. set_current_state changes the state of the currently executing process from TASK_RUNNING to TASK_INTERRUPTIBLE. In this case, as mentioned above, the schedule() function simply should schedule another process. But that happens only if the state of the task is TASK_RUNNING.

When the schedule() function is called with the state as TASK_INTERRUPTIBLE or TASK_UNINTERRUPTIBLE, an additional step is performed: the currently executing process is moved off the run queue before another process is scheduled. The effect of this is the executing process goes to sleep, as it no longer is on the run queue. Hence, it never is scheduled by the scheduler. And, that is how a process can sleep.

Now let's wake it up. Given a reference to a task structure, the process could be woken up by calling:

```
wake_up_process(sleeping_task);
```

As you might have guessed, this sets the task state to TASK_RUNNING and puts the task back on the run queue. Of course, the process runs only when the scheduler looks at it the next time around.

So now you know the simplest way of sleeping and waking in the kernel.

Interruptible and Uninterruptible Sleep

A process can sleep in two different modes, interruptible and uninterruptible. In an interruptible sleep, the process could be woken up for processing of signals. In an uninterruptible sleep, the process could not be woken up other than by issuing an explicit wake_up. Interruptible sleep is the preferred way of sleeping, unless there is a situation in which signals cannot be handled at all, such as device I/O.

Lost Wake-Up Problem

Almost always, processes go to sleep after checking some condition. The lost wake-up problem arises out of a race condition that occurs while a process goes to conditional sleep. It is a classic problem in operating systems.

Consider two processes, A and B. Process A is processing from a list, consumer, while the process B is adding to this list, producer. When the list is empty, process A sleeps. Process B wakes A up when it appends anything to the list. The code looks like this:

```
Process A:
1  spin_lock(&list_lock);
2  if(list_empty(&list_head)) {
3       spin_unlock(&list_lock);
4       set_current_state(TASK_INTERRUPTIBLE);
5       schedule();
6       spin_lock(&list_lock);
7  }
8
9  /* Rest of the code ... */
10 spin_unlock(&list_lock);

Process B:
100  spin_lock(&list_lock);
101  list_add_tail(&list_head, new_node);
102  spin_unlock(&list_lock);
103  wake_up_process(processa_task);
```

There is one problem with this situation. It may happen that after process A executes line 3 but before it executes line 4, process B is scheduled on another processor. In this timeslice, process B executes all its instructions, 100 through 103.

Thus, it performs a wake-up on process A, which has not yet gone to sleep. Now, process A, wrongly assuming that it safely has performed the check for list_empty, sets the state to TASK_INTERRUPTIBLE and goes to sleep.

Thus, a wake up from process B is lost. This is known as the lost wake-up problem. Process A sleeps, even though there are nodes available on the list.

This problem could be avoided by restructuring the code for process A in the following manner:

```
Process A:

1  set_current_state(TASK_INTERRUPTIBLE);
2  spin_lock(&list_lock);
3  if(list_empty(&list_head)) {
4          spin_unlock(&list_lock);
5          schedule();
6          spin_lock(&list_lock);
7  }
8  set_current_state(TASK_RUNNING);
9
10 /* Rest of the code ... */
11 spin_unlock(&list_lock);
```

This code avoids the lost wake-up problem. How? We have changed our current state to TASK_INTERRUPTIBLE, before we test the condition. So, what has changed? The change is that whenever a wake_up_process is called for a process whose state is TASK_INTERRUPTIBLE or TASK_UNINTERRUPTIBLE, and the process has not yet called schedule(), the state of the process is changed back to TASK_RUNNING.

Thus, in the above example, even if a wake-up is delivered by process B at any point after the check for list_empty is made, the state of A automatically is changed to TASK_RUNNING. Hence, the call to schedule() does not put process A to sleep; it merely schedules it out for a while, as discussed earlier. Thus, the wake-up no longer is lost.

Here is a code snippet of a real-life example from the Linux kernel (linux-2.6.11/kernel/sched.c: 4254):

```
4253  /* Wait for kthread_stop */
4254  set_current_state(TASK_INTERRUPTIBLE);
4255  while (!kthread_should_stop()) {
4256           schedule();
4257           set_current_state(TASK_INTERRUPTIBLE);
4258  }
4259  __set_current_state(TASK_RUNNING);
4260 return 0;
```

This code belongs to the migration_thread. The thread cannot exit until the kthread_should_stop() function returns 1. The thread sleeps while waiting for the function to return 0.

As can be seen from the code, the check for the kthread_should_stop condition is made only after the state is TASK_INTERRUPTIBLE. Hence, the wake-up received after the condition check but before the call to schedule() function is not lost.

Wait Queues

Wait queues are a higher-level mechanism used to put processes to sleep and wake them up. In most instances, you use wait queues. They are needed when more than one process wants to sleep on the occurrence of one or more than one event.

A wait queue for an event is a list of nodes. Each node points to a process waiting for that event. An individual node in this list is called a wait queue entry. Processes that want to sleep while the event occurs add themselves to this list before going to sleep. On the occurrence of the event, one or more processes on the list are woken up. Upon waking up, the processes remove themselves from the list.

A wait queue could be defined and initialised in the following manner:

```
wait_queue_head_t my_event;
init_waitqueue_head(&my_event);
```

The same effect could be achieved by using this macro:

```
DECLARE_WAIT_QUEUE_HEAD(my_event);
```

Any process that wants to wait on my_event could use either of the following options:

1. wait_event(&my_event, (event_present == 1));

2. wait_event_interruptible(&my_event, (event_present == 1));

The interruptible version 2 of the options above puts the process to an interruptible sleep, whereas the other (option 1) puts the process into an uninterruptible sleep.

In most instances, a process goes to sleep only after checking some condition for the availability of the resource. To facilitate that, both these functions take an expression as the second argument. The process goes to sleep only if the expression evaluates to false. Care is taken to avoid the lost wake-up problem.

Old kernel versions used the functions sleep_on() and interruptible_sleep_on(), but those two functions can introduce bad race conditions and should not be used.

Let's now take a look at some of the calls for waking up process sleeping on a wait queue:

1. wake_up(&my_event);: wakes up only one process from the wait queue.

2. wake_up_all(&my_event);: wakes up all the processes on the wait queue.

3. wake_up_interruptible(&my_event);: wakes up only one process from the wait queue that is in interruptible sleep.

Wait Queues: Putting It Together

Let us look at a real-life example of how wait queues are used. smbiod is the I/O thread that performs I/O operations for the SMB filesystem. Here is a code snippet for the smbiod thread (linux-2.6.11/fs/smbfs/smbiod.c: 291):

```
291 static int smbiod(void *unused)
292 {
293      daemonize("smbiod");
```

```
294
295     allow_signal(SIGKILL);
296
297     VERBOSE("SMB Kernel thread starting "
                "(%d)...\n", current->pid);
298
299     for (;;) {
300             struct smb_sb_info *server;
301             struct list_head *pos, *n;
302
303             /* FIXME: Use poll? */
304             wait_event_interruptible(smbiod_wait,
305                     test_bit(SMBIOD_DATA_READY,
                                 &smbiod_flags));
...
...             /* Some processing */
312
313             clear_bit(SMBIOD_DATA_READY,
                          &smbiod_flags);
314
...             /* Code to perform the requested I/O */
...
...
337     }
338
339     VERBOSE("SMB Kernel thread exiting (%d)...\n",
                current->pid);
340     module_put_and_exit(0);
341 }
342
```

As is clear from the code, smbiod is a thread that runs in a continuous loop as it processes I/O requests. When there are no I/O requests to process, the thread goes to sleep on the wait queue smbiod_wait. This is achieved by calling wait_event_interruptible (line 304). This call causes the smbiod to sleep only if the DATA_READY bit is set. As mentioned earlier, wait_event_interruptible takes care to avoid the lost wake-up problem.

Now, when a process wants to get some I/O done, it sets the DATA_READY bit in the smbiod_flags and wakes up the smbiod thread to perform I/O. This can be seen in the following code snippet (linux-2.6.11/fs/smbfs/smbiod.c: 57):

```
57 void smbiod_wake_up(void)
58 {
59     if (smbiod_state == SMBIOD_DEAD)
60         return;
61     set_bit(SMBIOD_DATA_READY, &smbiod_flags);
62     wake_up_interruptible(&smbiod_wait);
63 }
```

wake_up_interruptible wakes up one process that was sleeping on the smbiod_wait waitqueue. The function smb_add_request (linux-2.6.11/fs/smbfs/request.c: 279) calls the smbiod_wake_up function when it adds new requests for processing.

Thundering Herd Problem

Another classical operating system problem arises due to the use of the wake_up_all function. Let us consider a scenario in

which a set of processes are sleeping on a wait queue, wanting to acquire a lock.

Once the process that has acquired the lock is done with it, it releases the lock and wakes up all the processes sleeping on the wait queue. All the processes try to grab the lock. Eventually, only one of these acquires the lock and the rest go back to sleep.

This behavior is not good for performance. If we already know that only one process is going to resume while the rest of the processes go back to sleep again, why wake them up in the first place? It consumes valuable CPU cycles and incurs context-switching overheads. This problem is called the thundering herd problem. That is why using the wake_up_all function should be done carefully, only when you know that it is required. Otherwise, go ahead and use the wake_up function that wakes up only one process at a time.

So, when would the wake_up_all function be used? It is used in scenarios when processes want to take a shared lock on something. For example, processes waiting to read data on a page could all be woken up at the same moment.

Time-Bound Sleep

You frequently may want to delay the execution of your process for a given amount of time. It may be required to allow the hardware to catch up or to carry out an activity after specified time intervals, such as polling a device, flushing data to disk or retransmitting a network request. This can be achieved by the function schedule_timeout(timeout), a variant of schedule(). This function puts the process to sleep until timeout jiffies have elapsed. jiffies is a kernel variable that is incremented for every timer interrupt.

As with schedule(), the state of the process has to be changed to TASK_INTERRUPTIBLE/TASK_UNINTERRUPTIBLE before calling this function. If the process is woken up earlier than timeout jiffies have elapsed, the number of jiffies left is returned; otherwise, zero is returned.

Let us take a look at a real-life example (linux-2.6.11/arch/i386/kernel/apm.c: 1415):

```
1415  set_current_state(TASK_INTERRUPTIBLE);
1416  for (;;) {
1417      schedule_timeout(APM_CHECK_TIMEOUT);
1418      if (exit_kapmd)
1419          break;
1421       * Ok, check all events, check for idle
....       * (and mark us sleeping so as not to
....       * count towards the load average)..
1423       */
1424      set_current_state(TASK_INTERRUPTIBLE);
1425      apm_event_handler();
1426  }
```

This code belongs to the APM thread. The thread polls the APM BIOS for events at intervals of APM_CHECK_TIMEOUT jiffies. As can be seen from the code, the thread calls schedule_timeout() to sleep for the given duration of time, after which it calls apm_event_handler() to process any events.

You also may use a more convenient API, with which you can specify time in milliseconds and seconds:

1. msleep(time_in_msec);

2. msleep_interruptible(time_in_msec);

3. ssleep(time_in_sec);

msleep(time_in_msec); and msleep_interruptible(time_in_msec); accept the time to sleep in milliseconds, while ssleep(time_in_sec); accepts the time to sleep in seconds. These higher-level routines internally convert the time into jiffies, appropriately change the state of the process and call schedule_timeout(), thus making the process sleep.

I hope that you now have a basic understanding of how processes safely can sleep and wake up in the kernel. To understand the internal working of wait queues and advanced uses, look at the implementations of init_waitqueue_head, as well as variants of wait_event and wake_up.

Acknowledgement

Greg Kroah-Hartman reviewed a draft of this article and contributed valuable suggestions.

Kedar Sovani (www.geocities.com/kedarsovani) works for Kernel Corporation as a kernel developer. His areas of interest include security, filesystems and distributed systems.

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 Quad 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

Wherefore Art Thou, Oh Access Point?

Worse than a fallen soufflé is a wireless card with no Linux driver. Save the dinner for your guests with a few handy utilities. BY MARCEL GAGNÉ

Yes, François. The access point by the fireplace is much better for you to connect to. The ESSID? It's cmfireplace. You'll see it in the list if you scan for it. *Quoi? Mon Dieu*, François, you aren't actually editing a script, are you? Although I admire your desire to get comfortable with the shell, it would be much easier for you to select the scan for the access point, select, click and go.

Ah, I see! The Linux driver for your card doesn't support scanning. Yes, I had a similar problem with mine, but I have a solution. I'll show you in a little while, but time is short and our guests will be here any moment. To the wine cellar, François. Head to the South wing of the cellar and bring back the 1983 Batard Montrachet. *Vite!*

Welcome, *mes amis*, to *Chez Marcel*, home of exceptional Linux fare, fine wines and wonderful guests, of course. Before you arrived, my faithful waiter and I were discussing some problems we have experienced with our wireless cards. My own notebook's wireless card worked fine on my home network, but it worked only so well. The standard Linux Orinoco driver that supported the card didn't allow for things such as scanning. Every month, I went down to the TV studio to record a show, and every month I had to ask which wireless router I could use, because I had to enter the information manually into the network configuration file, ifcfg-eth2.

Of course, the Windows driver for the wireless network card supported these features, and as sometimes happens, manufacturers aren't 100% forthcoming with information or specifications to make full Linux support easy. I truly admire the incredible talent and energy of Linux developers who provide Linux with excellent drivers while working in a vendor black hole. Nevertheless, this lack of information was the impetus for the NdisWrapper Project, which makes it possible to use Windows Ndis drivers by way of a loadable Linux kernel module.

Here's how it works. First, you'll need to get a copy of NdisWrapper from the project's Web site (see the on-line Resources) to guarantee you use the latest version. That said, if you have a recent Linux distribution, check your CDs first. You may find you already have the software. Second, you need the Windows drivers that came with your card, specifically the INF file for that card.

Here's an example from my own Presario notebook, which came with a built-in LanExpress card. Under Linux, connectivity was supported by the Orinoco driver, but as I mentioned, scanning did not work. Because I never actually installed Windows on my notebook—it was there, but I put in a Linux CD before I ever booted the unit, so I never had Windows working on it—I went to the HP Web site and downloaded the driver file in a self-extracting EXE file. I used CrossOver Office to extract the package and then navigated to the folder where the package was located. Using NdisWrapper, I installed the driver by way of its INF file; this must be done as the root user:

```
ndiswrapper -i NetWlan.INF
Installing netwlan
```

Looking at the output above, it doesn't look like a lot has happened. By using the -l option, we can find out what drivers have been loaded and the status of those drivers:

```
ndiswrapper -l
Installed ndis drivers:
netwlan driver present, hardware present
```

The next step is to load the driver into the running kernel, which is done by loading NdisWrapper itself:

```
modprobe ndiswrapper
```

The net result of this can be seen by looking at the output of the dmesg command:

```
ndiswrapper version 1.2rc1 loaded (preempt=no,smp=no)
ndiswrapper: driver netwlan (LAN-Express,01/18/2002,1.07.29.20118) loaded
ACPI: PCI interrupt 0000:00:09.0[A] -> GSI 10 (level, low) -> IRQ 10
ndiswrapper: using irq 10
wlan0: ndiswrapper ethernet device 00:02:8a:a9:e6:eb using driver netwlan,
configuration file 1260:3873.5.conf
ndiswrapper (set_auth_mode:584): setting auth mode failed (C0010015)
wlan0: encryption modes supported: WEP
```

Excellent, we now have the Windows driver loaded into our Linux system and are ready to go. To have all this happen magically at boot time, I added the steps, minus the dmesg, to my rc.local file.

Now, you could get a list of wireless access points near you by using the `iwlist` command with the scan option. Assuming a wireless interface at eth2, the command would be `iwlist eth2 scan`. I then could use the iwconfig command to attach to my network of choice, assign an IP address and so on. However, it also would be nice for the whole desktop experience to have a graphical alternative, one that could scan for networks, report on the quality of the signal and then offer a means of connecting to the service you choose. The notebook, after all, also is a desktop tool.

One of the best such tools I've found is Pawel Nawrocki's Wireless Assistant. This is a great-looking little program that lets you scan for available wireless networks and then connect to them with a single click. Each identified network is identified by factors such as link quality and encryption. The appli-

Figure 1. Wireless Assistant is a tool to scan for and attach to wireless networks.

cation can be configured to handle WEP keys automatically, ignore various types of networks (ad hoc or encrypted), automatically run a script upon connection and more. Figure 1 shows the application in action.

The Wireless Assistant Web site (see Resources) has source code available as well as binary packages for several distributions. By the way, I must mention right now that the SourceForge site has only code available. If you want to track discussions on the package, the KDE-Apps.org (see Resources) site is the better place to look for that information. Building the application from source is a classic extract-and-build five-step:

```
tar -xjvf wlassistant-0.3.9.tar.bz2
cd wlassistant-0.3.9
./configure --prefix=/usr
make
su -c "make install"
```

The actual program name is wlassistant. When the package first starts, it automatically checks for your active network device. If it doesn't, click the Detect button. If you still are having problems, it is possible that the path to the wireless tools hasn't been set properly. Click the Configure button and a configuration dialog appears. In the left-hand sidebar, categories of options are listed while the actual changes take place in the larger right-hand window. Click the Paths button to confirm the pathnames to the wireless tools commands (Figure 2). You can either enter them manually or click the Detect button.

While you are in the configuration dialog, take some time to examine the other options at your disposal. When you are happy with the settings, click OK to return to the main Wireless Assistant window. If you haven't already done so, click Rescan to locate your available access points (Figure 1). As you can see from the image, a handful of networks are available for me to choose from. The display also shows whether an access point uses WEP encryption; this is always a good idea unless you actually want to provide open access to whomever comes your way.

Click the entry of your choice and a box appears so you can enter the root password (Figure 3). Your connection now is established. That's all there is to it.

When I was visiting clients and tying into a variety of networks, I created a little script that copied ifcfg-ethX and network back and forth, depending on the site I was visiting that day. It worked, but it wasn't the most elegant solution. In the world of wireless connections, this hasn't changed. If you are moving from access point to access point, office to office and then back home, you're going to want some kind of help in maintaining all those different profiles. This is true whether you are dealing with wireless or one of those old-fashioned wired connections, *non?*

This is the idea behind Per Johansson's netGo (Figure 4). netGo is a great little application that lets you create network profiles of all kinds and then switch between them with a single click. When not in use, the application docks into your system tray. The program itself is a Qt application, but it works equally well with KDE or GNOME and others.

To get your copy, head to the netGo Web site (see Resources). If you choose to build from source, this is your basic extract-and-build five-step, so nothing too scary here. To run the application, run the command `netgo`. You are asked to provide the root password at this point so that you can make network address changes later.

At first, the main window doesn't contain any profiles. To start the process, click the Add profile button and a new window appears (Figure 5). At the top, enter the profile name, for example, HomeLAN or CoffeeShop, and then choose an interface card; many notebooks have a built-in 10/100 Ethernet card in addition to the wireless card. For connections that require a static IP address, fill in the information in each of the fields for IP address, Netmask and so on.

When you are done, click the OK button to save your profile. If the connection you are setting up is wireless, click on the More options button. There, you can enter the network mode—ad hoc, managed, none—the ESSID and the WEP key. Notice also the Custom script field. This provides a means of automatically executing a series of commands, such as a custom firewall script, when bringing up the interface. Click the Back button to return to the

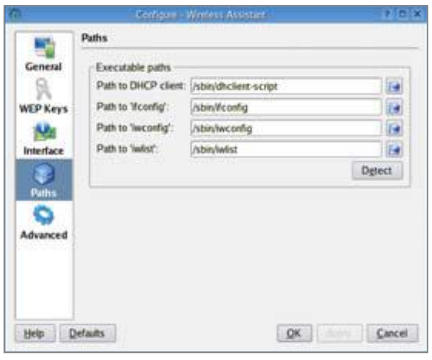


Figure 2. Use Wireless Assistant's configuration dialog to set the pathnames to the wireless tools.

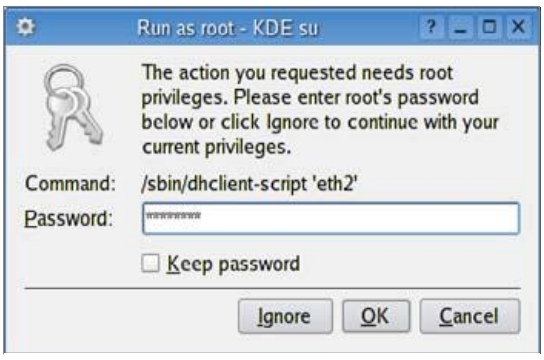


Figure 3. Before switching your network settings, you need to provide the root password.

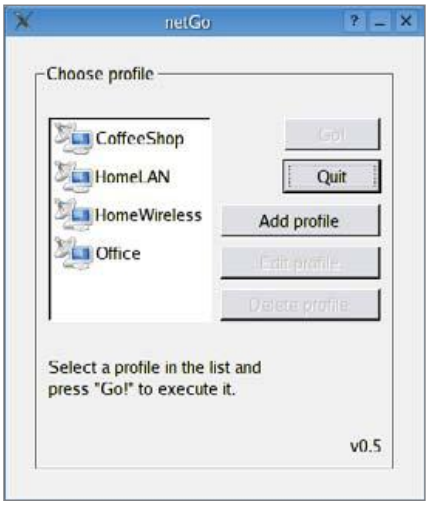


Figure 4. netGo makes it easy to set up and maintain a number of network profiles.

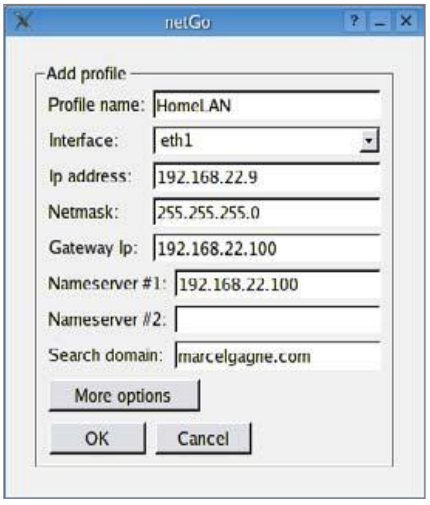


Figure 5. Entering connection details for a netGo profile.

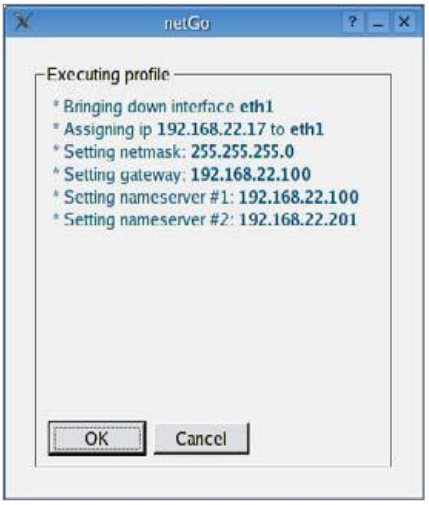


Figure 6. netGo reports back on the changes that were made as the profile executes.

main configuration screen.

Continue adding as many profiles as you need in the same manner. To activate a profile and apply your network settings, click on the profile name and then click the button labeled Go! to apply the changes. A status window then appears displaying the new settings (Figure 6).

The only real drawback I've found to netGo is it doesn't currently provide for taking down your second interface, so this still needs to be done manually if needed. But according to Per's Web site, this feature is in the works as I write this column.

It would appear, *mes amis*, that closing time has arrived, if that clock on the wall is to be believed. Nevertheless, I'm sure that François won't mind if we keep the doors open a little longer; time enough to refill your glasses one last time. We even can bring out a little of that double-butter Brie to accompany the wine. As we all are running wireless tonight, perhaps everyone can carry their notebooks outside to the patio where we can enjoy the evening before we all head home. Please 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/8398.

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 and a past Top-40 disc jockey. He writes science fiction and fantasy and folds a mean Origami T-Rex. He can be reached via 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.

64

The Intel® Xeon™ processor now provides high availability for your 64-bit applications and clusters.

The Intel® Xeon™ processor now works harder for your business than ever. With innovative features that enable power-saving options, flexible memory, I/O and storage configuration. And, of course, continued support for all your existing 32-bit applications.

How can clusters featuring the Intel Xeon processor serve you?
intel.com/go/xeon

Rackable Systems
1933 Milmont Drive
Milpitas, CA 95035
408-240-8300
www.rackable.com/xeon

Verari Systems, Inc.
9449 Carroll Park Drive
San Diego, CA 92121
858-874-3800
www.verari.com/VB1205_blade_server.asp

Ciara Technologies
9300 Trans Canada Highway
Montreal, Québec H4S 1K5
1-866-7VX-RACK
www.vxrack.com

Managing SSH for Scripts and cron Jobs

Anything that you can do from a shell command, you can do remotely with SSH. Here's how to set up keys for effective, secure remote tasks from cron jobs and scripts. BY JOHN OUELLETTE

Using insecure protocols leaves your data and connected machines vulnerable to attack. Remote server management requirements demand that security be given a top priority. This article explains my process for using OpenSSH in unattended scripts and cron jobs.

Most readers are familiar with the secure shell (SSH) protocol, which creates a secure tunnel for commands, data and passwords to travel across the network. Recently, my workplace was faced with some challenges in securely setting up scripts through cron. We use SSH because it resolves the major problems with rsh; rsh sends clear text over the network and has weak host-based authentication. But our challenge became how to deal with password prompts when using SSH in unattended jobs. For example, we run `df -k`, `top` and `swapon -s` to get remote server statistics and alert our team if problems exist.

If you also still are using rsh, the SSH client, ssh, makes a perfect replacement in scripts. Modifications typically are minor. For example:

```
for host in $servers
do
rsh $host df -k
done
```

simply becomes:

```
for host in $servers
do
ssh $host df -k
done
```

SSH supports a wide range of authentication systems, the most common being Kerberos, Rhosts, Public-key and Password. Because we didn't have a Kerberos infrastructure in place, our readily available options to solve this problem were to echo the password in the script, use Rhosts, use ssh-agent or use public keys.

Our first options presented some challenges and weaknesses. First, simply echoing the password in a script is not a simple task, as SSH does not read from standard input. To make it do so would require advanced scripting techniques. More important, you would need to put your password either in the script itself or in a file on the filesystem. Although you could set proper permissions, getting access to the password would be a relatively easy task for a determined intruder. It could be as simple as restoring data from a backup or even viewing the password on-screen. This method was not an option for us.

Second, we considered host-based authentication, which is how users executing the rsh command are granted access. Because users are granted or denied access based on the host they are logging in from, no password is needed. This solution may work in some situations where security concerns are light, but the ability to pretend to be another host, to IP spoof and to disrupt DNS does exist. Also, due to the fact that once a host has been impersonated successfully, all users have been compromised on the remote host, we decided we needed something more secure.

Our third option was to use ssh-agent. Before we discuss this option here, though, we need to cover public keys and their use. Instead of using a plain-text password, SSH has the ability to use public key cryptography. This means that when a client connects to a server, it has a conversation with the server and proves its identity based on advanced mathematical computations.

Let's walk through the setup to generate a set of public and private SSH keys to allow a user named scripts to log in from hostA to hostB, assuming the user exists on both hosts:

1) Generate the keys:

```
[scripts@hostA]$ssh-keygen  -t dsa
Generating public/private dsa key pair.

Enter file in which to save the key
(/home/scripts/.ssh/id_dsa):
Created directory '/home/scripts/.ssh'.
Enter passphrase (empty for no passphrase): XXXX
Enter same passphrase again: XXXX
Your identification has been saved in
/home/scripts/.ssh/id_dsa.
Your public key has been saved in
/home/scripts/.ssh/id_dsa.pub.
The key fingerprint is:
41:03:aa:dc:cc:b9:39:50:65:bc:ee:7b:36:d2:64:7a
scripts@hostA
```

2) Copy public key to hostB from hostA:

```
scp /home/scripts/.ssh/id_dsa.pub \
hostB:/home/scripts/.ssh/authorized_keys
```

scp is an encrypted replacement for rcp and simply copies files in a secured manner.

The authorized_keys file is a file that contains the public identities, or public keys, of users who can log in to your account by using public key authentication. All users maintain their own authorized_keys file, which typically lives in the hidden .ssh directory in a user's home directory. Users also may configure security restrictions to public keys here as well, which we review below.

The authorized_keys file is not created when you first run ssh-keygen to create your public and private keys. As a best practice, we recommend permissions of 600 for this file.

