

LINUX JOURNAL

Since 1994: The Original Magazine of the Linux Community

JULY 2006 | ISSUE 147 | www.linuxjournal.com

Ruby Everywhere, Including Rails

Getting Started with Ruby

Ruby the Enterprise Glue

Rails and More Rails

Ruby Has Many Gems

UNDER-ICE
Sonar
Visualizations

Ruby on Rails creator
<< David
Heinemeier
Hansson

PLUS

Creating a Searchable Site

Mashups
Game Rubies



Let's go **Xtreme**

Introducing Appro XtremeServers & Workstation with 8 DIMM Sockets per CPU

- 2-way or 4-way, Single or Dual-Core AMD Opteron™ processors
- **Largest memory capacity - 8 DIMM Sockets per CPU, up to 128GB**
- PCI-Express technology to increase I/O bandwidth and reduce system latency
- Outstanding Remote Management – IPMI 2.0 compliant
- Cable-free design, ready to run, simple to install, service and maintain
- Support for Windows® or Linux OS
- Ideal for memory-intensive and I/O-intensive applications



1U / 2U / 3U Servers
and Workstation

AMD Opteron™ Processors - AMD64 dual-core technology reduces memory latency and increases data throughput
- Dual-core processors with Direct Connect Architecture deliver the best performance per watt with little or no increase in power consumption or heat dissipation.

The Power of Being There.[®]

Times two.

Discover how the combined power of Avocent and Cyclades IT infrastructure management solutions can take you and your data center to the next level. KVM, serial and power – all over IP. Plus, Intelligent Platform Management Interface (IPMI) and embedded KVM. The Power of Being There[®]. ***Times two.***



Avocent.[®]



An Avocent[®] Company



Visit www.avocent.com/powerx2

Avocent, the Avocent logo, The Power of Being There and Cyclades are registered trademarks of Avocent Corporation or its affiliates. All other trademarks or company names are trademarks or registered trademarks of their respective companies. Copyright © 2006 Avocent Corporation.

CONTENTS

JULY 2006
Issue 147

Ruby

FEATURES

44 OPINIONS ON OPINIONATED SOFTWARE

The creator of Ruby on Rails doesn't rail but opines.

Kevin Bedell

46 INTRODUCTION TO RUBY

New to Ruby? Here's the primer you're looking for.

Reuven M. Lerner

52 RUGBYGEMS

A rich repository of modules called RubyGems awaits you.

Dirk Elmendorf

58 RUBY AS ENTERPRISE GLUE

How to pull together a heterogenous environment with Ruby.

Maik Schmidt

ON THE COVER

- Ruby on Rails Creator David Heinemeier Hansson, p. 44
- Getting Started with Ruby, p. 46
- Ruby as Enterprise Glue, p. 58
- Rails and More Rails, p. 66
- Ruby Has Many Gems, p. 52
- Under Ice Sonar Visualizations, p. 82
- Creating a Searchable Site, p. 78
- Mashups, p. 22
- Game Rubies, p. 30

ON THE COVER: PHOTO BY ANNA KNOTT

The competition doesn't stand a chance.



If you base deployment decisions on performance and price, Coyote Point's for you. We've cornered that market.

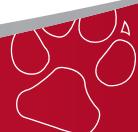
To prove it we asked The Tolly Group to evaluate our E350si application traffic manager against the competition. The results speak for themselves.

Throughput? Almost 40% more than others in our space. Cost of transactions per second? Up to four times less. Connection rate? In some cases, one-sixth the cost. One-sixth! And we're told Coyote Point is the #1 choice for today's open source networks.

But don't just take our word for it. Get the facts. Call 1.877.367.2696 or write info@coyotepoint.com for your free copy of the full Tolly Report.



www.coyotepoint.com



CONTENTS

JULY 2006

Issue 147

COLUMNS

22 REUVEN LERNER'S AT THE FORGE

Creating Mashups

30 MARCEL GAGNÈ'S COOKING WITH LINUX

A Gem of an Idea



34 DAVE TAYLOR'S WORK THE SHELL

Recognizing Blackjacks

36 JON "MADDOD" HALL'S BEACHHEAD

Sinking of the *USS Proprietary*

38 DOC SEARLS' LINUX FOR SUITS

Causes and Effects

96 NICHOLAS PETRELEY'S /VAR/OPINION

It's free. It's proprietary. No, it's two (click) two (click) two distros in one.

IN EVERY ISSUE

10 LETTERS

14 UPFRONT

42 NEW PRODUCTS

81 ADVERTISERS INDEX

INDEPTH

66 RAILS WRIT LARGE

If you want freedom, add constraints.

Scott Raymond



74 OPENSSL HACKS

The unsung OpenSSL command line.

Anthony J. Stieber

78 THE SEARCHABLE SITE

Take a glimpse at Webglimpse for search capabilities for your site.

Golda Velez



82 UNDER-ICE SONAR VISUALIZATION

What's going on down under you.

Richard R. Shell, Garner C. Bishop and Douglas B. Maxwell

92 SECURING OPENSSH

It's important to ask, who's watching the guard?

Matthew E. Hoskins



82 UNDER-ICE SONAR VISUALIZATION

Next Month

ULTIMATE LINUX BOX

This year's Ultimate Linux Box issue has something for almost everyone. Need the ultimate high-availability, robust server? How about the ultimate foundation for a do-it-yourself Linux box? While you're at it, customize that box as the ultimate Linux-based multimedia center. Best of all, we chose winners you can actually buy, not pie-in-the-sky fantasy boxes.

There's more. We'll tell you how to use hole-punching to create peer-to-peer protocols that aren't crippled by network address translation. We tackle the embedded Linux challenge for mobile phones. How can you use Linux to create just about any network-attached peripheral, even a DVD burner? You'll find out next month.

TYAN



Power of 4 + 4 TYAN Matchless 8-Processor Server

**M4881**

- Processor Expansion board for S4881
• Up to four AMD Opteron™ processor MP
• Up to 64GB of Registered (ECC) DDR400/333
166 DIMM sockets
• 12"x13" dimension with special architecture
• Two Hyper Transport™ v1.0 Connectors

**Thunder S4881**

- 8-Processor AMD Opteron™ Motherboard
• Up to eight or four AMD Opteron™ processor
• Up to 64GB of Registered (ECC) DDR400/333
• (4)SATA support RAID 0,1, 0+1
• Broadcom® BCM5704C Dual-Channel GbE controller
• IEEE 1394a (FireWire) ports
• ATI® RAGE™ XL with 8MB

**Transport VX50 B4881**

8-Processor AMD Opteron™ 5U Server

- 4/8-Processor HPC Computing Platform
- Support up to eight (8) AMD Opteron™ 8xx series
- 4 P model (B4881-4P);
Sixteen (16) DIMMs, supporting max.
64GB registered, ECC DDR400/333 memory
- 8-P model (B4881-8P);
Thirty-two (32) DIMMs, supporting max.
128GB registered, ECC DDR400/333 memory
- Two (2) gigabit Ethernet LANs,
Broadcom® BCM5704 controller
- Three (3) 64-bit, 133/100/66MHz PCI-X slots ,
Two (2) PCI-F slots (1"x16 + 1"x4 signal)
Total 5 usable PCI expansion slots

TYAN COMPUTER CORP.

Tyan Computer USA

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

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

www.tyan.com

Call Me!

So we can get close and personal 24/7...



JTL Network's staff works personally with your team to offer 24/7 support for your web hosting needs. We have a 98.7% customer satisfaction rating.

Linux Hosting Solutions

Starting From \$9.95/mo with No Set-up Fees.
Dedicated Servers from \$69/mo.
Colocation starting at \$35/mo.

- Redhat, Suse, Fedora or Centos Servers
- MySQL, PostgreSQL Database Support
- Online Control Panel for site and server management
- Remote Reboot
- Virus and Spam protection
- 24/7 Toll-Free Support



Real People Real Support™

since 1998

www.jtl.net/lj

1-877-765-2300

LINUX JOURNAL

Editor in Chief

Nick Petreley, ljeditor@ssc.com

Executive Editor

Jill Franklin

jill@ssc.com

Senior Editor

Doc Searls

doc@ssc.com

Art Director

Garrick Antikajian

garrick@ssc.com

Products Editor

James Gray

newproducts@ssc.com

Editor Emeritus

Don Marti

dmarti@ssc.com

Technical Editor

Michael Baxter

mab@cruzio.com

Senior Columnist

Reuven Lerner

reuven@lerner.co.il

Chef Français

Marcel Gagné

mgagnie@salmar.com

Security Editor

Mick Bauer

mick@visi.com

Contributing Editors

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

Proofreader

Geri Gale

VP of Sales and Marketing

Carlie Fairchild

carlie@ssc.com

Marketing Manager

Rebecca Cassity

rebecca@ssc.com

International Market Analyst

James Gray

jgray@ssc.com

Sales Coordinator

Lana Newlander

ads@ssc.com

Regional Advertising Sales

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

EASTERN USA: Martin Seto, +1 416-907-6562

SOUTHERN USA: Laura Whiteman, +1 206-782-7733 x119

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

Advertising Inquiries ads@ssc.com

Publisher

Phil Hughes

phil@ssc.com

Accountant

Candy Beauchamp

acct@ssc.com

Linux Journal is published by, and is a registered trade name of, SSC Media Corp.

PO Box 980985, Houston, TX 77253-3587 USA

Editorial Advisory Board

Daniel Frye, Director, IBM Linux Technology Center

Jon "maddog" Hall, President, Linux International

Lawrence Lessig, Professor of Law, Stanford University

Ransom Love, Director of Strategic Relationships, Family and Church History Department,

Church of Jesus Christ of Latter-day Saints

Sam Ockman, CEO, Penguin Computing

Bruce Perens

Dale Garbee, Linux CTO, HP

Danese Cooper, Open Source Diva, Intel Corporation

Subscriptions

E-MAIL: subs@ssc.com

URL: www.linuxjournal.com

PHONE: +1 713-589-3503

FAX: +1 713-589-2677

TOLL-FREE: 1-888-66-LINUX

MAIL: PO Box 980985, Houston, TX 77253-3587 USA

Please allow 4-6 weeks for processing address changes and orders

PRINTED IN USA

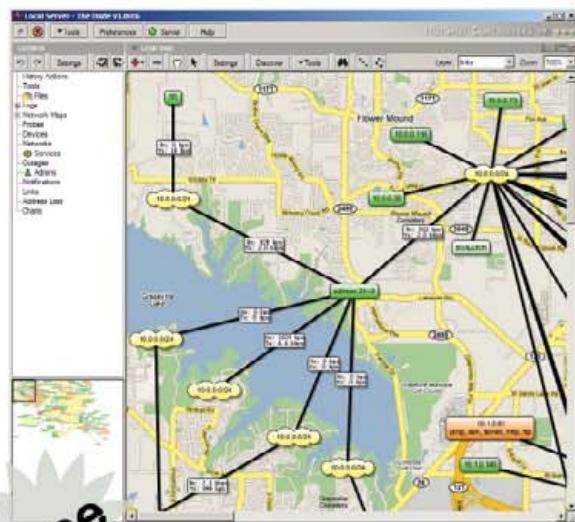
LINUX is a registered trademark of Linus Torvalds.

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 (P2P) protocol control
 - per client P2P tx/rx rules with burst support
 - 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
 - monitors 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:



Free

Download at www.mikrotik.com/dude

"The Dude" Network Monitor

- Auto network discovery and layout
- Discovers any type or brand of device
- Device, Link monitoring, and notifications
- Includes SVG icons for devices, and supports custom icons and backgrounds
- Easy installation and usage
- Allows you to draw your own maps and add custom devices
- Supports SNMP, ICMP, DNS and TCP monitoring for devices that support it
- Link usage monitoring and graphs
- Direct access to remote control tools for device management
- Supports remote Dude server and local client
- Runs in Linux Wine environment, MacOS Darwin, and Windows
- Best price/value ratio compared to other products (free of charge)

RouterBOARD 112

- Super-small embedded board
- 140mm x 85mm (5.51in x 3.35in)
- Ideal for wireless CPE solutions
- RouterOS Wireless AP/Firewall/Routing software already included
- Best price/performance/feature ratio compared to similar products
- 175MHz MIPS CPU
- 2 miniPCI (one on each side)
- 1 10/100 Ethernet MDI-X
- 64MB NAND storage
- 16MB SDRAM
- Speaker
- Serial port
- Passive PoE (also 12V PoE)
- Power input 11-48V



\$99

\$95



RouterBOARD 44

For the Router Builder

- up to 24 Ethernet ports in a PC
- no more straight/cross cable problems
- server quality VIA VT6105 chips

\$240



RouterBOARD 500 & RouterBOARD 564

The Wireless Switchboard AP

The bigger and more powerful RB532 accompanied with a RB564 daughterboard can be a complete multi-radio tower system, with 9 ethernets and 6 miniPCI slots.

CIARA

GET THE SERVER THAT TAKES YOUR BUSINESS AS SERIOUSLY AS YOU DO.

Intel® Xeon® Processors powering
Ciara VXPRO™ servers provide the
quality and dependability to keep up
with your growing business.

3 Good Reasons to Buy from Ciara Technology

Built To Order

Customized Servers, Blade & Storage:
Get the technology that's right for your business and not right for your supplier. With the capability to manufacture over 2500 systems a day, Ciara is suited to accommodate any customer requirement. Our record growth enabled us in February 2003 to inaugurate an all-new, ultra-modern manufacturing plant of 576,000 sq ft. Our systems are built under the ISO 9001-2000 certification.

Service...Service...Service

Incorporated in 1984, Ciara Technology is a world-class provider of computer systems including desktop, laptop, servers, storage and supercomputer clusters as well as other software and integration services. All our systems are serviced by Ciara's highly trained and certified technicians and system engineers. We are an accountable supplier - One single point of contact for all your technology needs.

Channel Partner Premier

Ciara has a strong working relationship with Intel, so we have access to information and support that give us – and you – significant advantages in deploying and managing your systems and applications. The result is a more flexible solution that meets your current needs, while enabling easy expansion to accommodate emerging technologies and new business growth.

Visit us

For all your computer needs

Give us a call

CIARA-TECH.COM

1-866-789-7225

TECH

DUAL CORE RACKMOUNT SERVER



Power Efficient
Intel® Xeon® LV
120Watts
per server

\$1099¹

DUAL CORE

VXPRO-R7230NH Affordable Starter Server

Ultra Affordable
Easy to Service

**Intel® Pentium® D Processor 920
(2.8GHz Dual Core Processor)**
800MHz Front Side Bus
2x2MB L2 Cache

1GB (2x512MB) ECC DDR2 533 Memory
(Expandable to 8GB)

One 80GB (7,200RPM) SATA150 HDD
(Up to 4 Hot Swap Hard Drives)
Floppy and CD-ROM included

1U Rackmount
3 Years Warranty Return to Depot

Upgrade Your System

Intel® Pentium® D Processor 930 (3.0GHz)
Intel® Pentium® D Processor 940 (3.2GHz)
Additional 80GB (7,200RPM) SATA150
Additional 1GB (2 x 512MB) ECC DDR2 533

Add \$99
Add \$140
Add \$72
Add \$109

\$2955¹

DUAL CORE

VXPRO-R7520BB2 Low Voltage - 4 Cores Server

Extremely Powerful
Ultra Low Power Consumption

**Dual Intel® Xeon® LV Processor
(2.0GHz Dual Core Processor)**
667MHz Front Side Bus
2MB of Shared L2 Cache

2GB (2x1GB) ECC/Reg DDR2 400 Memory
(Expandable to 16GB)

One 80GB (7,200RPM) SATA150 HDD
(Up to 4 Hot Swap Hard Drives)
Floppy and CD-ROM included

1U Rackmount
3 Years Warranty Return to Depot

Upgrade Your System

Additional 80GB (7,200RPM) SATA150
Additional 2GB (2 x 1GB) ECC/Reg DDR2 400

Add \$72
Add \$299

Celeron, Celeron Inside, Centrino, Centrino Logo, Core Inside, Intel, Intel Logo, Intel Core, Intel Inside, Intel Inside Logo, Intel SpeedStep, Intel Viv, Itanium, Itanium Inside, Pentium, Pentium Inside, Xeon and Xeon Inside are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. (1) Important Information. All prices, specifications and promotional offers are subject to change without notice. Clara cannot be responsible for typographical errors, photographies errors, pricing errors. All pricing in US dollar. Shipping and applicable taxes are not included.



letters



Change in Direction?

I am writing to express my concern with the sudden change in direction that *Linux Journal* appears to have taken. *LJ* has always stood apart from its competitors by covering diverse technical material that would be of no interest to the beginner or casual Linux user, but is the bread and butter of the hacker.

When the first competing magazines appeared on the market, it was apparent that their audience was less technically savvy people, perhaps interested in dipping their toes into the world of Linux.

The last three issues of *LJ* have shown a marked reduction in such diverse, technical articles. I was particularly surprised to find six pages dedicated to installing MediaWiki—a job that would take the average hacker no more than five minutes to do.

Conversely, the newsstand Linux magazines have been slowly moving toward accommodating the hacker, with more in-depth coverage of happenings in the kernel and distro worlds.

I urge you to take a closer look at what has made *LJ* such a unique publication over the last 12 years and strive to maintain the level of technicality and diverseness that is key to that uniqueness.

--

Jon Dowland

Appalled by Rant

I've been an *LJ* subscriber for six years and figured I had to reply to your recent /etc/rant [May 2006]. I am seriously considering terminating my subscription after reading this one, and I have only thought this since you have joined on as editor in chief. Sometimes I think these sections are written just to kill off some of the loyal readership you do have.

I just wanted to say that I am quite appalled by your latest /etc/rant. Usually, these sections are slightly tolerable, even though they have no place in a technical journal, but this one certainly takes the cake. I feel you expose your own ignorance when it comes to the AMD64 platform. Before someone comes out and declares all of Linux to be unsuitable for 64-bit use, perhaps he should try other distributions?

There are many distributions out there that will support AMD64 quite nicely, namely SUSE, Red Hat and Gentoo. Perhaps you should try one of these next time before writing such a damning article to an audience that just might be working to improve the situation.

As for your plugin issue, it really is quite simple. No 64-bit browser will run 32-bit plugins, which means no Flash. If you want Flash, run a 32-bit browser. (As for Java, there is a 64-bit plugin that will run in 64-bit compiled browsers.) Due to ABI differences, this is the case. (Somewhat akin to, say, running x86 plugins on a PPC system.)

If you have any questions, feel free to contact me. Also, if you would like some constructive articles for the journal, I'd also be willing to help turn things around over there (the way it used to be). If these sorts of articles are not fixed/changed, I will have no choice but to terminate my subscription.

I've got a dual-core Athlon X2 here with two gigs of RAM, GeForce FX 7800GT, and one terabyte of storage purring along nicely.

--

Gerald

Where did you get your 64-bit Java plugin for Firefox? It doesn't come with 64-bit Java, as does the 32-bit plugin. I do not expect 32-bit plugins to work with a 64-bit browser. I expect 64-bit plugins to be widely available and to work. As long as this isn't the case, distributions should automatically set up a 32-bit browser that is able to use 32-bit plugins (better still, install the plugins too). That's the way SUSE does it. Unfortunately, Ubuntu/Kubuntu didn't do that. Considering how long the AMD64 has been around, I'm shocked that there's popular software (such as Flash) that still isn't available for AMD64.—Ed.

Rave about "The 64-bit Question"

Excellent article! You told it like it was and did it with class and style! Don't like the new look though. Keep up the good work.

--

Gary Burt

No Kids?

My son and I were at Mr Negroponte's \$100 Laptop Keynote yesterday morning [LinuxWorld Boston 2006] and very much enjoyed it. We have been following the progress of the project since it first hit the wire!

The unfortunate thing we learned when we arrived was that LinuxWorld had raised the age restriction to 18 after we had registered in early January, therefore my son was not allowed in the Expo. The VP of LinuxWorld told me that the larger vendors demanded that children *not* be allowed on the floor because they interfere with selling their products. She did allow us to see the keynote, which was the highlight of an otherwise devastating day for my son. LinuxWorld states on its site that the 18-year-old age restriction is for safety and insurance reasons. That is *not* what I was told. Red Hat was the one company mentioned about pressuring LinuxWorld to up the age.

Mr Negroponte's excellent talk was all about the *children* of this World, Linux and Education, and unfortunately there was no EXPO education for my son today. I wanted you to know that the only person under 18 in his audience yesterday is a fan of his, and he wanted vendors to know that he wasn't able to enjoy the rest of the Expo. Thank you for listening.

--

Brad Fligor



Remember when the sky was the limit?

With Intel® Software Development Products, the Swinburne Center for Astrophysics is showing today's kids a universe filled with unlimited possibilities.



LINUX* APPLICATIONS HELP UNRAVEL THE ORIGINS OF THE UNIVERSE.

Swinburne University's Center for Astrophysics and Supercomputing in Melbourne, Australia, is helping develop the next generation radio telescope in order to collect enough data to perform modeling and simulations of our entire galaxy. Their goal is to make realistic, 3D, virtual-reality animations available to the general public, particularly school children. To do this, they used the Intel® compilers to deliver application performance improvements — saving them valuable development time and money¹. Whether you build applications for physics or financial analysis, Intel® Software Development Products help your Linux* applications reach for the stars.

Download a trial version today.
www.intel.com/software/products/nolimits

Copyright © Intel Corporation, 2006. All rights reserved. Intel, the Intel logo, are trademarks of Intel Corporation or its subsidiaries in the United States and other countries. *Other names and brands may be claimed as the property of others. ¹Performance and benchmark information were provided by Swinburne University. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing.

[LETTERS]

More Responses to "The 64-Bit Question"

In response to Nick Petreley's etc/rant "The 64-Bit Question" in the May 2006 issue, how is it that Linux has an opportunity to be "the first, best AMD64 desktop platform"? Solaris 10 has been running flawlessly on my W1100Z for months. Is your magazine trying to inform its readers and make them more productive or simply sell them something?

--
Ralph Wojtowicz

Get Off Your Arse

I could rant that my toilet tank is badly designed because *LJ*'s new format slides off of it and interferes with raising the seat. But the blame for that problem would be about as well placed as Nick Petreley's arrogant May 2006 /etc/rant that distro developers should get "off their arses" and give him a better AMD64 desktop Linux. Mr Petreley screwed up. He chose a hardware and software platform without researching whether the combination was well supported in his intended application. That's not the problem of the generous noncommercial distro developers who put their limited resources where they'll help the most colleagues, nor of the commercial distro developers, who have to make hard business decisions based upon demand. The way that Linux works is that if Petreley wants a distro that nobody else is making, it's up to him to get off of his own arse with like-minded contributors and make it happen.

--
Greg Bullough

Stay Away from Java?

After reading your etc/rant column, "The 64-Bit Question", you may have finally realized (hopefully) why some open-source advocates suggest staying away from Java.

You see, neither Java nor Flash has good support for running in 64-bit under AMD64 chips.

Neither Sun nor Macromedia is willing to release their source code freely enough for those who want to do the work for them. In another words, we can run Java because Sun allows us to run it. Obviously, we don't have the mercy (yet) from Sun to run Java on AMD64 chips.

If we are so dependent on Java and Sun decided tomorrow that nobody can run Java on any architecture other than Sparc, all of us have to have Sparc because, as I said, we've become "dependent" on it.

I am not saying we should throw out Java totally, but shouldn't we be cautious as to which programming tool we choose for our open-source projects?

--
Jack Cheung

Java runs fine on an AMD64. The downloadable version from Sun (at least at the time I tested it) simply didn't include a 64-bit plugin for 64-bit Firefox.—Ed.

Some Questions

I love *LJ* and find the tips very useful. Regarding GNOME bashing, try XFCE instead! My questions, however, pertain to the QEMU and User Mode articles in the May 2006 issue.

Is it possible to run the user-mode techniques on my Powerbook (PPC) running Ubuntu, and would I be able to run x86 or other architectures? How?

Also, I installed QEMU using Ubuntu's Synaptic, but every image that I tried to emulate with the commands given in the article cause the wm to crash and log me out. Any suggestions?

Keep up the good work and the good looks!

--
Brian O'Keefe

Missed Opportunity

I fully agree with your article [Nick Petreley's "The 64-Bit Question", May 2006]. Linux misses a big opportunity. So does OpenOffice.org. The 64-bit binary for AMDs is still not available, and glancing through Web sites, OOo 2.0.2 doesn't compile on x86_64. Therefore, one needs to install the 32-bit version, which in turn doesn't like the 64-bit Java, and the first time you start up OOo, it takes about ten minutes until its search for Java times out.

--
Thomas Spuhler

Have Hope

My right-hand man here in the USA reads your magazine regularly. He brought a copy along when he came to visit last week. You and those AMD chips—heh, heh, heh. The IT industry makes a business in circulating *slop*, does it not?

I just wanted to send along a note of thanks—for putting some gas on our fire through the years—for being human enough to respond when I have e-mailed. That is usually *not* the case with writers in the IT media.

I have been working away on a vision to *radically* fix the world's IT messes since 1995. To date, three systems designed and implemented from the ground up (aka Opus 1 through 3), plans for two more sitting on paper (aka Opus 3.5 and 4). Just because the seemingly *entire world* complains about things as they are, ya think they would get behind an effort to fix it all! Well, it takes other types of preparations to do a project like this—far more than simply writing the code, creating the IP, distributing and supporting it. So in that way we are making progress. Nick, the fix is coming, I tell you.

Nick, you seem to be one of a rare breed—those who are able to use their brains. So, thanks, keep doing so. Probably it is mostly for being able to help, to rescue, folks like you that keep me going—the folks that "get it" about the way things "should be". Just keep encouraging through your work as you can, Nick, we are out here, trying to make progress, trying to bring the solution. Steps forward are not always easy, not always fast, not always fun.

--
Michael Lueck

LINUX JOURNAL

At Your Service

MAGAZINE

PRINT SUBSCRIPTIONS: Renewing your subscription, changing your address, paying your invoice, viewing your account details or other subscription inquiries can instantly be done on-line, www.linuxjournal.com/subs. Alternatively, within the U.S. and Canada, you may call us toll-free 1-888-66-LINUX (54689), or internationally +1-713-589-2677. E-mail us at subs@linuxjournal.com or reach us via postal mail, Linux Journal, PO Box 980985, Houston, TX 77253-3587 USA. Please remember to include your complete name and address when contacting us.

LETTERS TO THE EDITOR: We welcome your letters and encourage you to submit them to ljeditor@ssc.com or mail them to SSC Editorial, 1752 NW Market Street, #200, Seattle, WA 98107 USA. Letters may be edited for space and clarity.

WRITING FOR US: We always are looking for contributed articles, tutorials and real-world stories for the magazine. An author's guide, a list of topics and due dates can be found on-line, www.linuxjournal.com/author.

ADVERTISING: *Linux Journal* is a great resource for readers and advertisers alike. Request a media kit, view our current editorial calendar and advertising due dates, or learn more about other advertising and marketing opportunities by visiting us on-line, www.linuxjournal.com/advertising. Contact us directly for further information, ads@linuxjournal.com or +1 206-782-7733 ext. 2.

ON-LINE

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

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

Data Center Solutions Built on OPEN STANDARDS

OSS-1000 1U OPEN Series

- Available with 1, 2, 4, or 8 CPU's per system
- Up to 2 terabytes storage and 64GB of RAM
- Tool-less 1U case design with rail kits
- 4 Hot-swap drive bays, SCSI/SATA-2
- Supports RAID levels 0,1, 5, 6, and 10
- 3 year next day warranty within the USA



OSS-3000 3U OPEN Series

- Available with 1, 2, 4, or 8 CPU's per system
- Up to 8 terabytes storage and 64GB of RAM
- Tool-less 3U case design with rail kits
- 16 Hot-swap drive bays, SCSI/SATA-2
- Supports RAID levels 0, 1, 5, 6, and 10
- 3 year next day warranty within the USA



866-664-STOR
Sales@OSShpc.com
OpenSourceStorage.com



diff -u

WHAT'S NEW IN KERNEL DEVELOPMENT

Paul Mundt has submitted patches to remove **RelayFS** from the kernel and migrate all of its functionality into a generic API that can be used by any filesystem. Thus, what started off as a highly specialized tool—a filesystem only for high-speed data transfer between user space and the kernel—has now been generalized into a service with wide-ranging applicability. Because RelayFS has been in the official kernel for some time, taking it out now already has produced some controversy. But, apparently, **Andrew Morton** promised the RelayFS developers that they could continue to make these large, sweeping developments, so long as no in-kernel code relied on RelayFS. User applications should not have been relying on RelayFS during this time of instability.

Intel has announced a new open-source project to support its **PRO/Wireless 3945ABG Network Connection mini-PCI express adapter** (IPW3945). It is not fully open source. There is a single binary-only part, containing the regulatory enforcement logic for all the countries where those adapters ship. In his announcement, **James Ketrenos** said that Intel had improved its licensing relative to earlier projects. The binary portion of this project uses the same license as the binary-only firmware, and James has said it is easier to understand and more permissive of redistribution than earlier licenses. The whole issue of regulatory enforcement and binary distribution is, as one might expect, controversial. The binary daemon must run as root, which makes any potential bugs a large security problem. Also, there is some question as to whether the FCC regulations actually require binary-only distribution, or only, as **Alan Cox** puts it, "that the transmitting device must be reasonably tamper-proof". Regardless of any controversy (that is sure to be ongoing), Intel can at least be credited with freeing the portions of its code it has freed and with making some effort to improve its license.

Bernhard Rosenkraenzer has forked the **cdrtools project** into his own **dvdrecord project**, and released version 0.3.1 with various enhancements, such as supporting 2.6 kernels, supporting writing to DVD-R and DVD-RW disks using purely free software and cleaning up the make system. Apparently, there have been massive flame wars precipitating the fork, and a number of kernel folks have said that Bernhard's work represents the only progress on the driver in recent

days. The conversation has turned toward whether his dvdrecord will support additional features, such as DVD RAM, DVD+R, DVD+RW and DVD+DL. It's unclear what the long-term direction of the project will be, because forks are always controversial, but at least there seems to be no immediate outcry that Bernhard forked the project unreasonably. It's still too soon for predictions, however.

Miklos Szeredi has created the **mountlo utility**, a tool that supports loopback mounting in user space. Until now, a filesystem image stored in a single file on disk could be mounted only via the kernel's own loopback support. Miklos' utility relies on FUSE (Filesystem in USErspace) to move the entire feature into userland and out of the kernel. Miklos considers it more of a pet project than anything else—an opportunity to play with FUSE and see what useful tools he could create.

The ever-vigilant **Christoph Hellwig** has pointed out that the **gdth driver** seems to be the only user of the **scsi_request interface** expected to be gone in Linux 2.6.17. The gdth maintainers have not responded to patch submissions, and unless someone steps forward to make sure gdth survives the interface change, it will be marked as "BROKEN" in 2.6.17. **Achim Leubner** has stepped forward to do some testing, but this leaves the question of driver maintainership up in the air. Typically, unmaintained code is rapidly deprecated in Linux.

Greg Kroah-Hartman has begun to document **ABI** (Application Binary Interface) levels of stability within the kernel. When a binary interface changes, user-space binaries that link to that interface break. And, because a given interface may have countless applications relying on it, it is considered virtually unacceptable to change the kernel ABI for that reason. Unfortunately, ABI changes are a fact of life. They do change, and have been changing, and the question as Greg sees it, is how to balance this with the needs of userland. Greg's idea is to provide enough indication about the future of a given interface that application developers will have time to rewrite their code before the change takes place. As one might expect, this is an incredibly controversial issue. Many top kernel people feel that the ABI is sacrosanct and should never be altered. But **Linus Torvalds**, in spite of having criticisms of the specific details of Greg's effort, seems generally to agree that ABI changes are inevitable, and that the kernel should do what it can to ease the burden placed on application developers.

—Zack Brown

Signs of GhandiCon 4

"First they ignore you, then they ridicule you, then they fight you, then you win." —Mahatma Gandhi

For many years, Linux and open source were synonymous with "threat" at Microsoft. Although that status may not have changed, the company has been moving gradually in a more accepting direction toward market opponents that include, in some cases, its own customers.

We've seen acceptance, for example, in the Identity Metasystem, which was a topic of my *Linux Journal* column in October 2005, and also with the company's embrace of blogging—a product of open standards and practices that grew up outside Microsoft's (or any vendor's) platform.

But with Microsoft's Open Source Software Lab, which started up in August 2005, acceptance was elevated above the purely de facto level. It became policy. Then, in April 2006, Microsoft opened the lab itself, through a new Web site called Port 25. Named for the router port for outbound e-mail, the site is open, interactive and intended to foster helpful communications with Microsoft customers that use open-source software as well.

Although this hardly means Microsoft will be developing open-source software any time soon (especially for Linux), it may help ease tensions in the heterogeneous environments where customers require productivity, and not merely coexistence.

I asked David Sifry, founder and CEO of Technorati (which runs on Linux), for some perspective on what it means. He said, "It's good to see Microsoft recognizing the impact and importance of open-source software, and I'm encouraged by its recent moves to understand and work with the large community of nonproprietary software developers out in the world. In the end, it will be good for Microsoft customers."

www.forbes.com/home/enterprisetech/2006/03/22/ballmer-microsoft-linux-cz_df_0322microsoft.html

—Doc Searls

RubyGarden

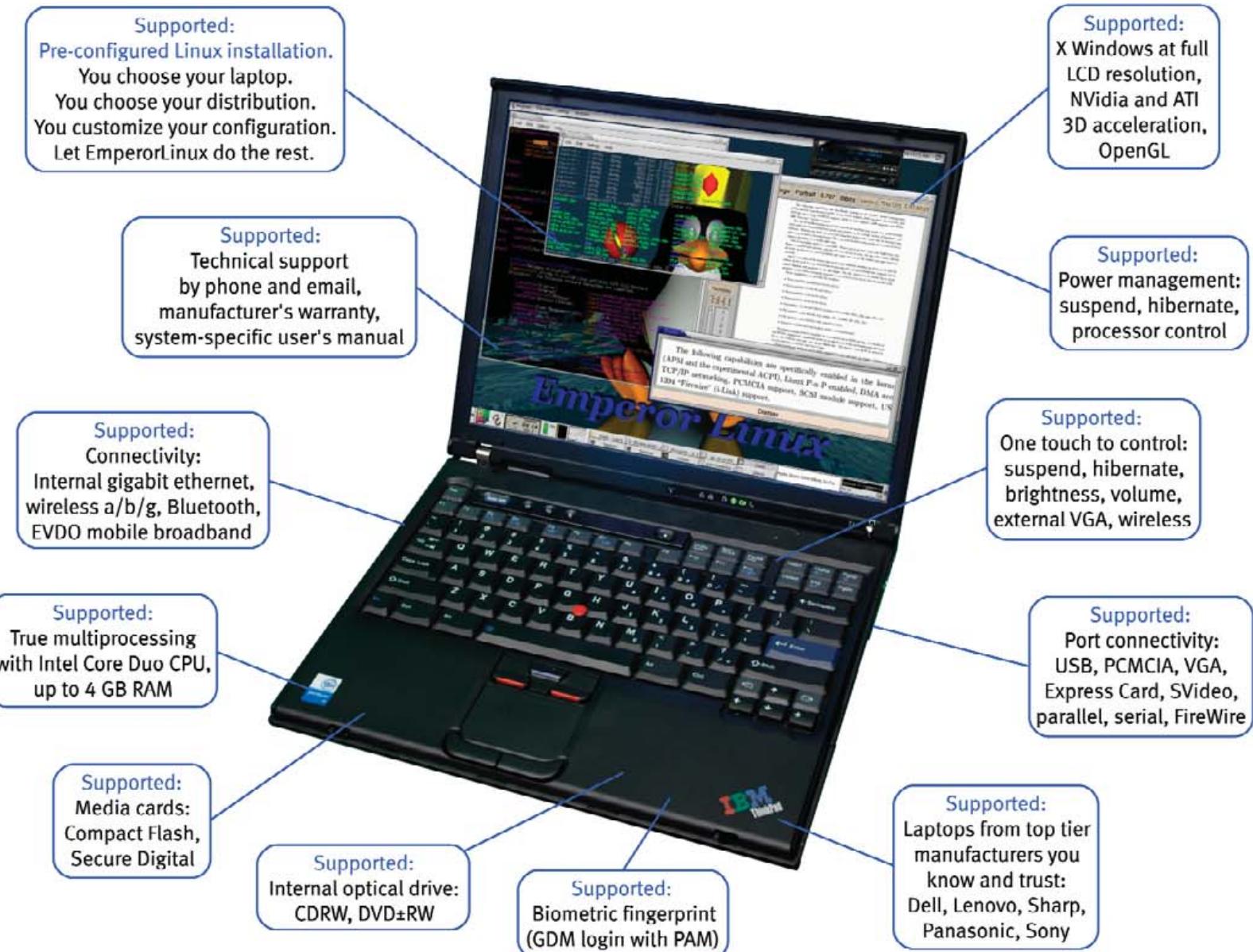
Visit www.rubygarden.org for all your Ruby needs, including links to other all-important Ruby sites, such as Pragmatic Programmers and RubyForge, as well as links to Ruby Webloggers.

RubyGarden provides an exhaustive listing of Ruby User Groups around the world at www.rubygarden.org/ruby?RubyUserGroups.

RubyGarden also recommends other sites of interest for Ruby User Groups, including ruby.meetup.com, where you can find a meetup group near you, and www.rubyholic.com, where Ruby groups can register and post meeting information.

You also can join the Google Ruby Brigade Group, at groups.google.com/group/Ruby-Brigades or by sending an e-mail to Ruby-Brigades-subscribe@googlegroups.com.

Linux laptops. Supported.



Since 1999, **EmperorLinux** has provided pre-installed Linux laptop solutions to universities, corporations, and individual Linux enthusiasts. We specialize in the installation of the Linux operating system on a wide range of the finest laptops and notebooks made by IBM, Lenovo, Dell, Sharp, Sony, and Panasonic. We offer a range of the latest Linux distributions, as well as Windows dual boot options. We customize each Linux distribution to the particular machine it will run upon and provide support for: ethernet, wireless, EVDO mobile broadband, PCMCIA, USB, FireWire, X-server, CD/DVD/CDRW, sound, power management, and more. All our systems come with one year of Linux technical support by both phone and email, and full manufacturers' warranties apply.

Visit www.EmperorLinux.com or call 1-888-651-6686 for details.

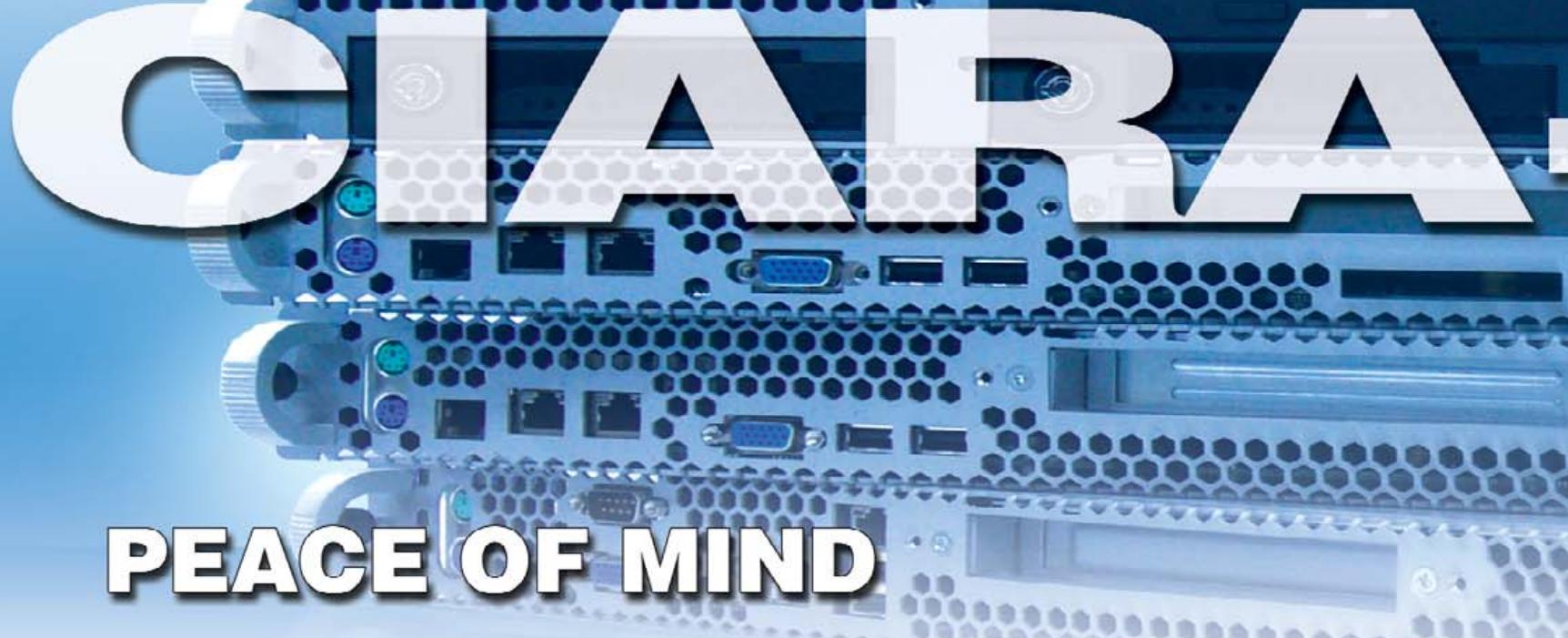
EmperorLinux

...where Linux & laptops converge

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

Model specifications and availability may vary.





