

LINUX JOURNAL

Since 1994: The Original Magazine of the Linux Community

NOVEMBER 2008 | ISSUE 175

ROADRUNNER'S
COMPUTING POWER

Supercomputing
at Your Fingertips
with CUDA

INCREASE
PERFORMANCE
WITH
SOFTWARE RAID

DEBUGGING
PROGRAMS
ON A **CELL
PROCESSOR**

INTERVIEW WITH

CORY DOCTOROW



Using Python
for Scientific
Computing

Simplifying GPU
and Accelerator
Programming

REVIEWED:

- » NolaPro
- » Popcorn Hour A-100

www.linuxjournal.com

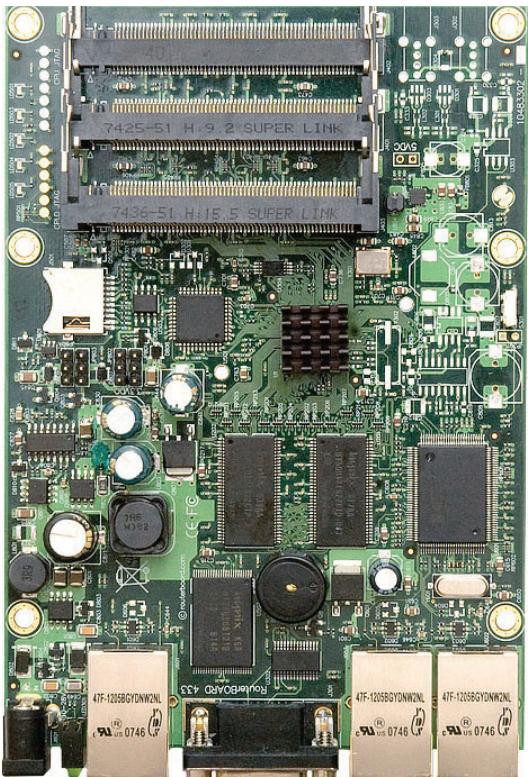
\$5.99US \$5.99CAN

11>



RouterBOARD 433

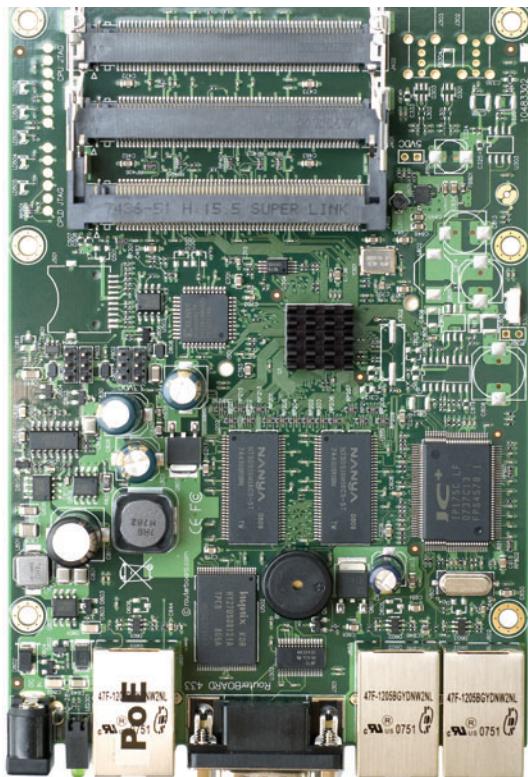
680MHz MIPS (overclock to 800MHz)



RB433AH

\$149

300Mhz



RB433

\$99

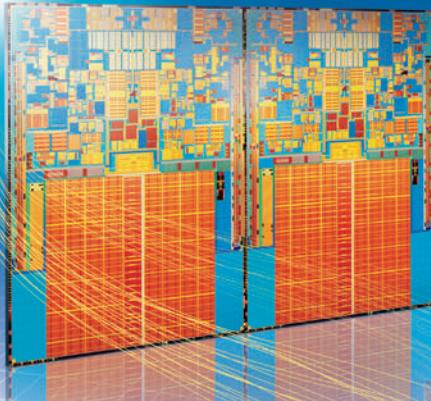
The RB433 is a high speed AP/router. It has a new generation 300Mhz Atheros CPU and 64MB of RAM. It is provided with three 10/100Mbit Ethernet ports and three miniPCI slots.

The RB433AH is a faster, enhanced version of the RB433 with a 680Mhz CPU, 128MB RAM and a microSD card slot for an additional memory card. It supports overclocking to 800Mhz, pushing the price/performance ratio to new standards.

These devices are preinstalled with the MikroTik RouterOS Firewall/QoS/Routing operating system. RB433 comes with a L4 license, and RB433AH with L5



YOUR MACHINES ARE VIRTUAL, YOUR GREAT PERFORMANCE IS REAL.



High-performance virtualization.
That's IT as it should be.

Intel® Xeon® with hardware-based virtualization technology allows you to load up to 2.5 times more virtual machines.¹

ACE POWERWORKS 1990

- Dual Quad-Core Intel® Xeon® processors
- Large Memory Footprint—16 DIMM Slots
- Outstanding 1600 MHz FSB Performance
- 1U Rack Optimized for HPC Applications



ACE POWERWORKS 2995

- Dual Quad-Core Intel® Xeon® processors
- High Memory Footprint—Up to 128GB
- 7 I/O Slots, 8 Hot Swap SAS/SATA Bays
- Excellent for Server Consolidation and Virtualization



25th Anniversary



www.acecomputers.com
1425 East Algonquin Rd
Arlington Heights, IL 60005
877-ACE-COMP (877-223-2667)

Celebrating 25 years of Vision™

Ace Computers is ranked as a "Top 10 Custom System Builder" by CRN™ magazine and is also rated as a top 5 Custom Server and Storage integrator and a member of the VARBusiness™ 500. Ace is an Intel® Premier Channel Partner and a Microsoft® Gold Certified Partner. Our unique position allows us to tailor a complete solution to YOUR challenges and become a true partner. Use what the US government, major universities and corporations do, Ace® Powerworks™ Servers.

Intel is not responsible for and has not verified any statements or computer system product-specific claims contained herein.



1. Intel measured results as of Aug 22, 07. Performance comparisons made using internal virtualization workload on Xeon 5300 against Xeon 7300. Intel VT requires computer system with a processor, chipset, BIOS, virtual machine monitor (VMM) and applications enabled for virtualization technology. Performance, functionality or other virtualization technology benefits will vary.

© 2008, Intel Corporation. All rights reserved. Intel, the Intel logo, Intel Xeon, and Xeon Inside are trademarks of Intel Corporation in the U.S. and other countries.

*Other names and brands may be claimed as the property of others.

CONTENTS

NOVEMBER 2008 Issue 175

FEATURES

56 THE ROADRUNNER

SUPERCOMPUTER:

A PETAFLOP'S

NO PROBLEM

IBM and Los Alamos

National Lab teamed up to build the world's fastest supercomputer.

James Gray

62 MASSIVELY PARALLEL

LINUX LAPTOPS,

WORKSTATIONS AND

CLUSTERS WITH CUDA

Unleash the GPU within!

Robert Farber

68 INCREASE

PERFORMANCE,

RELIABILITY

AND CAPACITY WITH

SOFTWARE RAID

Put those extra hard drives to work.

Will Reese

74 OVERCOMING THE

CHALLENGES OF

DEVELOPING

APPLICATIONS FOR

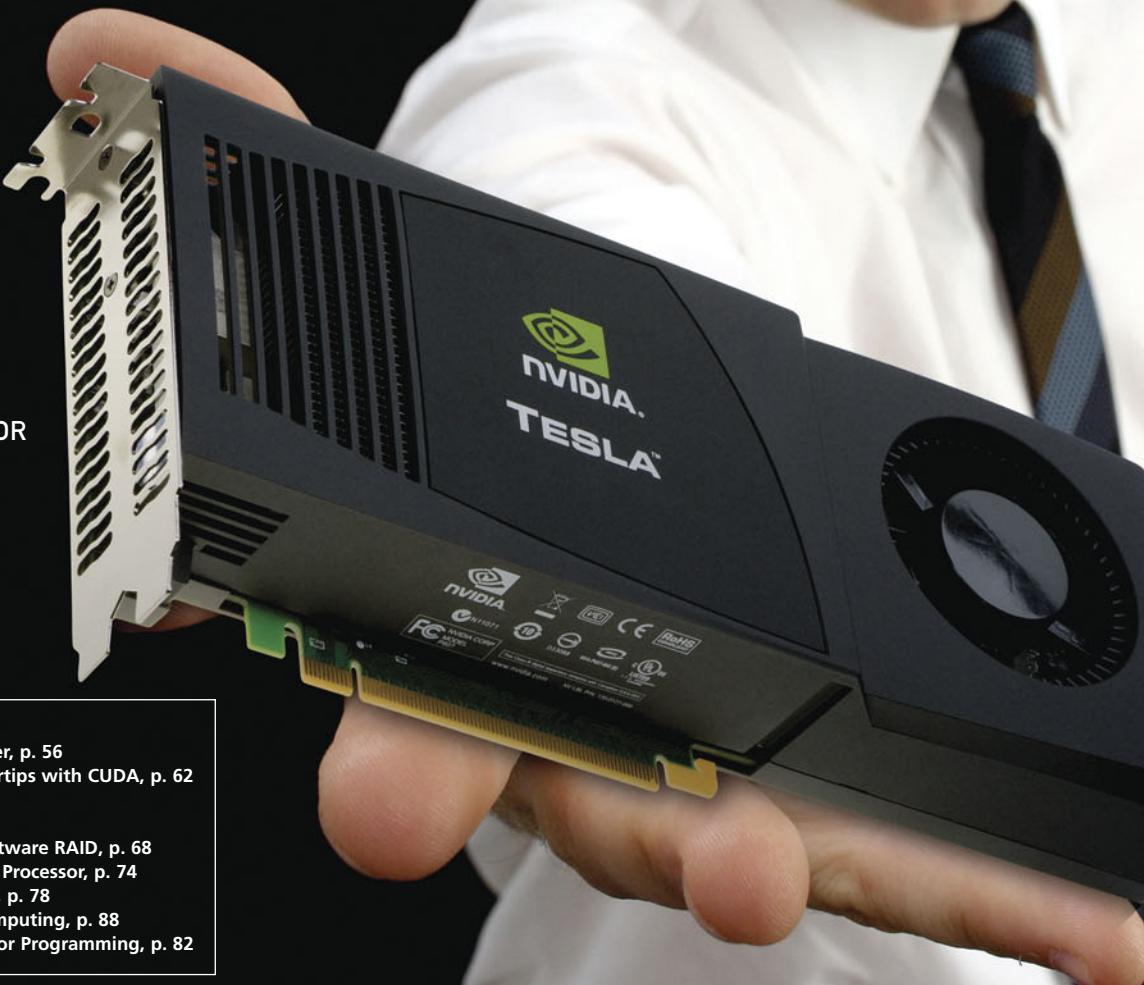
THE CELL PROCESSOR

Introducing techniques for troubleshooting programs written for the Cell processor.

Chris Gottbrath

ON THE COVER

- Roadrunner's Computing Power, p. 56
- Supercomputing at Your Fingertips with CUDA, p. 62
- Reviewed: NolaPro, p. 46
- Reviewed: Popcorn Hour, p. 50
- Increase Performance with Software RAID, p. 68
- Debugging Programs on a Cell Processor, p. 74
- Interview with Cory Doctorow, p. 78
- Using Python for Scientific Computing, p. 88
- Simplifying GPU and Accelerator Programming, p. 82



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, Inc.
540 Dado Street, San Jose, CA 95131
www.ironsystems.com



CALL: 1-800-921-IRON

CONTENTS NOVEMBER 2008

Issue 175

COLUMNS

- 8** SHAWN POWERS' CURRENT_ISSUE.TAR.GZ Sometimes, Fast Just Isn't Enough
- 20** REUVEN M. LERNER'S AT THE FORGE Book Roundup
- 24** MARCEL GAGNÉ'S COOKING WITH LINUX Warp-Speed Blogging



- 30** DAVE TAYLOR'S WORK THE SHELL Pushing Your Message Out to Twitter
- 32** MICK BAUER'S PARANOID PENGUIN Samba Security, Part I
- 36** KYLE RANKIN'S HACK AND / Memories of the Way Windows Were
- 96** DOC SEARLS' EOF Lincoln and Whitman's Unfinished Business

IN EVERY ISSUE

- 8** FROM THE EDITOR
- 12** LETTERS
- 14** UPFRONT
- 40** NEW PRODUCTS
- 42** NEW PROJECTS
- 81** ADVERTISERS INDEX

REVIEWS

- 46** TRACKING YOUR BUSINESS FINANCES WITH NOLAPRO Mike Diehl
- 50** THE POPCORN HOUR A-100 Daniel Bartholomew



INDEPTH

- 78** CORY DOCTOROW—LINUX GURU? Cory Doctorow on DRM, his new novel and more.

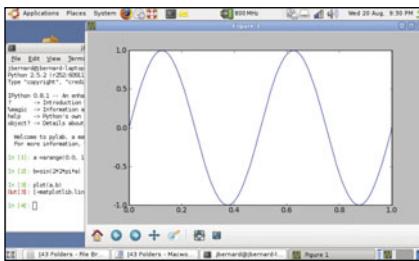
Dan Sawyer

- 82** HOW WE SHOULD PROGRAM GPGPUS Porting to GPUs without heroic programming effort.

Michael Wolfe

- 88** USE PYTHON FOR SCIENTIFIC COMPUTING Leverage the benefits of Python for scientific computing.

Joey Bernard



56 THE ROADRUNNER SUPERCOMPUTER

Next Month

GADGETS

Next month, we tackle small Linux gadgets in a big, big way. Whether your interests are in netbooks, gadgets or just using Linux with gadgets, it will be an issue you won't want to miss. Not sure how you feel about the future of embedded Linux? That's okay too, because we've got interviews with some of the movers and shakers in the industry. Watch for us on newsstands and in mailboxes next month!

USPS LINUX JOURNAL (ISSN 1075-3583) (USPS 12854) is published monthly by Belltown Media, Inc., 2211 Norfolk, Ste 514, Houston, TX 77098 USA. Periodicals postage paid at Houston, Texas and at additional mailing offices. Cover price is \$5.99 US. Subscription rate is \$29.50/year in the United States, \$39.50 in Canada and Mexico, \$69.50 elsewhere. POSTMASTER: Please send address changes to *Linux Journal*, PO Box 980985, Houston, TX 77098. Subscriptions start with the next issue. Canada Post: Publications Mail Agreement #41549519. Canada Returns to be sent to Blechup International, P.O. Box 25542, London, ON N6C 6B2

Polywell Linux Solutions

More Choices, Excellent Support Service, Great Value!
Serving Small and Medium OEM/VAR for More Than 20 Years

Netdisk 8000V 8TB High Performance RAID-5 NAS Storage



4U 24Bay Storage Server

24TB Dual or Quad Processors
Up to 128GB RAM, 4 x GigaLAN



Polywell OEM Services, Your Virtual Manufacturer
Prototype Development with Linux/FreeBSD Support
Small Scale to Mass Production Manufacturing
Fulfillment, Shipping and RMA Repairs

888.765.9686

linuxsales@polywell.com



Slim Terminal PC

NVIDIA GeForce® Graphics
Intel or AMD Processor
1G DDR2, starts at \$199

Fanless Mini-ITX

Fanless Mini-ITX with Intel Processor
1G DDR2, 80GB Hard Drive starts at \$299
NVIDIA GeForce® Graphics, Low-profile Add-on Available



Small 1U Server for Data Center ISP

64-bit QuadCore Phenom or Core2 Duo CPU
4 to 8GB DDR2-800, 2 x 500GB RAID HD
Linux Server Starts at \$499



- 20 Years of Customer Satisfaction
- 5-Year Warranty, Industry's Longest
- First Class Customer Service

www.polywell.com/us/Lx



Polywell Computers, Inc 1461 San Mateo Ave. South San Francisco, CA 94080 650.583.7222 Fax: 650.583.1974

Geforce, nForce and Nvidia are trademarks of NVIDIA Corporation. All other brands, names are trademarks of their respective companies.

LINUX JOURNAL

Since 1994: The Original Magazine of the Linux Community

Digital Edition Now Available!

Read it first

Get the latest issue before it hits the newsstand

Keyword searchable

Find a topic or name in seconds

Paperless archives

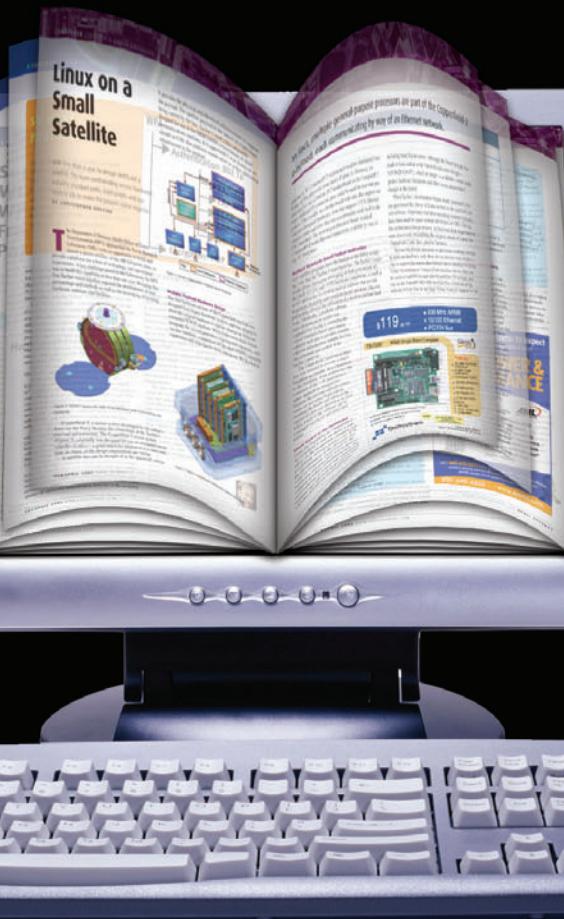
Download to your computer for convenient offline reading

Same great magazine

Read each issue in high-quality PDF

Try a Sample Issue!

www.linuxjournal.com/digital



LINUX JOURNAL

Executive Editor	Jill Franklin jill@linuxjournal.com
Associate Editor	Shawn Powers shawn@linuxjournal.com
Associate Editor	Mitch Frazier mitch@linuxjournal.com
Senior Editor	Doc Searls doc@linuxjournal.com
Art Director	Garrison Antikajian garrison@linuxjournal.com
Products Editor	James Gray newproducts@linuxjournal.com
Editor Emeritus	Don Marti dmarti@linuxjournal.com
Technical Editor	Michael Baxter mab@cruzio.com
Senior Columnist	Reuven Lerner reuven@lerner.co.il
Chef Français	Marcel Gagné mgagne@salmar.com
Security Editor	Mick Bauer mick@visi.com

Contributing Editors

David A. Bandel • Ibrahim Haddad • Robert Love • Zack Brown • Dave Phillips • Marco Fioretti
Ludovic Marcotte • Paul Barry • Paul McKenney • Dave Taylor • Dirk Elmendorf

Proofreader	Geri Gale
--------------------	-----------

Publisher	Carlie Fairchild publisher@linuxjournal.com
General Manager	Rebecca Cassity rebecca@linuxjournal.com
Sales Manager	Joseph Krack joseph@linuxjournal.com
Circulation Director	Mark Irgang mark@linuxjournal.com
Webmistress	Katherine Druckman webmistress@linuxjournal.com
Accountant	Candy Beauchamp acct@linuxjournal.com

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

Reader Advisory Panel

Brad Abram Baillio • Nick Baronian • Hari Boukis • Caleb S. Cullen • Steve Case
Kalyana Krishna Chadalavada • Keir Davis • Adam M. Dutko • Michael Eager • Nick Faloutsos • Ken Firestone
Dennis Franklin Frey • Victor Gregorio • Kristian Erik • Hermansen • Philip Jacob • Jay Kruizinga
David A. Lane • Steve Marquez • Dave McAllister • Craig Oda • Rob Orsini • Jeffrey D. Parent
Wayne D. Powel • Shawn Powers • Mike Roberts • Draciron Smith • Chris D. Stark • Patrick Swartz

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
Bruce Perens
Bdale Garbee, Linux CTO, HP
Danese Cooper, Open Source Diva, Intel Corporation

Advertising

E-MAIL: ads@linuxjournal.com
URL: www.linuxjournal.com/advertising
PHONE: +1 713-344-1956 ext. 2

Subscriptions

E-MAIL: subs@linuxjournal.com
URL: www.linuxjournal.com/subscribe
PHONE: +1 713-589-3503
FAX: +1 713-589-2677
TOLL-FREE: 1-888-66-LINUX
MAIL: PO Box 980985, Houston, TX 77098 USA
Please allow 4-6 weeks for processing address changes and orders
PRINTED IN USA

LINUX is a registered trademark of Linus Torvalds.

OUTRAGED by the high cost of Fibre Channel or iSCSI Storage ?

Switch to AoE !



Coraid Offers a Complete Line of Clustered Modular Storage Products:

- High Performance EtherDrive® SATA+RAID Storage Appliances with 1 GigE or 10 GigE Connections
- Clustered VirtualStorage™ Appliances (a Revolutionary Logical Volume Manager)
- Scalable NAS Gateways (File Sharing with EtherDrive® Storage)



Coraid's EtherDrive® Storage with AoE is fast, reliable disk storage that's easy to use. And it's much more affordable than iSCSI or Fibre Channel!

- Coraid products use open AoE (ATA-over-Ethernet) block storage protocol, for high performance without the TCP/IP overhead
- With AoE, your shared storage capacity is infinitely scalable – at a fraction of the cost of iSCSI or Fibre Channel storage
- We provide a 3-year warranty and free firmware upgrades on all our products, as well as support from first-rate engineers trained in our technology



EtherDrive® Storage has a field-proven track record and is 1000+ large data storage customers strong.



To learn more about this and Coraid's other products, go online or call

+1.706.548.7200 (toll free: **877.548.7200**)

www.coraid.com



SHAWN POWERS

Sometimes, Fast Just Isn't Enough

When I first started using computers, "High-Performance Computing" basically meant how fast a person could type. People were considered high performance if they could type faster than the IBM electric typewriter could smack out the letters. In fact, if you haven't seen a race between a person on a manual typewriter versus a person on an IBM Selectric "golfball" typewriter, you haven't lived.

Times quickly changed, however, and now instead of counting characters per second, we use terms like petaflops. Although arguably more funny to verbalize, petaflops don't really measure the same thing as characters per second. In an abstract notion, however, they both measure "how much stuff" a piece of hardware can churn out. This issue is thankfully dedicated to high-performance computing, not high-speed typing. If I pitched an issue focusing on the IBM Selectric typewriter, I have a sneaking suspicion it would be my last issue as Associate Editor.

Our writers came through this month and, indeed, focused on high-performance computing. In fact, James Gray takes us to the headquarters of the Roadrunner supercomputer. It's not exactly the type of system most people can build in their bedrooms, but a fascinating look at some real horsepower. If such setups seem too "pie in the sky" for you, fear not. We have a bunch of other articles that you can dig right into.

You can supercharge your programming by using CUDA and leveraging some of the GPU processor time to bend to your will. Robert Farber explains how. Or, perhaps you'd rather take advantage of your high-performance operating system and replace your hardware RAID setup with a Linux-powered software RAID system. Will Reese shows the advantages of doing such a thing, along with instructions on how to do it.

There's a lot to be said for writing good code, however. Often, if the code is good, even a regular desktop machine can act like a high-performance beast. Reuven Lerner

provides a big roundup of books that is sure to help along the way to some high-performance code. A word of warning, however; you may need to buy another bookshelf.

What if you're just a Linux desktop user like me? Well, we didn't leave you out. Marcel Gagné shows you how to streamline your blogging habits by utilizing the new microblogging services out there. Twitter? Identi.ca? Jisko? Yep, plus more. Marcel explains how to make microblogging as efficient and effective as possible, and these days, high-performance blogging is 140 characters or less.

If you hate the whole Twitter concept, fear not. Kyle Rankin shows how to streamline your desktop experience with Compiz. I'm suspicious that Kyle just wanted to prove Compiz was a legitimate addition to a business desktop, when, in fact, he just likes wiggly windows. He, of course, would deny any such thing. Check out his column, and see what you think.

Can a person be a high-performance device in the Open Source world? If so, Cory Doctorow would be a supercomputer when it comes to open standards and free information. This month, we have an interview with a man on the front line fighting against DRM. As you can imagine, he has some kind words to say about Linux. You won't want to miss it.

Just like every other month, we have our regular cast of columns, reviews and tech tips. Whether you're looking for information on installing and securing Samba or you're interested in solving programming tasks with Python, this issue will be one you'll want to read from cover to cover. As for me, I think I'm going to go dig out that old IBM Selectric and see if I can still type faster than it can print. For some reason, I suspect I might not be as awesome as I remember (but I am hot*).■

Shawn Powers is the Associate Editor for *Linux Journal*. He's also the Gadget Guy for LinuxJournal.com, and he has an interesting collection of vintage Garfield coffee mugs. Don't let his silly hairdo fool you, he's a pretty ordinary guy and can be reached via e-mail at shawn@linuxjournal.com. Or, swing by the #linuxjournal IRC channel on Freenode.net.

* www.linuxjournal.com/content/extra-shawn-powers-hot

Servers DIRECT®

MORE PRODUCTS, BETTER SERVICE, GUARANTEED.

RELIABILITY WHEN YOU NEED IT. IN OTHER WORDS, ALL THE TIME.

With ServersDirect® Systems based on the Intel® Xeon® Processor, you can count on high availability for your business-critical computing solutions.