At this point, userA should be able to log in to hostB without a password using public key technology. Now, of course, we still have the same problem with echoing the passphrase into a script. As I mentioned, SSH does not take input from standard input, so this represents the same scripting challenge as before. To eliminate the need to retype your password continually, SSH comes with ssh-agent. You use ssh-agent as follows, in combination with ssh-add:

```
[scripts@hostA scripts]$ ssh-agent bash
[scripts@hostA .ssh]$ ssh-add  id_dsa
Identity added: id_dsa (id_dsa)
```

We pass our shell to ssh-agent, and it inherits the keys we add with ssh-add. Now we need only type our passphrase once and we can use our key default key, id_dsa.pub, to be authenticated. An important note about using multiple keys in an interactive session with SSH is how you need to call SSH. For example, if you have created three private keys—your default key, id_dsa, and two other keys called backup and monitor to use for different tasks—you simply would call SSH with the -i parameter. This is done to make sure you're using your new key while logging in to the remote SSH server:

```
[scripts@hostA]$ssh -i backupkey hostB
```

The question to answer here is, "Do you want to manage keys or user accounts?"

When you are using ssh-agent, you may believe you simply type in `ssh -i backup` to use your backup identity. This is not quite the case, though, as the ssh-agent typically uses the key that is on the top of its key list. To get a listing of all the keys you have loaded in ssh-agent, run `ssh-add -l` for a listing of fingerprints of all identities currently loaded in the agent:

```
[scripts@hostA scripts]$ssh-add -l
1024 df:ab:8e:d7:e4:bd:35:f6:b3:2e:76:6b:dd:71:2f:fe monitor (DSA)
1024 4e:4c:00:ba:1e:5d:60:08:f2:b8:2e:d4:59:1e:ff:2f id_dsa (DSA)
1024 0a:72:24:9e:0a:cd:e2:e4:5f:93:cb:ac:66:78:03:f6 backup (DSA)
```

Because ssh-agent typically favors the key listed first, it favors the monitor key. To be able to use the backup key, you need to unset the shell variable SSH_AUTH_SOCK and then point SSH to the identity you want to use, as follows:

```
[scripts@hostA scripts]env -u SSH_AUTH_SOCK \
  ssh -i backup  hostB
```

After doing this, you will be using the proper key as intended.

Using ssh-agent is, of course, a great time saver for interactive use. When used in scripts, however, a human still needs to type in the passphrase at least once when the machine boots. This ends up being the best we can achieve with ssh-agent, even with scripts to automate most of the procedure. For more information on that topic, refer to *SSH, The Secure Shell: The Definitive Guide*. Ultimately, the ssh-agent option also did not meet our needs in deploying secure batch jobs, as our goal was to automate the jobs totally.

That left us with the option of using public keys without a password. The remainder of this article focuses on that setup, how to secure it further and some options to consider when using this setup. In any environment, thorough planning and review of security polices should occur before deploying or modifying security configurations.

The first method in securing our setup is to use the `from=` directive in the authorized_keys file. The syntax looks like this:

```
from="host1,host2" KEY
```

What this says is allow only users from host1 or host2 and authenticate them against the public key matching KEY. For example, to restrict logins from only hostA and hostB for our user scripts, the authorized_keys would look like this:

```
from="hostA,hostB" ssh-dss AAAAB..Aqbcw= scripts@hostA
```

This is by no means a foolproof restriction. As I mentioned before, it is possible to pretend to be another host and spoof an IP. But this restriction adds a layer of security and increases the effort needed to compromise our host. Notice that I intentionally shortened the key, which is quite long, due to space constraints. Be aware that the from= syntax is sensitive to short and long DNS names. HostA is not the same as HostA.somewhere.

Our second line of defense in securing our script setup is to use the `command =""` directive, also specified in the authorized_keys file. The syntax for this looks like:

```
command ="command",  KEY
```

This tells SSH to run *command* and then exit. It effectively limits your ability to run commands on the remote server. As you might expect, you can combine both of these in your authorized_keys file; simply make sure you separate the options by a comma:

```
from="hostA,hostB",command="/bin/df -k"  ssh-dss AAAAB3N...Aqbcw= userA@hostB
```

Now, should someone compromise this user and key, the worst that can be done is retrieving a listing of disk space on the remote host. In fact, this is the only command you can run with this key. In order to run multiple commands securely, you have a few options. First, consider calling a script instead of command. For example, run `top`, `df -k` and `hostname` from a shell script named myscript.sh and set `command="/path/to/myscript.sh"`. Second, if you need to run multiple commands at different times during the day to

the same host, you could create another key for your user. This time, use the -f option to specify a file other than the default:

```
[scripts@hostA]$ssh-keygen  -t dsa -f backupkey
Generating public/private dsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in backupkey.
Your public key has been saved in backupkey.pub.
The key fingerprint is:
14:ac:c5:5f:65:69:2f:8d:cf:0a:70:9e:5c:4e:c7:84 scripts@hostA
```

You would copy the contents of the new public key, backupkey.pub, to the authorized_keys file of the users on the hosts you want them to access, the same as the default key. Be sure to set the new `command=""` for your new key and the new command you want to run.

Now, you would use the -i parameter to make sure you're using your new key while logging in to the remote SSH server:

```
[scripts@hostA]$ssh -i backupkey hostB
```

TIP To add another public key on a remote host without overwriting your original authorized_keys file, you can run this command:

```
$cat ~/.ssh/newkey.pub | ssh -l user host "cat >> .ssh/authorized_keys"
```

Finally, you could create a separate user for this task. For example, you could create one user to monitor disks and one to automate backups. Each configuration has its advantages and drawbacks. The question to answer here is, "Do you want to manage keys or user accounts?" I prefer to have different keys and make a note of them with a comment.

One piece of the SSH key we have not considered yet is the comment field. The default comment for userA's public key created on HOSTA is userA@HOSTA. Basically, everything after the key is ignored as a comment. So to keep track of keys and what they are used for, I make a comment in the remote server's authorized_keys file. For example:

```
ssh-dss AAAAB3NzaC1kc3MAAA.....Jw= scripts@hostC-Key used for disk monitoring
```

Our third line of defense is the ability to limit the traffic we forward. We have three main options here that merit discussion. First, the no-port-forwarding option means what is says. When this key logs on, the ability to forward TCP/IP ports is denied. Forwarding ports is a great way to bypass firewalls; hence, the account used to run scripts should be given the ability to run only the commands necessary. The ability to forward TCP/IP ports is a potential security problem, so we restrict it.

Second, no-X11-forwarding tells the SSH process not to forward any X11 connections upon login. Any attempt to do so returns an error. We see that this is simply another way for an intruder to exploit our hosts, so we disable it. Again, we try to lock down what the account that logs in can do, but we also permit it to perform its job.

Third, no-agent-forwarding in the authorized_keys denies this key the ability to forward its ssh-agent and stored keys to another host. This reduces complexity and also takes away another avenue for a potential intruder to trespass.

The final option in the authorized_keys file we want to use is the no-pty option, which says not to allocate a pseudo-tty when logging in. Non-interactive commands continue to work using the associated key; however, you can no longer issue commands through an interactive session. Should an intruder gain access to your private key and somehow circumvent the other options, this option effectively ensures that he or she cannot issue interactive commands to do any damage.

With the above options in place, we have a reasonably restricted key that still can perform its job. Our final authorized_keys file looks like this:

```
from="hostA,hostB",command="/bin/myscript.sh",no-port-forwarding,no-X11-forwarding,no-agent-forwarding,no-pty
ssh-dss AAAAB3....o9M9qz4xqGCqGXoJw= scripts@hostA
```

Before we finish our discussion on options, let's look at two more that are not related directly to security. When running SSH in scripts, we use the -q and -o "BatchMode=yes" command-line options. The -q stands for quiet mode. The man page for sshd sums this up nicely: "Nothing is sent to the system log. Normally the beginning, authentication, and termination of each connection is logged." This is useful for suppressing warnings otherwise interpreted as command output. The -o "BatchMode yes" makes sure SSH does not prompt the user. So our script changes a little more:

```
for host in $servers
do
ssh -q -o "BatchMode=yes" $host df -k
done
```

Because we are specifying an option on the command line, we are certain the options will not be overridden as they take precedence. Typically, the global client configs are looked at first, usually /etc/ssh_config; then the local client configs, usually ~/.ssh/config; and finally the command line. As several versions and variations of SSH are available, always consult the man page for correct locations and syntax.

Ensuring proper options are set for each particular key and using a layered security approach goes a long way in making your servers less vulnerable to attacks. Setting the least privileges possible reduces the potential damage done during a successful attack. Using these methods, your data and networks become more secure and still run efficiently.

Resources for this article: www.linuxjournal.com/article/8400.

John Ouellette is a system administrator with nine years' experience in NT and UNIX. He believes the command line is king and loves chicken parmigiana. He can be reached at john_ouellette@yahoo.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 Pro Optimum 1U Server X9489

Dual Intel® Xeon™ Processors 3GHz
- Intel® E7320 Chipset Server Board
- 1GB ECC Registered DDRII 400 SDRAM (Up to 8GB)
- 2 x Seagate® 400GB SATA 7200RPM Hard Drive (Total 800GB Storage)
- 2 x 1 SATA Hot-Swappable Drive Bays
- Slim 24x CD-ROM & 1.44MB Floppy Drive
- On Board 2 Channel SATA RAID Controller (RAID 0, 1 Support)
- 2 x Intel® 10/100/1000 Gigabit Network Controllers
- 1U Rackmount Chassis w/420W Power Supply
- **3-Year Limited Warranty + First Year Onsite Service**

$2,399

ZT Pro Optimum 2U Server X9490

Dual Intel® Xeon™ Processors 3 GHz
- Intel® E7320 Chipset Server Board
- 1GB ECC Registered DDR333 SDRAM (Up to 8GB)
- 2 x Seagate® 250GB SATA 7200RPM Hard Drive (Total 500GB Storage)
- 6 x 1" SATA Hot-Swappable Drive Bays
- Slim 24x CD-ROM & 1.44MB Floppy Drive
- On Board 2 Channel SATA RAID Controller (RAID 0, 1, JBOD Support)
- Intel® 10/100/1000 Gigabit Network Controller
- 2U Rackmount Chassis w/550W Power Supply
- **3-Year Limited Warranty + First Year Onsite Service**

$1,999

ZT Pro Optimum 3U Server X9491

Dual Intel® Xeon™ Processors 3 GHz
- Intel® E7520 Chipset Server Board
- 1GB ECC Registered DDRII 400 SDRAM (Up to 16GB)
- 8 x Seagate® 300GB SATA 7200RPM Hard Drive (Total 2.4TB Storage)
- 8 x 1" SATA Hot-Swappable Drive Bays
- Slim 24x CD-ROM & 1.44MB Floppy Drive
- 8 Channel RAID Controllers (RAID 0, 1, 5, 10 , 50, JBOD Support)
- 2 x Intel® 10/100/1000 Gigabit Network Controllers
- 3U Rackmount Chassis w/550W Power Supply
- **3-Year Limited Warranty + First Year Onsite Service**

$3,999

1. OEM Computer Manufacturer

- 3 year warranty with lifetime tech support
- Reseller and volume pricing available.

New Accounts Receive Free Gift

- Personal attention (Dedicated Technical Sales Team)
- 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** (866-984-7687)

Purchaser is responsible for all freight costs on all returns of merchandise. Full credit will not be given for incomplete or damaged returns. Absolutely no refunds for merchandise returned after 30 days. All prices and configurations are subject to change without notice and obligation. Opened software is non-refundable. All returns have to be accompanied with an RMA number and must be in re-sellable condition including all original packaging. System's picture may include some equipments and/or accessories, which are not standard features. Not responsible for errors in typography and/or photography.

First Beowulf Cluster in Space

When a satellite's image-gathering power exceeds the bandwidth available to transmit the images, a Linux cluster right on the satellite helps decide which images to send back to Earth.

BY IAN MCLOUGHLIN, TIMO BRETSCHNEIDER AND BHARATH RAMESH

When President Eisenhower proposed the Open Skies Policy at the 1955 summit meeting to the Soviet delegation in Geneva, it was an unsuccessful move to legitimate the US' plans to launch the U2 spy plane a month later. Five decades later, Open Skies became a reality with the launch of Singapore's X-Sat. What could complement Open Skies better than open source? And it doesn't take a genius to understand that, when reliability is all important, the transparent and open nature of Linux source is an invaluable aid.

At the outset of the X-Sat Project, which focused on developing Singapore's first satellite, arguments were made for Linux but roundly were rejected. At that time, Linux was an esoteric outsider for embedded systems use and hadn't penetrated the consciousness of decision-makers in the area of space developments. Furthermore, Singapore generally is not known for risk taking, and truly there is something to be said in favour of this attitude where satellites are concerned. By contrast, VxWorks has excellent space heritage, although this is no guarantee for success.

Although the satellite's main computer runs VxWorks, Linux powers the data processing computer. This actually is a loosely coupled cluster called the parallel processing unit (PPU), and it is the first distributed example for Linux in space. The concept is to run satellite image-processing applications directly in space after a straightforward re-compilation and uploading procedure from ground-based Linux development platforms. Let's compare the main on-board computer (OBC) and the PPU (Table 1).

The OBC is so expensive because it utilises costly radiation-hardened components, whereas the PPU uses mostly commercial-off-the-shelf (COTS) components. Traditionally, reliability in space is ensured by using the most reliable individual components to survive the hostile environment. But the PPU embodies the relatively new concept—at least in space—of reliability through redundancy. Although each single component of the PPU is less reliable in space than are the OBC components, there are 20 copies of each PPU processor, so even if one after another fails, something still remains. The design almost eliminates single-point failures, where a single component failure could take out the entire system or multiple components. On top of this, the PPU is characterised through graceful degradation from a fully working 20-processor system down to a single processor. So, good design ensures that the probability of a single PPU processor still functioning at the end of design life matches the probability that the OBC still is functioning at the same time. And even with only a single surviving processor, it still thrashes the OBC.

X-Sat

X-Sat is a 100kg micro-satellite, roughly an 80cm cuboid, as shown in the CAD model of Figure 1. The satellite, an educational project at Nanyang Technological University, Singapore, carries three payloads: a 10m resolution multispectral (colour) camera for obtaining images in the Singaporean region, a radio link for an Australian-distributed sensor network and, most notably, the PPU. From the outset of the project, X-Sat has been an open satellite with details publicly available, and what better reason to use open-source software?

Table 1. X-Sat has both an on-board computer (OBC) and a parallel processing unit (PPU).

	OBC	PPU
Processors	2 x ERC32	20 x SA 1110
Configuration	Cold-redundant standby	Whatever you want
Peak performance [MIPS]	20	4,000
Total memory [MB]	8	1,280
Size [cm^3]	3,125	3,125
Power consumption [Watt]	Approx. 2	25
Hardware cost [US$]	50,000	3,500
Processing cost [US$ / MIPS]	2,500	0.88
Processing volume [cm^3 / MIPS]	156.25	0.78
Processing power [mW / MIPS]	50	6.25
Operating system	VxWorks	Linux
Costs for OS	A few thousand dollars	Free

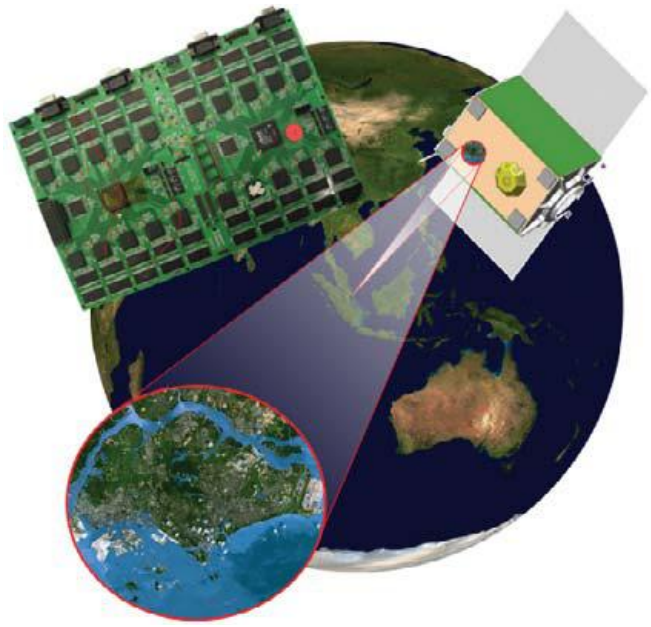


Figure 1. X-Sat is about 80cm in size and carries a colour camera, radio link and a Linux cluster.

Communication uses S-band with 4kb/s uplink and 500kb/s downlink as well as a unidirectional 50Mb/s downlink over X-band for image dumps. However, the X-band needs a dedicated 13m dish antenna for reception, and it works only when the satellite is over Singapore. In its intended sun-synchronous 685km orbit, this occurs for only a few minutes every day and leads to a major rationale for the PPU. Assuming a conservative duty cycle of 10% per orbit, the camera can generate 81GB of data per day, but only 12% of this can be downlinked. And with a three-year design lifetime and multimillion-dollar cost, each picture works out to be rather expensive.

If we want maximum value per picture and we have to throw away 88% of the images, which ones do we select? Anyone who has been to Singapore should remember the overcast skies. It turns out that 90% of satellite pictures over Singapore show only clouds and haze. Although this may excite meteorologists, it's a waste of money for us. We'd get value from only 10% of the 12% of pictures we download—a 1.2% success rate! So if there's any way of deciding before downloading whether a picture is obscured by clouds or even a way of cutting out the cloudy bits and downloading the rest, then this is valuable. Well, you guessed it, such applications do exist. They run on Linux Beowulf clusters, require many MIPS and happen to be a perfect fit for our PPU.

PPU Design

The PPU consists of two anti-fuse Actel field programmable gate arrays (FPGAs), known to be more radiation-resistant than other solutions from Xilinx or Altera. Each FPGA hosts ten processing nodes (PNs), each with a 206MHz StrongARM processor and 64MB of SDRAM. Individual FPGAs are connected to three Atmel 4MB serial Flash chips containing a bootloader, the OS kernel and filesystem images, which include selected image-processing applications. Of course, programs can be added dynamically while the satellite is in space, as though it were a regular Linux cluster.

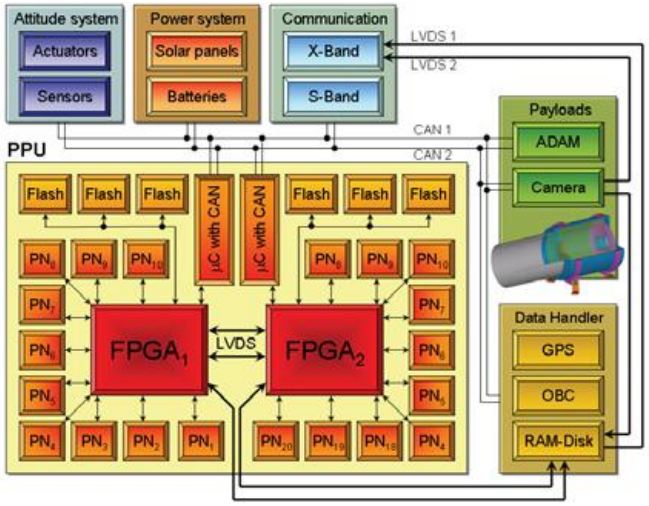


Figure 2. The cluster is based on two FPGAs, each connected to ten 206MHz StrongARM processors.

The PPU is connected to the rest of the satellite by fairly slow quad-redundant controller area network (CAN) links and two fast (200Mb/s) low-voltage differential signalling (LVDS) links for image data from the on-board camera. Figure 2 shows an overview of the hardware architecture. Most interesting to

mention is that the PPU also can take over satellite control from the OBC. In fact, this is one of the experiments that is supposed to validate that software and hardware COTS components can fulfill mission-crucial tasks.

Internally, the PPU resembles a cluster-based computing system with the FPGAs providing the interconnection network. In fact, these hubs themselves can offer image-processing capabilities. The cluster concept means we can sacrifice PNs to failure and yet carry on system operation regardless. It also gives each PN sufficient autonomy to run multiple algorithms simultaneously. As each FPGA has its own independent communication links, PPU operation can continue even with severe failures, such as destruction of an entire FPGA.

A parallel bus interfaces each PN to an FPGA. Given that ten PNs communicate with one FPGA, hardware I/O pins on the FPGA become a limitation. It is impossible to support ten full 32-bit buses. A 16-bit data bus is the next logical choice but results in a halving of the effective bus bandwidth. However, considerable effort was made to ensure that this slimmed interface operates efficiently, and it has resulted in a novel 17-bit data bus, which is discussed later. From the PN perspective, the FPGA is memory mapped into address space using an addressable window concept to reduce parallel bus requirements.

Booting

Booting of the PNs is sequential to reduce peak power on start up and consists of three stages. First, the StrongARM operates in the 16-bit access mode, executing code directly from the lowest address window of the FPGA. Although this translates into half-bandwidth memory access, the small size of the ARM assembler bootloader (512 bytes) makes it acceptable. The bootloader is a tiny ARM assembler coded routine of less than 5,122 bytes that executes directly out of the FPGA's lowest address window. It initialises the StrongARM, sets up SDRAM and then loads the second stage from serial Flash. The second stage retrieves the kernel and ramdisk from serial Flash, executes the kernel decompressor and boots Linux. Finally, the third bootloader stage consists of bzImage, which decompresses itself into the appropriate memory location and then executes the kernel, which then decompresses its ext2 initrd ramdisk.

The 17-Bit Bus Interface and Protocol

All communication to the PN occurs through FPGA. A kernel device driver plus a user-space library provide a standard interface API for Linux applications. The low-level driver maintains two filesystem character devices that implement interrupt handling and software receive/transmit buffers. In order to keep the driver efficient and simple, kernel preemption was disabled. The driver also periodically writes to a watchdog register in the FPGA, as a heartbeat signal, causing reboot on timeout.

In the PPU, writes from the PN to FPGA fall into two classes: control and message data. Message data normally is destined for another PN, whereas control data directs some action on the part of either the FPGA or PN. Similarly, reads of the FPGA by a PN also fall into these categories.

In case of message data writes from a PN to another PN via the FPGA, each item of data destined for a particular PN must be addressed. Either addressing information is part of each and every word transferred or it's set in advance. In the PPU, message paths are set in advance under PN control for efficiency reasons, assuming most transfers are large—which is without doubt the case for satellite images. But a 16-bit interface conveying 16-bit data messages must have a mechanism to distinguish between data and address packets. This could be achieved by writing these to separate address registers in the FPGA.

The situation for reading the FPGA is trickier, however. A 16-bit bus requires two reads for each message: one read to determine message type and/or length and another to convey the actual message. But because our messages have variable length, there is an immediate problem concerning the timing of such messages. The reason is an interrupt signal is used to indicate a 16-bit value waiting to be read and the PNs are under obligation to respond. So for long messages, the FPGA would read a sequence of 16-bit half-words. But it has no obvious means of distinguishing a 16-bit control word inserted into this sequence. We could prefix all half-words with a type header, but that would mean two reads per half-word of message—halving the bandwidth.

Our solution is a 17-bit bus with the StrongARMs operating in 32-bit access mode. Both raw data and commands share the physical link as half-words with their type differentiated by the state of a special 17th bit that indicates to the PN whether an incoming item is data or a control message. Most important, it does this without requiring any extra read cycles or extra bus bandwidth.

This approach wouldn't be of interest if the driver module couldn't take direct advantage of the load-store nature of the ARM and the fact that all instructions are conditional. The former implies that the 32-bit read from the 17-bit bus is loaded into an internal register before being moved to memory. The latter implies that if the 17th bit of the interface is wired to the most significant data bit, D31, rather than the more obvious choice of D16, it can be used to affect the zero flag. As a result, the data destination to one of two internal memory buffers can be controlled through conditional data moves. This is extremely efficient compared with an inefficient conditional branch that most other processors utilise. The following assembler code provides an example with r0, r1, r2 and r3 being the registers for the address of the FPGA data transfer, the control word buffer, the message word buffer and the type, respectively. In summary, the code for the optimised solution is 33% faster and uses one register less:

SCENARIO I—Default Read Method:

```
...
LDR   r4, [r0]          ; Load FPGA value
LDR   r5, [r3]          ; Load type register
TST   r5, #0x80000000   ; Check for D31
STREQ r4, [r1]          ; Z flag set (control)
STRNE r4, [r2]          ; Z flag not set (message)
B     _repeat           ; Loop again
```

SCENARIO II—Optimised Read Method:

```
...
LDR   r4, [r0]          ; Load FPGA value
STRMI r4, [r1]          ; N flag set (control)
STRPL r4, [r2]          ; N flag not set (message)
B     _repeat           ; Loop again
...
```

Software Error Detection and Correction

All satellites are subject to cosmic ray irradiation. Besides aging effects, the most frequent consequence is random bit-flip errors in SDRAM and the CPU. Left unchecked, these ultimately lead to large-scale data corruption. From a software perspective, the result of every calculation as well as every word in memory is suspect. It goes without saying that a mechanism to detect and correct such errors must be implemented in any space-based system.

Typical solutions for error detection and correction (EDAC) protection involves custom hardware checksum generators. But for our 20-processor PPU, a checksum solution is overly complex, so we utilise a less efficient but simpler multilayer software approach. An EDAC process periodically is scheduled in kernel space to provide error protection. A second EDAC process allows the two to be cross-checked for redundancy.

Process integrity verification in our system is performed for crucial code between scheduled runs of the EDAC processes. In addition, input and output values of protected software procedures are monitored. If unexpected values are detected, the system employs either a clean-up approach, retries the calculation, outputs a previously calculated value or uses the most significant bit-flip correction scheme. Which to use is configured on a per-function basis in a parametric verification table, which again is EDAC-protected.

C code is protected through a single header file and linkable library code. The function entry definition is inserted manually:

```
#define EDAC_CHECK \ entry_check_edac( __func__);
```

GCC resolves __func__ at compile time with the string name of the function being entered. The on-demand EDAC process is invoked prior to the function executed. A return re-definition is similar:

```
#define return(z)
return_check_edac( __func__,\ __builtin_return_address, z);
return(z);
```

The developer inserts this into the code, as in the sample program given below:

```
int calc(int x, int y) {
    EDAC_CHECK .....
    return(z); }
```

Using this, a malfunctioning program can't cause too much damage. But even if the kernel is involved, a loss of heartbeat triggers a reboot. To minimise the impact on other tasks, it's preferable that only one user application should operate on each node concurrently—but of course, this is at the user's discretion.

Applications and Algorithms

So, what is the PPU supposed to do after launch? Even though the hardware costs are almost insignificant with respect to the overall satellite budget, with a launch price of approximately 10,000 US$/kg, each gramme has to be strongly justified. Right now, the most essential PPU task is image compression

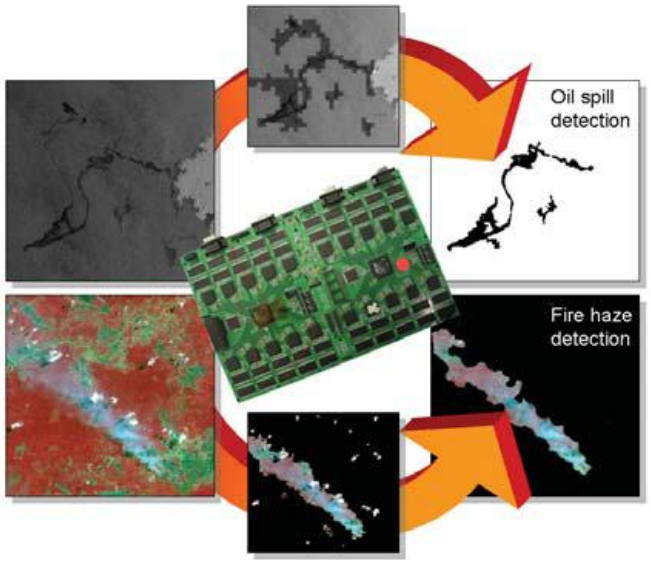


Figure 3. The processing power of the PPU makes it possible to detect oil spills and fires on the satellite without having to download all the raw data.

using a content-driven JPEG2000 scheme. But the major advantage of the PPU is its "standing watch" capability, in which the camera continuously monitors the Earth with image data evaluated and discarded immediately if it's not valuable. In case of detecting valuable information, which is under software control, the obtained scene is kept for subsequent transmission. But even more important, X-Sat can transmit the results of its findings instantaneously to mobile terminals on the ground—each the size and price of a conventional transistor radio. The implications of such a concept are understood easily if, for example, such a system was in place when the earthquake northwest of Sumatra, Indonesia, created a tsunami wave killing more than 285,000 people on Boxing Day 2004.

Currently, two specific applications are supported: the detection of oil spills and haze observation originating from man-made and natural fires. Both make use of the additional processing power available through the FPGAs to pre-process image data streamed into the individual processors. The images in Figure 3 are examples from a simulated acquisition campaign over a complete daylight period of one day's orbits. The raw data from a 10% duty cycle covers an area of approximately 3 million km^2. If only 0.001% of this data showed oil spills, this would be equivalent to 62 catastrophic *Prestige* oil spills. With a fully functional PPU, the processing time for simultaneous execution of both disaster-detection tasks is 25% of the total daily orbit time. In contrast, however, it allows the evaluation of the entire data instead of only a small subset on the ground.