PEACE OF MIND

With the **Dual-Core Intel® Xeon® Processor** in your **VXRACK™**, your servers have advanced reliability features built in.

3 Good Reasons to Buy a VXRACK™ Cluster Technology

Built To Order

Customizes Servers, Blade & Storage:
Get the technology that's right for your business and not right for your supplier. With a capability to manufacture over 2500 systems a day, Ciara is suited to accommodate any customer requirement. Our record growth enabled us in February 2003 to inaugurate an all-new, ultra-modern manufacturing plant of 576,000 sqf. Our systems are build under the ISO 9001-2000 certification.

Service...Service...Service

Incorporated in 1984, Ciara Technology is a world class provider of computer systems including desktop, laptop, servers, storage and supercomputer clusters as well as other software and integration services. All our systems are serviced by Ciara's highly trained and certified technicians and system engineers. We are an accountable supplier - One single point of contact for all your technology needs.

Channel Partner Premier

Ciara has a strong working relationship with Intel, so we have access to information and support that give us – and you – significant advantages in deploying and managing your systems and applications. The result is a more flexible solution that meets your current needs, while enabling easy expansion to accommodate emerging technologies and new business growth.

Visit us

For all your computer needs

Give us a call

CIARA-TECH.COM

1-866-789-7225

TECH

HIGH DENSITY BLADE SERVER



\$3835¹

DUAL CORE

\$2755¹

DUAL CORE

Power Efficient
Intel® Xeon® LV
120Watts
per server

VXBLADE-7520JR2 A Standard in the Datacenter

Powerful and Reliable
Mid to High End Server

Dual Intel® Xeon® Processor
(2.8GHz Dual Core Processor)
800MHz Front Side Bus
2x2MB L2 Cache

2GB (2x1GB) ECC/Reg DDR2 533
(Expandable to 12GB)

One 80GB (7,200RPM) SATA150 HDD

3 Years Warranty Return to Ciara

Upgrade Your System

Upgrade to 120GB (7,200RPM) SATA150
Additional 2GB (2 x 1GB) ECC/Reg DDR2 400
Upgr. to 8GB (4 x 2GB) ECC/Reg DDR2 400

Add \$35
Add \$299
Add \$899

Upgrade Your System

Upgrade to 120GB (7,200RPM) SATA150
Additional 2GB (2 x 1GB) ECC/Reg DDR2 400
Upgr. to 8GB (4 x 2GB) ECC/Reg DDR2 400

Add \$35
Add \$299
Add \$899



VXR-72 or VXR-96 Special Rack Enclosure

VXR-96 accommodates the following:
Up to 8 enclosure chassis
Up to 96 VXBLADE
1.1TB of memory
48 TB of local storage

VXR-72 accommodates the following:
Up to 6 enclosure chassis
Up to 72 VXBLADE
0.81TB of memory
36 TB of local storage

Optional Vertical Cable Management
Optional Infiniband Cable Management
Optional Color customization



LJ Index, July 2006

1. Minimum bandwidth in Mbps required for 1080p HDTV: **30**
2. Days required to download a standard 7.8GB DVD movie file over 56Kbps dial-up: **13**
3. Record number of terabits per second sent over a 160km link: **2.56**
4. Equivalent number of DVDs per second: **60**
5. Percentage of San Francisco Mayor Gavin Newsome wants covered by free Wi-Fi: **100**
6. Percentage of on-line consumers who express an interest in podcasts: **25**
7. Percentage of on-line households in North America that regularly download and listen to podcasts: **1**
8. Versions of UNIX running at Microsoft's Open Source Software Lab: **15**
9. Versions of Linux running at Microsoft's Open Source Software Lab: **50**
10. Billions of dollars in PC-based point-of-sale (POS) systems sold in North America in 2005: **6**
11. Linux percentage share of North American POS system sales in 2005: **9**
12. Linux percentage share of North American POS system sales in 2004: **5**
13. Millions of dollars in Linux sales in China in 2005: **11.8**
14. Percentage growth of Linux in China from 2004–2005: **27.1**
15. Expected compound annual growth percentage rate of Linux sales in China over the next four years: **34**
16. Projected millions of Linux sales, in dollars, at the end of the next four years in China: **51.1**
17. Percentage probability that open source will compete with closed-source products in all infrastructure markets by 2008: **80**
18. Percentage probability that 75% of all mainstream IT organizations will have formal open-source acquisition and management strategies: **80**
19. Percentage probability that mainstream IT organizations will consider open-source software in 80% of their infrastructure-based software investments: **70**
20. Percentage probability that open source will be included in mission-critical software portfolios within 75% of Global 2000 enterprises: **90**

Sources: 1: Wayne Caswell | 2–4: *Gizmag* | 5: New Media Musings | 6, 7: Forrester Research | 8, 9: Microsoft Open Source Software Lab | 10–12: *LinuxDevices.com* | 11–16: *Silicon.com*, sourcing International Data Corp. (IDC) | 17–20: Gartner

—Doc Searls

They Said It

Don't worry about people stealing an idea. If it's original, you will have to ram it down their throats.
—Howard Aiken, source: Marc Hedlund, radar.oreilly.com/archives/2006/03/entrepreneurial_proverbs.html

One of my many mottos of the past is "Only steal from the best." When you use someone else's idea that's the ultimate sign of respect. But it's important to say who you're stealing from, because they're the best, right?
—Dave Winer, www.scripting.com/2006/03/31.html#When:7:12:08AM

Think about it this way: if the water that's piped into your house had DRM on it and only allowed you to use it for showers, how would you wash your clothes? If you were only allowed to make ice cubes, how would you make iced tea? If you had to pay \$0.99 every time you wanted a glass of water?

Ideas and hope need to flow like water if a civilization is to continue its ascension toward greatness. Impediments to that flow will stall growth. Fortunately, like a solvent, the culture of open source will continue to expand, will wear away at these impediments, to restore the natural flow of social capital, of ideas, of hope. Those who get this first will rise, and rise quickly.
—Chris Messina, factoryjoe.com/blog/2006/03/18/because-of-open-source

It's simple. The Internet has won. Why negotiate terms of surrender?
—Bob Frankston, www.frankston.com/?name=GettingConnected

BIGGER IS BETTER



Thinkmate High-Capacity Storage Servers featuring Intel® Xeon® processors add up to 30 terabytes of raw storage to your network for a fraction of the price of NAS or SAN solutions.



2U Storage Server
8 Hotswap Bays
Up to 6 TB
Optional OS Drive
Serial ATA or SCSI
Approx \$1.45/GB



3U Storage Server
12 Hotswap Bays
Up to 9 TB
Up to 2 OS Drive(s)
Serial ATA or SCSI
Approx \$1.30/GB



3U Storage Server
16 Hotswap Bays
Up to 12 TB
Optional OS Drive
Serial ATA or SCSI
Approx \$1.25/GB



5U Storage Server
24 Hotswap Bays
Up to 18 TB
Up to 2 OS Drive(s)
Serial ATA or SCSI
Approx \$1.20/GB



8U Storage Server
40 Hotswap Bays
Up to 30 TB
Up to 2 OS Drive(s)
Serial ATA Only
Approx \$1.25/GB



THINKMATE

[800] 371-1212
WWW.THINKMATE.COM

©2006 Intel Corporation. Intel, Intel Inside, and Intel Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. All rights reserved.

Tabblo: Pictures of Linux at Work

If you're looking to make something more of your digital photos than the usual on-line gallery, Tabblo has ways to do it. In addition to your basic GIMP (or, to the less free, Photoshop) tools (crop, zoom, add text and so on), Tabblo lets you lay out, annotate and share "tabblos" (a pun off "tableaus") of photo sets, or output them in various print forms through off-line services—all from inside a browser.

As Tabblo founder and CEO Antonio Rodriguez puts it, "If you have a story to tell with pictures, we have ways for you to tell it—on-line and off." He also says Tabblo's own story would not be possible without Linux. To get that story, we did a brief interview:

LJ: Why this business, and why now?

AR: This is something the market has needed for a long time, but the conditions weren't right. The Web browser wasn't built to do serious multimedia processing, and until recently, it wasn't possible to do it on the server—at least not with economics that worked for anyone trying to build a business out of it. If you wanted to do server-side image editing at high scale, for example, the cost of the software and hardware involved was simply prohibitive. If you wanted to have a rich database of things like metadata-based navigation and unlimited undo, the cost of the Oracle licenses at volume would probably put you flat out of the business of providing any meaningful functionality for free in order to attract customers. And most important, if you wanted to store gigs and gigs of people's pictures, NetApp or EMC were the only games in town—at economics that make sense only for investment banks and NASA. I can say without any doubt that there was absolutely *no way* that we could provide the level of application given to users at launch for free without the benefit of our modified LAMP stack and its associated development practices. I am particularly sensitive to this because of the number of photo sites in the past that have actually tried to make money and have been undone by spiraling capital expenses and wrong assumptions about the likely uptake of revenue-generating products.

LJ: What's in your stack?

AR: For hardware, we use custom-made whitebox AMD64 boxes with plenty of horsepower per dollar of cost, running Debian AMD64 (which greatly lowers our sysadmin costs). We run our own clustered filesystem built mostly on top of the Apache 2.0 stack (as modules) and get throughput that commercial storage solutions can't touch, mostly because the software is written specifically for our workload. The image servers also leverage the Apache runtime, as well as



Tabblo Creator Antonio Rodriguez and Python Creator Guido van Rossum

ImageMagick. Our database is MySQL5, which is great for its ability to be set up in very funky configurations that let you optimize writes, reads or replication depending on your own needs. And, finally, our Web app is written in Python, which is both lightning fast to develop in and incredibly clean as a dynamically typed language.

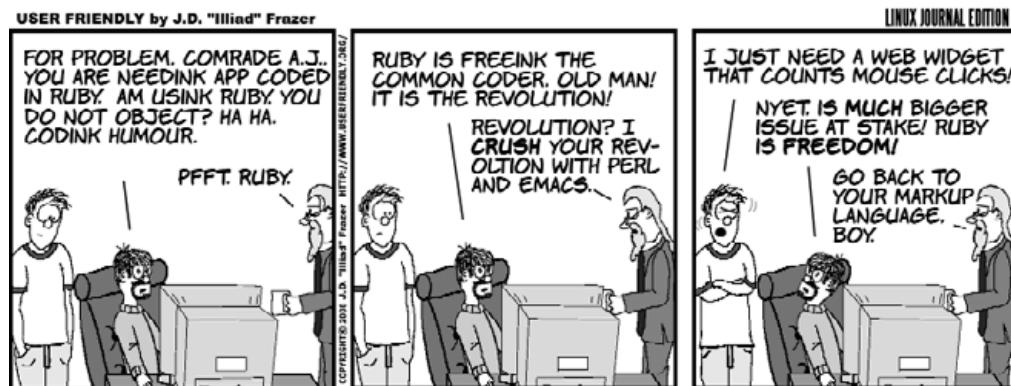
LJ: How do you plan to evolve the service?

AR: To start, we're going for composition, layout and effects. Next is printing and distribution. We also plan to make everything publishable to blogs though the metaweblog API.

Tabblo also makes the most of what's "mashable" from other Web services, over open APIs. When Antonio showed me Tabblo's beta service at O'Reilly's eTech Conference in February 2006, he populated my own Tabblo library by copying more than 6,000 photos from my collection at Flickr. He did this also to demonstrate the well-behaved nature of Flickr as a Net citizen. Rather than locking up customer data, Flickr is wide open, allowing users on other services to access and use photos, including all kinds of photo metadata, as well as tags. Tabblo, he told me, also aims to be equally responsible to what he calls "the open marketplace".

Thus, what's open about Linux-based infrastructure expands out to support whole marketplaces.

—Doc Searls



Servers DIRECT™ GO STRAIGHT TO THE SOURCE!

Get the server that takes your business as seriously as you do
Intel® Xeon™ Processor powering ServersDirect Systems provide the quality and dependability to keep up with your growing business.



\$2,399

1U Appliance Server SDR-1104T

Most affordable 1U Xeon Server and expandable for growing business



- Intel Xeon™ 2.8 Ghz with 800MHz 2MB Cache (Dual Processor Option)
- 1U rack designed
- Intel® E7520 (Lindenhurst) Chipset
- Kingston 1024MB DDR 333 ECC Reg.memory
- Western Digital 250GB SATA RAID Drive
- 4 x 1" Hot-swap SAS / SATA Drive Bays
- Integrated ATI Rage XL SVGA PCI video controller
- Intel® 82546GB dual-port Gigabit Ethernet Controller
- RAID 0, and 1 support

\$1,599

2U Xeon™ Application Server SDR-2116T

Optimal solution for mail and messaging, databases, and applications



- Intel Xeon™ 2.8 Ghz with 800MHz 2MB Cache (Dual Processor Option)
- 2U Chassis with 460W Power Supply
- Intel® E7320 (Lindenhurst VS) Chipset
- Kingston 1024MB DDR400 ECC Reg.memory
- Western Digital 250GB SATA RAID Drive
- 6 x 1" Hot-swap SATA drive bays
- Integrated ATI Rage XL SVGA PCI video controller
- One Intel® 82541PI Gigabit Ethernet controller
- Dual-channel SATA controller supporting RAID 0 and 1

\$1,199

3U Xeon™ Storage Server SDR-3309T

Provides optimal solution for Databases and applications



- Intel Xeon™ 2.8 Ghz with 800MHz 2MB Cache (Dual Processor Option)
- 3U Chassis with 650W Redundant Power Supply
- Intel® server board with Intel® E7520 (Lindenhurst) Chipset
- Kingston 1024MB DDR333 ECC Reg. (2x512MB)
- 1pc x Western Digital 250GB SATA RAID Drive
- 16 x 1" Hot-swap SATA drive bays
- Integrated ATI Rage XL SVGA PCI video controller
- Dual Intel® Gigabit Ethernet Controllers
- 2 SATA Channels with RAID 0 and 1

\$2,699

5U Advanced Storage Server SDR-5302T

The absolute best storage capacity available in a 5U format, at an incomparable cost/TB ratio



- Intel® Xeon™ Processor 3.0Ghz with 800FSB 2MB Cache (Dual Processor Option)
- 5U Chassis with 24 hot-swap bays & 950W redundant power supply
- Intel® E7520 (Lindenhurst) Chipset
- Kingston 1024MB DDR2 400 ECC Reg. (2x512MB)
- 12pcs x Western Digital 250GB SATA RAID Drive
- 1pc x 3Ware 9550SX-12 port RAID Controller Card
- Integrated ATI Rage XL SVGA PCI video controller
- Dual Intel® 82541PI Single-Port Gigabit Ethernet Controllers

\$5,999

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

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

1.877.727.7887 | WWW.SERVERSDIRECT.COM

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





REUVEN M. LERNER

Creating Mashups

It's a crime not to mashup two or more Web services to deliver more than they can deliver separately.

Last month, we started to look at the Google Maps API, which allows us to embed dynamic (and Ajax-enabled) maps into our Web applications. That article demonstrated how easy it is to create such maps, with markers on the screen.

This month, we try something far more ambitious. Specifically, we're going to join the ranks of those creating mashups, combinations of Web services that often (but not always) have a mapping component. A mashup is a combination of two or more Web APIs in a novel way, making information

**Listing 1.
mashup.rhtml, the First (Simple) Version of Our Map**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <script
src="http://maps.google.com/maps?file=api&v=1&key=
➥ABQIAAAAQQK9JhAXQ9eq-G55q\gu
➥1ExTnGAXA-Bs2i826H4DeSQaC3Vqy-xSjDFaTYA00N5YPQWIeUbqJMhhbA"
      type="text/javascript"></script>
  </head>
  <body>
    <h2>Here is your map</h2>
    <div id="map" style="width: 400px; height: 400px"></div>
    <script type="text/javascript">

      var map = new GMap(document.getElementById("map"));
      map.centerAndZoom(new GPoint(-87.740070, 42.037030), 13);

      map.addControl(new GSmallMapControl());
      map.addControl(new GMapTypeControl());

<% array = [-87.740070, -87.730000] %>
<% array.each_with_index do |item, index| %>

      var myMarker<%= index %> = new GMarker(new GPoint(<%= item %>,
42.037030));
      map.addOverlay(myMarker<%= index %>);

<% end %>
    </script>
    <h2>Here are the locations</h2>
    <ul>
<% array.each do |item| %>
      <li><%= item %></li>
<% end %>
    </ul>

  </body>
</html>
```

more accessible and informative than it would be on its own.

One of the first mashups I saw was the Chicago crime map. The Chicago Police Department publishes a regular bulletin of crimes that have taken place within the city, and their approximate locations. Using this map, you can determine how safe your block is from crime, as well as look for patterns in other areas of the city. This mashup took information from the Chicago Police Department's public information and displayed it on a Google Maps page.

I was living in Chicago at the time it came out, and (of course) used the listing to find out just how safe my neighborhood was. The information always had been available from the police department, but it was only in the context of a mapping application that I really was able to understand and internalize this data. And indeed, this is one of the important lessons mashups have taught us—that the synthesis of information and an accessible graphic display, can make a great deal of difference to end users.

When mapping software was first made available, there was no official way to use the maps for unofficial purposes. A number of enterprising developers looked at the JavaScript used to create the maps and reverse-engineered APIs for their own use. Google, as well as Yahoo and MapQuest, have since released APIs that make it possible for us to create mapping applications using their systems. This has made mashups with maps even more popular than ever, with a growing number of Web sites and blogs examining them.

This month, I demonstrate a simple mashup of Google Maps with Amazon's used-book service. The application will be relatively simple. A user will enter an ISBN, and a Google map of the United States will soon be displayed. Markers will be placed on the map indicating several of the locations where used copies of the book are available. Thus, if copies of a book are available in New York City, Chicago and San Francisco, we will see three markers on the map, one in each city. In this way, we'll see how two different Web APIs, from two different companies, can be brought together to create an interesting and useful display for end users.

This month's code examples assume you already have signed up for an Amazon Web services ID, as well as for a Google Maps ID. Information on where to acquire these IDs is available in the on-line Resources for this article.

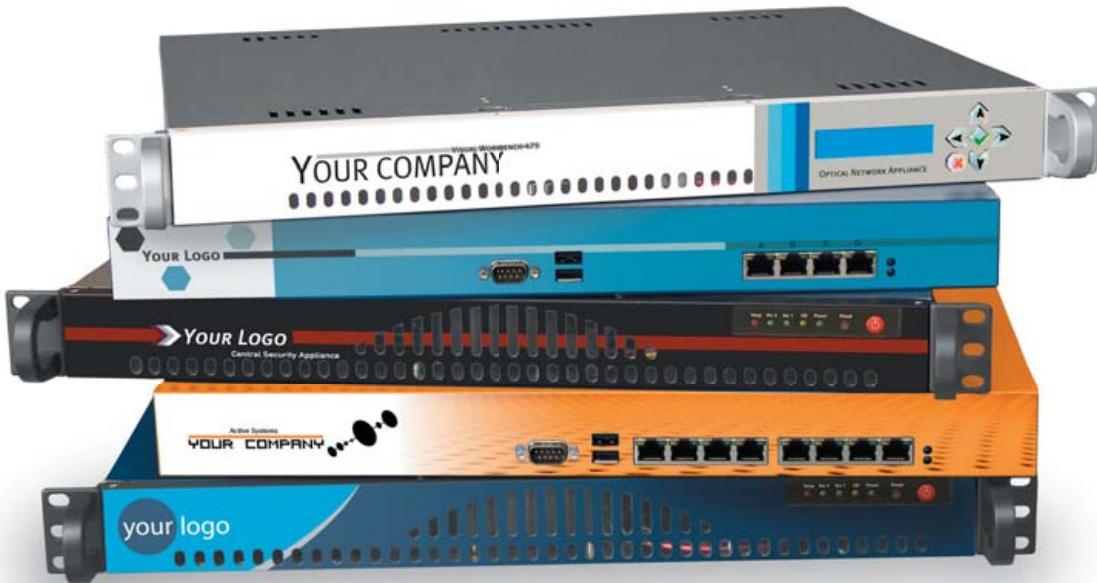
A Simple Map

Our first challenge is to create a map that contains one graphic marker for each location in a list. We already saw how to do this last month using PHP. This month, we begin by converting the program to ERB, an ASP- or PHP-style template that uses Ruby instead of another language. You can see the file, *mashup.rhtml*, in Listing 1.

One way to parse ERB files correctly on a server is by running Ruby on Rails, which uses ERB as a default templating mechanism. But for a small mashup like this, using Rails would be overkill. So, I decided to use a simple ERB (Embedded Ruby, for HTML-Ruby templates) by itself.

To make this work, I installed eruby in the cgi-bin directory of my server (see Resources). I then told Apache that any file

The Industry Leader for Server Appliances



Custom server appliances or off the shelf reference platforms,
built with your image and software starting under \$1,000.

From design to deployment, we handle it all.

Delivering an appliance requires the right partner. MBX Systems is the right partner. We understand that by putting your name on our hardware, you're putting your reputation in our hands. We take that seriously. We provide the services you need to service your customers. Better than the competition. You never even

need to touch the hardware. Engineering. Design. Deployment. We handle it all, so you can focus on what's important to you. Your software. Your sales. Your success.

Visit us at www.mbx.com or call 1-800-681-0016 today.



www.mbx.com | 1-800-681-0016

with an .rhtml extension should be parsed with eruby:

```
AddType application/x-httplib-eruby .rhtml
Action application/x-httplib-eruby /cgi-bin/eruby
```

After restarting the server, I was able to create HTML-Ruby templates without any problems, so long as they had an .rhtml extension. The file in Listing 1, `mashup.rhtml`, was a simple attempt at using my HTML-Ruby template to create a map. As with all Google Maps applications, our final output will be a page of HTML, including some JavaScript that invokes functions downloaded from the Google Maps server. Our Ruby code will be outputting JavaScript code, which will then execute in the user's browser.

To demonstrate that we can indeed do this for two fixed points, the ERB file defines an array of two latitudes, both within a short distance of my home in Skokie, Illinois:

```
<% array = [-87.740070, -87.730000] %>
```

Next, we iterate over the elements of this array, using the `each_with_index` method to get both the array element and the index within the array that we are currently on:

```
<% array.each_with_index do |item, index| %>
```

Now that we have both the latitude and a unique number for it, we can output some JavaScript:

```
var myMarker<%= index %> = new GMarker(new GPoint(<%= item %>, 42.037030));
map.addOverlay(myMarker<%= index %>);
```

What is happening in the above code isn't hard to understand, but it might be a bit complicated when you first read it. Basically, each iteration of our loop declares a new JavaScript variable. The first iteration creates `myMarker0`, and the second creates `myMarker1`. This is possible because we have the index of the current Ruby array element, and because we have made sure not to insert any spaces between `myMarker` and the Ruby output `<%= index %>`.

The `myMarkerX` variable is then defined to be a new instance of `GMarker`—that is, a marker on the Google map—located at a point defined by the latitude (the `item` variable) and longitude (a fixed value, 42.037030).

Finally, so that the user can see exactly where all of the points are, we print some text at the bottom of the page. The result is a map with two markers on it, and the location of each marker is listed in text.

Working with Addresses and Cities

This map is a nice start, but far from what we want to accomplish. And, one of the biggest impediments is the fact that Google Maps expects to get longitude/latitude pairs. Amazon's Web service does return information about third-party vendors, but it provides us with city and state information. So, we need a way to translate city and state names into latitude and longitude.

The easiest way to do this is to rely on someone else, who can translate an address into a longitude/latitude pair. Such geocoder services exist as Web services on the Internet; some of them are freely available, and others charge money. One of the best-known free geocoder services is at geocoder.us. To use this geocoder, we simply use a REST-style URL, as follows: `http://geocoder.us/service/rest?address=ADDRESS`, replacing ADDRESS with the place we want to go. For example, to find my house, we would say, `http://geocoder.us/service/rest?address=9120+Niles+Center+Road+Skokie+IL`.

The geocoder service returns an XML document that looks

Listing 2. `mashup2.rhtml`

```
<% require 'net/http' %>
<% require 'rexml/document' %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <script
src="http://maps.google.com/maps?file=api&v=1&key=
➥ABQIAAAQK9JhAXQ9eq-G55q
➥gu1ExTnGAXa-Bs2i826H4DeSQuC3Vqy-xSjDFaTYA00N5YPQWIeUbqJMhhbA"
      type="text/javascript"></script>
  </head>
  <body>
    <h2>Here is your map</h2>
    <div id="map" style="width: 400px; height: 400px"></div>
    <script type="text/javascript">

      var map = new GMap(document.getElementById("map"));
      map.centerAndZoom(new GPoint(-87.740070, 42.037030), 13);

      map.addControl(new GSmallMapControl());
      map.addControl(new GMapTypeControl());

      <% final_list = [ ] %>
      <% cities = ["Skokie,IL,US", "Longmeadow,MA,US",
      "Somerville,MA,US", "Old+Westbury,NY,US"] %>
      <% cities.each_with_index do |city, index| %>
        <% geocoder_response =
          Net::HTTP.get_response('brainoff.com',
          "/geocoder/rest/?city=#{city}") %>
          <% xml = REXML::Document.new(geocoder_response.body) %>
          <% longitude = xml.root.elements["/rdf:RDF/geo:Point/geo:long"].text %>
          <% latitude = xml.root.elements["/rdf:RDF/geo:Point/geo:lat"].text %>
          <% final_list << {'city' => city, 'longitude' => longitude,
          'latitude' => latitude } %>
          var myMarker<%= index %> = new GMarker(new GPoint(<%= longitude %>,
          <%= latitude %>));
          map.addOverlay(myMarker<%= index %>);
      <% end %>
    </script>

  </body>

  <h2>Your cities</h2>
  <table border="1">
    <tr>
      <th>City</th>
      <th>Longitude</th>
      <th>Latitude</th>
    </tr>
    <% final_list.each do |city| %>
      <tr>
        <td><%= city['city'] %></td>
        <td><%= city['longitude'] %></td>
        <td><%= city['latitude'] %></td>
      </tr>
    <% end %>
  </table>
  </body>
</html>
```

iSCSI that makes your IP SAN a whole lot richer...

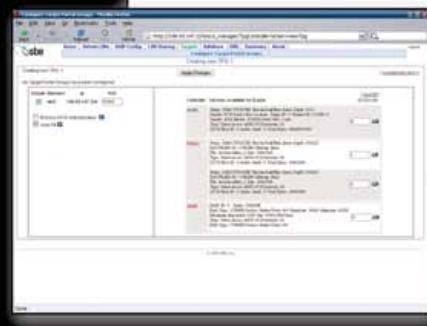
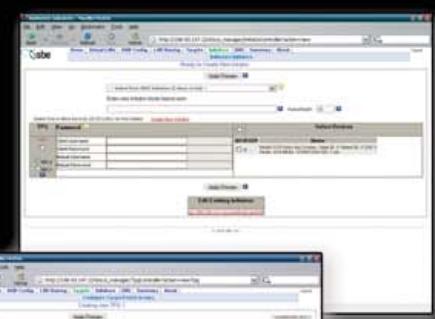


SBE's feature-rich iSCSI software solutions deliver enterprise-level reliability at affordable price points...

Highly scalable & fully standards compliant iSCSI Target and Initiator protocol stacks.

Early iSCSI deployments did not offer the high-quality transport and fault tolerance needed by enterprise storage managers. However, SBE's PyX iSCSI solutions were among the industry's first to provide the full error recovery features required by enterprise storage managers, namely Error Recovery Level 2 (ERL2). Providing broad Linux OS support, SBE's iSCSI protocol stack enables robust, cost-effective IP-based storage transport solutions delivering fault-tolerant, high availability networks with a rich mix of advanced feature sets, including:

- ▶ Proven performance from Wi-Fi to 10G with unlimited port count
- ▶ Aggregation of all available bandwidth across subnets and ports
- ▶ Multi-pathing I/O functionality for maximum redundancy and reliability
- ▶ Performance as fast as your underlying hardware can go
- ▶ Storage media and transport independent architecture
- ▶ Full interoperability with all compliant iSCSI initiators



like this:

```
<rdf:RDF>
<geo:Point rdf:nodeID="aid77952462">
  <dc:description>9120 Niles Center Rd, Skokie IL 60076</dc:description>
  <geo:long>-87.743874</geo:long>
  <geo:lat>42.046517</geo:lat>
</geo:Point>
</rdf:RDF>
```

Because the longitude and latitude are nicely compartmentalized inside of the XML, it's easy to extract it in our program and then insert it into the JavaScript that we generate. However, from looking through the geocoder.us documentation, it doesn't seem as though it is able to handle city names (that is, without street addresses).

Luckily, at least one free geocoder service handles city names, returning a similarly styled XML document. We submit the name of a city as follows, once again using a REST-style request: <http://brainoff.com/geocoder/rest?city=Skokie,IL,US>.

We get the following result:

```
<rdf:RDF>
<geo:Point>
  <geo:long>-87.762660</geo:long>
  <geo:lat>42.034680</geo:lat>
</geo:Point>
</rdf:RDF>
```

As you can see, the longitude and latitude points we got back from this query are slightly different. If we were looking to create a map for driving directions, this would be of greater importance. But, we already know that we'll be looking at the entire map of the United States for this application, and that being blocks away, or even two miles away, won't make any difference.

We can now update our ERB file, such that it has an array of cities, rather than longitude/latitude pairs, as you can see in Listing 2. We begin the file by importing two Ruby classes that will be needed to handle this additional functionality:

```
<% require 'net/http' %>
<% require 'rexml/document' %>
```

Although our starting (and centering) point begins at the same longitude/latitude location, we begin at zoom level 13, which will be large enough to show all of the cities.

We then define four cities, putting them in an array called cities, showing four of the US cities in which I have lived. Notice that each element of this array is a string containing a city name, state abbreviation and US (for United States). Also note that when the city name has a space, we must replace it with a + sign (or %20), so the Web service request works appropriately:

```
<% cities = ["Skokie,IL,US", "Longmeadow,MA,US",
  "Somerville,MA,US", "Old+Westbury,NY,US"] %>
```

We then iterate through these cities, using each as the argument to our Web service geocoder:

```
<% geocoder_response =
  Net::HTTP.get_response('brainoff.com', "/geocoder/rest/?city=#{city}") %>
```

The results of the geocoder Web service are in XML, as we saw earlier. To extract the results of this query from the XML, we use the REXML library that comes with Ruby. This allows us to retrieve the geo:long and geo:lat elements, and then grab the textual contents of the elements:

```
<% longitude = xml.root.elements[/rdf:RDF/geo:Point/geo:long].text %>
<% latitude = xml.root.elements[/rdf:RDF/geo:Point/geo:lat].text %>
```

Having done the hard work, we now insert the appropriate JavaScript:

```
var myMarker<%= index %> = new GMarker(new GPoint(<%= longitude %>,
<%= latitude %>));
map.addOverlay(myMarker<%= index %>);
```

Along the way, we collect city names and locations into an array named final_list. We can then use this to produce a list at the end of the document:

```
<% final_list.each do |city| %>
<tr>
  <td><%= city['city'] %></td>
  <td><%= city['longitude'] %></td>
  <td><%= city['latitude'] %></td>
</tr>
<% end %>
```

Sure enough, this produces a page with a Google map showing all of those locations, and with a list at the bottom.

Adding Amazon Information

Although the above is nice to have, the city information is still hard-coded. What we want is to be able to retrieve information about third-party sellers of a particular book. This means we must get an ISBN from the user, ask Amazon for third-party sellers of that book, and then get the city and state in which each of those sellers resides. Our code will remain largely the same, except for the way we define the cities array, which will be far more complicated. You can see the resulting code in Listing 3.

Getting an ISBN from the end user is fairly straightforward. At the top of the file, we import the CGI class:

```
<% require 'cgi' %>
```

Now we can retrieve an ISBN that the user entered:

```
<% cgi = CGI.new %>
<% isbn = cgi['isbn'] %>
```

We use this ISBN to find all of the third-party sellers with a copy of this book. (Actually, we're going to look at only up to ten of the third-party vendors; Amazon returns only ten items at a time, and we won't complicate our code by looking for additional pages of results.) We take each returned vendor and put it into our vendors array.

So, let's start by getting information about vendors of used copies of our book. We do this by sending Amazon a REST request for our ISBN:

```
amazon_params = {'Service' => 'AWSECommerceService',
  'Operation' => 'ItemLookup',
  'AWSAccessKeyId' => 'XXX',
  'ItemId' => isbn,
  'ResponseGroup' => 'Medium,OfferFull',
  'MerchantId' => 'All'}.map { |key,value|
  "#{key}=#{value}"}.join("&")

amazon_response = Net::HTTP.get_response('webservices.amazon.com',
  '/onca/xml?' <<
  amazon_params)
```

The above is my preferred technique for keeping track of names and values, especially when I'm passing a lot of them—I create a hash, joining



Are you
shocked
by the
high cost
of iSCSI &
Fibre Channel
storage?

AoE is your answer!

ATA-over-Ethernet = simple, low cost, expandable storage.

www.coraid.com



EtherDrive® SR1520

- RAID enabled 3U appliance with 15 slots for hot swap SATA disks
- Check out our other Storage Appliances and NAS Gateway



1. Ethernet Storage – without the TCP/IP overhead!
2. Unlimited expandability, at the lowest possible price point!!
3. You want more storage...you just buy more disks – it's that simple!!!

Visit us at www.coraid.com for more information.



1.706.548.7200

The Linux Storage People

www.coraid.com

Listing 3.

Adding Amazon Information

```

<% require 'cgi' %>
<% require 'net/http' %>
<% require 'rexml/document' %>
<% cgi = CGI.new %>
<% isbn = cgi['isbn'] %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Where you can find used copies of ISBN <%= isbn %></title>
  <script
src="http://maps.google.com/maps?file=api&v=1&key=
ABQIAAAQK9jhAXQ9eq-G55qgu1ExTnGAXa-Bs2i826H4De5QaC3Vqy-
xSjDFaTYA00N5YPQWIeUbqJMhhbA"
  type="text/javascript"></script>
</head>
<body>
  <h2>Here is your map</h2>
  <div id="map" style="width: 800px; height: 600px"></div>
  <script type="text/javascript">

    var map = new GMap(document.getElementById("map"));
    map.centerAndZoom(new GPoint(-87.740070, 42.037030), 13);

    map.addControl(new GSmallMapControl());
    map.addControl(new GMapTypeControl());

<% final_list = [ ] %>

<% amazon_params = {'Service' => 'AWSECommerceService',
'Operation' => 'ItemLookup',
'AWSAccessKeyId' => 'XXX',
'ItemId' => isbn,
'ResponseGroup' => 'Medium,OfferFull',
'MerchantId' => 'All'}.map {|key,value|
"#{key}=&#{value}"}.join("&")

amazon_response =
Net::HTTP.get_response('webservices.amazon.com',
                      '/onca/xml?' <<
                      amazon_params)

xml = REXML::Document.new(amazon_response.body)

# Get the vendors, and use that information to get their locations
cities = []
xml.root.elements.each("Items/Item/Offers/Offer/Seller/SellerId") do |seller|
  # Now get information about each vendor
  amazon_vendor_params = {'Service' => 'AWSECommerceService',
'Operation' => 'SellerLookup',
'AWSAccessKeyId' => 'XXX',
'SellerId' => seller.text}.map {|key,value|
"#{key}=&#{value}"}.join("&")

  vendor_response =
Net::HTTP.get_response('webservices.amazon.com',
                      '/onca/xml?' <<
                      amazon_vendor_params)
  vendor_xml = REXML::Document.new(vendor_response.body)

  vendor_city =
  vendor_xml.root.elements[/SellerLookupResponse/Sellers/
Seller/Location/City"].text

  vendor_state =
  vendor_xml.root.elements[/SellerLookupResponse/Sellers/
Seller/Location/State"].text

  cities << "#{vendor_city},#{vendor_state},US"
end

cities.each_with_index do |city, index|
  geocoder_response =
  Net::HTTP.get_response('brainoff.com',
"/geocoder/rest/?city=#{city.gsub(' ', '+')}")
  geocoder_xml = REXML::Document.new(geocoder_response.body)

  next if geocoder_xml.root.nil?

  longitude =
  geocoder_xml.root.elements[/rdf:RDF/geo:Point/geo:long].text
  latitude =
  geocoder_xml.root.elements[/rdf:RDF/geo:Point/geo:lat].text

  final_list << {'city' => city, 'longitude' => longitude,
'latitude' => latitude } %>

  var myMarker<%= index %> = new GMarker(new GPoint(<%= longitude %>,
<%= latitude %>));
  map.addOverlay(myMarker<%= index %>);

<% end %>
</script>

<body>

  <h2>Your cities</h2>
  <table border="1">
    <tr>
      <th>City</th>
      <th>Longitude</th>
      <th>Latitude</th>
    </tr>
    <% final_list.each do |city| %>
      <tr>
        <td><%= city['city'].gsub(" ", " ") %></td>
        <td><%= city['longitude'] %></td>
        <td><%= city['latitude'] %></td>
      </tr>
    <% end %>
  </table>
</body>
</html>

```

Feed a cold, starve a fever...
better yet:

Avoid a virus. Use BitDefender®!

At least that's what
41 million of our users say.

BitDefender is an award-winning producer of anti-virus software and data security solutions providing protection to satisfy the requirements of today's computing environment.

- **Best reaction to malware as certified by: Andreas Marx (www.av-test.org)**
- **Available for mail and file servers**
- **Answers to critical administrator needs: performance, stability and compatibility**
- **Compliant with FreeBSD 4 & 5 as well as all popular Linux distros**



BitDefender® 9 Standard
PC World Best Buy
March 2006



Now you should feel quite a BIT better.



Try it for free | www.bitdefender.com

the keys and values with = signs, and then the pairs themselves with ampersands (& signs). This gives me a string that I can hand to Amazon.

The XML response that I get back then contains a lot of information, including details about each offer. That's actually all I care about here; I'm not keeping track of the price of the book (which would be useful, of course), but rather the location of each used copy we can grab. But we can't get that right away; the ItemLookup request gets us only the seller IDs and some basic information about each one. We'll need to grab the seller ID from each offer node, then use that to perform a second Amazon request, obtaining information about the vendor:

```
xml.root.elements.each("Items/Item/Offers/Offer/Seller/SellerId") do
|seller|
  # Now get information about each vendor
  amazon_vendor_params = {'Service' => 'AWSECommerceService',
                           'Operation' => 'SellerLookup',
                           'AWSAccessKeyId' => 'XXX',
                           'SellerId' => seller.text}.map { |key,value|
    "#{key}=#{value}"}.join("&")

  vendor_response = Net::HTTP.get_response('webservices.amazon.com',
                                             '/onca/xml?' <<
                                             amazon_vendor_params)
  vendor_xml = REXML::Document.new(vendor_response.body)
```

This code sends a request to Amazon, gets an XML body back, and then looks for the City and State elements that a vendor will produce. Unfortunately, there's no fast and easy way to deal with countries outside of the United States, both with geocoding and with Amazon. Amazon's assumption seems to be that Canada is sort of like the United States, which is false. So, we'll always get the city and state and assume that it is in the United States. If our assumption turns out to be wrong, we'll allow ourselves to be corrected by the geocoder.

As we have grabbed information about each vendor, we have stuck the city and state information in the cities array. Now we're going to use that same array, just as we did in `mashup2.rhtml`—except now, the source is not a hard-coded list, but rather one that we put together from Amazon information. We had to make only two changes for things to work: a check that we didn't get nil from the geocoder (indicating there was an error, often because the vendor is in Canada), and a use of gsub to change space characters into + signs in the city name.

The results are quite nice to see, even if they're incomplete and a bit on the crude side: By going to a URL such as <http://maps.lerner.co.il/mashup3.rhtml?isbn=0812931432>, we can see where a number of used copies are located in the United States. This doesn't necessarily reflect the cost of the book, its condition, or the shipping charges—but it can be fun and interesting to see where different books have ended up, and which cities tend to have more (and fewer) used books.

Conclusion

Creating mashups, combinations of existing Web services, can be a great deal of fun, and can make it easier to see patterns in data by putting them on a map. It requires that you have a good understanding of the underlying technologies and their quirks—but with a bit of work, you'll see that creating such mashups can be fun and exciting, and even entertaining. Moreover, as the Web becomes increasingly interconnected, and as applications continue to blur the distinction between the desktop and the Web, we should expect to see more of such mashups, rather than fewer of them.■

Resources for this article: www.linuxjournal.com/article/9013.

Reuven M. Lerner, a longtime Web/database consultant, is currently a PhD student in Learning Sciences at Northwestern University in Evanston, Illinois. He and his wife recently celebrated the birth of their son Amotz David.



MARCEL GAGNÉ

A Gem of an Idea

A play on the word ruby can provide playtime for you. MARCEL GAGNÉ

What you have come up with, François? Yes, I am excited too, and I am very happy to hear that you found it so easy to come up with today's menu. I must confess, *mon ami*, I did have some concerns when you told me you wanted to put together the menu for the Ruby issue. You never told me that you were so interested in object-oriented programming. I'm rather curious as to what made you choose Ruby.

François? Why do you look so confused? What do you mean, I'm the one who is confused? Let me have a look at this menu of yours. Oh, I see. No, François, I am not laughing at you. Well, maybe a little. None of these programs have anything to do with Ruby. Yes, I can see that every one of them features rubies, but Ruby doesn't have anything to do with rubies. At least not directly, and as near as I can tell, none of these programs are written in Ruby. No matter, François. Our guests will be here any moment and we must be ready for their arrival. Don't worry about the menu. Everything will be fine.

Welcome, everyone, to *Chez Marcel*, the home of fine wine and exquisite Linux fare. Please, sit and make yourselves comfortable. François, please head down to the cellar and bring back the case of 2000 Castell de Falset Tinto

from Penedes, Spain. It is located midway on the right side of aisle four in the south wing. *Vite, mon ami!* Our guests are no doubt parched.

While my faithful waiter is in the cellar, I should tell you that the items on today's menu were all his choices. When I mentioned a Ruby special issue, he jumped at the chance and, through a thoroughly understandable misunderstanding, wound up putting together a collection of games with rubies in them. On that note, I'd like to introduce the first item on tonight's menu. Those of you who are running the KDE desktop will probably find you already have a copy of *KSokoban*, written by Anders Widell. The word *sokoban* is a Japanese word meaning "warehouse keeper". *KSokoban* can be seriously challenging, but the premise of the game is simple. Your warehouse keeper needs to push these large, red, ruby-like gems into proper storage locations in the warehouse. The storage locations are indicated by glassy, green circles. Every gem must be moved into storage (Figure 1).

Sounds easy—and it is. At first. With each successive level, *KSokoban* becomes increasingly difficult. The catch is that you can push only one gem at a time. Okay, there's more than just one catch. You also can't pull the gems, and you can't step over them. Push them up against a wall with no way to get behind them and you are stuck.

Ah, wonderful, François! You have returned with the wine. Please, pour for our guests. Enjoy the wine, *mes amis*. This red is actually a blend of several varietals—rich and complex.

Complexity in *KSokoban* comes with successive levels. There also are collections of levels, and even those collections vary in terms of complexity. If you find yourself overwhelmed, or if you would like a kinder, gentler version of *KSokoban* that your younger kids can enjoy, click Game on the menu bar and select *Microban* (easy) from the level collection. It's a great way to build up your confidence before submitting yourself to the Sasquatch levels.

Sebastien Delestain's *Gweled* is a Linux port/remeake of a game called *Diamond Mine*, also known as *Bejeweled*. This gem of a game features an array of precious stones from diamonds to emeralds to rubies and more (Figure 2). The idea is to line three identical stones in a row (diagonally does not work). Your only method for accomplishing this feat is to click on a stone and then another adjacent stone. This causes the stones to flip their position. This works only if the flip generates a three-in-a-row combination. Once three stones line up, those above fall and the grid is replenished. The object is to play until there are no further moves and, of course, to generate the highest possible score in that time frame.

Under the Preferences menu, you can select the size of the game grid. Mind you, this is a physical size. The number of jewels on the grid remains the same. You also can select a somewhat more difficult—and just a tad more nerve-wracking



Figure 1. *KSokoban* is a challenging and fascinating puzzle. Look for it as part of your KDE collection of games.

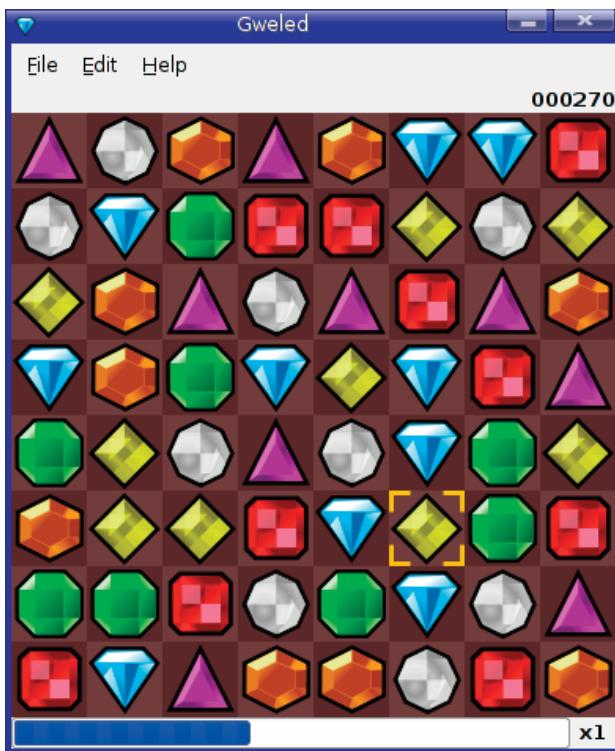


Figure 2. *Gweled* is attractive, fun to play and surprisingly addictive.

All Joe Wreschnig, the author of *Angry, Drunken, Dwarves*, can tell us is that they have decided to take this alcohol-fueled aggression out on their fellow dwarves by dropping gems on their heads.

ing—timed version of the game where you race against the clock for points. *Gweled* is easy, fun to play and makes for a highly successful time waster.

As you all enjoy the wine François is busy refilling, I note a happy and appreciative look on your faces, even from those of you who are on to your second or third glass. This makes the following item on tonight's menu somewhat ironic, a great arcade game called *Angry, Drunken, Dwarves*.

Why these dwarves are drunk and angry evades even the developer of the game. All Joe Wreschnig, the author of *Angry, Drunken, Dwarves*, can tell us is that they have decided to take this alcohol-fueled aggression out on their fellow dwarves by dropping gems on their heads. Rubies, emeralds, diamonds—they are all fair game in this arcade diversion (Figure 3).

During the course of the game, colored gems drop slowly from above (gravity must not be as strong deep

Hurricane Electric Internet Services...Speed and Reliability That You Can Depend On!

Flat Rate Gigabit Ethernet

1,000 Mbps of IP

\$13,000/month*

Full 100 Mbps Port

Full Duplex

\$2,000/month

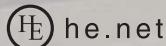
Colocation Full Cabinet

Holds up to 42 1U servers

\$400/month

Order Today!

email sales@he.net or call 510.580.4190



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



Figure 3. The computer's drunken dwarf doesn't look so tough.



Figure 4. Each angry, drunken, dwarf comes with his own picture, attack pattern and bio.

underground). As they drop, you use the cursor keys—or the keyboard in a multiplayer game—to rotate and position the gems. This is more than just a *Tetris* clone. *Angry, Drunken, Dwarves* is also a strategy game. Assemble the falling gems in a rectangular combination, and the gems will fuse into larger gems, or crystals. Circular, star-like gems are called break gems when they hit a gem (or crystal) of the same color. Crystals, as you might expect, are worth more than gems when smashed by a break gem.

Now, I did say that this was a multiplayer game, which means playing against another flesh-and-blood opponent or a computer-generated one. At the beginning of the game, each player chooses a dwarf. To help you choose a player, *Angry, Drunken, Dwarves* presents you with pictures, names and bios of the dwarf contestants (Figure 4). Each dwarf also has an attack pattern, which you can use to your advantage if you are smart. During game play, your opponent's rules are the same, but whenever you break gems, your opponent drops counter gems on you, which means you have to work twice as hard and twice as fast in order to clear them. As turnabout is indeed fair play, your opponents get the same treatment when they break gems.

There's quite a bit more to the game, but I'm going to let you discover that on your own. Before we move on, however, I want to point out a not-so-apparent aspect to the game that starts to show itself after you have played a few times. On the game selection screen, you will see a menu option labeled View Unlocks, which opens up some otherwise unseen features of the game.

Ah, *mes amis*, it is getting late, the stars are out and shining like jewels—sorry, I couldn't resist—and puns are a diamond a dozen. Sadly, we must start thinking about closing up...soon. François will happily refill your crystal glasses with the ruby-red Castell de Falset as you try your hands at some of tonight's sparkling games. On that note, 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/9014.

Marcel Gagné is an award-winning writer living in Mississauga, Ontario. He is the author of the all new *Moving to Ubuntu Linux*, his fifth book from Addison-Wesley. He also makes regular television appearances as Call for Help's Linux guy. Marcel is also a pilot, a past Top-40 disc jockey, 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 lots of other things (including great Wine links) from his Web site at www.marcelgagne.com.

Small business never stands still.

Choose the Dual-Core Intel® Xeon® Processor in your ZT X9000 series system and experience outstanding performance. The better to run more applications, support more users and get more done in less time.



ZT Dual-Core Server DC X9001

Dual-Core Intel® Xeon® Processor 5060
(2x2MB L2 Cache, 3.20 GHz, 1066MHz FSB, Intel® EM64T)
▪ Intel® 5000P Chipset Server Board
▪ 2GB DDR2 533MHz ECC/ Reg. SDRAM (Up to 64GB)
▪ 6x Seagate® 400GB SATAII Enterprise Hard Drive (RAID 5)
▪ 6x 1" Hot-swap SATAII Drive Bays
▪ Slim DVD-RW and Floppy Drive
▪ Intel® ESB2 6 SATAII (3Gbps) Host RAID Controller
▪ 8 Channel SATAII RAID Controller (RAID 0,1,5,10 Ready)
▪ Dual Intel® 82563EBI Gigabit Ethernet Controller
▪ 2U Rackmount Chassis w/ 500W Power Supply
▪ SuperDoctor III Server Management Software
▪ 3-Year Limited Warranty + 1-Year On-site Service

\$3,999



ZT Dual-Core Server DC X9002

2 x Dual-Core Intel® Xeon® Processor 5060
(2x2MB L2 Cache, 3.20 GHz, 1066MHz FSB, Intel® EM64T)
▪ Intel® 5000P Chipset Server Board
▪ 2GB DDR2 533MHz ECC/ Reg. SDRAM (Up to 64GB)
▪ 8x Seagate® 500GB SATAII Enterprise Hard Drive (RAID 6)
▪ 8x 1" Hot-swap SATAII Drive Bays
▪ Dual-layer DVD-RW / CD-RW and Floppy Drive
▪ Adaptec® 8 Channel 64bit SATAII RAID Controller (RAID 5, 6 Ready)
▪ Dual Intel® 82563EBI Gigabit Ethernet Controller
▪ 4U Rackmount/Tower Compatible Chassis
▪ w/ Triple Redundant 760W Power Supply
▪ SuperDoctor III Server Management Software
▪ 3-Year Limited Warranty + 1-Year On-site Service

\$5,999

- New powerful Intel® Dual-core processing server platform which maximizes Linux performance.
- Virtual technology, more secure, more interoperable, and more supportable.
- Save time and reduce operations costs by Dual Core technology.

Go to
Call

ztgroup.com/go/linuxjournal

866- ZTGROUP (866-984-7687)

Promote code : Lj0706

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. All rights reserved. All brands and product names, trademarks or registered trademarks are property of their respective companies. Celeron, Celeron Inside, Centrino, Centrino Logo, Core Inside, Intel, Intel Logo, Intel Core, Intel Inside, Intel Inside Logo, Intel SpeedStep, Intel Viiv, Itanium, Itanium Inside, Pentium, Pentium Inside, Xeon and Xeon Inside are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.





DAVE TAYLOR

Recognizing Blackjacks

Win or lose, we're almost done with our shell scripted *Blackjack* game.

Last month, we finally wrangled the issue of Aces being worth either one or 11 by recognizing the value of what I call lazy coding—trying to solve the specific situation rather than creating an elegant and sophisticated general solution. I've since talked with some programmers and found that many don't agree with my philosophy. Indeed, one chap basically said I was completely wrong, especially if the code's to be published—as this column obviously is—and that I should always be putting in maximal effort to present exemplary, elegant solutions to any problems encountered.

But what if, I countered, that would make a program two times longer? Or four times longer? He looked at me blankly and asked, "Since when does the length of a program matter when you're trying to solve a problem or model a particular behavior?" It's a fair question, but I have to think that's where all of our code bloat comes from—where no programs can be simple, efficient and

able to hammer our square peg into the round hole of shell scripting.

And So, Back to Vegas

Let's get back to our *Blackjack* game and see where we are vis-à-vis complexity, length and functionality. My suspicion is that we're going to be able to wrap up this project by the end of the next column, because it's really all the mechanics that we can create as a script; the game itself is always doomed to be a primitive command-line interface, because, well, it's a shell script. Could we layer an interface with Tk or some other graphical toolkit? Sure, but then, would you really want the underlying code base to be a shell script?

Our code is now getting into the category of seriously long shell scripts, weighing in at 177 lines. That's actually beyond my usual cutoff of 150 lines maximum for a script,

Early on in this column, I joked about eventually rewriting the popular game *Doom* as a shell script, but of course, it's the complexity that would kill us, even if we might well be able to hammer our square peg into the round hole of shell scripting.

streamlined, and everything does everything else, including word processors that can send e-mail, e-mail programs that can do rudimentary page layout, text editors that can convert ASCII to HTML, Web browsers that, well, seem to include just about every feature up to and including KitchenSink.app.

And, don't get me started on the Emacs versus vi philosophy either, okay? Let's just say that an editor that can replace your OS kernel might just be a wee bit over-implemented for the vast majority of users.

The real cost of all this complexity, however, isn't the length of the code. After all, gigs of disk space are dirt cheap nowadays, and even RAM seems to be not much more than a decent meal. The cost is in the complexity itself—in the fact that large, complex systems are harder to work with, harder to debug and harder to make bulletproof than simple systems. You need but look at the challenges Microsoft is facing as it tries to ship its new Windows OS, Vista, before spring 2010.

So, what does this all have to do with shell script programming? I suggest that this discussion is pivotal to all programming tasks, actually, and that if you can't figure out a simple and manageable solution within the world of shell scripting, it might well be time to move to a more sophisticated development environment. Early on in this column, I joked about eventually rewriting the popular game *Doom* as a shell script, but of course, it's the complexity that would kill us, even if we might well be

so we definitely need to wrap this up before it gets too much longer anyway.

The missing piece we'll deal with in this column is to test for a dealer blackjack and player blackjack situation. If you are dealt an Ace and a ten-point card (a ten or a face card) you have a blackjack. If the dealer has a blackjack, the game ends and the player loses. If the player gets a blackjack on the deal, the game ends and the dealer loses (without being able to get additional cards, because a 21-point hand that involves more than two cards does not tie a player's blackjack). If you have the extraordinary situation of both the player and dealer having a blackjack, it's a "push" or tie, just as it is in any other situation where both player and dealer have the same point value in their hands.

The key spot where we need to modify the code to test for blackjacks is where the hands are dealt, and we might as well deal both hands before adding the additional test. Here's the hand-dealing code:

```
player[1]=$(newdeck[1])
player[2]=$(newdeck[3])
nextplayercard=3          # player starts with two cards

dealer[1]=$(newdeck[2])
dealer[2]=$(newdeck[4])
nextdealercard=3          # dealer also has two cards
```

To see whether either hand is a blackjack quickly, we'll

use the handValue function:

```
handValue ${newdeck[1]} ${newdeck[3]}
```

Recall that because you can't return values in functions, handValue simply sets the global variable \$handvalue to the numeric value of the hand. This means the test is straightforward:

```
# test for dealer or player blackjack

handValue ${player[1]} ${player[2]}
playerhandvalue=$handvalue
handValue ${dealer[1]} ${dealer[2]}
dealerhandvalue=$handvalue

echo ""

if [ $playerhandvalue -eq 21 -a $dealerhandvalue -eq 21 ] ; then
    echo "Extraordinary! Both the dealer and player were dealt a blackjack!"
    echo "Unfortunately, this means you didn't win: it's a push (or tie)."
    echo ""
    exit 0
fi

if [ $dealerhandvalue -eq 21 ] ; then
    echo "Darn it! Dealer pulled a blackjack out of his hat. You lose."
    echo ""
    exit 0
elif [ $playerhandvalue -eq 21 ] ; then
    echo "NICE! You got a blackjack and won the game. Payout would be 3:2!"
    echo ""
    exit 0
fi
```

You can see that I simply create playerhandvalue and dealerhandvalue for these numeric tests, and then check whether both are 21, the dealer is 21 and the player is 21. That's all there is to it.

To test the new code, simply slip in either or both of the following lines right before the handValue call above:

```
player[1]=1 ; player[2]=11
dealer[1]=1 ; dealer[2]=12
```

Then you can, um, stack the deck and create the specific test situations you desire.

Almost Done

We've run out of space for this column, but all that's left is to implement the overall play logic. The game as it stands now shuffles the deck, deals cards for the player and dealer, lets the player add new cards ("hit" in *Blackjack* parlance) until he or she is satisfied or exceeds 21 points, and shows the dealer hand. Next column, we'll write the loop that lets the dealer play out its hand and determine who won the hand. No doubt we'll sneak in some interface features too, of course, but one way or the other, next column will be a wrap for this particular scripting project.

This means I'll be ready to tackle something new! If you have a particular scripting project you think could be an interesting study, please drop me an e-mail, and I'll consider it for a future column series. Otherwise, I've been eyeing the game *Yahtzee* pretty closely.■

Dave Taylor is a 26-year veteran of UNIX, creator of The Elm Mail System, and most recently author of both the best-selling *Wicked Cool Shell Scripts* and *Teach Yourself Unix in 24 Hours*, among his 16 technical books. His main Web site is at www.intuitive.com.



HPC Systems, Inc., ~ leading provider of Quad & 8-Way Opteron Servers.



HPC Systems, Inc. is a leading provider of dependable, cost effective Linux high performance computers based on AMD Opteron technology. We build to order (quad to 16-way) AMD Opteron servers for demanding enterprise applications and will debug, deploy, and install onsite to make your HPC servers fully operational in record time. Contact us today to get a no obligation consultation & quote.



Toll free: 888- 725- 3472 . Local: 408- 943- 8282.
Fax: 408- 943- 8341. Email: sales@hpcsystems.com



JON "MADDOD" HALL

Sinking of the USS Proprietary

There isn't much to salvage from the sunken hulls of proprietary design.

I hang out at the small bar and restaurant called *Aldeia dos Piratas* (Village of the Pirates) on my favorite beach. The owners are the Fafa Brothers, Jose and Chuy, and I regularly meet there to have dinner and discuss things with my friends, collectively known as The Dudes.

One day while I was having one of my favorite drinks, one of the Dudes came in very excited and told us that the latest victim of the sea, the *USS Proprietary*, was floundering offshore. Most of the people in the bar were not surprised. We had been expecting something like this for a while.

"Stupid people", said Dennis, one of the head Dudes. "We told them that the ship was too big and clumsy for the shallow waters. It was not able to turn in time and got caught on the shoals. They should have had a smaller, more nimble ship sail into the harbor."

I patiently explained to Dennis the economics of building ever-larger ships that could more efficiently carry ever-larger cargoes of goods, but he was unconvinced.

"Yes", he said, "I understand the issues, but think about the fact that larger ships can go only to the larger ports. Then you have to pay more to ship the goods to the areas where you really want them. Smaller ships can go closer to the other towns and reduce the overland transportation time and costs. You give up flexibility when you use the 'bigger is better' strategy."

Some of the other Dudes started chiming in. "Yes", said Jimmy, "and I wanted to ship just a small desk the other day. If I wanted to send it via one of the big vessels, they wanted me to 'consolidate' it with other items, and would not talk to me unless I needed to ship tons of cargo at one time. It would have taken me months of time to find enough furniture to make the large shippers even notice me. The smaller ship captains were eager to get my business, and I got my desk shipped in the time frame needed for my customer. So who gave me better service, the large ship or the smaller boat? I could not even get the proper telephone number for the larger shipper—I just kept being put on hold."

"What about risk?", I asked. "Don't you feel better about relying on the larger firms with the bigger boats?" Thiago snorted. "And what ship is lying offshore floundering? When I go to try to get restitution for my lost goods, whose throat can I get my hands around? Have you ever really read their warranties and liability limitations? I know where to find the small ship captains—they usually go to my church, my club and are on my Chamber of Commerce. They are local to me and value my business. The money I give them to ship my goods stays in the local economy, and often they come to my shop to buy food. When I give money to the owners of *Proprietary*, they take the money out of the local economy and I never see it again. It may even go out of the country, affecting our balance of payments." Thiago always was the deep thinker of the group.

It was now Chico's turn to add his thoughts, "What about the 'shipping drain'?" We all turned to him with a quizzical look. "Sure", he said, "we train lots of people in how to sail ships, but then they leave the area and go off to find jobs on these big ships. We lose the best people because they do not feel they can get a good job locally. If we could make it easier for them to get a decent living here, they would stay here and build up the shipping industry again. Remember, 'Think globally, ship locally.'"

"But guys", I said, "surely there are jobs that are just too big for a bunch of small- to medium-sized ships to tackle. Jobs that require something the size of the *USS Proprietary*." "Maybe", Dennis agreed, "but you really should think about it and balance the issues. Most of the time the smaller vessels would be just fine, and other times you might be able to break up your shipment to take advantage of the greater flexibility of having smaller boats, rather than having to put all of your shipment on one carrier. This is just one of the advantages of having control over your shipment, instead of relying on a consolidator who puts everything into one black box and refuses to let you exercise control."

With that we all stared out to sea, looking at the other hulks of boats lying on the shoals, whose cargoes went down never to be recovered:

- USS WANG
- USS Apollo
- USS Prime
- USS DG
- USS DEC
- USS Compact

And we wondered when the *Proprietary* would have its final death knell.■

Jon "maddog" Hall is the Executive Director of Linux International (www.li.org), a nonprofit association of end users who wish to support and promote the Linux operating system. During his career in commercial computing, which started in 1969, Mr Hall has been a programmer, systems designer, systems administrator, product manager, technical marketing manager and educator. He has worked for such companies as Western Electric Corporation, Aetna Life and Casualty, Bell Laboratories, Digital Equipment Corporation, VA Linux Systems and SGI. He is now an independent consultant in Free and Open Source Software (FOSS) Business and Technical issues.



Want to increase your ability to administer large numbers of PCs with a minimum amount of people?

Here's why.

Traditional Linux server-client architectures—especially thin client—are complicated, cumbersome approaches.

They require lots of servers (about one for every 30 clients), constant maintenance, teams of administrators and large capital investments.

They are also expensive, inefficient, complicated and not terribly secure.

IDEASsm. The better way

Integrity Diskless Enterprise Architecture Solutions (IDEAS)—based on Red Hat Enterprise Server V3 and 4-32 and 64-bit OS—is the simpler, more effective way to build diskless desktop computing.

It streamlines installation, scalability and system administration.

A single IDEAS OS server easily handles hundreds of users. Fewer servers means fewer system administrators.

Yet systems remain extremely powerful, easily handling the most demanding development and visualization applications without affecting operations of other OS-based systems on the network.

Then you need IDEASsm

Integrity Diskless Enterprise Architecture Solutions

CREM security

Diskless computing is an ideal solution for security-intensive operations because it eliminates classified removable electronic media (CREM). It's the ultimate security because there's no hard drive to track or manage.

A simple technology whose time has come

Diskless computing simplifies network computing (including implementation, maintenance, security and administration) without managing disk images. As a result, it improves your total cost of ownership (TCO).

To find out how diskless computing can simplify your network, come see us at www.IntegrityLinux.com...

and discover a new world of IDEASsm.



DOC SEARLS

Causes and Effects

If Congress doesn't favor Net Neutrality, what are our options? **DOC SEARLS**

Four years ago this month, in July 2002, Lawrence Lessig gave a landmark talk at OSCON called "Free Culture" (see the on-line Resources). In it, he detailed the damage caused to technology and culture by the insane growth of software patents, the limitless enlargement of copyright protection and other absurdities burned into law by clueless congresscritters. And, he chided us for doing nothing about it:

In an interview two days ago, [retiring Congressman J.C.] Watts said, "Here's the problem with Washington: if you are explaining, you are losing." It's a bumper-sticker culture. People have to get it like *that*, and if they don't, if it takes three seconds to make them understand, you're off their radar screen. Three seconds to understand, or you lose. This is our problem. Six years after this battle began, we're still explaining. We're still explaining and we are losing. They frame this as a massive battle to stop theft, to protect property. They don't get why re-architecting the network destroys innovation and creativity. They extend copyrights perpetually. They don't get how that in itself is a form of theft. A theft of our common culture. We have failed in getting them to see what the issues here are and that's why we live in this place where a tradition speaks of freedom and their controls take it away.

Now, I've spent two years talking to you. To us. About this. And we've not done anything yet. A lot of energy building sites and blogs and Slashdot stories. [But] nothing yet to change that vision in Washington. Because we hate Washington, right? Who would waste his time in Washington?

But if you don't do something now, this freedom that you built, that you spend your life coding, this freedom will be taken away. Either by those who see you as a threat, who then invoke the system of law we call patents, or by those who take advantage of the extraordinary expansion of control that the law of copyright now gives them over innovation. Either of these two changes through law will produce a world where your freedom has been taken away. And, If You Can't Fight For Your Freedom...You Don't Deserve It.

But you've done nothing.

Ah, but now we have. In 2005, our bumper sticker appeared and became a rallying cry for pro-Net legislation. Its name: Net Neutrality. The idea is that networks should be open and without prejudice about what they carry. In the

container cargo system of moving packets that comprises the Net, contents of containers should be of no interest to those who own and operate the various parts of the system.

This week, as I write this (in early April), many pro-Neutrality advocates (including myself) have just finished gathering at David Isenberg's Freedom To Connect (F2C) Conference in Washington, DC (see Resources). And they haven't been alone. The Big Boys missing in action when Larry Lessig gave his talk have been all over the case. Amazon, Google, Yahoo and Microsoft, among others, have been lobbying Congress to protect the Net's neutrality. Vint Cerf, one of the Net's fathers (and now VP and Chief Internet Evangelist at Google) said this in his prepared statement for a Senate commerce committee hearing on Net Neutrality:

Allowing broadband carriers to control what people see and do on-line would fundamentally undermine the principles that have made the Internet such a success. For the foreseeable future, most Americans will face little choice among broadband carriers. Enshrining a rule that permits carriers to discriminate in favor of certain kinds or sources of services would place those carriers in control of on-line activity. Allowing broadband carriers to reserve huge amounts of bandwidth for their own services will not give consumers the broadband Internet our country and economy need. Promoting an open and accessible Internet is critical for consumers.

Google believes that consumers should be able to use the Internet connections that they pay for in the way that they want. This principle—that users pick winners and losers in the Internet marketplace, not carriers—is an architectural and policy choice critical to innovation on-line. Google itself is a product of the Internet. We care passionately about the future of the Net, not just for ourselves, but because of all the other potential Googles out there. Indeed, we are not alone. Our concerns are shared by Internet companies, small businesses, end users and consumer groups across the country. The vibrant ecosystem of innovation that lies at the heart of the Internet creates wealth and opportunity for millions of Americans. That ecosystem—based upon a neutral open network—should be nourished and promoted.

At F2C, Representative Rick Boucher, a Virginia Democrat, said he and Rep. Edward Markey (D-Mass) would introduce an amendment to the Telecom Act rewrite (of the last Act, passed in 1996) that says that "if a telephone company or broadband provider of any kind decides to prioritize any content, then they have to offer that same fast-lane treatment to all content providers without charge."

This amendment, Boucher explained, would fortify Net Neutrality provisions already in the bill. Those provisions would give the FCC statutory authority to hear disputes relating to the commission's own principles, which include former FCC Chairman Michael Powell's four "Internet Freedoms" (to access content, to use applications, to attach devices and to obtain service plan information), which have since been watered down under current chairman Kevin Martin (see Resources).

During the conference, however, it became clear that, although most of those present agreed that Net Neutrality is a Good Thing to have, there was little agreement about the wisdom of burning it into law. I was one of those whose advocacy of Net Neutrality is tempered by misgivings about burning it into law (see Resources for a link to my article on the *LJ* Web site). That's what I said on stage at the show (during a two-person panel with *Cluetrain* co-author David Weinberger). Another who took the same general position was Michael Powell, who spoke the next day. I recorded some of what Powell said. Here's an excerpt:

First...there is a very shallow current understanding of tech. If you go give a quiz about the seven layers of the Internet—good luck. So, be careful of inviting the legislative process when they have a very *bad* understanding of the technical underpinnings. Because the secondary consequences of their errors can be enormous.

Number two: be careful. You live by the sword, you die by the sword. I'm a big believer in subversion or jujitsu when it comes to this stuff. Which is: let the weight of inertia be on your side. Which [means] I would rather try constantly to position my industry where I succeed if government does nothing, versus positioning it in a way where I need them to do something or I'm dead. Because, if you're in a position where you desperately need to do something or you're dead, start ordering your coffin and digging the plot....

And the other thing is, do you believe in a snapshot of time, you can trust the government—who, by the way, will have a vague definition because they don't get it well enough? If *this* community doesn't understand it, as I heard...I can tell you right now, very few people in Washington do. This means you're going to get a potentially very ambiguous, subject to massive

www.faircom.com/go/?track

Your packages get

**tracked &
delivered**

on time by FedEx.

**FairCom database
technology makes
it possible.**



FairCom®

Other company and product names are registered trademarks or trademarks of their respective owners. © 2006 FairCom Corporation

NEW!

TS-7300 High Security Linux FPGA Computer



\$219 qty 1 **\$189** qty 100

200 MHz CPU

- User-programmable Altera 2C8 Cyclone II FPGA
- PC/104 expansion bus
- Fanless, no heat sink
- 10 RS232 serial ports
- 55 DIO ports
- 2 10/100 Ethernet
- VGA video out
- matrix keypad & text mode LCD
- 2 USB ports
- 2 hot-swappable SDCard sockets
- SD is software RAID1 capable
- Linux, NetBSD
- Real Time extension

Design your solution with one of our engineers

- Over 20 years in business
- Never discontinued a product
- Engineers on Tech Support
- Custom configurations and designs w/ excellent pricing and turn-around time
- Most products stocked and available for next day shipping

See our website for options, peripherals and x86 SBCs

Technologic
SYSTEMS

We use our stuff.

visit our TS-7200 powered website at

www.embeddedARM.com

(480) 837-5200

COLUMNS

LINUX FOR SUITS

variations in interpretation, pile of law.

Now, personalities change, political power changes, congressmen come and go, and presidents change. You're living with something that's a perpetual cycle of interpretation. And, by the way, it's always easier to get a law on the books than ever to get it off again. If it goes on, be prepared that it's there for 15 to 20 years. And the other thing is, government has a way of turning on people. Ask Bill Gates. It may be about networks today, but those same principles can be used against innovative business models and applications in other contexts. And, I submit to you they *would* be. It might have to do with the interoperability of your new product with somebody else's product. It might have to do with servers and caching. Someone will think it's a good idea to apply the same basic principles to the other side of the community.

The biggest form of subversion may be what the Free Software and Open Source movements have been doing from the beginning: producing, rather than just consuming. Powerful things happen when the demand side supplies itself. That's what gave us the GNU tools, Linux, the LAMP stack and the Net itself. And, what thousands of geeks did with code, millions of consumers can do with media.

In the next several years, we'll see the supply side of the media market change radically. The ability to produce top-quality audio has been around for a while. But video will be the big difference-maker. Will premium deals between giant carriers and giant content producers even matter when the biggest source of content for everybody on the Net will be each other?

New businesses and business models will emerge then. These will need to be cage-free—ones that evolve to thrive in the Net's wide-open natural habitat.

For guidance in that direction, here's Lessig again, from the same "Free Culture" speech:

In the container cargo system of moving packets that comprises the Net, contents of containers should be of no interest to those who own and operate the various parts of the system.

Third, and lastly for the moment, I would say, be careful because you're playing *their* game. Regulatory battles are an art form, and these guys are the maestros. It's always a little like Br'er Rabbit. "Oh, woe is me. Throw me on the briar patch and don't regu—oh, regulate me." Uch. The average one of these incumbents, whether a cable or a phone company, have 40 lawyers in Washington dedicated to this work. Resources. Ability. One hundred years of skill. I'm not criticizing, only stating a reality. And, then I meet entrepreneurs who have 12 guys and can't afford a legal cost center to do research. Then, let me add the judicial process. Every decision you get from the Congress and the FCC will spend the next three and a half to four years in court.

The next day, Boucher and Markey's amendment was shot down, on partisan lines, 23–8.

Of course, that was just one fight among many. But it also suggests that our only options are what Powell calls "subversion or jujitsu".

Build a world of transparent creativity—that's your job, this weird exception in the 21st century of an industry devoted to transparent creativity, the free sharing of knowledge. It was not a choice in 1790; it was nature in 1790. You are rebuilding nature. This is what you do. You build a common base that other people can build upon....This is your enterprise. Create like it's 1790. That's your way of being. And you remind the rest of the world of what it was like when creativity and innovation were a process where people added to common knowledge. In this battle between a proprietary structure and a free structure, you show the value of the free.

Two hundred sixteen years later, our job is still putting revolution to work. ■

Resources for this article:
www.linuxjournal.com/article/9012

Doc Searls is Senior Editor of *Linux Journal*.

THE PENGUIN

Another server down. Another night at the office. Whether it's deploying a hundred new servers or installing the latest security patch, it doesn't matter. You're sleeping with the servers again. Penguin Computing® introduces Scyld ClusterWare™ Enterprise. Its centrally-managed, highly available architecture makes large pools of Linux servers act like a single, consistent

virtual system, significantly reducing complexity and time to deploy servers on the fly. Its highly secure environment makes your server farm a 'virtual fortress' and its simplified manageability cuts your maintenance windows and administration by up to 80%, dramatically improving TCO. So go on home, catch some zzzs and know that Penguin is standing guard.

MAY CAUSE DROWSINESS



REAL **LINUX**

PENGUIN HIGH DENSITY CLUSTER. The highest density, modular blade server architecture on the market. With powerful Scyld ClusterWare™ Enterprise for single point command and control, and AMD Dual Core Opteron™ for a highly productive user experience.

PLEASE VISIT US AT LINUX WORLD SAN FRANCISCO,
Moscone Center, August 14-17, 2006. www.penguincomputing.com



Penguin Computing and the Penguin Computing logo are registered trademarks of Penguin Computing Inc. Scyld ClusterWare and the Highly Scyld logo are trademarks of Scyld Computing Corporation. AMD Opteron and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices, Inc. Linux is a registered trademark of Linus Torvalds. ©2006 Penguin Computing, Inc. All rights reserved.

Autodesk's Discreet Inferno

Okay, friends, admit it. Like your humble editor, you've closeted a secret wish to work at Pixar ever since *Toy Story* first came out. Should you ever beat me to that dream job creating animated films or special effects, you'll feel right at home with the omnipresent Linux applications that bring everything to life. A case in point is Autodesk's new Linux port of Discreet Inferno 6.5, an interactive design system for high-resolution visual effects. Discreet Inferno has been used for high-speed compositing and advanced graphics applications in films and television shows, such as *Charlie & the Chocolate Factory*, *Hitchhiker's Guide to the Galaxy*, *CSI New York*, *Stargate: Atlantis* and many others. Interestingly (but not surprisingly), Autodesk first created a Linux version of Discreet Inferno just for Japan, which was such a blockbuster that it was released worldwide. In addition, according to a VP at Autodesk, the Discreet Inferno system on the Linux workstation "can offer up to five times the performance (per CPU) of previous SGI platforms such as the Onyx 2." So, good people of Autodesk, now that Linux has handily rocked your world, there's this little program called AutoCAD....

www.autocad.com/me

Kangaroo Media's Kangaroo.TV

It's official, folks—Linux has gone Joe Sixpack. That's because our beloved OS has infiltrated the world of NASCAR autoracing. Embedded Linux is the horsepower under the hood of Kangaroo.TV, a new device available for rental at NASCAR events that gives spectators a more entertaining autoracing experience. Marketed as NASCAR NEXTEL FanView, this wireless, handheld gadget provides racing fans a slew of event information, including ten live MPEG-4 video feeds (replays, highlights and in-car views), 64 AC3 audio feeds (driver-to-pit conversations and commentary) and a plethora of real-time stats. Kangaroo.TV utilized Trolltech's Qtopia as its application platform and GUI. You can give Kangaroo.TV a whirl for \$50 US a day, coming to a racetrack near you.

www.kangaroo.tv



ippimail's Free E-mail Service

Just say "ippimail"—the name of a new and free e-mail service—five times fast and you'll be hooked too. If the name alone doesn't grab you, the ippimail gang hopes that saving the world will. Here's how it works. ippimail's mission is to get as many people as possible to join its free, Hotmail-esque e-mail service, which utilizes 100% open-source technology. The more eyeballs it gets, the more banner advertising it can support. The firm then donates 45% of its profits to worthy charities and an additional 10% to the Open Source Development community. Though most of the charities are on its home turf in the United Kingdom, many work globally, and ippimail pledges to support charities wherever its users are. In addition, ippimail sees this project as a clever way to evangelize about the benefits of open source to the broader public. Its motto "feelgood email" seems to hit the nail right on the head.

www.ippimail.com





Micro/sys' SBC1491-ET

If your itch involves PC/104 single-board computers that boot quickly and can handle extreme heat and cold, Micro/sys' new SBC1491-ET might be the scratch. This mid-range, full-featured PC/104 (3.55" x 3.775") computer has a sub-five-second boot-up time and will operate in the temperature range of -40°C to +85°C. The SBC1491-ET offers standard PC features such as SVGA, dual serial ports and 10BASE-T Ethernet support, as well as 64MB of RAM, up to 576MB solid-state Flash and full AT compatibility. Micro/sys claims that this product is ideal "for applications not requiring Pentium speed" and offers PC compatibility while generating less heat, using less power and costing less. The processor is an STPC Atlas with operating speeds of 120MHz to 133MHz. Software development packages for the SBC1491-ET include Linux, Windows CE, VxWorks and DOS. Ahoy, Antarctica or bust!

www.embeddedsys.com

Heroix's Longitude

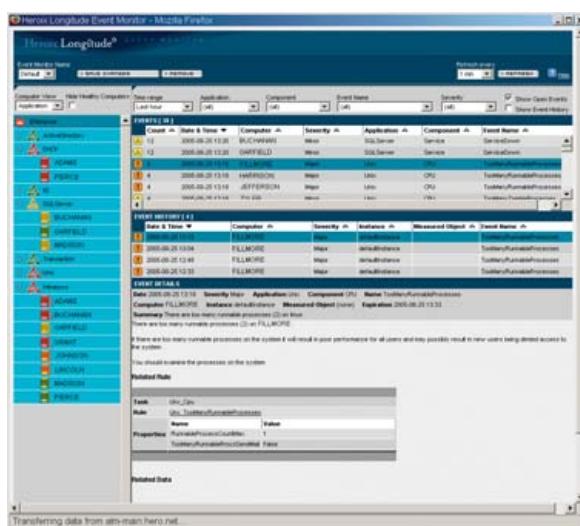
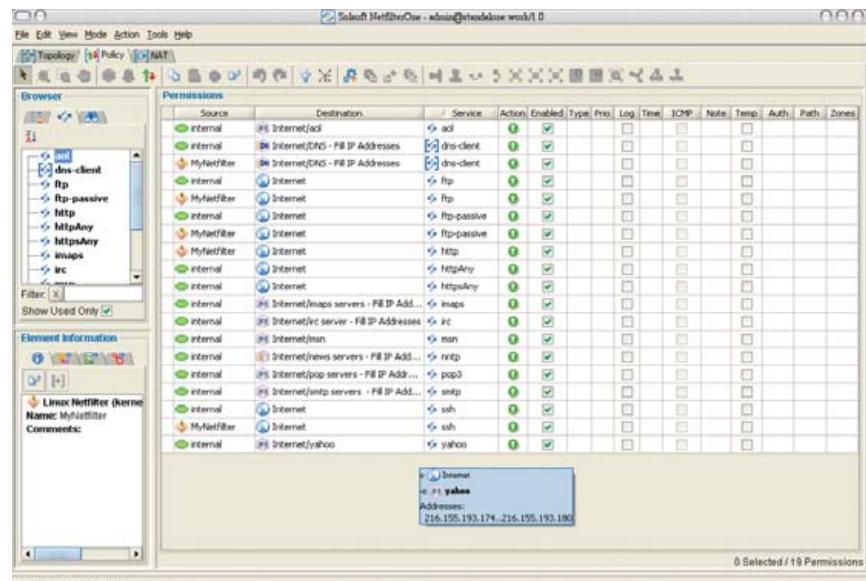
Heroix has upgraded its Longitude agentless performance monitoring and reporting software. The application is intended to monitor an enterprise's entire IT infrastructure, including OS, Web, database, J2EE, messaging, infrastructure, and user and business metrics, all right out of the box and without agent software on monitored computers. Innovations in Version 3 are event correlation across the entire infrastructure, enhanced SLA management via a new graphical event monitor, adjustable workload threshold settings to eliminate false alarms and a range of added monitoring coverage (Cisco devices, Dell OpenManage, HP Systems Insight Manager and SNMP device checking). Platforms monitored are Red Hat and SUSE Linux, Solaris, HP-UX, AIX and Windows. A free, 14-day trial is available at Heroix's Web site.

www.heroix.com

Solsoft's NetFilterOne

Linux Journal's security czar, Mick Bauer, has opined numerous times that a good Netfilter firewall GUI is not just a crutch for the less technical among us, but rather a tool that makes all of us IT administrators work more intelligently. Solsoft recently upgraded its own Netfilter GUI, NetFilterOne. The application is a centralized interface from which one can configure, deploy, enforce and audit rules and policies among Netfilter firewalls. The shiny new Version 1.2 offers the following three key new features: a new Tabular Policy Editor that allows users to manage all permissions in a unified view; bridge mode, also called transparent mode by other providers; and support for Linux Netfilter iptables-1.3.x. A free trial of NetFilterOne is available at Solsoft's Web site.

www.solsoft.com

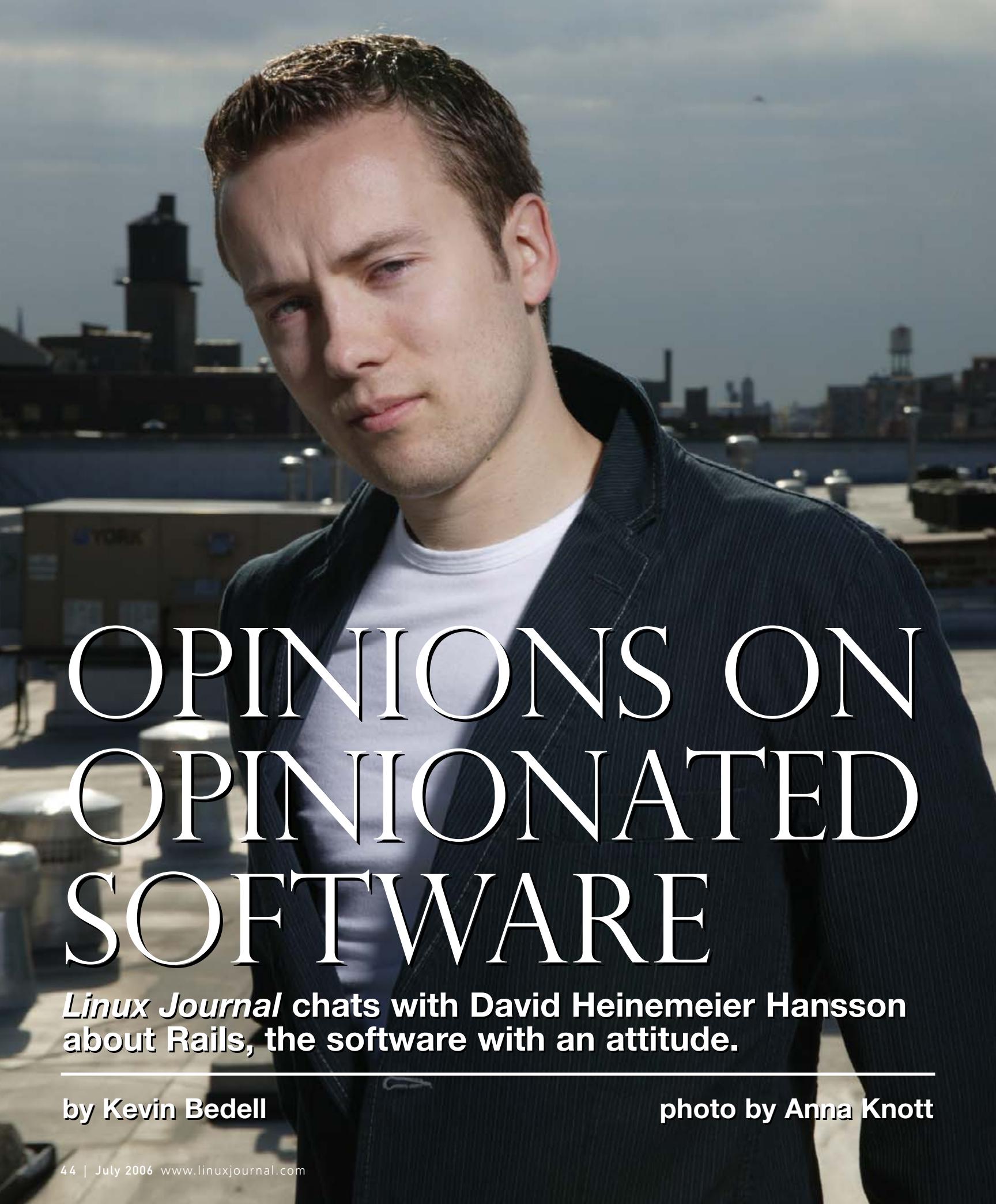


Arkeia Software's Network Backup

If your backup duties include files and directories named with non-Latin characters, you're in luck. Arkeia Software's newly released Network Backup Version 5.5 adds full support for Unicode/UTF-8 characters in the backup index. Unicode supports characters from all the world's major languages and is most commonly applied to languages such as Japanese, Chinese and Arabic.

www.arkieia.com

Please send information about releases of Linux-related products to James Gray at newproducts@ssc.com or New Products c/o *Linux Journal*, 1752 NW Market Street, #200, Seattle, WA 98107. Submissions are edited for length and content.



OPINIONS ON OPINIONATED SOFTWARE

***Linux Journal* chats with David Heinemeier Hansson about Rails, the software with an attitude.**

by Kevin Bedell

photo by Anna Knott

David Heinemeier Hansson is the creator of Rails and a partner at 37signals. He'll soon be moving from Copenhagen to Chicago as he leaves his place of birth to join the company headquarters.

The interview:

LJ: For our readers who are unfamiliar with Ruby and Rails, can you give us a short description of what they are and what makes them different from other development environments?

DH: Ruby is a dynamic and object-oriented programming language created in 1995 by Yukihiro Matsumoto. It has been described as a cross between Smalltalk and Perl, but I don't think that juxtaposition does it justice. Ruby is, more than anything else, a language for writing beautiful code that makes programmers happy.

Rails, then, is an attempt to mold the beauty and productiveness of Ruby into a solution for Web applications. We've sought to adhere to the same core principle that guided the development of Ruby: make the programmer happy!

This might all sound mighty fluffy, but only until you recognize that the single-most important factor in programmer productivity is motivation. And, happy programmers are certainly motivated programmers. Thus, if you optimize for happiness, you're optimizing for motivation, which ultimately leads to an optimization for productivity.

LJ: What is Rails? Why was it developed?

DH: Rails is an extraction from a solution to a real problem. It's not a science project. It's not something clever men sat down and designed in the highest of ivory towers. It's simply the generic pieces that were left after I tried to use Ruby to create Basecamp—the Web-based project management system from 37signals.

That means it's a very pragmatic, very targeted framework with a strong sense of direction. You might not share its vision, but it undeniably has one. I like to call that opinionated software. And Rails sure has a lot of opinions.

From one point of view, it could be said to be the collection of opinions I have about how Web applications should be constructed. Surely you can use Ruby on Rails without sharing *all* my opinions on how to create Web applications, but the more opinions you share, the less work is put upon you.

And, these opinions are surprisingly simple. They aim to give most people most of what they want, most of the time. It's a strong disagreement with the conventional wisdom that everything should be configurable, that the framework should be impartial and objective. In my mind, that's the same as saying that everything should be equally hard.

LJ: I've been reading about Active Record and the ORM (Object-Relational Mapping) capabilities (or how the application interfaces with databases) that are available using Ruby on Rails. Can you comment on this?

DH: Active Record has, by many, been called the crown of Rails. Its core mission is to make relational data mesh seamlessly with an object-oriented

domain model. And to do so with a minimum of explicit configuration.

So, you'll have a Person class that's automatically mapped to a people table (notice the cases and pluralization that Rails automatically figures out). This Person class will then have a first_name method if the people table has a first_name column. So, we're using reflection and conventions to escape the XML situps that plague frameworks of the old world.

Although the lack of explicit configuration scores high points with the Enterprise crowd used to Hibernate, EJBs and other Java frameworks, it's the mere notion of ORM that wins big with the PHP/.NET crowd. Active Record relieves you from the vast majority of all SQL writing. It's automatically constructed on the fly. No more three-line INSERTs, no more repetitive, tedious UPDATEs. The only SQL left is the bottleneck-clearing work where actual brainpower is involved on how to make this query go really fast.

LJ: For many of our readers to adopt Ruby and Rails (or convince their management to let them), they need real success stories. Where has Ruby and Rails been used to build scalable, production applications?

DH: Ruby on Rails has been a huge hit inside a lot of organizations. We have some 400 people signed up as working either partially or completely in a Rails-related job. So, like an iceberg, the bulk of the action happens below the surface.

But, we do have a good number of public success stories too. My own company, 37signals, has four widely popular applications used by hundreds of thousands to manage their projects (Basecamp), their personal life (Backpack), their to-do lists (Ta-Da List) and their collaborative writing (Writeboard). That suite has been the number-one poster child for Ruby on Rails and has helped win over a lot of doubters.

But 37signals is by no means the only small team doing big things with Ruby on Rails. The Robot Co-op has a suite of social networks that includes 43things, 43places and 43people. Together, these networks push more than two and a half million dynamic page views a day across their three-machine setup.

Odeo is running Ruby on Rails to power its podcasting portal in front of thousands of subscribers. Evan Williams created Blogger and knows a thing or two about running a huge, public site. He's at the wheel at Odeo.

Strongspace is just one of several Rails applications in the making from TextDrive. They provide gigabytes of secure hosting in the sky. It's a really cool and smooth site by the same guys that carry the title of being the official Rails hosting firm.

And, that's just a small taste. We have major applications in everything from e-commerce to productivity to content—you name it. There are very few kinds of Web applications left that Rails hasn't been used to create.

LJ: By the way, I've been a Backpack user for a while and I love it. Was it completely developed using Rails?

DH: Backpack is indeed 100% Ruby on Rails. When it launched, it was a mere 2,000 lines of code.

LJ: Java had been around for a while before it really penetrated the enterprise. It took the development of J2EE for it to establish itself as a true "enterprise development platform". The addition of transactional management, flexible deployment strategies and so on seemed required for it to mature into that role. Could you see Ruby and Rails eventually following a similar path, or do you think its role will be something different?

DH: We have a wide enterprise audience that uses Rails simply because it gets the job done, faster. I think we've seen the peak of Java in the enterprise. I'm sensing an understanding that while Java and the J2EE gang certainly has its uses in legacy integration, huge distributed setups that require two-phase commits and so on, it's way overkill for the majority of applications created in the enterprise.

Dave Thomas from the Pragmatic Programmers recently expressed this as "cracking nuts with a sledgehammer". Yes, a few special jobs do need sledgehammers. But you don't need to use it [a sledgehammer] for all the other jobs that need to get done.

That's why having a company standard on something like Java and J2EE seems so nonsensical. Why would you use the heaviest and slowest machinery to solve the 80% of the business that would rather have its valuable software two, three, five or ten times faster? Or, whatever the multiplier is in your environment. So, keep the big guns in store for that last 20% that actually requires it.

LJ: Is there anything else you think is important to tell our readers about Ruby and Rails?

DH: Give it a try! We've fought hard to make Ruby on Rails the easiest Web-application framework to try out. Get Ruby, get RubyGems (the apt-get of Ruby libraries), gem install rails, rails my_application, and you have your application skeleton running and ready to produce.

It's hard to relay in words just how fast and easy it is to get started. So, I would invite your readers to check out the 15-minute video on the Rails Web site where we build a complete, if simple, blogging engine.■

Resources for this article:
www.linuxjournal.com/article/9015.

Introduction to Ruby

**EVERYTHING YOU NEED TO KNOW
TO START PROGRAMMING IN RUBY.**

REUVEN M. LERNER

We programmers are lucky to be working today. I say this because there are so many excellent programming languages from which to choose, especially in the Open Source world.

One of the most talked-about languages is Ruby. Ruby isn't actually all that new. Yukihiro "Matz" Matsumoto released the first public version in 1995, and it has grown in popularity ever since. As the Ruby on Rails framework for Web development has become increasingly popular, interest in Ruby has soared along with it.

Ruby often has been described as a cross

between Perl and Smalltalk, and I don't think this is a bad way to look at it. Certainly, if you have experience with both Perl and object-oriented programming, you probably will feel right at home working with Ruby.

In this article, I introduce the basics of Ruby, showing how it is similar to other high-level languages and where it adds its own, special twist. By the end of this article, I hope you'll know enough about Ruby to try it out for some small applications. If you're like me, you'll quickly discover that Ruby is surprisingly compact and elegant, making it possible to write maintainable code quickly and easily.

The Basics

Downloading and installing Ruby is fairly easy, particularly because a recent version (1.8.2) is included with many distributions of Linux. You either can use that version or install the latest version (1.8.4) from the main Ruby site. As an open-source product, you shouldn't be surprised to find that the main Ruby site (www.ruby-lang.org) offers the source code in .tar.gz format. Additional formats, such as RPMs and Debs, are available from the official repositories for your favorite distribution.

If you want to install the latest version of Ruby from source, download and unpack the .tar.gz file:

```
$ cd Downloads  
$ tar -zxf ruby-1.8.4.tar.gz
```

Now use the standard configure program to find the system configuration automatically, make to compile it and then make test to ensure that the compiled version of Ruby works correctly:

```
$ ./configure && make && make test
```

If all goes well, the final line of output from the above commands will read `test succeeded`. Now you can become the root user and install Ruby onto your system:

```
$ su  
# make install
```

This installs a variety of Ruby programs and libraries onto your computer.

Interactive Ruby: Irb

The Ruby language itself exists as an executable called `ruby`, which you can run manually by typing it on the command line:

```
$ ruby
```

However, this version of Ruby is designed for non-interactive use. To test code or experiment with the Ruby language, there is `irb`, the interactive Ruby shell. `irb` is something like a debugger, in that it takes input from a user (terminated by pressing the Enter key) and executes it. For example, type:

```
$ irb
```

And, `irb` responds with its prompt:

```
irb(main):001:0>
```

Now we can type a bit of Ruby:

```
irb(main):001:0> print "Hello, world"
```

And, `irb` responds with:

```
Hello, world=> nil
```

The above output indicates that `print` displays `Hello, world` on the screen and returns a `nil` value; `nil` is Ruby's way of representing a null value, much like `undef` in Perl, `None` in Python and `NULL` in SQL.

Like many other high-level languages, Ruby allows us to assign values to variables without pre-declaring them. Thus, we can write:

```
greeting = "Hello, world"  
print greeting
```

Ruby also can do math, using the familiar operators `+`, `-`, `*` and `/`:

```
5 + 3  
60 - 23  
60 * 23  
10 / 2
```

I have omitted the call to `print` in the above lines, because it's unnecessary in `irb`. However, in a standalone Ruby program, no output would be sent to the screen (or elsewhere) without using `print`.

If you are a seasoned Perl programmer, you might be somewhat surprised to discover the result of the following:

```
5 / 2
```

The above returns 2 because both 5 and 2 are integers, and Ruby assumes you want to perform integer arithmetic. To get a floating-point result, you must ensure that at least one of the numbers is a float:

```
5 / 2.0
```

Sure enough, that returns 2.5. Unlike many other languages, Ruby requires a leading 0 for fractional numbers; you must say `0.5`, rather than `.5`.

You can convert a string to an integer or float using the `to_i` and `to_s` methods:

```
"5".to_i  
"5".to_f
```

All objects in Ruby have a similar `to_s` method, which turns the object into a string.

One datatype in Ruby that surprises some newcomers is the symbol. You can think of symbols as special kinds of strings that take up far less room in memory, especially when they are used in multiple locations. Symbols, which begin with a colon (for example, `:reader`) cannot always be used in place of strings, but they allow programmers to make programs more readable. They also are used on occasion to refer to objects and methods, as I explain later in this article.

Interpolation and Methods

Like many other high-level languages, Ruby lets us interpolate values inside of double-quoted strings. (Single-quoted strings are taken literally, as is the convention in many other languages.) For example:

```
name = "Reuven"  
"Hello, #{name}"
```

The above expression is equivalent to:

```
Hello, Reuven
```

Within the `#{ }`, we can put any Ruby expression, not only a variable name:

```
name = "Reuven"  
print "Hello, #{name}. Your name is #{name.length} letters long."  
print "Backwards, your name is '#{name.reverse}'."  
print "Capitalized, your backwards name is '#{name.reverse.capitalize}'."
```

As you can see, interpolation lets us put arbitrarily complex expressions within a double-quoted string. But wait a second—what are we doing with the expressions `name.length`, `name.reverse` and `name.reverse.capitalize`?

The answer is that strings, like everything in Ruby, are objects. Nearly anything we will do with a string is expressed as a method, rather than as a standalone function. If you want to reverse a string, get its length, capitalize it or break it apart, you will invoke a method

on the object using Ruby's `object.message` syntax. For example:

```
name.reverse
```

The above code returns a new string object, whose value is the reverse of `name`. `Name` itself is not altered in the process. Because this new returned object is also a string, we can invoke any string method on it, including `capitalize`, as we saw before. Ruby programmers often end up chaining methods together to accomplish a task.

Methods invoked on an instance of an object are often referred to as `Object#method` in Ruby documentation. So, the above method would be referred to as `String#reverse`.

How do we know to which methods a particular object will respond? One way is to ask the object what class it is:

```
name.class
```

We also can ask an object whether it is a member of a particular class:

```
name.is_a?(String)
```

This might look a bit strange, both because of the question mark in the method name and the parameter that comes after it. But it works just like the other methods we have invoked so far. We send an `is_a?` message to `name`, which returns a Boolean (true or false) response. The argument to `is_a?` is a class name, which is `String`.

If we would prefer not to look up the API documentation for Ruby strings, we simply can ask the object itself what methods it will respond to:

```
name.methods
```

This returns an array (that is, a list) of methods to which `name` responds. We will look at arrays in a moment, but it's important to realize that `name.methods` is not a string; rather, it's an array whose contents happen to be strings. However, arrays respond to a built-in sort method, which returns a new array whose contents are ordered:

```
name.methods.sort
```

I probably invoke `OBJECT.methods.sort` at least once each day, rather than look through a book or on-line API for Ruby.

Arrays and Hashes

If you have worked with Perl or Python in the past, you won't be surprised to learn that Ruby has built-in arrays (as mentioned above) and hashes. We create an array with square brackets:

```
an_array = [1, "two", true]
```

An array can contain any number of objects, and each object can be of any type, including another array. The above array contains three objects (of types `Fixnum`, `String` and `Boolean`, respectively). Each item in an array has a unique index; the first element has an index of 0. We can retrieve items as follows:

```
an_array[1]
```

The above expression returns "TWO", the item with an index of 1 in `an_array`. Arrays are mutable, meaning that we can replace any of the items by assigning to that index:

```
an_array[1] = "TWO"
```

We can use a negative index to count from the back of the array; thus `an_array[-1]` returns the Boolean value `true`. We also can view a subset of

the original array by passing two indexes separated by a comma, indicating the first and last index that we want:

```
an_array[0,1]
```

To combine all of the elements of an array into a string, we can use the `join` method, for example:

```
an_array.join(", ")
```

The above code creates a single string, whose contents are the values from `an_array`, with ", " between each pair of elements.

Hashes are similar to arrays, except that instead of storing values using an ordered, numeric index, they are stored with keys, for example:

```
my_hash = {'a' => 1, 'b' => 2}
```

We can now retrieve either of the two values, by using its key:

```
my_hash['a']
my_hash['b']
```

The above lines of code return the numbers 1 and 2, respectively. As with arrays, we can store any object as a value in a hash; it doesn't have to be an integer.

We can retrieve the keys and values of a hash with the `Hash#keys` and `Hash#values` methods, respectively. (Later, I explain how to iterate over the keys and values to retrieve contents from a hash.) Sometimes, however, we simply want to know if a particular key exists in a hash. This is easily accomplished with `Hash#has_key?`, which takes a string as a parameter and returns a Boolean value. The following code thus would return true:

```
my_hash.has_key?("a")
```

Conditionals

Every language lets us execute code conditionally. In Ruby, this normally is done with an `if` statement. Consider the following (somewhat contrived) example:

```
if server_status == 0
  print "Server is in single-user mode"
elsif server_status == 1
  print "Server is being fixed"
elsif network_response == 3
  print "Server is available"
else
  print "Network response was unexpected value '#{network_response}'"
end
```

Notice that Ruby does not require parentheses around the condition. And although the condition does not have to return a Boolean value, Ruby will produce a warning if you try to use `=` (that is, assignment) in the condition, rather than `==` (that is, comparison). The `==` comparison operator works on all objects; there are no separate text comparison and numeric comparison operators as in Perl. This is true for `<` and `>` also, which can be used to compare strings as well as numbers. Finally, Ruby does not use opening or closing braces; instead, it closes the conditionally executed block of code with `end`.

As with Perl, you can use `if` and `unless` as suffixes to make a statement conditional:

```
print "We won!" if our_score > their_score
print "Here is your change of #{amount_paid - price}!"
unless amount_paid <= price
```

SUPERMICRO®

Xeon 5000/5100 Server Solutions

• 64GB FBD Memory • Multiple-Core
• High Efficiency Power



1U SuperServer 6015B-8+

- ✓ Dual Intel® Dual-core Xeon™ 5000/5100 sequence
- ✓ SUPER●®X7DBR-8+ serverboard
- ✓ Intel 5000P Chipset/1333MHz FSB
- ✓ Up to 64GB fully-buffered DIMM (FBD)
- ✓ 1 PCI-X 133MHz or 1 PCI-E x16 or 1 PCI-E x8
- ✓ Adaptec dual-channel U320 SCSI
- ✓ AOC-LPZCR2 (Zero-channel RAID) support
- ✓ ATI ES1000 16MB PCI graphics controller
- ✓ Dual Gigabit LAN ports
- ✓ 4 x 3.5" hot-swap SCA drive bays w/SAF-TE
- ✓ 1 slim floppy & 1 slim DVD-ROM drive
- ✓ 700W cold-swap power supply w/I²C
- ✓ 4 x 4cm fans w/optimal fan speed control
- ✓ IPMI 2.0 with KVM-Over-LAN support

2U SuperServer 6025B-8R+

- ✓ Dual Intel® Dual-core Xeon™ 5000/5100 sequence
- ✓ SUPER●®X7DB8+ serverboard
- ✓ Intel 5000P Chipset/1333MHz FSB
- ✓ Up to 64GB fully-buffered DIMM (FBD)
- ✓ 3 PCI-X 133/100MHz, 2 PCI-E x8 & 1 PCI-E x4
- ✓ Adaptec dual-channel U320 SCSI
- ✓ AOC-LPZCR2 (Zero-channel RAID) support
- ✓ ATI ES1000 16MB PCI graphics controller
- ✓ Dual Gigabit LAN ports
- ✓ 8 x 3.5" hot-swap SCA drive bays w/SAF-TE
- ✓ 1 slim DVD-ROM drive, optional USB 2.0 ports & FDD
- ✓ 700W cold-swap power supply w/I²C
- ✓ 3 x 8cm hot-swap cooling fans
- ✓ IPMI 2.0 with KVM-Over-LAN support

4U SuperServer 7045B-3

- ✓ Dual Intel® Dual-core Xeon™ 5000/5100 sequence
- ✓ SUPER●® X7DB3 serverboard
- ✓ Intel 5000P Chipset/1333MHz
- ✓ Up to 32GB fully-buffered DIMM (FBD)
- ✓ 3 PCI-X 133/100MHz, 2 PCI-E x8 & 1 PCI-E x4
- ✓ Adaptec 9410 controller for 8 SAS/SATA ports
- ✓ AOC-LPZCR2 (Zero-channel RAID) support
- ✓ ATI ES1000 16MB PCI graphics controller
- ✓ Dual Gigabit LAN ports
- ✓ 8 x 3.5" hot-swap SAS drive bays w/SES2
- ✓ 90° rotatable module: 2 USB 2.0 ports
- FDD, 2 x 5.25" drive bays
- ✓ 100% cooling redundancy: 6 fans & air shroud
- ✓ 1 slim DVD-ROM drive, optional USB 2.0 ports & FDD
- ✓ 650W redundant cooling fan w/I²C
- ✓ IPMI 2.0 with KVM-Over-LAN support

AMAX Corp.
1-800-800-6328
www.amax.com

Arrow Electronics
1-888-427-2250
www.arrownacp.com

ASI
1-800-2000-ASI
www.asipartner.com

Bell Micro.
1-800-232-9920
www.bellmicro.com

Ingram Micro
1-800-456-8000
www.ingrammicro.com

MA LABS
1-408-941-0808
www.malabs.com

Synnex Inc.
1-800-756-5974
www.synnex.com

Tech Data
1-800-237-8931
www.techdata.com

You also can do things like:

```
if inputs.length < 4
  print "Not enough inputs!\n"
end
```

And, also:

```
if not my_hash.has_key?("debug")
  print "Debugging is inactive.\n"
end
```

Loops

Ruby does have some looping operators, such as for and while. But the real fun and excitement is in doing things such as this:

```
5.times {print "hello\n"}
```

Think about it—we're invoking a method on a number, using the standard Ruby method-invocation syntax. The times method for integers executes a block of code a particular number of times. So, the above line of code executes five times, printing the word hello (followed by a new line) each time.

Blocks can take parameters as well, between pipe ()| characters:

```
5.times {|iteration| print "Hello, iteration number#{iteration}.\n"}
```

We similarly can iterate over the elements of an array with the each method:

```
an_array = ['Reuven', 'Shira', 'Atara', 'Shikma', 'Amotz']
an_array.each {|name| print "#{name}\n"}
```

A variation of the each method, called each_with_index, requires a block that takes two parameters. The first parameter is the item, and the second is the index:

```
an_array = ['Reuven', 'Shira', 'Atara', 'Shikma', 'Amotz']
an_array.each_with_index {|name, index| print "#{index}: #{name}\n"}
```

At a certain point, blocks become difficult to read in this syntax. Ruby provides an alternate syntax, replacing the curly braces with do and end:

```
an_array = ['Reuven', 'Shira', 'Atara', 'Shikma', 'Amotz']
an_array.each_with_index do |name, index|
  print "#{index}: #{name}\n"
end
```

We can iterate over a hash in several ways. One way is to use the type of iteration that Perl and Python programmers have used for years, getting the hash's keys (via Hash#keys, which returns an array) and then grabbing the value that goes with the key:

```
state_codes = {'Illinois' => 'IL', 'New York' => 'NY',
               'New Jersey' => 'NJ', 'Massachusetts' => 'MA',
               'California' => 'CA')

state_codes.keys.each do |state|
  print "State code for #{state} is #{state_codes[state]}.\\n"
end
```

Of course, we might want to sort the keys before iterating over them:

```
state_codes.keys.sort.each do |state|
  print "State code for #{state} is #{state_codes[state]}.\\n"
end
```

Ruby provides an easier way to perform this task, the each_pair method:

```
state_codes.each_pair do |state, code|
  print "State code for #{state} is #{code}.\\n"
end
```

Classes and Methods

Finally, we can put this all together in defining a class and some methods. We can create a class in irb, or anywhere else in Ruby, simply by saying:

```
class Simple
end
```

Sure enough, we've managed to create a class in only two lines. Is this enough to create an object of type Simple? Let's see:

```
foo = Simple.new
foo.class
```

It would seem so; our variable foo claims that it is of class Simple. We didn't specify what object Simple inherits from, so it automatically inherits from Object, the ultimate Ruby superclass. Ruby supports only single inheritance, which is stated in the class definition as:

```
class SimpleArray < Array
end
```

We already have defined two classes, which is nice, but we haven't defined any methods specific to those classes. Ruby allows us to open up a class at any time, adding or replacing methods in a class. We define a method with the def statement, indicating whether the method takes any parameters, for example:

```
class Simple
  def id_squared
    return self.object_id * self.object_id
  end
end
```

The method we have defined is quite simple, and it does something that I don't expect we would ever want to do—namely, it takes the object's unique ID (available via the inherited method object_id) and returns its doubled value (which will likely be an instance of Bignum).

If we type the above definition into irb, something amazing happens: our foo variable of class Simple now responds to the method Simple#id_squared! Yes, Ruby allows us to modify methods on the fly and to open up existing classes. We could, for example, modify the built-in Array or String classes, replacing the built-in methods with some of our own.

Finally, we might want to store some state in our object. This is done via instance variables. In Ruby, instance variables are preceded with the @ character, which might be a bit confusing if you are coming from the Perl world:

```
class Simple
  def initialize
    @simple_data = [ ]
  end
end
```

The special initialize method is invoked whenever we create a new instance of Simple. So if we once again define foo to be an instance of Simple:

```
foo = Simple.new
```

we can see that foo now has an instance variable defined, by invoking:

```
foo.instance_variables
```

The above returns an array:

```
["@simple_data"]
```

How can we assign to @simple_data? And how can we retrieve its value? One way is to define a number of methods: one for writing this instance variable and one for retrieving its value. But a shorthand way would be to use the attr_reader and attr_writer methods:

```
class Simple
  attr_reader :simple_data
  attr_writer :simple_data
end
```

The above code tells Ruby we have an instance variable named @simple_data, and that we would like to have methods created that will allow us to read and set its value. You can see here how symbols allow us to refer to an instance variable by something that is not a string, but not the literal variable either. With this in place, we can do things like:

```
foo = Simple.new
foo.simple_data = 'abc'
foo.simple_data = [1, 2, 3]
print foo.simple_data.join(' ', ')
```

Conclusion

Ruby has become extremely popular in the last year or two, in no small part because of the growth of Ruby on Rails among Web developers. Even without Rails though, Ruby deserves much of the attention it has received. The fact that all data is stored in objects, the compactness and elegance of the method and block structures, and the very large number of objects included in the standard library all make for an impressive language.

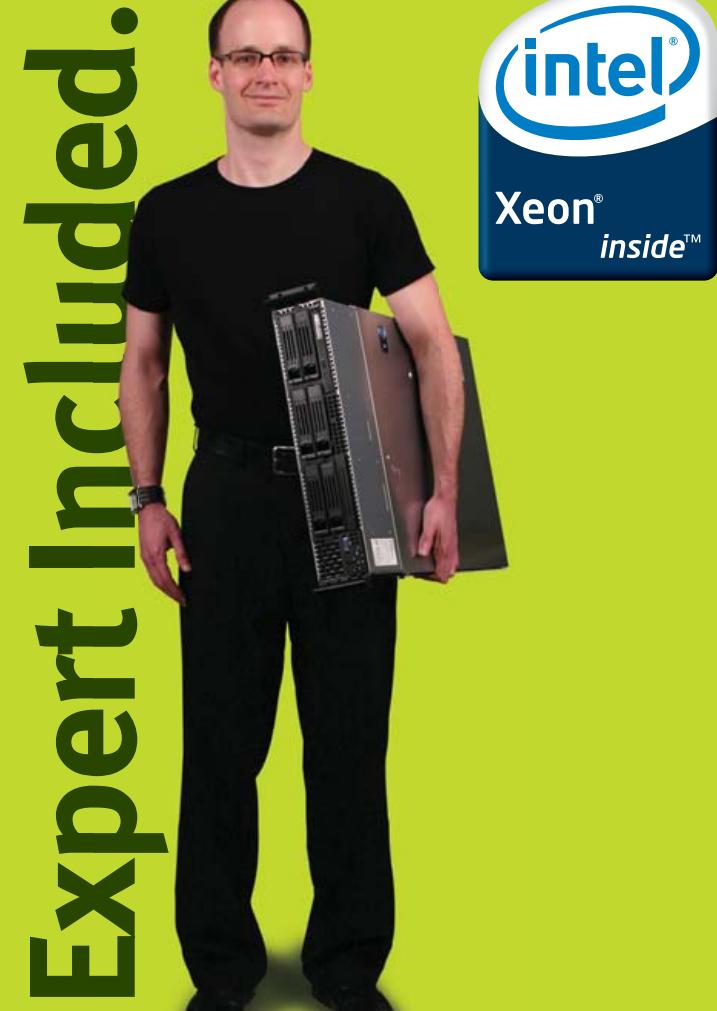
This article didn't have space to go into some additional features that will be of interest to many Ruby programmers, such as modules, class variables, input/output with files, networking, XML parsing, the RubyGems library available on the Internet and built-in support for regular expressions. Ruby is a rich language, but it is fairly consistent and easy to learn—assuming you already have some background with object-oriented programming, which I think is the greatest hurdle to understanding Ruby.

Ruby still has a number of issues to resolve, including its relatively slow speed and a lack of Unicode support, but these are being addressed for future versions, and the community is one of the strongest that I've seen.

I have been using Ruby more and more during the last year and have grown to be quite impressed with the language. I suggest that you give Ruby a whirl as well. Even if you don't make it your primary programming language, it will get you thinking in new ways, and it might make programming in other languages more enjoyable too. ■

Resources for this article: www.linuxjournal.com/article/9017.

Reuven M. Lerner, a longtime Web/database consultant, is currently a PhD student in Learning Sciences at Northwestern University in Evanston, Illinois. He and his wife recently celebrated the birth of their son Amotz David.



Our experts draw on years of experience to provide manageable, reliable products. Dave likes the new Rackform iServ R242 with Dual-Core Intel® Xeon® processors because it offers up to 2x the performance at less than 100W per processor. He knows that IT organizations can count on Intel® Virtualization Technology to streamline their infrastructure, optimize server utilization, and reduce total costs.

The Rackform iServ R242 features:

- Dual-Core Intel® Xeon® processors
- 8 FBDIMMs with 32GB of memory capacity
- IPMI 2.0 support with optional KVM-over-IP
- 6 hot-swap SAS or SATA 3.5" drive bays
- Optional redundant power
- Optional Local Control Panel

When you partner with Silicon Mechanics you get more than great Intel performance – you get an expert like Dave.



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



RubyGems

A comprehensive guide to finding, creating and using Ruby resources called gems. **Dirk Elmendorf**

RubyGems is a system for managing Ruby software libraries. Ruby code packaged in this manner is called a gem. When you find Ruby software you want to use in a project, gems offer a means of downloading, installing and managing the software.

History

Ruby's connection with Perl caused converts to ask an obvious question "Where is the CPAN (Comprehensive Perl Archive Network) for Ruby?" If you have done any Perl programming or used Perl software, you likely have downloaded something from CPAN to make the software work. As it is the de facto standard for sharing libraries in Perl, access to CPAN makes it easier to re-use code developed by others. This tool allows the developer to focus on new problems instead of re-inventing the wheel.

As it turns out, package management is not as simple as it sounds. It gets even more complicated when you are trying to solve a problem for a variety of platforms and operating systems (Ruby runs on a *nix/*BSD/Mac OS X/WinX). There have been several attempts at building a working system.

Ryan Leavengood is credited with creating the very first RubyGems project back in 2001 (see the on-line Resources). The project got started but did not really gain enough momentum to take off. Other solutions were attempted, but they did not really catch on enough to dominate the field.

In November 2003, Rich Kilmer, Chad Fowler, David Black, Paul Brannan and Jim Weirch got together at a Ruby conference and started coding. Their goal was to create a solution once and for all. They obtained permission to use the existing name RubyGems from Leavengood, even though they did not use any code from the previous project.

RubyGems set out to solve several problems. The focus was on simplifying the process of installing, removing, updating and managing Ruby libraries. The developers added an interesting twist by allowing the system to manage multiple versions of the same library easily. Using the versioning

scheme from RubyGems, it is possible to provide very powerful control over the selection of which version of a library your code will actually use.

Getting Started

There are plans to include RubyGems as part of the core distribution of Ruby, but until that happens, you need to install it. Your Linux distribution may have a package (RPM, Deb and so on) for RubyGems. In the event that it does not, you can install it from source easily, assuming you have Ruby and the development headers for Ruby already installed on your Linux box.

You can do the following as a user: go to rubyforge.org/projects/rubygems, and download the current version (0.8.11 at the time of this writing):

```
tar xzf rubygems-0.8.11.tgz  
cd rubygems-0.8.11
```

You must be root to install the software (assuming you want it to be available to all users):

```
ruby setup.rb all
```

Now that RubyGems is installed, you should have the gem command (gem is the command used to interact with the RubyGems package system). Test it out by running:

```
gem list
```

It should show a single package—sources (0.0.1) installed. The gem command is the way you interact with the RubyGems package system.

User Tasks

Now that you have the gem command, you can begin installing gem packages. You need to be root to install or modify gems, but any user can query the system to find out what is installed. When you want to find software, you can always check out RubyForge (see Resources). It is the main clearinghouse for Ruby open-source software.

One of the most popular RubyForge projects is Ruby on Rails. The Rails gem (and the gems it depends on) can be installed with the following command:

```
gem install rails --include-dependencies
```

Another very popular project is RMagick. RMagick is a useful Ruby interface to ImageMagick (see Resources), and it can be installed with the following command:

```
gem install rmagick
```

This gem includes non-Ruby code. When you install it, it will compile the C code as part of the installation process. If you do not have compile tools installed, the installation will fail.

RubyGems features a number of useful features, including:

```
gem search rails --remote gems.rubyforge.org
```

This returns a list of all the packages and versions available on RubyForge that have the word rails in the title of the package. Here are a few more, well, er, gems:

- `gem update`: updates all the current versions of gems to their latest version.
- `gem cleanup`: removes old versions of gems that are installed.
- `gem uninstall`: removes a given gem from the repository.

Because I try to keep up with the most current version of the gem software, I usually `gem update` and then `gem cleanup` the repository to get rid of old libraries. Doing this keeps the gems directory a little cleaner and makes it easier to sort through if and when you need to look in the directory.

Developers

Now that you have some software installed, you will want to use it. To get started, you may want to read the documentation on the gems to learn their API. If you have installed rdoc on your system, gem automatically generates the rdoc (Ruby documentation) for all of the gems you install. You can view this documentation in two different ways. The first one is to run the command:

```
gem_server
```

This automatically launches a Ruby-based Web server on port 8808. You can add the -p option to launch the server on a different port. This makes it easy for you to use your Web browser to browse the documentation for all of the gems that are installed. The `gem_server` can be stopped by pressing Ctrl-C. Also, be aware that the server accepts connections from all hosts that are able to connect to that port. So, if you are concerned about opening a port on your server, you may want to try the alternate means of access.

The other way to access this documentation is to navigate to the place on the filesystem where gem has generated it. In most cases, it will be in `/usr/lib/ruby/gems/1.8/doc`, but in the event that gem has been installed in a different path, you can ask gem where the correct directory is:

```
gem environment gemdir
```



ASA COMPUTERS

Want your business to be more productive?

The ASA Servers powered by the Dual Core Intel® Xeon™ Processor provides the quality and dependability to keep up with your growing business.

Hardware Systems For The Open Source Community—Since 1989

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

ASA Low Budget Low Voltage-Starting @ \$1025

1 of 2 Intel Xeon LV 1.67GHZ Dual Core
1 GB DDR2 PC2-3200 ECC REG
1 of 160 GB SATA Drive
Dual Gigabit LAN, IPMI Card, CD



ASA 14" Deep Appliance Server-Starting @ \$995

Intel Xeon LV 1.67GHZ Dual Core
1 GB DDR2 PC2-3200 ECC REG
40GB SATA Drive, One GigE
CD, FD, Second HD, Your Logo



ASA 1U 4 Drives SCSI Server- Starting @ \$1491

1 of 2 Intel Xeon LV 1.67GHZ Dual Core
1 GB DDR2 PC2-3200
1 of 4 36GB SCSI Drive
Dual Gigabit LAN, IPMI Card, CD



ASA 6 Hot Swap SAS/SATA/Dual Xeon Dual Core

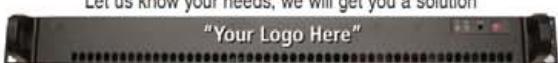
2 Intel Xeon 2.0GHZ Dual Core
Supports up to 16GB DDR2 ECC REG
Supports SAS/SATA
CD, FD, IPMI Card, 2 Gigabit Ethernet
Call for Pricing & Customization



Your Custom Appliance Solution

Let us know your needs, we will get you a solution

"Your Logo Here"



ASA Collocation

\$75 per month for 1U Rack - 325 GB/month

ASA Collocation Special

First month of collocation free.*

Storage Solutions

IDE, SCSI, Fiber Raid solutions
NAS, DAS, iSCSI, SATA, SAS
3Ware, Promise, Adaptec
JMR, Kingston/Storcase solutions

Clusters

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

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



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



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

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

This command gives you the base directory where gem is installed. The documentation is stored in the doc subdirectory of that directory. When you access the files this way, you do not get the summary overview that you get from the gem_server; instead you get only a directory listing of all the gems that are installed.

In order to make your Ruby scripts able to use the Ruby libraries you have now installed, you need to use Ruby's require mechanism to load in the code. The simplest way to use RubyGems is to call the following lines:

```
require 'rubygems'
require 'RMagick'
```

This loads all the RubyGems code and automatically allows you to use the latest gem version of RMagick that you have installed. If the code is available locally, it will be included from there instead.

If you would like to tie your software to a specific version of the library, a different function must be called:

```
require 'rubygems'
require_gem 'RMagick', '>=1.10'
require_gem 'rake', '>=0.7.0', '<0.9.0'
```

These statements tell Ruby to use RMagick as long as it is greater than or equal to 1.10. The second line allows any version of rake as long as it is greater than or equal to 0.7 and less than 0.9.0. The version statement supports a number of operators: =, !=, >, >=, <, <= and ~>. The last one is a special operator. It assumes that you are following a RubyGems standard for versioning.

X.Y.Z

You increase X when you release a version that is incompatible with the previous version. You increase Y when you release a version with a new feature that is otherwise compatible. You increase Z when you release a fix for the software.

This allows the ~> requirement to select within a special range. So for example: 1.0, 1.0.1, 1.0.2, 1.1 are all ~> 1.0, and 1.1, 1.1.2 are ~> 1.1.

This lets you support minor changes in the gem version without having to change the require statements in your code.

A word of advice: if you are putting in require statements that are tied to a version, make sure you have a central place for calling and organizing them. This will make it easier to determine what other software you depend on and to adjust version requirements later when they need to change.

Building Your Own

So far, gems have really been about using other people's software in your code. If you decide you have a library that might be useful to other people, you easily can package it up as a gem.

Now that you know how to use gems, you might want to know how to build them. The process of turning your code into a gem is a two-part process. The nice thing about that process is you do not have to modify your code to make it available as a gem. The first part is getting your library set up in a directory structure that is suitable for conversion to a gem. I'm going to be using an existing project called IPAdmin (see Resources) as my example of how this works.

The directory structure is organized as follows:

- /ipadmin/lib: this directory contains all of the Ruby code related to the project.
- /ipadmin/pkg: this is where the gem will be generated.
- /ipadmin/tests: this is where any unit or other tests should be stored.
- /ipadmin/README: this file should contain a summary of the project—especially the license under which it is being released (feel free to add a separate file for the license).

This is the bare minimum layout you need to build up a gem.

More complex projects (rake for example) add the following directories:

- /rake/bin: this is for any command-line scripts that are part of the project.
- /rake/doc: additional documentation about the project.

This shows how some projects (rake, capistrano) are able to add in new command-line tools once they are installed on a system.

RMagick includes a special directory:

- /RMagick/ext: this is where non-Ruby source should be stored if it is going to be compiled.

This is another power option. RubyGems supports shipping non-Ruby source code in the gem. When the user installs this "source" gem on the destination computer, gem attempts to compile the extra code as part of the installation. The advantage of shipping a gem this way is that the non-Ruby code will bind to the actual libraries that are installed on the destination computer. This is exactly what happens when you install RMagick. If you do not have the proper libraries (ImageMagick) or a compiler, the install will fail. To get around the problem of not being able to compile the code, it is possible to ship a precompiled version of the gem. In this case, the source files are compiled and then simply included in the gem.

Once you have your code set up in the correct directory structure, you can focus on the other part of the process of gem building—the gem specification. This is basically a manifest that gives gem all the information it needs about the gem being built. You can build a gem spec as a standalone file, but it is easier to work with if you make it a Rakefile. This simplifies the building process.

There is a Rakefile in the main directory of IPAdmin:

```
require 'rubygems'
Gem::manage_gems
require 'rake/gempackagetask'

spec = Gem::Specification.new do |s|
  s.platform = Gem::Platform::RUBY
  s.name      = "ipadmin"
  s.version   = "0.2.2"
  s.author    = "Dustin Spinhirne"
  s.email     = "dspinhir @nosspam@yahoo.com"
  s.summary   = "A package for manipulating IPv4/IPv6 address space."
  s.files     = fileList['lib/*.rb', 'test/*'].to_a
  s.require_path = "lib"
  s.autorequire = "ip_admin"
  s.test_files = Dir.glob('tests/*.rb')
  s.has_rdoc = true
  s.extra_rdoc_files = ["README"]
end

Rake::GemPackageTask.new(spec) do |pkg|
  pkg.need_tar = true
end

task :default => "pkg/#{spec.name}-#{spec.version}.gem" do
  puts "generated latest version"
end
```

This is a good example of a standard Rakefile for a gem. Here you can see that it is including RubyGems and adding some tasks from rake. The main spec handles providing all the information about the gem that is being built. The last task adds a simple helper that allows you to run rake in the directory and automatically build a gem.

Each of the lines in the spec has a special meaning. The entire list of options that can be set is available from the Gemspec Reference on the RubyGems Manuals site (see Resources).

PGI® Compilers are building the 64-bit applications infrastructure.

PGI Fortran, C and C++ compilers deliver world-class performance on a wide spectrum of 64-bit scientific and engineering applications. With PGI you get an easy-to-use integrated suite of dual-core and MPI-capable compilers, debugger, and profiler to simplify porting and tuning of 64-bit applications for AMD64 and EM64T processor-based workstations, servers and clusters. With comprehensive cross-platform support for Linux and 64-bit Windows operating systems on both Intel and AMD processors, PGI delivers a uniform development environment across your key target systems. The leading independent software vendors in structural analysis, computational chemistry, computational fluid dynamics, and automotive crash testing have chosen PGI compilers and tools to build and optimize their 64-bit applications.

Visit www.pgroup.com to learn what PGI Compilers and Tools can do for you.



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

Specification Explained

platform determines for what platform the gem is meant. If you are just using pure Ruby, it can stay with this default. This flag becomes very important when you are shipping precompiled gems.

name, version, author, email and summary provide basic information about the gem and its author. This is how users can find out who is responsible for the code.

files defines the list of files that are to be included in the gem. The FileList command is provided by rake, which does two things that make life easier. First, it handles globs (*) and patterns meaning that you can grab a lot of files easily. It also understands that certain files should be excluded. By default, it excludes CVS, svn, bak and core files.

require_path is set to determine what directories should be searched for code. The value for this would change if you were building extensions in the ext.

autorequire designates which file will be loaded when require_ipadmin is called in code. ipadmin.rb in this module handles requiring the other three libraries that ship with ipadmin.

test_files is a list of files that should be executed when the gem is installed if the user adds the -t argument to the gem install. This is a way to provide safety checks to make sure everything worked after the gem is installed.

has_rdoc is a way to tell gem you have included rdoc tags in the code. If this flag is false or missing, gem will not generate documentation automatically.

extra_rdoc_files allows you to include other files in the documentation that is generated by gem. In this case, the README file is being linked into the documentation. If you had other documents, they could be listed here.

Because IPAdmin is a very simple project, it does not include one very useful command: add_dependency. If you build a gem that depends on another gem, this command allows these dependencies to be specified. You even can tie it to a version number in the same way you can with require_gem. When you install a gem that has a dependency, gem checks to see if it is met. If it is not met, gem offers to install it. To add a dependency on rake, you could add this to the spec definition:

```
s.add_dependency("rake", ">=0.7.0")
```

Signing Gems

Thanks to a patch from Paul Duncan, the latest version of RubyGems (0.8.11) now has some features to support signing your gems using a public/private key. This introduces some new options for the gem specification (signing_key and cert_chain). This change also allows you to install gems in a high-security mode that will install only gems that are signed by trusted sources. Because the feature itself is very new, some pieces of infrastructure to make it useful in the greater scheme of things are missing—namely, an easy way to build up a chain of trust so that end users do not have to add certificates for every single gem author out there. That being said, these features might be useful if you want to control gems inside your network across a lot of servers. You could download them once and sign them with an internal certificate. Then, you could update all your servers by requesting gems from the server where you distribute these signed gems. Duncan has written a great overview of getting started with gem signing on the RubyGems Manuals site (see Resources).

Distribution

Now that you have a gem, you probably want to share it. There are several ways to distribute your code. The simplest way is to host the file. When people want to install it, they can download the file and run gem in the same directory.

The second option is to host the project at RubyForge.org. RubyGems

ships with RubyForge as the default source for gems. RubyForge even runs a special script so that once you upload your new gem to your account, it automatically is available to all users of RubyGems.

Assuming you do not want to use RubyForge, there are two options left to make it possible to distribute your gem via RubyGems. First, you need to run your own server. The easiest way to do that is to simply fire up gem_server. It automatically shares gems with anyone who connects to it.

The other option is to cd to a directory inside of the webroot of an existing Web server. Create a directory called gems, and copy all the gems you want to distribute into that directory.

Run the following command, and replace DIR with the full path to the directory above the gems directory. This creates yaml and Yam.Z files:

```
generate_yaml_index.rb -d DIR
```

You need to re-run the script anytime you modify the gems you are serving. Keep in mind that if you use either of these options, your users have to add the --source URL_OF_YOUR_SITE to the gem install command. This allows gem to search that site for gems.

Packaging

RubyGems is a package management system unto itself. If your system does not already have package management, this is a huge improvement. On the other hand, if your Linux system has package management, RubyGems can add some complexity. This is largely a side effect of RubyGems being completely separate from the host packaging system. According to the RubyGems Web site, the problem is related to the version-per-directory layout. This apparently conflicts with the Filesystem Hierarchy Standard (see Resources). Hopefully, some sort of middleground will be found, because the joy of having a good package management system is having a single place to make sure everything is up to date and works properly together. The risk is really related to gems that install non-Ruby code. For example, I believe it is possible to install a gem and then have the host package system replace a shared library that is managed by the host system with an incompatible version, which would render the gem useless.

In the long run, I hope that someone comes up with a good solution to the problem. So far, I have not been affected seriously by this potential issue. I use apt to manage Ruby and the rest of the system, and I use RubyGems to manage the gems I need. The one problem I had was more related to user error. I failed to install a library that RMagick required. The compilation of the RMagick extension failed, but I did not see the error because it scrolled by too fast, and the gem reported that it was installed. Eventually, I figured out what was going on, and no computers were harmed in the process. It could be argued that this problem may have been prevented if I were doing everything in apt, because it would have installed the missing library as soon as I installed RMagick. On the other hand, because a lot of the Rails and other Ruby gems seem to be updating frequently, it has been nice to be able to keep up with the latest version of the Ruby software instead of having to wait for new Debs to be released.

Conclusion

Package management for Ruby got off to a rocky start. Now that we have RubyGems, it is hard to imagine working without it. RubyGems crams a lot of features into a very tiny package. It has made it a lot easier to find, distribute and manage a wide variety of Ruby software. Now that you have made it through this brief introduction, you can start using gems in your own development.■

Resources for this article: www.linuxjournal.com/article/9019.

Dirk Elmendorf is one of the founders of Rackspace Managed Hosting (www.rackspace.com). He is currently addicted to Ruby on Rails, and by the time you read this he will be happily married to Annie Tiemann!

Expand Your HPC Flexibility.



Enjoy great dual-core performance plus the flexibility to run 32-bit and 64-bit applications with Dual-Core Intel® Xeon® Processors in your **AMAX HPC Servers**.

AMAX HPC servers leverage leading edge technology to provide an optimized solution containing innovative dual core and low voltage processing technologies and built-in Virtualization capability for deployment by many data centers throughout the world. Our ISO-9001 certified quality assurance facilities, with highly skilled engineers, provide customers with reliable, secure, cost effective, and flexible computing solutions!

AMAX HPC servers are deployed by clients in a wide spectrum of industries, including:

- Semi-conductors
- Telecommunications
- Network Security
- Entertainment
- Medical
- Printing
- Bio-chemical
- Banking
- Petroleum
- Education
- Government
- R&D Labs

To learn more about AMAX HPC Servers, visit www.amax.com.

See us at Boston Siggraph Aug 1 - Aug 3 Booth #930
See us at SF LinuxWorld Aug 14 - Aug 17 Booth #110

Toll Free: 1.866.626.2948 ext.0706 | WWW.AMAX.COM

©2006 AMAX Information Technologies, 1565 Reliance Way, Fremont, CA 94539. All rights reserved. Performance may vary depending on the system configuration and software applications used. Celeron, Celeron Inside, Centrino, Centrino Logo, Core Inside, Intel, Intel Logo, Intel Core, Intel Inside, Intel Inside Logo, Intel SpeedStep, Intel Viiv, Itanium, Itanium Inside, Pentium, Pentium Inside, Xeon and Xeon Inside are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

AMAX
AMAX Information Technologies
An ISO-9001 Certified Company

RUBY AS ENTERPRISE GLUE

How Ruby can glue together a vast number of enterprise resources. **Maik Schmidt**

Dynamic languages, formerly known as scripting or glue languages, always have been a valuable tool in every serious enterprise developer's toolbox. In the past, hordes of programmers have used Perl, Python, Tcl and the like to integrate disparate databases, message queues, LDAP repositories, Web services and so on. But, there's a new kid on the block called Ruby. In this article, I show how to solve common enterprise integration problems much more quickly and elegantly than with any other programming language available today.

A REAL ENTERPRISE INTEGRATION PROBLEM

To make things more tangible, let's solve a typical real-world problem. A provider of mobile telecom services wants to offer a new tariff based on the user's geographical position. People pay a lower fee when they use their cell phones within a radius of 500 meters around their home address.

To fulfill this requirement, the team developing the billing application needs a new HTTP service. The service gets a customer ID and should return the coordinates belonging to the customer's address in XML format. Our company already has a customer database, and it has access to a SOAP localization service. The target system architecture looks like Figure 1, and it's our task to build the new HTTP service.

Building it step by step, first we modify the customers database and build an access layer for it. Then, we implement a binding for the localization service, and finally, we hide all this behind a nice HTTP interface. As you might have guessed from the article's title, we use Ruby to do all of this.

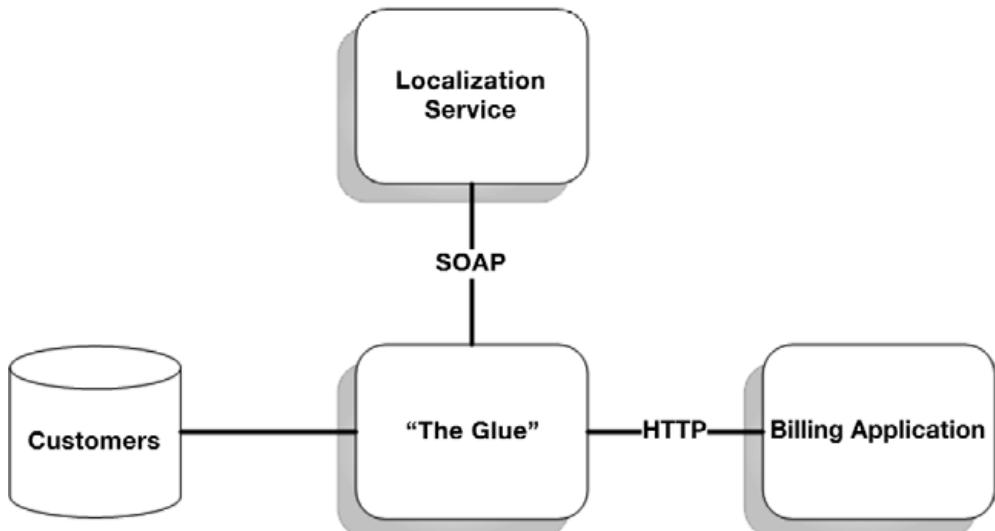


Figure 1. The Glue That Binds

NAS — LINUX BASED. PERFORMANCE TUNED.

You chose a Linux environment for many reasons, not the least of which includes its performance, reliability, security and open environment. Why not choose Network Attached Storage (NAS) that meets the same criteria? Introducing the Aberdeen AberNAS 170 and AberNAS 270 series of Linux based NAS appliances. They deliver all the features you love about Linux, in the highest performing, most competitively priced NAS in the marketplace today. Just look at these features:

- Best fit for Unix/Sun Micro/Linux environments or mix of Windows and non-Windows environments
- Browser based GUI management
- Substantially faster NFS and CIFS performance compared to Windows Storage Server
- Extremely virus resistant—Disc-On-Memory (DOM) operating system provides integrity, avoids corruption and allows for easy upgrade
- Embedded iSCSI target with unlimited initiators
- Schedulable volume snapshots
- PCBackup Utility included allows multiple file versions and instant backup of any modified file



ABERDEEN ABERNAS 170 SERIES

1U NAS with Hardware RAID

- Up to 2TB of Network Attached Storage in a 1U format
- Intel® Pentium® 4 processor 531 supporting Hyper-Threading Technology, 3.0E GHz with 800MHz FSB and 1MB Cache
- 1GB Dual Channel DDR400 SDRAM (Low Latency)
- Single controller for all drives with RAID level 0, 1 (10), 3 and 5
- 4 x Hot-Swap Hard Drives with single partition capability
- Slim DVD Drive
- 300W Power Supply
- Linux based proprietary NAS software (see above for features and benefits)
- Disc-On-Memory (DOM) operating system provides integrity and avoids corruption

• 5-Year Limited Warranty

AberNAS 170 **\$3,195**

- 1TB of Network Attached Storage
- 4 x 250GB Hot-Swap SATA Hard Drives with 8MB cache

AberNAS 176 **\$3,895**

- 1.6TB of Network Attached Storage
- 4 x 400GB Hot-Swap SATA Hard Drives with 8MB cache

AberNAS 178 **\$4,495**

- 2TB of Network Attached Storage
- 4 x 500GB Hot-Swap SATA II 3Gb/s Hard Drives with 16MB cache



ABERDEEN ABERNAS 270 SERIES

2U NAS with Hardware RAID

- Up to 4TB of Network Attached Storage in a 1U format
- Intel® Pentium® 4 processor 531 supporting Hyper-Threading Technology, 3.0E GHz with 800MHz FSB and 1MB Cache
- 1GB Dual Channel DDR400 SDRAM (Low Latency)
- Single controller for all drives with RAID level 0, 1 (10), 3, 5 and 6
- 8 x Hot-Swap Hard Drives with single partition capability
- Slim DVD Drive
- 460W Redundant Power Supply
- Linux based proprietary NAS software (see above for features and benefits)
- Disc-On-Memory (DOM) operating system provides integrity and avoids corruption

• 5-Year Limited Warranty

AberNAS 271 **\$4,995**

- 2TB of Network Attached Storage
- 8 x 250GB Hot-Swap SATA Hard Drives with 8MB cache

AberNAS 272 **\$6,995**

- 3.2TB of Network Attached Storage
- 8 x 400GB Hot-Swap SATA Hard Drives with 8MB cache

AberNAS 273 **\$7,995**

- 4TB of Network Attached Storage
- 8 x 500GB Hot-Swap SATA II 3Gb/s Hard Drives with 16MB cache

"powerful enough to tackle the most cutting-edge applications"

CRN Test Center Recommended
August 22, 2005 — Aberdeen Stonehaven A381

Finalist: Best Servers

LinuxWorld Product Excellence Awards
August 8-11, 2005 — Aberdeen Stonehaven A261

"solid redundancy for data drives, OS, and power"

PC Magazine
April 26, 2005 — Aberdeen AberNAS 211

"staggering ... eye-opening ... the highest WebBench numbers to date"

PC Magazine
February 8, 2005 — Aberdeen Stonehaven A261S

"extremely approachable and easy to use at a very affordable price."

CRN Test Center Recommended
January 10, 2005 — Aberdeen AberSAN i100

"terrific for video serving or other storage-intensive tasks"

PC Magazine
October 5, 2004 — Aberdeen XDAS



THE CUSTOMERS DATABASE

Customers are stored in a MySQL database called customers. It basically consists of only two tables: customer and address (Listing 1). Every entry in the customer table refers to an entry in the address table and vice versa. Both tables have a primary key called id that is generated automatically by the database.

Listing 1. `create_db.sql`

```
drop table if exists customer;
create table customer(
    id int unsigned not null auto_increment,
    forename varchar(64) not null,
    surname varchar(64) not null,
    created_on timestamp not null,
    primary key(id)
);

drop table if exists address;
create table address(
    id int unsigned not null auto_increment,
    customer_id int unsigned not null,
    street varchar(64),
    house_number varchar(16),
    postal_code varchar(16),
    city varchar(64),
    state char(2),
    primary key(id),
    foreign key (customer_id) references customer(id)
        on delete cascade
);
```

Because we have to store the coordinates of every address, we add a new table called locations (Listing 2). It contains the longitude and latitude belonging to every address.

Listing 2. `add_location.sql`

```
drop table if exists locations;
create table locations(
    id int unsigned not null auto_increment,
    address_id int unsigned not null,
    longitude double not null,
    latitude double not null,
    primary key(id),
    foreign key (address_id) references address(id)
        on delete cascade
);
```

Alternatively, we could add longitude and latitude columns to the address table, but our solution is less invasive. Perhaps there are applications performing SQL statements such as select * from address, and we do not want to break them.

That's all we have to do on the database side for the moment. Now, we'll see how to access the tables in a Ruby program.

ACCESSING THE DATABASE

There are many ways to access relational databases. For example, you can use the database's native interface or an abstraction layer

such as DBI. But in an object-oriented language like Ruby, an object-relational mapper (ORM) is by far the most convenient tool. ORMs map rows in a database table to objects and vice versa without a single SQL statement.

ActiveRecord is the most advanced ORM for Ruby and implements one of Martin Fowler's enterprise application patterns (see the on-line Resources). He defines it as follows: "[An active record is] an object that wraps a row in a database table or view, encapsulates the database access and adds domain logic on that data." Simply put, an active record is a class that provides the typical CRUD methods (Create, Retrieve, Update, Delete) for a single row in a database table.

ActiveRecord is part of the famous Ruby on Rails project, but it's completely independent of the rest and can be obtained and installed separately. We'll use it to map our three tables to classes.

If you've worked with an ORM before, you probably expect some boring configuration files now. How do we map a table to a Ruby class? How do we map a column to an attribute? Don't be afraid! You don't need all this, because ActiveRecord prefers convention over configuration. The short program in Listing 3 is all you need to map the Customer, Address and Location classes to the according tables.

Listing 3. `database.rb`

```
require 'rubygems'
require 'active_record'

class Customer < ActiveRecord::Base
  set_table_name 'customer'
  has_one :address
end

class Address < ActiveRecord::Base
  set_table_name 'address'
  belongs_to :customer
  has_one :location
end

class Location < ActiveRecord::Base
  belongs_to :address
end

ActiveRecord::Base.establish_connection(
  :adapter => 'mysql',
  :database => 'customers'
)
```

You do not have to do a lot. Derive every class from ActiveRecord::Base, and you get accessors for every column for free. These accessors have the same names as the corresponding columns. For example, the Customer class has accessors called id, forename, surname and created_on.

ActiveRecord maps a class to a table having the same name in plural form by default. A class named User is mapped to the users table, and a class named Location is mapped to the locations table. When you work with a legacy database, you cannot choose table names yourself. In such cases, specify the table name with the set_table_name method as we did for our two legacy tables.

Every table must have a numerical primary key called id that is filled by the database automatically. You can change the name of the primary key with the set_primary_key method, but if your legacy tables contain

complicated primary keys spanning several columns, ActiveRecord might not be the right tool for your job. ActiveRecord really shines when you adhere to its conventions.

Use belongs_to, has_one, has_many and has_and_belongs_to_many to declare relationships between the different classes. Naming is important for specifying relationships too. Note the naming scheme we have used for the foreign keys. In the address table, for example, the foreign key is called customer_id. By loose convention, many developers built the name of a foreign key column by appending _id to the name of the table to which the key refers. If you do this too, there's nothing more to be done.

In the last lines of Listing 3, we establish the connection to the MySQL database. If you need to, you can pass :host, :username and :password options.

Listing 4 shows how to insert a new customer and address into the database. It's all very intuitive, and we have to clarify only a few details. In line 7, we store a customer in the database. The save method automatically creates a new customer ID. We use this ID in line 10 to associate the address with the customer. ActiveRecord creates accessors for depending tables automatically—that is, all instances of the Customer class have an address attribute that refers to the according entry in the address table. What could be easier?

Listing 4. `create_customer.rb`

```
require 'database'

customer = Customer.new(
  :forename => 'Homer',
  :surname => 'Simpson'
)
customer.save

address = Address.new(
  :customer_id => customer.id,
  :street => 'Main Street',
  :house_number => '42',
  :postal_code => '75244',
  :city => 'Dallas',
  :state => 'TX'
)
address.save
```

We can find our new customer with one of the following statements:

```
customer = Customer.find(1)
customer = Customer.find_by_forename('Homer')
customer = Customer.find_by_surname('Simpson')
```

ActiveRecord dynamically creates tons of useful find methods. For example, Address.find(:all) iterates over all entries in the address table. In addition, you can search for arbitrary combinations of column values—that is, there are methods such as find_by_forename_and_surname.

Gone are the days when you had to fiddle with LEFT OUTER JOIN clauses and the like. ActiveRecord hides all this nasty stuff, and it even has many more useful features, such as single table inheritance and validations. It has been ported to nearly all database systems available today and is constantly enhanced by a big community.

LOCALIZING THE CUSTOMER'S ADDRESS

Now we know how to store the coordinates belonging to a customer's address in the database. The next thing to do is to calculate

Expert Included.



Our experts draw on years of experience to provide efficient yet powerful products. Jack likes systems based on the Dual-Core AMD Opteron™ processor because he knows that its Direct Connect Architecture provides optimized memory performance, balanced throughput, expandable I/O, and more linear symmetrical multiprocessing. He relies on AMD PowerNow!™ technology with Optimized Power Management to decrease overall system power consumption without compromising system performance.

When you partner with Silicon Mechanics you get more than a great AMD solution – you get an expert like Jack.



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

those coordinates. Normally, this would be a difficult problem and would call for a digital map. Luckily, we can delegate this job to a SOAP localization service.

Listing 5. loc_service.wsdl

```
<definitions
  name="LocServiceImplementationDescription"
  targetNamespace="example.com/wsdl/loc_service.wsdl"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="example.com/wsdl/loc_service.wsdl"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <message name="locate_in">
    <part name="street" type="xsd:string"/>
    <part name="house_number" type="xsd:string"/>
    <part name="postal_code" type="xsd:string"/>
    <part name="city" type="xsd:string"/>
    <part name="state" type="xsd:string"/>
  </message>

  <message name="locate_out">
    <part name="longitude" type="xsd:double"/>
    <part name="latitude" type="xsd:double"/>
  </message>

  <portType name="LocServiceInterface">
    <operation name="locate">
      <input message="tns:locate_in"/>
      <output message="tns:locate_out"/>
    </operation>
  </portType>

  <binding
    name="LocServiceBinding"
    type="tns:LocServiceInterface">
    <soap:binding style="rpc"/>
    <operation name="locate">
      <soap:operation soapAction="locate"/>
      <input>
        <soap:body namespace="urn:LocService"/>
      </input>
      <output>
        <soap:body
          namespace="urn:LocService"
          use="encoded"/>
      </output>
    </operation>
  </binding>

  <service name="LocalizationService">
    <documentation>
      Calculates coordinates of a given address.
    </documentation>
    <port
      binding="tns:LocServiceBinding"
      name="LocServicePort">
      <soap:address
        location="http://localhost:2000"/>
    </port>
  </service>
</definitions>
```

SOAP is a Remote Procedure Call (RPC) protocol standardized by the W3C. It allows you to create and use objects on a remote host as if they were part of your local process. Method calls and their parameters are turned into XML documents and are sent over a network layer. In the receiving process, they are converted back into method calls again. Return values and exceptions are represented as XML documents also and are sent back to the calling process. Although SOAP is independent of the transport layer, most applications use HTTP or HTTPS.

Listing 6. loc_service.rb

```
require 'soap/wsdlDriver'
include SOAP

class LocalizationService
  def initialize(wsdl_file)
    factory = WSDLDriverFactory.new(wsdl_file)
    @loc_service = factory.create_rpc_driver
  end

  def locate(address)
    @loc_service.locate(
      address.street,
      address.house_number,
      address.postal_code,
      address.city,
      address.state
    )
  end
end
```

Listing 7. Standalone SOAP Server

```
require 'soap/rpc/standaloneServer'

class LocalizationServer < SOAP::RPC::StandaloneServer
  def on_init
    @log.level = Logger::Severity::DEBUG
    add_method(
      self,
      'locate',
      'street',
      'house_number',
      'postal_code',
      'city',
      'state'
    )
  end

  def locate(street, house_number, postal_code, city, state)
    [97.03, 32.90]
  end

  server = LocalizationServer.new(
    'localization', 'urn:LocService', '0.0.0.0', 2000
  )
  trap(:INT) { server.shutdown }
  server.start
end
```

Fortunately, you normally do not have to know all these nitty-gritty details to use a SOAP service. It's sufficient to know where you can find it on a network, what methods it supports and what transport layer it uses. For this purpose, you'll usually use the Web Service Description Language (WSDL). The localization services' interface is described in Listing 5. Even if you're not familiar with WSDL, you should have no problems finding the definition of the locate function of the LocalizationService service. It takes an address (street, house number, postal code, city and state) and returns a two-element array containing its longitude and latitude.

Ruby has excellent support for SOAP because of the SOAP4R library (see Resources). It implements version 1.1 of the SOAP specification and is easy to use. If you've worked with SOAP before, you probably know what to do with a WSDL file. Normally, you'd use it to create skeleton code for a SOAP server or client you're going to implement. SOAP4R comes with a tool called wsdl2ruby.rb that turns a WSDL file into Ruby code. It can create code both for accessing a service having the interface described in the file and for creating a server that implements the interface.

We need a client that uses the localization service, and we could generate the complete code from the WSDL file with wsdl2ruby.rb. But in a dynamic language like Ruby, we don't need this intermediate step. It's easier to derive the client from a WSDL file on the fly. Listing 6 demonstrates how to do this.

The initialize method expects a WSDL file and creates a driver factory from it. This factory creates a driver (a synonym for proxy) for every service binding that has been specified in the WSDL file. We choose the RPC driver and treat the instance variable @loc_service as if it were a local object of class LocalizationService. In the locate method, we simply delegate the work to the localization service.

You need to run a standalone SOAP server to make these examples work, as shown in Listing 7.

BUILD AN HTTP SERVER

In a final step, we build an HTTP server that returns the coordinates belonging to a particular address as an XML document. It takes some time to calculate the coordinates, and the localization service isn't free either. Hence, we calculate coordinates only if necessary and store them locally in our database.

Back in the old days of the Internet, you had to use standards like the Common Gateway Interface (CGI) to create dynamic Web sites. Whenever a client requested a nonstatic page, the Web server called an external program—often a Perl or bash script—to create the content. The server passed it the current environment, including the client's request parameters, and returned the script's output to the requesting client. This approach causes a severe performance overhead, because the scripts have to be started as separate processes.

CGI programs have more disadvantages. First, they cannot easily maintain a state, because they are shut down immediately after they have done their work. Second, they are often a security problem, because they run in a more or less uncontrolled environment.

With the advent of Java, an alternative technology became fairly popular—servlets. Servlets are little code snippets that are executed by a so-called servlet container. They are loaded into memory only once and can be reused as often as necessary. This increases performance tremendously, and it allows developers to manage state information in the servlets. Eventually, the servlet container controls the environment of the servlets and can prevent them from performing unwanted operations such as deleting files.

Ruby ships with WEBrick (see Resources), a fantastic framework for creating HTTP servers. It allows you to follow the more or less obsolete CGI approach, but it strongly encourages the use of Ruby servlets. In Listing 8, you can see a servlet that implements the main logic of our service.

We have derived our servlet from class AbstractServlet. The



"Company With A Vision"

1-877-25-SERVER : www.genstor.com

GREAT MINDS – GREAT SOLUTIONS
Customized Solutions For
SERVERS – STORAGE – APPLIANCES
Linux - FreeBSD - x86 Solaris – MS etc.

SERVERS
LOW POWER – BUDGET – HIGH DENSITY
(1U, 2U, 3U and above - Xeon, P4, Opteron, VIA)



2U Dual Xeon/ Opteron Server
Upto 24GB RAM
Upto 8 SATA/ SCSI HDD
Starting @ \$ 2000.00
Call for your custom needs

STORAGE
SATA-NAS-DAS-iSCSI -SAS Storage Solutions



5U Dual Xeon/
Opteron SATA Storage
Upto 24GB RAM
Upto 26 SATA HDD
Upto 13TB Storage
Starting @ \$ 3930.00
Call for your custom needs

CALL FOR BEST PRICED iSCSI SOLUTIONS

CUSTOM APPLIANCE SOLUTION
Prototype – Certifications - Logo Screen Printing



Custom Turnkey OEM Appliance Solutions
From Prototype to Drop Ship
Custom OS/ Software Image Installs
No Quantity is small to Customize
Call for your Custom OEM Solutions

Contact Genstor For Your Hardware Needs:

Genstor specializes in customizing hardware around the OS of your choice (Linux, *BSD, x86 Solaris etc.) with Intel and AMD offerings. Please contact Genstor sales@genstor.com for all your hardware needs.

GENSTOR SYSTEMS, INC.

780 Montague Exp. # 604, San Jose, CA 95131
Phone: 1-877-25 SERVER or 1-408-383-0120
Email: sales@genstor.com Fax: 1-408-383-0121

Prices, Products and Availability subject to change without notice.

WEBrick server calls the do_GET method whenever it receives a GET request for a certain URL. Accordingly, it calls do_POST, do_PUT and so on for other HTTP request methods. WEBrick

Listing 8. servlet.rb

```
require 'rubygems'
require 'builder'
require 'database'
require 'loc_service'
require 'webrick'
include WEBrick
include WEBrick::HTTPServlet

class LocalizationServlet < AbstractServlet
  def initialize(server, wsdl)
    super(server)
    @loc_service = LocalizationService.new(wsdl)
  end

  def do_GET(req, res)
    customer_id = req.query['customer_id']
    customer = Customer.find(customer_id)
    address = customer.address
    if address.location.nil?
      lon, lat = @loc_service.locate(address)
      address.location = Location.new(
        :longitude => lon, :latitude => lat
      )
      customer.save
    end
    res['content-type'] = 'text/xml'
    res.body = to_xml(address.location)
    res.status = 200
  end

  def to_xml(location)
    xml = ''
    doc = Builder::XmlMarkup.new(
      :target => xml, :indent => 2
    )
    doc.position(
      :longitude => location.longitude,
      :latitude => location.latitude
    )
    xml
  end
end
```

Listing 9. server.rb

```
require 'servlet'

server = HTTPServer.new(:Port => 4242)
server.mount(
  '/', LocalizationServlet, 'loc_service.wsdl'
)
trap('INT') { server.shutdown }
trap('TERM') { server.shutdown }
server.start
```

always passes a Request and a Response object to the method it calls. Request objects contain all query parameters and headers that were sent by the client. It's your task to fill the Response object with a body and all headers that should be sent back.

In our case, the servlet logic reads like a pseudo-code specification. We try to read the geographical position of the customer having the ID customer_id from the database. If we cannot find it, we localize the customer's address using the localization service and store the coordinates in the database, so we do not have to localize it again. Next, we turn the coordinates into an XML document. At the end of the method, we set the content type, the HTTP status code and the body.

You do not have to define an initialize method for a servlet, but if you do, it always gets the server instance as its first argument. In our case, we also expect the name of the WSDL to be used to initialize the localization service.

The to_xml method converts a location into an XML document. Too often, developers use raw strings to create XML documents. I recommend never doing that, even for apparently trivial documents. Creating XML documents never is as easy as it seems, because you have to care about difficult concepts, such as well-formedness and character set encodings. Hence, we use Jim Weirich's XmlBuilder class (see Resources) to create the result document.

Now we have a servlet that implements our main logic, but a servlet alone won't cut it. We still have to create an HTTP server that controls it. Listing 9 is everything we need. We specify the port on which the server is listening and map our LocalizationServlet to the path /. In addition, we make the server terminate whenever it receives a SIGINT or SIGTERM signal.

A FINAL TEST RUN

It's time for a final test run. Point your favorite browser to http://localhost:4242/?customer_id=1 or use a command-line tool such as wget or curl to test our newly created service:

```
mschmidt:/tmp $ curl http://localhost:4242/?customer_id=1
<position longitude="97.03" latitude="32.9"/>
mschmidt:/tmp $
```

That's exactly the result we have expected. We're done!

CONCLUSION

There's no doubt, regarding enterprise integration, Ruby is ready for prime time. Even in this short article, we were able to cover some of the most important enterprise technologies, such as relational databases, SOAP and HTTP. You also can integrate your existing Java code, access LDAP servers, create XML-RPC services or manipulate XML documents with ease.

Ruby cannot compete in many respects with platforms such as J2EE or .NET, but it doesn't have to, and it doesn't want to. Its strengths are flexibility, maintainability and speed of development. Although the Ruby platform might not be the biggest compared to other dynamic languages, it might well be the one that's growing fastest. And, most important, it's a lot of fun!■

Resources for this article: www.linuxjournal.com/article/9018.

Maik Schmidt has worked as a software developer for more than a decade. He makes a living from creating complex solutions for mid-size enterprises. Besides his day job, he writes book reviews and articles for computer science magazines and contributes code to open-source projects. Recently he has written his first book called *Enterprise Integration with Ruby*.

Cool. Efficient. Reliable.



**Meet your
data center dream date.**

With challenges ranging from power shortages to thermal issues, today's data center managers are seeking cooler, more efficient and more reliable solutions.

Rackable Systems' award-winning servers and storage products are designed to reduce power consumption and increase reliability. Customers deploying Rackable Systems servers with DC power and the AMD Opteron™ processor experience monthly power savings of up to 30% and heat reduction in the chassis of up to 40%. And with density levels as high as 88 dual-core servers per cabinet, this all adds up to the most data center-friendly solution out there.

Time to meet your dream date.

To learn more and set up a first date, visit www.rackable.com/dreamdate.



**Rackable®
systems**

408.240.8300 www.rackable.com

Rails Writ Large

Ruby on Rails 1.1 and the paradox of how constraints can lead to greater freedom.

SCOTT RAYMOND

Ruby on Rails is a framework for Web application development that promises—and delivers—a powerful, productive and fun platform for building dynamic Web sites. A framework can be thought of as a library—a collection of functions for your application to use—but it's also more than that, it's a system of constraints for your code. Why would constraints be a good thing? Because by embracing constraints for a specific purpose, you actually enable creativity by focusing energy on the problem at hand. The Rails framework is a set of constraints that enable effective Web development. To get a feel for how it works, let's look at the parts that make up Rails.

ActiveRecord

Like most Web application frameworks, Rails follows the Model-View-Controller (MVC) design pattern, which divides your code into three logical layers. The model layer consists of domain objects, backed by a database, and the Rails component for that job is ActiveRecord. Note the three major features of ActiveRecord: associations, callbacks and validations. Associations allow you to define relationships between your ActiveRecord classes, such as one-to-one, one-to-many and many-to-many. Here's how it looks:

```
class User < ActiveRecord::Base
  has_many :projects
  has_one :address
  belongs_to :department
end
```

Details that normally would require configuration (table names, foreign key names and so on) are inferred automatically, and object attributes are created automatically for every column in the database. Rails calls this convention over configuration. Callbacks provide a robust set of hooks into the life cycle of your objects, where you can add behavior. For example, when user record is saved for the first time, send a welcome e-mail:

```
class User < ActiveRecord::Base
  after_create :send_welcome_email
  after_update :update_audit_log
end
```

Validations are a special kind of callback that make standard data validation routines a cinch:

```
class User < ActiveRecord::Base
  validates_presence_of :name
  validates_format_of :phone, :with => /^[0-9]{3}-[0-9]{3}-[0-9]{4}$/i
  validates_confirmation_of :email
  validates_acceptance_of :terms_of_service, :message => "must be accepted"
  validates_inclusion_of :age, :in => 0..99
end
```

By keeping your associations, callbacks and validations rules in the ActiveRecord class definition, you make it easier to create reliable, maintainable code.

ActionController

ActionPack has two subcomponents that work together closely, ActionController and ActionView. ActionController classes define actions—public methods that are accessible from the Web. Actions always end in one of two ways: either with a redirect (an HTTP response header sent back, causing the client to be forwarded to another URL) or with a render (some content being sent back to the client, usually an HTML file). When an action does a render, ActionView is invoked. Take a look at an example controller, with three actions:

```
class MessagesController < ApplicationController::Base
  def list
    @messages = Message.find :all
  end

  def show
    @message = Message.find params[:id]
  end

  def create
    @message = Message.create params[:message]
    redirect_to :action => :show, :id => @message.id
  end
end
```

The first action uses an ActiveRecord object to find all messages in the database and then renders the template messages/list.rhtml. The second action looks up one particular message by its ID and shows it. The third action also uses the ActiveRecord object, this time to save the parameters passed in from an HTML form. Then, it sends an HTTP redirect response, sending the user back to the show action.

Controllers and actions are mapped to URLs using routes. The default route is :controller/:action/:id, so without any additional configuration, the URL for the actions above would be /messages/list, /messages/show/1 and /messages/create.

In addition to actions, controllers also can have filters, which allow you to interrupt actions, and caches, which allow actions to execute faster. For example:

```
class MessagesController < ApplicationController::Base
  before_filter :authenticate, :except => :public
  caches_page   :public
  caches_action :show, :feed
end
```

ActionView

ActionView is Rails' system for formatting the output of your application—usually HTML files. The primary mechanism is ERB, Embedded Ruby, which will be familiar to anyone who has used PHP or JSP-like syntax. Any template file with an .rhtml extension can have embedded Ruby snippets, inside of <% %> and <%= %> tags. The first kind doesn't output anything, the second does. For example:

Figure 1. Add an Item to the Cart

```
<% for message in @messages %>
<h2><%= message.title %></h2>
<% end %>
```

You also can create template partials to extract commonly used chunks of markup, and helpers are Ruby functions available within your templates to provide handy functionality, like drop-dead easy Ajax. Lastly, special templates called layouts can hold markup that is common to the whole project (like HTML headers and footers).

Rails 1.1

The first public release of Ruby on Rails was version 0.5, in July 2004. More than a year later (and with nearly every line of code changed) Ruby on Rails 1.0 was announced in December 2005. That milestone was preceded by an intense push of polishing and testing to ensure that it was a solid release—so you might expect that the Rails Core team has coasted down the tracks since then, enjoying the phenomenal success of its software.

You might expect that, but you'd be wrong. In fact, they haven't slowed down one bit, and the next major release of Rails has just been announced. It's the biggest release to date, with more than 500 enhancements, fixes and tweaks. The majority of the 500 changes subtly polishes existing features, but some of them are superstars that promise to change the way your applications are developed. I've poured through the change logs to find the most interesting parts, and they can be lumped into three major groups: powerful Ajax, richer domain models and easy Web services.

Powerful Ajax

Arguably, the most significant new features in Rails 1.1 redefine the way Rails handles Ajax. Rails already had top-notch support for creating Ajax applications—it works by sending small snippets of HTML to a page to be inserted. Now, it also can return JavaScript to the browser to be evaluated. That means updating more than one page element in one step is a snap.

The kicker is that instead of writing the JavaScript by hand, it can be generated by Rails, using Ruby syntax. That's where RJS, Ruby-generated JavaScript, comes into play. In addition to .rhtml (Ruby HTML) templates, you can create .rjs (Ruby JavaScript) ones. In them, you can write Ruby

code that will generate JavaScript code, which is sent as the result of an Ajax call and evaluated by the browser.

Let's look at an example to see how this can be used. The on-line store IconBuffet uses RJS for its shopping cart (see www.iconbuffet.com/products/amsterdam to try it out). When a product is added to the cart, three separate page elements need to be updated to reflect the change. Before RJS, that would entail a dozen lines of JavaScript and multiple round-trips to the server. But now, it can be accomplished in one pass, with no custom JavaScript.

The Add to Cart button uses the standard Ajax link helper, just like before:

```
<%= link_to_remote "Add to Cart", :url => { :action =>
"add_to_cart", :id => 1 } %>
```

Clicking the link triggers the add_to_cart action, which updates the session and renders its template, add_to_cart.rjs:

```
page[:cartbox].replace_html :partial => 'cart'
page[:num_items].replace_html :partial => 'num_items'
page["product_#{params[:id]}"].addClassName 'incart'
```

The template is rendered into JavaScript that is sent back to the browser and evaluated, which updates the three page elements accordingly. You may be wondering where this page object came from—it's passed to RJS templates to represent JavaScriptGenerator, and it has many tricks up its sleeve:

1) Pop a JavaScript dialog box:

```
page.alert 'Howdy'
```

2) Replace the outerHTML of an element:

```
page.replace :element, "value"
```

3) Replace the contents of an element:

```
page.replace_html :element, "value"
```

4) Insert text:

```
page.insert_html :bottom, :list, '<li>Last item</li>'
```

5) Simulate a redirect with:

```
window.location.href: page.redirect_to url_for(...)
```

6) Call a JavaScript function:

```
page.call :alert, "Hello"
```

7) Assign to a JavaScript variable:

```
page.assign :variable, "value"
```

8) Call an effect:

```
page.visual_effect :highlight, 'list'
page.visual_effect :toggle, "posts"
page.visual_effect :toggle, 'comment', :effect => :blind
```

9) Show an element:

```
page.show 'status-indicator'
```

10) Hide elements:

```
page.hide 'status-indicator, 'cancel-link'
```

11) Refer to an element by ID:

```
page['blank_slate']
page['blank_slate'].show
```

12) Get elements with CSS selectors:

```
page.select('p')
page.select('p.welcome b').first
page.select('p.welcome b').first.hide
```

13) Insert some JavaScript:

```
page << "alert('hello')"
```

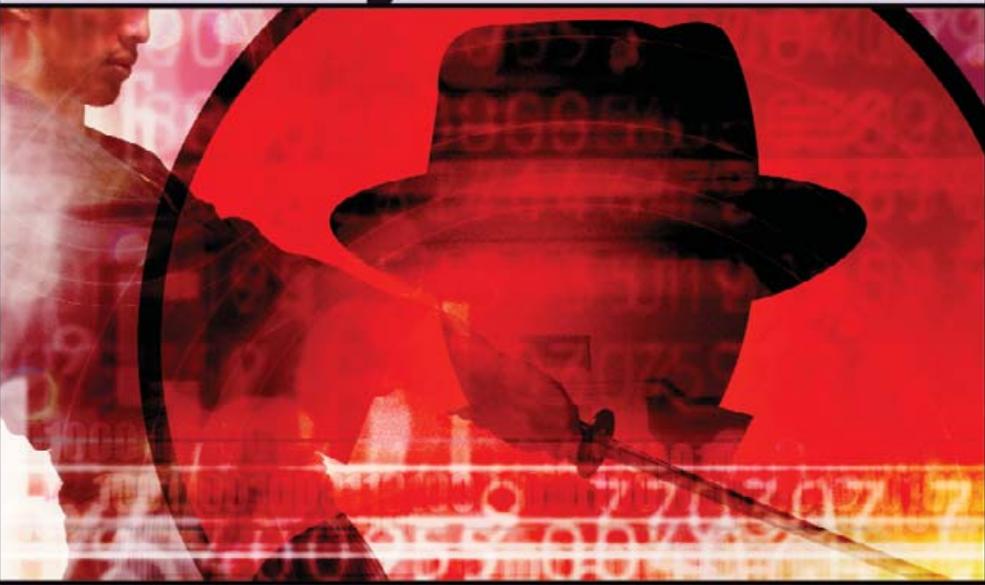
14) Make a draggable:

```
page.draggable 'product-1'
```

15) Make a droppable:

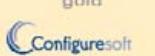
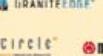
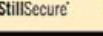
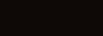
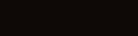
```
page.drop_receiving 'wastebasket', :url => { :action => 'delete' }
```

Hone your skills.



Defeat the attackers roaming your cyber streets!
 Black Hat USA will once again gather the world's information and computer security elite to share their knowledge and experience with you.
 Six days. Thirty-three classes. Sixty presentations.

Black Hat®
Briefings & Training USA 2006
 July 29 - August 3 • Caesars Palace • Las Vegas

sponsors		
platinum		
		
gold		
		
silver		
		
		
		
		
		

www.blackhat.com

16) Make a sortable:

```
page.sortable 'todolist', :url => { action => 'change_order' }
```

17) Delay execution:

```
page.delay(20) { page.visual_effect :fade, 'notice' }
```

Enumerable methods also can be used, and they'll generate the equivalent JavaScript code:

```
page.select('#items li').collect('items') do |element|
  element.hide
end
```

which generates this JavaScript:

```
var items = $$('#items li').collect(function(value, index)
  { return value.hide(); });
```

In addition to having .rjs files in your views directory, you also can write inline RJS. For example:

```
def create
  # (handle action)
  render :update do |page|
    page.insert_html :bottom, :list, '<li>Lastitem</li>'
```

```
      page.visual_effect :highlight, 'list'
    end
  end
```

Of course, you don't want to pollute your controllers with a lot of view-specific code, so you also can write RJS Helpers that can be called from update blocks. For example:

```
module ApplicationHelper
  def update_time
    page.replace_html 'time', Time.now.to_s(:db)
    page.visual_effect :highlight, 'time'
  end
end

class UserController < ApplicationController
  def poll
    render :update { |page| page.update_time }
  end
end
```

Debugging RJS can be tricky, because if a Ruby exception occurs, no error will be visible in the browser. To get around that, set config.action_view.debug_rjs = true, and you'll be notified of RJS exceptions via alert().

You may have noticed that the output of the RJS templates makes use of a great new feature of Prototype: methods of the Element class



Don't complicate a simple task

Keep basic tasks just that with handheld, stationary and vehicle-mounted wireless data collection from AML. While others are busy reinventing the wheel, we're keeping things simple, from our products to our personal service. Visit us at www.amltd.com or call **1.800.648.4452** for a real live person.

M7100 Wireless Family



AML
NEVER COMPLICATED™



are mixed into all HTML elements that are referenced by \$() and \$\$(). That means instead of writing Element.show("foo"), you now can write \$("foo").show(). It's a small change that makes writing JavaScript code more natural and Ruby-like. The methods available are visible(), toggle(), hide(), show(), visualEffect(), remove(), update(html), replace(html), getHeight(), classNames(), hasClassName(className), addClassName(className), removeClassName(className), cleanWhitespace(), empty(), childOf(ancestor), scrollTo(), getStyle(style), setStyle(style), getDimensions(), makePositioned(), undoPositioned(), makeClipping() and undoClipping().

Ruby-generated JavaScript also uses another fantastic new feature of Prototype, the Selector class and its corresponding \$\$() function. Like the \$() function, \$\$() is used to reference HTML elements, but this one matches elements by CSS selector strings. For example:

```
// Find all <img> elements inside <p> elements with class
// "summary", all inside the <div> with id "page". Hide
// each matched <img> tag.
$$('div#page p.summary img').each(Element.hide)

// Attributes can be used in selectors as well:
$$('form#foo input[type=text]').each(function(input) {
  input.setStyle({color: 'red'});
});
```

If you're not convinced by now, take it from me, RJS and the new additions to Prototype will revolutionize the way Ajax is done in Rails.

Rich Domain Models in ActiveRecord

So far, we've looked at advancements in the controller and view layers of Rails. Let's turn now to ActiveRecord, which also got a lot of love in this release. First up, a new type of association.

Prior to 1.1, Rails supported many-to-many relationships with has_and_belongs_to_many. For example:

```
class Author < ActiveRecord::Base
  has_and_belongs_to_many :books
end
class Book < ActiveRecord::Base
  has_and_belongs_to_many :authors
end
```

That works fine, to a point. The difficulty comes in when you need data or behavior for the association itself. The solution is to make an explicit join model for the association. Take a look at this alternative:

```
class Author < ActiveRecord::Base
  has_many :authorships
  has_many :books, :through => :authorships
end
class Authorship < ActiveRecord::Base
  belongs_to :author
  belongs_to :book
end
class Book < ActiveRecord::Base
  has_many :authorships
  has_many :authors, :through => :authorships
end
Author.find(:first).books.find(:all, :include => :reviews)
```

The new :through option of has_many allows you to specify an explicit association join model, so you can have the ease of has_and_belongs_to_many but get full power of ActiveRecord for the Authorship model.

The :through option also can be used where the intermediate association is a has_many. For example:

```
class Firm < ActiveRecord::Base
  has_many :clients
  has_many :invoices, :through => :clients
end
class Client < ActiveRecord::Base
  belongs_to :firm
  has_many   :invoices
end
class Invoice < ActiveRecord::Base
  belongs_to :client
end
```

Without the :through option, getting all invoices for a firm would require multiple SQL hits to the database or a custom SQL query. Now, ActiveRecord handles the join automatically and leaves a clean API to access the associations.

Another new association option that further enriches your domain models is polymorphic associations. This solves the problem of having a model that could share relationships with multiple other models. With polymorphic associations, the model defines an abstract association, which can represent any other model, and ActiveRecord keeps track of the details. Take a look at this example:

```
class Address < ActiveRecord::Base
  belongs_to :addressable, :polymorphic => true
end
class User < ActiveRecord::Base
  has_one :address, :as => :addressable
end
class Company < ActiveRecord::Base
  has_one :address, :as => :addressable
end
```

Any developer experienced with SQL has run into the "n+1 queries" problem, where looking up a set of records, each with a related record, causes a large number of queries to the database. The solution is SQL JOIN statements, but writing them by hand quickly gets complicated, especially after more than one join. Rails 1.1 significantly reduces that pain, with cascading, bottomless eager loading. Now, queries like Author.find(:all, :include => { :posts => :comments }) will fetch all authors, their posts and the comments belonging to those posts in a single query. For example:

```
Author.find(:all, :include => { :posts => :comments })
Author.find(:all, :include => [ { :posts => :comments }, :categorizations ])
Author.find(:all, :include => { :posts => [ :comments, :categorizations ] })
Company.find(:all, :include => { :groups => { :members => :favorites } })
```

The next major new feature of ActiveRecord is nested with_scope. This feature allows your dealings with ActiveRecord objects to be more clearly understood—especially important for code with security implications. Here's an example:

```
Developer.with_scope :find => { :conditions => "salary > 10000", :limit => 10 } do
```



~~I CAN'T GET MY DATA OUT OF OUR CRM SYSTEM.~~

~~CUSTOMIZING OUR CRM SYSTEM IS OFF-LIMITS.~~

~~OUR CRM SYSTEM IS SO INFLEXIBLE.~~

SugarCRM™ gives you control over your CRM implementation that proprietary, closed source CRM restricts. Now you can fully manage your business critical data.

Start using the most powerful commercial open source CRM solution built on the modern LAMP architecture for managing sales, marketing, support and group collaboration activities today.

Get hooked on SugarCRM. Visit: www.sugarcrm.com/linuxjournal or call +1 408.454.6941.

Commercial Open Source Customer Relationship Management



SUGARCRM®

COMMERCIAL OPEN SOURCE

www.sugarcrm.com • www.sugarforge.org

1.87 SUGAR CRM • +1.408.454.6941

```
# SELECT * FROM developers WHERE (salary > 10000) LIMIT 10:
Developer.find :all

# parameters are merged
Developer.with_scope :find => { :conditions => "name = 'Jamis'" } do
  # SELECT * FROM developers WHERE (( salary > 10000 ) AND ( name = 'Jamis' )) LIMIT 10
  Developer.find :all
end

# inner rule is used. (all previous parameters are ignored)
Developer.with_exclusive_scope :find => { :conditions => "name = 'Jamis'" } do
  # SELECT * FROM developers WHERE (name = 'Jamis'):
  Developer.find :all
end

end
```

The last major addition to ActiveRecord provides convenient syntax for accessing calculations and statistics, without writing custom SQL. For example:

```
Person.count
Person.count :conditions => "age > 26"
Person.count :include => :job, :conditions => "age > 26 AND job.salary > 60000"
Person.average :age
Person.maximum :age
Person.minimum :age, :having => "min(age) > 17", :group => :last_name
Person.sum :salary, :group => :last_name
```

Easy Web Services

The third big category of changes in Rails 1.1 involves creating Web services—specifically, embracing certain aspects of the HTTP protocol so that REST-style APIs can be implemented very easily.

The first piece of the equation is a new method for your actions, respond_to. This method parses the HTTP Accept header sent from the client, so that one action can return multiple response formats. For example:

```
class MessagesController < ActionController::Base
  def list
    @messages = Message.find :all
    respond_to do |type|
      type.html # using defaults, which will render messages/list.rhtml
      type.xml { render :xml => @messages.to_xml } # generates
      XML and sends it with the right MIME type
      type.js # renders index.rjs
    end
  end
end
```

In this example, a typical browser requesting /messages/list will get the HTML version of the data back, as usual. But an Ajax request for the same URL might send an Accept header of application/javascript—triggering the RJS template to be used instead. Yet another client might want to interact with your application as a Web service API, so it requests application/xml, and the same action handles that format as well. If you've wondered how hard it would be to add an API to your Web application, it's never been easier.

The above example includes a new option for the render method, :xml. It works just like render(:text => text), but sets the content-type to application/xml and the charset to UTF-8. You can specify the content-type header manually with the :content_type option. For example:

```
render :action => "atom.xml", :content_type => "application/atom+xml"
```

Arrays, hashes and ActiveRecord now have a to_xml method, and every object has to_json. These powerful additions mean that providing machine-readable versions of your application data can be accomplished with a few keystrokes. For example:

```
message.to_xml
message.to_xml(:skip_instruct => true, :skip_attributes => [ :id, bonus_time, :written_on,
  replies_count ])
firm.to_xml :include => [ :account, :clients ]

[1,2,3].to_json => "[1, 2, 3]"
"Hello".to_json => "\"Hello\""
Person.find(:first).to_json => "{\"attributes\": {\"id\": \"1\", \"name\": \"Scott Raymond\"}}"
```

The above examples demonstrate how easily you can enable a read-only API, but what if you want to accept input from the API as well? Well, it's remarkably simple:

```
class MessagesController < ActionController::Base
  def create
    @message = Message.create params[:message]
    redirect_to :action => :show, :id => @message.id
  end
end
```

But wait—isn't that the same code as the non-API version of the action? Indeed, Rails now examines the HTTP content-type header of the incoming POST and parses the input accordingly into the params object—just as if the data came from a Web form. By default, requests submitted with the application/xml content type are handled by creating an XmlSimple hash with the same name as the root element of the submitted XML. XML data is handled automatically, but what if you want to handle other formats as well? Enter pluggable parameter parsers:

```
ActionController::Base.param_parsers['application/atom+xml'] = Proc.new do |data|
  node = REXML::Document.new data
  { node.root.name => node.root }
end
```

There's (a Lot) More Where That Came From

Clearly we've barely scratched the surface of the new features in Rails 1.1—to say nothing of Rails as a whole. But you should now have a taste of Rails' latest additions. For more in-depth information and community (including books, screencasts, documentation, tutorials, demo applications, Weblogs, mailing lists and IRC channels), head to rubyonrails.com.■

Scott Raymond is a contributor to the Rails project and a professional Rails developer and consultant. He has been creating Web applications for ten years—in every role from Intern to IT Director, for clients ranging from indie-rock bands to Fortune 100 multinationals. He writes at scottraymond.net and will release a book with O'Reilly later this year. Scott holds a BA in Linguistics from the University of Kansas.

15 Years of BREAKING the rules...

Now we're Finally MAKING the Rules!

Join LinuxWorld as we celebrate the 15th Anniversary of Linux—and its amazing evolution from a kernel to industrial-strength corporate applications. Come to San Francisco—to the main event for the Linux community—and dive into the open technologies that are still transforming IT all these years later.

Go Deep in technical training

Brainstorm with the best

Check Out the latest products and services

Feed Your Mind in visionary keynotes

Feel the Energy of the entire Linux and open source community

Go Deep in 100+ Sessions on Key Linux and Open Source Topics

- Iron-Clad Security for Open Environments
- Managing Mixed Environments
- Virtualization
- Scalable Open Source Applications
- Desktop Linux
- Grid Computing
- Mobile Linux
- Web Services and SOA
- VoIP
- IT / Line-of-Business Alignment
- High-Performance Open Source
- The Business of Open Source
- Kernel and System Development

Register by July 14 and save

Enter priority code D0102 and save up to \$500
www.linuxworldexpo.com/register

Brainstorm with Top Open Source Experts

Alan Boda, Software Consultant, HP

Fabrizio Capobianco, CEO, Funambol, Inc.

Dave Dargo, Senior VP for Strategy and CTO, Ingres

Chris DiBona, Open Source Programs Manager, Google

Scott Handy, VP of Worldwide Linux & Open Source, IBM

Greg Kroah-Hartman, Linux Kernel Subsystem Maintainer, SuSE Labs / Novell

Dan Kusnetzky, Executive VP of Marketing, Open-Xchange Inc.

Eben Moglen, Chairman, Software Freedom Law Center

Bernard Traversat, Director of Advanced Development, Sun Microsystems

Register by July 14

to save up to \$500

Enter priority code D0102



Don't Miss the Biggest

Linux Event Ever!

Hundreds of products,
services, and training
sessions all under one roof!

OPEN. For Business.

KEYNOTE SPEAKERS



Lawrence Lessig
Professor of Law,
Stanford Law School



Guru Vasudeva
AVP & Enterprise Chief
Architect, Nationwide



Peter Levine
CEO, XenSource



Greg Besio
Corporate VP, Mobile
Devices Software, Motorola

**Check out the latest products and services
from hundreds of key exhibitors**

AMD
CA
Dell
EMC

HP
IBM
Intel
Novell

Oracle
Unisys
VMware
and more!

Feed Your Mind in Visionary Keynotes
by Lawrence Lessig and other Linux and
open source luminaries

Feel the Energy of a full slate of 15th
Anniversary Events

**Discover the
Full Power of
Today's Linux
Get the Skills,
Solutions, and
Insight Your
Business Needs**

**Get complete details at www.LinuxWorldExpo.com
and sign up for the event where Linux history is made.**

Register by July 14 with priority code D0102 and save up to \$500.



Conference: August 14 – 17, 2006

Expo: August 15 – 17, 2006

Moscone Center,
San Francisco

Plus expanded content with

OpenSolutions
WORLD™

LinuxWorld is open to business professionals only. No one under 18 years of age will be admitted.



© 2006 IDG World Expo Corp. All rights reserved. LinuxWorld Conference & Expo, LinuxWorld and
OpenSolutions World are trademarks of International Data Group, Inc. All other trademarks are
property of their respective owners.

D0102



PLATINUM SPONSOR



PLATINUM SPONSOR



SILVER SPONSOR



SILVER SPONSOR



MEDIA SPONSOR

OpenSSL Hacks

OpenSSL contains a command-line tool to do nearly everything possible within the OpenSSL library. Even better, it's probably already installed on your system. ANTHONY J. STIEBER

OpenSSL is a powerful Secure Sockets Layer cryptographic library. Apache uses it for HTTPS encryption, and OpenSSH uses it for SSH encryption. But, you don't have to use it as a library. It's also a multi-purpose, cross-platform crypto tool.

Here's a little background on OpenSSL. Before OpenSSL, there was ssleay, an SSL implementation written by Eric A. Young. It was successively improved and eventually became OpenSSL, similar to how NCSA HTTPd became the Apache Web server. Today, OpenSSL supports dozens of cryptographic algorithms and protocols with hundreds of options.

OpenSSL has a lot of features. In addition to the SSL client and server features, OpenSSL also has:

- US federal government NIST FIPS 140-2 Level 1 validation.
- TLS, the next generation of SSL.
- X.509 key and certificate generation.
- X.509 certificate authority.
- S/MIME encryption.
- File encryption and hashing.
- UNIX password hashes.
- Nine different commercial cryptographic hardware devices.
- Cryptographic performance testing.
- Thirty-six commands.
- Six message digest algorithms.
- Nine cipher algorithms with four block modes (where applicable).
- Multiple crypto protocols.

Although OpenSSL is complex, a lot of that complexity can be avoided. The remainder of this article concentrates on features that are easy to use, and in only a few lines of shell code.

This article uses the same section titles as in my earlier GnuPG article ("GnuPG Hacks", *Linux Journal*, March 2006, page 52) to make it easier to compare OpenSSL and GnuPG.

Getting Started

First, let's confirm that OpenSSL is installed and in your path. Many Linux distributions, even some small distributions, include OpenSSL. Usually OpenSSL is located in /usr/bin/, like most bundled packages.

In all of the examples, the shell prompt is set to \$.

First, type:

```
$ openssl version
```

Note that there are no leading dashes for the version option.

You should get something like this:

```
OpenSSL 0.9.7d 17 Mar 2004
```

The exact version number, date and other details may vary. At the time of this writing, the current version is OpenSSL 0.9.8a. The examples we're using should work for most versions of OpenSSL.

If you typed openssl with no command-line option, you'll get this:

```
OpenSSL>
```

If this happens, simply type quit or press Ctrl-C to exit safely. It's only the OpenSSL internal command-line prompt. It doesn't have command-line editing and has no explicit help function. But, it will type a list of commands if you type a command it doesn't know. It's better to avoid it for now.

Binary File Shields

Binary files typically are sent in e-mail using MIME. However, if your e-mail software doesn't support MIME, like most command-line e-mail, you're stuck with uuencode, or you can use OpenSSL's base64 encoding. Base64 is the same encoding used by the much more complicated MIME protocol, but it's not directly MIME-compatible.

To wrap a file in base64 text encoding, type:

```
$ openssl base64 < filename.bin > filename.txt
```

To unwrap a base64-encoded file, type:

```
$ openssl base64 -d < filename.txt > filename.bin
```

Note that OpenSSL doesn't care about the filename extension.

OpenSSL, unlike GnuPG or MIME, also can encode short strings, as follows:

```
$ echo "The Linux Journal" | openssl base64
VGhlIEpbnV4IEpvXJuYWwK
```

Decoding is similar:

```
$ echo "VGhlIEpbnV4IEpvXJuYWwK" | openssl base64 -d
The Linux Journal
```

Note the -d option, which specifies decoding.

Better Checksums

sum and cksum are the traditional UNIX checksum programs. They work fine, as long as as you don't need cross-platform compatibility or security, and you don't mind that occasionally two completely different files will have the same checksum value.

Although Linux systems often have md5sum installed, the MD5 algorithm suffers from a relatively new vulnerability and shouldn't be used anymore.

If it's installed, the more secure sha1sum should be used. Several different programs go by this name, however. Some versions can hash only one file at a time, some can't handle stdin or have some other deficiency. If you run into any of these problems or simply want consistent, known, good cross-platform software, consider OpenSSL.

The OpenSSL hash output format is a bit different from GnuPG, but numerically identical. OpenSSL format always identifies the algorithm used and also outputs a lowercase hexadecimal string with no whitespace. Some find this format easier to use.

Here are some examples:

```
$ openssl sha1 filename
SHA1(filename)= e83a42b9bc8431a6645099be50b6341a35d3dceb

$ openssl md5 filename
MD5(filename)= 26e9855f8ad6a5906fea121283c729c4
```

As in my previous "GnuPG Hacks" article, the above examples use a file that contains the string: "The Linux Journal". Note that there is no period in the string.

If you have a problem replicating these results, here is the ASCII-annotated hexadecimal representation of the file. Note the newline at the end automatically added by vi:

```
The Linux Journal \n
54 68 65 20 4c 69 6e 75 78 20 4a 6f 75 72 6e 61 6c 0a
```

OpenSSL, unlike GnuPG, doesn't have SHA-512, but OpenSSL does have MD2, MD4 and MDC2. These are older algorithms that are provided for backward compatibility. Like MD5, they shouldn't be used anymore.

Quick and Clean Encryption

Although not OpenSSL's strength, it also can encrypt files. The flexibility of OpenSSL makes it a bit more complicated than GnuPG.

OpenSSL has very few defaults, so more options have to be used. There are also many algorithms from which to choose. Some algorithms, like DES and RC4-40, are kept only for backward compatibility and shouldn't be used anymore. Strong algorithms you should use include bf, which is the Blowfish algorithm, and -aes-128-cbc, which is the US NIST Advanced Encryption Standard (AES) with 128-bit keys running in Cipher Block Chaining (CBC) mode.

Here is an example:

```
$ openssl enc -aes-128-cbc < filename > filename.aes-128-cbc
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
```

As with GnuPG, OpenSSL asks for a passphrase twice, which will not echo to the screen.

Decryption is also a bit more complicated:

```
$ openssl enc -d -aes-128-cbc -in filename.aes-128-cbc > filename
enter aes-128-cbc decryption password:
```

Note the -d in this example, which specifies decryption.

OpenSSL, unlike GnuPG, does not automatically detect the file type or even what algorithm, key length and mode were used to encrypt a file. You need to keep track of that yourself. In my example, I've put

Table 1. Password and passphrase strengths compared with estimated time to crack.
Note: time to crack is very rough. Your crackage may vary.

Type	Bytes	Characters	Bits/Character	Total Bits	Time to Crack
Base64 [A-Za-z0-9+/=]	6	8	6	48	Minutes to hours
Base64 [A-Za-z0-9+/=]	9	12	6	72	Years
Base64 [A-Za-z0-9+/=]	12	16	6	96	Decades
Base64 [A-Za-z0-9+/=]	15	20	6	120	Uncrackable?
Diceware Passphrase		8 words	12.9 per word	120	Uncrackable?

that information in the filename extension. OpenSSL won't manage the files and file extensions for you, you have to specify where you want the outgoing data written.

If you don't specify the correct algorithm, OpenSSL either may spew garbage or complain about a bad magic number. Either way, without the correct options, your data won't decrypt properly. To be fair, this is simply not something OpenSSL was designed to do, but it does work.

Passphrases

Before we go much further, we should discuss the importance of passphrases. In most cryptosystems, the passphrase is the secret that keeps the other secrets. It's usually the weakest point. So, creating strong

Linux Laptops Starting at \$990

**DON'T BE SQUARE!
GET CUBED!**



R³
Technologies™
Making Used to Bring Technologies to Life

309.34.CUBED
shoprcubed.com

passphrases is important, but it's also difficult, unless you have the right tools. Using OpenSSL, you can create a strong passphrase quickly.

A simple guide to passphrases is that longer is usually better, and eight characters is not long enough (Table 1). The goal is to make a secret that you can remember but that someone else won't know, can't guess or won't eventually stumble upon.

Generating a Passphrase

OpenSSL can create very strong random passphrases like this:

```
$ openssl rand 15 -base64
wGcwstkb8Er0g6w1+Dm+
```

If you run this example, your output will be different from the example, because the passphrase is randomly generated.

The first argument of 15 is the number of binary bytes to generate, and the second argument of -base64 specifies that those binary bytes should be encoded in base64 characters. For 15 bytes, the output always will be 20 characters, plus a newline character.

The base64 character set consists only of uppercase and lowercase letters A-Z, the numbers 1-9 and the three punctuation characters: plus, slash and equals. This is intentionally a limited character set, but more complexity in the character set is not necessarily better. Adding only one additional character makes up for the difference in security. For example, an eight-character fully printable ASCII password is about as strong as a nine-character base64 password.

Although not as quick as using OpenSSL rand, the Diceware passphrase generator produces strong and often easy-to-memorize passphrases. I highly recommend it.

Encrypted Passwords

Here's something that GnuPG can't do. OpenSSL has a built-in command for creating encrypted Linux passwords exactly like those made by /bin/passwd.

Skip this paragraph to avoid pedantic cryptography. Although commonly called encrypted, Linux passwords are actually hashed using either MD5 or the old UNIX password hash (based on the DES encryption algorithm). This allows Linux not to know your password even though it knows when you provide the correct password. When you set your password, Linux hashes your password and stores it in /etc/shadow. When you log in, Linux takes the password you provide and hashes it again, and then compares that hash to what's stored in /etc/shadow for your account. If they match, you provided the correct password and can log in. If they don't match, you don't know your password, and because only the hash was stored, the computer still doesn't know your password either.

Here's why making your own password hash is useful. Let's say you need a new password on another computer. Perhaps it's a new account or you forgot the old password and asked the system administrator to reset it for you. If you can talk to the sysadmin in person, this works well, but what if the sysadmin is somewhere else? Perhaps you've never even met your sysadmin. How do you communicate your new password? E-mail is not secure at all, and telephones are scarcely better. Postal mail takes days

Table 2. Hash and Block Cipher Performance
(The numbers are in 1,000s of bytes per second using 1,024-byte blocks.)

	AMD K6-2 300MHz, Linux 2.6.12, OpenSSL 0.9.7g	AMD Athlon 1.333GHz, Linux 2.4.27 OpenSSL 0.9.7d	PowerMac G5 1.6GHz, Darwin Kernel Version 8.0.0, OpenSSL 0.9.7b
md5	26,733.93k	169,207.13k	76,921.71k
sha1	12,665.41k	113,973.25k	76,187.82k
blowfish cbc	9,663.40k	56,940.54k	44,433.14k
aes-128 cbc	5,134.78k	31,355.90k	54,987.78k

Table 3. Public Key Crypto Performance

	sign/s	verify/s	
rsa 1024 bits	30.8	563.1	AMD K6-2 300MHz, Linux 2.6.12, OpenSSL 0.9.7g
dsa 1024 bits	61.6	50.7	AMD K6-2 300MHz, Linux 2.6.12, OpenSSL 0.9.7g
rsa 1024 bits	239.9	4,514.6	AMD Athlon 1.333 GHz, Linux 2.4.27, OpenSSL 0.9.7d
dsa 1024 bits	498.2	410.6	AMD Athlon 1.333 GHz, Linux 2.4.27, OpenSSL 0.9.7d
rsa 1024 bits	148.8	2,739.1	PowerMac G5 1.6 GHz, Darwin Kernel Version 8.0.0, OpenSSL 0.9.7b
dsa 1024 bits	289.5	234.6	PowerMac G5 1.6 GHz, Darwin Kernel Version 8.0.0, OpenSSL 0.9.7b

(and has other security problems). Fax machines, text messaging and most pagers aren't secure either.

It gets worse. Maybe you don't even trust that sysadmin. Sure, the sysadmin usually has root, but someone else knowing your password bypasses even root accountability. Maybe you want to use the same password on other computers, and you don't trust those sysadmins either.

So, do this instead:

```
$ openssl passwd -1
Password:
Verifying - Password:
$1$zmUy5lry$aG45DkcaJwM/GNlpBLTDy0
```

Enter your new password twice; it won't echo. If you have multiple accounts, run the above example multiple times. The output is the cryptographic hash of your password. The hash is randomly salted so that every time it's run, the output will be different, even if the password is the same.

In my example, the password hash is:

```
$1$zmUy5lry$aG45DkcaJwM/GNlpBLTDy0
```

Your password hash probably will be completely different, except for the initial \$1\$.

This password hash can now be e-mailed, faxed, text messaged or even spoken over the phone to a sysadmin to set as your password hash.

After the sysadmin receives your password hash, it can be entered into /etc/shadow manually or with chpasswd. The latter requires a temporary new file, call it newpassword, with your loginid and password

hash like this:

```
LoginidHere:$1$ywrU2ttf$yjm90XTIBnoKJLQK2Fw5c/
```

The file can contain multiple lines for other accounts too.

Then, the sysadmin runs this as root:

```
chpasswd --encrypted < newpassword
```

Now, the new password is set. It's a good idea to change your password once you log in, unless you use a really strong passphrase. This is because the password hash, once exposed, is subject to off-line brute-force attacks, unless the password is really long.

This method of resetting passwords can be quite secure. For example, using this technique, someone could take the password hash published above, create an account, then tell the person the loginid and hostname and that the password hash above was used. Only the person who originally created the password hash could know the password, even though the password hash was published in a magazine.

By the way, the password for that hash is 28 completely random base64 characters long, so it should be extremely difficult to crack. But please, don't create any accounts with that hash because the password is also published, it's:

```
HXzNnCTo8k44k8v7iz4ZkR/QWKM2
```

That password and hash were created like this:

```
$ openssl rand 21 -base64  
HXzNnCTo8k44k8v7iz4ZkR/QWKM2  
$ openssl passwd -1 HXzNnCTo8k44k8v7iz4ZkR/QWKM2
```

These examples use the MD5 password hashes now common on Linux systems. If you need to use the old UNIX password hash, simply leave off the -1. For example:

```
$ openssl passwd  
Password:  
Verifying - Password:  
xcx7DofWC0LpQ
```

The password for this last password hash example is: TheLinux.

Crypto Benchmarking

The many algorithms that OpenSSL supports makes it well suited for cryptographic benchmarking. This is useful for comparing the relative performance of different cryptographic algorithms and different hardware architectures using a consistent code base. And, it has a built-in benchmarking command.

The `openssl speed` command, by default, runs through every single algorithm in every single supported mode and option, with several different sizes of data. The different sizes are important because of algorithm start-up overhead.

A complete speed run takes about six minutes, regardless of hardware performance, and produces 124 lines of performance data with 29 lines of summary.

However, note that cryptographic algorithm

performance is extremely dependent on the specific implementation. For higher performance, OpenSSL has x86 assembly code for several algorithms. Other architectures, such as ia64, SPARC and x86-64, have much less assembly code, and most others have none. The assembly code is located in these OpenSSL source code directories: `crypto/*/asm`. Excerpts of the speed report for three different systems are shown in Tables 2 and 3.

Learning More

This is only a sample of what OpenSSL offers at the command line.

Some documentation is available at the OpenSSL Web site under the Documents and Related sections. There are also several mailing lists available under the Support section.

OpenSSL is written in and for C/C++, but it has been adapted to other programming languages, including Ruby. In addition, the FIPS 140-2 Level 1 validation in March 2006 makes OpenSSL a new contender in the enterprise and government cryptography markets.■

Resources for this article: www.linuxjournal.com/article/9020.

Anthony J. Stieber is an information security professional geek who obsesses over UNIX systems, cryptology, physical security and things he can't tell you. He is currently thawing out in Minneapolis, Minnesota, USA. This is his second published article.

DEDICATED SERVERS!

- » **100Mbps Dedicated Port**
- » **1200GB/mo Bandwidth**
- » **1GB RAM DDR400!**
- » **160 GB SATA2 Hard Drive**

\$59 /mo **\$99** /mo

64bit, 2.8GHz XEON!

64bit, 2.97GHz Celeron D

Upgradable to Dual HT!

cari.net/linuxjournal or 888.221.5902

betterServers. betterService. carinet

The Searchable Site

How to use Webglimpse to search and add search-based ads to your site. GOLDA VELEZ

Back when I was a curious undergrad, I attended a seminar on a tiny imaginary creature called Maxwell's Demon. This clever beastie can make an engine run off hot air, but only if it knows the location and speed of all the hot air molecules. Essentially, the Demon transforms knowledge into energy (strictly speaking, information plus heat into work, or usable energy). I think the Demon stuck in my mind because it demonstrated in a physical way the value of information, especially organized information.

A Web site rich in useful content attracts visitors because of its valuable information. Adding a search engine multiplies that value. And, what if you don't have your own content-rich site? Do you have, perhaps, a collection of favorite bookmarks on a particular subject? Using Webglimpse, you can create a form on your own Web site that allows users to search those Web sites. Now your work of researching and selecting those high-quality, subject-specific sites you bookmarked can help attract visitors to your own site. In this article, I describe how to use Webglimpse to enable users to search the content of your chosen Web sites, and how to generate ad revenue quickly from your traffic.

The Webglimpse Story

Webglimpse is a creature of several parts: a spider and manager, written in Perl, and Glimpse, the core indexing/search algorithm written in C. Glimpse was created first, by Udi Manber and Sun Wu, Computer Science professors who wanted to apply the neat new search algorithm for finding fuzzy patterns that they had developed (and released as agrep) when Sun Wu was Manber's student. Glimpse was originally written in 1993 as "a tool to search entire filesystems", useful for people who have ever misplaced a file or old e-mail message somewhere on their hard drive.

Webglimpse was wrapped around Glimpse a few years later, as a way to apply the powerful searching and indexing algorithms to the task not of searching the entire Web, but of combining browsing and searching on individual Web sites. Written by grad students Pavel Klark and Michael Smith, Webglimpse introduced the notion of searching the "neighborhood" of a Web page, or the set of pages linked from that page. Meanwhile, Manber and another student, Burra Gopal, continued to add features and refine Glimpse to make it optimally useful in its new context.

I arrived on the scene at about this point. I'd just quit my job debugging assembly network code at Artisoft in order to start my own company doing something with discovery and categorization of information on the Internet. In the early Web of 1996, Webglimpse stood out as the most promising search tool. It was newborn and still rough around the edges, so when Udi Manber accepted my offer to help with the project, my first job was to rewrite the install. I became more and more involved with Webglimpse, adding features and supporting users, and in January 2000 the University of Arizona granted my company exclusive license to resell it. I didn't feel I could make Webglimpse my primary focus and still make a living if it were open source, but I did make the decision always to distribute all the source code and to keep it free for most nonprofits. As a result, many users were willing to provide feedback and patches, and I was able to provide free, even resalable, licenses to anyone who helped with the project.

Making It Work

Having been around for many years, Webglimpse runs on almost any Linux version and configuration. The only real prerequisites are a Web server with Perl 5.004 or above and shell access to that server.

Full details regarding installation are available on the Webglimpse home page (see the on-line Resources), so I mention only a few tips here. If you find Webglimpse already installed on your system, check the version. Most of the preinstalled copies out there are old (v. 1.6, circa 1998), and it's likely you have rights to upgrade. The simplest way to check the version of Webglimpse is to run a search and view the source of the results page. The version number is in a comment line at the beginning of the search results.

At the time of this writing, Webglimpse 3.0 is beta testing a new FTP-only install. You can try this version, or install the older 2.0 if you have SSH access to your server. To go the SSH route, first download trial version tarballs from the site. Follow the Installation Instructions linked at the top of the download page, which tell you first to compile and install glimpse by the usual steps:

```
./configure  
make  
make install
```

then to install Webglimpse by running its installation script:

```
./wginstall
```

The script walks you through the usual choices as to installation directory and where to put the cgi scripts. It also tries to parse an Apache configuration file if it finds one, and it confirms with you the key Domain name and DocumentRoot settings for your server. Because Webglimpse can index local files on your hard drive and map them to URLs, it needs to know how to translate filesystem paths to URLs. This is such a key point that in the Web administration interface, there is a screen devoted to testing URL→file and file→URL translations to make sure it is set up correctly.

Other settings during the install are security-related. In order for the archive manager to run from the Web, the directory where archives are placed needs to be Web-writable. The most secure way to do this is *not* to make it world-writable, but rather owned by the Web user, which is the user name that your Web server runs as. Most often this is www or nobody. You can tell by examining the process list:

```
ps aux | grep httpd
```

which will show a number of lines something like:

```
nobody    873  0.1  0.5 16492 11416 ?  
      S 18:03  0:00 /usr/local/apache2/bin/httpd  
nobody    874  0.0  0.5 16492 11416 ?  
      S 18:03  0:00 /usr/local/apache2/bin/httpd  
nobody    875  0.0  0.5 16552 11620 ?  
      S 18:03  0:00 /usr/local/apache2/bin/httpd
```

The first column is the user name the Web server is running as—in this case nobody. Now you can answer the prompt when wginstall asks, and if you are running wginstall as a user that is capable of changing ownership, it sets the ownership for you. If not, become root after the install is complete and change ownership manually. Supposing you installed to the default location of /usr/local/wg2, you would run:

```
chown -R nobody /usr/local/wg2/archives
```

to make the archives directory Web-writable.

Once the install is complete, it is time to choose the files you want to index and create the search form. Webglimpse calls this Configuring an Archive.

Configuring an Archive

Upon completing the install, you will see something like the following:

```
*****  
Done with install! You may use  
http://mycoolserver.com/cgi-bin/wg2/wgarcmin.cgi
```

or:

```
/usr/local/bin/wgcmd  
to configure archives at any time.  
(The web version currently has more features)
```

Run wgcmd to create new archive now? [Y]:

Once you are familiar with Webglimpse, the command-line tool is very handy for managing multiple archives or quickly setting up new ones. Your first time, I'd recommend using the Web version. So you enter N to not run wgcmd, and instead open the wgarcmin.cgi URL in your browser, and enter the user name and password you chose during the install. This brings you to the archive manager, which will later list all the archives you have configured. If this is your first time installing, the list is empty, so press Add New Archive. Now you should see the New Archive screen shown in Figure 1.

Here you can enter a title and description, and optionally select a category and language. The language doesn't restrict the sites you can include, but it does select a template and character set for the search form and results page. Then you will click on one of the buttons at the bottom:

- Index by Directory: lets you index files already on your Web server in a specific directory.
- Index by Site: lets you index everything at a particular Web site, either on your server or somewhere else on the Net. Use this for dynamic files on your own server.
- Index by Tree: lets you index everything linked to from a particular starting page, with configurable settings for how many and which "hops", or links, to follow.

After entering the specific directory or URL to index and entering settings, such as maximum number of pages, you will come to the main control screen for managing an archive. Here you can add additional sources of pages to index, so that one archive can combine local files, remote sites and trees of remote pages on multiple sites if so desired. Figure 2 shows the archive ready to go. Once you press the Build Index button, the spider goes out and gathers remote pages, filters out the HTML tags, and then runs glimpseindex to create a block-level inverted index for fast searching.

Finally, you can get a search form to include in your pages by clicking on the text link Add a search box or page

The screenshot shows the 'Web Glimpse Administration' interface with a teal header bar. Below it is a white form titled 'New Archive'. The 'Title' field contains 'My Cool Archive'. The 'Category' dropdown menu is set to '/Computers/Security' and lists several options: Firewalls, Unix, Java (which is selected), NT, and Hackers. The 'Subcategory' dropdown menu is empty. The 'Description' text area contains the text: 'French language pages on the subject of Security issues for Java apps. The best collection on the net!' Below the form are three buttons: 'Index by Directory' (highlighted in yellow), 'Index by Site', and 'Index by Traversing Link'. At the bottom are three help links: 'Help on Dir', 'Help on Site', and 'Help on Tree'.

Figure 1. The New Archive screen allows you to enter a title and description, choose category and select the language in which the search results should be presented.

to your website. This link takes you to a page with the source code for three example search forms for this archive, ranging from a bare input box to an advanced search form supporting regular expressions and making all the options visible to the user. The simple version of the search form, supporting *all*, *any* or *exact phrase* searching, is shown in Figure 3.

Ultra Dense, Powerful, Reliable... Datacenter Management Simplified!

15" Deep, 2-Xeon/Opteron or P4 (w/RAID) options



Customized Solutions for... Linux, BSD, W2K

High Performance Networking Solutions

- Data Center Management
- Application Clustering
- Network and Storage Engines

Rackmount Server Products

- **1U Starting at \$499:** C3-1GHz, LAN, 256MB, 20GB IDE
- 2U with 16 Blades, Fast Deployment & more...



iron
SYSTEMS™

Iron Systems, Inc.
540 Dado Street, San Jose, CA 95131
www.ironsystems.com

CALL: 1-800-921-IRON

The screenshot shows the 'Manage Archive #2: My Cool Archive' page. It includes fields for 'Title' (My Cool Archive), 'Category' (I), 'Language' (Français), 'Description' (French language pages on the subject of security issues for Java apps. The best collection on the net!), 'Archive Dir' (/home/godar/test/archives/2), 'Optional - include Sponsored SearchFeed links' (checkbox checked), 'SF partner ID' (text input), 'SF track ID' (text input), 'Number of results to include' (text input), and 'Additional relevance keywords for this archive' (text input). Buttons at the bottom include 'Search Archive', 'Save Changes', 'Build Index', and 'Delete Archive'. A note at the bottom states: 'For security reasons, some changes cannot be made through the web interface. The following tasks must be done through a telnet session: Add a search box or page to your website Reindex automatically from your crontab'.

Figure 2. The Manage Archive page lets you combine Web pages from different sources into one searchable index.

You also can get the same forms by pressing the Search this Archive button or entering the URL directly to the Webglimpse cgi (<http://mycoolserver.com/cgi-bin/wg2/webglimpse.cgi?ID=2>). Normally they are generated in the language of the archive, but we're showing them in English here.

We Know Linux!
Your Embedded Linux Partner

- Single Board Computers
- Custom Drivers
- Flash Disk
- USB Support
- Fanless CPUs
- PCMCIA
- PC/104
- IP Router/Firewall

2.6 Kernel

Real-Time Starter Kits Enclosures Panel PCs X-Windows CAN HTTP Server Data Acquisition

EMAC can fulfill all of your Embedded Linux needs, from credit card size systems to rack-mounts.

Since 1985
OVER
21
YEARS OF
SINGLE BOARD
SOLUTIONS

EMAC, inc
EQUIPMENT MONITOR AND CONTROL

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

The screenshot shows the 'Search My Cool Archive with WebGlimpseII' page. It includes fields for 'Search for' (text input), 'Look for' (dropdown menu: 'ALL the words'), 'Optional - only find matches WITHOUT the words:' (text input), and 'For ALL word queries, require that all keywords appear within a span of [] words.' Below these are instructions for 'Advanced search'.

Figure 3. The search templates that come with Webglimpse can be modified to fit your Web site. Only the form action and the names of the input tags need to be preserved.

Making It Pay

So, now you have a searchable archive of your chosen sites, the coolest collection of links anywhere on your particular subject. Users everywhere can take advantage of your research and use your server to search through your highly optimized index. If you want, you now can serve ads in order to generate revenue and support your site. Back on the Manage Archive screen shown in Figure 2, check the box labeled Optional - include Sponsored SearchFeed links. Then, click on Set up/manage Account, which is a link to set up an account with Searchfeed.com. An on-line advertising and content provider company, Searchfeed.com provides sponsored search results that are supposed to be relevant to the keywords on which the user searches. Once your account is set up, simply enter the partner ID and track ID provided by Searchfeed.com and choose how many ads should appear at the top of your search results. It's pretty simple to set up. To get the most out of your ads, you can use the suite of on-line tools provided by Searchfeed.com to monitor what keywords users are searching on, which ads they are clicking on and how much you make from each click.

Customizing

Whether or not you choose to add sponsored links to your search results, very likely you will want to wrap them in the "skin" of your site—your own look, feel and navigation menus. To accomplish this, you need to edit the file named wgoutput.cgi in the archive directory. (The location of the archive directory is shown on the Manage Archive screen.) This file contains the snippets of HTML code that go above, below and in between individual search results. You also can include your own header and footer files instead of typing in the HTML.

In some cases, you also may want to customize the ranking order of your search results. Webglimpse, unlike some search engines, doesn't claim to know what "percent relevant" a particular page is to the user. Instead, it lets you see under the hood how it calculates relative relevance of search results, and if you like, you can implement your own customized relevance ranking formula(s). Simply edit the file .wgrankhits.cfg in that same archive directory with a snippet of Perl code using these available variables:

```
# Available variables are:
#
# $N          # of times the word appears
# $LineNo     Where in the file the word appears
# $TITLE      # of matches in the TITLE tag
# $FILE       # of matches in the file path
# $Days       Date (how many days old the file is)
# $META       Total # of matches in any META tag
# $LinkPop    Link popularity in the site (how
#             many times other pages link to it)
# %MetaHash   Hash with the # of times the word
#             appears in each META tag, indexed
#             by the NAME= parameter.
# $LinkString actual url of link
```

Advertiser Index

```
# The following uncommented lines
# are the actual ranking formulae that will be used

# This is the default ranking, it gives high weight
# to keywords in the title, plus some weight to
# regular hits, link popularity and freshness

$TITLE * 10 + $N + $LinkPop + 5/($Days + 1)
```

By making use of the \$LinkString variable, for instance, you can make sure that selected regions of your site always appear above others. In the Webglimpse home page, for example, we add this term to the default formula to make sure that pages in the /docs directory appear first in the search results:

```
+ ($LinkString =~ /\docs\//)*1000
```

Troubleshooting

By now you may have an inkling of what the strengths and weaknesses are of Webglimpse: a bunch of neat features that are directly configurable by the user, and a bunch of neat features combined in a somewhat ad hoc manner. Webglimpse has, depending on your perspective, enjoyed or suffered from a great deal of tweaking to make it able to perform a lot of different tasks. The next version, which is in the works at the time of this writing, is intended to be simpler to install and maintain, and even to have an FTP-only install for users without shell access to their servers. Be that as it may, the most common problems you are likely to run into with the current version are as follows:

1. Permissions issues—these occur when you sometimes re-index from the Web administration interface, and sometimes from a shell or from your crontab. You can re-index any archive either by pressing the Build Index button in the Manage Archive screen or by running the script `./wgreindex` from the archive directory. The best thing to do is decide on one way to re-index, stick to it and make the archive owned by the user who will run the re-index script.
2. URL/file translation issues—these occur mainly when the DocumentRoot is not correctly specified. You can check what file a given URL will be translated into or vice versa by pressing the Test Path Translations button on the main Web administration screen. All the applicable settings for local and remote domains are stored in this file: `/usr/local/wg2/archives/wgsites.conf`. You can edit `wgsites.conf` directly, or make changes by pressing the Edit Domain Configuration button in the Manage Archives screen.

More troubleshooting tips are available in the Documentation and How-tos page (see Resources).

Acknowledgements

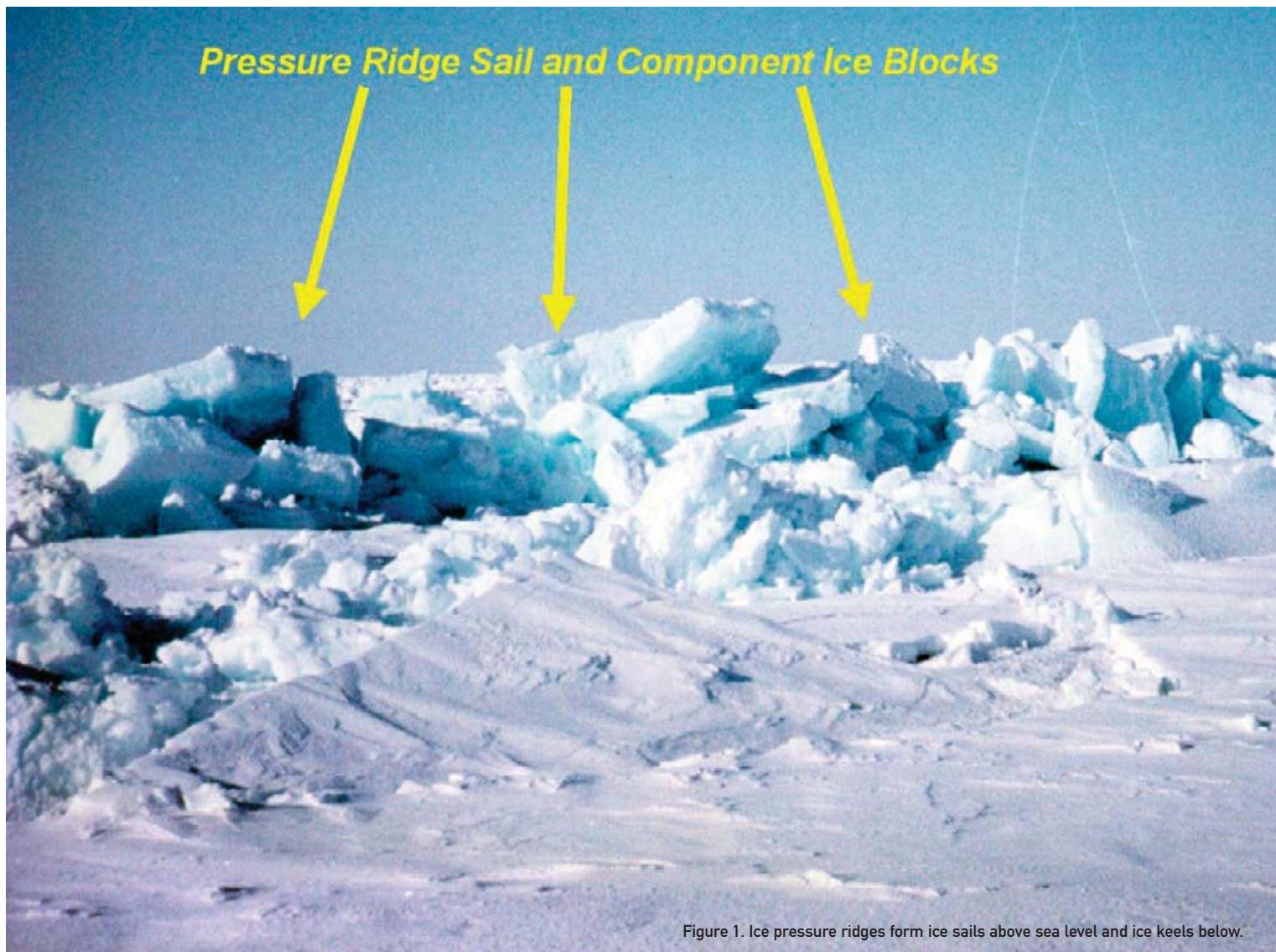
The author thanks Udi Manber for trusting me with these wonderful creations! I am still trying to take care of them as they deserve! Thanks also to Sun Wu, Burra Gopal, Michael Smith and Pavel Klark, the co-creators of Webglimpse and Glimpse, and to all the users who have submitted bug reports, patches, translations and suggestions over the years. ■

Resources for this article: www.linuxjournal.com/article/9021.

Golda Velez is a programmer and mom living in Tucson, Arizona. She has been the primary maintainer of Webglimpse since 1997. She is also the owner and founder of Internet WorkShop, a Web hosting and consulting company started in 1995.

For advertising information, please contact our sales department at 206-782-7733 ext. 2 or ads@ssc.com.
www.linuxjournal.com/advertising

Advertiser	Page #	Advertiser	Page #
ABERDEEN, LLC www.aberdeencorp.com	59	LINUXWORLD CONFERENCE & EXPO www.linuxworldexpo.com	73
AMAX www.amax.com	57	LPI www.lpi.org	95
AML www.amltd.com	69	MBX www.mbx.com	23
APPRO HPC SOLUTIONS appro.com	C2	MICROWAY, INC. www.microway.com	C4
ASA COMPUTERS www.asacomputers.com	53, 83	MIKROTIK www.routerboard.com	7
BITDEFENDER LLC www.bitdefender.com	29	O'REILLY OSCON conferences.oreillynet.com	91
BLACK HAT BRIEFINGS (CONFEX PARTNERS LTD) www.blackhat.com	68	OPEN SOURCE STORAGE www.opensourcestorage.com	13
CARINET www.carinet.net	77	PENGUIN COMPUTING www.penguincomputing.com	41
CIARA TECHNOLOGY www.ciara-tech.com	8, 9, 16, 17	THE PORTLAND GROUP www.pgroup.com	55
CORAID, INC. www.coraid.com	27	RACKABLE SYSTEMS www.rackable.com	65
COYOTE POINT www.coyotepoint.com	3	RACKSPACE MANAGED HOSTING www.rackspace.com	C3
CYCLADES, AN AVOCENT COMPANY www.cyclades.com	1	R CUBED TECHNOLOGIES www.rcubedtech.com	75
EMAC, INC. www.emacinc.com	80	SBE, INC. www.sbei.com	25
EMPERORLINUX www.emperorlinux.com	15	SERVERS DIRECT www.serversdirect.com	21
FAIRCOM CORPORATION www.faircom.com	39	SILICON MECHANICS www.siliconmechanics.com	51, 61
GENSTOR SYSTEMS INC. www.genstor.com	63	SUGARCRM, INC. www.sugarcrm.com	71
HPC SYSTEMS, INC. www.hpcsystems.com	35	SUPERMICRO www.supermicro.com	49
HURRICANE ELECTRIC www.he.net	31	TECHLOGIC SYSTEMS www.embeddedx86.com	40
INTEGRITY LINUX SYSTEMS www.integritylinux.com	37	THINKMATE www.thinkmate.com	19
INTEL www.intel.com/software/products	11	TYAN COMPUTER USA www.tyan.com	5
IRON SYSTEMS www.ironsystems.com	79	TUX MAGAZINE www.tuxmagazine.com	85
JTL NETWORKS www.jtl.net/lj	6	ZT GROUP INTERNATIONAL www.ztgroup.com	33
LAYER 42 NETWORKS www.layer42.net	89		



Under-Ice Sonar Visualization

The Naval Undersea Warfare Center provides naval and environmental scientists with an integrated 3-D display of under-ice sonar information.

RICHARD R. SHELL, GARNER C. BISHOP AND DOUGLAS B. MAXWELL

In World War II, the Arctic became an active theater of operations for German and Soviet subs, which occasionally ducked under the ice to escape detection. After the war, interest in cold-water acoustics led to work on sonar and navigation instruments that could be applied to submarines operating in the Arctic. As a result of the Cold War and the corresponding growth of naval concern about the possibilities of nuclear warfare in the Arctic, an under-ice capability for nearly all US submarines was implemented.

With the appearance of the first nuclear submarine, USS *Nautilus*, the Arctic Ocean beneath the pack ice finally could be explored fully and used

for naval operations. Today, under-ice operations are standard for submarines of the US and other nations. Furthermore, studies suggest that sonar could be used as a tool for detecting and localizing under-ice oil spills. So, for both strategic and environmental reasons, the study of under-ice sound properties is important.

For more than two decades, the Naval Undersea Warfare Center, located in Newport, Rhode Island, has been investigating and modeling the under-ice sound environment. Current work involves the use of 3-D visualization to aid in the understanding of complex scattering that results from the

impact of sound energy on the ice blocks making up an underwater pressure ridge. These pressure ridges, called ice sails above the water's surface and ice keels below the surface, are formed when floating Arctic ice collides (Figure 1).

Current 3-D visualization work builds on a previous effort that was designed conceptually to show submarine commanders target location volumes created by the rendering of data from towed submarine sound sensors. Subsequently, this software has been modified and greatly enhanced to display environmental information in all parts of the world, including under the Arctic ice pack. The enhanced 3-D display is capable of immersive stereopsis viewing in multiple environments, including fixed shore-based facilities, such as a 3-D CAVETM, or on mobile systems, such as a laptop using a head mounted display (HMD, Figure 2).

It is anticipated that through the use of these high-level graphics techniques that both rapid target identification, be it tactical or environmental,



Figure 2. 3-D Visualization Work at the Naval Undersea Warfare Center

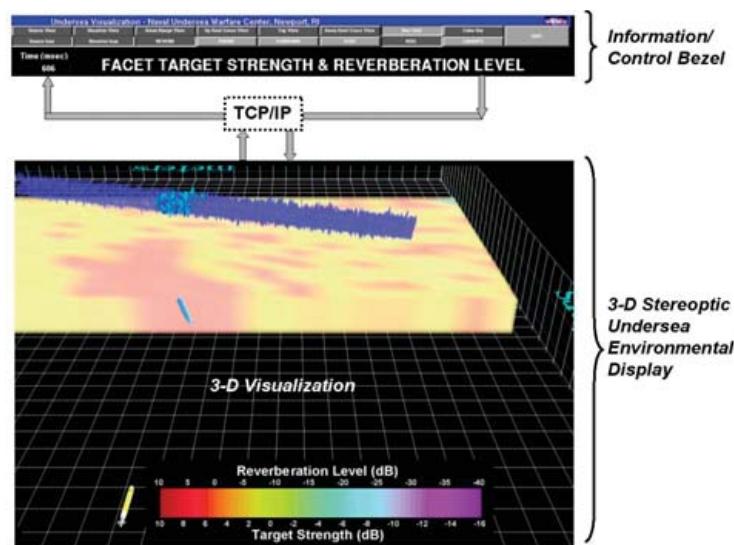


Figure 3. Baseline Undersea Environmental Visualization Architecture

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

Hardware Systems For The Open Source Community—Since 1989

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

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

AMD Opteron® Value Server-S795

1U 14.3" Deep
AMD Opteron 140 1M Cache
1 GB DDR ECC REG PC-3200
1 of 2 40GB SATA Drive
2 X 10/100/1000 NIC
Options: CD, FD, or Second Drive, Raid ADD Your Logo



iSCSI Dual AMD Opteron® 1U to 8U, Call for Pricing
1TB to 30TB of iSCSI Storage
Dual AMD Opteron 246
1 GB DDR ECC REG PC-3200
Dual GigE LAN
Redundant PS, Hot-Swap Drives
RAID Options, RAID 5, 10, 50
More Customization is available



1U SCSI Quad AMD Opteron® Starting @ \$2850

1of 4 AMD Opteron 848
2 GB DDR ECC REG PC-3200
1 of 3 36GB SCSI Drive
2 GigaE, CD, FD,
Remote Management Card (IPMI)



30TB AMD Opteron® Storage Solution- Starting @ \$26,395

30TB SATA Storage in 8U
Includes all Raid Cards, Raid 5, 10
Dual AMD Opteron 246
2 GB DDR, ECC REG PC-3200
Dual GigE, FD, CD



Your Custom Appliance Solution

Let us know your needs, we will get you a solution



Custom Server, Storage, Cluster, etc. Solutions

Please Contact us for all type of Storage solutions, NAS, DAS, iSCSI, Fiber RAID, SATA, SAS.

*Free shipping on selected servers and all notebooks.



2354 Calle Del Mundo, Santa Clara, CA 95054

www.asacomputers.com

Email: sales@asacomputers.com

P: 1-800-REAL-PCS | FAX: 408-654-2910

Prices and availability subject to change without notice.

Not responsible for typographical errors. All brand names and logos are trademark of their respective companies.

Listing 1.

pfb Conversion Code Snippets

```

Load Ice Keel Node and Store as pfb file

/**  read input file(s)          **/


i = 0;
group = pfNewGroup();
for (i = 0; i < num_files - 1; i++)
{
    printf("Make keel: %s\n", files[i]);
    bot_switch = (pfSwitch*)
        LoadKeel(files[i],limits,i);
    pfAddChild(group, bot_switch);
    printf("adding switch to group\n");
}
node = (pfNode *)group;

/** optimize input file (optional)      **/
node = optimize(node);

/*
 * write output file
 */
pfdStoreFile(node, files[num_files - 1]);

/*
 * indicate success
 */
return 0;
}

Convert Ice Keel to Performer Node

***** LOAD AND CREATE A 3D SURFACE *****/
pfSwitch *LoadKeel( const char *file_name,
                    float *limits, long numfile )
{
/* Declare Local Variables */
pfSwitch *root;
pfGroup *depth_group;
pfGroup *mag_group;
pfLOD *lod_ptr;
pfDCS *dcs12;
pfGeode *ice_geode;
pfCoord coord;
long lod_cols;
long lod_rows;
pfMaterial *material;
long i;
long j;
long status;

/* Create work space to create surface */
arena = pfGetSharedArena();

/* Load vertices, normals and colors */
status = load_data(file_name);
if( status != OK )
{
    exit (1);
}

/* Create the KEEL geode */
ice_geode = MakeKeel();

/* Create a group to hold all Depth and
 * Magnitude Features of Surface
 */
depth_group = pfNewGroup();
root = pfNewSwitch();

/* Add ice geode to group */
magflag = 0;
pfAddChild( depth_group, ice_geode );
dcs12 = pfNewDCS();
coord.xyz[PF_X] = 0.0f;
coord.xyz[PF_Y] = 0.0f;
coord.xyz[PF_Z] = 0.0f;
pfAddChild( dcs12, depth_group );
pfAddChild( dcs12, mag_group );
pfDCSScaleXYZ( dcs12, 1.0f, 1.0f, 1.0f );
pfAddChild( root, dcs12 );
pfDelete(dcs12);

/* Return 3D Surface Switch */
limits[0] = -1;
limits[1] = 1;
limits[2] = -1;
limits[3] = 1;
limits[4] = 0;
limits[5] = 0;
limits[6] = 0;
limits[7] = 1;
limits[8] = 1;
limits[9] = 1;

return(root);
}

```

and data prospecting will allow for a better understanding of the complex sound behavior in areas of interest to the Navy.

Undersea Environmental Visualization Software

Although the original software was written to run under the Silicon Graphics IRIX operating system, at the time of this writing, the new Undersea Environmental Visualization (UEV) version is compatible with and has been

tested and developed under Red Hat Linux 7.0 through 9.0. Linux was chosen as the operating system for several reasons. First, it is compatible with current and future submarine combat systems. Second, it is a generic UNIX operating system, which means software and script files developed under Linux can be transferred readily to UNIX operating systems such as HP-UX and IRIX. Third, it is an open-source operating system with a large user community that can be tapped for system optimization and maintenance.

The UEV system is composed of two main modules, the bezel and the main 3-D display application. These two modules communicate with each other by way of TCP/IP sockets. Figure 3 illustrates this architecture.

Separate modules were chosen for the display of the 2-D and 3-D data to allow separate viewing media to be used for each display, thus achieving the highest resolution for both. In its expanded form, the bezel also supports a 2-D overhead view. Still, this system is flexible enough to allow both displays to be shown simultaneously on a single screen, as shown in Figure 3. This simultaneous view does not support a 2-D overhead view, but it does support all the expanded version's functionality.

The bezel is a digital information and 3-D scene control program. The variables passed between the bezel and the main program include 3-D oceanographic/topographic maps, 3-D ice cover data, including ice keels, ice keel target strength data and 3-D sound propagation data, along with vehicle position data. The bezel for the UEV display was written using the XForms library. XForms is a GUI toolkit based on Xlib for the X Window System. It features a rich set of objects, such as buttons, scrollbars and menus, integrated into an easy and efficient object/event callback execution model that allows fast and easy construction of applications. In addition, the library is extensible and new objects easily can be created and added to it. Furthermore, Xforms was chosen for the prototype version of the UEV software because it is a stable and easy-to-use application programmers interface (API). In addition, absolutely no recoding is needed for operation under Linux.

Communication between the bezel and the main 3-D display happens by way of sockets that are established as datagrams in which messages sent over the network are self-contained packets delivered asynchronously between the two applications. This asynchronous form of communication was chosen because the data update rate between the two programs is slow enough that this primitive form of intra-program communication was sufficient. These links are primitive in their construction, requiring the specific IP address of the machines running the bezel and 3-D main application. Again, the reality, at least for research and development at Navy labs, is fast and inexpensive implementation is the driving force behind the creation of prototype software. This is so because software often doesn't advance past the prototype stage—the cost associated with programming elegance is a luxury.

However, a requirement for the follow-on UEV software is it must operate under Microsoft Windows as well as Linux. The Xlib version of XForms is no problem for Linux, but it is a big problem for Windows unless it is operated in the Cygwin environment. Although this is an option, the preference is for code that runs natively in both the Microsoft Visual C++ and Linux environments.

Our solution is the future conversion of the bezel to the Fast Light Tool Kit (FLTK), which will solve multiple problems. Because FLTK compiles under both Microsoft Visual C++ and Linux, the same software can be used for both systems. Second, the transfer of information between the bezel and main application can be converted from clunky TCP/IP sockets to a more elegant shared memory method. Finally, the bezel code can be brought into the 21st century by conversion of its XForms C routines to FLTK C++ methods. The conversion process currently is underway and is drawing in large part on the Open Inventor-based software that NUWC, Virginia Tech and the Naval Research Laboratory (NRL) jointly developed for the

TALOSS Project. As the system evolves to rely more and more on 3-D interaction with the 3-D environment, the bezel controls will become less important and may disappear entirely. Most likely, they will be replaced by a virtual toolchest and a gestural-based interface.

The 3-D UEV display receives its mapping and navigational information from an under-ice canopy database that is loaded at startup and updated based on the evolution of the acoustic situation. The under-ice canopy database consists of an ice volume of uniform depth with one or more embedded ice keels. The area of acoustic coverage determines the extent of the ice canopy.

All under-ice acoustic information is pre-rendered as OpenGL Performer binary (pfb) files. Construction of the pfb files begins with using Matlab

A graphic advertisement for TUX magazine. It features a yellow sticky note pinned to a blue background. The note has a red diagonal banner at the top that reads "Free Subscriptions!" in white. The note itself contains handwritten text:
Dear Bill,
It's over between us.
I've found someone new.
Someone I can depend on.
Someone who is fun for
a change. Thought you might
like to see his picture.
-Sandy

To the right of the note is a small photo of a penguin. At the bottom, the TUX logo is displayed, consisting of the letters "TUX" in a stylized font with a penguin icon integrated into the letter "U". Below the logo, the text reads:
The first and only magazine for the new Linux user.
Your digital subscription is absolutely free!
Sign up today at www.tuxmagazine.com/subscribe

Listing 2.

Code Snippet—Nodal Structure of Ice Facets/Volume Reverberation

```

Adding Target Strength & Reverb Nodes

/* Target Strength & Reverberation MAKING ROUTINES */

/* Load the target strength reflected keels */
nts = 0; /* Number of keel target strengths */
for (i = startpfb; i < stoppfb; i=i+5){

    sourcetime[nts] = i;

    /* Create load name for pfb file */
    sprintf(nts_name,"./PFB/out%d.pfb",i);

    /* Create Target Strength switch */
    ts_switch[nts]= pfNewSwitch();

    /* Load Target Strength pfb file */
    ts_group = pfdLoadFile(ts_name

    /* Create a coordinate node for
     * target strength */
    dcs6 = pfNewDCS();
    coord.xyz[PF_X] = -3300.0f;
    coord.xyz[PF_Y] = -650.0f;
    coord.xyz[PF_Z] = -125.0f;
    pfDCSCoord( dcs6, &coord );

    /* Add target strength node to
     * coordinate node */
    pfAddChild( dcs6, ts_group);

    /* Scale for good display */
    pfDCSScaleXYZ( dcs6, 0.201f,0.101f,0.2f);
    pfDCSScaleXYZ( dcs6, 5.0f,3.3f,15.0f);

    /* Add Target Strength in position node
     * to nodal switch */
    pfAddChild( ts_switch[ts], dcs6);

    /* Increment the Target Strength counter */
    nreflects = nreflects + 1;
}

/* Load the reverberation field corresponding
 * to TSkeels */
nreverbs = 0;
for (i = startpfb; i < stoppfb; i=i+5){

    /* Create load name for pfb file */
    sprintf(reverb_name,"./REVERB/reverb%d.pfb",i);

    /* Create Reverberation switch */
    reverb_switch[nreverbs]= pfNewSwitch();
}
/* Load Reverberation pfb file */
reverb_group = pfdLoadFile(reverb_name);

/* Stack the reverb planes to form a volume */
for(j=150; j<190; j++)
{
    /* Create a coordinate node for the reverb
     * plane */
    dcs6 = pfNewDCS();
    coord.xyz[PF_X] = -450.0f;
    coord.xyz[PF_Y] = 300.0f;
    coord.xyz[PF_Z] = j*-1.0f;
    pfDCSCoord( dcs6, &coord );

    /* Add target strength node to coordinate
     * node */
    pfAddChild( dcs6, reverb_group);

    /* Scale for good display */
    pfDCSScaleXYZ( dcs6, 3.5f,3.5f, 0.01f);
    pfDCSScaleXYZ( dcs6, 1.6f,4.3f,0.01f);

    /* Add Reverb plane node to nodal switch */
    pfAddChild( reverb_switch[nreverbs], dcs6);
}