GO STRAIGHT TO THE SOURCE! 1.877.727.7887 | www.ServersDirect.com



SDR-S1208-T00

STARTING AT \$599

- Supermicro Mini 1U Rackmount Server with 260W Power Supply
- Supermicro Server Board w/Intel® 946GZ Chipset
- Support up to a Dual-Core Intel® Xeon® 3000 Series processor
- TPM Support
- 1x 3.5" Internal Drive Bay
- 2x Intel® 82573 PCI-e Gigabit LAN Port



SDR-S1305-T04

STARTING AT \$999

- Supermicro 1U Rackmount Server with 520W Power Supply
- Supermicro Server Board w/Intel® 5000V Chipset
- Dual Intel® 64-bit Xeon® Quad-Core or Dual-Core 5400/5300/5200 sequence
- Support up to 16GB DDR2 667 & 533 FB-DIMM
- 4 x SATA Hot-swap Drive Bays
- Dual-port Gigabit Ethernet Controller



SDR-S1330-T04

STARTING AT \$1,519

- Supermicro 1U Rackmount Server with 650W High-efficiency Redundant Power Supply
- Supermicro Server Board w/Intel® 5400 Chipset
- Dual Intel® 64-bit Xeon® Quad-Core or Dual-Core 5400/5300/5200 sequence
- Support up to 64GB DDR2 667 & 533 FB-DIMM
- 4 x SATA Hot-swap Drive Bays
- Dual-port Gigabit Ethernet Controller



SDR-C2205-T00

STARTING AT \$699

- 2U Rackmount Server with 300Watt Power Supply
- Supermicro Server Board w/Intel® 3200 + ICH9R Chipset
- Support up to a Quad-Core Intel® Xeon® 3300/3200 series processor
- Up to 8GB unbuffered ECC / non-ECC DDR2 800/667 SDRAM
- 4x 3.5" Internal Drive Bay
- Dual-port Gigabit Ethernet Controller



SDR-C2300-T06

STARTING AT \$1,149

- 2U Rackmount Server w/600W Power Supply
- Supermicro Server Board w/Intel® 5000P chipset
- Dual Intel® 64-bit Xeon® Quad-Core or Dual-Core
- Support up to 16GB DDR2 667 & 533 FB-DIMM
- 6 x 3.5" Hot-swap Drives Trays
- Dual-port Gigabit Ethernet Controller



SDR-S2301-T08

STARTING AT \$1,599

- 2U Rackmount Server w/700W High-Efficiency Redundant Power Supply
- Supermicro Server Board w/Intel® 5000P chipset
- Dual Intel® 64-bit Xeon® Quad-Core or Dual-Core
- Support up to 64GB DDR2 667 & 533 FB-DIMM
- 8 x 3.5" Hot-swap Drives Trays
- Dual-port Gigabit Ethernet Controller



SDP-CP200-T04

STARTING AT \$899

- Pedestal Server with 600W Power Supply
- Intel® Server board with 3000 Server Chipset
- Support up to a Quad-Core Intel® Xeon® 3300/3200 series processor
- Up to 8GB unbuffered ECC / non-ECC DDR2 800/667 SDRAM
- 4x 3.5" Internal Drive Bay (Hot-swap Optional)
- Dual-port Gigabit Ethernet Controller



SDR-IP300-T06

STARTING AT \$1,199

- Intel Pedestal Server with Redundant (1+0) 650W Power Supply
- Supermicro Server Board w/Intel® 5000V Chipset
- Dual Intel® 64-bit Xeon® Quad-Core or Dual-Core
- Support up to 16GB DDR2 667 & 533 FB-DIMM
- Support up to 6 x 3.5" Hot-swap Drives Trays
- Dual-port Gigabit Ethernet Controller



SDP-C5300-T24

STARTING AT \$2,969

- 5U Rackmount Servers with Redundant 1350W Power Supply
- Supermicro Server Board w/Intel® 5000P Chipset
- Dual Intel® 64-bit Xeon® Quad-Core or Dual-Core
- Support up to 32GB DDR2 667 & 533 FB-DIMM
- Support up to 24 x 3.5" Hot-swap Drives Trays
- Dual-port Gigabit Ethernet Controller

SERVERS DIRECT CAN HELP YOU CONFIGURE YOUR NEXT HIGH PERFORMANCE SERVER SYSTEM - CALL 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 logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, Pentium, and Pentium III Xeon are trademarks of Intel Corporation or its subsidiaries in the United States and other countries.



**Dual-core.
Do more.**

Bring your website to life with 1&1!



1&1 Website Builder

Create an eye-catching
website in minutes!
No HTML knowledge
needed!

Welcome to Laura's Website!

Welcome to my website! Hi there-

I created this website to talk about my favorite bands and share pictures with my friends. There's a page with handy links for buying tickets to concerts. I also listed some tips that will help you find new bands with great music.

Check back soon because I'll be putting up more pictures after the concert this weekend!

Home
Favorite Bands
Lyrics
Links
Tips
Photos
Contact

A screenshot of a website titled "Welcome to Laura's Website!". The site features a blue header with yellow flowers, a sidebar with navigation links, and a main content area with text and images. A sidebar on the left lists various website sections like Home, Favorite Bands, and Contact.

1 Choose your layout

The interface shows a "Design Template" section with a search bar and a preview area displaying various website layout templates. A large preview window shows a website design with a blue header and a grid of images.

World's #1 Web Host

With a wide variety of products and hosting packages, superior data center technology, unparalleled reliability, special offers, great prices and a 90-Day Money Back Guarantee, it's no wonder 1&1 is the world's biggest and fastest growing web host!



Overnight Prints™

For a limited time, get 100
FREE personalized business
cards with your 1&1 Home or
Beginner Package.*

Promote your website!

Use your FREE business
cards to share your
website address.

**Limited Time
Offer!**

1&1		Go Daddy	Verio
Best Value: Compare for yourself.	BEGINNER	ECONOMY PLAN	HOSTING 1000
Included Domain Names (.com, .net, .org, .info or .biz)	1 Domain FREE	\$ 1.99/year	1 Domain Free
Web Space	10 GB	10 GB	5 GB
Monthly Transfer Volume	300 GB	300 GB	200 GB
E-mail Accounts	600	100	100
Mailbox Size	2,000 MB	100 MB	5,000 MB
Website Builder	8 Pages	\$ 4.99/month for 5 Pages	9 Pages
Photo Gallery	✓	✓	✓
Blogging	✓	✓	✓
90-Day Money Back Guarantee	✓	—	30-Day Risk Free Trial
Support	24/7 Toll-free Phone, E-mail	24/7 Phone, E-mail	24/7 Phone, E-mail
Price Per Month	\$ 3.99	\$ 4.29	\$ 11.95

© 2008 1&1 Internet, Inc. All rights reserved.

Visit 1and1.com for details. Prices based on comparable Linux web hosting package prices, effective 8/26/2008.

* Offer valid for new customers only and applies exclusively to 1&1 Home and Beginner Packages. Limit 1 voucher per customer. For full promotional offer details, visit www.1and1.com. Product and program specifications, availability, and pricing subject to change without notice.

2 Arrange your pages

My WebsiteBuilder Website

- Home
- Favorite Bands
- Lyrics
- Links
- Tips
- Photos
 - Family
 - Me
 - Contact

3 Add text and pictures

Designs

Color Scheme:

Main Image:

Navigation Menu:

A WebsiteBuilder Website

4 Publish your website

Welcome to Laura's Website!

Check back soon because I'll be putting up more pictures after the concert on Saturday.

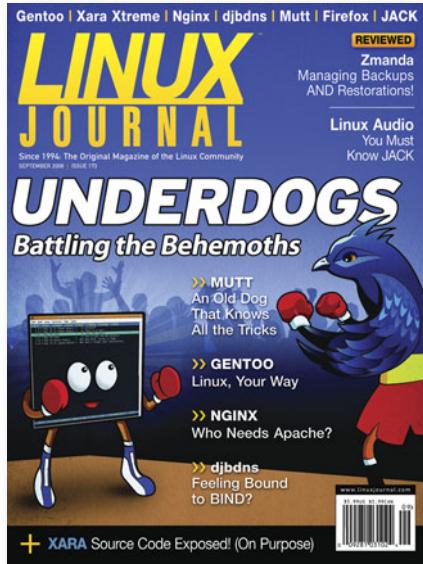
→ Laura

Call 1.877.go1and1
Visit us now 1and1.com



1&1

letters



Bashing Arithmetic

I was surprised to see Dave Taylor imply that the magic bash variable \$RANDOM is a feature of \$(()) arithmetic syntax [see "Movie Trivia and Fun with Random Numbers", *LJ*, August 2008]. It is just a magic variable that can be used anywhere. Likewise, there is no need to use both \$(expr) and \$(()); one or the other is sufficient. In particular, lines like:

```
pickline=$(expr $(( $RANDOM % 250 )) + 1 )"
```

could have been simplified to:

```
pickline=$(( $RANDOM % 250 + 1 ))
```

I might also mention in passing that double quoting is redundant in variable assignment, even if the expression would normally be unsafe without quoting.

--
Peter Samuelson

Dave Taylor replies: Thanks for your note. I didn't mean to imply that \$RANDOM was part of the \$(()) notation, but I will say that in my experience it's far more useful in that context than elsewhere I use it. Finally, although the double quotes are occasionally unneeded, I find

that a consistent style (for example, always quote variable assignments) helps with debugging.

End of TV?

In reading Doc Searls' article on the end of analog TV and seeing his background [see "What Happens after Next February?", *LJ*, September 2008], I had to speak up and say, "Yes, I was there too in those days." I have tinkered with radio, most of my life—K9LD. Wire recorders, 21 tube television sets, tall towers with long yagis. I started in computers when it was diodes and telephone relays, DEC, right on up until now. I would rather write programs than anything else.

Point being, I believe TV will just die. You can download your movies and watch them when you want. Every home has a PC running something. I subscribe to Netflix, and would rather that instead of mailing out the DVDs, they would put it all on a point-and-click basis—ah, those copyright laws though. Someone will come up with a scheme to where your programming, which is all we are interested in with television anyway, will be selected and viewed on your computer—whether it's 20 inches, handheld or, dare I say it, built in to your glasses.

--
Larry Dalton

Honest Writing and Technical Criticism, Please

I'm partway through reading Eric Pearce's article on a 16TB backup NAS (essentially), and I really have to compliment his writing [see "One Box. Sixteen Trillion Bytes.", *LJ*, August 2008]. It's simple, direct and honest ("here are the things I thought about doing, but didn't have the time to try", and "FYI: I'm not sure about these options in practice, but maybe they could improve performance"), and I truly appreciate that.

I find that very often technical people (microbiologists, analytical chemists, IT workers and so on) paralyze themselves

into quiescence because they want to present not just a problem but a solution—and not just any solution, but a very thoroughly thought-out and "perfect" and utterly defensible solution. The honesty and practicality that Eric shows in his article is a kind of triumph of what he actually accomplished over the common tendency to "self-paralysis". It's his writing this down as he did that really impressed me.

I'm glad there is a forum like *Linux Journal* where the writers can be that open and (I urge all of us, including myself when I'm ornery!) the readers keep their criticisms technical.

--
redeschene

It's a Vendor Thing

I am not a computer specialist and neither do I have any interest in computer code. But, I use a computer most of the day, every day. Having been stuck with Windows (which I don't like because of the way everything I do is controlled by Microsoft), I recently bought a small laptop with Linux as the operating system. It is an absolute disaster area. For a start, it is incompatible with 3 mobile broadband (I have read a number of blogs and even the experts agree on that). I have had no success in loading Java, which is essential for the work I do. And, I can't even load a 56K modem for emergency use. In short, it is totally useless to me, and I am going to have to load up Windows XP instead—much against my wishes. I had hoped that Linux was a serious competitor to Microsoft, but in reality, it is light-years away, strictly for computer specialists. Of course, I could spend days and days reading up on how to make it work, but why should I? I only want to use the computer, not re-invent it. Kernels, shells, command prompts—these things are of no interest to me whatsoever. It's back to the dark days of MSDOS all over again.

--
Richard Bonny

Shawn Powers replies: I feel your pain. It is so frustrating to buy a computer, especially one preloaded with Linux, only to have it fail during normal, everyday tasks. You didn't mention the brand or vendor of your laptop, but I could name a handful of "Linux-friendly" vendors shipping laptops that seem crippled when they arrive.

My suggestion would be to purchase a laptop from a vendor like EmperorLinux—one that is known for retro-fitting Linux into computers and doing it well. As for the laptop you currently own, there still might be hope, but I'd need more details to point you in the right direction.

It's frustrating as a Linux evangelist

when vendors sell pre-installed computers that don't work quite right. I assure you, it's not a Linux thing, but rather a vendor thing. If a vendor shipped a Windows notebook without the drivers, I'd venture to guess it would be even less useful than your Linux laptop.

Yay for Mobile LinuxJournal.com, ELinks and Mutt!

I was excited to read about the mobile version of the *LJ* Web site [go to m.linuxjournal.com to try it out], as it will be perfect not only for my Nokia N800, but also my new Acer Aspire One running Linpus Linux Lite. Speaking of the One, I really enjoyed the articles by Marcel Gagné and Victor Gregorio regarding text-mode browsers and the Mutt e-mail pro-

gram, respectively [see "Browsers with the Speed of Lightning" and "Power Up Your E-Mail with Mutt", *LJ*, September 2008]. After trying several browsers, I settled on ELinks and have been trying it out on my Aspire One. I just installed Mutt and will be trying to configure it as soon as I send this message.

Thanks for another great issue! I just subscribed, and my first print issue should be arriving next month.

--
William Parmley

Supporting the "Made for Linux" pcHDTV

I was very interested when I read "Over-the-Air Digital TV with Linux" by Alolita Sharma in the July 2008 issue of *LJ*.

Expert included.

When it comes to efficiency, as Vice President of Operations for Silicon Mechanics, Eva really gets it. She recognizes that the Bladeform 9100 Blade Server Platform is engineered for efficiency at every level.

When efficiency means server density—The 7U Bladeform 9100 enclosure holds 14 blade servers. You get unparalleled density.

When efficiency means processing power—Each Bladeform 9110 Blade Server supports 2 Quad-Core Intel® Xeon® 5000 Series processors. You get a possible 672 cores in a 42U rack.

When efficiency means performance per watt—Because blade servers share chassis resources like power supplies, the blade form factor is inherently efficient. And the power supply modules for the Bladeform 9100 have 93% maximum efficiency. You get maximized performance per watt.

When efficiency means price point—Blade servers reduce deployment, management, and energy costs, and with the Bladeform 9100 Platform you won't have to pay more to realize those benefits. You get blade technology at prices that match equivalent 1U installations.

When you partner with Silicon Mechanics, you get more than a blade server platform with efficiency engineered in from top to bottom—you get an expert like Eva.



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

**See the Silicon Mechanics
Bladeform 9100 Blade Server Platform at
www.siliconmechanics.com/9100**

Silicon Mechanics and the Silicon Mechanics logo are registered trademarks of Silicon Mechanics, Inc. Intel, the Intel logo, Xeon, and Xeon Inside, are trademarks or registered trademarks of Intel Corporation in the US and other countries.



**Powerful.
Efficient.**

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 +1-713-589-3503. E-mail us at subs@linuxjournal.com or reach us via postal mail, Linux Journal, PO Box 980985, Houston, TX 77098-0985 USA. Please remember to include your complete name and address when contacting us.

DIGITAL SUBSCRIPTIONS: Digital subscriptions of *Linux Journal* are now available and delivered as PDFs anywhere in the world for one low cost. Visit www.linuxjournal.com/digital for more information or use the contact information above for any digital magazine customer service inquiries.

LETTERS TO THE EDITOR: We welcome your letters and encourage you to submit them at www.linuxjournal.com/contact or mail them to Linux Journal, 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 +1 713-344-1956 ext. 2.

ON-LINE

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

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

I purchased the Pinnacle PCYV HD Pro Stick and was disappointed to find that they have recently changed chipsets. The 801e now utilizes the DIBcom 0700C-XCCXa-G. It seems that the community has just recently started reverse-engineering the stick. I have decided to keep the unit, anxiously awaiting the community support. In the meantime, I will be supporting the "made for Linux" pcHDTV. Love *Linux Journal*! Keep up the great articles.

--
Adam Roland

Xara Xtreme Correction

I'm a little baffled at the article on Xara Xtreme included in the September 2008 *Linux Journal*. How old is that article? It states: "Until last year, Xara X was a professional, closed-source, Windows-only commercial app..."

Huh? Xara Xtreme has been available for Linux since October 2005. The article also fails to note that development on the open-source version ended about two years ago, owing to the fact that the rendering engine was being kept proprietary, and thus, FOSS contributors lost interest. I'm guessing this article was written two years ago, at least. You might want to check the date on your mayonnaise if this kind of stuff is slipping by. Care to comment?

--
Alan C. Stegerman

Dan Sawyer replies: When I got your comment, my immediate reaction was, "that can't be right", so naturally, I returned to my notes and dug around on the Web. Dating the open sourcing to last year was an oversight. I wish I had a good excuse, but I don't. The

proper date is there in my notes, and I should have seen it when I was fact-checking the article before I sent it in. It's a gross oversight—thank you for pointing it out. I'd rather be corrected on an error, so that people don't carry away inaccurate information from one of my articles.

As for the development controversy, I hadn't heard about it before your letter. After receiving your message, forwarded on to me by my editor, I dug. And dug. And dug. And eventually, I stumbled upon a blog that mentioned the matter in passing and linked to the developer's listserv group. Here's what I learned.

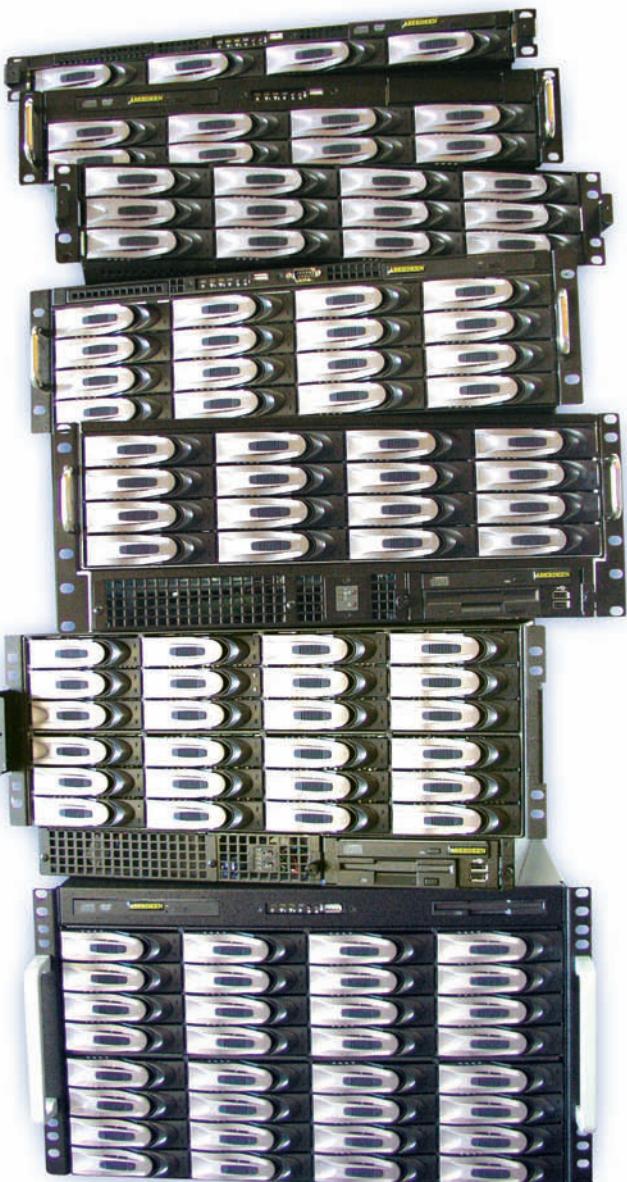
The latest I can ascertain is that there was, at some point this time last year, an effort to port Xara Xtreme to Cairo away from CDraw, in order to fix the problem (the acquisition of Xara by another company evidently poached the effort to open source the CDraw library), and that most community involvement has stalled for the time being until that fix is back on-line. Xara either currently hosts or has made space to host the Cairo fork (the information I can find is unclear on this point). This doesn't change my opinion that it's a project that deserves a lot more attention (in fact, I think it reinforces the point). The code base is still available; the listserv is still running; and the SVN is still accepting commits. Xara open sourced a hell of a program, and it'll be a crying shame if its hiatus turns into a death on the vine.

Thanks for bringing the matter to my attention.

LJ pays \$100 for tech tips we publish. Send your tip and contact information to techtips@linuxjournal.com.



MIRROR MIRROR IN THE RACK



4-DRIVE 1U ABERNAS

Up to 4TB Capacity

- Dual-Core Intel® Xeon® Processor
- 2GB DDR2 Memory
- 300W Power Supply
- From 1TB to 4TB

Starting at **\$2,495**

8-DRIVE 2U ABERNAS

Up to 8TB Capacity

- Dual-Core Intel Xeon Processor
- 2GB DDR2 Memory
- 500W Redundant Power
- From 2TB to 8TB

Starting at **\$3,995**

16-DRIVE 3U ABERNAS

Up to 16TB Capacity

- Dual Quad-Core Intel Xeon Processors
- 2GB DDR2 Memory
- 650W Redundant Power
- Quad LAN and SAS Expansion
- From 8TB to 16TB

Starting at **\$7,995**

24-DRIVE 5U ABERNAS

Up to 24TB Capacity

- Dual Quad-Core Intel Xeon Processors
- 2GB DDR2 Memory
- 950W Redundant Power
- Quad LAN and SAS Expansion
- From 12TB to 24TB

Starting at **\$10,495**

32-DRIVE 6U ABERNAS

Up to 32TB Capacity

- Dual Quad-Core Intel Xeon Processors
- 2GB DDR2 Memory
- 1350W Redundant Power
- Quad LAN and SAS Expansion
- From 16TB to 32TB

Starting at **\$13,495**

40-DRIVE 8U ABERNAS

Up to 40TB Capacity

- Dual Quad-Core Intel Xeon Processors
- 2GB DDR2 Memory
- 1350W Redundant Power
- Quad LAN and SAS Expansion
- From 20TB to 40TB

Starting at **\$15,995**

Features & Benefits:

- With available NAS-to-NAS Mirroring and Auto-Failover over LAN
- Native Linux-based OS featuring iSCSI
- Supports SMB, CIFS, NFS, AFP, FTP
- DOM-based (Disc-On-Module) OS
- Independent Data Drive RAID
- RAID 0, 1, 5, 6 Configurable
- Convenient USB Key for Recovery
- NAS-2-NAS Replicator
- PCBackup Utility
- iSCSI Target Capable
- All 3U+ models support expansion via cost effective JBODs

"Aberdeen surpasses HP ... markedly higher scores ...
AberNAS 128 boasts outstanding features"

Network Computing—Aberdeen AberNAS 128



**Quad-core.
Unmatched.**

diff -u

WHAT'S NEW IN KERNEL DEVELOPMENT

The reason for taking time on the transition is because PowerPC is a very big, very active project with trees, branches and processes of its own. Paul's estimate is that six months should be enough time to do the transition. But in the meantime, Benjamin will be doing a sudden two-week takeover of the project while Paul is on vacation. This involves setting up a new git tree and relevant branches, and starting to accept patches from the large group of contributors. It's not clear whether success in the two-week period will shorten the overall transition, but it certainly will be a good test run.

OProfile also probably is changing maintainership. Because this involves both a kernel and user-space code set, there are several issues. **Robert Richter** will be taking over from **Philippe Elie** as maintainer of the kernel portion. It's also unclear whether Philippe is on board with this change, so it may turn out that OProfile has some deeper communication issues between the developers. The user-space portion of OProfile also seems to be changing hands, though not as formally as with kernel code.

The **kmemcheck** code now has an official maintainer—or actually two official maintainers. **Vegard Nossum** and **Pekka Enberg** will share maintainership. The kmemcheck patch is a cool little debugging tool that logs whenever memory is read that had not been

written previously. Clearly, there's a bug if we're trying to read data from a location to which we never wrote.

The **ZFS** filesystem may start to have more of a presence in the official kernel tree. **Sun** has released a read-only version of the filesystem under the GPL. This is pretty cool, but not as cool as if Sun had released the whole thing. A read-write version could be useful to anyone, but the read-only version will be useful really only to people who've been using **Solaris**. In its full form, ZFS is a very interesting filesystem—it can handle multiple exabytes of storage, pools multiple block devices seamlessly together, checksums everything and snapshots the history of all data changes on the filesystem. **Alan Cox** speculates that ZFS may be one of the only things keeping Solaris alive as an operating system, and that this is why Sun doesn't want to GPL the full code. Even if a read-write version were open sourced though, there still are patents covering parts of ZFS, and those patents currently are being fiercely litigated by Sun and **NetApp**. Even if the source code were fully available, the problem of getting permission to use the patents might be too thorny to overcome. In the meantime, **Kevin Winchester** has volunteered to port the read-only code into Linux, and he'll almost certainly have help—or at least advice—from **Christoph Hellwig**. **Ricardo Correia** also is attempting to rewrite ZFS from scratch as a **FUSE-based filesystem**. Ricardo's work already has made great strides, to the point where **Patrick Draper**, for one, has been able to use it for all his data. However, anyone trying

it out, he cautions, should be sure to back stuff up.

Karsten Keil submitted **mISDN**, a driver intended to replace the **I4L architecture** for **passive ISDN cards**. The new architecture is apparently a big improvement, as folks like **Tillman Schmidt**, who maintain the old I4L drivers, are very excited to get started using it.