Launch into a New Space Era

From an engineer's perspective, X-Sat and its PPU couldn't succeed without Linux: almost all current application developments in the area of remote sensing use Linux, as do most modern cluster systems. So, sometime in early 2007, if you tilt your head back at the right time, you might be caught on camera, processed and downloaded, thanks to Linux.

Resources for this article: www.linuxjournal.com/article/8399.

Ian's been using Linux since about 1856 and weaned his kids at the penguin's electronic teat. His interests include satellites and signal processing, and his career objective is to lose his job and become a missionary in China.

Although Timo didn't try to wean his daughter on the penguin, he uses Linux for most of his number-crunching problems on Beowulf clusters and in the future even more extensively in space. Timo's research focus is remote sensing and various image-processing problems, well, unless he's gone traveling.

Bharath designs high-performance systems for Hewlett-Packard. Not being very high-performance himself, he relies on the pet monkey under his desk to come up with hardware designs. Occasionally, it also writes articles for magazines with penguins on their cover.

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

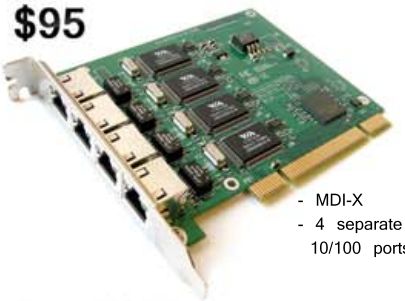
- Nstreme wireless polling protocol
  - no decrease in speed over long distances (as seen with the 802.11 ack packet bottleneck)
  - polling improves speed and eliminates contention for access to the wireless bandwidth
  - access point control over Nstreme clients tx data to optimize use of the wireless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

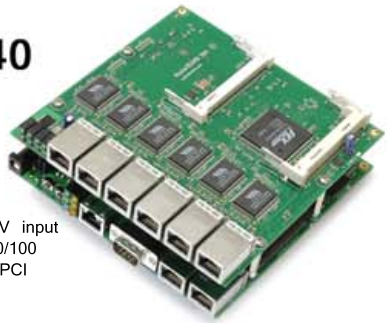
Eight ports Four ports

RouterBOARD 11/14/18

Multi radio tower !

MiniPCI to PCI adapters for multi radio system. Tested with sixteen radios in one Router/AP.

$240



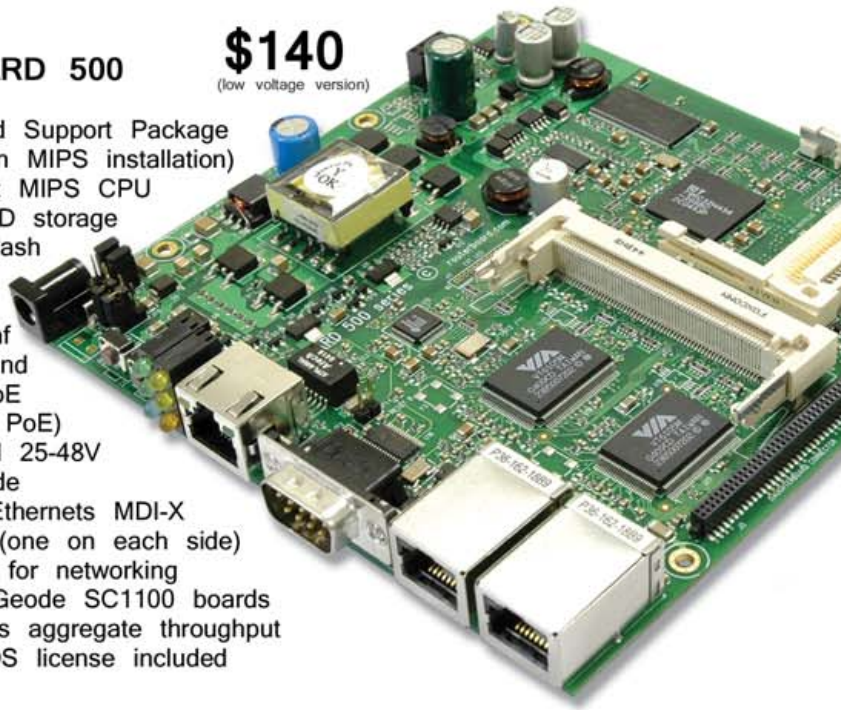
- PoE
- 10-56V input
- 9x 10/100
- 6x mPCI

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

Independent Identity

Can a free market in identity systems emerge from a confusing array of vendor-specific silos? Doc sees hope from an unlikely source. **BY DOC SEARLS**

I've been delivering the closing keynote at Digital ID World (DIDW) since October 2002, when the show began. I've given it five times so far. From the beginning I've had a case to make. Here it is:

1. In a truly free marketplace, vendors in a category compete openly for a customer's business, based on information the customer supplies at his or her discretion to any or all of them. Customer data isn't isolated in vendor silos, and customers aren't forced to go from one silo to another and interact separately with each vendor's customer relationship management system, its lame marketing agendas and its locked-up data.

2. Customers won't have full power in any marketplace unless they control their own data, including data about their relationships with vendors, and selectively make private data available to vendors, with explicit permissions regarding each vendor's use of it. Drummond Reed of Cordance and Identity Commons calls this CoRM, or company relationship management.

3. Personal identity control and CoRM are powers customers have not experienced since the Industrial Age began and they first became "consumers". As fully empowered customers, they will blow the CRM blinders off vendors, release a wave of entrepreneurship, differentiation and innovation and make markets grow in all directions. This won't be a market revolution but, rather, the dawn of a real market—where demand and supply have equal power.

4. CoRM requires identity services that do not yet exist. Those services also serve other purposes—SSO, or single sign-on, authentication, security, privacy and so on. But whatever those services may be, they share a point of origin: the individual. This invites adjectives such as grass-roots, lightweight, user-centric, next-generation, human origin, bottom-up and distributed. I like the literal meaning of the word independent. Every speech I make on this subject is a literal declaration of customer independence.

5. Independent Identity is a human quality more than an organizational one. Establishing and ubiquitizing Independent Identity therefore needs to be a grass-roots movement that grows on open standards and with support from the Open Source community. For that reason, it only makes sense for Independent Identity to grow from the bottom up rather than from the top down, from individuals rather than from organizations.

Andre Durand of Ping Identity Corp., on whose advisory board I now serve, inspired this thinking in 2001, when he was putting the company together. I wrote about Andre and Ping in "Identity from the Inside Out", in the May 2002 issue of *Linux Journal*. In it I describe Andre's three-tier view of identity:

> At the center is tier one: that's your core identity alone. You're in charge of it. Outside that is tier two. This is the identity issued to you by the government, by retailers, by airlines, by insurance companies, by credit-card companies. Every piece of plastic in your wallet is a tier-two identity. Tier three is the cloud of highly presumptuous identity information held by direct marketers and others who hope you may be the one consumer in 50 who responds to a promotional message.

When Andre laid this out for me the first time, I knew instantly that we could liberate markets simply by developing the means for individuals to assert their sovereign identities. It was a moment of revelation—like Neo had, just before he said "It's about choice" to the architect of The Matrix. In fact, *The Matrix*, one of my favorite movies ever, suddenly made sense as an allegory for the false world invented by marketing, where consumers live in blissful ignorance of their role as batteries whose energy maintains that world.

By the middle of 2004, however, I had begun to lose faith. Even Ping Identity was caught up with the rest of the supply side in a conversation about federation. In an interview last year, Eric Norlin, Director of Marketing at Ping Identity, said federation "leaves the distributed environment as it is, but seeks to let the end users link together those pieces and still have control over their privacy, what gets shared, and how." He said the Liberty Alliance spec, for example, "is purposefully designed to be opaque. [It] tries to accomplish one thing...to allow one end user to link one account to another account. But in order to protect the privacy of the end user...neither side of that account—either Company A or Company B—would know who the end user is that is being passed between them."

I've summarized this as "large companies having safe sex using customer data".

Federation is a big company concern, essentially a silo-to-silo "solution"—so far. Liberty Alliance isn't the only consortium devoted to federation. The other big one is WS-I, where WS stands for Web services. WS-I was founded by Microsoft, IBM and Verisign, partly in response to the Liberty Alliance, which was founded by Sun, partly in response to Microsoft's Passport. Anyway, the more I heard about federation, the more depressed I became about the prospects for Independent Identity.

Then, during LinuxWorld Expo in August 2004, I met Kaliya ("Identity Woman") Hamlin at a San Francisco Giants baseball game. She told me she worked with Identity Commons, a grass-roots organization with a market-opening plan that leveraged some standards, notably XRI and XDI, and

encouraged new ones, such as i-Names. A series of conversations with various Identity Commons people armed me with plenty of fodder for my keynote at DIDW in October 2004. I wrote up the story behind that keynote in "What's Your i-Name?", my column for the January 2005 issue of *Linux Journal*. In that piece, I made something of a bet about Independent Identity: "We're not going to get that from the big vendors, for the same reason we didn't get Linux from big computer makers: big suppliers in any category have trouble pioneering anything that's good for everybody and not only for them."

I was wrong about that. When the conversation started to heat up after DIDW, the Neo role was being played by a character with the unlikely title of "Architect", working inside the most unlikely company of all: Microsoft. Kim Cameron is his name, and his architecture is the Identity Metasystem. Note that I don't say "Microsoft's Identity Metasystem". That's because Kim and Microsoft are going out of their way to be nonproprietary about it. They know they can't force an identity system on the world. They tried that already with Passport and failed miserably.

Kim came to Microsoft by way of acquisition, when Microsoft bought Zoomit, a Toronto company specializing in the "metadirectory" field. I came to know Kim through Craig Burton of The Burton Group, back in the early 1990s. Craig and his organization saw non-interoperable directories as a problem that could be solved only by a system that was intentionally inclusive and respectful toward all directories and their countless differences. Craig labeled the required system a metadirectory and called for vendors to fill the market's need for one. Kim and Zoomit stepped forward and developed a metadirectory product, along with a lot of deep thinking about directories and related issues, including security and identity. During that time I became good friends with Kim, an occasional consultant to Zoomit.

Microsoft acquired Zoomit at about the time it was becoming clear that Passport was failing. At Microsoft Kim took a leading role in re-thinking the company's approach to identity. It became clear to him that taking a "meta" approach would open a whole new marketplace—for Microsoft and everybody else:

> At first I didn't think it was possible. But while I was scrounging around one day I ran across this one protocol that was so simple I could hardly believe it. I saw how it could work like a conduit for the simple exchange of tokens and how it could bridge many different identity systems.

Craig Burton took a natural interest in Kim's identity work at Microsoft. Before DIDW/2004, Craig began telling me that Kim's architecture had the potential to seed and support—in an open way—the kind of grass-roots movement I'm looking for.

So I made sure Kim got to connect with the other grass-roots advocates attending the conference: Drummond Reed, Fen LaBalme, Mark LeMaitre, Kaliya Hamlin, Jan Hauser and Owen Davis of Identity Commons and also, for several on that

list, of Seattle-based Cordance; Dick Hardt of Sxip; Marc Canter, currently of Broadband Mechanics but perhaps best known as a founder of Macromind, which later became Macromedia; Simon Grice of MiDentity; and Phil Windley of Brigham Young University, also the former CIO of Utah and the author of a new book from O'Reilly on digital identity. All are open-source and open standards advocates, and all consider open-source involvement essential to the success of whatever it was Kim and Microsoft had in the works, which still wasn't clear at the time.

An informal group began to form. Meetings followed in Seattle and other places. And right after DIDW/2004, Kim also began posting his Seven Laws of Identity. He did this on an installment plan to give everybody time to talk about each one before moving on to the next. His First Law appeared on November 16, 2004, and his Seventh Law appeared in March 2005. In summary form, here they are:

1. User Control and Consent: digital identity systems must reveal information identifying a user only with the user's consent.

2. Limited Disclosure for Limited Use: the solution that discloses the least identifying information and best limits its use is the most stable, long-term solution.

3. The Law of Fewest Parties: digital identity systems must limit disclosure of identifying information to parties having a necessary and justifiable place in a given identity relationship.

4. Directed Identity: a universal identity metasystem must support both "omnidirectional" identifiers for use by public entities and "unidirectional" identifiers for private entities, thus facilitating discovery while preventing unnecessary release of correlation handles.

5. Pluralism of Operators and Technologies: a universal identity metasystem must channel and enable the interworking of multiple identity technologies run by multiple identity providers.

6. Human Integration: a unifying identity metasystem must define the human user as a component integrated through protected and unambiguous human-machine communications.

7. Consistent Experience across Contexts: a unifying identity metasystem must provide a simple consistent experience while enabling separation of contexts through multiple operators and technologies.

The Laws serve two purposes. The first is to guide conversation and development in an emerging marketplace. The second is to guide conversation and development inside Microsoft. Kim says he often finds himself saying stuff like, "No, that would break the Fifth Law" or "That misses the point of the Seventh Law."

Microsoft is and will remain an issue. In October and November 2004, Marc Canter and others wondered out loud about how we could ever trust the company or its partners, such as Sun Microsystems. Craig Burton acknowledged the problem in a December 4th blog posting:

> Marc contends that people don't want to get locked into standards owned by Microsoft or Sun. Kim wants to look beyond the past and create a "big bang" of distributed computing that would eclipse the petty Microsoft bashing. In Marc's defense, Microsoft is an unabashed bully. The leaders of Microsoft—Bill Gates and Steve Ballmer—lead the bully behavior. I have personal experience of this behavior from both of them. Microsoft doesn't and isn't going to play fair anytime in the future—in general.

Perspective: Craig fought Microsoft when he was at Novell in the 1980s—and usually won. One high-level Microsoft executive told me years later that Craig was perhaps the only leader at a competing company that truly understood how to compete and win against Microsoft. Craig continues:

> I say "in general" for a reason. There are good people with vision and integrity at Microsoft. Kim Cameron is one of those people. You can't go wrong working with Kim. Further, it is just ludicrous to think that Microsoft is of one voice and has an overarching plan with which to rule everything always and forever....I have said before to Kim that working at Microsoft is like working inside ten tornadoes. I am changing that to a thousand tornadoes. Each tornado (or hailstorm if you like) has its own path, thinking and objective. They seldom cross paths and are too busy dealing with the issues at hand to even talk to each other. Microsoft is a thousand tornadoes deep.
>
> Microsoft bashing aside, when two people like Marc and Kim get together and collaborate, expect good things to happen that go beyond the history of giants—even the giant of all time—Microsoft.

The most significant push-back we received was from Dave Winer, whom Kim considered to be an important role model. Dave had success at launching standards, including XML-RPC, SOAP—which Dave and his company, Userland, co-developed with Microsoft—and RSS, and he made it all happen without the clout of a large organization behind him. RSS was an especially interesting case, because it established a service, syndication, that is rapidly moving toward ubiquity.

On his blog, Scripting News, Dave wrote:

> Doc Searls...offered RSS and podcasting as examples of technologies that were simple, therefore successful, and suggests that identity, if it were to be approached the same way, might have similar success. Bzzzt. Wrong. RSS was not easy, it was hard, for exactly the same reasons identity is hard. Too many cooks spoil the broth. Two ways to do identity is one too many.
>
> Politics spoiled identity and would have spoiled RSS had the major players not converged on RSS 2.0. The difference this time was that there was a Switzerland, me, to guide RSS through its gauntlet, and I clearly wasn't in bed with any of the major publishers or vendors. The Harvard connection didn't hurt because it's a highly respected university that hadn't been involved in tech standards. Had identity had that kind of championship it might not be the mess it is today.

I didn't think identity was spoiled in the least. But I also realized something. Identity needs a Dave Winer: an independent developer and free-range technologist who tirelessly advocates something that can work for everybody.

On December 30th, Steve Gillmor called. He was hunting up guests for the "Gillmor Gang" scheduled the next day: New Year's Eve. I suggested bringing in the "Identity Gang"—or as many as wanted to come on the show. Steve said yes, and I sent out an e-mail to nine people, including Dave Winer, whose experience, example and skepticism I thought were essential. To my surprise, nearly all of them, including Dave, agreed and took part in the show.

The conversation was all over the place. But it served as a public meeting to which many could listen and link. The "Identity Gang" show was energizing. Starting on January 1, 2005, I saw Independent Identity issues being discussed—not only in blogs and podcasts but in trade pubs and in halls at conferences—with considerable optimism. In the past, discussions always seemed to go sideways into energy-draining digressions on privacy, crypto and other muddy subjects, such as "Microsoft Sucks".

Kim bled off a lot of steam by publishing his Fifth Law on New Year's Day. Craig Burton wrote this about it:

> The Law of Pluralism is contrary to the laws of customer control.
>
> Let's be clear: the Law of Pluralism requires operating system independence—by definition. This means the Microsoft Identity Architect is calling for a system that is not necessarily Windows-centric by design. This—of course—is the only way such a system can really work. But consider the implications.
>
> A cross-platform identity metasystem is sun-spot hot and—with the other laws being discussed here—changes everything.

The Identity Metasystem looked to each of us—so far as we could understand it, which wasn't enough—as though it had the makings of a Net-native system that would embrace and accommodate everybody's separate efforts. It helped especially that Drummond Reed and the Identity Commons people already were figuring out ways of working with the Identity Metasystem.

Kim also demonstrated InfoCards, a Microsoft identity implementation that can work within the Metasystem. Everybody was eager to think about or find other implementations—so nobody would confuse the InfoCard implementation with the Identity Metasystem architecture. At one point I asked if it was possible for InfoCards, or anything Microsoft was doing in the Identity Metasystem framework, to plug in to Firefox. Kim said, "Yes, of course." I invited Mitchell Baker to the next meeting we held, and she and Kim agreed that it ought to be workable.

The Identity Gang has grown since then. DIDW gave us a room to use on May 8th, the day before the show started. About 40 people met all day around a large table. Kim

explained the Identity Metasystem in more detail than we had heard before. There was a lot of discussion, including plenty of skepticism, but more than enough positive energy to keep everybody interested.

Since then, Kim and his team have published a whitepaper titled "Microsoft's Vision for an Identity Metasystem". The paper outlines the architecture in some detail. Here are the key paragraphs:

> The encapsulating protocol used for claims transformation is WS-Trust. Negotiations are conducted using WS-MetadataExchange and WS-SecurityPolicy. These protocols enable building a technology-neutral identity metasystem and form the "backplane" of the identity metasystem. Like other Web services protocols, they also allow new kinds of identities and technologies to be incorporated and utilized as they are developed and adopted by the industry.
>
> To foster the interoperability necessary for broad adoption, the specifications for WS-* are published and are freely available, have been or will be submitted to open standards bodies and allow implementations to be developed royalty-free.
>
> Deployments of existing identity technologies can be leveraged in the metasystem by implementing support for the three WS-* protocols above. Examples of technologies that could be utilized by way of the metasystem include LDAP claims schemas; X.509, which is used in Smartcards; Kerberos, which is used in Active Directory and some UNIX environments; and SAML, a standard used in inter-corporate federation scenarios.

Figure 1 shows the graphic illustration.

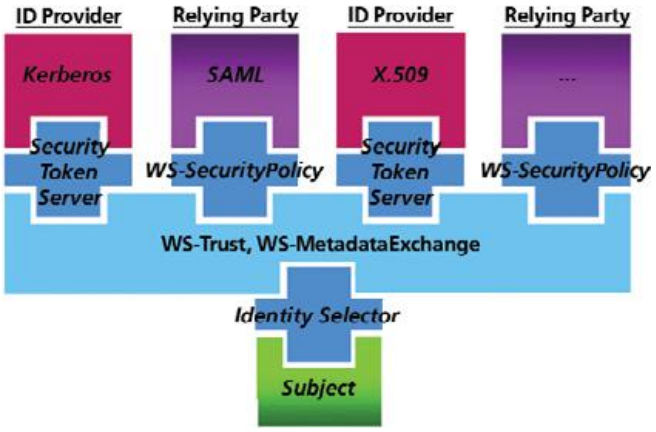


Figure 1. The WS-* Architecture

The Independent Identity would be one of the ID providers, part of the metasystem. It could be hosted locally, on a Linux box in the clouds, on a phone—wherever someone wanted to put one.

I've told Kim that he and Microsoft need to do more before my constituency—the Linux and Open Source development communities—takes a serious interest in the Identity Metasystem. I said, "If you don't have an open-source license or if you start talking about IP Frameworks, my readers will leave the room." The term IP Frameworks was used by somebody from another part of Microsoft, in respect to the WS-* standards process. Kim replied:

> It's essential to have the Open Source community involved. And I wish we were already at some point in the future when we have some of these things cleared up. But we're talking about slow processes here. The standards process is incredibly complicated. You get a bunch of companies together, and the process is like nuclear disarmament. The only way to get a royalty-free standard is to negotiate IP in such a way that nobody can sue because it's a standoff. What an "IP Framework" means to me—though I am not a lawyer and can's speak for Microsoft on IP issues—is that everybody puts their IP in and agrees not to charge royalties. Ironically, the biggest concern may not be each company's IP, but submarine patents that can surface later and screw up everybody.

As for open-source licensing, Kim has said encouraging things to me privately, but for now there's nothing to report. I'm hoping, for everybody's sake, there will be an accepted open-source license or licenses in place by the time you read this.

Meanwhile, everybody in the open-source camp seems to be scratching their own itches, each in their own simplifying way.

Open ID says, "This is a distributed identity system, but one that's actually distributed and doesn't entirely crumble if one company turns evil or goes out of business." Its identities are URL-based.

LID's goal is to "empower individuals to keep control over and manage their digital identities, using VCards, FOAF and GPG. It is very REST-ful and fully decentralized. It is also a great mechanism to add accountability to REST-based Web services, even if no (human) digital identities are involved." Johannes Ernst, one of LID's creators, says "it's the simplest scheme there is, so simple that, just like a few other folks have done already, you can probably implement it yourself over the weekend and add five new profiles to it that we didn't even think of." There are several LAMP and J2EE implementations available for download.

Sxip is more ambitious and has several parts. Sxip.net (Sxip Network) is "a simple, secure and open digital identity network that offers a user-centric and decentralized approach to identity management. This key piece of Internet infrastructure, based on a network architecture similar to DNS, can be used by people to develop their own identity management solutions, enabling distinct and portable Internet identities." Sxip.com (Sxip Identity) "provides identity management solutions that leverage the Sxip Network and drive Identity 2.0 infrastructure. Sxip empowers individuals to create and manage their on-line digital identities and enables enterprises to instantly provision and manage their users." Sxip.org provides developer resources, including a Subversion code repository.

Moebius "builds on the success of e-mail-based identity systems by adding a few important but incremental improvements while laying the foundation for more advanced identity systems in the future. Moebius is more convenient to use, easier to deploy and safer for all concerned, without requiring expensive investments in new infrastructure or adoption of

untried, centralized identity systems."

There are other open-source and Linux-related efforts around identity, and I'm sure this article will flush out those who feel offended by their exclusion. I invite them to join the Identity Gang or whatever name it uses by the time you read this.

Things are moving so fast, in so many directions at once and with so many individuals involved, that it's impossible to cover the subject completely. This piece sets a record of frustration for me, personally. I've been working on it since January, and I've rewritten it countless times. I was going through a series of rewrites when I missed my deadline last month. And I almost decided to make it a *Linux Journal* Web site piece several days ago, when I still wasn't sure if Kim and his Identity Metasystem would take the heat from interested skeptics like Julian Bond and Dave Smith—or from those of us (a percentage that rounds to "everybody") who see a lock-in agenda behind every Microsoft move. So I posted some tough public questions on IT Garage. Kim has met every challenge with grace, humor and backbone. I'm sure he needs all three to make this project fly inside Microsoft.

I also want to thank Microsoft for giving Kim a way to apply his genius. If this thing flies, high fives are due all around.

The best any of us can do is stay true to our own principles and purposes. Kim's are embodied in his "meta" understanding of the world. The man is the best includer I've ever known. Microsoft is lucky in the extreme to have him working there. Mine are embodied in an NEA understanding of the Net—as something Nobody can own, Everybody can use and Anybody can improve.

There is an improvement I want to see, and it's something only Independent Identities can produce. I want anybody to be able to pay for anything on a voluntary basis, because I believe the voluntary ability to pay whatever one wants is at the heart of a free and open marketplace. I also believe we haven't experienced that power since the Industrial Revolution put huge suppliers in charge, even of democratic governments. We certainly haven't had it since the invention of the price tag.

I'm not saying I want to turn every store into a commodities pit where everybody haggles over prices. I'm not saying "Let's get rid of fixed prices." I am saying, let's give consumers the power to be customers. I am saying, let's start by making this work in markets where no prices have yet been set, where sellers and buyers don't yet have the means for discovering what their goods are worth, where—because that mechanism is absent—most of the goods are free, as in beer.

I have two markets in mind: "podsafe" (non-RIAA, Creative Commons-licensed) music and podcasts. I would like to be able to express my willingness to pay for music I like and for podcasts I like and to do that at my discretion, quickly and easily. And, to extend that ability to other services that welcome voluntary payment, such as public radio and TV, churches, charities and so on.

I would like that capability to be built in to my browser, as

a plugin probably, and my RSS aggregator. Later, I'd like to see it in cell phones and other mobile devices.

I would like the Open Source community to step into those markets with me and say, "We have a way that anybody can pay anybody for anything, on their own or mutually agreeable terms." And free has to be an alternative. Free still has to be okay.

You might say I'm talking about a more robust shareware market here. One where suppliers don't beg or cajole, make goods scarce or call those who get goods for free "pirates". I'm talking about making the Net as open and responsible as a farmers market: a place where customers are as unlikely to filch from an artist's site as they are to take an apple from a farmer's cart. And where artists of all kinds still can give away all they like.

Can this be done? I don't know. I can think of a hundred reasons why it can't. I'm sure the rest of you can think of more. Between me writing the last sentence and this one, Johannes Ernst wrote this to me:

> The trouble though, is, that we are miles away from being able to understand what the technical requirements are for such a transformational system, because we haven't thought through the transformational applications that need to be supported.
>
> Yet I feel certain there's a way of doing this, and experimenting with it, and seeing what works and what doesn't, and showing the world how a free and open marketplace can work.

I want to give the old choose-your-silo system a bad case of Innovator's Dilemma. We need something disruptive here. Something simple and new. An invention that mothers necessity.

We won't get it if we get bogged down in long-winded digressions about privacy and crypto and the big awful companies that want to keep their hands—oops, credit and membership cards—in our pockets. Those are legitimate and necessary concerns, but they are secondary to the purpose of establishing methods and protocols and technologies for the assertion of Independent Identity. And for changing the world by saving markets from the producerist mentality that has kept everybody, producers included, in darkness for more than a century.

I also feel certain that forces far more nefarious than Microsoft are hell-bent on putting the Net genie back in the telco and cableco bottles—and turning it into the distribution system for "protected content" they imagined when they made sure the "information superhighway" had asymmetrical driveways to every "consumer's" home.

If we don't want that, we have to show we're customers and not just consumers. And real customers don't just shop in silos.

Resources for this article: www.linuxjournal.com/article/8401.

Doc Searls is senior editor of *Linux Journal*.

servers ▶ DIRECT™

More products, better service, guaranteed.

GO STRAIGHT TO THE SOURCE!

Introducing the Servers Direct Blade System with the power of Intel® Xeon™ Processor

Increased computing power in a smaller footprint and simplified maintenance help you expand your enterprise solution to meet the most intense application demands.

SDB-1100H Servers Direct Blade System

Featuring a Server Direct Server Compute Blade powered by dual ® Xeon™ 800FSB Processors.

Benefit of using Servers Direct Blade System:

- More power, bandwidth, and processing performance to meet the demanding requirements of departmental workloads
- Deliver world-class performance for peak server workloads
- Future 64-bit-enabled applications
- High performance small form factor SCSI hard drives (RAID 1 with 2 HDDs)
- Ethernet I/O for demanding, data-intensive applications
- Ability to easily add hotswappable SCSI hard drives, and additional Ethernet, or Fibre Channel I/O for increased application performance (RAID 1E requires use of the HDDs on the SBX82 as well as the HDDs on the SBESCSI)

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 (Lindenhurst VS) Chipset
- Kingston 512MB DDR333 ECC Reg. RAM (2x256MB)
- 1pc x 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 Xeon Processing Server
SDR-2103T

High-density 2U platform optimized for performance and flexibility; ideal for Web hosting, data center, terminal services and High Performance Computing (HPC)

- Intel Xeon Processor 3.0Ghz with 800FSB 1MB Cache (Dual Processor Option)
- Intel Extended Memory 64 Technology
- 1U Intel Chassis with 700W PFC power supply
- Intel® Server Board SE7520JR2
- Kingston 512MB DDR333 ECC Reg. RAM (2x256MB)
- 6pcs xSeagate 160GB SATA/150 W/ncq 7200rpm 8MB Cache
- Intel SRCS16 6Channel SATA RAID Controller Card
- Integrated ATI Rage XL SVGA PCI video controller
- 2x Intel® PRO/1000 MT Server Network Connections (Intel® 82546GB controller)

$3,499

5U File 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 (Lindenhurst) 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

$4,999

3U Clusterable SATA SAN Nodes
SDR-3303T

Provides a flexible, price and performance advantages to the storage needs of Small to Medium Business (SMB) market segment.