/* Increment the Reverb counter */
nreverbs = nreverbs + 1;
}

***** TS and Reverb SWITCHES ADDED TO SCENE *****
for (k = 0; k < nreflects; k++)
{
    pfAddChild(ViewState->sceneGroup,
    reflect_switch[k]);
    ++loaded;
    pfSwitchVal(reflect_switch[k], PFSWITCH_OFF);
}

for (k = 0; k < nreverbs; k++)
{
    pfAddChild(ViewState->sceneGroup,
    reverb_switch[k]);
    ++loaded;
    pfSwitchVal(reverb_switch[k], PFSWITCH_OFF);
}

```

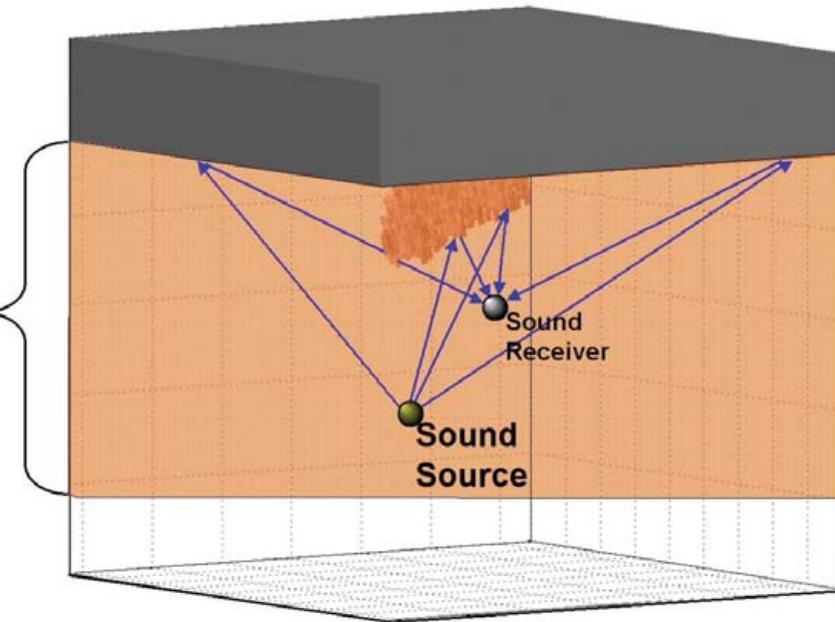


Figure 4. 3-D Sound Behavior

7.0.1 on a Linux platform. Matlab is a flexible interactive tool for doing numerical computations with matrices and vectors but is capable of displaying this information graphically, in both 2-D and 3-D forms. Therefore, by using a combination of Matlab and C-based transformation code, the under-ice information that comes out of a FORTAN-based model, developed by G. Bishop, is messaged into a form that is compatible with the OpenGL Performer-based 3-D UEV display.

The transformation starts with a Matlab routine that calculates all polygonal surfaces and their normals. It then outputs this information to the C-coded routines that convert the information to pfb file format. The pfb conversion is a modification of the Silicon Graphics utility pfConvert that is available for both IRIX and Linux. The code snippets shown in Listing 1 were added into pfConvert.c to read in the polygonal information generated by the Matlab code. The pfConvert routine then uses its own libraries to output the data to a Performer pfb file. The 3-D main application combines all tactical, navigation and acoustic information into a comprehensive 3-D picture. It renders the picture using the computer platform-independent scenegraph, OpenGL Performer. The use of OpenGL Performer was necessitated by the need for an efficient and cost-effective means of displaying both planar and volumetric data within the same 3-D display. OpenGL Performer provided the least labor-intensive means of achieving this integration, although open-source freeware, such as OpenSceneGraph, could provide an inexpensive alternative to OpenGL Performer in future releases of the software.

Understanding High Frequency Under-Ice Acoustic Scattering

Understanding the behavior produced by the scattering of sound energy in complex environments, such as under the Arctic Ocean pack ice, is an area of great interest to the US Navy and other navies. Insight into this complex acoustic environment is aided greatly by the simultaneous visualization of the in-water acoustic reverberation and the associated acoustic scattering from the ice keel.

Acoustic reverberation, which also could be called unwanted sound noise, simply is the re-echoing caused by sound bouncing off surfaces in all directions. In the case of the under-ice environment, these surfaces are the ice canopy and ice keels (Figure 4). Because sound energy bounces off objects three-dimensionally, it can be represented

GLOSSARY OF TERMS

Acoustics: *the physical properties of sound as they pertain to a particular situation.*

Acoustic scattering time interval: *the time interval in which a portion of the ice keel is being impacted by sound energy.*

Data prospecting or data mining: *the non-trivial extraction of implicit, previously unknown and potentially useful information from data.*

Ensonification: *impacting an object with sound energy.*

Ice canopy: *the ice cover formed by the ice pack.*

Ice pack: *a large expanse of floating ice.*

Pack ice: *ice making up the ice pack.*

Sonar: *the use of sound waves to detect underwater objects.*

Stereopsis: *3-D vision, binocular vision and stereopsis mean the same thing.*

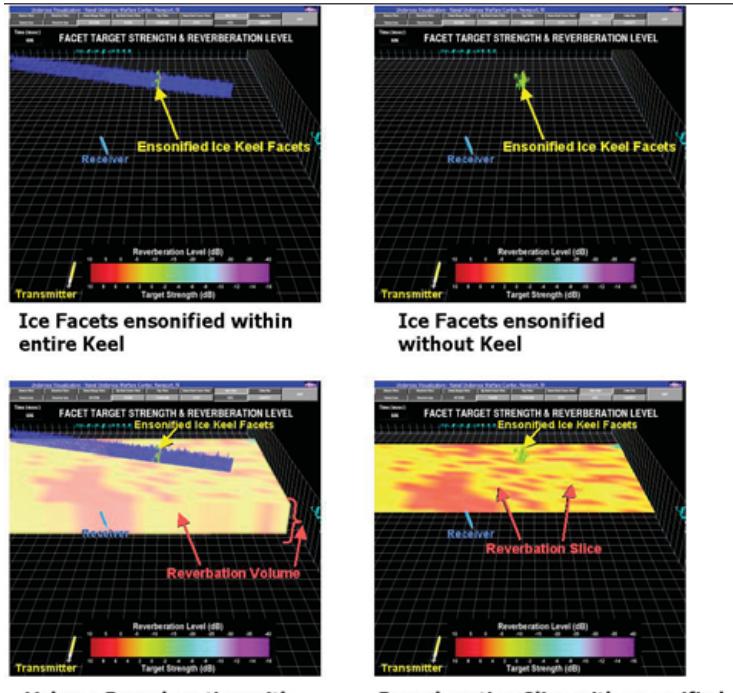


Figure 5. Graphic Illustration of Display Options Provided by Nodal Structure

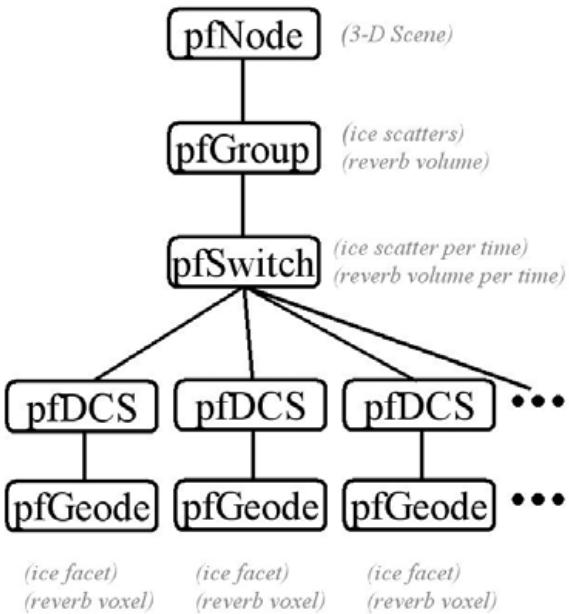


Figure 6. Sonar Visualization Scenegraph Structure

as a volume. This volume is referred to as the reverberant volume and can be represented in 3-D by individual volume elements, called voxels. Each voxel is color-coded to match the intensity level of the sound energy reaching it. Similarly, the intensity of the sound energy bouncing directly off the embedded ice blocks, called acoustic scattering, also is color-coded for intensity. The central idea behind the UEV

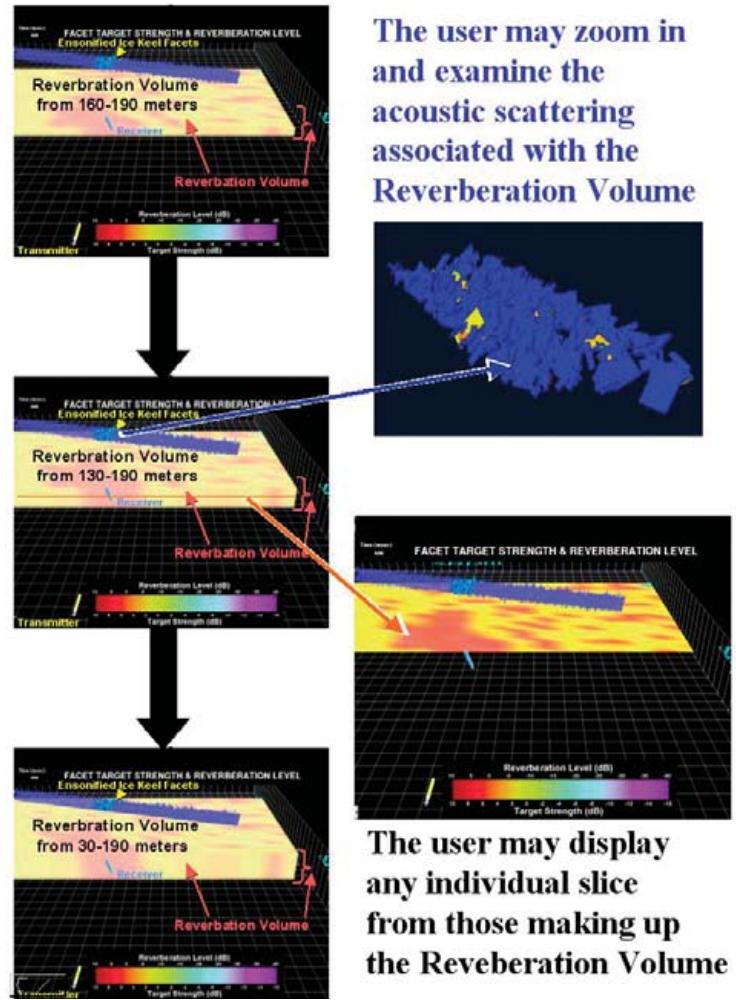


Figure 7. 3-D Manipulation of Ensonified Ice Facets and Volume Reverberation

software is to create an animated display that enables the user to interpret better the behavior of ice block scattering, as well as some of the space-time properties of the reverberant volume.

Displaying High-Frequency Under-Ice Acoustic Scattering

Individual nodes within the scenegraph represent the 3-D display of the information of interest, that is, the acoustic scattering from the surfaces of the ice blocks. Each block is formed as a six-faced polygonal surface, with each face colored to represent the target strength of the acoustic scattering from that face. To conserve memory and decrease rendering time, only those surfaces above a predetermined threshold are lit for any given acoustic scatter time interval. Turning the faces of the keel on or off is accomplished through the use of switches attached to each facial node. The reverberation associated with a given acoustic scatter interval is represented by a color-coded volume consisting of thin stacks of voxel volumes representing the reverberation for a given water depth. Again, these component reverberation volumes are addressed individually as nodes within the scenegraph. Figure 5 graphically illustrates the code snippets for the nodal structure of the ice facets and reverberation volume given in Listing 2. The advancement or regression of the acoustic scatter-reverberation display is controlled by the bezel. The display can be set to either continuous update

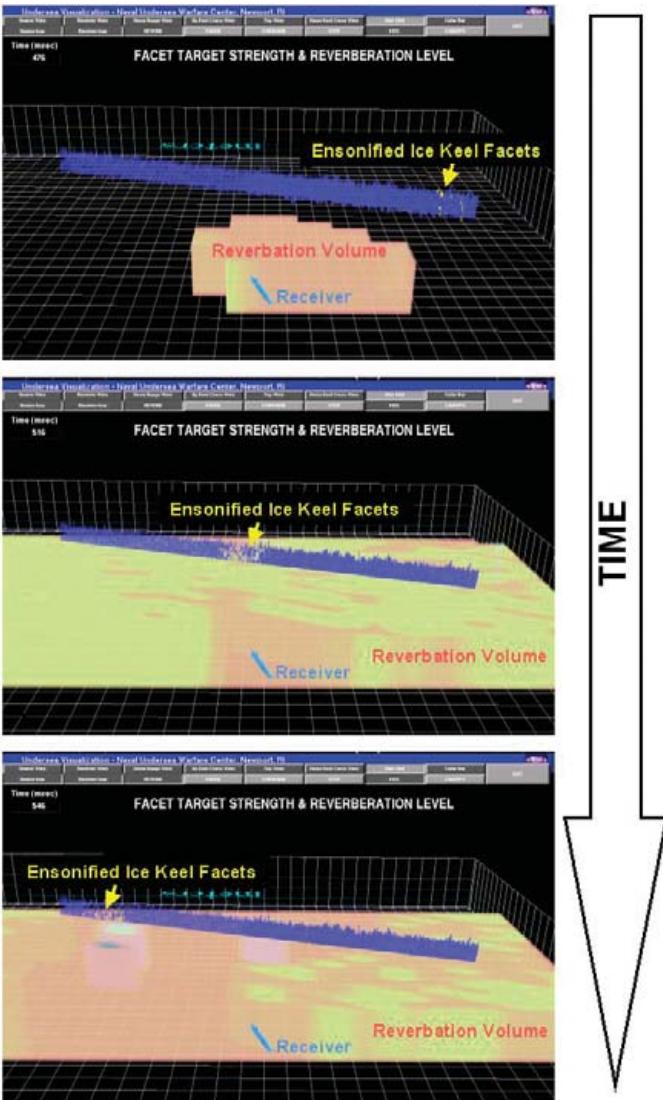


Figure 8. 3-D Progression of Sound Signal along the Ice Keel

mode or manual step mode. For any given time interval, the user can view any combination of ice block scattering and reverberation information, including blocks ensonified, lit within the entire ice keel; only the ice blocks of interest lit; the entire reverberation volume; or a user-selected depth slice of the reverberation volume.

The 3-D scene navigation also allows the user to zoom in and manipulate a particular region of the acoustic environment of interest. This data mining allows for a better understanding of what processes are occurring in a particular region. Figure 7 illustrates a zoomed view of the target strength for the region of an ensonified ice keel. Notice that the scene can be manipulated by the user and viewed from several different angles. Figure 7 also illustrates the building of the reverberant volume for the same acoustic scatter interval, as well as the selection of a single reverberation slice for display.

Finally, the user can observe the simultaneous evolution of both the target strengths of the faces of the blocks composing the ice keel and the accompanying water reverberation. Figure 8 illustrates the progression of the acoustic signal down a length of the ice keel and the accompanying reverberation at the receiver, in this case a submarine's acoustic system.

Advanced 3-D Scene Navigation Software Upgrades

The complexity of a data set such as ice keels lends itself perfectly to immersive visualization. The keel visualization environment displays sound energy in a volumetric form. Placing the user's viewpoint within that volume with a first-person perspective, rather than the current third-person perspective, should assist the observer.

Currently, the manipulation of the 3-D scene is dependent upon the standard keyboard and mouse. Planned upgrades to the under-ice visualization environment include a transition to an immersive display and 3-D gestural interface. The immersive display we have in mind is a head-mounted display. This design is portable and lends itself easily to use in tight quarters, such as those on a submarine.

When operating within a purely 3-D environment, the two-dimensional user interface becomes cumbersome. An elegant solution is to use a gestural interface in the form of a standard game pad with a sourceless inertial tracker attached. The object is to allow the user to position the viewpoint easily in any way. This functionality allows the user to better understand the complex visual representations of target strength and reverberation levels. Figure 9 demonstrates a prototype of the next step in the evolution of the UEV software.

Summary and Conclusions

A flexible, modularized, 3-D data fusion visualization system to study and data-mine under-ice acoustic information has widespread applicability for both military and civilian use. Building upon work originally sponsored by the Office of Naval Research (ONR) and years of Arctic experience, the Naval Undersea Warfare Center is developing pioneering visualizations for

LAYER 42™

- Redundant UPS and generator
- Nationwide network
- Free tech support

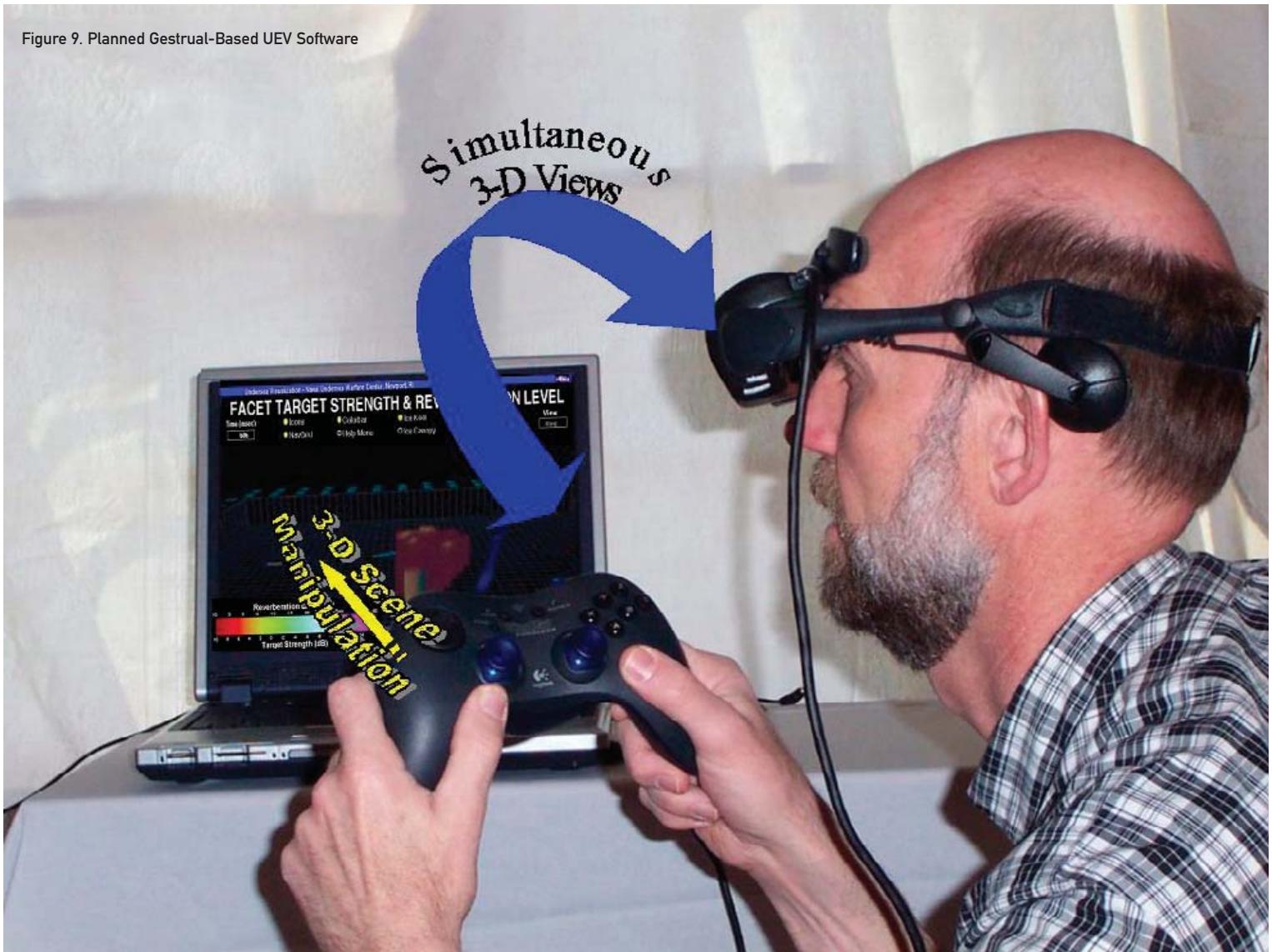
2U 256kbps ~80GB \$60/mo.	4U or Mid-tower 256kbps ~80GB \$80/mo.
1/4 Rack 512kbps (14U) ~165GB \$200/mo.	1/2 Rack 1mbps (28U) ~330GB \$350/mo.

www.layer42.net

All prices include 100Mbps port, Firewall,
24x7 Monitoring and DNS hosting

408-450-5740 2312 Walsh Ave., Santa Clara, CA 95051

Figure 9. Planned Gestural-Based UEV Software



the integrated understanding of the acoustic properties of complex ocean environments, such as those under the pack ice in the Arctic Ocean and in shallow water coastal regions throughout the world.

Because the Linux operating system is so adaptable and generic, it has been easy to transform the software originally written for Silicon Graphics machines to run now on Linux.

Richard R. Shell is an electrical/computer engineer at the Naval Undersea Warfare Center Division, Newport, Rhode Island. He is a co-recipient of the NUWC 2003 Special Achievement Center Award for Technology Innovation for his work as part of the team that developed the TALOSS 3-D submarine undersea battlespace visualization software featured in the November 2003 issue of *Linux Journal*.

Furthermore, because the US Navy operates in both the Linux and Microsoft Windows environments, Linux has provided a cost-effective means to update, compile and test the UEV software. We have found that if the software compiles under Linux without errors, it almost always compiles under Visual C++, as long as the laborious process of ensuring all the Microsoft Windows paths are set up properly.

Garner C. Bishop received his BS in mathematics from Bates College in 1968 and his MS and PhD degrees in theoretical physics from the University of Connecticut in 1976 and 1982, respectively. In 1983, he began employment with the Naval Undersea Warfare Center, and from 1994 to 1996 he was a Visiting Lecturer in underwater acoustics at the University of Massachusetts, Dartmouth. He is a member of the Acoustic Society of America.

Linux provides a unique environment for rapid prototyping, which results in software compatible with both UNIX and Microsoft Windows operating systems, thus qualifying as a valuable fleet software expeditor. ■

Resources for this article:
www.linuxjournal.com/article/8532

Douglas B. Maxwell is a mechanical engineer and research scientist at the NAVSEA Division Newport's Center for Advanced System Technology. His areas of expertise include design synthesis in virtual environments and synthetic training applications. He lives with his wife and dachshund in Newport, Rhode Island.

EIGHTH ANNUAL

OSCON

OPEN SOURCE CONVENTION

OPENING INNOVATION

OSCON 2006 will feature the projects, technologies, and skills that you need to write and deploy killer modern open source apps.



Linux

Perl Conference 10

PHP Conference 6

Python 14 Conference

Emerging Topics

Java

Ruby

Security

Databases

Business

JavaScript and Ajax

Web

Windows

REGISTER NOW AND SAVE 10%

(Use discount code os06linuxj)

conferences.oreilly.com/oscon

Securing OpenSSH

A tutorial on the holistic approach to securing OpenSSH. MATTHEW E. HOSKINS

If you are a systems administrator of one or more Linux/UNIX systems, OpenSSH is likely one of the most important tools in your toolbox. OpenSSH can be used interactively or embedded in scripts to provide secure remote access and file transfer between systems. But, alas, danger lies ahead. With OpenSSH, it is perilously easy to create sloppy trust relationships or, worse, leave yourself open to common automated attacks.

Basic Security

One of the basic tenets of system security has always been to run only the minimum required services and limit their access only to those who need it. Linux systems make this pretty easy to do, and like most things in the *nix world, you can do it in a number of ways. If you're still reading this article, we assume you need to run OpenSSH. Let's work on limiting access.

On most Linux distributions, you have the choice of handling this at the kernel firewall level or using TCP Wrappers (/etc/hosts.{allow|deny} files).

iptables

A simple iptables firewall rule to limit OpenSSH only to your local subnet could be:

```
iptables -A INPUT -s ! 192.168.0.0/255.255.0.0
→ -p tcp -m tcp --dport 22 -j REJECT --reject-with
→ icmp-port-unreachable
```

This tells the kernel to reject any TCP/IP packets not coming from a specific subnet aimed at port 22. (Substitute your own numbers as needed.) In all likelihood, you will have additional iptables rules to protect other applications on this system, so integrate the above into your overall firewall design.

TCP Wrappers

TCP Wrappers is a common feature on most Linux distributions, and OpenSSH has built-in support. To implement the same rule above using TCP Wrappers, add the following line to your /etc/hosts.deny file:

```
sshd: ALL EXCEPT 192.168.0.0/255.255.0.0
```

For most situations, it is very unlikely that OpenSSH needs to be dangling out for the world-wide Internet to poke at. Further, it is unlikely that OpenSSH even needs to be available to your entire organization. I highly recommend that access be limited, using either of the above methods, to the smallest audience possible. One common method is to designate one or more systems specifically for the purpose of centralized administration. Configure the rest of your systems to accept OpenSSH connections only from these dedicated and highly secured systems.

To test whether you have limited access to OpenSSH successfully, you can try to connect using an ssh client, or simply connect to the system using a command-line telnet client, for example:

```
$ telnet mylinuxbox 22
```

where mylinuxbox is the hostname, and 22 is the port.

If OpenSSH is responding and open, you should see something like the following:

```
SSH-1.99-OpenSSH_3.9p1
```

If it is successfully blocked, the connection will be refused. Try this from multiple locations, and confirm that connections are blocked as you expect.

Avoiding Automated Attacks

Over the years, OpenSSH has had a few remotely exploitable vulnerabilities due to bugs. These always have been fixed quickly, with updated versions distributed promptly. Unfortunately, a lot of networked systems are not updated. Also, many systems simply are poorly secured. Standard accounts with simple or blank passwords are depressingly commonplace.

All of these are tantalizing targets for crackers and worm authors. A number of malicious worms exist that scan for any SSH servers and then try common vulnerabilities and passwords. If you must have your OpenSSH port open to the world, consider changing its TCP port number. This can be accomplished by changing the Port directive in the sshd_config file, which is commonly in /etc/ssh/. Although any live person trying to crack your systems would likely have little difficulty detecting this, it is usually enough to escape the attention of large-scale automated attacks and worms.

Also, you might consider rebuilding your OpenSSH from sources after modifying the version string. This also can help confuse automated attacks, because many vulnerabilities are dependent on specific versions of OpenSSH. This is done by modifying the version.h file in the OpenSSH distribution and recompiling. Change the SSH_VERSION and SSH_PORTABLE #define lines to anything you like, such as:

```
#define SSH_VERSION      "MyCompany_Vers00001"
#define SSH_PORTABLE      "foo"
```

After changing the sources, build and install. This changes the version banner shown in the above telnet test.

Changing the port and version string does *nothing* to improve security; they simply reduce the risk of being the victim of a common exploit or worm. There is no substitute for securing accounts and making sure your systems are running the latest patch levels of software. If you are using Linux, use a current distribution. Use automated update tools like yum, apt or up2date to keep your system running the latest version of all software and libraries including OpenSSH.

You can confirm that users are selecting good passwords by using a cracking tool like John the Ripper (see the on-line Resources). John the Ripper (JTR) uses the one-way encrypted passwords in /etc/shadow to try to crack passwords. Generally speaking, easier passwords cracked by JTR are more likely to show up in a brute-force dictionary attack commonly used by some worms. JTR supports a wide range of input file formats for those using LDAP, NIS, AFS Kerberos and other authentication services.

Avoid Using Passwords

If possible, stop using passwords for remote logins. You can accomplish this by setting up trust relationships using public key authentication. These trust relationships can be set up between any two user accounts, on the same system or across the network.

The first step to setting up one of these relationships is to create a private and public key, known as a key pair. Each key is stored in a separate file. By convention, the filenames are the same, but the public key file has an extra .pub extension. The private key stays secret and usually remains in the home directory of the account that will initiate the connection.

OpenSSH will refuse to use any private key file with lax file permissions. To form the trust relationship, the public key is appended to the ~/.ssh/authorized_keys file (usual location on most Linux builds of OpenSSH) on the destination side of the trust relationship. The authorized_keys file contains all the public keys trusted by the owner's account. These keys can be secured further by providing a passphrase, which is used to encrypt the private key. If a passphrase is specified, it needs to be provided before the private key can be used to make a connection. In the example below, we create a trust relationship between two accounts, account1 on a system named localbox and account2 on a system named remotebox.

Create account1's key pair (we will specify a passphrase when asked). The passphrase is not actually displayed as you type it, but it is shown below for demonstration. We specify that the pair should be based on a 2048-bit private key (the default is 1024, which recently has been proven to be much easier to crack):

```
[account1@localbox account1]$ ssh-keygen -t rsa -b 2048
Generating public/private rsa key pair.
Enter file in which to save the key (/home/account1/.ssh/id_rsa):
<ENTER>
Created directory '/home/account1/.ssh'.
Enter passphrase (empty for no passphrase):
This%%Is%%%My%%%Passphrase<ENTER>
Enter same passphrase again:
This%%Is%%%My%%%Passphrase<ENTER>
Your identification has been saved in /home/account1/.ssh/id_rsa.
Your public key has been saved in /home/account1/.ssh/id_rsa.pub.
The key fingerprint is:
63:9f:75:d5:80:93:67:5e:d8:69:5a:69:69:4f:a0:76 account1@localbox
```

```
[account1@localbox account1]$ cat /home/account1/.ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,9668032A1B3D23DA

2IKH5xt+LR2/nftd8aPi1KPZ4Fn20upJKU1NUAK82HXBb59TPyzmr/05i56ZiPh0
UHJCScF+G/k44zheUnyWu/+65iVuzTaa+6q4eYaTna8ElA+PemfwQF08L780Nku
+2VJeEiyzXcro9J6oYQGhs+qWfHalmbX6w6IwuAh6jjEhCn+LT1Zs1yjGFu5NHDr
sWSMNgounPS/ay8ZlNIV+iQB5l/v+bNWic92ICYbZW0C+o2XKpNvACdUDpA/gfqk
b20utNozBfGAggmU2EJ5HGq8YQMCjVk0ZN/NVjGqUuACQmW0nB5jYrGE/ka/20xN
X4jZwxXGxNeS0PMdx1B5XXXXXXXXBmCxw0AZ0SVYdoSbapZJKMsOdEo1FU4IFEewm
NlgFXfe7vsdFseFsaSdfEkChaHaMadeYouLooktoqz+mfkbwDpC8msnKcuRJShR9
09rZnoKW/ddvrJv5oIQYFqu7LYJr5ygpW3T3oiFgzsIg//1hAW4h0sTCBWUIGSkk
IWTxKFooQ3ABarsIzChqPBazXnSJ5IVXXXlglmH9ce5Z6YMN0UdfWBn4Mgc2gNB
xCK4KINXMJ7MjsmLxkwuAbqcTq4SBCiopr6PLL5G2KK6ljXLdJe4oLDdb/GtxlH
m0XNuAwLD1eE5FA0USQLGDkeUThisIsNotARealPrivateKeyc0BrhdykHI5Sh7
/hVytVr9t6r6lRBEBsMi1T+P2uBAuahNGCyDs+5F1P14y2RJ5GAldVdM8bXOfnfso
pHiXerStpKw6j85wxoKryzSAGs2eWXtrQKk8IMkDQug=
-----END RSA PRIVATE KEY-----
```

```
[account1@localbox account1]$ cat /home/account1/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAIAE54Q3QZLkpRaNN83WUMBGBE+/xZ88ud
+S1M74awZa8P6H3TNCEnr3Lt158bsgjl3z0W5DXWvNRgszoQeBJYNVHlsVIUCp0Vj
Vr7aDun0PiUVI2y658XB1bup5+isgMNuyB7BskVKs9aWYQNCbJVNo+3wcVA9x+De5
Z7yBapi6E= account1@localbox
```

This created the two key files /home/account1/.ssh/id_rsa and /home/account1/.ssh/id_rsa.pub. Now, we copy the public key to account2@remotebox. You can do this with any method you find convenient. We use scp to copy the file, and then use cat to append to it:

```
[account1@localbox account1]$ scp /home/account1/.ssh/id_rsa.pub
->account2@remotebox:./account1.id_rsa.pub
account2@remotebox's password:
id_rsa.pub          100%   243    538.1KB/s   00:00
```

Now we append account1's public key to account2's authorized_key file and set proper permissions on files:

```
[account2@remotebox account2]$ mkdir ~/.ssh
[account2@remotebox account2]$ chmod 700 ~/.ssh
[account2@remotebox account2]$ cat account1.id_rsa.pub >>
->~/.ssh/authorized_keys
[account2@remotebox account2]$ chmod 600 ~/.ssh/authorized_keys
[account2@remotebox account2]$ rm account1.id_rsa.pub
[account2@remotebox account2]$ ls -l ~/.ssh/
-rw----- 1 account2 account2 243 Dec  9 09:42 authorized_keys
```

By placing the public key on the destination account2, we prove the identity of account1 when it tries to connect. As you can see below, we are prompted for the passphrase when we make the connection:

```
[account1@localbox account1]$ ssh account2@remotebox
Enter passphrase for key '/home/account1/.ssh/id_rsa':
This%%Is%%%My%%%Passphrase<ENTER>
[account2@remotebox account2]$
```

After the basic trust relationship is set up, you can disable password authentication by locking the password using the passwd command and the -l option.

You can restrict this trust further by adding options to the authorized_keys file. In the example below, we include the from= option to add host-based checking, so the private key will be accepted only from predefined network addresses. We also add the command= option, which specifies the one and only command that will be executed after authentication to the remote account. This feature is most useful from inside scripts to create special-purpose trust relationships to automate tasks between systems or accounts:

```
[account2@kettle account2]$ cat .ssh/authorized_keys
from="localbox",command="/bin/df -k" ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAIAE54Q3QZLkpRaNN83WUMBGBE+
/xZ88ud+S1M74awZa8P6H3TNCEnr3Lt158bsgjl3z0W5DXWvNRgszoQeBJYNVHlsVIUCp0Vj
Vr7aDun0PiUVI2y658XB1bup5+isgMNuyB7BskVKs9aWYQNCbJVNo+3wcVA9x+De5Z7yBapi6E= account1@localbox
```

```
[account1@localbox account1]$ ssh account2@remotebox
Enter passphrase for key '/home/account1/.ssh/id_rsa':
This%%Is%%%My%%%Passphrase<ENTER>
```

```
10:32:15 up 8 days, 15:13, 16 users, load average: 0.15, 0.12, 0.09
Connection to remotebox closed.
[account1@localbox account1]$
```

After the uptime command is executed, we are immediately dropped back to account1. This relationship is now locked in to performing only one operation. This could be a single command, script or a restricted shell. The command= option can be used to allow certain users to perform selected elevated privilege operations in a controlled manner. Similar functionality can be gained using tools like sudo, but OpenSSH allows these operations over the network, which is useful in some circumstances. If you have problems with any of these examples, you can run ssh with debugging turned on, using the -vvv command-line option. This usually helps in figuring out what is going wrong; most commonly, file permissions cause OpenSSH to ignore key files.

To facilitate the use of public key authentication in scripts, you can create a key pair for each remote operation and then specify which one to use by invoking ssh with the -i option and the filename of the private key.

For interactive sessions, you can use a program called ssh-agent to cache your decrypted private keys in memory, so you need to provide the passphrase only once. On some Linux distributions, ssh-agent is already hooked into the login process to start automatically. If so, all you have to do is use ssh-add to cache your private keys. If you are interested in ssh-agent, take a look at the ssh-agent and ssh-add man pages for detailed explanation of their uses.

One final comment about public key authentication: with the price and ease of use of modern USB Flash drives, you can consider keeping your passphrase-protected private keys on removable media rather than in your home directory. This lets you use them from multiple locations while maintaining only one copy.

OpenSSH Options

OpenSSH has a number of configuration options that can be employed to reduce risk. Most Linux distributions come with good middle-of-the-road settings. In some cases, these can be adjusted for better security. These include:

- Logging: turn up the logging level to INFO or DEBUG. Examine your logs on a regular basis, or better yet, implement one of the many available log analysis tools to notify you of anomalies:

```
LogLevel INFO
```

- Privilege separation: make sure the UsePrivilegeSeparation option is enabled:

```
UsePrivilegeSeparation yes
```

- Protocol: limit OpenSSH to accept only protocol version 2:

```
Protocol 2
```

- Root logins: prevent root logins using passwords. This limits root access only to nonpassword methods like public key authentication. If you never directly log in as root, set this to no. You always can log in as a regular user and su to root or use sudo. Setting this

to without-password allows only nonpassword methods, such as public key authentication:

```
PermitRootLogin without-password
```

- File permissions: make sure StrictModes is enabled to prevent the use of insecure home directory and key file permissions:

```
StrictModes yes
```

- Reverse name checking: require OpenSSH to check proper reverse hostname lookups. Note that you must have proper name lookups working (that is, DNS or /etc/hosts):

```
VerifyReverseMapping yes
```

- Prevent port forwarding: these two options prevent OpenSSH from setting up TCP port and X11 forwarding; if you do not need these features, disable them:

```
AllowTcpForwarding no
X11Forwarding no
```

- Disable all host-based authentication: always make sure these are disabled. These methods assume that the network can be trusted and allow .rhosts-style authentication based on hostname or IP. Never use these methods as primary authentication:

```
IgnoreRhosts yes
HostbasedAuthentication no
RhostsAuthentication no
RhostsRSAAuthentication no
```

Conclusion

Using the above suggestions, you will be able to tighten access controls and eliminate sloppy trust relationships. As I mentioned earlier, staying current with updates or patches is critical to system security. In order to stay current, you need to be informed of when updates are released, so I suggest a multipronged approach. First, automate apt, yum or up2date to check nightly and report on missing updates. Second, subscribe to your Linux distribution's security mailing lists. Third, subscribe to one of the many security discussion groups, such as SecurityFocus. If you build OpenSSH from sources, join the OpenSSH mailing list to watch for updates.

System security requires a holistic approach. The methodologies provided in this article form the basis for securing OpenSSH, one small component of a modern complex Linux system.

After you believe you have secured your system, use scanning tools like the free Nessus Vulnerability Scanner (see Resources) and peer review from colleagues to check your work. ■

Resources for this article: www.linuxjournal.com/article/9023.

Matthew E. Hoskins is a Senior UNIX System Administrator for The New Jersey Institute of Technology where he maintains many of the corporate administrative systems. He enjoys trying to get wildly different systems and software working together, usually with a thin layer of Perl (locally known as "MattGlue"). When not hacking systems, he often can be found hacking in the kitchen. Matt is a member of the Society of Professional Journalists and can be reached at matt@njit.edu.

Growing a World of Linux Professionals



We at the Linux Professional Institute believe the best way to spread the adoption of Linux and Open Source software is to grow a world wide supply of talented, qualified and accredited IT professionals.

We realize the importance of providing a global standard of measurement. To assist in this effort, we are launching a Regional Enablement Initiative to ensure we understand, nurture and support the needs of the enterprise, governments, educational institutions and individual contributors around the globe.

We can only achieve this through a network of local "on the ground" partner organizations. Partners who know the sector and understand the needs of the IT work force. Through this active policy of Regional Enablement we are seeking local partners and assisting them in their efforts to promote Linux and open source professionalism.

We encourage you to contact our new regional partners listed above.

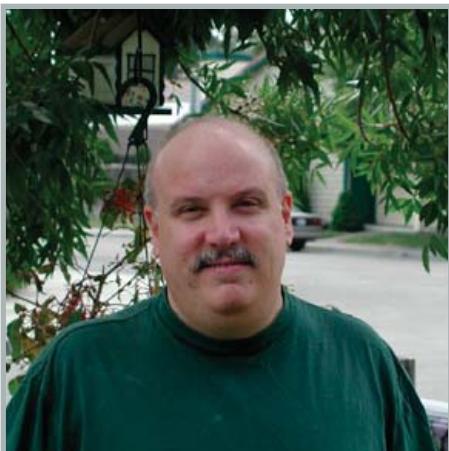
Together we are growing a world of Linux Professionals.



Stable. Innovative. Growing.

It's free. It's proprietary. No, it's two (click) two (click) two distros in one.

Freespire is Linspire's new inspired response to the factors that are holding back the best desktop distribution around.



Nick Petreley, Editor in Chief

This column has a new name, /var/opinion. I changed it for three reasons. First, as much as I enjoy a good rant (and I will continue to rant whenever the mood strikes), I've found too much to rave about lately. I find it constraining to produce a rant when I get excited about something good. Second, I named it /var/opinion because a column is, by definition, an opinion. We editors and writers take that for granted, but not all readers are as savvy about publishing practices. Some readers express dissatisfaction when someone expresses an opinion in a column. I'm not finding fault with these readers for doing so, because they simply may not know better. The title is meant to remind them that an editorial column is opinion. The two are inseparable. Finally, as many readers were so kind to point out, /etc/rant was not LSB-compliant. So I changed the location of the column to /var.

Linspire recently announced that it will ship a community-driven distribution called Freespire sometime in August 2006. When it gets here, I'll decide if it's worthy of a rant or a rave. But

the idea is spot on, and worthy of a rave.

Let's get the self-serving nature of this move out of the way. Linspire may grab a lot of pieces from Debian and other sources in order to produce the Linspire distribution, but the company still has to do a lot of work fine-tuning the distribution to make it easier to use than the competition. Linspire is attempting to off-load that work to the volunteer community. It sounds like Linspire is exploiting volunteers, but if so, Linspire is just joining a very big club. Red Hat created Fedora Core 5 in part to off-load the development of the core distribution to volunteers. Countless Debian-based distributions exploit the hard work of Debian volunteers. So, unless you want to spread the blame across just about every distribution available, Linspire doesn't deserve any criticism for doing the same.

Let me squash another probable criticism. Is Freespire Linspire's response to the threat of an Ubuntu/Kubuntu revolution? Probably. But again, what difference does that make if we Linux users benefit?

Here's the cool thing about Freespire. The Linspire folks are adopting some of the most significant improvements you'll find in other distributions, such as Ubuntu. For example, you will be forced to create a user account and will no longer log in as root by default.

More important, the plan is to make it customizable. Those who are anal about licensing issues can have a completely open-source version of Freespire. Those who want to use Java, Flash, play DVDs or do anything else that requires an alternative license or proprietary software can get Freespire with support for these additions.

Maybe you're a license freak, in which case you don't use Java, Flash or play DVDs on Linux. Sorry, but I'm not only unafraid of most licenses (some are unacceptable, but not most), I actually use and pay for proprietary software. For example, there's a new proprietary graphics program out there called Pixel, and it looks

fantastic. It's almost like an incredibly inexpensive Photoshop clone. It is far more intuitive and friendly than GIMP. So, to me, it's worth every penny. Visit www.kanzelsberger.com/pixel for more information, and don't be a cheapskate if you like what you see.

I also want Flash support for my browser, I want to play DVDs, and I can't live without Java. Aside from applications like Jedit and Eclipse, my favorite Java applet is the doppler radar applet you can find on the National Weather Service site (NOAA). Just visit www.srh.noaa.gov and select your location, click on the radar map and then select one of the radar types (composite reflectivity or base reflectivity, among others). I went tornado chasing while I lived in Missouri, and I found this feature to be invaluable. By the way, I'm told NOAA has adopted Linux as its standard platform.

Back to Freespire. If Freespire attracts a community of developers, it could easily turn out to be my distribution of choice. Why? I think Linspire is by far the best desktop distribution available. Linspire blows away every other distribution in terms of ease of use.

In spite of all this praise, I don't use Linspire on a daily basis. One reason I don't use it is because Linspire takes so long to stay up to date with KDE and hardware support. It always has an older version of KDE, and it doesn't even install on my latest dual-core AMD64 machine.

Just as the Fedora developers have been able to keep Fedora more up to date than Red Hat was able to do internally, perhaps the Freespire developers will be able to keep Linspire current, and therefore more attractive to people like me. And, maybe even you. Keep an eye on Freespire. I will.■

Nicholas Petreley is Editor in Chief of *Linux Journal* and a former programmer, teacher, analyst and consultant who has been working with and writing about Linux for more than ten years.



Rackspace – Managed Hosting Backed by Fanatical Support™

Fast servers, secure data centers and maximum bandwidth are all well and good. In fact, we invest a lot of money in them every year. But we believe hosting enterprise class web sites and web applications takes more than technology. It takes Fanatical Support.

Fanatical Support isn't a clever slogan, but the day to day reality our customers experience working with us. It's how we have reimagined customer service to bring unprecedented responsiveness and value to everything we do for our customers. It starts the first time you talk with us. And it never ends.

Contact us to see how Fanatical Support works for you.

1.888.571.8976 or visit www.rackspace.com



Thanks for honoring us with the
2005 Linux Journal Readers' Choice Award for
"Favorite Web-Hosting Service"



The World's Fastest InfiniBand Switch

From a Company You've Trusted for 23 Years

Microway's **FasTree™** DDR InfiniBand switches run at 5GHz, twice as fast as the competition's SDR models. FasTree's non-blocking, flow-through architecture makes it possible to create 24 to 72 port modular fabrics which have lower latency than monolithic switches. They aggregate data modulo 24 instead of 12, improving nearest neighbor latency in fine grain problems and doubling the size of the largest three hop fat tree that can be built, from 288 to 576 ports. Larger fabrics can be created linking 576 port domains together.



▲ 72 Port FasTree™ Configuration

Working with PathScale's InfiniPath HTX Adapters, the number of hops required to move MPI messages between nodes is reduced, improving latency. The modular design makes them useful for SDR, DDR and future QDR InfiniBand fabrics, greatly extending their useful life. Please send email to fastree@microway.com to request our white paper entitled *Low Latency Modular Switches for InfiniBand*.

Microway's **QuadPuter®** includes four AMD single or dual core Opteron™ processors, 1350 Watt redundant power supply, and up to 5 redundant, hot swap hard drives-all in 4U. Dual core enables users to increase computing capacity without increasing power requirements, thereby providing the best performance per watt. Constructed with stainless steel, QuadPuter's RuggedRack™ architecture is designed to keep the processors and memory running cool and efficiently. Hard drives are cooled with external air and are front-mounted along with the power supply for easy access and removal. The RuggedRack™ is also available with an 8-way motherboard and up to 128 GB of memory for power- and memory-hungry SMP applications.



Call us first at 508-746-7341 for quotes and benchmarking services. Find technical information, testimonials, and newsletter at microway.com.

◀ QuadPuter® Navion™ with hot swap, redundant power and hard drives and dual core Opterons, offering the perfect balance between performance and density



508.746.7341 microway.com