The **touchscreen driver** continues to graft new and exciting touchscreen hardware onto its list of supported devices. **Alastair Bridgewater** recently added support for the **eGalax touchscreen** used in the **HP tx1305us tablet PC** as well as in other devices. Part of the problem with supporting all these different pieces of hardware is that the driver has to detect them by probing their various behaviors. This can lead to some very subtle weirdness, as Alastair discovered when his first patch submission was rejected on the grounds that it didn't programmatically detect the eGalax hardware.

David Altobelli has submitted code supporting the **HP iLO/iLO2 management processor**, used by system administrators to access servers remotely instead of at the console.

Michael Buesch has written a driver supporting the **GPIO pins** on all **Brooktree 8xx chips**. GPIO (General-Purpose Input/Output) pins are used for both input and output, depending on how they're configured. Michael's code supports that configuration. After a bit of trouble locating the appropriate maintainer, **Andrew Morton** finally pointed Michael to **David Brownell**, who could receive the patch.

—ZACK BROWN

YOU Are LinuxJournal.com

Yes, that's right, LinuxJournal.com is all about you. We want to know what makes you happy, and we want to share it with all our readers. You may have noticed something a little different with our Web articles—you get to rate them. What better way to let us know what works and give other readers the benefit of your experience? So, feel free to judge, rate, applaud, rant or

anything else that strikes you. Did I mention, it's all about you?

One thing we already know is that you are fans of OpenOffice.org. Be sure to check out our regular tutorial articles on-line, and you might learn a new trick or two. Whether you need to unlock the mysteries of line spacing in OpenOffice.org Writer or understand Calc functions, the answers

are there. You also may happen upon a new piece of software that will quite literally broaden your horizons. If you missed Mike Diehl's article "Exploring Space with Celestia", go check it out at www.linuxjournal.com/content/exploring-space-celestia. It is one of my personal favorites.

I hope to see YOU around the Web!

—KATHERINE DRUCKMAN

1.	Percentage of employees in Florida who go to work when they're sick because they need the money: 44
2.	Percentage of employees in Ohio who go to work when they're sick because they need the money: 50
3.	Percentage of Delta Airlines domestic fleet that will have Internet access for passengers by mid-2009: 100
4.	Exceeded average Internet-generated IP traffic in terabytes per day in 2007: 9,000
5.	Expected exceeded average Internet-generated IP traffic in terabytes per day in 2012: 21,000
6.	Price in dollars per megabyte of wireless texting: 1,000
7.	Price in dollars per megabyte of wireless voice: 1
8.	Price in dollars per megabyte of wireline voice: .1
9.	Price in dollars per megabyte of residential Internet: .01
10.	Price in dollars per megabyte of backbone Internet: .0001
11.	Percentage of the Top 50 Most Reliable Hosting Company sites that run on Windows: 18
12.	Percentage of the Top 50 Most Reliable Hosting Company sites that run on Linux: 50
13.	Percentage of the Top 10 Most Reliable Hosting Company sites that run on Linux: 60
14.	Percentage of the Top 5 Most Reliable Hosting Company sites that run on Linux: 80
15.	Percentage of the Top 3 Most Reliable Hosting Company sites that run on Linux: 100
16.	Maximum thousands of Ultra-Mobile PC (UMPC) shipments in 2007: 500
17.	Projected millions of UMPCs to ship in 2012: 9
18.	Millions of ASUS Eee PCs sold in the first half of 2008: 1.7
19.	Total 2007 Linux ecosystem spending in billions of dollars: 21
20.	Projected 2011 Linux ecosystem spending in billions of dollars: 49

Sources: 1, 2: NPR | 3: Delta Airlines
 4, 5: "Managing Proliferating Traffic Growth", Pieter Poll, PhD, Chief Technology Officer, Qwest
 6–10: "Network neutrality, search neutrality, and the never-ending conflict between efficiency and fairness in markets", by Andrew Odlyzko, Digital Technology Center, University of Minnesota
 11–15: Netcraft.com report covering July 2008
 16–18: IDC, via *Investors Business Daily*
 19, 20: "The Role of Linux Servers and Commercial Workloads", by IDC for the Linux Foundation

What They're Using

Alan Robertson's Eee PC Tote-Bag Hack

I ran into Alan Robertson at LinuxWorld in August 2008, at a keynote given by an IBM executive. Alan works for IBM, where his title is Senior Software Engineer, Business Resilience. That's biz-speak for Alan's leading work on High-Availability Linux. It was while talking about the latter that Alan let slip that he uses an ASUS Eee PC.



Alan Robertson

I was fascinated by the extreme polarity in scale between Linux-HA—with its heartbeat-enabled death-of-node detection, cluster management and other mission-critical industrial-grade uses—and the hot little portable. When we got to a table where he could show off the machine, Alan pulled out of his knapsack what looked like a colorful striped purse. It was an accessory bag Alan had hacked to hold the Eee PC as if it were nothing more than an extra battery. It was way cool.



Eee PC Bag Hack

"The deal is...it's supposed to be just for cables that you carry with your regular laptop, but it happens to fit the Eee PC", he said. That's because the bag comes with three pockets on each side, and Alan removed the two lines of stitching on one side, giving it one pocket for the Eee PC and three for its accessories. The result is an all-in-one case that's custom-hacked for the Eee PC, which fits snugly inside the de-stitched side, across from its relatively tiny AC power supply and whatever other small stuff you'd like to carry with it.

Alan's Eee PC has the new 9-inch screen, 20GB of solid-state storage and stock Xandros distro. Beyond its dimensional virtues, Alan likes the Wi-Fi signal-finding utility, because "it shows you the DHCP negotiation" and "all that stuff you need to know" if something isn't working—which it too often doesn't with Wi-Fi. He also showed off the Eee PC's multi-touch, which I hadn't seen working before outside of an iPhone. And, he noted how well Skype is integrated with Linux. "Skype already knows about the Webcam...so if I were video-conferencing you, it would just come up and work."

Alan's main laptop for work is still a Lenovo ThinkPad 360p. He notes that the airline seat power supply for that one weighs more than the Eee PC does by itself. Meanwhile, his Eee PC is a nice little all-in-four (pockets) portability solution.

If you're interested in Alan's bag hack, you want BuiltNY's "charger bag". It comes in four color themes: brown/mint green, wood-grain black/slate, black/powder blue, and Alan's version, called stripe #7/lava. BuiltNY retails the bag on its Web site for \$25, but at the time of this writing, Amazon sells it for \$13.97. Either way, it's in alignment with the Eee PC's own downscale cost.

—DOC SEARLS

Keeping the Kernel Klean

Operating systems drive devices. Linux is driven by open-source imperatives. So, naturally, Linux's kernel developers have a problem with closed-source kernel modules. And, just as naturally, they've hacked up a statement they hope will discourage the closed and encourage the open.

On his blog, Greg Kroah-Hartman explained, "As part of the Linux Foundation Technical board...we wanted to do something that could be seen as a general 'public statement' about them that is easy to understand and point to when people have questions". Here it is:

Position Statement on Linux Kernel Modules, June 2008

We, the undersigned Linux kernel developers, consider any closed-source Linux kernel module or driver to be harmful and undesirable. We have repeatedly found them to be detrimental to Linux users, businesses and the

greater Linux ecosystem. Such modules negate the openness, stability, flexibility and maintainability of the Linux development model and shut their users off from the expertise of the Linux community. Vendors that provide closed-source kernel modules force their customers to give up key Linux advantages or choose new vendors. Therefore, in order to take full advantage of the cost savings and shared support benefits open source has to offer, we urge vendors to adopt a policy of supporting their customers on Linux with open-source kernel code.

We speak only for ourselves, and not for any company we might work for today, have in the past or will in the future.

Below that are 176 names. The Linux Foundation has a slightly

broader statement:

The Linux Foundation recommends that hardware manufacturers provide open-source kernel modules. The open-source nature of Linux is intrinsic to its success. We encourage manufacturers to work with the kernel community to provide open-source kernel modules in order to enable their users and themselves to take advantage of the considerable benefits that Linux makes possible. We agree with the Linux kernel developers that vendors who provide closed-source kernel modules force their customers to give up these key Linux advantages. We urge all vendors to adopt a policy of supporting their customers on Linux with open-source kernel modules.

Either way the message is clear.

—DOC SEARLS

They Said It

The slow adoption of Vista among businesses and budget-conscious CIOs, coupled with the proven success of a new type of Microsoft-free PC in every region, provides an extraordinary window of opportunity for Linux....We'll work to unlock the desktop to save our customers money and give freedom of choice by offering this industry-leading solution.

—Kevin Cavanaugh, Vice President for IBM Lotus Software, blogs.zdnet.com/open-source/?p=2754

Linux should stop copying Windows circa 2001 and rather look at what Apple is doing these days around usability and design. I understand that to gain acceptance for new software it makes it easier for users if you mimic the behavior of the old software, but at some time, you need to step out and innovate in the user interface.

This "innovation" might just be driving maximum consistency in look and feel. I want to at least feel that the system is held together by well-engineered common design principles

and APIs rather than aging string and bubble gum.

Open-source projects like WordPress have attracted excellent graphic designers to build themes and skins. We need more graphic designers involved with open source. We need to build more software so that others can make it look great.

There are certainly many in the general Open Source community who understand this and are working toward this goal.

—Bob Sutor, VP Open Source and Standards, IBM, www.sutor.com/newsite/blog-open/?p=2455

NERDS WILL RULE ALL GALAXIES

—Xeni Jardin, twitter.com/xenijardin/statuses/870654762

Here's another neutral, open network: the electricity grid. It's transparent, open, anybody can do anything on it. As long as you know the protocols, you can plug any technology in to it. We can imagine a world where when you

plugged something in, the network asked, "Is it a Panasonic or a Sony TV you're plugging in?" "Is it radio or television?" "Is it pay TV or free TV?" And then, depending upon the answer to these questions, allocate the resource according to that information. It's possible to imagine an electricity grid like that, but would it be better?

—Lawrence Lessig, testifying to the FCC at a Stanford hearing in April 2008, www.lessig.org/blog/2008/04/testifying_fcc_stanford.html

Well, the e2e advocates are essentially arguing that end-to-end in engineering is the equivalent of the perfect, competitive market that economists know and love....But in fact, that's not the way the real world works. It's neither the economist nirvana of perfect competition nor is it the engineers' nirvana of e2e. It doesn't work that way.

—Gerald Faulhaber, then the chief economist at the FCC, during a Stanford panel on e2e (end-to-end) in December 2000, cyberlaw.stanford.edu/e2e/papers/e2e.panel5.pdf

Lifting the Fog from Cloud Computing

Back in August 2008, at LinuxWorld in San Francisco, the big buzzword was "Cloud Computing". It's a neat concept, but after a week of hearing folks talk about "in the cloud", I was about at the end of my rope. To add insult to injury, it seemed that the San Francisco fog confused many folks, and "Cloud Computing" started to be used synonymously with "Grid Computing", "Clustered Virtualization" and "My Company Is Cool".

For clarity's sake, I thought a brief vocabulary lesson was in order. Cloud computing is indeed a viable, exciting idea—but it helps if we all know what we're talking about.

Cloud Computing Explained

The idea behind cloud computing is that services, not servers, are offered to the end user. If people need a Web server, they buy Web services from the "cloud", and have no idea what is actually offering them the serving. The "cloud" essentially hides the server infrastructure from the client, and ideally scales on the fly and so on. Much of the confusion in terminology happens, because the cloud of services almost always is powered by a grid of computers in the background. Cloud computing itself, however, is just the abstraction of services away from servers themselves.

Cloud Advantages

The advantage is that a vendor can offer more reliable, diverse and scalable services to a user without the cost of dedicating hardware to each user. This allows for more graceful temporary spikes (Slashdot, Digg and so on), while not letting servers sit idle during low times. Because the back end is transparent to the user, those actual grids of computers in the background

can be geographically diverse, and often-times virtualized for easy migration, all without any end-user interaction. Ideally, it offers a reliable "service" to the end user, at a lower cost, and gives vendors flexibility in the back end, so they can manage servers in the most efficient way possible.

Cloud Computing's Dirty Little Secret

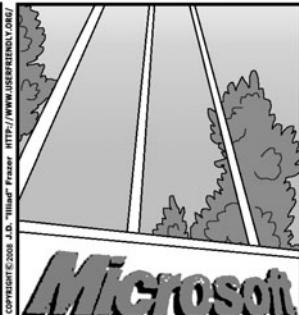
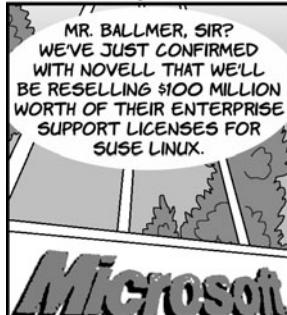
Most people don't realize that cloud computing ultimately is shared hosting. Vendors avoid terms like "shared hosting", because that implies multiple people sharing a single computer. By its strictest definition, however, cloud computing certainly could be run from a single back-end server. With current scalability and virtualization technologies, vendors have much more robust ways to serve to the "cloud", and the traditional hangups with shared hosting are largely eliminated. Still, it's important to understand what cloud computing really is, so you don't get fooled into buying more or less than what you truly need.

Questions to Ask Your Cloud Hosting Provider

- What sort of back-end servers are you running?
- Do you have the ability to fail over to a secondary data center behind the cloud, transparent to me?
- How do you differ from traditional shared hosting? (This one should spark some heated retorts!)
- How well do you scale, and how does pricing work for occasional spikes?

—SHAWN POWERS

USER FRIENDLY by J.D. "Illiad" Frazer



LINUX JOURNAL EDITION



*Sometimes
you have to
outrun the
competition.*



VersaLogic's Cheetah Single Board Computer (SBC) delivers swiftness and agility to embedded computing applications. Offered in Pentium® M and ULV Celeron® M versions, this product makes good on its promise of providing high-performance in a compact, ruggedized package. The Cheetah offers flexible RAM options and scalable processing for optimum application performance with minimal power draw. Available in standard (0° to +60°C) and extended (-40° to +85°C) temperature versions, the Cheetah features 10/100 Ethernet, high performance video capabilities, an on-board CompactFlash® socket and extensive integrated I/O. Customization is available on quantities of 100 pieces or more.

Contact us and discover how for more than 30 years we've been perfecting the fine art of extraordinary support, and on-time delivery: One customer at a time.

1.800.824.3163
1.541.485.8575

 VERSALOGIC
CORPORATION
www.versalogic.com/che



REUVEN M. LERNER

Book Roundup

Read a good book about Web development lately? Reuven has and is happy to share his latest favorites with you.

Remember the Web? You know, that combination of Internet technologies that will make paper obsolete? Well, it's true that many newspaper and magazine publishers are hurting, and that many people (including me) now read things on-line that we previously read on paper. And, for Web developers in particular, the Web provides a treasure trove of information, ranging from blogs and articles to forums and IRC channels.

Based on the flurry of content that is available on-line, and the reported death of print media, you might think that the number and quality of Web-related books also has declined in recent years. From what I can tell, however, the opposite is true. I continue to see a large number of high-quality books from a number of different publishers, on topics that are relevant and interesting. Even better, I have found many of these books to be useful in my day-to-day work, giving me perspective on technologies I already knew and teaching me many things that I hadn't previously known.

This month, I describe a few of the books that have most recently crossed my desk, which I have found to be the most helpful and interesting. This is an admittedly skewed sample. During the past few years, most of my work has been with Ruby on Rails, PostgreSQL and JavaScript, and also on looking at how to make Web sites more explicitly "social", with a large number of interactions among the participants.

Ruby on Rails

I have been working with Ruby on Rails for several years now and continue to see it as a breath of fresh air. After years in which I had to work directly with a database, writing explicit SQL queries, it's a delight to be able to think about things at the object level, rather than at the row-and-column level. Admittedly, some performance issues exist with Rails, but when you want to create a Web application quickly and easily, nothing else comes close. (Actually, that's not exactly true. If I hadn't been smitten by Rails, I probably would be using Django today. And, there are several Rails-like application frameworks out there, including CakePHP and Catalyst, both of which have their fans.)

The now-classic, standard text for beginning Rails programmers is the second edition of *Agile Web Development with Ruby on Rails*, written by

Dave Thomas and David Heinemeier Hansson, and published by the Pragmatic Programmers (ISBN 978-0977616633). This edition is due to be superseded by a third edition by the time you read this column, so some of my comments might not be relevant anymore. But, I generally have had mixed feelings about the first and second editions of this book. On the one hand, it successfully got me excited about Rails and provided me with a useful introduction back when I was first starting with it. At the same time, I kept searching for an introductory book with a slightly different style that I could recommend to beginning Rails developers.

One particularly strong introductory Rails book is *Simply Rails 2*, by Patrick Lenz (ISBN 978-0980455205). This book introduces a large number of key Rails features, including the MVC architecture, associations, testing, Ajax, some elements of Prototype and script.aculo.us, and even debugging with tools, such as ruby-debug. Because Lenz aims to cover less ground in his book, he manages to make a more effective tutorial than *Agile Web Development with Ruby on Rails*. Frequently interspersed code samples, screenshots and diagrams make this a particularly accessible book.

Once you have finished with *Simply Rails 2*, you will need a reference guide that describes the Rails architecture in greater depth, with a more comprehensive list of features. Obie Fernandez's book, *The Rails Way* (ISBN 978-0321445612), is an excellent, detailed and thick resource, although I think it mentions (and emphasizes) routes and RESTful architecture a bit early for many newcomers. That's probably fine, given that this book is not meant to be a tutorial, but rather a reference and advanced guide. So, although I strongly recommend this book and use it several times a week as a reference for my own work, it doesn't do the trick as an introduction or tutorial.

Good complements to *The Rails Way*, and to one another, are three similarly named books from two different publishers: *Rails Cookbook*, by Rob Orsini (ISBN 978-0596527310); *Rails Recipes*, by Chad Fowler (ISBN 978-0977616602); and *Advanced Rails Recipes*, by Mike Clark (ISBN 978-0978739225). Each of these books follows the now-standard format of having many short chapters, each addressing a common problem developers might encounter. Each of these books contains



multiple recipes that have given me ideas, and even when I haven't used the recipe directly, I have found them to be useful food for thought. I'm also not sure whether the third book is truly as advanced as its title would indicate.

Finally, the book *Deploying Rails Applications*, by Ezra Zygmuntowicz, Bruce Tate and Clinton Begin (ISBN 978-0978739201), has a great deal of practical advice about taking a Rails application and making it available to the general public. The authors guide you through using version control (with Subversion), configuring one or more production servers, deploying with Capistrano and identifying bottlenecks. The book assumes you are using MySQL, which means people who use PostgreSQL (like me) can ignore some of the advice. There also is a chapter on Windows deployment that I expect most readers of this column can ignore.

The authors assume you want to deploy your application with a combination of Mongrel (for dynamic, Ruby-generated content) and nginx (for static content), while ignoring such possibilities as Phusion (aka mod_rails), an Apache module that some sites have been using with great success. Then again, Zygmuntowicz is the founder of Engine Yard, a Rails hosting company that has been enjoying great success, so it might be wise to follow his lead. Regardless of the specific implementation choices the authors suggest, this book helps put each aspect of Rails deployment into perspective, and it's good reading for people who plan to make their applications public.

JavaScript and CSS

It used to be that a Web developer could get away with working only on the server side, allowing designers to handle everything from JavaScript to CSS. But, as Web applications have become more dynamic and have incorporated more Ajax, it has become increasingly important for all developers to understand and master these technologies.

For all of the buzz around JavaScript, seasoned Web developers know that it is a flawed language, partly in its design and partly in its implementation. One of

the best-known JavaScript gurus, Douglas Crockford, recently wrote a short (but excellent) book, *JavaScript: The Good Parts* (ISBN 978-0596517748). My understanding of JavaScript has been helped a great deal by Crockford's previous writing and lectures, and the book has helped me to appreciate this language more. *JavaScript: The Good Parts* has helped me understand why I have been so frustrated by JavaScript in the past—beyond issues of browser compatibility. By ignoring the bad parts of JavaScript, the frustration level drops considerably.

Even if you take Crockford's advice into consideration, you almost certainly will want to choose a JavaScript library, such as Prototype, jQuery, YUI or Dojo. I have explored some of these libraries in this column over the past few years, with a heavy emphasis on Prototype, because it is included with the Rails framework. However, there are many good things to say about the others—and in many cases, you might have no say over which library you use. For example, I have started to do some work with the open-source Moodle system for on-line learning, which uses YUI as its toolkit. Similarly, the Django framework for Web development now includes the Dojo toolkit, so you should expect to work with Dojo if you do any Django development.

Although I have been very impressed with Yahoo's YUI toolkit, my default JavaScript library remains Prototype, in no small part because of its close relationship with Ruby on Rails. (The Prototype developers are part of the Rails core team, and development between the two is synchronized.) The Web site for Prototype (prototypejs.org) has excellent documentation and even a downloadable PDF version of the API. But, this wasn't sufficient for some introductory classes that I recently gave about Prototype programming, so I have been on the lookout for a high-quality tutorial.

Thus, I was pleased to discover *Practical Prototype and script.aculo.us*, by Andrew Dupont (ISBN 978-1590599198), which is one of the best programming books I have read in the last few months. It introduces Prototype (and



"Finalist Best Management Tool"

Programmable Linux Based Console Server Solutions



Service Processor Management	Network UPS Tools
DRAC	Secure Tunneling
iLO	Port Forwarding
ALOM	Nagios Checks
RSA	Embedded IPMI
	Flexible Pinouts

opengear.com

its companion GUI toolkit, script.aculo.us) with on-target, practical and illuminating examples—along with a sense of humor I found refreshing, without getting in the way.

Modern Web pages are styled with cascading stylesheets (CSS), a technology that is quite simple to understand in theory, but it can become complicated when it comes to execution. One book that seems to offer a gentle introduction to CSS is *The CSS Anthology*, by Rachel Andrew (ISBN 978-0975841983). This book is a cross between a tutorial and a cookbook, allowing you to learn CSS in the context of bite-sized lessons and ideas.

Once you have understood the basics of JavaScript and CSS, and what they can do for your dynamic Web sites, you might want to look at *Dynamic HTML: The Definitive Reference*, by Danny Goodman (ISBN 978-0596527402). This book looks at the DOM—the document object model browsers use to work with HTML—and explains how you can modify its appearance, as well as manipulate its elements, using a combination of JavaScript and CSS. This book doesn't use any JavaScript library, so some of the JavaScript code might seem long and unwieldy if you are used to the brevity of Prototype. But, for sheer comprehensiveness, nothing beats this book.

The similarly comprehensive latest edition of David Flanagan's *JavaScript: The Definitive Guide* (ISBN 978-0596101992) is useful when you want to better understand what JavaScript is doing or how an object is defined. Douglas Crockford has noted that this is the only JavaScript book that actually gets the details right. That might be true, but it doesn't change the fact that the text can be quite dense. Don't try to learn JavaScript from this book, but you should have it around afterward, either to refresh your memory or to gain a clearer understanding of how things work.

Finally, I have found *The Ultimate CSS Reference*, by Tommy Olsson and Paul O'Brien (ISBN 978-0-980285857), to be a useful list of CSS selectors, properties and values. Especially useful is the table indicating the degree to which each browser complies with the CSS standard for each property. The information in this book is available elsewhere, but it is particularly handy if I am not connected to the Internet (which, I'm ashamed to admit, is sometimes the case), or if I just want to skim through a number of related items. My main complaint with this book is its lack of an index, and although I realize you can read and search much of it on-line, adding 5–10 pages of index would have changed this from a good book into an excellent one.

Social Networking

Facebook, LinkedIn and other social-networking

sites are all the rage, and I even have looked at some aspects of these sites in this column.

If you're interested in creating your own social-networking site, two books might be of interest to you. The first, *RailsSpace*, written by Michael Hartl and Aurelius Prochazka (ISBN 978-0321480798), describes how to create a simple social site with users, friends and Ajax-based blogs. The source code to RailsSpace has been made available, so you can use the code to build your own, real-world site, rather than merely go through the tutorials. Even better, one of the authors (Hartl) has founded an open-source project called Insoshi (www.insoshi.com), which offers a downloadable framework for creating social networking sites.

A second book on the subject, *Practical Rails Social Networking Sites*, by Alan Bradburne (ISBN 978-1590598412), is a bit more ambitious in the projects it aims to do, showing examples not only of users and friendship links, but also of e-mail, discussion forums, a photo gallery, user-created themes and a mobile interface. The code from this book is used to power the RailsCoders site (www.railscoders.net), and it similarly can be downloaded and used to power a real-world site.

No matter what toolkit you use, or whether you decide to create a social site on your own, you should consider the wider implications of what you are doing. *Designing for the Social Web*, by Joshua Porter (ISBN 978-0321534927), is one of the most interesting books I've read on the subject to date, and it's full of practical advice on how people participate in a site. A slightly less practical, but no less interesting book is Clay Shirky's *Here Comes Everybody* (ISBN 978-1594201530). His analysis of the social Web, and how groups are now collaborating before they fully know each other and define themselves, is full of interesting anecdotes, but also cautionary tales about what the designers of such sites should consider when deploying.

Conclusion

Perhaps the printing press is going the way of the dodo bird. But for the time being, there are plenty of books that I find not only useful and interesting, but also essential as I go about my daily life as a developer and consultant. If you have read a particularly interesting and useful book you think I should know about, please send me e-mail. I am always happy to learn about new, high-quality books, and if it turns out to be particularly useful, I'll be happy to share it with other readers of this column.■

Reuven M. Lerner, a longtime Web/database developer and consultant, is a PhD candidate in learning sciences at Northwestern University, studying on-line learning communities. He recently returned (with his wife and three children) to their home in Modi'in, Israel, after four years in the Chicago area.