- Intel 3U SAN Storage Enclosure w/700W Redundant Power Supply
- Integrated Intel SE7501HG2 Server Board
- 2x Integrated SAN Intel Xeon 3.06Ghz/533FSB Processor
- Integrated SAN 2x256MB Compact Flash Memory Cards w/SAN Mgt Software
- 3 X Integrated Intel SRCS16 6-Channel SATA RAID Cards
- Kingston 1024MB DDR266 ECC Reg. RAM (2x512MB)
- 16pcs x Western Digital WD2500SD RAID SATA
- Integrated SAN Management Software & Storage System

$13,599

Big business power, small business price tag.

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**

Internet Radio to Podcast with Shell Tools

Combine several standalone programs with some shell "glue" and record your favorite Internet radio show while you sleep. **BY PHIL SALKIE**

It all started because I wanted to listen to "Hour of the Wolf" on WBAI radio—it's a cool science-fiction radio program hosted by Jim Freund that features readings, music, author interviews and good "I was there when..." kind of stories. Unfortunately for me, WBAI broadcasts from Long Island, New York, and is too far away from me to receive well. Plus, the show is on Saturday mornings from 5 to 7AM EST—not a really welcoming timeslot for us working folks.

Then, I discovered that WBAI has streaming MP3 audio on its Web site, which solved the reception problem. That left the Oh-Dark-Hundred problem—I'm normally settling into a deep sleep at that hour. And science-fiction buff or no, I'm not going to be catching Jim live any time soon.

The Search

What I needed was a VCR for Internet radio. Specifically, I wanted to capture the stream and save it to disk as an MP3 file, named with the show name and date. I would need to add the proper MP3 ID tags so I could load it into my Neuros audio player for convenient listening. It also would be awfully nice if I could let RSS-compatible software know that I've captured these files. That way, they would show up in a Firefox live bookmark or could be transferred to an iPod during charging. The ultimate effect would be to create an automatic podcast—a dynamically updated RSS feed with links to saved recordings—by snipping a single show out of an Internet media stream at regular intervals.

So, off I went to Google to search for "mp3 stream recording" and "tivo radio" and so on. I found many packages and Web sites, but nothing seemed quite right. Then, I heard a voice from my past—that of the great Master Foo in Eric S. Raymond's "The Rootless Root", which said to me: "There is more UNIX-nature in one line of shell script than there is in ten thousand lines of C." So, I wondered if I could accomplish the task using the tools already on the system, connected by a simple shell script.

Collecting the Tools

You see, I already could play the stream by using the excellent MPlayer media player software. Due to patent problems, Fedora Core 3 doesn't ship with MP3 support, so I previously had downloaded and built MPlayer from source as part of the process of MP3-enabling my system. On a side note, MPlayer makes extensive use of the specific hardware features of each different CPU type, so it performs much better as a video player if it is built from source on the machine where you plan to use it. The command:

```
mplayer -cache 128 \
-playlist http://www.2600.com/wbai/wbai.m3u
```

served admirably to play the stream through my speakers. All that was left to do was convince MPlayer to save to disk instead. The MPlayer man page revealed -dumpaudio and -dumpfile <filename>, which work together to read the stream and silently save it out to disk, forever and ever. There's no time-out, so it captures until you kill the MPlayer process. Therefore, I wrote this script:

```
#!/bin/bash

mplayer -cache 128 \
  -playlist http://www.2600.com/wbai/wbai.m3u \
  -dumpaudio -dumpfile test.mp3 &
# the & sets the job running in the background

sleep 30s

kill $! # kill the most recently backgrounded job
```

which nicely captured a 30-or-so-second MP3 file to disk. The & character at the end of the `mplayer` command above is critical; it makes MPlayer run as a background task, so the shell script can continue past it to the next command, a timed sleep. Once the sleep is done, the script then kills the last backgrounded task, ending the recording. You may need to adjust the -cache value to suit your Internet connection or even substitute -nocache.

Now that part one was accomplished, I was on to part two—inserting the MP3 ID tags. Back on Google, I found id3v2, a handy little command-line program that adds tags to an MP3 file—and it's already in the Fedora Core distribution! It's amazing, the things that are lurking on your hard drive.

Creating a Podcast

I now had the tools in place to capture and tag my favorite shows. With that in place, I was left with the task of coming up

with some way to make a syndication feed from the stack of files. It turns out that RSS feeds are simple eXtensible Markup Language (XML) files that contain links to the actual data we want to feed, whether that be a Web page or, as in this case, an MP3 file.

Another quick look at Google brought me to the XML::RSS module for Perl. It's a complete set of tools that both can create new RSS files and add entries to existing ones. At this point, I thought I was almost done and put together a nice code example that almost worked. In true project timeline tradition, however, the last 5% of the project turned out to require 95% of the total time.

RSS: Worms in an XML Can

Once I had a script that did all I wanted it to do, I sent it in to *LJ* along with a first version of this article. *LJ* Editor in Chief Don Marti pointed out that I was missing one key component: my program was generating an RSS version 1.0 feed, but all the podcast-aware programs look for a version 2.0 feed—specifically for an XML tag named enclosure. Naturally, I assumed it would be a trivial change to my software, merely switching versions and adding the enclosure tag. I soon learned, however, that the XML::RSS Perl module can write RSS 2.0 but cannot read it. Several sleepless nights ensued, until I determined that Perl tools were available that could read RSS 2.0 but not write it. So, it was time to add some glue.

I started by adding two Perl modules to my system—you can install them (as root) with:

```
perl -e "install XML::RSS,XML::Simple" -MCPAN
```

You probably will be okay with answering any questions it asks with the default. If you haven't used the Comprehensive Perl Archive Network (CPAN) yet, it asks quite a few setup questions, such as choosing several mirror sites that are close to you. Otherwise, it simply asks about a dependency or two; say yes.

After the two modules and their required dependencies are installed, you need to create a new XML file with information about the show you want to capture. The great thing about XML is you can use any text editor to make a file that is readable by both humans and machines, making it easy to create, view, test and modify RSS feed files. Let's start with this skeleton, containing a basic title section:

```
<?xml version="1.0" encoding="UTF-8"?>

<rss version="2.0">

<channel>
<title>Hour of the Wolf</title>
<link>http://www.hourwolf.com</link>
<description>Science Fiction Talk Radio
  with Jim Freund</description>
<generator>WBAI Stream Capture
  using Linux shell tools</generator>
</channel>
</rss>
```

If you never have played with XML before, this is a good time to get your feet wet. A quick look at the file shows data items surrounded by HTML-like tags, where each <something> tag has a corresponding </something> to close the something section. This becomes more confusing later, though, when we add the alternate syntax, which looks like <tagname a="A" b="B" />.

Applying the Glue

Once I had gathered all the tools I needed, I added a few droplets of shell magic to arrive at this simple script:

```
#!/bin/bash
# catchthewolf - capture "Hour of the Wolf"

# For capturing the stream
DATE=`date +%F`  # Save the date as YYYY-MM-DD
YEAR=`date +%Y` # Save just the year as YYYY
FILE=/home/phil/wolf.$DATE.mp3 # Where to save it
STREAM=http://www.2600.com/wbai/wbai.m3u
DURATION=2.1h # enough to catch the show, plus a bit
#DURATION=30s # a quick run, just for testing

# For the RSS syndication
XML="/home/phil/wolfrss.xml" # file for the RSS feed
ITEMS=15  # Maximum items in RSS list
XTITLE="Hour of the Wolf - $DATE Broadcast"
XDATE=`date -R` # Date in RFC 822 format for RSS
```

Streaming Formats

When streaming radio first came out, it often was transmitted in proprietary data formats, making it tough for Linux users to listen. Now most streams are MP3, but there still may be something in a different format that you want to capture, such as BBC Radio's RealPlayer streams—see the on-line Resources for a link. Assuming that it's something MPlayer can handle, we simply can rearrange our process a bit. Tell MPlayer to write audio data to the disk in the form of a WAV file and then encode it using lame for MP3 or oggenc for ogg files. Be aware, though, that lame is not included with Fedora, again due to patent issues.

The audio capture commands then would look like:

```
# Use mplayer to capture the stream
# at $STREAM to the file $FILE
/usr/local/bin/mplayer -really-quiet -cache 500 \
    -ao pcm:file="$FILE.wav" -playlist $STREAM &
# the & turns the capture into a background job

sleep $DURATION  # wait for the show to be over

kill $! # kill the stream capture

# Encode to .ogg, quality 2, and tag the file
oggenc -q 2  -t $TITLE -a $AUTHOR -l $ALBUM \
  -n "1/1" -G "Radio" -R 16000 -o $FILE $FILE.wav

rm $FILE.wav # Remove the raw audio data file
```

followed by the original call to the Perl script. No need to use id3v2 here, as both the lame and oggenc encoders insert tags as part of the encoding process. We wind up with the same result as capturing an MP3 stream directly. But because of the intermediate WAV file's large size, we need much more disk space during the actual capture process. The optional -R 16000 specifies the sample rate of the captured WAV file—this is needed only if MPlayer does not correctly detect the speed of the incoming audio stream and your captured MP3 sounds like whale song or chipmunks. You probably want to comment out the rm command until you're sure the encoding is working the way you want it to and remove the WAV files manually until then.

```
i=\$i;o=\$o;m=\$m # replace "$" in the perl script

# For the id3v2 Tags
AUTHOR="Jim Freund"
ALBUM="WBAI Stream Rip"
TITLE="Hour of the Wolf - $DATE"

# Use mplayer to capture the stream
# at $STREAM to the file $FILE
/usr/local/bin/mplayer -really-quiet -cache 128 \
    -dumpfile $FILE -dumpaudio -playlist $STREAM &
# the & turns the capture into a background job

sleep $DURATION  # wait for the show to be over

kill $! # end the stream capture

# Tag the resulting captured .mp3
id3v2 -a "$AUTHOR" -A "$ALBUM" \
    -t "$TITLE" -y $YEAR -T 1/1 -g 255 \
    --TCON "Radio" $FILE

# Add a new entry in the rss file,
# keep the file to a max of $ITEMS entries,
# and change the file's date to right now.
/usr/bin/perl -e "use XML::RSS; use XML::Simple; \
    $i=XMLin('$XML');$o=$i;bless $o,XML::RSS; \
    $m=$i->{channel}{item};if((ref $m)ne ARRAY) \
    {$o->add_item(%$m);} else \
    {foreach $m (@{$m}) {$o->add_item(%$m);}} \
    $o->channel(lastBuildDate=>'$XDATE', \
    pubDate=>'$XDATE'); \
    $o->add_item(title=>'$XTITLE', \
    link=>$o->{'channel'}{'link'}, \
    pubDate=>'$XDATE', \
    enclosure=>{url=>'file://$FILE', \
    length=>(stat('$FILE'))[7], \
    type=>'audio/mpeg'}, mode=>'insert'); \
    pop(@{$o->{'items'}}) \
    while (@{$o->{'items'}}>$ITEMS); \
    $o->{encoding}='UTF-8'; $o->save('$XML');"

echo "Caught the wolf."
```

This doesn't look too simple, though. Let's dissect this script a bit to see how it all works. Notice the back-ticks (`) around the date commands. They take whatever is enclosed in the `` marks and run it as a command and then replace the entire `whatevercommand` with the output from that command. If I had needed the date only once, I could have written:

```
FILE=wolf.`/bin/date +%F`.mp3
```

or even:

```
/usr/local/bin/mplayer -dumpaudio \
  -dumpfile "wolf.`/bin/date +%F`.mp3" \
```

But because I wanted the date for the filename, the tag and the RSS feed, I stored it in the $DATE shell variable. That

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

Highly SCYLD

www.scyld.com

What Is This Thing Called RSS?

RSS stands for Rich Site Summary.

RSS stands for RTF Site Summary.

RSS stands for Really Simple Syndication.

Everything else about RSS is as confused as its acronym. The idea started out as the ability to read headlines from Web sites without having to download the entire front page. RSS is implemented in eXtensible Markup Language (XML), which makes it easily read and written by both humans and computers. That means the format for the RSS file is standardized—unfortunately, the content is not. There are at least four versions of RSS floating around—0.9, 0.91, 1.0 and 2.0—that have similarities, differences and interoperability issues galore. The basic RSS file contains a title, a publication date and a group of items. Each item has its own title, date and link to the file containing the article content. The variations between versions mean that any software wanting to read or write these files has to be programmed specifically to understand each version—there is not enough backward compatibility to let things simply work.

Even the version numbering is odd—version 2.0 is descended from version 0.91, not version 1.0. Version 1.0 is the most feature-rich and extensible, supporting dynamic definitions of the tag names through links to special machine-readable Web pages. Version 2.0 extends the original concept to allow more complex summaries that include images and music rather than only lines of text; it does so through the use of the enclosure tag. Enclosures work like attachments to e-mail messages. When the RSS-aware program downloads the site summary, it notices the attachments and downloads them too. This extends the concept of a summary to being a list of contents, plus the contents itself—far from the original concept of RSS, but this is becoming its biggest use today.

makes it much easier to change the script around too. I now have several scripts that capture streams, and the only things that have to change are the variable assignments at the top.

Back-ticks are one of the shell's tools that allow us to merge simple commands into powerful assemblies. You can play with this more by using the echo command. Try, for example:

```
echo "wolf.`date +%F`.mp3"
```

to see what the filename would be in that last call to MPlayer.

We use the +%F formatting option to date, because the default date string is full of spaces. Also, my USA locale's date string has / characters in it—not the best thing to try to put inside a filename. Furthermore, the yyyy-mm-dd format means the files sort nicely by date when you list the directory. The RSS feed wants its date in RFC 822 format, so we wind up calling /bin/date three times in all.

Notice also that I'm giving the exact path to some of the executable commands. I do this so that when the script runs as a timed task, it won't have my personal shell's path settings. If you're unsure where a file lives, find it with which:

```
[phil@asylumhouse]$ which date
/bin/date
```

You're safe to leave off /bin and /usr/bin, but any other path should be specified explicitly, as should paths to any executable that exists as different versions in multiple locations.

The call to id3v2 tags the file as track 1 of 1, with proper author, album, title and year entries. The predefined genre number of 255 means Other. The --TCON entry fills in Radio in place of one of the predefined genres on any software that understands version 2 MP3 tags.

Lastly, the one-line Perl script at the end is a compressed version of this:

```
#!/usr/bin/perl

use XML::RSS; use XML::Simple;

$in=XMLin('/home/phil/wolfrss.xml');
$out=$in; # copy the parsed RSS file's tree
bless $out, XML::RSS; # make the copy an XML::RSS

# blessing doesn't copy the items.  Drat!
$item = $in->{channel}{item};
if ((ref $item) ne ARRAY) { # only one item in feed
  $out->add_item(%$item);
} else { # a list of items - foreach the list
  foreach $item (@{$item}) {
    $out->add_item(%$item);
  }
}

# Encoding doesn't transfer either.
$out->{encoding}='UTF-8';

# Date the file so client software knows it changed
$date = `date -R`;
$out->channel( lastBuildDate=>'$date',
     pubDate=>'$date');
```

Scale up performance. Scale down costs.

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

Appro *HyperBlade* Clusters

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

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

Appro **BladeDome** Remote Server Management

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

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

APPRO
HPC Cluster Solutions

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

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

AMD 64 Opteron™

```
# Add our newest captured file
$file = "/home/phil/wolfcaught.mp3";
$out->add_item( title => "Hour of the Wolf",
     link => $out->{'channel'}{'link'},
     pubDate => '$date',
     enclosure => { url=>"file://$file",
       length => (stat($file))[7],
       type => 'audio/mpeg'
     },
     mode => 'insert');

# Don't have more than 15 items in the podcast
while (@{$out->{'items'}} > 15) {
         pop(@{$out->{'items'}};
}

# Write out the finished file
$out->save('/home/phil/wolfrss.xml');"
```

Here I use XML::Simple to read and parse the existing .RSS file and XML::RSS to add our new item and write the modified version. The bless function tells Perl that the XML::Simple object $out now should be treated as an XML::RSS object. The only reason this does anything useful is the two modules use nearly identical variable names internally, derived from the tag names of the incoming RSS file.

This bless function copies over almost anything in the RSS file's header, but it doesn't bring over item or encoding tags. So I then copied over each item in a foreach loop, added today's date as the build and publication date and added the just-captured file as a new item. This item has a Web page link that is copied from the header, today's date as publication date and the all-important enclosure tag. The enclosure has a URL, in this case a file:// reference, because we are doing everything on the local filesystem. It also has a file length and a MIME type, audio/mpeg.

Shell variables replace all the quoted strings, and the super-sneaky shell variables $i, $o and $m get replaced by \$i, \$o and \$m. In other words, everywhere you see $i in the Perl script, the Perl interpreter actually gets the Perl variable name $i. Without that bit of substitution, the shell would replace each $i with a null string or, worse yet, whatever the shell variable i happened to hold before the script was executed. The reference to the actual MP3 file is a URL, file:///home/phil/wolf.2005-03-19.mp3, not merely a filename. When we enter the RSS feed file into Firefox or a feed aggregator program, we refer to it using URL notation as well, file:///home/phil/wolfrss.xml.

Why Not Just Do It in Perl?

It may seem strange that I'm calling a scripting language from another scripting language. The point is that I'm using each to do the things it's best at. Bash is designed to execute commands, and it's really easy to start a background process, find out its process ID and kill it again. On the other hand, trying to add an XML entry in Bash using the more basic string-handling tools, such as sed and grep, would have been, well, exactly the kind of thing that drove Larry Wall to write Perl in the first place.

Now that we have a script, we make the file executable and run it:

```
chmod +x catchthewolf
./catchthewolf
```

which results in a properly tagged MP3 file and a new entry in the wolfrss.xml RSS feed. When testing, you can uncomment the 30-second test line to make sure everything's working properly, but be sure to comment it back out before trying to catch a show. Now all that's left is to get our computer to run this thing at 5AM on Saturday. That's done by using the system's cron utility—invoke `crontab -e`— and adding an entry like this:

```
MAILTO=phil # Testing: mail script output to me

# Catch hour of the wolf 5AM Saturdays
59 4 * * sat /home/phil/catchthewolf
```

crontab's editor is most likely to be set to vi-style commands, so you have to use i to start typing and <Esc>:wq to save-and-exit. When you're done, you should see this message:

```
crontab: installing new crontab
```

which says you're all set. Check `man 5 crontab` for more information on how to make jobs repeat every day, once a month or whatever. You also want to make sure your user name is in the file /etc/cron.allow—the list of who can run jobs on the system's scheduler. If you're running on a remote system, verify with the administrators that you're allowed to run cron jobs.

To see the resulting podcast, point your RSS-aware software at the XML file the script creates. In Firefox, use Bookmarks→Manage Bookmarks→Add Live Bookmark, and remember to enter the URL starting with file:// and not the filename itself.

That's a Wrap

By taking two programs already on the hard drive, downloading two Perl modules and writing a few lines of shell script, we have assembled a homebrew Webcast recording system that saves our favorite programs for us to listen to whenever we choose. It also lets us know what it has done by popping up live bookmarks in Firefox and automatically transfers the recordings to our MP3 player. Some scripts for capturing other Internet radio shows will be available on the *Linux Journal* FTP site (see the on-line Resources). Now I just have to remember to delete the older files before my hard drive fills up with leftover Webcasts.

Thanks to Anne Troop, Jen Hamilton and Chris Riley for their many shell-scripting hints over the years; to Anne's friend Janeen Pisciotta for finding "Hour of the Wolf" for us in the first place; and to *LJ* Editor in Chief Don Marti for the cool podcast idea.

Resources for this article: www.linuxjournal.com/article/8402.

Phil Salkie is an industrial controls guru who has liked science fiction and radio drama since childhood. He has been a Linux fanatic since 2.0.12 or so and has the most wonderful, tolerant family—e-mail him at phil@asylumhouse.org.

DON'T BE SQUARE! GET CUBED!

L Series Laptop - LS1250-L
Light &Thin Performance
Starting at $1,562.54

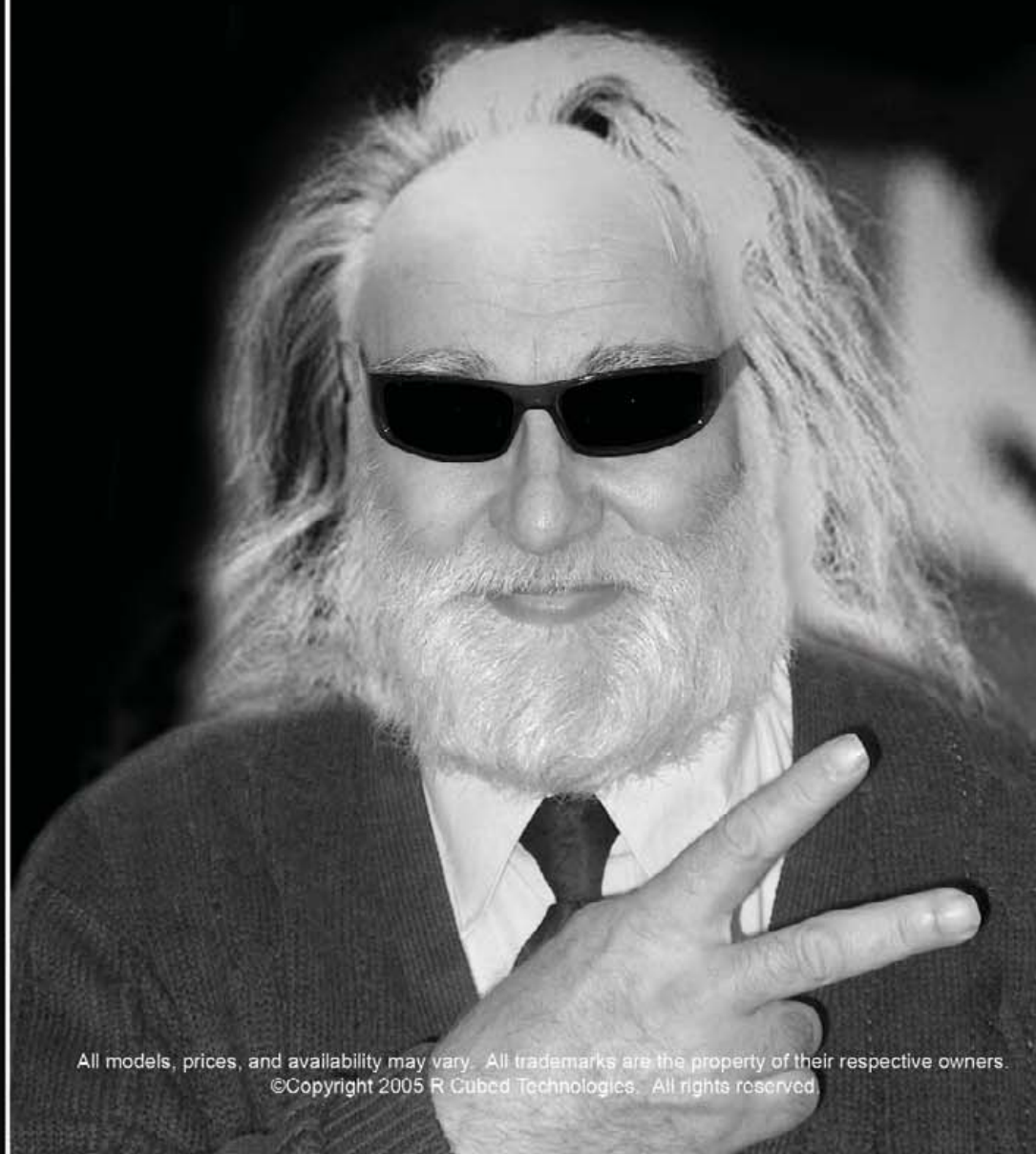
G Series Laptop - GW1550-L
Essential Technology on a Budget
Starting at $1,302.54

X Series Laptop - XW1550-L
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 with 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



Auditing Wi-Fi Protected Access (WPA) Pre-Shared Key Mode

Understand the risks of two wireless security technologies by experimenting with cracking tools.

BY JOHN L. MACMICHAEL

Although the implementation of wireless networks has increased exponentially, the focus on network and information security has not kept pace. Empirical evidence suggests that fewer than one-third of wireless networks have implemented any sort of data encryption, be it wired equivalent privacy (WEP) or Wi-Fi protected access (WPA). Those network administrators and home users who have implemented these encryption methods may have been lulled into a false sense of security. WEP is known to be easily exploited, and substantial although relatively unknown problems exist with WPA when used in consumer mode. This article focuses on data confidentiality provided through encryption by reviewing the flaws in WEP and examining the issues surrounding WPA. Tools that demonstrate the risk of using WPA in pre-shared key (PSK) mode are explored.

A Little History

WEP was ratified as an IEEE standard in 1999. It was designed to provide moderate protection against eavesdropping on data in transit and unauthorized access to the network resources. This protection was provided through an encryption scheme that utilized a flawed implementation of the RC4 stream cipher. The actual key size of the implementation was misleading, because the keys were 40-bit and 104-bit, with a 24-bit initialization vector (IV) added to the key. This led to the misnomer of 64-bit and 128-bit keys.

WEP suffered from a poor implementation of the key scheduling algorithm and transmitted the flawed IVs in the clear. A general acknowledgement that WEP was not an appropriate method of securing a wireless network came after Fluher, et al., published *Weaknesses in the Key Scheduling Algorithm of RC4* in 2001 and the Shmoo Group released the beta version of Airsnort. Capturing approximately five million data packets statistically would ensure the collection of approximately four thousand weak IVs. From this information, Airsnort could discern most WEP keys. These statistically weak interesting IVs received wide recognition within the industry, and as a result, most vendors made changes to their WEP firmware and software implementations that filtered or removed weak IVs.

Older versions of Airsnort and other tools that attacked WEP by examining interesting IVs became unusable as an attack vector against most wireless equipment produced after 2002. In 2004, Korek released a new WEP statistical cryptanalysis attack and while still based on the weaknesses in the key scheduling algorithm, the Korek attack removed the requirement for collection of interesting IVs. This attack has been coded into several tools, most notably Aircrack, WepLab and the newest version of Airsnort. Each tool functions slightly differently, but each requires as few as half as many packets to break WEP than the previous generation of WEP cracking tools.

802.11i and WPA

The IEEE recognized that WEP was not a sufficient method to protect wireless communications and set to work creating a new security standard, 802.11i, also known as WPA2. 802.11i was ratified as a draft standard in early 2004 and includes a robust set of security standards. The 802.11i architecture contains 802.1x for authentication and port-based access control, AES (advanced encryption standard) block cipher and CCMP (counter mode CBC MAC protocol) for keeping track of associations and providing confidentiality, integrity and origin authentication.

Of these robust requirements, AES is the most computationally intensive, and the 802.11b/g hardware that had been fielded for WEP was not up to the task of implementing the AES block cipher. It is likely that companies that fielded enterprise-wide wireless implementations would be concerned about fielding new equipment that was not backwards-compatible; legacy 802.11 hardware would not be capable of interoperating with new 802.11i hardware. This would cause companies either to field all new equipment at once or face a nightmare of interoperability.

Enter the Wi-Fi Alliance, a nonprofit industry association devoted to promoting the growth of wireless local area networks (WLANs). The Wi-Fi Alliance created the WPA specification as a bridging solution that would alleviate the concerns of WEP while providing a bridge to 802.11i. WPA was designed to conform to the majority of the 802.11i specifications. The major exception was WPA would not implement AES for encryption and would continue to use RC4. This methodology ensured that WPA would be backward-compatible with 802.11-certified hardware and forward-compatible with 802.11i hardware. In essence, it would provide a bridge as vendors brought new equipment on-line, allowing companies to

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℠

leverage the WPA standard while migrating to newer equipment in a phased manner.

WPA Modes

WPA solves several problems inherent in WEP. By implementing the Temporal Key Integrity Protocol (TKIP), the issues of privacy and encryption are mitigated, as the use of a RADIUS or Kerberos authentication server mitigates the problem of client-to-AP authentication and unauthorized network access. The TKIP protocol greatly expands the size of the keys, allows for per-user keying, creates an integrity-checking mechanism and removes the predictability in the WEP key scheme.

WPA can be implemented in two versions, WPA-Enterprise and WPA-Personal. WPA-Enterprise uses the 802.1x authentication framework with TKIP key encryption to prevent unauthorized network access by verifying network users through the use of a RADIUS or authentication server and ensures per-user-based keying. Thus far, WPA-Enterprise has not been prone to any attacks on the confidentiality of the per-user key. An intruder that could divine the key would find it unusable on all but the computer from which it was stolen.

WPA-Personal also uses the TKIP key encryption mechanism but uses a pre-shared key (PSK) instead of a per-user key generated from an authentication server. This mode often is referred to as WPA-PSK. In WPA-PSK, users must share a passphrase that may be from eight to 63 ASCII characters or 64 hexadecimal digits (256 bits). Similar to WEP, this passphrase is the same for all users of the network and is stored on the AP and client computer. WPA-PSK was designed for personal or small-business environments in which an authentication server is not required. In actual implementation, several mid-sized firms use WPA-PSK instead of WPA-Enterprise in an effort to simplify enterprise management.

Problems with WPA-PSK

In November 2003, Robert Moskowitz, a senior technical director at ICSA Labs (part of TruSecure) released "Weakness in Passphrase Choice in WPA Interface". In this paper, Moskowitz described a straightforward formula that would reveal the passphrase by performing a dictionary attack against WPA-PSK networks. This weakness is based on the fact that the pairwise master key (PMK) is derived from the combination of the passphrase, SSID, length of the SSID and nonces. The concatenated string of this information is hashed 4,096 times to generate a 256-bit value and combine with nonce values. The information required to create and verify the session key is broadcast with normal traffic and is readily obtainable; the challenge then becomes the reconstruction of the original values. Moskowitz explains that the pairwise transient key (PTK) is a keyed-HMAC function based on the PMK; by capturing the four-way authentication handshake, the attacker has the data required to subject the passphrase to a dictionary attack. According to Moskowitz, "a key generated from a passphrase of less than about 20 characters is unlikely to deter attacks."

In late 2004, Takehiro Takahashi, then a student at Georgia Tech, released WPA Cracker. Around the same time, Josh Wright, a network engineer and well-known security lecturer, released coWPAtty. Both tools are written for Linux systems and perform a brute-force dictionary attack against WPA-PSK networks in an attempt to determine the shared passphrase. Both require the user to supply a dictionary file and a dump file that contains the WPA-PSK four-way handshake. Both function similarly; however, coWPAtty contains an automatic parser while WPA Cracker requires the user to perform a manual string extraction. Additionally, coWPAtty has optimized the HMAC-SHA1 function and is somewhat faster. Each tool uses the PBKDF2 algorithm that governs PSK hashing to attack and determine the passphrase. Neither is extremely fast or effective against larger passphrases, though, as each must perform 4,096 HMAC-SHA1 iterations with the values as described in the Moskowitz paper.

Audit—System Preparation

To perform the audit, we need a libpcap file that contains the WPA-PSK four-way authentication handshake and the program WPA Cracker or coWPAtty. Capturing the four-way handshake in the libcap-compatible dumpfile format is the most challenging part of the exercise. It requires a wireless NIC that is capable of rf monitor mode and a set of modified wireless drivers that allow packets to be passed up through the interface.

libpcap is either pre-installed or available as a package for most modern Linux distributions and is the de facto standard for low-level network monitoring. The libpcap network library provides a system-independent interface for user-level packet capture. The steps for installation are straightforward for those that prefer to compile vice install packages. Download the latest libpcap file from SourceForge.net and then expand the libpcap file, configure, make and make install. When compiling your code, the filename depends on the version you downloaded:

```
# tar zxvf libpcap-current.tar.gz
# cd libpcap-2005.06.01
# ./configure && make && make install
```

Now that the system has the ability to capture the network data, a method is needed to read the data from the air. Most modern Linux distributions ship with one or more wireless drivers, but few ship with the modified drivers that allow raw monitor mode or rfmon. rfmon is a sniffing mode that allows the wireless NIC to report data from the 802.11 layer. Although few major distributions ship with rfmon-capable drivers, many live CD security distributions, such as Knoppix-STD, Auditor and Whoppix, have precompiled modified wireless drivers as well as compiled binaries of the audit tools.

The modified driver to be used is dependent on the type of chipset. For example, the Prism2-based cards may use the wlan-ng drivers or Host-AP drivers, and Orinoco cards and clones can use the patched orinoco_cs drivers. Orinoco cards that use the Orinoco drivers greater than version 0.15 have built-in monitor mode, while Atheros-based cards may use the MadWiFi drivers. This list is not inclusive, and there are many possible options in the form of driver patches, standalone packages that build driver modules outside of the kernel tree and kernel mainline drivers that are part of the kernel source itself. It is assumed that readers have the ability to install a driver for their particular cards and distributions that permits wireless monitor mode.

Capturing the Wireless Data

Several methods can be used to capture the wireless traffic that contains the WPA-PSK four-way handshake of interest. tcpdump allows for network monitoring and data acquisition, but it does not readily provide meaningful AP data. Kismet is arguably the best tool for wireless data capture, auditing traffic, network detection and general wireless sniffing. Specifically, Kismet can log the packet data into a dump file required for this demonstration, but it is overkill for this situation. The most elegant method of capture is to use airodump, which is part of the Aircrack 2.1 suite written by Christopher Devine. Aircrack can handle large capture files and displays meaningful AP information to include SSID, total number of unique IVs and packet size. Aircrack is available in the Tar File Gzipped format (tgz). Install by following these steps to build the Aicrack suite of tools; the specific tool of interest in this situation is airodump:

```
# tar zxvf aircrack-2.1.tgz
# cd aircrack-2.1
# make
```

With the tools compiled, wireless traffic now can be captured. The wireless NIC first must be placed in rf monitor mode. For example, if using the patched version of the Orinoco driver, the following commands would be issued, where <AP channel> is the channel of interest:

```
# iwpriv eth0 monitor 1 <AP channel>
```

The wireless NIC then is enabled:

```
# ifconfig wlan0 up
```

Finally, commands to capture traffic would be issued:

```
# airodump wlan0 datafilename
```

Airodump continuously displays the AP SSID and packet capture information on the specified channel. To reduce the amount of captured data, the MAC address of the AP may be appended after the datafilename. To exit airodump, use the Ctrl-C command.

Although airodump happily captures traffic, the four-way handshake is not captured until a client-to-AP association occurs. This is a random occurrence from the attacker's point of view, but forced reassociations can be accomplished by executing a death attack using a tool such as void11 that forces the de-authentication of wireless clients from their associated APs. The wireless client automatically attempts reassociation, which allows the capture of the WPA-PSK four-way handshake. Assuming the handshake has been captured, it is time to execute the brute-force dictionary attack.

coWPAtty Execution

coWPAtty requires that OpenSSL be installed on your system. After downloading coWPAtty, install it using the following steps:

```
# tar zxvf Cowpatty-2.0.tar.gz
# cd cowpatty
# make
```

You now have built the coWPAtty binary. Execute the binary by supplying the libpcap that includes a captured four-way handshake, a dictionary file of passphrases from which to guess and the SSID of the network. The options are:

- -f: dictionary file
- -r: packet capture file
- -s: network SSID

The binary is executed with the following command:

```
# ./cowpatty -r datafilename \
 -f dictionaryfile -s SSID
```

If there is no WPA four-way exchange, the following message is displayed:


```
End of pcap capture file, incomplete TKIP four-way
exchange.
Try using a different capture.
```

If the file did contain the four-way handshake, the following is displayed:

```
coWPAtty 2.0 - WPA-PSK dictionary attack.
<jwright@hasborg.com>
Collected all necessary data to mount crack against
passphrase.  Loading words into memory, please be
patient ... Done (XX words).  Starting dictionary
attack. Please be patient.
```

coWPAtty continues the intensive and relatively slow process of testing each dictionary word as a passphrase by using the PBKDF2 function and making 4096 SHA-1 passes on each passphrase in the supplied data set. coWPAtty updates its progress until it reports either it has found the WPA-PSK passphrase or it was unable to identify the WPA-PSK passphrase from the supplied dictionary file. As noted in the documentation, coWPAtty is not fast, due to the number of repetitions required for each passphrase. Expect approximately 45 keys per second in actual use.

For users who care to demonstrate this tool but are unable to capture the network data, coWPAtty includes a sample packet capture file, named eap-test.dump, that was generated from an AP with SSID somethingclever and a PSK of family movie night. To demonstrate the attack utilizing the supplied file, enter the following command ensuring that the supplied dictionary has the phrase somethingclever included:

```
# ./cowpatty -r eap-test.dump \
-f dictionaryfile -s somethingclever
```

Conclusion

This article examined some of the vulnerabilities within WEP and WPA and provides the tools and method for auditing WPA pre-shared key mode passphrases. To do this, we examined the framework and flaws in WEP and reviewed the risks associated with using WPA-PSK passphrases of less than 20 characters. It has been demonstrated that although the method to crack the WPA-PSK is not trivial, it also is not beyond the reach of an average Linux user. Home users can lessen their security risks by using a passphrase significantly greater than 20 characters or, alternatively, by using WPA-Enterprise and incorporating an authentication server. Corporate users should implement an authentication server, use per-user keying and refrain from implementing WPA in PSK mode.

Resources for this article: www.linuxjournal.com/article/8405.

John L. MacMichael (CISSP, GSEC, CWNA) is a Naval Officer and Information Professional who works in the field of Information Assurance. He considers himself a journeyman Linux user and utilizes a variety of distributions both at work and home, including Slackware, Debian, Red Hat and several live distros; he has yet to find his favorite. He invites your comments at johnny@757.org.

Compression Tools Compared

Use top-performing but little-known lossless data compression tools to increase your storage and bandwidth by up to 400%.

BY KINGSLEY G. MORSE JR.

Data compression works so well that popular backup and networking tools have some built in. Linux offers more than a dozen compression tools to choose from, and most of them let you pick a compression level too. To find out which perform best, I benchmarked 87 combinations of tools and levels. Read this article to learn which compressor is a hundred times faster than the others and which ones compress the most.

The most popular data compression tool for Linux is gzip, which lets you choose a compression level from one to nine. One is fast, and nine compresses well. Choosing a good trade-off between speed and compression ratio becomes important when it takes hours to handle gigabytes of data. You can get a sense of what your choices are from the graph shown in Figure 1. The fastest choices are on the left, and the highest compressing ones are on the top. The best all-around performers are presented in the graph's upper left-hand corner.

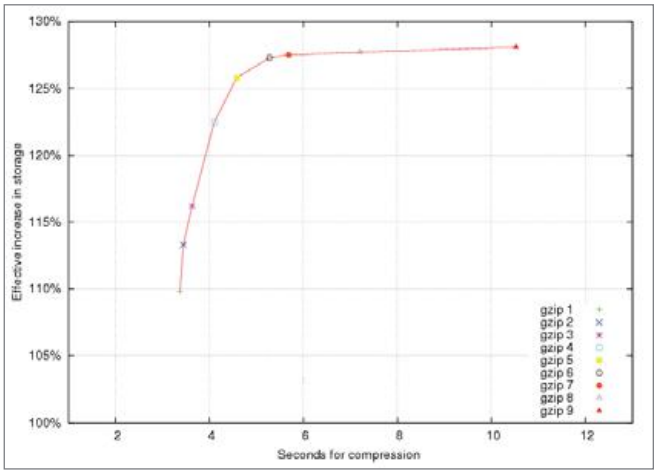


Figure 1. Increasing the compression level in gzip increases both compression ratio and time required to complete.

But many other data compression tools are available to choose from in Linux. See the comprehensive compression and decompression benchmarks in Figures 2 and 3. As with gzip, the best performers are in the upper left-hand corner, but these charts' time axes are scaled logarithmically to accommodate huge differences in how fast they work.

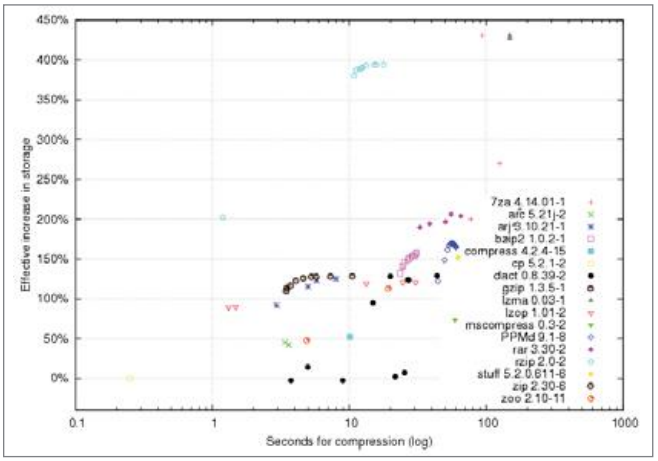


Figure 2. Performance of Many Utilities, Compression

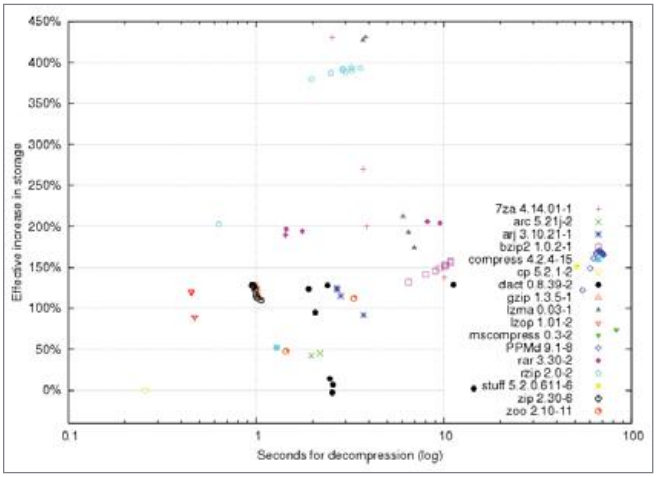


Figure 3. Performance of Many Utilities, Decompression

Better Backups

The tools that tend to compress more *and* faster are singled out in the graphs shown in Figures 4 and 5. Use these for backups to disk drives. Remember, their time axes are scaled logarith-

mically. The red lines show the top-performing ones, and the green lines show the top performers that also can act as filters.

Check whether the data compression tool that you want is installed on both computers. If it's not, you can see where to get it in the on-line Resources for this article. Remember to replace `a/dir` in the following examples with the real path of the data to back up.

Unless your data already is in one big file, be smart and consolidate it with a tool such as tar. Aggregated data has more redundancy to winnow out, so it's ultimately more compressible.

The Benchmarks

How compactly data can be compressed depends on what type of data it is. Don't expect big performance increases from data that's already compressed, such as files in Ogg Vorbis, MP3 or JPEG format. On the other hand, I've seen data that allows performance increases of 1,000%!

All benchmarks in this article used the same 45MB of typical Linux data, containing:

- 24% ELF 32-bit LSB
- 15% ASCII C program
- 11% gzip compressed data
- 8% ASCII English text
- 7% binary package
- 4% directory
- 2% current ar archive
- 2% Texinfo source text
- 2% PostScript document text
- 2% Bourne shell script
- 2% ASCII text
- 21% various other data types

This data set was chosen because it is more representative of the demands made on today's Linux systems than the data used in the traditional Canterbury and Calgary test data, because this data set is bigger and contains Linux binaries.

I used the same lightly loaded AMD Athlon XP 1700+ CPU with 1GB of RAM and version 2.4.27-1-k7 of the Linux kernel for all tests. Unpredictable disk drive delays were minimized by pre-loading data into RAM. Elapsed times were measured in thousandths of a second. I'm not affiliated with any of the tools, and I strove to be objective and accurate.

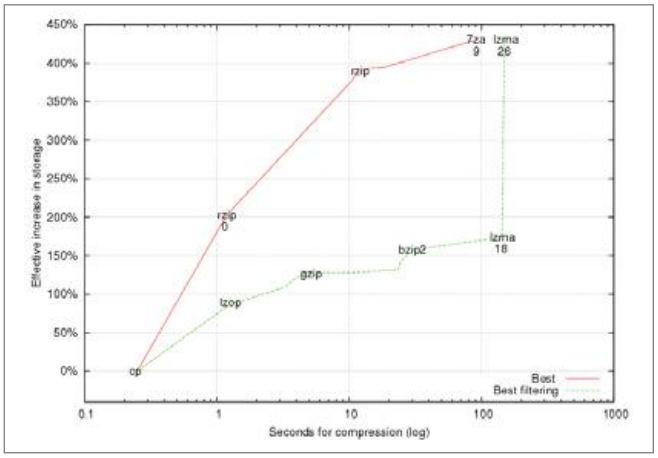


Figure 4. Best Utilities for Backups, Compression

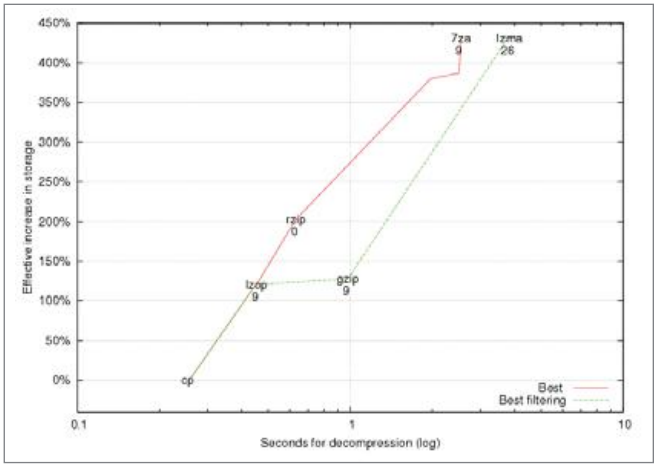


Figure 5. Best Utilities for Backups, Decompression

Filters

Filters are tools that can be chained together at the command line so that the output of one is piped elegantly into the input of the next. A common example is:

```
$ ls | more
```

Filtering is crucial for speeding up network transfers. Without it, you have to wait for all the data to be compressed before transferring any of it, and you need to wait for the whole transfer to complete before starting to decompress. Filters speed up network transfers by allowing data to be simultaneously compressed, transferred and decompressed. This happens with negligible latency if you're sending enough data. Filters also eliminate the need for an intermediate archive of your files.

But be aware that the redundancy that saps your performance also may make it easier to recover from corruption. If you're worried about corruption, you might consider testing for it with the cksum command or adding a limited amount of redundancy back into your compressed data with a tool such as parchive or ras.

lzop often is the fastest tool. It finishes about three times faster than gzip but still compresses data almost as much. It finishes about a hundred times faster than lzma and 7za. Furthermore, lzop occasionally decompresses data even faster than simply copying it! Use lzop on the command line as a filter with the backup tool named tar:

```
$ tar c a/dir | lzop - > backup.tar.lzo
```

tar's c option tells it to create one big archive from the files in a/dir. The | is a shell command that automatically pipes tar's output into lzop's input. The - tells lzop to read from its standard input, and the > is a shell command that redirects lzop's output to a file named backup.tar.lzo.

You can restore with:

```
$ lzop -dc backup.tar.lzo | tar x
```

The d and c options tell lzop to decompress and write to standard output, respectively. tar's x option tells it to extract the original files from the archive.

Although lzop is impressive, you can get even higher compression ratios—much higher! Here's how. Combine a little-known data compression tool named lzma with tar to increase storage space effectively by 400%. Here's how you would use it to back up:

```
$ tar c a/dir | lzma -x -s26 > backup.tar.lzma
```

lzma's -x option tells it to compress more, and its -s option tells it how big of a dictionary to use.

You can restore with:

```
$ cat backup.tar.lzma | lzma -d | tar x
```

The -d option tells lzma to decompress. You need patience to increase storage by 400%; lzma takes about 40 times as long as gzip. In other words, that one-hour gzip backup might take all day with lzma.

This version of lzma is the hardest compressor to find. Make sure you get the one that acts as a filter. See Resources for its two locations.

The data compression tool with the best trade-off between speed and compression ratio is rzip. With compression level 0, rzip finishes about 400% faster than gzip *and* compacts data 70% more. rzip accomplishes this feat by using more working memory. Whereas gzip uses only 32 kilobytes of working memory during compression, rzip can use up to 900 megabytes, but that's okay because memory is getting cheaper and cheaper.

Here's the big but: rzip doesn't work as a filter—yet. Unless your data already is in one file, you temporarily need some extra disk space for a tar archive. If you want a good project to work on that would shake up the Linux world, enhance

rzip to work as a filter. Until then, rzip is a particularly good option for squeezing a lot of data onto CDs or DVDs, because it performs well and you can use your hard drive for the temporary tar file.

Here's how to back up with rzip:

```
$ tar cf dir.tar a/dir
$ rzip -0 dir.tar
```

The -0 option says to use compression level 0. Unless you use rzip's -k option, it automatically deletes the input file, which in this case is the tar archive. Make sure you use -k if you want to keep the original file.

rzipped tar archives can be restored with:

```
$ rzip -d dir.tar.rz
$ tar xf dir.tar
```

rzip's default compression level is another top performer. It can increase your effective disk space by 375% but in only about a fifth of the time lzma can take. Using it is almost exactly the same as the example above; simply omit compression level -0.

Better Bandwidth

Data compression also can speed up network transfers. How much depends on how fast your CPU and network are. Slow networks with fast CPUs can be sped up the most by thoroughly compressing the data. Alternatively, slow CPUs with fast connections do best with no compression.

Find the best compressor and compression level for your hardware in the graph shown in Figure 6. This graph's CPU and network speed axes are scaled logarithmically too. Look where your CPU and network speeds intersect in the graph, and try the data compression tool and compression level at that point. It also should give you a sense of how much your bandwidth may increase.

For example, if you have a 56Kbps dial-up modem and a 3GHz CPU, their speeds intersect in the light-yellow region labeled lzma 26 at the top of the graph. This corresponds to

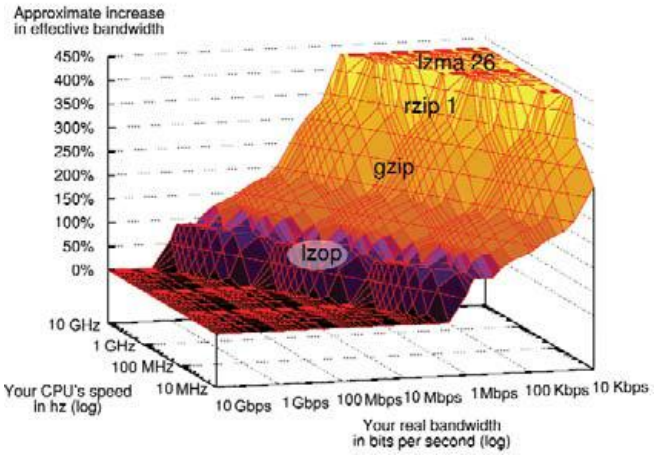


Figure 6. Best Compressors for Improving the Bandwidth of Various Hardware

Network Transfer Estimates

To find the best compressors for various CPU and network speeds, I considered how long it takes to compress data, send it and decompress it. I projected how long compression and decompression should take on computers of various speeds by simply scaling actual test results from my 1.7GHz CPU. For example, a 3.4GHz CPU should compress data about twice as fast. Likewise, I estimated transfer times by dividing the size of the compressed data by the network's real speed.

The overall transfer time for non-filtering data compression tools, such as rzip, simply should be about the sum of the estimated times to compress, send and decompress the data.

However, compressors that can act as filters, such as gzip, have an advantage. They simultaneously can compress, transfer and decompress. I assumed their overall transfer times are dominated by the slowest of the three steps. I verified some estimates by timing real transfers.

using lzma with a 2^{26} size dictionary. The graph predicts a 430% increase in effective bandwidth.

On the other hand, if you have a 1GHz network, but only a 100MHz CPU, it should be faster simply to send the raw uncompressed data. This is depicted in the flat black region at the bottom of the graph.

Don't assume that you always should increase performance the most by using lzma, however. The best compression tool for data transfers depends on the ratio of your particular CPU's speed to your particular network's speed.

If the sending and receiving computers have different CPU speeds, try looking up the sending computer's speed in the graph. Compression can be much more CPU-intensive. Check whether the data compression tool and scp are installed on both computers. Remember to replace user@box.com and file with the real names.

For the fastest CPUs and/or slowest network connections that fall in the graph's light-yellow region, speed up your network transfers like this:

```
$ cat file \
| lzma -x -s26 \
| ssh user@box.com "lzma -d > file"
```

ssh stands for secure shell. It's a safe way to execute commands on remote computers. This may speed up your network transfer by more than 400%.

For fast CPUs and/or slow networks that fall into the graph's dark-yellow zone, use rzip with a compression level of one. Because rzip doesn't work as a filter, you need temporary space for the compressed file on the originating box:

```
$ rzip -1 -k file
$ scp file.rz user@box.com:
$ ssh user@box.com "rzip -d file.rz"
```

The -1 tells rzip to use compression level 1, and the -k tells it to keep its input file. Remember to use a : at the end of the scp command.

rzipped network transfers can be 375% faster. That one-hour transfer might finish in only 16 minutes!

For slightly slower CPUs and/or faster networks that fall in the graph's orange region, try using gzip with compression level 1. Here's how:

```
$ gzip -1c file | ssh user@box.com "gzip -d > file"
```

It might double your effective bandwidth. -1c tells gzip to use compression level 1 and write to standard output, and -d tells it to decompress.

For fast network connections and slow CPUs falling in the graph's blue region, quickly compress a little with lzop at compression level 1:

```
$ lzop -1c file | ssh user@box.com "lzop -d > file"
```

The -1c tells lzop to use compression level 1 and to write to standard output. -d tells it to decompress. Even with this minimal compression, you still might increase your hardware's effective bandwidth by 75%.

For network connections and CPUs falling in the graph's black region, don't compress at all. Simply send it.

C Libraries

If you want even more performance, you may want to try calling a C compression library from your own program.

Table 1. C Libraries

Tool	Library
lzop	liblzo1
gzip, zip	zlib1g, zlibc, zlib, zziplib
bzip2	libbzip2
7za, lzma	lzma

Resources for this article: www.linuxjournal.com/article/8403.

Kingsley G. Morse Jr. has been using computers for 29 years, and Debian GNU/Linux has been on his desktop for nine. He worked at Hewlett-Packard and advocates for men's reproductive rights. He can be reached at change@nas.com.

RoboNexus

IDG WORLD EXPO

roboticstrends

The Largest Robotics Event in the Western Hemisphere

October 6-9, 2005
San Jose Convention Center, San Jose, CA

EXPO Magazine's best new show 2005 Winner




The International Business Development, Educational and Consumer Event for Personal, Service and Mobile Robotics

- 50 Robotics Visionaries and Thought Leaders
- 50,000 Square Foot Exposition
- 5 Comprehensive Professional Development Conferences:
  - Business Development and Opportunities Conference
  - Emerging Robotics Technologies and Applications Conference
  - Robotics Design, Development and Standards Conference
  - Robotics Education and Instruction Conference
  - Consumer Robotics and Entertainment Event

New for RoboNexus 2005!

- Business-to-Business and Consumer Entertainment Expo Areas
- Service Robotics Summit
- Service Robotics Pavilion
- "Robotics at Home" Demo Stage
- Robotics Innovators Awards –the 'Robi'

www.robonexus.com

Founding Sponsor

iRobot

Premier Sponsor

Premier Media Sponsor

POPULAR science

Gold Sponsors

HITEC

Corporate Sponsors

evolution robotics

ESS

LEGO education

mBus machineBus.com

OLogic

Openware Robotics

ShopBot

ROGUE robotics

SYSTRONIX Embedded Java Spoken Here

Media Sponsors

BATTERY POWER PRODUCTS & TECHNOLOGY

CIRCUIT CELLAR

e-DRIVE

Entertainment Engineering

GLOBAL FUTURES

VEHICLE

Nuts & Volts

LINUX JOURNAL

NetworkWorld

PC WORLD

Popular Mechanics

TUX

SERVO

VisionSystems DESIGN

Association Sponsors

Acceleration Studies Foundation

OMG

GETI

802.1x on Linux with xsupplicant

Many of the well-known problems in 802.11 security are quite old and can be addressed by using 802.1x appropriately. Here's the client side.

BY MATTHEW GAST

When WEP's flaws became apparent, the wireless industry started developing new protocols to address the published weak points. These new protocols grew up around the IEEE 802.1x framework, which is a way of using the Extensible Authentication Protocol (EAP) and all of its methods on a LAN link. 802.1x client software programs, called supplicants, were brought to market by operating system vendors as well as by third-party developers.

Linux, however, initially was left out of the 802.1x frenzy. Network administrators who supported power users were forced to rely on manual WEP-based solutions with MAC address filtering or VPNs to secure Linux before supplicants were widely available. Happily, now two open-source supplicants are bringing high-quality wireless security to Linux. This article describes the process of setting up xsupplicant, which is also known as Open1X.

Wireless Extensions

The wireless extensions API originally was designed to provide a unified way of having programs interact with drivers. Like any API, it saves developers from having to know the details of how to interact with every card. 802.1x supplicants, for example, are able to use a wireless extensions system call to set keys, rather than using card-specific calls for every card that exists.