EmperorLinux

...where Linux & laptops converge



Powerful Linux: The Raptor



Quad-core
QX9300

- Based on the ThinkPad W700 by Lenovo
- Highest performance NVidia 3-D graphics available on a laptop with WUXGA widescreen
- High performance quad-core Core 2 Extreme, 8 GB RAM
- ThinkPad T500 & W500 series laptops are also available

Features include:

- Up to 3.06 GHz Core 2 Extreme (dual-core) or 2.53 GHz Core 2 Extreme QX9300 (quad-core)
- 17" WUXGA LCD w/ X@1920x1200
- 160-320 GB @ 7200 rpm or 64 GB SSD, up to 8 GB RAM
- DVD±RW or Blu-ray, ethernet, 802.11a/g/n, Bluetooth
- One year Linux tech support - phone and email
- Three year manufacturer's on-site warranty
- Choice of pre-installed Linux distribution:



Powerful Linux



Rhino E6500/M6300

- Dell Latitude E6500/Precision M6300
- Up to 17" WUXGA w/ X@1920x1200
- NVidia Quadro FX 3600M graphics
- 2.2-2.8 GHz Core 2 Duo/Extreme
- Up to 8 GB RAM
- 60-320 GB @ 7200 rpm / 64 GB SSD
- DVD±RW or Blu-ray
- Starts at \$1350

Tablet Linux



Stingray ST5112

- Fujitsu Stylistic ST5112
- 12.1" XGA digitizer w/ X@1024x768
- 1.33 GHz Core 2 Duo
- Up to 4 GB RAM
- 80-120 GB hard drive
- Full tablet feature support: pressure sensitive pen, handwriting recognition
- Starts at \$2450

Rugged Linux



Tarantula CF-30

- Panasonic Toughbook CF-30
- Fully rugged MIL-SPEC-810F tested: drops, dust, moisture, & more
- 13.3" XGA TouchScreen
- 1.6 GHz Core 2 Duo
- Up to 4 GB RAM
- 80-320 GB hard drive
- Call for quote

www.EmperorLinux.com

1-888-651-6686

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



MARCEL GAGNÉ

Warp-Speed Blogging

Telling everyone you know about the many fascinating things in your life may sound like a daunting task for even the most dedicated blogger, requiring many hours at the computer and thousands of words. One way to speed things up is limiting your messages to 140 characters or less. Impossible? Not at all. Welcome to microblogging.

What are you doing, François? You have been sitting there working on that message for almost an hour. Surely your cousin doesn't need to know every detail regarding our wine cellar—after all, you told me you wanted to let him know about yesterday's wine, not all of them. *Quoi?* This is a different cousin? And, you had to let your parents know, and your aunts, uncles and friends? François, I know you have a large family and an extensive list of friends, but this is insane. If you need to keep everyone in the loop, there are better ways. You need a system that lets you contact everyone you know at once, keeping them up to date with short, pointed posts.

Mais oui, of course, you can do that with e-mail, but your messages do tend to go on, just as your blog posts do, *mon ami*—not that you aren't exceptionally witty and entertaining, François. To help you be succinct, I have just the thing, and you'll learn all about it shortly. I can see our friends approaching the restaurant even now. You'll have to finish that e-mail later.

Ah, welcome, *mes amis*, to *Chez Marcel*, where fine wine meets the best in free and open-source software. While François shows you to your tables, allow me to tempt your taste buds with the very thought of tonight's special wine selection. My faithful waiter and I were submitting this exceptional 2007 Terres Blanches Muscat Sec to quality control earlier today, and we both can confirm that it is truly wonderful. Now, François, please head to the cellar and bring back enough for everyone, so that we may all give in to temptation. *Vite, mon ami!*

While we wait for François to return, let me introduce you to the items on today's menu. I have three packages to show you, all of them examples of microblogging servers (and services). Microblogging (MB) is an interesting mix of blogging and instant messaging. Posts are generally updates sent either publicly or to a list of people who "follow" your published updates. The updates are limited to 140 characters, the traditional length of SMS phone text messages. The 140-character limit forces you to be brief, but it also makes it possible to follow the updates of a great many people. It doesn't take long to write 140 characters,

and it doesn't take long to read either.

The best known MB service on the Internet is probably Twitter, followed distantly by Jaiku. Although those two may be the best known, they aren't necessarily the best in terms of functionality or features. Today, I show you three MB services built entirely on free software. In all cases, I assume you have a working Apache server and MySQL installation. Depending on where you want to locate each service, you may need to update your Apache configuration as well.

Excellent, François! Good to have you back. Please, pour the wine for our guests. This Muscat is fantastic, *mes amis*, crisp and medium-bodied with a sweet/spicy nose and some lovely citrus flavors hitting the palate. Enjoy.

The first MB service I have to show you is arguably the one that has received the most attention of late. Launched in July 2008 by Evan Prodromou, Identi.ca is an MB service built on the free and open-source Laconica software. Laconica supports a Twitter-like API, can be updated via SMS or Jabber/XMPP, and it allows you to register and log in using OpenID.



Figure 1. Laconica is an exciting entry into the world of microblogging software, and it looks to correct Twitter's limitations with an open design.

To keep things moving along, and because I want to show you more than one MB alternative, I'm going to redirect you to my own cookingwithlinux.com Web site for details of the server-side installation of these packages, including Laconica. Here's the full path to the installation instructions: www.cookingwithlinux.com/free_microblogging.html.

After installing Laconica, point your browser to your new Laconica server and register your first account by clicking the Register link. You'll be transported to a page where you can fill in your nickname, password and other information (Figure 2).

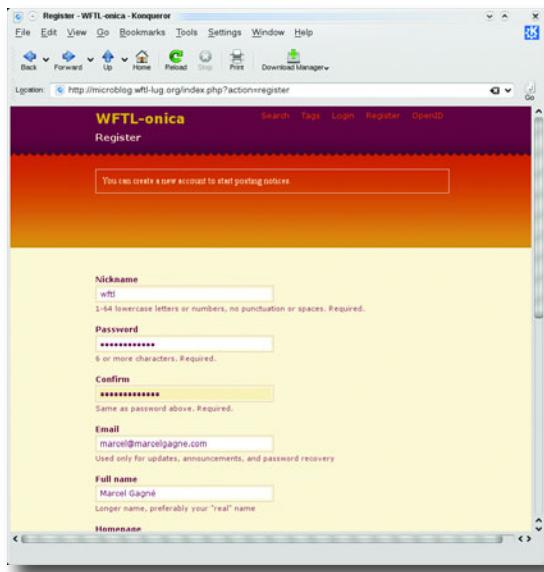


Figure 2. Registration is the same for all users. There is no Web-based administration.

Once you have registered your new account, a message will be sent to your e-mail address. Once you receive this message, you'll find a link that will return you to your Laconica site and provide confirmation (Figure 3). This is done to make sure the person who signed up for the account is a real user and not a spambot.

You don't actually need to enter any additional information, but you'll probably want to make some customizations before you start. This is all done under Settings, which you can find in a link on the top right of the page. Enter your full name, a Web site address if you have one, and a personal bio telling the world about yourself and your interests. Remember to keep it to 140 characters. Click Save, then click the Home link at the top of the page. This is your main page for entering updates. There are three tabs here. Under Personal, you will see updates from all the people to whom you are subscribed, including your own. Replies are posts

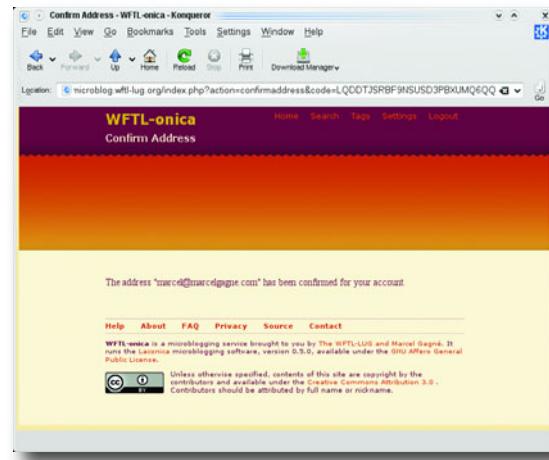


Figure 3. Once you respond to the confirmation link you received in the e-mail message, you are ready to start microblogging.

directed to you specifically—the convention is to add an @ sign in front of a person's nickname, followed by your message. At the end of each post, there is a little arrow you can click for the same results.

Finally, there's the Profile tab. This serves a couple functions. One is to show the world the information you have included in your settings. The second provides a list of the people whose posts you subscribe to, an archive of your previous updates, some statistics regarding how long you have been active, what your last post was and how many updates you have committed (Figure 4).

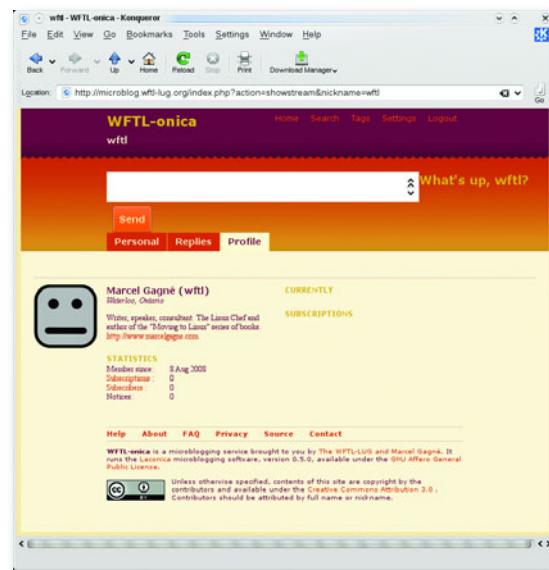


Figure 4. You are ready to start sending updates. As you enter your message in the What's up? field, the form keeps track of how many characters you have left.

To change your avatar, make sure you have a small image to upload (a representative avatar or a picture of yourself). Then, click on Settings at the top of the page. A number of tabs will be displayed labeled Profile, Email, OpenID and so on. The one you want is labeled Avatar. Once there (Figure 5), you can click the browser button to locate an image on your personal system. Keep it relatively small, then click the Upload button.



Figure 5. To upload a picture or avatar, enter its path or navigate to its location using the file manager button.

From here on, and including any earlier posts you may have made, your avatar will be displayed along with your posts (Figure 6). Speaking of posts, remember that unless you specify otherwise in the Settings, your posts are public and can be read by anyone, regardless of whether they "follow" you or not.



Figure 6. Now, all your updates are prefixed with your new avatar. Wait a minute. Isn't that François?

The growth of Laconica and Identi.ca has been nothing short of impressive. The code is being actively and enthusiastically developed by Evan and his legion of followers, which means you can expect great things from Laconica in the near future.

The next two packages aren't particularly well

known, but each is as impressive as Laconica in its own way. Both are, however, much simpler to set up. Let's start with Jisko (Figure 7).

Jisko seems like just another MB service, but it does have some interesting differences. For starters, you can attach files to your updates. Jisko doesn't tie you to its Web page; you can send and receive updates via Jabber/XMPP. Jisko also gives a nod to the giant gorilla (or perhaps I should say whale) of microblogging services, namely Twitter. I'll show you how that works shortly. Last but not least, Jisko also has a cute little mascot named Jisko.



Figure 7. Part of the beauty behind Jisko is that you can follow and update Twitter automatically if you have an account there.

Assuming you've taken care of the Apache server configuration side of things, all that's left to do is point your browser at your Jisko installation. This could be a simple URL like this one: <http://yoursite.dom/jisko>.

Your first order of business is to register an account. There are no admin accounts per se, so enter whatever nickname you want to use, along with your e-mail address and a password (Figure 8). Click Register to finish the process.

As with Laconica, a confirmation e-mail message will be sent along with a link for you to acknowledge. I found the text of this message to be somewhat amusing, so I thought I'd share it with you: "Someone (probably you) has requested an account on Jisko." Who else could it be? I guess it's pretty rare that somebody would create an account for you and not tell you the details. After acknowledging the e-mail, you can log in and start posting updates. Find and invite friends to join, and start updating each other on every little thing of interest that happens. It doesn't take long to fill 140 characters.

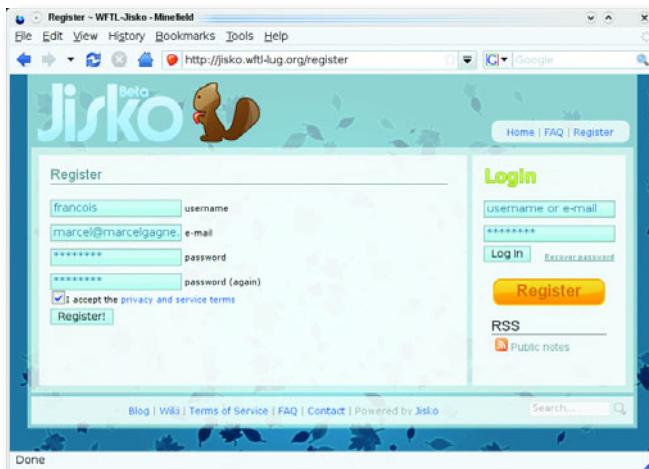


Figure 8. Registering an account with Jisko takes only a few seconds.

Odds are, you will want to make some changes. Click Settings at the top and upload a custom avatar or small image of yourself. Update your profile to include a few things about yourself. Change the visual theme and add a background image. You even can enter your Twitter user name and password. Once done, Jisko shows your Twitter updates and updates Twitter when you post to it. You'll update two different microblogging services and save time. Did you catch that last part, François?

Before we continue, I'd like to give you some incentive to explore the Jisko code and perhaps make your own little changes. Let me tempt you with something simple. Jisko has a cool little logo that rotates to show different colors with each page load. As you might have guessed, this, and the fact that this is completely free and open software, opens itself up for some personal customization. As I did with the restaurant, you can create your own logo, or just put it in rotation with the existing ones (Figure 9). The images are in a subdirectory called img. Logos are in a further subdirectory called logos.

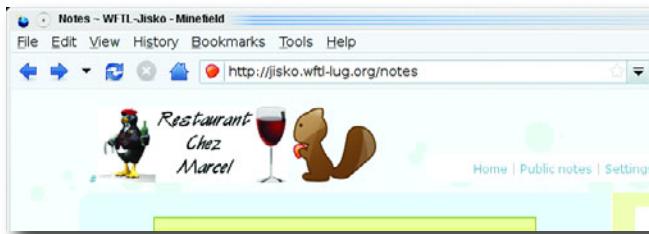


Figure 9. Why not customize Jisko with your own company or organization's logo?

Once you have created your image, you need to tell Jisko about it, which simply means another entry in the config.php file. Look for the following:

```
$globals['logo'] = array('logo_green.png',
->'logo_blue.png', 'logo_orange.png');
```



ASA COMPUTERS

Want your business to be more productive?
The ASA Servers powered by the Intel Xeon Processor provide 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.)

1U Server - ASA1401i

- 1TB Storage Installed. Max - 3TB.
- Intel Dual core 5030 CPU (Qty-1), Max-2 CPUs
- 1GB 667MGZ FBDIMMs Installed.
- Supports 16GB FBDIMM.
- 4X250GB htswap SATA-II Drives Installed.
- 4 port SATA-II RAID controller.
- 2X10/100/1000 LAN onboard.

2U Server - ASA2121i

- 4TB Storage Installed. Max - 12TB.
- Intel Dual core 5050 CPU.
- 1GB 667MGZ FBDIMMs Installed.
- Supports 16GB FBDIMM.
- 16 port SATA-II RAID controller.
- 16X250GB htswap SATA-II Drives Installed.
- 2X10/100/1000 LAN onboard.
- 800w Red PS.

3U Server - ASA3161i

- 4TB Storage Installed. Max - 12TB.
- Intel Dual core 5050 CPU.
- 1GB 667MGZ FBDIMMs Installed.
- Supports 16GB FBDIMM.
- 16 port SATA-II RAID controller.
- 16X250GB htswap SATA-II Drives Installed.
- 2X10/100/1000 LAN onboard.
- 800w Red PS.

5U Server - ASA5241i

- 6TB Storage Installed. Max - 18TB.
- Intel Dual core 5050 CPU.
- 4GB 667MGZ FBDIMMs Installed.
- Supports 16GB FBDIMM.
- 24X250GB htswap SATA-II Drives Installed.
- 24 port SATA-II RAID. CARD/BBU.
- 2X10/100/1000 LAN onboard.
- 930w Red PS.

8U Server - ASA8421i

- 10TB Storage Installed. Max - 30TB.
- Intel Dual core 5050 CPU.
- Quantity 42 Installed.
- 1GB 667MGZ FBDIMMs.
- Supports 32GB FBDIMM.
- 40X250GB htswap SATA-II Drives Installed.
- 2X12 Port SATA-II Multilane RAID controller.
- 1X16 Port SATA-II Multilane RAID controller.
- 2X10/100/1000 LAN onboard.
- 1300 W Red Ps.

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

ASA Computers
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 typographic errors.

intel
Xeon inside™
Powerful.
Efficient.



Figure 10. Sweetter supports plugins that allow it to update both Twitter and Jaiku.

The last package I want to look at is Sweetter (Figure 10). I hesitate to call it just another MB system, because it does differ in some interesting ways. Unlike the others, it does not require an Apache server, nor does it require MySQL (although you can use it if you like). Sweetter can run independently, from a personal account, on a port of your choosing, such as 8080 (the default). If you aren't expecting a huge load on your MB service and you are keeping it between friends or coworkers, you may find that the built-in SQLite database is all you need.

Sweetter also supports plugins that allow you to post updates not only to Sweetter, but to Twitter and Jaiku as well. This feature isn't always on, however. If you want your posts to hit multiple services, click the Show plugins link directly below the update box on the main page. Authentication fields for Twitter and Jaiku will appear. Make sure you click the check box beside each service you use before clicking Send.

Here's another Sweetter-unique feature. Aside from your usual MB posts, Sweetter also is a kind of on-line to-do list. Be careful how you post though. Sweetter may complain that you are using it like an instant-messaging service, and it doesn't like that.

There's something about Sweetter that says it's more for fun than serious microblogging (if such a thing is possible). That would be the system of voting on other users' posts. Each time you post an update, others can vote that post up or down. You can see a plus and minus sign below each one. Once a vote has been cast, it goes toward your Karma, in both a positive and negative sense. Higher Karma is, of course, what you want. Aside from being popular with the voters, your output also is taken into account. More updates mean more Karma.

All three of these services are worthy of your



Figure 11. Each Sweetter update is subject to a vote by other readers. The number of posts you make and the votes you receive all go toward your Karma.

attention, and all of them have something to offer that makes them unique. What you wind up using will depend on what you find most useful.

With closing time nearly upon us, we still have plenty of time to chat with, and otherwise update, our many friends and family on this superb wine. After all, you can send a great many people a number of updates in the time we have left. That's the beauty behind staying within that 140-character limit—short and sweet. When it comes to wine, however, I'm sure no one wants to make it short and sweet. Wine is meant to be savored, like friendship. And, when you savor a glass of wine with friends, it's twice as wonderful. François, please make sure our friends' glasses are all refilled before we say, *Au revoir*. Please, *mes amis*, raise your glasses and let us all drink to one another's health. *A votre santé! Bon appétit!*

Marcel Gagné is an award-winning writer living in Waterloo, Ontario. He is the author of the *Moving to Linux* series of books from Addison-Wesley. 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 marcel@marcelgagne.com. You can discover lots of other things (including great Wine links) from his Web sites at www.marcelgagne.com and www.cookingwithlinux.com.

Resources

Jisko: jisko.org

Locanica: laconica.ca

Sweetter: sweetter.net

Marcel's Web Site: www.marcelgagne.com

Cooking with Linux: www.cookingwithlinux.com



Systems



ZT Systems delivers something different:
a unique

BALANCE

of world-class server performance and cost advantage
joined with extensive flexibility in delivery and support.

ZT Systems combines the design and manufacturing prowess of a world-class server manufacturer with layers of flexibility to give you the customized server solutions you need fast – all at very competitive prices.

- Advanced high-performance servers with outstanding reliability.
- Platform customization that meets your exact hardware and imaging needs.
- Extensive customized programs that support your ever-changing mission without being over-engineered.



ZT Systems 1401Ti-84 Twin Node Datacenter Server
featuring the Intel® Xeon® Processor 5400 Series

Scalable, Customized Data Center Server Solutions
www.ztsystems.com/inteldatacenter





DAVE TAYLOR

Pushing Your Message Out to Twitter

Use the shell to generate movie trivia from a movie database.

Holy cow, have we really been working on the movie trivia Twitter stream for almost a year now? This surely must be the longest time-per-line-of-code project in the history of software development.

Previous columns have combined to give us a set of shell scripts that scrape the Internet Movie Database (IMDb, www.imbd.com) for its top 250 list, then pull out the year of release for each film (inconveniently, not available on the same page as the top 250 list), randomly select reasonable incorrect guesses for release year and output the question in the form: "IMDB Top 250 Movie #81: Was 2001: A Space Odyssey released in 1968, 1973 or 1975?"

That's rather more work behind the scenes than you might realize. Of course, if you've been reading all the previous columns, you actually have firsthand knowledge of just how many hoops we've had to

It turns out, there's a simple way to invoke Twitter through a command-line Web utility and get access to much of the basic data.

jump through (but fortunately, no "hogsheads of real fire").

This month, we're going to turn our attention to the Twitter microblogging service, as we finally have the ability to produce the desired message. Now, we just need a mechanism to publish it to Twitter.

Hacking the Twitter API

I could take the long way and actually read through the Twitter API to learn how to make specific calls and interact with the service elegantly and appropriately, but that sounds like work, doesn't it?

It turns out, there's a simple way to invoke Twitter through a command-line Web utility and get access to much of the basic data. My tool of choice? Curl, a slick utility that makes it easier to work with Internet services through the command line.

The URL is twitter.com/statuses/update.json, but the way you pass data is a bit tricky. You need to send it as a name=value pair in the connection stream, not as a GET value or other URL-based

technique. The variable it seeks is status, and you can test it by doing this:

```
$ curl --data-ascii status=testing \
http://twitter.com/statuses/update.json
```

The problem is immediately obvious in the return message:

```
Could not authenticate you.
```

Ah, well, yes, we haven't specified our user ID. Those credentials aren't sent via the URL or --data-ascii, but instead through a basic HTTP auth. So, in fact, you need to do this:

```
$ curl --user "$user:$pass" --data-ascii \
status=testing "http://twitter.com/statuses/update.json"
```

Now, of course, you need a Twitter account to utilize. For this project, I'm going to continue working with FilmBuzz, a Twitter account that I also set up to disseminate interesting film-industry news.

For the purposes of this article, let's assume the password is DeMille, though, of course, that's not really what it is. The user and pass variables can be set, and then we invoke Curl to see what happens:

```
$ curl --user "$user:$pass" --data-ascii status=testing
http://twitter.com/statuses/update.json
{"truncated":false,"in_reply_to_status_id":null,"text":
"\"testing\"","favorited":null,"in_reply_to_user_id":null,
"source":"web","id":880576363,"user": {"name":
"\"FilmBuzz\"","followers_count":214,"url":null,
"profile_image_url":"http://s3.amazonaws.com/
/twitter_production/profile_images/55368346/
/FilmReelCloseUp_normal.JPG","description":"\"Film
trivia game, coming soon!\"","location":"Hollywood, of
course","screen_name":"FilmBuzz","id":15097176,"protected":false},
"created_at":"Thu Aug 07 16:51:49 +0000 2008"}
```

The return data gives you a nice sneak peek into the configuration of the FilmBuzz twitter account and tells you that what we sent wasn't too long (truncated:false), but otherwise, it's pretty forgettable stuff, isn't it?

How do you ignore output with a command in the shell? Reroute the output to /dev/null, of course:

```
$ curl --user "$user:$pass" --data-ascii status=testing
http://twitter.com/ses/update.json > /dev/null
% Total % Received % Xferd Average Speed Time Time Current
Dload Upload Total Spent Left Speed
100 505 100 505 0 0 1127 0 --::-- --::-- --::-- 0
```

Nope, that doesn't work, because then Curl wants to give us some transactional stats. How do we mask that? Use the --silent flag to Curl. Now, we're just about there:

```
$ curl --silent --user "$user:$pass" --data-ascii status=testing
http://twitter.com/statuses/update.json > /dev/null
$
```

That's it. Let's put that into a shell script, so we can simply invoke tweet with the message we want to send out to followers of FilmBuzz. (And, if you aren't yet following FilmBuzz on Twitter, why the heck not? Go to twitter.com/FilmBuzz, and click follow.) Actually, we need to take into account one more thing: although our messages on the command line are quite likely to have spaces within them, we can't send a name=value pair with spaces. Instead, a quick invocation of tr lets us convert all spaces into + signs, the commonly accepted HTTP encoding:

```
#!/bin/sh
# Twitter command line interface
user="FilmBuzz" ; pass="DeMille"
curl="/usr/local/bin/curl"

$curl --silent --user "$user:$pass" --data-ascii \
"status=$(echo $@ | tr ' ' '+')" "http://twitter.com/
➥statuses/update.json"
> /dev/null
exit 0
```

This isn't a particularly robust solution, because what happens if we have a + character in the message we want to transmit? It gets lost. That can be addressed by first checking to see whether there are + signs and converting them to the safe HTTP-encoding

equivalent, %2B. You can't do that with tr, however, because it wants 1:1 substitution, so we'll use sed instead and pull the substitution onto its own line for better style too:

```
msg=$(echo $@ | sed 's/+/%2B/g;s/ /+/g')

$curl --silent --user "$user:$pass" --data-ascii \
"status=$msg" "http://twitter.com/statuses/update.json" > /dev/null
```

Problem solved. Now you can do simple things like:

```
$ tweet 'Tweeting from the command line? Well, sure!'
```

Nice. Now, to put everything together. Oh. We've run out of space. Again. Okay, next month. We finally have all the building blocks we need. Remember, sign up for Twitter, and then follow @FilmBuzz, so you can see the fruit of this work. ■

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, and he also offers up tech support at AskDaveTaylor.com. Follow Dave on Twitter through twitter.com/DaveTaylor.