The wireless extensions interface has gone through several versions. WPA support was added in wireless extensions version 18 (WE-18). Some distributions using the 2.6 kernel already have WE-18 support. Older kernels need to be patched, however. My test laptop runs Slackware, which still is using the 2.4 kernel. The 2.4 kernel has support for version 16 of wireless extensions, but patches are available for version 2.4.30. Patch download locations appear in the on-line Resources for this article. Begin by applying two patches to the kernel source:

```
# patch -p1 ~/iw249_we17-13.diff
patching file include/linux/netdevice.h
patching file include/linux/wireless.h
patching file include/net/iw_handler.h
patching file net/core/dev.c
patching file net/core/wireless.c
# patch -p1 ~/iw240_we18-5.diff
patching file include/linux/wireless.h
patching file net/core/wireless.c
```

To keep modules straight, I often find it helpful when patching kernels to edit the Makefile to include an extra version number in addition to the patch level. My wireless extensions 18 kernel is built as 2.4.30WE18.

The most common tools used with wireless extensions are the wireless toolset, and the most common tool you will use is iwconfig. Wireless tools version 28 is the current version and supports WE-18. Grab the source code from the Web site (see Resources). A simple `make` command builds the tools.

Getting the Driver Going

Many cards are supported under Linux, but a handful of drivers have captured the bulk of the popularity:

- MADwifi, the Multi-band Atheros Driver for Wi-Fi: Atheros-based cards have some of the best hardware support for 802.11a networking. Chances are good that if your card supports 802.11a, it uses an Atheros-designed chip.

- Intel IPW drivers for Centrino chipsets: Intel sponsors open-source driver development projects for the various Centrino chipsets. Due to the sheer number of Centrino chipsets on the market, these drivers are widely used.

- orinoco_cs: the first widely used 802.11 card was the Orinoco Gold card, based on the Hermes chipset. These cards were sold under a variety of names, and they all performed quite well in their day. Although the radio performance and throughput of these cards is no longer cutting-edge, the driver is well understood and often serves as a testbed for new ideas.

This article is not meant to be a definitive treatment of working with drivers. I use Atheros-based cards because I have an 802.11a network at home and want a dual-band card for packet analysis. Therefore, I am writing about MADwifi.

MADwifi has not released any packaged source files. To use the driver, you must download the code from CVS. The build files distributed with MADwifi use your current kernel. If you have patched the kernel to update wireless extensions, reboot before building MADwifi:

```
$ cvs -z3 -d:pserver:anonymous@cvs.sourceforge.net:
➥/cvsroot/madwifi co madwifi
$ cd madwifi
$ make

root@bloodhound:/home/user/madwifi# make install
```

High Performance on Wall Street

September 26, Monday Roosevelt Hotel, New York, NY

Madison Avenue at East 45th Street next to Grand Central Station

Back by Popular Demand. Register now for Free Show and Low Cost Conference.

High Performance Computing, Grid, Blade, Cluster, high-speed networking, scalable storage, Linux systems will all be there.

2004 SPONSORS

Platinum Sponsor

Sun microsystems

Gold Sponsors

CTC Cornell Theory Center

intel

IBM

Silver Sponsors

egenera

Enigmatec Corporation

Novell

The 2005 High Performance on Wall Street will return to the Roosevelt Hotel, New York by popular demand. Attendees reported this was the best New York show to network, see, examine, compare HPC systems.

This annual event is focused on the New York financial markets, Wall Street, and IT management. Wall Street will increase IT spending 7% to 8% this year, reports Forrester Research.

Plan to see HPC, Linux, Grid, Blade, Utility, Open Source solutions for IT management in the financial sector.

Wall Street IT chiefs are looking for reduced total-cost-of-ownership and greater return-on-investment using HPC applications. The big savings are in consolidations, time-saving deployment, and money-saving Grid applications to upgrade aging legacy systems.

The 2005 High Performance on Wall Street is a focused, no-waste, intimate market. The Roosevelt Hotel is in the heart of the New York market.

The major financial firms headquartered in New York will be there – Merrill Lynch, Lehman Brothers, Morgan Stanley, JP MorganChase, DeutscheBank, Salomon Smith Barney/Citibank. Meet key HPC, Linux, Grid, and Utility IT directors, architects, and technologists at this networking event.

Register online for the free show: highperformanceonwallstreet.com

Show and Conference Hours

Show:
8 am – 3:30 pm
Grand Ballroom Foyer

Conference:
9 am – 4:50 pm
Grand Ballroom

2004 Media and Association Sponsors:

Linux Journal · Linux Magazine · STP Information Services · waters · DWT · Global Investment Technology · Reference Data Review · Windows in Financial Services · WSTA Wall Street Technology Association · zapthink · OpenEnterprise Trends.com · DevTown Station The Software Developers' Training Portal · Transaction Networks & Technologies · ClusterWorld · Integration Developer News Preparing Developers for the Next Economy · A-Team Insight · Inside Market Data · Hedge Fund & Investment Technology

Conference Producer: LighthousePartners

PR Partner: massing pr

For more information on the event, to be a sponsor or exhibitor, contact:

Flagg Management Inc
353 Lexington Ave,
NY, NY 10016
(212) 286 0333,
flaggmgmt@msn.com

Conference Management:
Pete Harris,
Lighthouse Partners
(718) 237 2796,
pete@lighthouse-partners.com

highperformanceonwallstreet.com

Atheros-based cards do not use firmware. Instead, they have a binary-only object called the hardware abstraction layer (HAL). Atheros has interpreted FCC regulations in such a way that requires the HAL to be kept closed-source. The HAL serves the same purpose as firmware on other cards—it implements low-level operations for the driver. The HAL is distributed as a uuencoded file, so you must install the uudecode program to install the HAL. It probably is in the shell archive utilities package for your distribution, but the location may vary. The OpenBSD Atheros driver includes an open-source, reverse-engineered HAL, but it has not been ported yet to Linux.

The kernel modules built as part of the process are installed in your modules directory. The driver includes its own 802.11 support layer composed of the modules wlan, wlan_wep, wlan_tkip and so on. The hardware-specific part of MADwifi is composed of modules that begin with the prefix ath_: the driver ath_pci, the HAL ath_hal and rate adaptation algorithms (ath_rate_*). All the modules are installed in the net/ directory.

Testing the Driver

In addition to having up-to-date wireless support in the kernel, you need to have a properly configured wireless networking subsystem. Many "wireless" problems encountered when dealing with 802.1x on Linux are PC card configuration problems. When the card is inserted, you should get a high-pitched beep indicating that Card Services has loaded the right driver. A second beep is used to communicate the status of the card configuration, so a second lower beep is fine because the configuration of the card hasn't been set up yet.

If the card is recognized and the right driver is loaded, try firing up a wireless network with no encryption and no authentication. Configure association to the network with iwconfig, and bring up the card with ifconfig. The MADwifi driver creates interfaces that begin with the prefix ath, so my interface is ath0. Depending on the driver you use, your interface may be different. When the card first comes up, you can see it scan for the network as the frequency reported by iwconfig changes. When the card successfully associates to a network, it reports the access point MAC address as well as the operating frequency. At that point, you should be able to ask the network for an IP address, using whatever tool is favored by your Linux distribution:

```
# iwconfig ath0 essid "clearnet"
# ifconfig ath0 up
# iwconfig ath0
ath0  IEEE 802.11g  ESSID:"etherclear"
      Mode:Managed  Frequency:2.412 GHz  Access Point: 00:0B:0E:2F:0A:40
      Bit Rate:12 Mb/s   Tx-Power:50 dBm   Sensitivity=0/3
      Retry:off   RTS thr:off   Fragment thr:off
      Power Management:off
      Link Quality=39/94  Signal level=-56 dBm  Noise level=-95 dBm
      Rx invalid nwid:107  Rx invalid crypt:0  Rx invalid frag:0
      Tx excessive retries:22  Invalid misc:22   Missed beacon:0
# dhcpcd -d -t 10 ath0
dhcpcd: MAC address = 00:20:a6:4c:ca:4b
dhcpcd: your IP address = 172.16.199.84
```

If you can associate to a network, your card is functional. Although it is not necessary to find out if you can obtain an IP address from an unencrypted network, it is helpful to know that the frame handling and network stacks are working and that DHCP service is configured correctly on the network. With the wireless network system having basic functionality, we can move on to providing security for it.

xsupplicant

Two major supplicants exist for Linux: xsupplicant, also known as Open1X, and wpa_supplicant. This article discusses only the former. Before getting to work on xsupplicant, check the version of OpenSSL on your system. xsupplicant requires OpenSSL 0.9.7 or later to provide transport layer security (TLS) support. All the commonly used 802.1x authentication protocols require TLS, either for authentication directly with digital certificates (EAP-TLS) or as a protective tunnel for some other form of authentication (TTLS or PEAP). You need a development version of the packages to get the expected headers.

Download the source code from SourceForge (see Resources) At the time of this writing, the current release is 1.2pre1:

```
$ tar -xzf Xsupplicant-1.2pre1.tar.gz
$ cd xsupplicant
$ ./configure --with-madwifi-path=~/madwifi
. . .
Adding MADWIFI WPA support.
. . .
$ make

# make install
```

As a result of the build, three executables are installed. The only one you are likely to use is /usr/local/sbin/xsupplicant.

Certificate Wrestling

Secured EAP authentication generally depends on digital certificates. Certificate data is encoded using either the privacy-enhanced mail (PEM) format or the distinguished encoding rules (DER). My experience is that xsupplicant likes its certificates in PEM format, but many certificate authorities hand out certificates in the DER format. Fortunately, OpenSSL is quite good at converting between formats:

```
# openssl x509 -inform DER -outform PEM \
-in MyCA.der -out MyCa.pem
```

To see the actual data encoded within the certificate, you can use the openssl command to print textual output:

```
# openssl x509 -in MyCA.pem -text
```

How exactly you obtain the certificate is up to your network administrator. Many certificate authorities make the root certificate available on a Web page.

Configuring xsupplicant

When run, xsupplicant searches for its configuration file in /etc. The config file, /etc/xsupplicant.conf, is not installed by

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 a 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
1.706.548.7200

CORAID

default, but it's easy enough to copy over:

```
# cp xsupplicant.conf /etc/xsupplicant.conf
```

Specify the user identity, possibly the password and the root CA certificate in the configuration file. Each network can have its own configuration by bracketing the entire network configuration. A simple configuration for a network that uses PEAP with MSCHAP-V2 for inner authentication might look something like this:

```
dynamic-wep
{
  allow_types=all
  identity = testuser
  eap_peap {
       root_cert = /usr/local/etc/myCA.der
       random_file = /path/to/random/source
       allow_types = eap_mschapv2
       eap-mschapv2 {
         username = testuser
         password = "testpw"
       }
  }
}
```

Linux has two random number devices, /dev/random and /dev/urandom. Both pull random numbers from a system entropy pool, but the former device returns only strong random numbers. As a result, I highly recommend using /dev/random as the random number device file. Many 802.1x implementations can cope with relatively large delays while waiting for a response. At the Interop Labs in Las Vegas in May 2005, we authenticated a user account through a multi-hop global distributed RADIUS system, so end-to-end latency was much higher than on most networks.

For testing purposes, certificate validation can be disabled by setting the root_cert location to NONE. Although useful for testing purposes, disabling certificate authentication removes the protections provided by the certificate and should not be done for normal deployments.

Running xsupplicant

Once xsupplicant is configured, you finally can authenticate to the network. Start by connecting to the network that you want to attach to with iwconfig and bringing up the interface. I have found that it helps to give xsupplicant a dummy WEP key so it knows that it will be connecting to an encrypted network as well. Three commands do the trick:

```
# iwconfig ath0 key 1234567890123456789012345 6
# iwconfig essid "batnet"
# ifconfig ath0 up
```

The wireless interface name is driver-dependent. My interface is ath0, but yours may not be.

In the current version of xsupplicant, it is mandatory to supply an interface with the -i option. When testing, I generally find it helpful to log debug messages with -d and keep the process in the foreground with -f. To see a full list of what can be printed, use --help:

```
# xsupplicant -w -dasic -i ath0 -f
```

Debug messages print out each frame that is sent and received, as well as provide processing information with each sent or received frame. At the end of the process, the key information is processed. For example, a dynamic WEP key looks like this:

```
Processing EAPoL-Key!
[INT] Key Descriptor   = 1
[INT] Key Length       = 13
[INT] Replay Counter   = 41 2F BB 2D 00 00 00 D6
[INT] Key IV           = 66 15 69 E2 B2 8C 0E 89 7C D3 94 8C 93 25 43 1B
[INT] Key Index (RAW)  = 80
[INT] Key Signature    = 49 C1 15 B8 E9 D0 87 53 A6 FD 5D 76 CB 51 9D 65
[INT] EAPoL Key Processed: unicast [1] 13 bytes.
[INT] Using peer key!
[INT] Successfully set WEP key [1]
[INT] Successfully set the WEP transmit key [1]
```

Configuring and Using WPA

WPA is triggered by a command-line option and is configured by two options in the global section of the configuration file. WPA allows you to specify the type of encryption used for unicast (pairwise) and broadcast or multicast (group) frames. Both options can be set in the configuration file and can take values of wep40, wep104, tkip, ccmp or wrap. At this point, however, only the RC4-based ciphers—WEP and TKIP—work reliably. Set up the two lines of configuration like this:

```
wpa_pairwise_cipher = tkip
wpa_group_cipher = tkip

network-config
{
  . . .
}
```

To use WPA at run time, you must have configured support in the driver for your card as well as the main configuration file. WPA is not simply the new encryption routines of TKIP and it does affect the association process and key distribution. Due to the level of driver support required, you need to specify a driver with the -D option, and you must use a driver that has WPA support compiled in:

```
# xsupplicant -dasic -i ath0 -D madwifi
```

Resources for this article: www.linuxjournal.com/article/8404.

Matthew Gast is the author of the leading technical book on wireless LANs, *802.11 Wireless Networks: The Definitive Guide* (O'Reilly Media). He currently is Director of Consulting Engineering for an advanced wireless systems company, where he helps customers understand new security protocols and standards and how to use them to build secure wireless LANs. He can be reached at matthew.gast@gmail.com, but only when he is close to sea level.

Monarch Furia with AMD Athlon 64 X2 Processors

Monarch Computer announced the availability of the Monarch Furia featuring the AMD Athlon 64 X2 dual-core processor. The new Furia workstations and desktops handle both 32-bit and 64-bit applications. With an on-die dual-core x86 PC processor, Monarch's new workstations offer inter-core communication at CPU speeds, as well as direct access to memory controller and HyperTransport technology.

CONTACT Monarch Computer Systems, 5242 Royal Woods Parkway, Suite 160, Tucker, Georgia 30084, 800-611-0875, www.monarchcomputer.com.

HW400c/2 Communication Controller

The newest addition to SBE's HighWire series of communications, the HW400c/2 is an intelligent PICMG 2.16 CompactPCI I/O processor. It features a 1GHz PowerPC processor, up to 1GB of SDRAM, two PCI Telecom Mezzanine Card (PTMC) sites and Gigabit Ethernet and H.110. Designed to be a blade platform for telecom infrastructure applications, such as media gateways, softswitches and remote node controllers, the two expansion sites are designed to support PTMC Configuration 2 and Configuration 5 modules in addition to standard PMC boards. The core processing architecture on the HW400c/2 is based on the 1GHz Freescale MPC7447A PowerPC processor and Marvell Discovery III system controller. Up to 1GB of ECC DDR memory is supported in addition to on-board Disk-on-Chip Flash filesystem storage.

CONTACT SBE Corporate Headquarters, 2305 Camino Ramon, Suite 200, San Ramon, California 94583, 925-355-2000, sbei.net.

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.

Nokia 770 Internet Tablet

The Nokia 770 Internet Tablet is a dedicated device for Internet browsing and e-mail communications in a pocket-size format. The Nokia 770 features a high-resolution 800 x 480 widescreen display with zoom and on-screen keyboard, making it well suited for viewing on-line content over Wi-Fi. Aside from Wi-Fi, the device also can connect to the Internet utilizing Bluetooth wireless technology via a compatible mobile phone. The 770 runs on the Linux-based Nokia Internet Tablet 2005 Software Edition, which includes many popular open-source technologies. In conjunction with the release of the Nokia 770, the maemo development platform (www.maemo.org) now is available to provide open-source developers with tools and opportunities to collaborate with Nokia on future devices and OS releases in the Internet Tablet category.

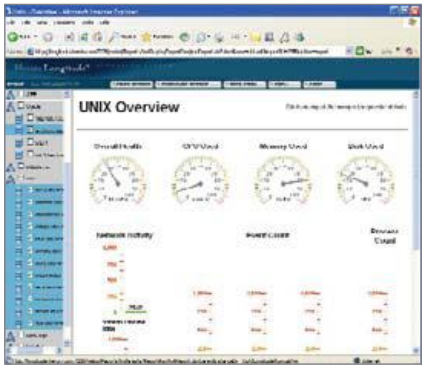
CONTACT Nokia, Keilalahdentie 2-4, FIN-00045 Nokia Group Finland, +358 (0) 7180 08000, www.nokia.com.

PL-01025 1U Embedded Development Platform

WIN Enterprises, Inc., introduced the PL-01025, a high-performance, rack-mountable 1U embedded development platform designed for Internet/network appliance OEMs. Featuring the supplemental processing power of the SafeXcel 184x co-processor, the PL-01025 supports a Pentium M processor and up to 8GB of DDR RAM. It also offers a CompactFlash socket, Gigabit Ethernet and a PCI-X slot. Other features include 10 Gigabit Ethernet (10/100/1000) and four 10/100 Ethernet ports, as well as digital I/O (four in, four out), serial interface and an IDE connector for a 2.5\200\235 HDD.

CONTACT WIN Enterprises, 300 Willow Street South, North Andover, Massachusetts 01845, 978-688-2000, www.win-ent.com.

Heroix Longitude



Heroix's Longitude is an agentless, multiplatform OS and application monitoring and reporting system. Event displays, graphical dashboard views and performance reports and graphs supply information about the overall system so IT personnel can manage performance and capacity issues before IT service levels are affected. Based on industry standards, Longitude is 100% Web-enabled and is equipped with more than 250 prepackaged operational metrics for monitoring the performance of Windows, Linux and UNIX systems, as well as application, Web, database and messaging servers. More than 125 prepackaged reports and the intuitive dashboard allow users to assess an overview of historical performance problems and then drill down to view problem details. Longitude requires little to no configuration and can be up and running in the production environment within 15 minutes.

CONTACT Heroix, 57 Wells Avenue, Newton, Massachusetts 02459, 800-229-6500, www.heroix.com.

Archos PMA400

REVIEWED BY DOVID KOPEL

PRODUCT INFORMATION

Vendor:
Archos

URL:
www.archos.com

Price:
$799 US

THE GOOD

- Lightweight, compact, stylish and well made.
- Removable battery, ten-hour life span.
- Good user interface.

THE BAD

- Poor DivX video syncing.
- Limited media format support.
- Weak Wi-Fi

Until now, people looking to buy a media player have had to choose either speed and mobility or storage and connectivity. The Archos PMA400, however, aims to provide users with all the essentials in one player that previously had been separate entities. The PMA, or Personal Media Assistant, is an MP3 player, PDA and 30GB hard drive all in one. As such, the PMA400 seems to be the first of its kind and the start of a new class of mobile devices.

The PMA400 comes with a tiny Hitachi hard drive that has a roomy 30GB to spare. The display is a 3.5" TFT touchscreen with 320x340 resolution, which is better than your average media player. What really sets the PMA apart, though, is it runs embedded Linux along with Qtopia, the standard GUI for Linux-based PDAs. Qtopia has turned the PMA from a simple MP3 player into a more functional PDA. To top it all off, the PMA is equipped with an internal 802.11b wireless card. When you combine a PDA, storage and the Internet, you have a fairly powerful device.

The PMA has a sleek and elegant design that doesn't detract from its usability; the design is comfortably ergonomic. The removable Li-Ion battery allows the unit to play around four hours of video or ten hours of music.

The PMA is controlled by a four-way directional pad with a center confirmation button. On the outside of the pad is a cancel button that also is used to put the unit into standby mode. It also has a button that scrolls through open windows and takes the user to the home screen and to the menu.

Users should find themselves using the touchpad much less than expected, because nearly everything can be accessed through the navigation pad. I have never been happy using handwriting recognition or an on-screen keyboard, however, which happens to be the case here. For quick input, the stylus isn't terrible, but writing long e-mails or even inputting several contacts can be grueling. I was dissatisfied with the provided stylus, as it was bulky and awkward to use and annoying to take out. Because the stylus is the only means of inputting data I found it to lack the quality it should have.

The Qtopia interface has not changed much since the one used in the Sharp Zaurus 5600, which is good and bad. When booting up cold, not from standby, you may want to find something else to do, as it takes about a minute and a half for Qtopia to get up and running. When you aren't using the unit for a short duration, the screen dims and eventually shuts off. The unit is fairly quick in response, though, and can handle multitasking decently. Coming from a Palm OS device, it is a great feeling that you

can run more than one application at a time. Qtopia is based on Qt, a powerful and fast open-source graphics library for the X Window System made by Trolltech. Qtopia makes the PMA400 a perfect candidate for a superior mobile media center, when it is given the right software.

The Archos bundled software is plentiful and smoothly integrated with Qtopia. The software covers all aspects of media playing and recording. The PMA also is able to handle the playback and recording of both audio and video. It comes with several accessories for input and output, both audio and video. Support for MPEG was fine, but when testing DivX, I experienced massive time gaps between audio and video that seemed to be nearly ten or so seconds. This made watching a movie rather intolerable. After trying several DivX movies, I became fed up. After speaking with Archos representatives about this DivX support problem, they suggested I install the new firmware update, available from the Archos Web site. Sadly, I have noticed no difference since applying the updates.

The PMA400 supports both images and documents. I successfully viewed several PDF files as well as many images through the included applications. The photo application has some nice zooming features that work well with the d-pad controls. The only thing I felt was lacking in terms of the photo software was a slide-show feature.

Overall, I found the image and document support to be disappointing. According to the fact sheet that came with the unit, the PMA400 is supposed to support Microsoft Word, PowerPoint and Excel documents, but it doesn't. I was informed by Archos that the document types are not supported by default. Rather, you must install Qword, Qsheet and Qpresenter. According to the Archos representative with whom I dealt, the applications do not come with the PMA and are optional. However, there is no trace of them on the Web site as of this writing.

The newest and by far most innovative idea that I have seen in a media player is that of Wi-Fi. The PMA400 is equipped with an 802.11b wireless Internet card. This is integrated into the device, and Qtopia has a graphical utility for configuring the wireless network, including up to 128-bit encryption, which I found to be nice. Additionally, you can have the PMA search for available access points. I happened to find the wireless signal on the PMA400 to be horrible compared to other devices. Right on top of the access point it was showing only 11 / 92 for signal strength.

Overall, the support for Internet connectivity in general was fairly poor. There are three ways of connecting the PMA400 to the Internet. You can connect through Wi-Fi, Infrared or USB. As I have mentioned, the Wi-Fi support was nothing to brag about. To get support for the classic wired Ethernet, you must purchase an adapter.

Once I finally established a connection, I wanted to see what the PMA400 had to offer me in terms of Internet applications. Two applications of this

sort are pre-installed on the PMA400, the Opera Web browser and an e-mail application. Personally, I feel that Opera on Qtopia handles scaling poorly and makes viewing fairly difficult. As for the e-mail application, it is rather difficult to compose a message with the tools provided; namely, the PMA does not have a QWERTY keypad. Instead you must use an on-screen keyboard. With a small screen, this type of keyboard can be difficult to use. I also found it frustrating that no word completion feature is offered. The e-mail application was fine for receiving messages, but as I said before, it is not a viable method for composing messages.

When transferring files to the PMA through my LAN, I received an average speed of 100kb/ps, which was even slower than CNET's bandwidth tester over the Internet. That is almost exactly half the bandwidth that I achieved with my laptop. I felt like I was using dial-up when Web browsing with the PMA. For anyone who is serious about using a mobile device for wireless Internet, get a laptop.

The Qtopia Desktop personal information manager really impressed me, however. Once you are connected to the Internet, you can do a network sync to the PMA with little to no hassle. All I needed to do was connect and obtain my IP address, which is found under current network information in the configuration. I then entered the PMA's IP address and was asked to accept or deny access. Apparently, the PMA has a firewall that prevents any intrusion by asking the user for confirmation. After I accepted, I was pleased to see that the single contact I made on the computer was transferred over to the PMA. Although the feature is not essential, it is nice to be able to sync your PMA wirelessly over the network.

The PIM applications offered on the PMA are fairly standard and work as expected. I was able to beam a contact from my Treo 650 to the PMA400 with no problem at all. I also was able to beam a text file, which was supported.

In conclusion, the PMA400 has great potential, but it doesn't succeed at all the ambitious tasks it takes on. It is a great MP3 player and a decent PDA. The really good things about the unit are the Qtopia interface and the music aspects. However, I found its wireless and video capabilities to be severely lacking. The DivX support is sluggish and poorly synced, and photo and document support is not at the level it should be.

Some individuals will find the PMA400 to their liking, but I believe that the unit, on the whole, is only a beginning step toward the next-generation multifunctional device. The Linux back end provides developers with the power to make additional applications for the PMA400 and expand its functionality. However, at the time of this review, the SDK was unavailable for testing. All in all, I suggest that you save your money for a rainy day and something worth the $799.

Dovid Kopel is a longtime supporter, user and developer of free software. He is the project manager of mUnky and now is the COO of namethatJam.com.

LETTERS CONTINUED FROM PAGE 7

Multimedia Lock-in?

It appears that law-related issues are starting to crack down on the Linux multimedia scene. According to the MPlayer site, "Multimedia is a patent minefield." I would like to know where we (the Linux people) stand with the patent rights and all the other hub-a-baloo with DVDs and the like. I think the Linux community would embrace such a write-up. What direction do we see the "Multimedia scene" going for Linux? Are we going to be out cold?

--
Darin Riedlinger

You can create your own media in patent-free formats you can use on any OS. See xiph.org for details. In the USA, the infamous Digital Millennium Copyright Act lets the non-Linux OS vendors lock in their customers with multimedia formats. Join eff.org to support legal reform to let people legally switch OSes without losing access to content they bought.—Ed.

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

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
Order online by 9/30/05
Get $500 Off
Enter PROMO CODE: LJ0905

LEVANTA®
www.levanta.com
1.877.LEVANTA

Memory Ordering in Modern Microprocessors, Part II

Anybody who says computers give only right answers hasn't seen what happens when several SMP processors, each with its own cache, try to get at the same data. Here's how to keep the kernel's view of memory correct, no matter what architecture you're on.

BY PAUL E. MCKENNEY

	Loads Reordered After Loads?	Loads Reordered After Stores?	Stores Reordered After Stores?	Stores Reordered After Loads?	Atomic Instructions Reordered With Loads?	Atomic Instructions Reordered With Stores?	Dependent Loads Reordered?	Incoherent Instruction Cache/Pipeline?
Alpha	Y	Y	Y	Y	Y	Y	Y	Y
AMD64	Y			Y				
IA64	Y	Y	Y	Y	Y	Y		Y
(PA–RISC)	Y	Y	Y	Y				
PA–RISC CPUs								
POWER	Y	Y	Y	Y	Y	Y		Y
SPARC RMO	Y	Y	Y	Y	Y	Y		Y
(SPARC PSO)			Y	Y		Y		Y
SPARC TSO				Y				Y
x86	Y	Y		Y				Y
(x86 OOStore)	Y	Y	Y	Y				Y
zSeries				Y				Y

Table 1. Summary of Memory Ordering

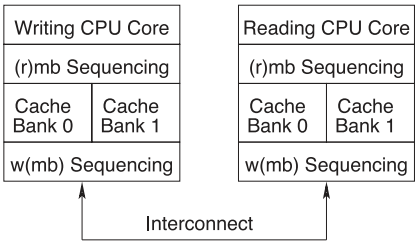


Figure 1. Why smp_read_barrier_depends() Is Required

The first installment of this series was an overview of memory barriers, why they are needed in SMP kernels and how the Linux kernel handles them [August 2005]. This installment gives an overview of how several of the more popular CPUs—Alpha, AMD64, IA64, PA-RISC, POWER, SPARC, x86 and zSeries, otherwise known as IBM mainframe—implement memory barriers. Table 1 is reproduced here from the first installment of this series for reference.

Alpha

It may seem strange to say much of anything about a CPU whose end of life has been announced, but Alpha is interesting because, with the weakest memory-ordering model, it reorders memory operations the most aggressively. It therefore has defined the Linux kernel memory-ordering primitives that must work on all CPUs. Understanding Alpha, therefore, is surprisingly important to the Linux kernel hacker.

The difference between Alpha and the other CPUs is illustrated by the code shown in Listing 1. This smp_wmb() on line 9 guarantees that the element initialization in lines 6–8 is executed before the element is added to the list on line 10, so that the lock-free search works correctly. That is, it makes this guarantee on all CPUs except Alpha.

Alpha has extremely weak memory ordering, such that the code on line 20 of Listing 1 could see the old garbage values that were present before the initialization on lines 6–8.

Figure 1 shows how this can happen on an aggressively parallel machine with partitioned caches, so that alternating cache lines are processed by the different partitions of the caches. Assume that the list header head is processed by cache bank 0 and the new element is processed by cache bank 1. On Alpha, the smp_wmb() guarantees that the cache invalidation performed by lines 6–8 of Listing 1 reaches the interconnect before that of line 10. But, it makes absolutely no guarantee about the order in which the new values reach the reading CPU's core. For example, it is possible that the reading CPU's cache bank 1 is busy, while cache bank 0 is idle. This could result in the cache invalidates for the new element being delayed, so that the reading CPU gets the new value for the pointer but sees the old cached values for the new element.

One could place an smp_rmb() primitive between the pointer fetch and dereference. However, this imposes unneeded overhead on systems such as x86, IA64, PPC and SPARC that respect data dependencies on the read side. An smp_read_barrier_depends() primitive has been added to the Linux 2.6 kernel to eliminate overhead on these systems. This primitive may be used as shown on line 19 of Listing 2. However, please note that RCU code should use rcu_dereference() instead.

It also is possible to implement a software barrier that could be used in place of smp_wmb(), which would force all reading CPUs to see the writing CPU's writes in order. However, this approach was deemed by the Linux community to impose excessive overhead on extremely weakly ordered CPUs, such as Alpha. This software barrier could be implemented by sending interprocessor interrupts (IPIs) to all other CPUs. Upon receipt of such an

IPI, a CPU would execute a memory-barrier instruction, implementing a memory-barrier shoot-down. Additional logic is required to avoid deadlocks. Of course, CPUs that respect data dependencies would define such a barrier simply to be smp_wmb(). Perhaps this decision should be revisited in the future when Alpha fades off into the sunset.

The Linux memory-barrier primitives took their names from the Alpha instructions, so smp_mb() is mb, smp_rmb() is rmb and smp_wmb() is wmb. Alpha is the only CPU where smp_read_barrier_depends() is an smp_mb() rather than a no-op. For more detail on Alpha, see the reference manual, listed in the on-line Resources.

AMD64

Although AMD64 is compatible with x86, it offers a slightly stronger memory-consistency model, in that it does not reorder a store ahead of a load. After all, loads are slow and cannot be buffered, so why reorder a store ahead of a load? Although it is possible in theory to create a parallel program that works on some x86 CPUs but fails on AMD64 due to this difference in memory-consistency model, in practice this difference has little effect on porting code from x86 to AMD64.

The AMD64 implementation of the Linux smp_mb() primitive is mfence, smp_rmb() is lfence and smp_wmb() is sfence.

IA64

IA64 offers a weak consistency model, so that in absence of explicit memory-barrier instructions, IA64 is within its rights to reorder memory references arbitrarily. IA64 has a memory-fence instruction named mf, as well as a half-memory fence modifier to load and store some of its atomic instructions. The acq modifier prevents subsequent memory-reference instructions from being reordered before the acq, but it permits prior memory-reference instructions to be reordered after the acq, as fancifully illustrated by Figure 2. Similarly, the rel modifier prevents prior memory-reference instructions from being reordered after the rel, but it allows subsequent memory-reference instructions to be reordered before the rel.

These half-memory fences are useful for critical sections, as it is safe to push operations into a critical section. It can be fatal, however, to allow them to bleed out.

The IA64 mf instruction is used for the smp_rmb(), smp_mb() and smp_wmb() primitives in the Linux kernel. Oh, and despite persistent rumors to the contrary, the mf mnemonic really does stand for memory fence.

PA-RISC

Although the PA-RISC architecture permits full reordering of loads and stores, actual CPUs run fully ordered. This

Listing 1. Insert and Lock-Free Search

```
 1 struct el *insert(long key, long data)
 2 {
 3       struct el *p;
 4       p = kmalloc(sizeof(*p), GPF_ATOMIC);
 5       spin_lock(&mutex);
 6       p->next = head.next;
 7       p->key = key;
 8       p->data = data;
 9       smp_wmb();
10       head.next = p;
11       spin_unlock(&mutex);
12 }
13
14 struct el *search(long key)
15 {
16       struct el *p;
17       p = head.next;
18       while (p != &head) {
19            /* BUG ON ALPHA!!! */
20            if (p->key == key) {
21                 return (p);
22            }
23            p = p->next;
24       };
25       return (NULL);
26 }
```

means the Linux kernel's memory-ordering primitives generate no code; they do, however, use the GCC memory attribute to disable compiler optimizations that would reorder code across the memory barrier.

Listing 2. Safe Insert and Lock-Free Search

```
 1 struct el *insert(long key, long data)
 2 {
 3       struct el *p;
 4       p = kmalloc(sizeof(*p), GPF_ATOMIC);
 5       spin_lock(&mutex);
 6       p->next = head.next;
 7       p->key = key;
 8       p->data = data;
 9       smp_wmb();
10       head.next = p;
11       spin_unlock(&mutex);
12 }
13
14 struct el *search(long key)
15 {
16       struct el *p;
17       p = head.next;
18       while (p != &head) {
19            smp_read_barrier_depends();
20            if (p->key == key) {
21                 return (p);
22            }
23            p = p->next;
24       };
25       return (NULL);
26 }
```

POWER

The POWER and PowerPC CPU families have a wide variety of memory-barrier instructions:

- sync causes all preceding instructions, not only memory references, to appear to have completed before any subsequent operations are started. This instruction, therefore, is quite expensive.

- lwsync, or lightweight sync, orders loads with respect to subsequent loads and stores, and it also orders stores. However, it does not order stores with respect to subsequent loads. Interestingly enough, the lwsync instruction enforces the same ordering as does the zSeries and, coincidentally, the SPARC TSO.

- eieio, enforce in-order execution of I/O, in case you were wondering, causes all preceding cacheable stores, which are normal memory references, to appear to have completed before all subsequent cacheable stores. It also causes all preceding non-cacheable, memory-mapped I/O (MMIO) stores to appear to have completed before all subsequent non-cacheable stores. However, the stores to cacheable memory are ordered separately from the stores to non-cacheable memory, which, for example, means that eieio does not force an MMIO store to precede a spinlock release.

- isync forces all preceding instructions to appear to have completed before any subsequent instructions start execution. This means that the preceding instructions must have progressed far enough that any traps they might generate either have happened or are guaranteed not to happen. Furthermore, any side effects of these instructions—for example, page-table changes—are seen by the subsequent instructions.

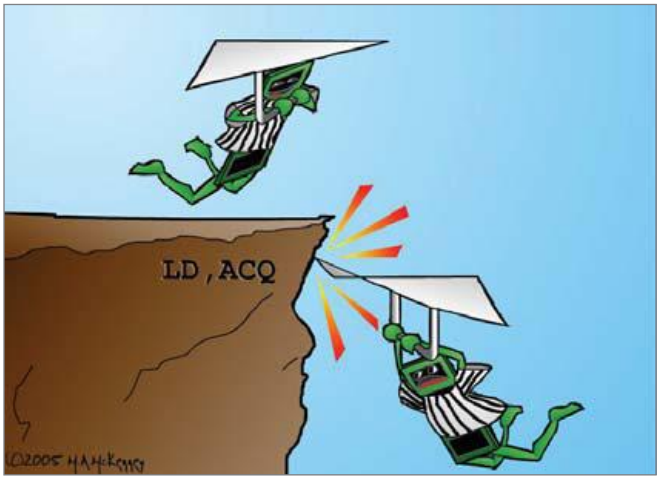


Figure 2. Half-Memory Barrier

Unfortunately, none of these instructions line up exactly with Linux's wmb() primitive, which requires all stores to be ordered. It does not require the other high-overhead actions of the sync instruction. But there is no choice: ppc64 versions of wmb() and mb() are defined to be the heavyweight sync instruction. However, Linux's smp_wmb() primitive cannot be used for MMIO, because a driver must carefully order MMIOs in UP as well as SMP kernels. So, it is defined to be the lighter-weight eieio instruction, which may be unique in having a five-vowel mnemonic. The smp_mb() primitive also is defined to be the sync instruction, but both smp_rmb() and rmb() are defined to be the lighter-weight lwsync instruction.

Many members of the POWER architecture have incoherent instruction caches, so a store to memory is not necessarily reflected in the instruction cache. Thankfully, few people write self-modifying code these days, but JITs do it all the time. Furthermore, recompiling a recently run program looks like self-modifying code from the CPU's viewpoint. The icbi instruction, instruction cache block invalidate, invalidates a specified cache line from the instruction cache and may be used in these situations.

SPARC RMO, PSO and TSO

Solaris on SPARC uses total-store order (TSO); however, Linux runs SPARC in relaxed-memory order (RMO) mode. The SPARC architecture also offers an intermediate partial store order (PSO). Any program that runs in RMO also can run in either PSO or TSO. Similarly, a program that runs in PSO also can run in TSO. Moving a shared-memory parallel program in the other direction may require careful insertion of

memory barriers; although, as noted earlier, programs that make standard use of synchronization primitives need not worry about memory barriers.

SPARC has a flexible memory-barrier instruction that permits fine-grained control of ordering:

- StoreStore: order preceding stores before subsequent stores. This option is used by the Linux smp_wmb() primitive.
- LoadStore: order preceding loads before subsequent stores.
- StoreLoad: order preceding stores before subsequent loads.
- LoadLoad: order preceding loads before subsequent loads. This option is used by the Linux smp_rmb() primitive.
- Sync: fully complete all preceding operations before starting any subsequent operations.
- MemIssue: complete preceding memory operations before subsequent memory operations, which is important for some instances of memory-mapped I/O.
- Lookaside: same as MemIssue but applies only to preceding stores and subsequent loads, and even then only for stores and loads that access the same memory location.

The Linux smp_mb() primitive uses the first four options together, as in:

```
membar #LoadLoad | #LoadStore | #StoreStore | #StoreLoad
```

This fully orders memory operations.

So, why is membar #MemIssue needed? Because a membar #StoreLoad could permit a subsequent load to get its value from a write buffer, which would be disastrous if the write goes to an MMIO register that induces side effects on the value to be read. In contrast, membar #MemIssue would wait until the write buffers were flushed before permitting the loads to execute, thereby ensuring that the load actually gets its value from the MMIO register. Drivers instead could use membar #Sync, but the lighter-weight membar #MemIssue is preferred in cases where the additional function of the more-expensive membar #Sync are not required.

The membar #Lookaside is a lighter-weight version of membar #MemIssue, which is useful when writing to a given MMIO register that affects the value read next from that same register. However, the heavier-weight membar #MemIssue must be used when a write to a given MMIO register affects the value read next from some other MMIO register.

It is not clear why SPARC does not define wmb() to be membar #MemIssue and smb_wmb() to be membar #StoreStore, as the current definitions seem vulnerable to bugs in some drivers. It is quite possible that all the SPARC CPUs that Linux runs on implement a more conservative memory-ordering model than the architecture would permit.

SPARC requires a flush instruction be used between the time that an instruction is stored and executed. This is needed to flush any prior value for that location from the SPARC's instruction cache. Notice that flush takes an address and flushes

only that address from the instruction cache. On SMP systems, all CPUs' caches are flushed, but there is no convenient way to determine when the off-CPU flushes complete, although there is a reference to an implementation note.

x86

The x86 CPUs provide process ordering so that all CPUs agree on the order of a given CPU's writes to memory, so the smp_wmb() primitive is a no-op for the CPU. However, a compiler directive is required to prevent the compiler from performing optimizations that would result in reordering across the smp_wmb() primitive.

On the other hand, x86 CPUs give no ordering guarantees for loads, so the smp_mb() and smp_rmb() primitives expand to lock;addl. This atomic instruction acts as a barrier to both loads and stores. Some SSE instructions are ordered weakly; for example, clflush and nontemporal move instructions. CPUs that have SSE can use mfence for smp_mb(), lfence for smp_rmb() and sfence for smp_wmb(). A few versions of the x86 CPU have a mode bit that enables out-of-order stores, and for these CPUs, smp_wmb() also must be defined to be lock;addl.

Although many older x86 implementations accommodated self-modifying code without the need for any special instructions, newer revisions of the x86 architecture no longer require x86 CPUs to be so accommodating. Interestingly enough, this relaxation comes just in time to inconvenience JIT implementors.

zSeries

The zSeries machines make up the IBM mainframe family previously known as the 360, 370 and 390. Parallelism came late to zSeries, but given that these mainframes first shipped in the mid-1960s, this is not saying much. The bcr 15,0 instruction is used for the Linux smp_mb(), smp_rmb() and smp_wmb() primitives. It also has comparatively strong memory-ordering semantics, as shown in Table 1. This should allow the smp_wmb() primitive to be a no-op, and by the time you read this, this change may have happened.

As with most CPUs, the zSeries architecture does not guarantee a cache-coherent instruction stream. Hence, self-modifying code must execute a serializing instruction between updating the instructions and executing them. That said, many actual zSeries machines do in fact accommodate self-modifying code without serializing instructions. The zSeries instruction set provides a large set of serializing instructions, including compare-and-swap, some types of branches—for example, the aforementioned bcr 15,0 instruction—and test-and-set, among others.

Conclusion

This final installment of the memory-barrier series has given an overview of how a number of CPUs implement memory barriers. Although these overviews should by no means be considered a substitute for carefully reading the architecture manuals (see Resources), I hope that it has served as a useful introduction.

Acknowledgements

I owe thanks to many CPU architects for patiently explaining the instruction and memory-reordering features of their CPUs, particularly Wayne Cardoza, Ed Silha, Anton Blanchard, Tim Slegel, Juergen Probst, Ingo Adlung and Ravi Arimilli. Wayne deserves special thanks for his patience in explaining Alpha's reordering of dependent loads, a lesson that I resisted quite strenuously!

Legal Statement

Resources for this article: www.linuxjournal.com/article/8406.

Paul E. McKenney is a Distinguished Engineer with IBM's Linux Technology Center. He has worked on NUMA and SMP algorithms and, in particular, RCU for longer than he cares to admit. In his spare time, he jogs and supports the usual house-wife-and-kids habit.

Linux Groupware Roundup

Is "organize company to-do list" on your to-do list? Get a jump start on your research into collaboration servers and clients. BY FRANCIS LACHAPELLE AND LUDOVIC MARCOTTE

For the vast majority of its users, the Internet represents e-mail and instant messaging. However, even if it is considered to be a mission-critical service that should perform at an optimal level, e-mail doesn't solve every communication problem. Scheduling a meeting with a few coworkers, for example, can be a tedious task when you don't know the availability of others or the room where the meeting might take place.

Groupware is software that facilitates communication and collaboration through e-mail, calendaring and scheduling, notes, contacts and task management. Good groupware solutions offer not only a Web interface for accessibility from everywhere but also compatibility with native clients on major platforms such as Linux, Apple Mac OS X and Microsoft Windows.

Usually, the more features a groupware solution offers, the less scalable it is. Groupware solutions providing many features generally are suitable for groups composed of no more than a few hundred users. Those that offer basic groupware functionalities, such as e-mail, contact and calendaring, are likely to satisfy the requirements of large deployments, up to thousands of users.

Along with dominant and established products such as Microsoft Exchange, IBM Lotus Notes and Novell GroupWise, excellent proprietary and commercial alternatives are available for Linux: Kerio MailServer, Scalix and Samsung Contact among others. Over the past few years, the Open Source community also has demonstrated a growing interest in messaging, calendaring and scheduling solutions. This article focuses on the status of these open-source efforts. It presents an overview of the various standards related to groupware software and examines the most promising projects being developed by the community.

Drafts and Standards

Today, the most supported and implemented standard in calendaring and scheduling is iCalendar, which defines a common format for openly exchanging calendaring and scheduling information across the Internet. Along with iCalendar, two additional standards were proposed, iCalendar Transport-Independent Interoperability Protocol (iTIP) and iCalendar Message-Based Interoperability Protocol (iMIP). RFC 3283, titled "Guide to Internet Calendaring", summarizes the relationship between the three standards: "iCalendar is the language used to describe calendar objects. iTIP describes a way to use the iCalendar language to do scheduling. iMIP describes how to do iTIP scheduling via e-mail."

iMIP has seen some success but is not used commonly today. A real-time Calendar Access Protocol (CAP) also was proposed, but it eventually expired after only a few implementations of it were put in place. Because it is universally considered to be not a good concept, CAP is being abandoned.

Although a trend of using WebDAV to share and edit iCalendar data emerged from various calendaring software vendors—Apple iCal, Mozilla Sunbird and Novell Evolution—the Internet Task Force published the CalDAV specification. The draft proposes a standard to model calendar events as HTTP resources in iCalendar format. Commitments to support CalDAV have been made by various open-source groupware solutions, but the majority of commercial products still has to adopt the upcoming standard.

More recently, GroupDAV emerged as an effort to create a simple protocol to connect open-source groupware clients to open-source groupware servers. More precisely, GroupDAV focuses on three popular clients: KDE Kontact, Novell Evolution and Mozilla Sunbird. Similar to CalDAV, the proposed model uses HTTP and WebDAV to store groupware data,

such as events and tasks, using the iCalendar standard, but it also stores contacts using the vCard standard.

Back Ends/Web Interfaces

E-mail service probably is the most solicited service in any groupware solution. Most organizations have a solid e-mail system and are interested in adding groupware-type functionalities on top of the existing infrastructure. To this end, the development version of Kolab2 makes heavy use of Cyrus IMAP Server's capabilities, including access control lists, annotations and shared folders. It stores every single object, such as a contact, event, note or task, in an e-mail message in the appropriate object's type folder. Kolab2 provides all groupware features and uses solid open-source server components, including Postfix, Cyrus IMAP Server, OpenLDAP and ProFTPd.

Kolab2 does not include a Web interface beside its administration interface, but connectivity is being added to most of Horde's excellent modules. The Horde Project combines a powerful PHP-based application framework with modules such as the Webmail program IMP, the calendar manager Kronolith and the contact manager Turba.

Installing Kolab2 is relatively easy to do, thanks to OpenPKG, a component that also makes the project deployable on many distributions. Kolab2 does not support CalDAV nor GroupDAV, and adding support for one of these protocols is hard, due to the nature of how objects are stored in Kolab2. In addition, you cannot update a message in IMAP; whenever a modification is done, identity is lost.

Formerly SKYRiX groupware server, OpenGroupware (OGo) is a feature-full groupware solution that sits side by side with an existing e-mail infrastructure. OGo provides group calendars, contacts, tasks, resources, projects and documents management and a Webmail client. OGo also provides GroupDAV support. Built on top of the SOPE application server, OGo has a well-structured architecture. Installation is relatively easy, as binary packages are offered for most distributions.

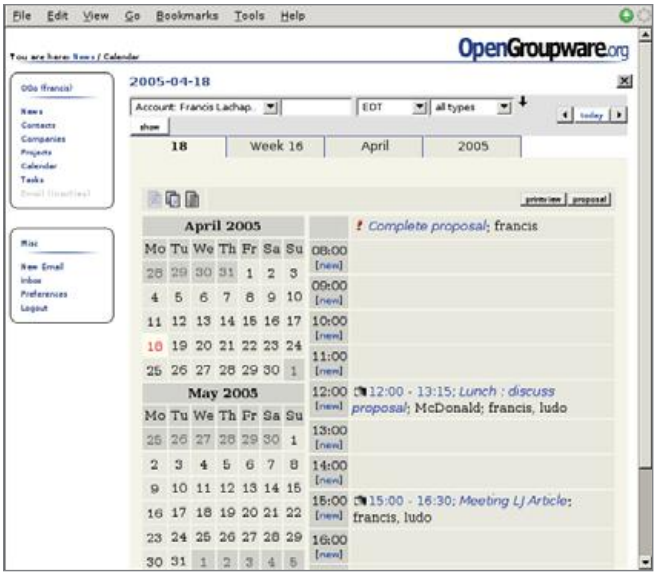


Figure 1. OpenGroupware offers calendars, document management and other features and supports the GroupDAV standard.

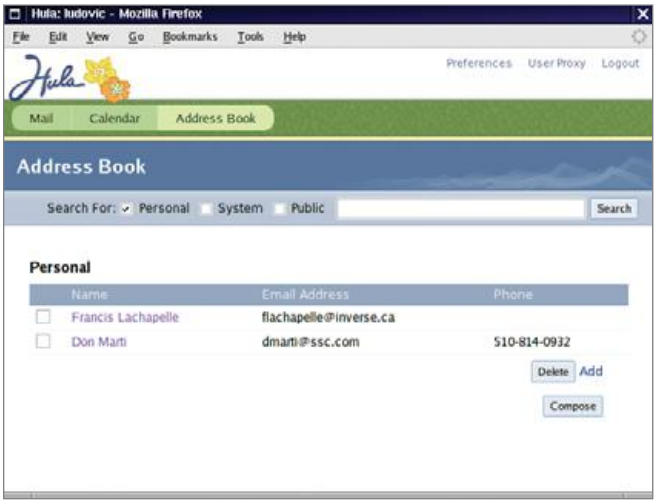


Figure 2. Hula, once Novell NetMail, has a simple, efficient Web interface. The Hula Address Book is shown here.

Formerly SUSE OpenExchange, OPEN-XCHANGE (OX) provides the same kind of functionalities that OGo offers, and it also sits on top of an existing e-mail infrastructure. Built around mainly Java-oriented components, OX provides a rich

Web interface to its groupware features. Although a little behind other projects with regard to client interoperability, OX might be a natural choice for those comfortable with Java technologies.

If you don't have a robust e-mail infrastructure or if you are not particularly tied to it, the Hula and Citadel Projects are interesting groupware solutions. The new Hula Server Project, formerly the proprietary Novell NetMail product, is a complete mail and calendar server. It provides SMTP, POP3, IMAP and calendar services as well as a simple and efficient Web interface. The Hula Server features CalDAV support, and GroupDAV support is being added by Martijn van Beers. The installation and configuration of the Hula Server is easy to do, as packages are available for many distributions and the software offers a rich Web interface for managing all components of the system.

Citadel is a multithreaded groupware server implementing all mail standard protocols, although it can integrate with an existing mail transfer agent. Standard groupware functionalities such as mail, calendar, contacts, notes and tasks are supported. It offers a Web interface through WebCit in addition to a text interface. Citadel also joined the GroupDAV effort and already provides a working implementation.

Moving to a different sector, universities aggressively are integrating portal engines in their infrastructures, especially uPortal. Offering groupware functionalities in the portal is appealing. Projects such as University of British Columbia (UBC) Webmail and UBC Address Book have matured and are well integrated in uPortal. For calendaring services, the University of Washington (UW) Calendar Project can be integrated as a portlet in the portal engine, although the support is preliminary. Support for native clients, such as Novell Evolution or MeetingMaker, also is planned.

Additional projects are worth mentioning but should still be considered experimental: exchange4linux, OpenOffice.org Groupware and its Glow client and Chandler. Well funded by the Mellon Foundation, Chandler eventually could become a key player.

Development activity also is dissipated among a cluster of overlapping projects based on PHP, such as eGroupWare,

Table 1. Groupware Servers and Their Functionalities

	Kolab2	OGo	OX	Hula	Citadel	UW Calendar
Standard						
iCalendar over WebDAV	X	X	X			
CalDAV				X		
GroupDAV		X		X	X	
SyncML		X	X			
Feature						
Contact/Address Book	X	X	X	X	X	
Calendar	X	X	X	X	X	X
E-mail	X	X	X	X	X	
Notes	X				X	
Tasks	X	X	X		X	

phpGroupWare and more.groupware. Despite their impressive number of features, these projects lack maturity and cannot be scaled for enterprise-wide deployments. In addition, most of them don't support clients other than a Web browser.

Table 1 presents the groupware servers described above and lists their respective functionalities. Some of those functionalities currently are in development.

Native Clients

Even if most of the groupware contenders provide a Web interface to every feature they offer, users often prefer a native client. Native clients provide access to standard groupware features, such as contacts, e-mail, notes and calendaring. On Linux, three native groupware clients are taking the lead over others: KDE Kontact, Novell Evolution and Mozilla Thunderbird and Sunbird.

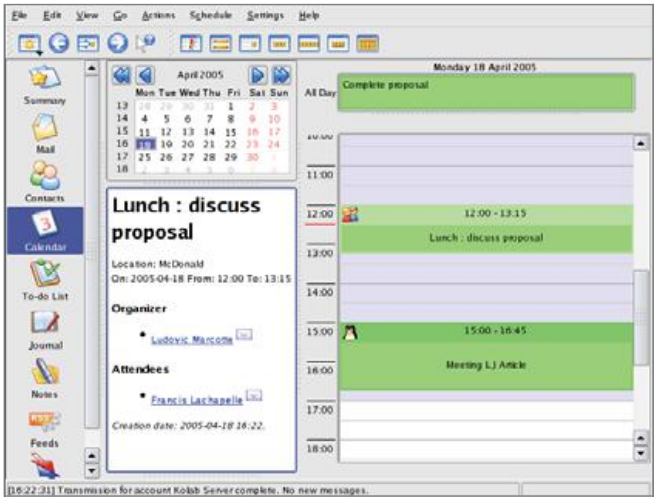


Figure 3. Kontact is a KDE-based native client for Kolab.

Kontact, KDE's personal information management suite, contains e-mail, calendar, contacts, notes and news components. As of the Kontact 1.1 release, included in KDE 3.4, GroupDAV support is included.

Novell Evolution is a popular groupware client that offers e-mail, calendaring, contacts and task management in one application. The Noodle Project aims to improve the compatibility between Evolution and OpenGroupware. The developers recently adopted GroupDAV, which not only will allow Evolution to work with OGo but also with all groupware servers implementing the proposed standard, including Citadel.

The Mozilla Project, with Thunderbird and Sunbird, is coming along nicely with great cross-platform applications. Thunderbird already is a mature e-mail and contacts management application, and Sunbird is maturing quickly. Stelian Pop has started adding GroupDAV support to Sunbird, making interoperability with various groupware possible. There also is an effort called SyncKolab to add Kolab synchronization capabilities to Thunderbird and its calendar extension. This project is progressing rapidly, and Kolab2 support is in the works.

Another client gaining maturity is Aethera, a localized, multiplatform application developed by TheKompany.com. Although currently offering support only for Kolab1 and Citadel, it eventually may support GroupDAV.

Native clients for platforms such as Microsoft Windows and Apple Mac OS X also might become options with regard to the groupware solution. Commercial connectors currently are available for Microsoft Outlook—Toltec Connector for Kolab2, OXlook for OPEN-XCHANGE and ZideLook for OpenGroupware—but the usage of cross-platform open-source clients such as Mozilla Thunderbird and Sunbird certainly is an economically appealing option.

Mobile Clients

Although a Web interface is attractive for accessing groupware-related information, it sometimes can be difficult to have Web access. Most mobile workers have cellular phones or handheld devices that offer contact management, notes and scheduling. The need to synchronize these devices to a groupware product is growing and solutions are emerging.

Part of the GNOME platform, MultiSync is a modular

program to synchronize calendars, contacts and other information between programs on your computer and cellular phones or handheld devices. MultiSync supports Novell Evolution, which can connect to many groupware solutions, as well as many devices such as Palm, Zaurus, PocketPC and many Sony-Ericsson phones.

KDE's universal syncing application, KitchenSync, is similar to MultiSync. Due to their similarity, the two projects are being merged into a new project called OpenSync. Part of the freedesktop.org collaborative zone, the OpenSync Project is creating a new API, libraries and synchronization plugins that eventually will become the standardized synchronization framework used by projects such as GNOME and KDE.

On the other hand, projects such as OpenGroupware and OPEN-XCHANGE support Palm synchronization through the HotSync manager. They now have started to add support for SyncML, an XML-based standard allowing you to synchronize PIM-related information from your mobile device directly with the groupware server.

Conclusion

A proliferation of groupware clients and servers now is available. Good proposed standards, such as GroupDAV and SyncML, need to be adopted by more projects and vendors in order to ease interoperability among native clients, mobile devices and groupware servers. We also should see efforts to merge soon among groupware developers, as there likely are too many solutions available currently.

Scalability remains to be seen, especially for a large amount of users—20,000 users and beyond. Projects such as SOGo, also based on the SOPE application server, address scalability by reducing the features of projects such as OpenGroupware, so they can scale to many thousands of users. This project, which started in August 2004, is promising in this regard.

Migration from existing groupware is another problem, particularly when Microsoft Exchange is involved. The OpenGroupware Project was started to address this issue, and hopefully progress will be made toward this adoption barrier.

In our next article, we will pick one of the groupware servers mentioned in this article and detail the installation and configuration steps required to deploy it as well as the native client's configuration.

Resources for this article: www.linuxjournal.com/article/8333.

Francis Lachapelle (flachapelle@inverse.ca) holds a Bachelor's degree in Computer Engineering from McGill University. He currently is a systems architect for Inverse inc., an IT consulting company located in downtown Montréal.