On the Web, Articles Talk!

Every couple weeks over at LinuxJournal.com, our Gadget Guy Shawn Powers posts a video. They are fun, silly, quirky and sometimes even useful. So, whether he's reviewing a new product or



showing how to use some Linux software, be sure to swing over to the Web site and check out the latest video: www.linuxjournal.com/video.

We'll see you there, or more precisely, vice versa!

TECH TIP Finding Which RPM Package Contains a File

To search a list of RPM files for a particular file, execute the following command:

```
$ ls RPMS-TO-SEARCH | \
xargs rpm --query --filesbypkg --package | \
grep FILE-TO-SEARCH-FOR
```

Replace RPMS-TO-SEARCH with the names of the RPM files to search, and replace FILE-TO-SEARCH-FOR with the name of the file to search for. The --filesbypkg option tells the rpm command to output the name of the package as well as the name of the file.

—DASHAMIR HOXHA



MICK BAUER

Samba Security, Part I

Build a secure file server with cross-platform compatibility.

It recently occurred to me that in the eight years or so I've been writing this column, I've never covered file servers. I've covered secure file transfer, for example, via scp, rsync and vsftpd, which certainly is important. But, I've not covered file serving, specifically, allowing users to mount persistent "network volumes" that let them use networked server disk space as though it were a local disk. This has all sorts of productivity- and operations-related benefits: it's (usually) easy for end users to use, and it makes data easier to access from multiple systems and locations and easier to back up and archive.

As it happens, there's a rich toolkit available to Linux users for building, securing and using file servers, mainly in the form of Jeremy Allison and Andrew Tridgell's Samba suite of daemons and commands, plus various graphical tools that supplement them. For the next few columns, I'm going to show you how to build a secure Samba file server using both command-line and GUI tools.

Does that sound like a good Paranoid Penguin project? Good enough, I hope, to forgive me for ignoring file servers for so long. (So many things to secure, so little time!)

What We Want to Achieve

Obviously, no series of articles can cover everyone's file server needs or wants. So, before I begin, let's agree on some requirements of my

With a file server, I'm going to pay particular attention to protecting the data itself: the integrity, availability and confidentiality of my files, while at rest on the server and in transit over the network.

choosing (hopefully, some or all of these coincide with yours). It seems reasonable to focus on the following: security, convenience and cross-platform compatibility.

It goes without saying that in a security project, security is foremost among my preoccupations. With a file server, I'm going to pay particular attention to protecting the data itself: the integrity, availability and confidentiality of my files, while at rest on the server and in transit over the network. I also

want to protect the server itself, both to protect the data and to prevent the server from being misused in other ways.

The trio of goals I listed above (confidentiality, integrity and availability) is part of classic information security dogma. In just about any information security scenario you can think of, C, I and A are important one way or another.

The goal of simply preventing a system from being used in unexpected or unwanted ways by unauthorized persons though, is what I like to call exclusivity. If I go to the trouble of building and maintaining a file server, even if the data itself is 100% boring and useless (please do not insert a joke about Paranoid Penguin column archiving here), I want such a server to be used *exclusively* by me and the users I specifically designate.

Even if it's some sort of public file server (for which, by the way, asynchronous file transfer technologies, such as FTP, HTTP and rsync are *much* more securable than Samba), I still want that server to be used *exclusively* for that purpose. I don't want it being used as someone's pirate IRC server, warez repository or proxy for attacking other systems.

I just mentioned that the type of file server I'm talking about (the kind to which you can "map drives") isn't suitable for public file serving. This is because the two dominant tools for this, Samba and NFS, historically have relied on RPC, a protocol that involves the dynamic assigning of TCP and UDP listening ports on a per-client, per-connection basis, which requires a large range of ports to be opened through any firewalls that might be in the way. Alas, opening UDP and TCP ports 1025 through 65,534 in both directions through a firewall is an awful lot like not using a firewall at all, even if you limit source or destination IP addresses.

On the one hand, more current versions of NFS (versions 3 and 4) allow the server/daemon to use a single TCP port for all connections by concurrent users. However, much of the world seems to be stuck on NFS v2. Worse still for our purposes here, there never has been good support for NFS outside the world of UNIX and UNIX-like platforms.

And, this brings us to our other two requirements: convenience and cross-platform compatibility.

Samba over the Internet?

If Samba is so very convenient, you may wonder, why not use IPsec or some other VPN/encryption tool to secure its use on the Internet? This is actually possible. As it's been a while since I covered IPsec in this space, and in the intervening years, IPsec support has been added to the Linux kernel, I just may end this series with a quick tutorial on doing Samba over IPsec.

However, Samba is a very "chatty" protocol—it generates a lot of packets even if you're using it only for small files or shares. This causes problems not so much for your Internet link as for Samba performance: Samba can be very sensitive to dropped or delayed packets, which is more likely in your modestly sized Internet uplink than over your exponentially bigger Local Area

Network (LAN) fabric.

So, trying to get Samba working over IPsec may or may not be worth your time, and it may or may not warrant my covering it in this series of columns. Have no fear, one way or another, you can expect me to provide a tutorial on using the Linux kernel's IPsec functionality in a future Paranoid Penguin column.

I want to be able to map network drives/volumes, because this is much, much more convenient than manually copying files to and from the file server every time one changes, even automatically via some script. I want a file share that allows me to "work" on files that "reside" in a central location; I don't want working copies of my files being maintained on umpteen different systems.

I've alluded to the fact that NFS allows you to map network volumes in a very similar way as with Samba. However, I have to acknowledge the ugly reality that I am sometimes required to operate Windows systems. My job requires it and so does my video-gaming habit. So, I need a file server that supports both Linux and Windows clients. (As it happens, the one we're going to build also will support FreeBSD, NetBSD, Solaris, Mac OS X and practically the entire rest of the *nix world!)

To summarize, we're going to build our file server with Samba, because it's convenient and it supports different client platforms. And, we're going to build it as securely as possible.

Samba Security Terms and Concepts

I've explained in gory detail why Samba, firewalls and the Internet don't go well together. So, how do you secure Samba for LAN use?

Samba security is a surprisingly complex topic, which is why this is a multipart article. You've got many choices to make if you want to use Samba securely. Is your Samba server also going to be an NT domain controller, or will it participate in an existing domain or workgroup? Will you permit guest access, or will all users of every share need to be authenticated first? Or, will you allow both private and public shares?

Don't worry if the previous questions make little sense to you. That's why the Paranoid Penguin is here. For most of the rest of this

article, I discuss these concepts in detail. Only then will you be ready to explore the mysteries of smb.conf and the NMB daemon.

First, let's get some definitions out of the way:

- **SMB:** the Server Message Block protocol, the heart of Samba. SMB is the set of messages that structure and use file and print shares.
- **CIFS:** short for the Common Internet File System, which in practical terms is synonymous with SMB.
- **NetBIOS:** the API used to pass SMB messages to lower-level network protocols, such as TCP/IP.
- **NBT:** the specification for using NetBIOS over TCP/IP.
- **WINS:** Microsoft's protocol for resolving NBT hostnames to IP addresses; it's the MS world's answer to DNS.
- **Workgroup:** a peer-to-peer group of related systems offering SMB shares. User accounts are decentralized—that is, maintained on all member systems rather than on a single central server.
- **NT domain:** a type of group consisting of computers, user accounts and other groups (but not other domains). It is more complex than a workgroup, but because all domain information is maintained on one or more domain controllers rather than being distributed across all domain members, domains scale much better than workgroups.
- **Active Directory:** Microsoft's next-generation domain system. Samba can serve as an Active

Directory client via Kerberos, but you can't control an Active Directory tree with a Samba server as you presently can do with NT domains. Active Directory server support will be introduced in Samba v4.

- User-mode security: when a Samba server's shares are authenticated by local workgroup user names and passwords.
- Share-mode security: when each share on a Samba server is authenticated with a share-specific password that isn't explicitly associated with a user name.
- Guest access: when a Samba server allows anonymous connections to a given share via a shared guest account with no password.

Here's what you need to take away from that list of definitions.

First, the protocols. SMB, aka CIFS, is the protocol that defines the network filesystem—its structure and its use. NetBIOS provides an API through which SMB messages may be transmitted.

If Samba is so very convenient, you may wonder, why not use IPsec or some other VPN/encryption tool to secure its use on the Internet?

ted over networks, and which may be used by servers to "advertise" services and by clients to "browse" those services. NetBIOS can use any of a number of lower-level network protocols as its transport, but the most important of these is TCP/IP; NetBIOS over TCP/IP is called NBT. WINS provides centralized name services (mappings of hostnames to IP addresses), where needed.

Next, server roles. A Samba server can authenticate its transactions either on a per-share basis, using share-specific passwords and inferred/implicit user names, or on a per-user basis, using either a dedicated local user database (in user-mode security) or some networked authentication scheme, such as LDAP, NIS, NT Domains or Active Directory. The server can be in a workgroup, in which case it needs to maintain its own database of all the workgroup's user information, or it can be in an NT Domain or an Active Directory, in which all user information is managed centrally.

When you want to share data with maximum

convenience and minimum security, for example, read-only files containing nonsensitive data, you can put it on a share with guest access. Users connecting to such a share will not be prompted for any user name or password.

The bad news is that this is only a fraction of what you need to know in order to understand SMB/Samba services. The good news is, NT Domains and Active Directory are out of scope for this series of articles. We're going to focus on using workgroups for our secure Samba file server.

Workgroups don't scale well, because each server in a workgroup needs to maintain all user information for the entire workgroup, and you must somehow keep this information (passwords and so forth) consistent across all workgroup members (except where only guest access or share-mode access is permitted).

However, for the usage scenario I've described—creating a file share or two I can reach from anywhere in my house—I'm not going to have very many users or even more than one server, necessarily, and the simplicity of setting up a standalone/workgroup server trumps the complexity-laden power that comes with NT Domains. If your needs differ, hopefully this series of articles nonetheless will make it easier for you to figure out Samba's NT Domain support on your own, if that's what you really need instead.

So, to express our project in the terms I've just defined, in subsequent articles, I'm going to walk through the process of configuring a standalone (workgroup) Samba server operating with user-mode security, using a dedicated local user database. Our example server will host a combination of guest shares, read-only shares restricted by user and shares that can be read by only some users, but that can be written to (changed) by others.

First though, we have to make sure you've got the software you need in order to pull that off.

Getting Samba Software

On your Samba server, you're going to need your distribution's packages for Samba's libraries; the Samba daemons smbd, nmbd and winbind; the Samba client commands smbclient, smbmount and so forth (which are useful even on servers for testing Samba configurations); and also the Web-based configuration tool SWAT (Figure 1). Naturally, nearly all these things are contained in packages whose names don't correspond neatly with the names of their component daemons, libraries and so forth, but I give some pointers on those shortly.

First, a word about SWAT, which requires a modest security trade-off for Ubuntu users.

Although normally in Ubuntu the user root can't log in directly, Samba requires this to be possible, so you need to set a root password on any Ubuntu box that runs SWAT.

Like so much else about Samba, this is not something I recommend doing on any Internet-facing Ubuntu box. However, SWAT is such a useful and educational tool, I feel pretty confident in stating that in non-Internet-facing environments, the mistakes SWAT will help you avoid probably constitute a bigger threat to system security than SWAT does.

As I mentioned, Samba packages are included in all major Linux distributions. In Debian and its derivatives, such as Ubuntu, you'll want to install the following deb packages: samba, samba-common, samba-doc, smbclient and swat (plus whatever packages you need to satisfy dependencies in any of these).

In SUSE, you'll want to install samba, samba-client, samba-winbind and samba-doc. (SWAT is included with one of these, probably samba.)

In Red Hat Enterprise Linux and its derivatives, you need samba, samba-client, samba-common and samba-swat.

Installing these binary packages should involve installation scripts that put startup scripts, symbolic links and so forth in the correct places for everything to work (at least, after you configure Samba to serve something). Using SWAT is the best way to get up and running quickly—not because it does very much work for you, but because its excellent help system makes it super-convenient to summon the pertinent parts of Samba's various man pages.

There are two SWAT quirks I should mention. First, SWAT must be run by an Internet super-server, such as the old Berkeley inetd or the newer xinetd. Ubuntu configures inetd automatically when you install the swat package, but if your distribution of choice does not, you need a line like this in /etc/inetd.conf:

```
swat stream tcp nowait.400 root /usr/sbin/tcpd /usr/sbin/swat
```

Second, to get SWAT's help links to work under SUSE 11.0, you may need to create the following symbolic links while logged in to a terminal window as root:

```
ln -s /usr/share/doc/packages/samba/htmldocs/manpages  
➥/usr/share/samba/swat/help  
ln -s /usr/share/doc/packages/samba/htmldocs/using_samba  
➥/usr/share/samba/swat/help  
ln -s /usr/share/doc/packages/samba/htmldocs/index.html  
➥/usr/share/samba/swat/help  
ln -s /usr/share/doc/packages/samba/htmldocs/manpages.html  
➥/usr/share/samba/swat/help
```



Figure 1. SWAT

Conclusion

And with that, we're ready to start configuring our Samba server! Or we would be, if we weren't out of time and space for this month. The links in the Resources section, not to mention SWAT's aforementioned excellent help links, should help you get started before we continue this series in my next column. Until then, be safe! ■

Mick Bauer (darth.elmo@wiremonkeys.org) is Network Security Architect for one of the US's largest banks. He is the author of the O'Reilly book *Linux Server Security*, 2nd edition (formerly called *Building Secure Servers With Linux*), an occasional presenter at information security conferences and composer of the "Network Engineering Polka".

Resources

Christopher R. Hertel's On-Line Book
Implementing CIFS, a Comprehensive Source of Information on All Things CIFS/SMB-Related:
www.ubiqx.org/cifs

"The Official Samba 3.2.x HOWTO and Reference Guide": us1.samba.org/samba/docs/man/Samba-HOWTO-Collection

Did you know Linux Journal maintains a mailing list where list members discuss all things Linux? Join LJ's *linux-list* today: <http://lists2.linuxjournal.com/mailman/listinfo/linux-list>.



KYLE RANKIN

Memories of the Way Windows Were

Still arranging windows by hand? Learn how to take advantage of Compiz's window memory, so it can do the work for you.

I'm a half-organized person. On one hand, if something of mine has a place, I can be pretty anal about making sure I put it back every time I use it. On the other hand, if something doesn't have a place, it inevitably ends up in a pile or a junk drawer. I've learned that if I want to be organized, I must give everything a home.

The same rule applies to my desktop environment. Back when I used to use Windows, I didn't have much of a choice—everything ended up stacking up on the same desktop, either maximized or at some arbitrary size. Once I started using Linux though, I discovered this interesting multiple desktop model. With Linux, I could assign windows into certain groups and then arrange each group on a particular desktop. The main downside to this much

With window memory, every window I use on a regular basis can be assigned a location, a size and a desktop.

organization was that every time I opened a window, I usually needed to resize it and move it to a particular desktop. That's a lot of manual work on my part, and it wasn't long before I discovered that certain window managers supported window memory. With window memory, every window I use on a regular basis can be assigned a location, a size and a desktop.

My History with Window Memory

My first exposure to window memory was with the Enlightenment window manager. Its window memory was quite easy to use and to set. All you had to do was right-click on a window, and you could check off attributes that Enlightenment would remember the next time you opened the window. In addition to having certain sets of terminals and Web browser windows open on certain desktops, I also was able to have windows always stay on top or stick across all desktops. Although it did require a little setup, by the time I was finished arranging my windows once, everything I used on a regular basis had its place

on my desktops.

I stayed with Enlightenment for quite some time, even though I was eyeing this new window manager called Fluxbox as a potential replacement. It wasn't until Fluxbox added window memory, however, that I made the switch. Fluxbox's window memory worked a lot like Enlightenment's—right-click on a title bar and toggle the attributes you want to remember. As with Enlightenment, these attributes were assigned based on the window title, so if you had two windows with the same title (say, xterm, with no extra arguments), they both would take those same settings.

I used both Enlightenment and Fluxbox for years, but I kept eyeing the GNOME and KDE desktops all the cool kids were using. For me, window memory was the crucial requirement though, and it wasn't until I made the switch to using Ubuntu that I decided to give one of the "standard" desktop environments a fair shake. Out of the box, it didn't seem like Compiz had any window memory, and this was a major strike against it in my book. However, almost a year later, I still am using Compiz, and I have to credit the advanced window memory that I discovered buried in advanced Compiz settings for keeping me here.

CompizConfig Settings Manager

By default, at least in Ubuntu, there are only so many settings you can tweak in Compiz. Compiz provides a tool, however, called CompizConfig Settings Manager (or CCSM) that gives you very detailed control over many different aspects of both Compiz eye-candy effects as well as a lot of the important settings for the window manager itself. The major downside to CCSM, however, is that there are almost too many options—if you don't know exactly what you are looking for, expect to spend some time digging through different categories. Even window memory settings are split between two different categories.

Ubuntu didn't install CCSM automatically for me, but I was able to install it with a quick trip to the package manager, and it should be packaged for other distributions that include Compiz. Once it is

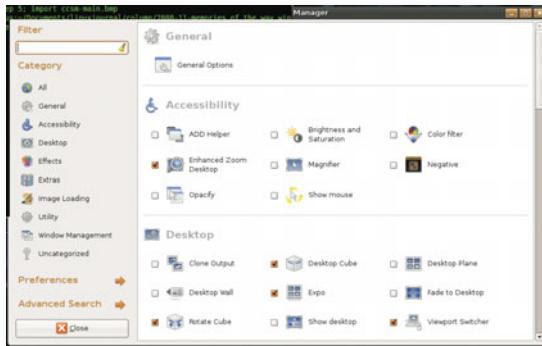


Figure 1. Default CCSM Window

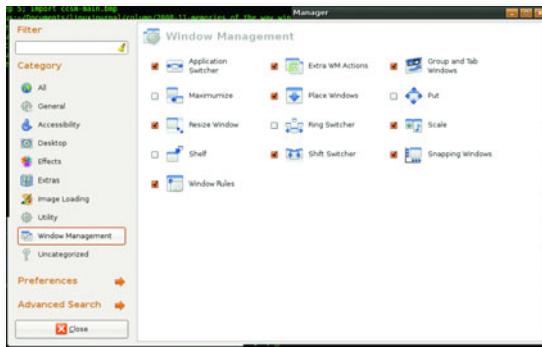


Figure 2. Window Management Configuration Options

installed, you either can type `ccsm` in a terminal window or click System→Administration→Advanced Desktop Effects Settings. As I mentioned, the default window can be a little daunting (Figure 1) and is split into a narrow left pane that displays the categories and a larger right pane that shows all the particular settings you can configure for the category.

Everything you need to configure window memory in `ccsm` can be found in the Window Management category, and once you click on that category in the left pane, you drill down into a much more manageable set of options (Figure 2). For some reason, `ccsm` splits window memory into two different sets of options, Window Rules and Place Windows. The Window Rules options allow you to configure window attributes like stickiness and geometry, and the Place Windows options let you control the viewport and location where the window is placed.

Window Rules

Once you click the Window Rules icon, you will see the first set of window attributes Compiz can remember, split into tabs for Matches and Size Rules. Matches (Figure 3) contains standard window attributes, such as Above, Below, Sticky and

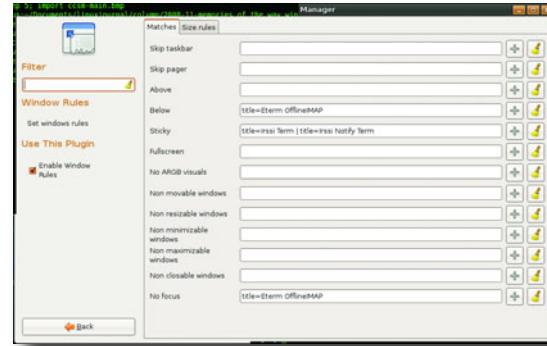


Figure 3. Window Matching Rules

Fullscreen, that you could set on a window manually, as well as many options you can't, such as non-movable, non-resizable, non-maximizable and no focus. To assign one of these attributes to a window, you need to add some sort of identifier in the field next to the attribute.

Compiz Window Identifiers

Compiz can match windows based on a number of different attributes documented at wiki.compiz-fusion.org/WindowMatching, such as window type, role, class, title, xid and state—all of which you can find out about with the `xprop` command-line utility. Simply run `xprop`, and then click on the particular window for which you want information. Even though there are lots of possible attributes to match, probably the easiest attribute to use is the window title. To figure out a window's title, either view its title bar, or alternatively, run:

```
xprop WM_NAME | cut -d\'' -f2
```

and then click on the window of interest. Compiz doesn't necessarily need the full title of the window, just some identifying information. So for instance, if you want Mozilla Firefox to be sticky, you could add `title=Mozilla Firefox` to the Sticky option, or you simply could add `title=Firefox`.

You also can add multiple window attributes to each of these fields and separate them with a `|` for "or" or `&` for "and". So if I wanted both xterms and aterms to be sticky, I would add the following to the sticky field:

```
title=xterm | title=aterm
```

Note that I use the `or` operator. If I had used the `and` operator, only windows with both `xterm` and `aterm` in their titles would be sticky.

You even can use basic regular expressions to match windows (so `^` and `$` would match the

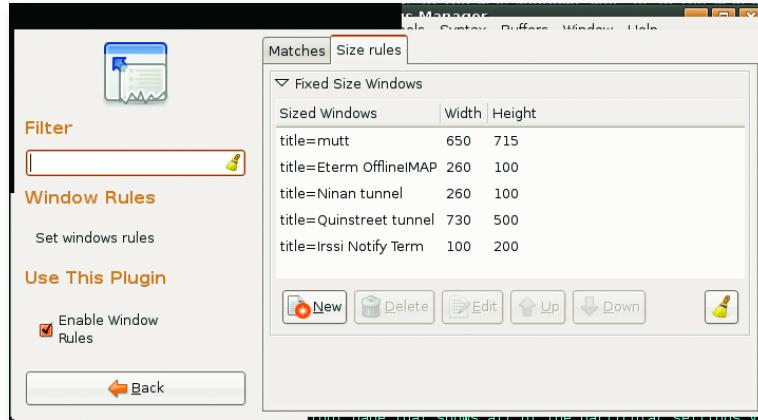


Figure 4. Window Size Rules

beginning and the end of a string, respectively), along with more advanced nested expressions. All of these more advanced options are documented on the Compiz Wiki page mentioned above.

The second tab in the Windows Rules window is labeled Size Rules and allows you to force windows to a particular geometry (Figure 4). The configuration is pretty straightforward. Click New to add a new size rule, input the attribute to match your particular window (in Figure 4, I match only on title), and then add the width and height in pixels for that window. Once you finish your changes, click the Back button at the bottom of the left pane to return to the main ccsm screen.

Place Windows

The CompizConfig Settings Manager makes a logical distinction between window geometry and settings and actual window placement. Click Place Windows from the main ccsm screen, and you will

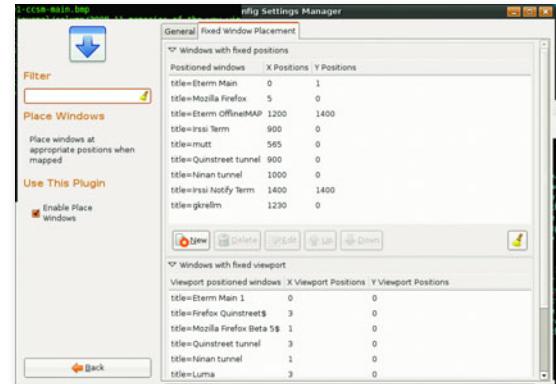


Figure 5. Window Placement Configuration Window

You also can configure a window to appear only in a particular viewport. Again, you can match on window attributes and then specify the X and Y viewport positions to use. In my case, I have a desktop that is five viewports wide and only one viewport high, so my Y viewport setting was always set to 0, and I could choose between 0–4 for my X viewport.

Now that I have all of these window management options, how do I use them? Well, I always have liked my first desktop to be for normal terminals, so I have my default terminals always open there at a particular size. I also have a different terminal I use exclusively for IRC. I want that to be available on all desktops, so I set that to sticky and also move it to a particular position on the desktop. I like my second desktop to be for Web applications, such as Firefox, so I configure them for that viewport. I also like to segregate terminals, browsers and IM clients I use exclusively for work, so I have all of those open automatically on a special desktop.

Other options you could consider might be to move GIMP and all of its windows to a special desktop. I also run a number of scripts in the background that do things such as sync my e-mail to a local directory. These scripts actually open their own small terminal in the background, so I move them to a desktop specifically created for them and also have them configured to not steal focus. I find I naturally end up assigning window memory for windows once I get tired of positioning them every time. The time it saves me in the long run makes up for the initial configuration, and it also saves me from always Alt-tabbing through the junk drawer that is the default user desktop. Most important, it helps keep this Linux user organized—at least half the time.■

Kyle Rankin is a Senior Systems Administrator in the San Francisco Bay Area and the author of a number of books, including *Knoppix Hacks* and *Ubuntu Hacks* for O'Reilly Media. He is currently the president of the North Bay Linux Users' Group.