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

Native XML Data Storage and Retrieval

A new generation of databases creates a new set of decisions and several full-featured ways to build queries. BY GEORGE FEINBERG

The design and implementation trade-offs within a native XML database make a significant impact on the performance, scalability and features available to applications that use it. This article focuses on the granularity of stored XML documents and indexing as two of the most critical design considerations. Berkeley DB XML from Sleepycat Software (www.sleepycat.com/products/xml.shtml) is the basis for this discussion.

The basic functions of an XML database are to store documents, query over documents and handle query results. Of course, indexes are required to obtain acceptable query performance.

In a relational database, pieces of a relational table are stored, queries are SQL and results are tabular. This abstraction and standardization is useful from an application developer's perspective. Developers have less visibility into precisely how documents are stored and indexed and how a query can leverage the combination of storage format, indexes and query language to answer a question quickly.

The same concepts exist in a native XML database, such as Berkeley DB XML. In this case, the data is the XML document and the query may be an XPath or XQuery expression. The results may be XML documents, DOM, SAX or a proprietary form. Within a native XML database, mechanisms for storage, indexing and querying are not obvious from the perspective of an application developer, yet they are critical to the function, performance and scalability of the overall system.

A native XML database exposes a logical model of storing and retrieving XML documents; however, its internal storage model may not be equivalent to the document. Indexing is a crucial component of any database. Without intelligent indexing, a database is little better than a filesystem for information retrieval. Query processing builds on both storage format and indexes but is beyond the scope of this article.

Storage Formats and Granularity

Most native XML databases are oriented toward storing XML documents, where a key issue is the granularity with which the document is stored. In database terms, granularity can be described in several different ways: external access, internal addressability and concurrency.

A distinction is made between access granularity and addressability. Addressability refers to objects that can be named and accessed directly, without navigation, within the system. Access may be provided through a DOM to a system with an addressable granularity of an XML document, by parsing the document. In this sense, access granularity is user-visible, while addressability is an internal concept. Concurrency means how objects can be modified concurrently, if such a feature is supported.

Intact Document Storage

There are two major choices in terms of how to store a document—intact or not intact. Systems that store XML documents intact usually parse the XML in order to ensure it is well formed and valid but otherwise store documents unchanged. This is useful for applications that require retrieval of the entire byte-for-byte document or for round tripping. Furthermore, for relatively small documents that tend to be retrieved and processed whole, such a system is ideal. The major issue for intact document storage is how to address target documents within a collection of documents. There are two primary mechanisms to do this: a unique identifier, such as name or document ID, or a query expression, such as XQuery. The first results in exactly one document, whereas the latter may return many documents in a result set.

For a large collection, it must be possible to target a small set of result documents in a query. For intact document storage, this implies an indexing mechanism. If a document is parsed upon insertion into a collection, it can be indexed as well, based on the system's indexing specifications. Indexes in this type of system use document granularity addressing. It is desirable to avoid parsing documents in order to resolve a query. Additional parsing can be avoided if the query can be answered definitively from indexes and the access granularity desired by the application is at the document level, as opposed to DOM granularity access.

A clear disadvantage of intact document storage is that for certain applications and queries, it can take a long time and a large amount of memory to process a request. This is mostly due to the need to parse documents to satisfy a query. Optimizations, such as references to offsets within a document, can be made, however, for read-only documents.

The advantages of intact document storage include its simplicity and byte-for-byte round tripping. Berkeley DB XML has an option to store documents intact.

Fine Granularity Storage

Some native XML databases, such as Berkeley DB XML, store documents with granularity finer than the document. The properties of such systems include: addressability is subdocument level, access granularity is subdocument level and concurrency granularity may or may not be finer than document level.

Storing documents in pieces offers a number of advantages, including:

- Ability to reference an element or other object within a document directly.
- Ability to retrieve partial documents without parsing.
- Efficient querying, without parsing, by materializing only those parts of a document necessary to evaluate the query.
- Ability to modify a small piece of a large document.

The decision to store documents in pieces results in more choices:

- Degree of round tripping supported, if any.
- What information is stored or the data model of the storage.
- Granularity of addressability.
- Support for partial document modification, without rewriting the entire document.
- Physical format of information.

Fine-grained document storage systems must choose the degree of round tripping supported if it is a requirement to be able to return the original document, byte for byte. Virtually any decomposition of a document for storage results in loss or change of information, such as reordering of attributes, or a change in the XML declaration. This is because there is not a 1:1 mapping from XML infoset to bytes in a document. That is, there are bytes within an XML document that are not considered relevant to the infoset and, therefore, may not even be passed through by a parser.

To support round tripping, a fine-grained document storage system must track entity references that are expanded during parsing, as well as ignorable white space and namespace prefix mappings. Such mechanisms are unimportant in terms of querying and retrieval of partial documents, but for some applications, they can be critical for document serialization. Because the degree of round tripping implies extra cost, some systems export configuration options to determine handling of these issues.

Data Model

Intact document storage has the vastly simplifying advantage of being unconcerned with the data model of the XML documents it stores. Fine-grained document storage must decide on the data model, which is tied closely to query processing and query language support. For example, XQuery's data model is typed, and type information can appear in XQuery expressions. XPath 1.0 expressions, however, are not richly typed, so no additional type information is necessary.

A simple example of the data model issue is DOM vs. XQuery. The DOM is relatively simple. Where most every object is a node, some nodes have names, some have values and some have children and siblings. The DOM essentially is a tree with little semantic information, and virtually all of its information is contained in the XML document itself. Conversely, the XQuery data model is typed. XQuery does support simple, well-formed XML; however, it also supports type information, as obtained from a schema-validated document, where the schema information comes from outside the document.

It is possible to choose a storage data model equivalent to the XML infoset or DOM, but then the powerful type facilities of XPath 2.0 and XQuery 1.0 are not fully available. A schema-validated document has type information available at the time it is parsed and validated. A system where parsing, validation and querying occur at the same time has no problem obtaining type information to satisfy the query. However, in a fine-grained storage system, the parsing and query events are not related. This means that at the time of the query, type information must be found if it is to be used for the query. There are several choices for how a system can implement types:

- Store type information with each document and typed object and materialize it for querying.
- Store references to relevant schema files and reload (parse) them for querying.
- Map each type to the nearest atomic type in the XML Schema recommendation and store that information.
- Don't support type information at all, which limits queries and forces them to use their own, complex type definitions.

Granularity of addressability is tied closely to the data model. At one extreme is the choice of DOM objects as the addressable unit. This means that each DOM node, be it a document, element or attribute value, is an addressable and separately stored object. Although simple, this approach is quite expensive in terms of memory, disk space and CPU. There are other, coarser-grained solutions. One is to use the element as an addressable unit and associate its attributes and child text nodes. Another is to address elements and text nodes and associate attributes with elements. The former may be better for locality of reference, if an element and its attributes and text nodes are likely to be referenced together.

Native XML databases that store documents as fine-grained nodes must assign addressable node identifiers (node IDs) to addressable units. Node IDs are used to retrieve specific nodes during processing. When it comes to physical storage, size matters. Smaller nodes and node IDs mean better locality of reference and fewer disk accesses to read and write data.

Berkeley DB XML stores nodes in a B-tree, where node IDs are allocated in document order, which also is an iteration order on the B-tree. This means that once a node is located, serialization or child navigation can occur by way of iteration rather than by additional lookup operations.

With the appropriate sorting/comparison function, a node ID that is a B-tree key can take on many physical forms. It can be as simple as an integer, or it can be a complex array or string. Node numbering is one of the more interesting and important design choices in a native XML database. There are node numbering schemes that have the ability to allow insertion and removal of arbitrary nodes without renumbering and to allow query-relevant operations to be performed based solely on node numbers and indexes, eliminating node lookups.

Berkeley DB XML uses a numbering scheme that allows some direct relationship comparisons and attempts to minimize the need to materialize nodes for navigation. The scheme also avoids renumbering when a document is modified partially.

One advantage of fine-grained storage is the ability to modify some parts of documents without touching the rest. There is a significant performance and scalability benefit in such "surgical" changes; however, it can be difficult to do efficiently. Many systems do not support partial modification of documents, and if they do, it is only through a well-defined interface such as XUpdate, as opposed to a direct DOM manipulation.

A partial modification can render a document invalid, or worse, malformed. Re-parsing for validation, however, negates much of the benefit of partial modifications. Insertion or removal of an addressable object, such as an element, affects the system's node numbering scheme, as described above. Indexes also are affected and must be updated. A database may choose to revalidate or parse after a modification or allow the application to request it explicitly.

Fine-grained document storage has a disadvantage in serialization of an entire document. In this situation, an iterator must traverse the addressable pieces of the document. If this is a common operation, it may be worth optimizing or caching the serialized document for reuse, which creates a possible concurrency problem. Document serialization can be optimized by maintaining addressable units in document order, keeping names in stored nodes rather than name IDs and using coarser granularity, which leads to fewer objects retrieved from disk.

Indexing

Proper specification and use of indexes can increase query speeds by orders of magnitude. However, indexes consume space on disk and in the cache—a classic space versus speed trade-off. Under certain situations, the presence of indexes slows operations. When frequently updating indexed data, time spent re-indexing can offset the benefit of indexed access.

The data models for querying XML imply that virtually all indexes deal with elements, attributes and their respective text content, as well as possible data types represented by their value strings. However, there is no standard or convention regarding how to specify indexes or even what is indexed and how. Different XML databases have made different choices regarding indexes in these areas:

- Index Type—structure, value, full-text.
- Index Scope—document, collection.
- Index Target—document, node.
- Index Control—automatic, voluntary, required.

Index Type

Structural indexes are used for tracking structure and path information, such as "track existence of all element nodes with the path /a/b/c" or "track all paths to the node c." Such indexes are useful for navigational portions of queries. Some indexes reduce the result set to a smaller set of possible results, rather than give a single definitive result. For example, the index above that tracks all paths /a/b/c can be positive about its answer to the query /a/b/c. The index that tracks all paths to c cannot be definite, because it also contains entries for paths such as /e/f/c.

Value indexes are used to track all values for specific elements or attributes. A value index on the element "color" would have an index entry for every separate instance of color and would be useful for a query such as `//color[.="green"]`. In addition, value indexes may be typed so that comparisons can be performed correctly. The typed data model of XPath 2.0 and XQuery 1.0 brings a long list of potential data types from the XML Schema recommendation, such as xs:date, xs:time and various numeric formats. Support for typed indexes allows applications to use them directly rather than modify their content to map, for example, xs:datetime to integer, so that range-

based comparisons can be used.

Full-text indexing is a large topic unto itself. There is a working draft for full-text extensions to XQuery, but it is not yet in general use. Some native XML database products implement what they call full-text indexing, which minimally is a word index over a document. Because there is no standard, a full-text index requires a proprietary query language or extension as an interface.

Index Scope

Most native XML databases store documents in a collection. The scope of a given index could be collection-wide or it could be restricted to a single document. A native XML database system can choose the index scope it implements. Queries against a collection can return documents or sets of nodes within documents. In order to support efficient restriction of a query to a manageable set of documents, the system must support indexes at the collection scope. This does not mean that it is not also possible to have indexes at the document scope, which contain entries that apply only to a given document.

Index Target

Related to scope is the target or the object referenced by an index entry. It can be a document or an object within a document. An index is capable of pointing down to the addressable unit in the system, but such granularity is not always necessary and can be expensive. Because navigational operations within a document stored with fine granularity are not as expensive as those used for intact document storage, due to parsing, it can be sufficient to return the document element for further navigation. Although this is possible, it is the case that most database systems with fine-grained document storage reference directly to nodes in indexes rather than to the containing documents.

Index Control

Another dimension of index type is how indexes are specified. Voluntary indexes are specified explicitly by an interface to the system. These indexes allow for some experimentation to find the minimal useful set of indexes. Some systems have automatic indexes, where a well-defined set of indexes always is created, except for those that are disabled explicitly, by way of configuration or interface. The system also may have required indexes, which cannot be disabled because they are necessary for proper functioning of the system.

Summary

This article has highlighted the importance of storage granularity and indexing within the design of a native XML database. These core choices drive the performance, scalability and features available within the system.

George Feinberg is the architect for Sleepycat Software's Berkeley DB XML. Prior to that, Feinberg was one of the architects of the eXcelon native XML database, now called XML Information Server (XIS) and owned by Progress Software. He was eXcelon's representative to the W3C and the XML Schema working group. Feinberg's previous experience includes serving as an operating system designer and developer for the Open Software Foundation (now The Open Group), Hewlett-Packard and a storage system startup.

A System Monitoring Dashboard

This simple set of shell scripts keeps you informed about disks that are filling up, CPU-hog processes and problems with the Web and mail servers.

BY JOHN OUELLETTE

For about a year, my company had been struggling to roll out a monitoring solution. False positives and inaccurate after-hours pages were affecting morale and wasting system administrators' time. After speaking to some colleagues about what we really need to monitor, it came down to a few things:

- Web servers—by way of HTTP, not only physical servers.
- Disk space.
- SMTP servers' availability—by way of SMTP, not only physical servers.
- A history of these events to diagnose and pinpoint problems.

This article explains the process I developed and how I set up disk, Web and SMTP monitoring both quickly and simply. Keeping the monitoring process simple meant that all the tools used should be available on a recent Linux distribution and should not use advanced protocols, such as SNMP or database technology. As a result, all of my scripts use the Bash shell, basic HTML, some modest Perl and the wget utility. All of these monitoring scripts share the same general skeleton and installation steps, and they are available from the *Linux Journal* FTP site (see the on-line Resources).

Installing the scripts involves several steps. Start by copying the script to a Web server and making it world-executable with `chmod`. Then, create a directory under the root of your Web server where the script can write its logs and history. I used webmon for monitor_web.sh. The other scripts are similar: I used smtpmon for monitor_smtp.sh and stats for monitor_stats.pl. monitor_disk.sh is different from the others because it is the only one installed locally on each server you want to monitor.

Next, schedule the scripts in cron. You can run each script with any user capable of running wget, df -k and top. The user also needs to have the ability to write to the script's home. I suggest creating a local user called monitor and scheduling these through that user's crontab. Finally, install wget if it is not already present on your Linux distribution.

My first challenge was to monitor the Web servers by way of HTTP, so I chose wget as the engine and scripted around it. The resulting script is monitor_web.sh. For those unfamiliar with wget, its author describes it as "a free software package for retrieving files using HTTP, HTTPS and FTP, the most widely used Internet protocols" (see Resources).

After installation, monitor_web.sh requires only two choices for the user, e-mail recipient and URLs to monitor, which are labeled clearly. The URLs must conform to HTTP standards and return a valid `http 200 OK` string to work. They can be HTTP or HTTPS, as wget and monitor_web.sh support both. Once installed and run the first time, the user is able to get to localhost/webmon/webmon.html and view the URLs, the last result and the history in a Web browser, as they all are links.

Now, let's break down the script; see monitor_web.sh, available on the *LJ* FTP site. First, I set all the variables for system utilities and the wget program. These may change on your system. Next, we make sure we are on the network. This ensures that if the server monitoring the URLs goes off-line, a massive number of alerts are not queued up by Sendmail until the server is back on-line.

As I loop through all the URLs, I have wget connect two times with a timeout of five seconds. I do this twice to reduce false positives. If the Web site is down, the script generates an e-mail message for the recipient and updates the Web page. Mail also is sent when the site is back up. The script sends only one message, so we don't overwhelm the recipient. This is achieved with the following code:

```
wget $URL  2>>   $WLOG

if (( $? != 0 ));then

  echo \<A HREF\="$URL"\>$URL\<\/A\> is down\

          $RTAG $EF.\
          $LINK Last Result $LTAG  >> $WPAGE

  if [[ ! -a down.$ENV ]];then

        touch /tmp/down.$ENV
        mail_alert down

  else
         echo Alert already sent for $ENV \
         - waiting | tee -a $WLOG
  fi

fi
```

I have included the HTML for green and red text in the script, if you choose not to use graphics. Again, the full script is available from the *Linux Journal* FTP site.

With the Web servers taken care of, it was time to tackle disk monitoring. True to our keep-it-simple philosophy, I chose to create a script that would run from cron and alert my team based on the output of `df -k`. The result was monitor_disk.sh. The first

Do I Have That Perl Module Installed?

An easy way to check whether you have any Perl module installed is by issuing this from the command line:

```
$ perl -e "use Net::SMTP";
```

If nothing prints, you have that module installed. If you're missing the module, you get an error that looks like this:

```
$perl -e "use Net::OTHER";
Can't locate Net/OTHER.pm in @INC (@INC contains:
/usr/lib/perl5/5.8.3/i386-linux-thread-multi /usr/lib/perl5/5.8.3
/usr/lib/perl5/vendor_perl/5.8.0 /usr/lib/perl5/vendor_perl .) at -e line
1.
BEGIN failed--compilation aborted at -e line 1.
```

This error indicates a lack of the module in question.

real block of code in the script sets up the filesystems list:

```
FILESYSTEMS=$(mount | grep -iv proc |\
grep -iv cdrom | awk'{print $3}')
```

I ignore proc and am careful not to report on the CD-ROM, should my teammates put a disk in the drive. The script then compares the value of Use% to two values, THRESHOLD_MAX and THRESHOLD_WARN. If Use% exceeds either one, the script generates an e-mail to the appropriate recipient, RECIPIENT_MAX or RECIPIENT_WARN. Notice that I made sure the Use% value for each filesystem is interpreted as an integer with this line:

```
typeset -i  UTILIZED=$(df -k $FS | tail -1 | \
awk '{print $5}' | cut -d"%" -f1)
```

A mailing list was set up with my team members' e-mail addresses and the e-mail address of the on-call person to receive the critical e-mails and pages. You may need to do the same with your mail server software, or you simply can use your group or pager as both addresses.

Because our filesystems tend to be large, about 72GB–140GB, I have set critical alerts to 95%, so we still have some time to address issues when alerted. You can set your own threshold with the THRESHOLD_MAX and

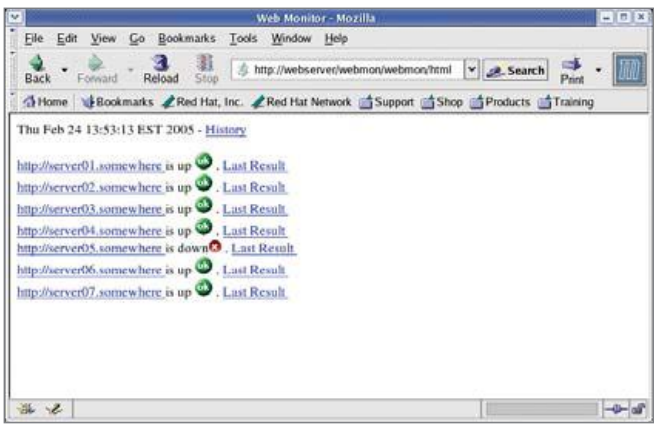


Figure 1. monitor_web.sh in action. Run the script from cron to regenerate this page as often as needed.

THRESHOLD_WARN variables. Also, our database servers run some disk-intensive jobs and can generate large amounts of archive log files, so I figured every 15 minutes is a good frequency at which to monitor. For servers with less active filesystems, once an hour is enough.

Our third script, monitor_smtp.sh, monitors our SMTP servers' ability to send mail. It is similar to the first two scripts and simply was a matter of finding a way to connect directly to a user-defined SMTP server so I could loop through a server list and send a piece of mail. This is where smtp.pl comes in. It is a Perl script (Listing 1) that uses the NET::SMTP module to send mail to an SMTP address. Most recent distributions have this module installed already (see the Do I Have That Perl Module Installed sidebar).

monitor_smtp.sh updates the defined Web page based on the success of the transmission carried out by smtp.pl. No attempt is made to alert our group, as this is a trouble-shooting tool and ironically cannot rely on SMTP to send mail if a server is down. Future versions of monitor_smtp.sh may include a round-robin feature and be able to send an alert through a known working SMTP server.

Listing 1. smtp.pl tests outgoing mail through the SMTP server.

```
#!/usr/bin/perl -w
#
# Title  :    smtp.pl
# Author :    John_Ouellette@yahoo.com
# Files  :    smtp.pl
# Pupose :    Send email through SMTP server
#             Called from monitor_smtp.sh
#
# Submit as

use Net::SMTP;

my $rcpt    = $ARGV[2] || 'mygroup@somewhere';
my $sender  = $ARGV[1] || 'root@host01';
my $host    = $ARGV[0];


#Start Script

my $smtp =Net::SMTP->new($host, Debug => 1);
my $input="test msg for server $host";

$smtp->mail("$sender");
$smtp->to("$rcpt");
$smtp->data();
$smtp->datasend("To: $rcpt\n") ;
$smtp->datasend("From: $sender\n") ;
$smtp->datasend("Subject: $host test\n") ;
$smtp->datasend("$input");
$smtp->dataend();
$smtp->quit;
```

Finally, we come to our stats script, monitor_stats.pl. This script logs in to each host and runs the commands:

```
df -k
swapon -s
top -n 1 | head -n 20
hostname
uptime
```

It then displays the results in a browser (Figure 2) and saves the result in a log, again sorted by date on the filesystem. It serves as a simple dashboard to give quick stats on each server.

The benefit of this monitoring design is threefold:

1. We have a history of CPU, disk and swap usage, and we easily can pinpoint where problems may have occurred.

2. Tedious typing to extract this information for each server is reduced. This comes in handy before leaving work to resolve potential problems before getting paged at night.

3. Management quickly can see how well we're doing.

We are using the insecure rsh protocol in this script to show you how to get this set up quickly, but we recommend that you use SSH with properly distributed keys to gain security.

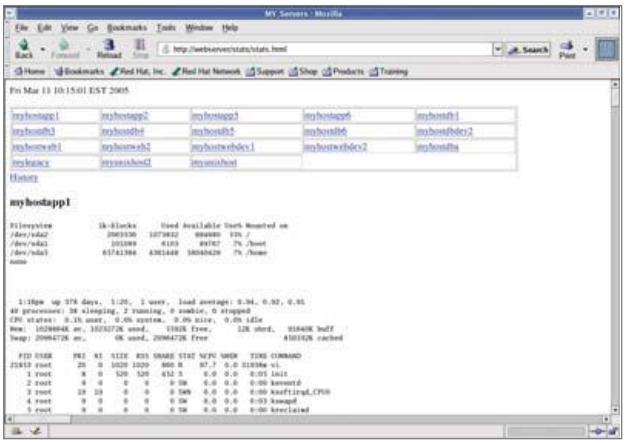


Figure 2. monitor_stats.pl in Action

Conclusion

With the use of this new system monitoring dashboard, my team's productivity has increased and and its confidence in monitoring has soared, because we no longer are wasting time chasing down false positives. A history of system performance has been a real time saver in diagnosing problems. Finally, easy installation allows users with basic skills to conquer a complex system administration problem in one business day.

Resources for this article: www.linuxjournal.com/article/8269.

John Ouellette is a system administrator with nine years of experience in Microsoft Windows NT and UNIX. He believes the command line is king and loves chicken parmigiana. He can be reached at john_ouellette@yahoo.com.

The Free Software Foundation at 20

Development projects and legal groundwork both are essential to protecting your freedom to use your computer your way. **BY PETER BROWN**

This year we celebrate the 20th anniversary of the birth of the Free Software Foundation (FSF). You might think that we would be satisfied that free software has become so popular during that time. Indeed, many of us now can use an operating system and software tools that are free. But there is much work to do and many threats to defend against—threats posed by software patents, treacherous computing, hardware with secret specifications and attractive but non-free software development platforms such as Sun's Java.

Much of what the FSF does today is based on the fundamental principle that freedom is the most important goal we seek. Having good software, as the advocates of "open source" say they aim for, is not enough.

The FSF ethos, engendered by its founder Richard Stallman, has been to call upon developers and users to demand freedom. This has inspired many in our community to achieve tremendous things, and the FSF today is surrounded by a strong community of support.

The FSF receives the bulk of its funding from its Associate Membership. You can become a member of the FSF for an annual sum of $120 US or $60 US if you're a student. As a member you receive an FSF membership card, which is a credit-card-size bootable CD-ROM distribution. You also receive an annual membership gift, the FSF news bulletin, e-mail forwarding and the knowledge that you help fund our core work.

The FSF is the home of the GNU Project, which is the project to build a UNIX-like operating system that is free to all its users. The creation of this GNU operating system, commonly using the Linux kernel, was possible thanks to the creative and ethical hackers that we refer to as the GNU Maintainers and GNU Developers. Today, much work continues to develop and maintain the GNU Project and document its constituent parts. The FSF provides the framework and resources for this ongoing effort.

Beyond the GNU Project, the Free Software community has developed a vast array of free software tools and applications. To promote this software and make it available to anyone for free, we maintain the FSF Free Software Directory (see the on-line Resources). The Directory is a complete listing of all the stable free software programs and now contains more than 4,000 entries. The Directory is the de facto portal for the distribution of free software worldwide, and it was built without commercial advertising.

The FSF holds the copyright on major parts of the GNU/Linux operating system. We carefully collect and process contributers' legal paperwork and register our copyrights with the US Library of Congress' Copyright Office. We hold these copyrights so that we can take action to keep GNU/Linux free. We undertake this work through what we call the FSF Licensing and Compliance Lab. As the author and guardian of the GNU General Public License, the FSF provides detailed resources about free software licensing and takes action when we receive your reports of license violations. So, if you are developing free software and need some help with your licensing, check out our resources. If you see a GPL violation, please read our reporting guidelines. If you still have questions, you can contact us at licensing@fsf.org.

The FSF seeks to encourage the adoption of free software and not exclude anyone in the process. We encourage businesses to use and develop free software and contribute back to the community. Many corporations now develop free software and have profitable business models based on it. That is good, but it is important to remember that only human beings can be trusted to value the ethics of free software—we mustn't depend upon business interests to look after us.

Organizing for free software is one of the major elements of the work of FSF President Richard Stallman. Richard travels constantly with a busy schedule of speaking engagements in an effort to educate and warn people of the threats of proprietary software. Freedom has to be fought for, and one of the major problems we face is the fact that so many still do not perceive the threats.

FSF's message and insistence on freedom over the practical is hard for some to agree with and is controversial in an age when many believe freedoms are for sale. Therefore, we depend heavily on those that do get it to join us and help fund our work.

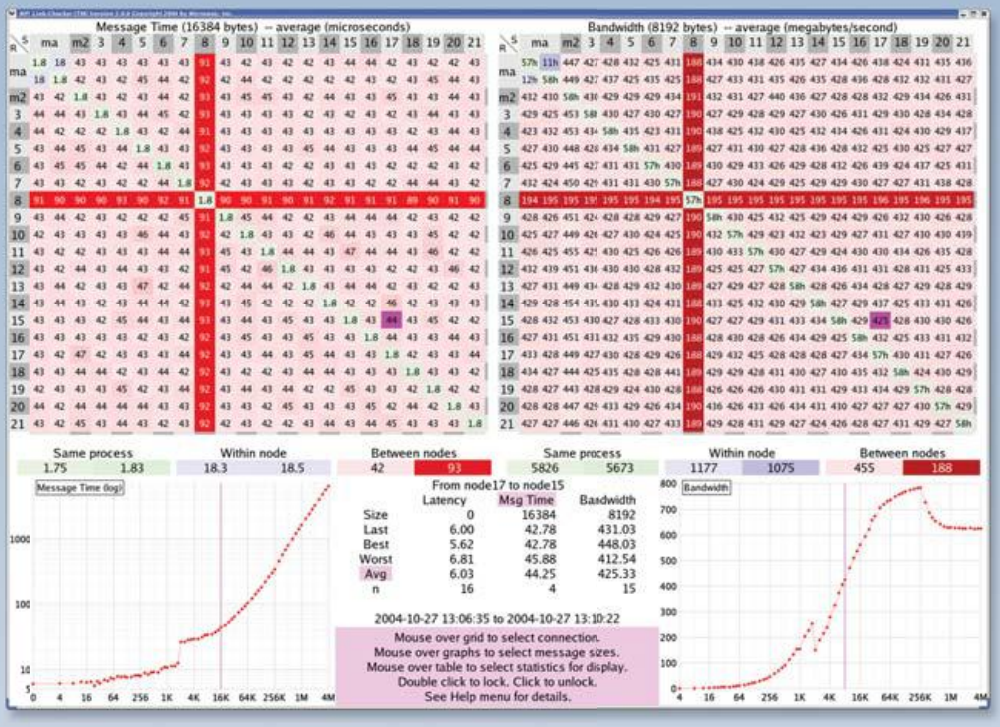
You also can help by working on our high-priority projects: GNU Compiler for Java, GNU Classpath, LinuxBIOS and GPLFlash. Or, make your voice heard in Europe against software patents and in the US against the extension of the reach of the Hague Treaty.

Happy anniversary and happy hacking to all who got us this far!

Resources for this article: www.linuxjournal.com/article/8409.

Peter Brown has worked at the FSF since 2001 as manager of the FSF Licensing and Compliance Lab. He became the Executive Director in 2005 and previously worked as a director of *New Internationalist Magazine*.

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™