GENSTOR

"Company With A Vision"



GNS 2450 *Green* Storage Appliance

Green
Scalable
Reliable
Performance
Plug & Play

Genstor Systems GNS 2450

is built on fully optimized Highly Efficient dual-core AMD Opteron™ processor model 2214 HE running Open-E® DSS™ (Data Storage Server).

GNS 2450 is an all-in-one IP-Storage with NAS and iSCSI (both target and initiator) functionality. Excellent price performance value, enhanced management and superior reliability is great for organizations of all sizes.

► Hardware & Software

- One or two Highly Efficient dual-core AMD Opteron™ processor-based platform
- 8GB RAM (Expandable to **64GB**)
- Dual GbE NICs (Optional 10GbE)
- 2U rackmount chassis with 12 SATA HDD bays with redundant power supplies
- Open-E® DSS™ (Data Storage Server) with NAS and iSCSI functionality

► Features & Benefits

- Administration:
Powerful, secure and easy intuitive GUI for configuration and remote management
- Network Management:
Along with standard Network Gateway support, the **GNS 2450** also supports DHCP Client, Multiple NIC, Adapter Fault Tolerance (AFT), Adaptive Load Balancing (ALB), Proxy Settings, IP-Sec, etc.
- Flexibility & Scalability:
The **GNS 2450** storage appliance is highly scalable and flexible to take care of your future expandability needs

Price: \$5200 (for 6TB raw storage)

GENSTOR SYSTEMS, INC.
780 Montague Expressway # 604
San Jose, CA 95131
Phone: 408-383-0120
Fax: 408-383-0121



Please contact Genstor sales@genstor.com for your custom hardware needs

1-877-25-SERVER • www.genstor.com • sales@genstor.com

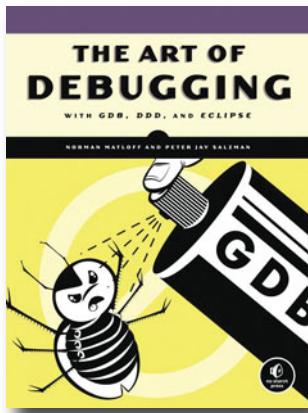
Open-E, Open-E DSS and the Open-E logo are registered trademarks of Open-E, Inc.

AMD, AMD Arrow logo, and AMD Opteron are trademarks or registered trademarks of Advanced Micro Devices, Inc.

Super Talent's Pico D USB Drive

The folks at Super Talent have added a new model, the Pico D, to its line of ultra-diminutive USB Flash drives. The Pico series of USB drives, measuring in at 1.4 inches, is an inch shorter than most USB drives on the market today, says the company. Other features include a pivoting lid that won't get lost, shock and water resistance and transfer speeds up to 30MB/sec. Not to mention that the little guy is kinda cute too.

www.supertalent.com



Matloff and Salzman's *The Art of Debugging with GDB, DDD, and Eclipse* (No Starch)

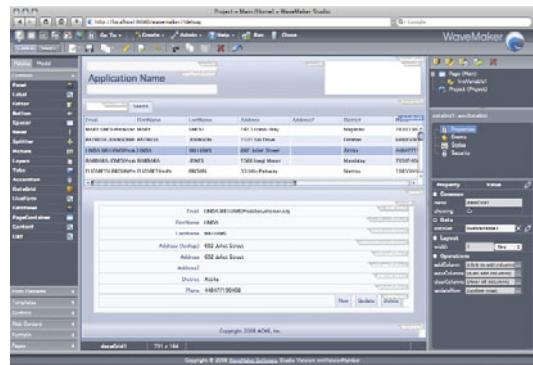
Despite the importance of debugging, many new programmers are unaware of techniques that reduce the time needed to find and fix programming errors. To the rescue is the new book *The Art of Debugging with GDB, DDD, and Eclipse* by Norman Matloff and Peter Jay Salzman, published by No Starch Press. Leveraging the open-source development tools GDB, DDD and Eclipse, the book illustrates an assortment of real-world coding errors—from simple typos to major logical blunders—to discuss how to manage memory, understand core dumps and trace programming errors to their root cause. The book also covers topics that the publisher says other debugging books omit, such as threaded, server/client, GUI and parallel programming.

www.nostarch.com

WaveMaker's Visual Ajax Studio 4.0

As the Ajax wave surges on, one of its well-known providers, WaveMaker, is adding new Ajax-related tools, such as Visual Ajax Studio 4.0. The product is an open-source development tool that "makes it easy to build visually stunning Web applications". The company claims that "with just 15 mouse clicks and zero coding, a developer can build and deploy a sleek, Web-based application". The new 4.0 release offers faster development time, better-quality applications, the company's own Live Layout data display, enhanced drag-and-drop capabilities and IDE-quality editing that exposes the source code and offers syntax highlighting. Visual Ajax Studio 4.0 supports Linux, Mac OS X 10.5 and Windows XP and Vista.

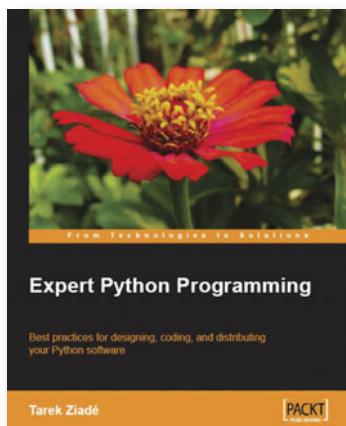
www.wavemaker.com



Perforce Software's SCM System

Perforce 2008.1 is the latest release of Perforce Software's software configuration management (SCM) system, which versions and manages source code and all digital assets. The 2008.1 release extends visual differencing functionality to images, enabling enterprise developers and engineers to manage all content with one SCM solution. Image differencing supports most common image files, including TIFF, JPG and GIF, and can be extended to support other image formats through the Qt API. Another new feature is improved remote access, which is accomplished via the Perforce Proxy, a self-maintaining proxy server that offloads file decompression to the client.

www.perforce.com



Tarek Ziadé's *Expert Python Programming* (Packt)

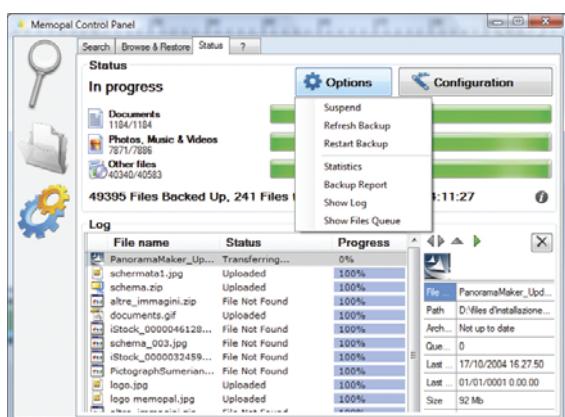
Just in the nick of time for the justifiably hyped Python 3000 is Tarek Ziadé's *Expert Python Programming*, "a practical tour of Python application development", says publisher Packt. The book starts with setting up the best Python development environment and moves on to agile methodologies and applying proven object-oriented principles to one's design. Other skills readers will learn include writing efficient syntax, writing an application based on several eggs, distributing and deploying applications with zc.buildout and applying design patterns. *Expert Python Programming* is for developers who already are building Python applications but want to build better ones by applying best practices and new development techniques.

www.packtpub.com

SEH's PS56 WLAN Print Server

SEH says that its new PS56 WLAN Print Server interface card will make your network printing more secure. Utilizing the highly secure WPA and WPA2 encryption standards, the IPv6-enabled PS56 will connect all HP output devices with an EIO port to a wireless 802.11g network. Because the WPA and WPA2 standards have not yet been cracked, offers SEH, they are regarded as the safest protection for WLANs. To enhance security further, the PS56 also includes TLS/SSL encryption and several IEEE 802.1X authentication methods. The interface card simply slides into the respective slot on the printer unit and is easily configurable and manageable.

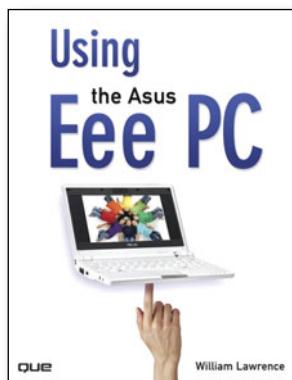
www.seh.de



Memopal's On-Line Backup Utility

European companies often get the jump on their North American counterparts regarding the addition of Linux compatibility. A fine example is Italy's Memopal, which now offers a Linux version of its on-line backup utility. Memopal offers automatic and continuous backup to a remote server via a secure Internet connection, a service that has been lacking in the Linux space. The company claims that its Memopal Global File System archiving technology provides a distributed filesystem that supports up to 100 million terabytes of storage, transparent read-write compression, hot-add scalability and more. In beta at the time of this writing, Memopal for Linux supports Ubuntu 8.04 and Debian Etch.

www.memopal.com



William Lawrence's *Using the Asus Eee PC* (Que)

While other laptops hog the mainstream media glory, the Linux-based ASUSTeK's Eee PC is the underdog "little PC that could". To get to know this now darling of the Linux community, get your hands on William Lawrence's *Using the Eee PC* from Que. The book covers everything from turning on the machine and connecting it to the Internet, as well as how to upgrade and update it. The machine-book combination will help you convert your loved ones to Linux while keeping their after-hours tech-support calls to you at a minimum.

www.informit.com

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

Fresh from the Labs

VDrift—Open-Source Drift Racing Simulator

(vdrift.net)

To start off this month, I deal with my petrol-head side straightaway and look at the racing simulator *VDrift*. For those who read my column last month, you may recall I made a brief mention of this project. This month, I take a more in-depth look at it. To quote the Web site:

VDrift is a cross-platform, open-source driving simulation made with drift racing in mind. It's powered by the excellent Vamos physics engine. It is released under the GNU General Public License (GPL) v2. It is currently available for Linux, FreeBSD, Mac OS X and Windows (Cygwin).

This game is in the early stages of development but is already very playable. Currently the game features:

- 19 tracks: Barcelona, Brands Hatch, Detroit, Dijon, Hockenheim, Jarama, Kyalami, Laguna Seca, Le Mans, Monaco, Monza, Mosport, Nurburgring Nordschleife, Pau, Road Atlanta, Rudskogen, Spa Francorchamps, Weekend Drive and Zandvoort.
- 28 cars: 3S, AX2, C7, CO, CS, CT, F1, FE, FF, G4, GT, M3, M7, MC, MI, NS, RG, RS2, SB, T73, TC, TL, TL2, XG, XM, XS and Z06.
- Compete against AI players.
- Simple networked multiplayer mode.
- Very realistic physics.
- Mouse-/joystick-/keyboard-driven menus.



Determination is needed when approaching the notoriously dangerous and difficult corkscrew section at Laguna Seca.



One of the prettier areas available in *VDrift* is street racing through Detroit.



Scenery like this makes you grin from ear to ear.

Installation I was pleasantly surprised by this project's main installation method, as it eschews all of the usual repository and source stuff and uses Autopackage instead. I've always been a big fan of Autopackage, because it combines the perks of a Windows-style package installation (double-click, Next, Next, Next, Finish—you get the idea) with the added structural benefits of a UNIX-style architecture. There is a source file buried deep under several layers of the Web page, but the choice given for Linux on the main page is an Autopackage, so we'll stick with that here and hopefully annoy some pedantic

Debian developers in the process—the natural enemy of the Autopackage!

Grab the package and save it somewhere locally. Once it's downloaded, you need to flag it as executable (don't worry, this is just a once-off), either by turning the executable option on for the file in the file manager of your choice, or by entering the following at the command line:

```
$ chmod u+x VDrift-2007-03-23-full-2.package
```

Now, you can run the package simply by clicking on it, and follow the Next, Next, Next prompts. You can choose to install it locally or system-wide, depending on whether you have a root password. Note that you can run this from the command line, but it's a bit like mixing 12-year-old Scotch with Coke—it just defeats the purpose. If this is your first time with an Autopackage, before *VDrift* installs, Autopackage installs itself to your system along with a neat Add/Remove Programs-style utility called Manage third-party software in your system menu, where you can remove *VDrift* (or any other Autopackages) later if you want. Don't worry; this also is a one-time-only process. Autopackages will skip straight through to installing after you have Autopackage on your system.

During the installation, Autopackage checks your system for compatibility, and if it encounters any problems, it tells you in the installation window. If you are missing any needed requirements, you can install them in the meantime and run the Autopackage again simply by clicking on it. In terms of libraries, the documentation says you need the following:

- libSDL: simple direct media layer.
- libglew: OpenGL extension utilities.
- sdl-gfx: graphics drawing primitives library for SDL.
- sdl-image: image file-loading library for SDL.

- sdl-net: low-level network library for SDL.
- vorbisfile: file-loading library for the Ogg Vorbis format.
- libvorbis-dev: the Vorbis General Audio Compression codec.

Once the installation process is over, *VDrift* should install itself under your menu, somewhere along the lines of Games→Simulation→*VDrift*.

Usage The first thing you should do is crank up the graphics as much as humanly possible. The default graphics level is very conservative, and even with the graphics turned up, it still has the occasional feel of "ye olde Pentium 133". So, head to the Options→Display section, and then go to the Advanced section below. Texture size, Anisotropic filtering, Antialiasing and Lighting quality will all have a big effect on the look of the game. Back in the main Display section, you can switch between full-screen and windowed mode, as well as change the resolution. If you want to make life easier, you can choose between either miles or kilometers per hour, and enabling the track map really helps when driving somewhere unfamiliar.

***VDrift* Controls**

- W: gear up.
- S: gear down.
- Up arrow: accelerate.
- Down arrow: brake.
- Left/right arrows: steering.
- Spacebar: handbrake.
- F1–F6: camera angles.

Given the general emphasis on physics, you really do get the feeling that this game is meant to be controlled by a steering wheel. If you have one, please, plug it in. In the Controls section, you can tweak any wheel, pedals, joystick or joypad options under Joystick Options as well as adjust the force

feedback settings. If you're a poor bloke like myself, and you can't afford a steering wheel and are stuck with a keyboard, you'll want to turn on the driving aids, such as traction control, ABS and the auto-clutch. I also found that to have any feel without constantly spinning off, I had to change Speed Affect on Steering to 100%.

Enough of this boring setup rubbish though, let's drive! Head into Practice Game, and select a car and a track. Remember, the physics engine is very harsh on driving and will show no quarter to any need-for-speed arcade-racer types. Do not glue the accelerator down! Just keep dabbing at it to begin with—particularly if you're using a keyboard—until you gain more confidence. I found the MC (Mini Cooper) and the XG (which appears to be some kind of BMW, perhaps a 5 series) to be the easiest cars to drive, and the easiest track is Weekend Drive, which had easy corners and will give you an initial feeling for the game.

The default view is behind the wheel, and at a fairly low graphics level, it will not give much of a speed sensation. Things just don't look that fast even in real life unless you have a lot of objects whizzing by your side, and *VDrift* is fairly minimal in terms of roadside distractions. I recommend keeping a close eye on the speed to begin with instead of just feeling it, so you can compare it to real life. In real life, would you take that corner at 73mph? No, you'll understeer into the railing or your back end will swing out. So, practice for half an hour on something long and twisty, such as the Nordschleife Nurburgring circuit, which is simply epic and one of the longest circuits in the world. This track is incredibly hard, and you'll keep coming off, but the corners are endless, and you'll learn to adapt very quickly to the harsher aspects of this game. Mind you, I am a bit of a masochist, so if it puts you off, try another track!

The early development bugs do show through almost immediately in *VDrift*. I found that restarting a track would cut the sound out, and when I tried to change graphical options, things kept resetting and would rarely stay the same between game sessions. When I was driving, I often ran into the kind of jumping physics that have



quantity 1 pricing starts at **\$449**

 Powered by a
200 MHz ARM9 CPU

- Low power, Industrial Quality Design
- Mountable aluminum frame
- 64MB SDRAM (128MB opt)
- 512MB Flash w/ Debian Linux
- Programmable FPGA - 5K LUT
- 7" Color TFT-LCD Touch-Screen
- 800x480 customizable video core
- Dedicated framebuffer - 8MB RAM
- Audio codec with speaker
- Boots Linux 2.6 in about 1 second
- Unbrickable, boots from SD or NAND
- Runs X Windows GUI applications
- Runs Eclipse IDE out-of-the-box

 Our engineers can
customize for your LCD

- Over 20 years in business
- Never discontinued a product
- Engineers on Tech Support
- Open Source Vision
- Custom configurations and designs w/ excellent pricing and turn-around time
- Most products ship next day

 See our website for our
complete product line

 **Technologic**
SYSTEMS

We use our stuff.
visit our TS-7800 powered website at
www.embeddedARM.com
(480) 837-5200

always plagued large 3-D games, especially when you stray from the track. Several times after coming off a tricky hairpin, I had to restart the track. Racing is still pretty rudimentary, and it's not always obvious what you're doing, so things are still best in one-player mode. But, look past this, because you can see that so much passion and research went into this project, with its amazing touches and brilliant locations.

Yes, this game does have a lot of bugs, but it's allowed to—it's in development. You may curse and swear at this incomplete game, which is often ugly and quite harsh to play (and definitely not friendly to beginners). But, ten seconds later, a moment will come when everything looks beautiful, you're in the zone, control is coming naturally and an authentic touch by a big fan will put a huge grin on your face. There really is a large sense of ambition with this game, and if you can look past the early development flaws, you'll find a real gem. The code currently is being rewritten as a side project called Refactor, so I *really* hope this game sees the support and development it deserves, because it could be brilliant—definitely one to watch.

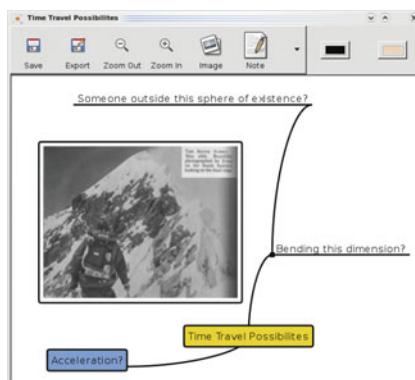
CharTr—Mind Mapping Tool

(code.google.com/p/chartr)

CharTr is an artistic piece of software made for fun to give mind mappers good usability. For those unfamiliar with mind mapping, Wikipedia says the following:

A mind map is a diagram used to represent words, ideas, tasks or other items linked to and arranged radially around a central key word or idea. It is used to generate, visualize, structure and classify ideas, and as an aid in study, organization, problem solving, decision making and writing.

Currently, its stated features are



Here's me going off on a creative bent with CharTr.

as follows:

- Basic mind map with curved links.
- Link folding.
- Colors.
- Outline box of several selected nodes.
- Audio/text/images embedded as notes.
- Automatic saving.
- SVG, PNG, PDF and PS export.
- Numerous keyboard shortcuts (with an eased keyboard navigation, vim-like).
- Idea bookmarking.
- Search for text in nodes.
- Math equations.

Installation CharTr does have a few obscure requirements, so you should look through your repositories. You need Python, PyGTK, Cairo, GStreamer, Numpy and python-plastex for mathematical equations. Once you have these sorted out, head to the Web site where you have a choice of a source tarball or Debian package.

If you grab the .deb package, install

it by entering the following in a terminal from whichever directory contains the file:

```
$ sudo dpkg -i chartr_0.16_i386.deb
```

Now, run CharTr by entering:

```
$ chartr
```

If you get the source version, download and extract the tarball, and then open a terminal in the new CharTr directory.

You need to invoke Python manually, by entering the following:

```
$ python chartr.py
```

Usage Once inside, click that big shiny New button, and a new window appears, called a Map. In the big expanse of white, left-clicking brings up a text cursor allowing you to type in some text. Press Enter, and the text is placed inside a box. The first of these is yellow, allowing for a central idea from which others ideas can flow. If you click on the original box and add some text somewhere else on the map, it is placed in a blue box, and a black line links to it. Right-clicking lets you move the map around, and if you look at the toolbar at the top, you can zoom in and out, as well as add images. If you check the drop-down box toward the right, you also can add bits of audio, notes or some already-provided icons—very handy! Once you've finished making a mind map, you can export it to a picture file. Check the documentation page at code.google.com/p/chartr/wiki/CharTrDocumentationEn for more information on general usage.

All in all, this is a nice and simple application with some great aesthetics that will find favor with students and teachers alike. It's still buggy for the moment, but I hope to see it included in major distros, especially educational ones. ■

John Knight is a 24-year-old, drumming- and climbing-obsessed maniac from the world's most isolated city—Perth, Western Australia. He can usually be found either buried in an Audacity screen or thrashing a kick-drum beyond recognition.

Brewing something fresh, innovative or mind-bending? Send e-mail to John Knight at knight.john.a@gmail.com.

DO IT WITH DRUPAL

A 3 DAY SEMINAR
NEW ORLEANS, LA
DECEMBER 10 - 12, 2008

Still getting your head around Drupal? Learn how to harness the power of this flexible open-source social content management system and web application framework.

- Learn from the world's top Drupal experts
- Examine and dissect successful Drupal sites
- Discover new site-building strategies
- Hear from community-building experts
- Connect with other Drupal professionals

Early registration and hotel room discounts available.

Find speakers, schedule, and specifics at
DoItWithDrupal.com

brought to you by
 **Lullabot™**

SOFTWARE

Tracking Your Business Finances with NolaPro

NolaPro is an easy-to-use, Web-based accounting program. MIKE DIEHL

I've been doing side work for years, and recently, I've become voluntarily self-employed. Like most small and even large companies, one of my biggest problems is tracking who owes me money and whom I need to pay. And, I kind of want to know whether I'm actually making a living. This is where a good accounting program becomes essential.

About six months ago, I started looking for an accounting program in order to track my side work. I needed something that was easy to use, ran on Linux and preferably was Web-based. It had to be easy to use, because I'm not an accountant or a financial analyst; I'm a nerd. The program had to run on Linux, or I wouldn't have anywhere to run it. I wanted it to be Web-based, because I didn't want to have to install the software on all my machines, and I wanted to be able to log time and charges from remote locations. The Holy Grail would have been if it also would integrate with eGroupWare.

I looked at a lot of accounting packages during my search. The ones that worked at all didn't seem very mature. I had resolved simply to deal with buggy software, when I was offered the opportunity to review NolaPro from Noguska LLC (www.nolapro.com/index-5.html).

NolaPro is a Web-based accounting program that runs on Linux using Apache, MySQL and PHP. NolaPro can track customers, orders, accounts payable, inventory, employees and, everyone's favorite, accounts receivable. NolaPro also has a point-of-sale system, a B2B module, an e-commerce module and a general ledger—all of this in one very mature, polished software package.

NolaPro originally was an in-house program used to track Noguska's print-

ing operations. Eventually, Noguska decided to release the program to the public as a free download. As the company doesn't charge money for the program, it doesn't have an advertising budget for it, which explains why I'd never heard of it when I was looking for an accounting program. Even so, the NolaPro user base is growing, and the company actively supports the program. One of the company's motivations for releasing the product is to help introduce its customers to Linux as a flexible and low-cost system, thus reducing its costs as well as its customers' costs.

Noguska offers fee-based custom development and integration services to NolaPro users. The results of this paid development find their way into the next release of NolaPro. So, the entire user community benefits when a company pays to have a particular feature added to NolaPro. Donovan Lee, Noguska's CEO, described it to me as a Robin Hood business model where the richer, larger companies pay for features they need, and the smaller, poorer companies benefit from the results. The results in this case are that NolaPro has the features companies want, not the features the software companies think they want.

Installing the program was pretty easy. I simply downloaded a tarball and extracted it into my Web server's htdocs directory. Then, I pointed my Web browser at the resulting subdirectory, and the installation program presented me with a checklist of things I had to do to make the program work (Figure 1). In my case, I had to make some configuration changes to PHP. The program led me by the hand, and when all of the prerequisites were met, I was presented with an initial login screen—easy.

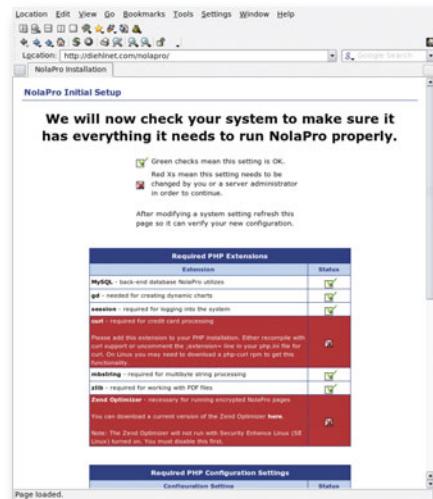


Figure 1. NolaPro Initial Setup

Once I had the program running, I was anxious to get some of my customers into the system (Figure 2). The customer module lets me store just about anything I could care to know about my customers. The interface isn't fancy, but it is intuitive.

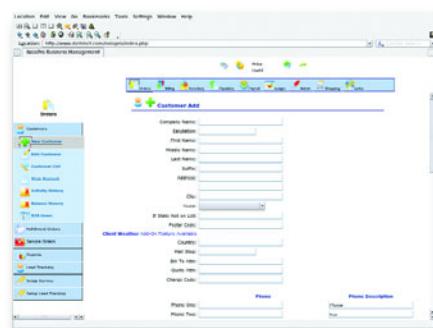


Figure 2. Adding a Customer

But, billing is where the money's at, right? Clicking on Billing and then New Invoice was all it took to start a new customer invoice (Figure 3). The

Customer Name field let me type the name of the company I wanted to invoice, and as I typed, it would search my customer list and fill in the best match. This feature is one of those unexpected gems that make a program easy to use. Once I find the right company, I press Enter and then can enter the terms of the invoice (Figure 4). Clicking Next lets me enter the specifics of the invoice on a line-item basis (Figure 5). The resulting invoice is a professional PDF file that I can send to my clients and that I can track to ensure timely payment.

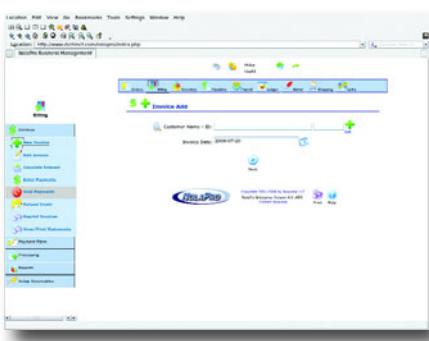


Figure 3. Adding a New Invoice

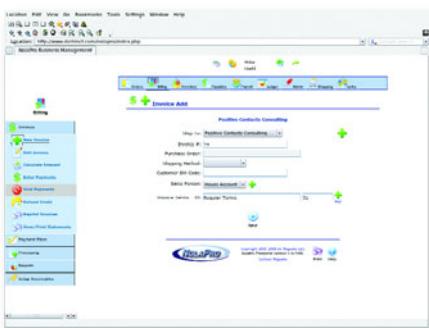


Figure 4. Adding the Invoice Terms

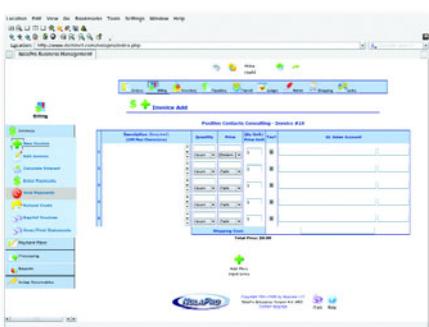


Figure 5. Entering Invoice Specifics on Line-Item Basis

The Payables module is almost as intuitive as the Billing module, although it's not as fun to input where all your money is going as it is to input where it's coming from. This is one area where I had to e-mail NolaPro support. I couldn't seem to add a vendor to the database. It turned out that I had to enter the company name as well as the first and last name of a contact at the company. The support staff returned my e-mail within a few minutes, and I was on my way. I don't write a lot of checks, and I'm too small to need purchase orders, but it's nice to see that NolaPro will track them just the same.

The Orders module was something I discovered almost by accident. Being a computer nerd, I tend to think in terms of "projects", not "orders". But, once I realized that a programming project was nothing but a service order, I found that NolaPro's Orders module would allow me to track all my billable hours against a given order, and when the work was done, I could convert the order into a

ready-made invoice. I had been tracking my billable time in an OpenOffice.org spreadsheet, which was less than ideal.

NolaPro also allows me to enter an estimated cost for each order, which lets me make sure all my projects come in under budget. For orders that entail shipping a product from inventory, NolaPro allows you to create a fulfillment order, so you can estimate shipping costs and modify inventory counts.

Something else I used to track in a spreadsheet was inventory. My VoIP business maintains an inventory of telephone adapters, and it's nice to be able to tell how many I have on hand and where I've stashed them. NolaPro's Inventory module lets me track inventory items that may be stored at multiple locations. It also allows me to set the minimum and maximum number of each inventory item to keep in stock, as well as minimum ordering quantity. I get a substantial discount when I order phone adapters in quantity, for example, and NolaPro allows me to take that

Linux Certified

Linux Laptops: The New LC2000 Series

- High Performance
- Amazing ROI
- Robust
- Fully Compatible
- Cost Effective

Open Source Training, Services and Products 1-877-800-6873 www.linuxcertified.com

fact into consideration when I re-order.

Once a company grows beyond a certain size, it inevitably needs to hire employees. NolaPro has a very mature employee-tracking system. Sure, it tracks your employees' names and phone numbers. It tracks how many hours they work, and it even has an optional time-card function, so they can log their own hours. Then, things get complicated. NolaPro even tracks benefits and taxes. Perusing the whole Payroll section was a daunting experience that left me with a new appreciation for the human resources representative at my old place of employment. With NolaPro, you can track employee benefits, company contributions, deductions, direct deposits and pension plans. The program would let you configure just about any benefits package you could imagine.

Additionally, I thought it was a nice touch that you even can generate phone lists and birthday lists of all your employees.

Each of NolaPros's modules includes its own reporting function. For example, the Billing section includes an Invoice Aging report that shows which invoices are more than 30 days old. The Orders section includes a report that shows open orders and how much money you have tied up in them.

NolaPro's Ledger features go well beyond anything I need as a small-time entrepreneur. Here's where you find a general ledger, budgeting and bank statement reconciliation. Most of the GL functions are tied to a set of standardized general accounts, such as cash on hand, sales income and so forth. The Budgeting module then allows you to set financial targets for each GL account on a per-month basis. The Ledger functions obviously are geared toward companies that are large enough to have an accountant or bookkeeper.

A couple modules are included in NolaPro that I haven't had occasion to use, but I think they are worth mentioning briefly. The Point of Sale module allows a user to sell items from inventory and accept cash, check or credit card as payment, without having to create an invoice. The B2B module allows business partners to log in and view outstanding orders, invoices and payment history. The B2B module also allows partners to pay bills via credit card. I've been told that Noguska will be releasing

a Web Services API that will allow third-party developers to integrate with NolaPro. I've already got some ideas on what to do with the API when it's available. The API should be available sometime in the fall of 2008.

Of course, many features are available as add-ons for a nominal fee. For example, the Daily Accounting Summary add-on shows a daily snapshot of checking balances, receivables aging totals, payments received, orders completed and payables aging totals together on one page for easy viewing, and it costs only \$2.00. On the other hand, the Employee Time Tracking add-on I mentioned earlier provides a way for employees to record hours worked, and it costs \$299. It should be noted that standalone time-tracking systems can cost much more.

Now that I've used the program daily for a couple weeks, I've discovered a work flow that just seems to fit the types of businesses that I run.

When I take on a new programming or consulting project, I create a service order for the project. Then, as I work on the project, I can log my time and note what work was performed on the project. When the project is finished, I can convert the order, history and all, into an invoice that I then send to my client.

My VoIP business is a bit more complicated, because it has an inventory, hardware shipments and regular, recurring billing. I'm able to keep track of how many units I have in inventory. Then, when I take on a new customer, I create a Fulfillment Order for the initial setup and hardware delivery. This order automatically commits a unit from inventory and warns me if I need to re-order. Then, I create a payment plan for the new customer, which automatically creates an invoice for the customer each quarter.

In the time I've been using NolaPro, I've had a few occasions to contact technical support. I've been told that, officially, the free e-mail and telephone support is done on a best-effort basis. In practice, I found NolaPro's e-mail support to be very responsive. I sent in one question at midnight and had a reply by 10am the next day. The context-sensitive help has complete descriptions of each function and includes example screenshots. NolaPro's on-line forums are organized by feature and seem to have a high signal-to-noise ratio. Finally,

there's the video training library. Yes, free video training. If you want to learn how to add a new service order, you can click on a few hyperlinks from the NolaPro home page. From there, you can watch and listen to a video of someone actually performing that task. The video describes the process, and you clearly can see what menu items to use. It's more than just a slideshow with a brief outline. Some of the videos are from older versions of the program, which means the menus have a slightly dated layout, but even so, the video training is well done and quite usable.

NolaPro is able to handle a fairly broad range of business types, but it doesn't do everything. Some of my consulting customers require regular incremental invoicing on projects, even before the project is finished. I discovered that NolaPro won't let me split an order up and invoice part of it. I also discovered that the invoicing module tracks inventory items by count, whereas I need to track them individually by serial number. I was a bit disappointed by the fact that when I asked the program to e-mail an invoice to a customer, it sent a very generic e-mail with the invoice attached as an oddly named .pdf file, and none of this was configurable. I've resolved this by simply exporting the .pdf files and attaching them to my own e-mail messages. Finally, I don't know any two people who do their budgets the same way. NolaPro's budgeting capability seemed to be too closely tied to the GL accounts, which I found cumbersome to wrap my head around. With these weaknesses in mind, NolaPro is a very powerful program, and I'm sure that people in different lines of business would find other things they felt needed to be improved. No software program can do everything to everyone's satisfaction.

For me, accounting is one of the necessary evils of doing business. My job is to solve technical problems and to deliver a service, and the tools need to stay out of the way of the real work. NolaPro is a powerful and intuitive software tool, and I've found the program to be a pleasure to use. It's certainly on my list of keepers. ■

Mike Diehl is a recently self-employed Computer Nerd and lives in Albuquerque, New Mexico, with his wife and three sons. He can be reached at mdiehl@diehlnet.com.

Linux News and Headlines Delivered To You

Linux Journal topical RSS feeds NOW AVAILABLE



http://www.linuxjournal.com/rss_feeds

HARDWARE

The Popcorn Hour A-100

Watch out! Here comes the Networked Media Tank. DANIEL BARTHOLOMEW

Linux-powered devices are everywhere. They're so ubiquitous that Linux often isn't even mentioned. It's just *there*. One case in point is the Popcorn Hour A-100. Linux isn't mentioned in the technical specifications of the device at all. What is mentioned are the formats and codecs and other capabilities of the device, and what a device it is.

The Popcorn Hour is a reference hardware implementation for a new Linux-based middleware layer from Syabas Technology called the Networked Media Tank. According to its Web site: "The Networked Media Tank (NMT) is a state-of-the-art integrated digital entertainment system that allows you to watch, store and share digital content on your home network."

Think of tank in terms of fish, not Abrams or Sherman. The NMT is designed to be able to access all your media, no matter what computer or networked storage system it resides on, and display it on whatever television you connect to it. According to Syabas, other devices are in the works from other manufacturers that will be using the same NMT middleware and similar hardware.



Figure 1. Popcorn Hour and Everything That Comes in the Box

The Popcorn Hour is not designed to compete with your TiVo or MythTV box; it can connect to on-line video streams and podcasts, but it doesn't do live over-the-air (or cable) television. The closest device it competes with, at least partially, is the Neuros OSD. The Neuros can encode video though, which is something the Popcorn Hour can't do—it's strictly a playback device.

For me, the encoding capabilities of the Neuros OSD are not needed. I've already digitized most of my DVD library. I have a mix of media in both MPEG-4.2 and H.264 formats, and the Popcorn Hour's support of both formats was one of the things that attracted me to it over the MPEG-4.2-only Neuros OSD. For more on the differences between MPEG-4.2

When MPEG-4 Isn't MPEG-4

Many people are familiar with what generically is known as MPEG-4 or MP4 video. Popular implementations of this are DivX and Xvid, both of which have found wide use on file-sharing sites. Technically though, DivX and Xvid implement MPEG-4 Part 2. Much like MPEG-2 before it, the MPEG-4 standard encompasses several different audio, video and file format standards. There isn't space to go into too much detail on MPEG-4 here, so see en.wikipedia.org/wiki/MPEG-4 for more information (a lot more). The main point I want to make about MPEG-4 Part 2 is that even though when the first implementations were released it was hailed as an excellent video codec that was far better than MPEG-2 video, MPEG-4 Part 2's two main modes are known as the Simple or Advanced Simple Profiles. In other words, they're the children of the MPEG-4 world. The "all-grown-up" video codec of MPEG-4 is Part 10, which is also known as MPEG-4 AVC (Advanced Video Coding). The International Telecommunications Union calls it H.264.

To avoid confusion, when I refer to MPEG-4 Part 2 in this article, I call it MPEG-4.2 instead of Xvid or DivX or the generic MP4. And, when I'm talking about MPEG-4 Part 10, I refer to it by the ITU name, H.264.

Much as MPEG-2 is the format used on DVDs, H.264 video is the preferred video format of Blu-ray discs. It also is becoming the preferred video format for small devices, such as cell phones. This is because H.264 was designed to provide video quality equivalent to MPEG-4.2 at half the bandwidth. This efficiency comes at an increased processing cost both to encode and decode, but since the standard was formalized a few years ago, several chipsets have been developed to do the decoding in hardware. Thus, even extremely small and low-power devices, such as cell phones, can play back H.264-encoded video easily.

This is exactly what the Popcorn Hour does. It utilizes a Sigma Designs SMP8635 chip, which, according to the manufacturer, provides MPEG-4.2, H.264, VC-1, WMV9 and MPEG-2 decoding at up to 1080p resolution.

and H.264, see the When MPEG-4 Isn't MPEG-4 sidebar.

Another thing that caught my eye on the Popcorn Hour were the various outputs, which include composite, component, S-Video and HDMI. I currently have a standard-definition

television, but we plan on replacing it with an LCD-HDTV before the end of the year, and the Popcorn Hour will work on both. It can output NTSC, 480p, 720p, 1080i and 1080p among others.



Figure 2. The Ports on the Back of the Popcorn Hour

The Popcorn Hour casing is sleek and rather empty inside. Only about one-third of the case is the actual hardware for the device. The other two-thirds is reserved for installing your own ATA hard drive.



Figure 3. The actual hardware takes up very little of the case.

Under full operation, and especially with a hard drive installed, the Popcorn Hour becomes quite warm, so although the nice flat top of the unit seems to be begging for something to be stacked on it, it's probably a good idea to give it a little breathing room all the way around.

Installing a hard drive into the Popcorn Hour is essential if you want to use it as a file server or BitTorrent client. For those uses, there needs to be some sort of local storage. Installing a hard drive is as easy as sliding it in and screwing it to the bottom of the case. All of the necessary screws and cables are provided. One nice touch is that the top of the case is attached with thumbscrews, so accessing the interior is easy.

Initial Setup

Setting up the Popcorn Hour is easy. For basic operation, you simply plug it in and turn it on. From the factory, the Popcorn Hour is set to detect the appropriate video output settings automatically, so after a few seconds of my television blinking, the startup screen appeared and the main interface loaded.

The Popcorn Hour interface is attractive, but it is not flashy or fancy. It is very serviceable, and it gets the job done.

If you have DHCP set up on your network, the Popcorn Hour will connect automatically. Otherwise, you need to put in the appropriate network settings manually. On my network, the Popcorn Hour was given the IP address 192.168.1.148, which I mapped in my /etc/resolv.conf file to popcorn, so in the examples in this article, that's what I use.

After the home screen appeared, my first step was to visit the Setup section to set my default output settings—so it doesn't have to scan for the correct setting every time it starts—and to install the extra apps to the hard drive along with setting other preferences.

The initial step of installing the apps to the hard drive involves making sure all firmware updates have been applied. The firmware upgrading process was easy and took only a few minutes. Once the firmware is at the latest version, the hard drive apps can be installed.

The apps can be installed over the Internet or from a USB key (if the Popcorn Hour doesn't have an Internet connection). After installing the apps, my next step was connecting it to my NFS server. The process was easy once I actually read the documentation for the proper URL formatting. By default, when specifying an NFS address, such as nfs://fileserver/mnt/files, the Popcorn Hour tries to connect over UDP. For my setup, I had to use nfs-tcp://fileserver/mnt/files, and then it worked.

For my setup, I installed the NFS server and BitTorrent client. There is also a Samba server option, but I don't have any Windows machines, so I didn't enable it. Once installed and started, I was able to mount the Popcorn Hour NFS share from any of my machines with the following:

```
mount -t nfs popcorn:/opt/sybhttpd/localhost.drives/HARD_DISK
→ /mnt/popcorn
```

The brief documentation that comes with the Popcorn Hour is Windows-centric, so I had to use the showmount command to discover what the export was called on the Popcorn Hour, like so:

```
danielb@d610:~$ showmount -e popcorn
Export list for popcorn:
/opt/sybhttpd/localhost.drives/HARD_DISK 192.168.1.0/
→ 255.255.255.0
```

The on-line forum and wiki also came in very handy whenever I had similar questions.

Testing the Popcorn Hour

One of my first orders of business after the initial setup was out of the way was to test the capabilities of the Popcorn Hour to see whether it could play all the formats advertised.

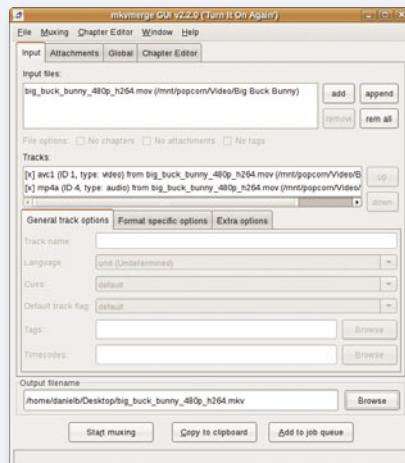
For video tests, I used *Big Buck Bunny* from the Open Movie Project. This Creative Commons-licensed animated short

Matroska

According to the Matroska home page, "Matroska aims to become the standard of multimedia container formats." A lofty goal to be sure, but it's making progress due to the tremendous flexibility the container format has.

The main trick that Matroska has over other container formats is that it can support multiple audio and video streams inside a single file. This enables you to, for example, have multiple selectable languages to go along with the video portion of the file. If you think that sounds like a DVD, you're on the right track. Many people feel that the era of physical formats, like DVDs and Blu-ray discs, is coming to an end. In fact, many of them feel that Blu-ray is the *last* physical format. The only problem with this line of thinking is that if the world moves away from physical formats, whatever replaces them should be able to do everything (or almost everything) they can do. This includes multiple languages, alternate video streams, subtitles in various languages and other features. The LGPL-licensed Matroska is trying to become that format.

Matroska containers can contain nearly



The mkvmerge program is very easy to use.

any audio and video format, and one of the ways for putting those formats into a .mkv file on Linux is with mkvmerge. The mkvmerge tool can be downloaded as part of the mkvtools package from www.bunkus.org/videotools/mkvtoolnix/downloads.html.

Follow the directions for your distribution to install it.

To change the container of any video file from whatever it is to Matroska, simply launch the mkvmerge GUI, and click the Add button to open the

file you want to convert. By default, it will save the output .mkv file to the same directory. If you want to change that, click on the Browse button in the lower-right corner of the window, and choose where you want to save it. Finally, click the Start muxing button, and mkvmerge will begin the process of extracting the audio and video from the existing container and putting it all into a Matroska container. Because the tool is not converting the audio or video to a different format, the process is lossless and does not take long.

If you want to do the muxing from the command line, the GUI tool offers a Copy to Clipboard button that gives you the command and all of the options that it will do when you press the Start muxing button. The general command is this:

```
mkvmerge -o "destination-file.mkv" -a 1 -d 0 -S
→ "original-file.avi" --track-order 0:0,0:1
```

At the end of the process, you will have a Matroska container with whatever audio and video you copied out of the other container inside of it.

For more on Matroska, see matroska.org.

is available at bigbuckbunny.org in several video formats and sizes. The sizes range from 1920x1080 pixels (1080P) at the high end down to 320x180 pixels at the low end. For each size, there is a selection of container formats, including AVI, Ogg, M4V and MOV. Each container format has a different video format inside it, including MPEG-4.2, H.264, MS MP4 and Theora. The audio is in MP3, AAC, AC3 and Vorbis formats. These various versions provide a pretty good test suite for the capabilities of any video player.

The only container format it doesn't have is Matroska (.mkv). This, however, was not an insurmountable problem, because it is simple to create it, thanks to mkvmerge. For more on this and the Matroska container format, see the Matroska sidebar.

The most dominant video format of the past ten years (or more) has been MPEG-2, the format DVDs use. Thankfully, the Open Movie Project has me covered there too. *Big Buck Bunny* can be downloaded as a DVD ISO file from Archive.org (www.archive.org/details/BigBuckBunny), which can be burned to a DVD or simply opened by the Popcorn Hour (more on playing ISO files later).

The Fast Forward button on the remote works as advertised for all supported video files. Rewind works with varying levels of success on .avi, .mov and .m4v files. Rewind did not work at all with the .mkv files I have. Hopefully, a firmware upgrade will fix this.

All the video formats I tried worked with the exception of Microsoft's implementation of MPEG-4.2 (msmp4) and the Theora-encoded version. I also noticed some slight jitteriness with the 1080p versions when I played them off my NFS server. They played without a hitch when the files were on the local hard drive. The scaling that the Popcorn Hour applied to the large files to shrink them down to my television screen looked flawless. There was pixelation (naturally) when I played the smaller versions that had to be scaled up, but they still were very watchable. Audio playback likewise was flawless.

For audio tests, I tried a variety of files, including MP3, FLAC, Ogg Vorbis and M4a. The Popcorn Hour had no problems with the MP3 and FLAC files, but it would not play the Ogg or M4a files. The failure of the M4a files was surprising, as both the M4a container and the AAC audio format are listed as supported. More than 95% of my music is in either MP3

or FLAC format (with FLAC being the preferred format), so the lack of M4a and Ogg support is not that big of a deal, but I hope they will be enabled at some point.

One thing I should note is that the Popcorn Hour does not play any sort of DRM-infected content. So, if you've been purchasing things from iTunes and/or similar digital stores that cripple their content, the Popcorn Hour probably is not a good purchase.

BitTorrent

One of the big selling points of the Popcorn Hour is the built-in BitTorrent client. My experience with it has been mixed.

On one hand, the BitTorrent client works. On the other hand, it's very painful to use with the remote. I do like the fact that I can check on the status of torrents I am downloading or seeding right on my television, but anything more than that (like setting up the schedule) is difficult at best. Thankfully, there is a Web interface that is much easier to use.

The Web interface has all the features of the TV interface with the addition of an upload interface to add new torrents. To add torrents to the list, you first need to download the .torrent file to your desktop and then connect to the Web-based torrent front end and upload the file through that. The address for the Web-based front end is at popcorn:8883/torrent/bt.cgi.

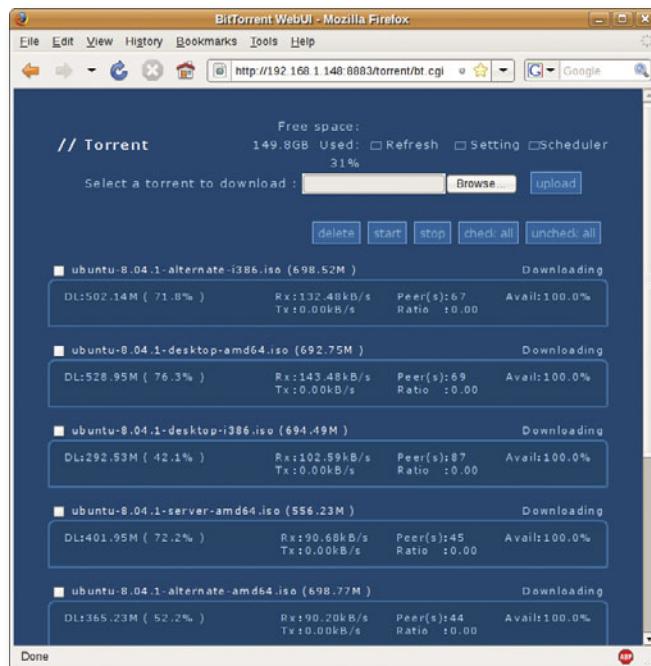


Figure 4. The BitTorrent service can be controlled via a Web front end.

Content Providers

The Online Services area has a lot of content preconfigured for it. The biggest section is the Media Service Portal, which is filled with dynamically updated content from various providers, such as YouTube, Revision3, DLTv, SayaTV, Vuze, Mevio and even a selection of podcasts from the NBC, CBS and CNN news networks.

There also are slots for adding podcasts (video or audio) of your own. Simply choose the Edit link from the Online Services

page, and then enter the title of the podcast and the RSS feed URL. After saving your changes, when you select the newly created entry, the Popcorn Hour will fetch and parse the RSS and give you a link to the audio or video file. Select it, press the Enter button, and the podcast will play after a delay. The length of the delay is dependent on your Internet bandwidth and the size of the file you are downloading. In my testing with some large-format (480p and up) video podcasts, the Popcorn Hour had trouble downloading the files and even crashed a couple times. So, although it's nice that the Popcorn Hour can connect to RSS podcast feeds that I specify directly, a much more reliable solution for me is to download them on my local file server and then access them through the NFS share.

DVD Playback

The Popcorn Hour can play DVDs either from a USB DVD-ROM drive plugged in to one of the USB ports or from a DVD ISO file. There is one huge caveat to this ability, however; it can't play encrypted DVDs, which basically covers nearly all commercial DVDs and ISO files of those DVDs. The only DVDs I found in my collection that can be played directly by the Popcorn Hour are the ones I purchased at the dollar store and *Big Buck Bunny* from the Open Movie Project.

Assuming you have some unencrypted DVDs or ISO files, playing them is similar to using any off-the-shelf DVD player. One note though: playing an ISO of a DVD off of a network

Low Cost Panel PC

PPC-E7

- Cirrus ARM9 200MHz CPU
- 3 Serial Ports & SPI
- Open Frame Design
- 3 USB 2.0 Host Ports
- 10/100 BaseT Ethernet
- SSC-I2S Audio Interface
- SD/MMC Flash Card Interface
- Battery Backed Real Time Clock
- Up to 64 MB Flash & 128 MB RAM
- Linux with Eclipse IDE or WinCE 6.0
- JTAG for Debugging with Real-Time Trace
- WVGA (800 x 480) Resolution with 2D Accelerated Video
- Four 12-Bit A/Ds, Two 16-Bit & One 32-Bit Timer/Counters



2.6 Kernel

Setting up a Panel PC can be a puzzling experience. However, the PPC-E7 Compact Panel PC comes ready to run with the Operating System installed on Flash Disk. Apply power and watch either the Linux X Windows or the Windows CE User Interface appear on the vivid color LCD. Interact with the PPC-E7 using the responsive integrated touch-screen. Everything works out of the box, allowing you to concentrate on your application, rather than building and configuring device drivers. Just Write-It and Run-It. Starting at \$495.

For more info visit: www.emacinc.com/panel_pc/ppc_e7.htm

Since 1985
OVER
23
YEARS OF
SINGLE BOARD
SOLUTIONS

EMAC, inc.
EQUIPMENT MONITOR AND CONTROL

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

share takes a lot of bandwidth, so you'll have your best luck with NFS and a wired connection as opposed to Samba and/or a wireless connection.

myiHome

For people who don't want to go to the trouble of setting up an NFS or Samba server, there is another option for sharing media from your computer to the Popcorn Hour: myiHome. The myiHome application can be downloaded from www.networkedmediatank.com/download/myihome.html.

There are versions for Windows, Mac OS and Linux.

For Linux, you simply download a tar.gz file. Untar it, and you will have a folder named something like

myiHomeLinux-v5.0.2. Inside this folder is a startserver.sh script. Running this script starts myiHome. To stop it, simply press Ctrl-C, and it will quit.

Once the server is started, it shows up in the list of media sources on the Popcorn Hour automatically. On Linux, the server automatically looks in your home directory for folders named My Videos, My Music and My Pictures.

After connecting to a myiHome server, you can set various preferences from the television interface, such as selecting the photo folder to use as a slideshow source when playing back music and vice versa, among other things.

When navigating music or video folders on a myiHome server, you have the option to play the contents randomly with the Shuffle command. There also is a Search button that you can use to find specific tracks or photos.

The Remote

Navigating with the remote is easy; you simply use the arrow keys on the remote and the Enter key to select things—much like navigating DVD menus. To go up a directory, use the Return key. The combination number/letter keys are used during initial configuration



Figure 5. The Popcorn Hour Remote

when setting network settings and configuring shares, and I haven't had much use for them since.

The DVD-specific keys (Menu, Title, Angle and so on) work as expected with DVDs, but are useless otherwise.

My biggest complaint about the remote is the location of the Page Up and Page Down keys. They're way up in the top-right corner. A much better location would have been next to the arrow keys.

There is a Suspend BT button on the remote that the manual says is for suspending all BitTorrent traffic with one click. It does not work at the time of this writing, but the promise is that a firmware update will enable it (hopefully, by the time you read this). I hope it's sooner rather than later, because although it isn't hard to suspend BitTorrent traffic, it does take several clicks with the remote to do it, and you have to stop whatever you're watching or listening to.

There also are some colored buttons at the bottom of the remote that don't do anything at the moment.

One trick I learned was that if you want to play everything in a folder, first navigate into the folder and then press the Play button. If you want to play only a single file, highlight the file and press the Enter button. Also, when playing a file, you need to press the Stop button before you can use the Return or Home buttons.

Conclusion

The Popcorn Hour is a very capable little box. It plays a wide variety of music and video formats—provided they aren't encumbered with DRM.

Right now, there are several unfinished pieces, but thankfully, firmware updates are coming regularly, and each one unlocks more functionality. Despite the rough spots, I have to admit I am perfectly happy with the core functionality, and I recommend it (as long as you don't have a lot of DRM-infected content). It is well worth the modest purchase price of \$179.■

Daniel Bartholomew lives with his wife and children in North Carolina. He can be found on-line at daniel-bartholomew.com.

Resources

Popcorn Hour Web Site: popcornhour.com

Networked Media Tank and Popcorn Hour Forum:
networkedmediatank.com

The NMT Wiki: networkedmediatank.com/wiki

The NMT Quick-Start Guide:
support.popcornhour.com/UserFiles/Popcorn_Hour/file/NMT_Quick_Start_Guide_Rev1_0.pdf

Sybas Technology: sybas.com

Details of Sigma Designs SMP8635 Chip:
www.sigmadesigns.com/public/Products/SMP8630/SMP8630_series.html

THE SOFTWARE BUSINESS 2008 CONFERENCE

Conference and Workshops for Executives and Managers of Software and SaaS Companies

Oct. 30-31 Marriott San Francisco

www.SoftwareBusinessOnline.com

Software Business 2008, the software industry's leading annual learning and networking event, will focus on current strategic business, financial and technology issues and growth opportunities facing executives and managers of software companies. This two-day conference serves owners, chief executives, presidents, vice presidents, division directors and department managers of leading and fast-growing software companies, located throughout North America, who are conducting business domestically and worldwide. Make your plans today to join more than 200 large, mid-sized and small software companies at the premier event for the software industry.

- Learn first-hand the latest strategies for success in the software business from top executives, pros and analysts
- Gain authoritative insight from top financial analysts and investment bankers on software company M&A activity and company financings
- Find out how to turn the latest technology developments into new business and improved performance for your company
- Network with key executives and managers from leading software companies engaged in a variety of markets
- Learn exciting new marketing and sales strategies for software companies
- Find out the best revenue models for licensing, services, e-business and online sales of your software offerings
- Hear about hot new markets and growth opportunities for software companies
- Learn new ways to build your franchise, and protect it

Exceptional Presentations By:

Jean-Pierre Garbani • Forrester Research
Ken Bender • Software Equity Group
David Lawee • Google
Alex Lintner • Intuit
Bernie Anzarouth • Constellation Software
John Ciacchella • Deloitte Consulting LLP
Robert Shimp • Oracle
Jeffrey M. Kaplan • THINKstrategies, Inc.
George Hoyem • Blueprint Ventures
Javier Rojas • Kennet Partners
Colleen Smith • Progress Software Corp.
Billy Marshall • rPath
Guy Smith • Silicon Strategies Marketing
Jim Watson • CMEA Ventures
Tae Hea Nahm • Storm Ventures
Alex Dodd • M Squared Consulting
Theresa Bui Friday • Palamida
Rick Sklarin • Crimson Consulting
Michael Gorriarán • Microsoft Corp.
Todd Rowe • Business Objects
John Gunn • Aladdin Knowledge Systems
Howard Lubert • SafeHatch, LLC
Henry Bruce • The Rock Annand Group
Brian Turchin • Cape Horn Strategies, Inc.
Ken Holec • Jobs2Web
Richard Muirhead • Tideway
Chetan Saiya • Assetlink
Christopher Cabrera • Xactly Corp.
Bill Auvil • FrontRange Solutions
Ari Takanen • Codenomicon Ltd.
Ward Carter • Corum Group Ltd.
Treb Ryan • OpSource
Bob Norton • C-Level Enterprises

WWW.SOFTWAREBUSINESSONLINE.COM

7355 E. Orchard Road, Suite 100, Greenwood Village, CO 80111 USA

Phone +720 528 3770

Fax +720 528 3771

THE ROADRUNNER SUPERCOMPUTER: A PETAFLOP'S NO PROBLEM

IBM and Los Alamos National Lab built Roadrunner, the world's fastest supercomputer. It not only reached a petaflop, it beat that by more than 10%. This is the story behind Roadrunner. **JAMES GRAY**



Figure 1. Part of the Roadrunner team at Los Alamos National Lab

In 1995, the French threw the world into an uproar. Their testing of a nuclear device on Mururoa Atoll in the South Pacific unleashed protests, diplomatic friction and a boycott of French restaurants worldwide. Thanks to many developments—among them Linux, hardware and software advances and many smart people—physical testing has become obsolete, and French food is back on the menu. These developments are manifested in Roadrunner, currently the world's fastest supercomputer. Created by IBM and the Los Alamos National Laboratory (LANL), Roadrunner models precise nuclear explosions and other aspects of our country's aging nuclear arsenal.

Although modeling nuclear explosions is necessary and interesting to some, the truly juicy characteristic of the aptly named Roadrunner is its speed. In May 2008, Roadrunner accomplished the almost unbelievable—it operated at a petaflop. I'll save you the Wikipedia look-up on this one: a petaflop is one quadrillion (that's one thousand trillion) floating-point operations per second. That's more than double the speed of the long-reigning performance champion, IBM's 478.2-teraflop Blue Gene/L system at Lawrence Livermore National Lab.

Besides the petaflop achievement, the story behind Roadrunner is equally incredible in many ways. Elements such as Roadrunner's hybrid Cell-Opteron architecture, its applications, its Linux and open-source foundation, its efficiency, as

well as the logistics of unifying these parts into one speedy unit, make for a great story. This being *Linux Journal*'s High-Performance Computing issue, it seems only fitting to tell the story behind the Roadrunner supercomputer here.

Many Pounds of Stuff

"You want to talk about challenges; it is the logistics of dealing with this many pounds of stuff", said Don Grice, IBM's lead engineer for Roadrunner. "The folks in logistics have it down to a science." By the time you read this, the last of Roadrunner's 17 sections with 180 compute nodes—250 tons of "stuff" on 21 semitrucks—will have left IBM's Poughkeepsie, New York, facility, bound for Los Alamos National Laboratory's Nicholas Metropolis Center in New Mexico.

The petaflop accomplishment occurred at "IBM's place", where the machine was constructed, tested and benchmarked. In reality, Roadrunner achieved 1.026 petaflops—merely 26 trillion additional floating-point calculations per second beyond the petaflop mark. Roadrunner's computing power is equivalent to 100,000 of today's fastest laptops.

The Roadrunner is one of the most complex projects undertaken by both IBM and its partners. IBM produced each of Roadrunner's two server blades in two different locations and assembled them into so-called tri-blades in a third. The tri-blades then were shipped to Poughkeepsie to become part of

FEATURE The Roadrunner Supercomputer



Figure 2. Early Bird's-Eye View of Roadrunner (Phase 1 Test Version) in Spring 2007 at Los Alamos National Lab

Roadrunner. Despite this logistical hurdle, the project was completed on schedule and at budget.

IBM also had to find partners for the entire interconnect fabric, make it scale and obtain the desired performance. The company also worked with various Linux and other open-source communities to build a coherent software stack. Fears that the high level of coordination among partners, such as Emcore, Flextronics, Mellanox and Voltaire, wouldn't work out were proved unfounded. "They all pulled together in a tremendous, tremendous way", said Grice. "There isn't any aspect of the machine that isn't doing what it was supposed to."

Before Roadrunner Ran

Of course, a project of Roadrunner's magnitude requires many smart people at both IBM and LANL, who have collaborated for six years to develop and build it. The team at LANL consisted of 171 people, with a group of similar size on the IBM side. "Los Alamos and IBM have formed a very close partnership", commented Andy White, LANL's Project leader. "We have been able to work together to work through many problems", he added.

According to White, the project planning began in 2002, when LANL decided to pursue supercomputers with accelerators (in the end, the Cell processors) to achieve its modeling needs. They had begun hearing about the Cell processor and were intrigued about the potential for its applications. LANL determined that it essentially wanted a very large Linux cluster and realized that with the accelerators, they could reach the petaflop.

IBM and LANL jointly worked on Roadrunner's overall design; IBM implemented the code—that is, the computational library (ALF) and the arithmetic software (DaCS) for hybrid systems. The Los Alamos group was tasked with ensuring that its applications would run on the machine. The system modeling group spent an entire year analyzing its applications and the performance characteristics of the machine, making sure that both LANL's classified work and all kinds of interesting science applications would work well. The group built four applications related to its nuclear-physics modeling. These include the Implicit Monte Carlo (IMC) code (the Milagro application suite) for simulation of thermal radiation propagation, the Sweep3D kernel, the SpaSM molecular dynamics code and the VPIC particle and cell plasma physics code. LANL's White says that these applications were the basis for asking the question "Can we program [Roadrunner] and can we get accelerated performance on this system?"

There were no shortage of significant intellectual challenges to making Roadrunner do its job. One was to prove that the aforementioned applications could run on an accelerated Roadrunner—without having it to run on! In September 2006, IBM delivered a base system to LANL for testing, but without accelerators. The applications could be tested on the Cell processor but not on the complete node or system. White explained:

The performance and architecture laboratory team actually was able to model the entire system [complete with acceleration] and predict pretty much dead-on what has

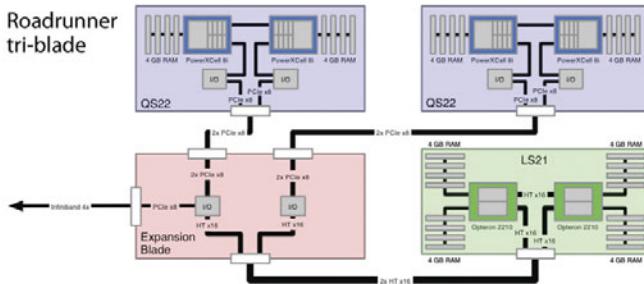


Figure 3. The hybrid Opteron-Cell architecture is manifested in a tri-blade setup. The tri-blade allows the Opteron to perform standard processing while the Cell performs mathematically and CPU-intensive tasks.

happened when the code was run on the full system. The fact that we were able to pass two serious technical assessments in October [2007] and show people that we can program the machine, the codes can get good speed up, they're accelerated, and we can manage the machine, etcetera, without actually having the machine on hand, I think was a tour de force.

Another challenge involved the networking. While working with the base system in late 2006 and early 2007, the concern arose that Roadrunner's computing horsepower would cause the network to be a bottleneck. Thus, said White, "the nodes were redesigned in-flight", with the new ones offering a 400% increase in performance from the Opteron to the Cell processors, as well as out to the network, vis-à-vis the original design. All of this was done at the same original contract price.

The \$110 million Roadrunner was completed on schedule, just in time to qualify for the June 2008 edition of the Top 500 list of the world's most powerful computer systems.

Speed = Hybrid Architecture + Software

You may be surprised to learn that Roadrunner was built 100% from commercial parts. The secret formula to its screaming performance involves two key ingredients, namely a new hybrid Cell-Opteron processor architecture and innovative software design. Grice emphasized that Roadrunner "was a large-scale thing, but fundamentally it was about the software".

Despite that claim, the hardware characteristics remain mind-boggling. Roadrunner is essentially a cluster of clusters of Linux Opteron nodes connected with MPI and a parallel filesystem. It sports 6,562 AMD dual-core Opteron 2210 1.8GHz processors and 12,240 IBM PowerXCell 8i 3.2GHz processors. The Opteron's job is to manage standard processing, such as filesystem I/O; the Cell processors handle mathematically and CPU-intensive tasks. For instance, the Cell's eight vector-engine cores can accomplish acceleration of algorithms, much cooler, faster and cheaper than general-purpose ones. "Most people think [that the Cell processor] is a little bit hard to use and that it's just a game thing", joked Grice. But, the Cell clearly isn't only for gaming anymore. The Cell processors make each computing node 30 times faster than using Opterons alone.

LANL's White further emphasized the uniqueness of Roadrunner's hybrid architecture, calling it a "hybrid hybrid", because the Cell processor itself is a hybrid. This is because the Cell has the PPU (PowerPC) core and eight SPUs. Because the

PPU is "of modest performance" as the folks at LANL politely say, they needed a core for running code that wouldn't run on the SPUs and improved performance. Thus, the Cells are connected to the Opteron.

The system also carries 98 terabytes of memory, as well as 10,000 InfiniBand and Gigabit Ethernet connections that require 55 miles of fiber-optic cabling. 10GbE is used to connect to the 2 petabytes of external storage. The 278 IBM BladeCenter racks take up 5,200 square feet of space.

The machine is composed of a unique tri-blade configuration consisting of one two-socket dual-core Opteron LS21 blade and two dual-socket IBM QS22 Cell blade servers. Although the Opteron cores each are connected to a Cell chip via a dedicated PCIe link, the node-to-node communication is via InfiniBand. Each of the 3,456 tri-blades can perform at 400 Gigaflops (400 billion operations per second).

See Figure 3 for a schematic diagram of the tri-blade.

The hybrid, tri-blade architecture has allowed for a quantum leap in the performance while utilizing the same amount of space as previous generations of supercomputers. Roadrunner takes up the same space and costs the same to operate as its two predecessors, the ASC Purple and ASC White machines before it. This is because performance continues to grow predictably at a rate of 1,000% every 10–11 years. Grice noted how just three of Roadrunner's tri-blades have the same power as the fastest computer from 1998. Put another way, a calculation that would take a week on Roadrunner today would be only half finished on an old 1 teraflop machine that was started in 1998.

Such quantum leaps in performance help boggle the minds of many scientists, who see their careers changing right before their eyes. If they have calculations that take too long today, they can be quite sure that in two years, the calculation will take one-tenth of the time.

Neither IBM's Grice nor LANL's White could emphasize enough the importance and complexity of the software that allows for exploitation of Roadrunner's hardware prowess. Because clock frequency and chip power have plateaued, Moore's Law will continue to hold through other means, such as with Roadrunner's hybrid architecture.

Roadrunner Runs

Roadrunner was put together in its full configuration on May 23, 2008. On May 26, it reached the petaflop. "Running a petaflop just three days after being assembled is pretty amazing", said White.

Clearly a petaflop isn't the limit. Not only was the original petaflop achievement actually 1.026 petaflops, since then, Roadrunner has done better. In June 2008, LANL and IBM ran a project called PetaVision Synthetic Cognition, a model of the brain's visual cortex that mimicked more than one billion brain cells and trillions of synapses. It reached the 1.144 petaflop mark. Calculations like these are the petaflop-level tasks for which Roadrunner is ideal.

"It's hard to overstate how exciting it is to see the science we'll be able to do with Roadrunner", said White. In mid-2009 the bulk of Roadrunner's nodes will enter "classified" mode for the rest of its life, allowing only authorized personnel to know what it's doing. Nevertheless, scientists and their groupies will be happy to learn about some of Roadrunner's

FEATURE The Roadrunner Supercomputer

non-military duties. First, in August 2008, LANL ordered two additional connected units for Roadrunner, dubbed the Turquois Network, which will be available and "in the open all the time", according to White. These units should be running by October 2008. In addition, during early 2009 before Roadrunner goes classified, LANL will utilize several other so-called unclassified open science codes as test loads as part of Roadrunner's stabilization and integration process. The ten codes that have been selected for this purpose must prove their ability to work on Roadrunner. Although some of these codes are based on the above-mentioned VPIC and SPaSM, others are new and untested. "It remains to be seen whether others can write codes that actually will run on the system", stated White.

LANL received 29 proposals for access to Roadrunner, of which two were weapons-related and eight were non-weapons-related. A sampling of the fascinating selected projects include investigations of the formation of metallic nanowires with an atomic-force microscope, the phylogenetics of the early infection states of HIV and, finally, dark energy and matter.

Although the chance to utilize Roadrunner's power is enticing, one must consider the extra tweaking to take advantage of the hybrid architecture. "It can be tricky", said White. With a more conventional machine, codes don't require much change from one Linux cluster to another. Fortunately, for those scientists whose proposals have been accepted, LANL is offering extra funds to support code development to the hybrid architecture. This December, LANL will evaluate the progress of each project and allocate compute time in early 2009 based on those results.

Linux Was a Natural Fit

No, I didn't forget to mention that Roadrunner runs on Linux—Red Hat to be exact. From the beginning, the LANL team knew it wanted Linux due to the open nature of its mission and what it sought to accomplish. IBM's Grice added that LANL "has always been interested in Linux things, so it was a natural fit. We did think about [other operating systems] but we didn't think very hard."

Technically, Linux was a good fit too. The teams didn't need to concern themselves about running either the Cell processor or the LS21 blade server, nor is scalability a major issue, as it didn't come down at the node level. Rather, it is about using all of the nodes together, which means a low level of strain on the operating system. IBM's Linux Technology Center was instrumental in making Linux work on the Roadrunner.

Beyond Linux, Grice praised other open-source communities for their "tremendous cooperation". He explained how they excitedly dived into the unique challenges presented by Roadrunner and its hybrid architecture and surpassed all expectations. Some of the open-source applications include the Moab scheduler and Torque resource manager.

To the surprise of IBM and LANL, most potential software "issues" never turned into problems. However, one challenge presented by open source is the numerous streams that aren't always compatible with each other. Thus, the teams had to hold themselves back in some places and experiment in others to keep a stack that was coherent with itself. Nevertheless,

result was satisfying and scaled effectively.

"The notion that there were separate communities who all pulled together, and then it all locked in together as one whole stack, that I think is a fantastic story", said Grice.

Keeping the Bird Cool

In general, "power and cooling are second only to the software complexity", emphasized Grice. Power is the real problem for driving HPC forward. Roadrunner solves these issues through the efficiency of its design. Especially due to the efficiency of the Cell processors, Roadrunner needs only 2.3MW of power at full load running Linpack, delivering a world-leading 437 million calculations per Watt. This result was much better than IBM's official rating of 3.9MW at full load. Such efficiency has placed Roadrunner in third place on the Green 500 list of most efficient supercomputers.

Otherwise, Roadrunner is air-cooled, utilizing large copper heat sinks and variable-speed fans.

What Comes after Roadrunner?

Despite Roadrunner's quantum leap into petascale computing, it is merely the beginning of an exciting trend. IBM's Grice spoke of efforts in Europe to re-invigorate supercomputing there, with plans in the pipeline for multi-petaflop machines on-line by 2010. IBM also is planning in the tens of petaflops with Los Alamos and Sandia National Laboratories, including a 50-petaflop machine slated for delivery in 2012 or 2013. "We're going to have an exaflop in 11 years", adds Grice, "so we just have to figure out how to power it". The trend has been amazingly linear, and given the advances in hybrid computing, it likely will continue unabated.

Roadrunner also will raise expectations, and hybrid computing will trickle down, making the once-impossible possible. Climate-change scientists will heap more elements to their models, pharmaceutical companies will model the effects of drugs in the body, and Hollywood's special-effects will become even more mind-blowing.

As this future unfolds, the Roadrunner teams at IBM and Los Alamos National Lab should be confident in their accomplishment of building the world's fastest supercomputer—the first-ever petaflop machine. It was an incredible achievement in planning, hardware, software and logistics that has set the global standard for supercomputing. It will be interesting to see what the team will accomplish next. ■

James Gray is Linux Journal Products Editor and a graduate student in environmental sciences and management at Michigan State University. A Linux enthusiast since the mid-1990s, he currently resides in Lansing, Michigan, with his wife and cats.

Resources

IBM Fact Sheet on Roadrunner: www-03.ibm.com/press/us/en/pressrelease/24405.wss

Roadrunner Home Page at Los Alamos National Lab: www.lanl.gov/orgs/hpc/roadrunner/index.shtml

The Green 500 List: www.green500.org



Experience Lightning Without The ThunderSM



► WhisperStation-Pro™

1 TERAFLOP IN A COOL, FAST, RELIABLE PLATFORM!

Whether it's Wall Street, Main Street or Your Street, Microway's new Nvidia-powered WhisperStation-Pro is energy-efficient, designed for superior performance, and best of all - QUIET.

Originally designed for a group of power hungry, demanding engineers in the automotive industry, WhisperStation-Pro incorporates two AMD® Opteron™ or Intel® Xeon® quad-core processors and high-efficiency power supplies. Ultra-quiet fans and internal sound-proofing produce a powerful, but silent, computational platform.

WhisperStation-Pro configured with one Quad core processor, 4 GB high speed memory, 250 GB drive, dual-GigE, NVIDIA® Quadro™ FX570 graphics and 20" LCD – **starts at \$1995.**

You can have it configured to your exact needs with NVIDIA GeForce® or Quadro graphics adapters (including SLI®), NVIDIA Tesla™ GPU, any Linux distribution, or Windows® dual-boot. Also, there is plenty of room for RAID storage expansion. From a home based workstation for financial wizards, to a superior gaming or design station, WhisperStation-Pro fits the bill and your budget.

Visit www.microway.com for more technical information.

Hear Yourself Think Again!

Call our technical sales team at 508-746-7341 and customize your WhisperStation-Pro today.

WhisperStation™ ▶
3D Elite SLI
For Gamers



Microway
Technology you can count on™

Massively Parallel Linux Laptops, Workstations and Clusters with CUDA

Use an NVIDIA GPU or a cluster of them to realize massive performance increases.

ROBERT FARBER

NVIDIA's CUDA (Compute Unified Device Architecture) makes programming and using thousands of simultaneous threads straightforward. CUDA turns workstations, clusters—and even laptops—into massively parallel-computing devices. With CUDA, Linux programmers can address real-time programming and computational tasks previously possible only with dedicated devices or supercomputers.



Do you want to use hundreds of processing cores with Linux today? If so, take a look at running CUDA on the NVIDIA graphics processor in your Linux system. CUDA is a way to write C code for various CUDA-enabled graphics processors. I have personally written a C language application that delivers more than 150GF/s (billion floating-point operations per second) on a Linux laptop with CUDA-enabled GPUs. That same application delivers many hundreds of GF/s on a workstation and teraflop performance on a Linux cluster with CUDA-enabled GPUs on each node!

Do you need to move lots of data and process it in real time? Current CUDA-enabled devices use PCI-E 2.0 x16 buses to move data around quickly between system memory and amongst multiple graphics processors at GB/s (billion bytes per second) rates. Data-intensive video games use this bandwidth to run smoothly—even at very high frame rates. That same high throughput can enable some innovative CUDA applications.

One example is the RAID software developed by researchers at the University of Alabama and Sandia National Laboratory that transforms CUDA-enabled GPUs into high-performance RAID accelerators that calculate Reed-Solomon codes in real time for high-throughput disk subsystems (according to “Accelerating Reed-Solomon Coding in RAID Systems with GPUs” by Matthew Curry, Lee Ward, Tony Skjellum and Ron Brightwell, IPDPS 2008). From their abstract, “Performance results show that the GPU can outperform a modern CPU on this problem by an order of magnitude and also confirm that a GPU can be used to support a system with at least three parity disks

Do you want to use hundreds of processing cores with Linux today?

with no performance penalty.” I’ll bet the new NVIDIA hardware will perform even better. My guess is we will see a CUDA-enhanced Linux md (multiple device or software RAID) driver in the near future. [See Will Reese’s article “Increase Performance, Reliability and Capacity with Software RAID” on page 68 in this issue.]

Imagine the freedom of not being locked in to a proprietary RAID controller. If something breaks, simply connect your RAID array to another Linux box to access the data. If that computer does not have an NVIDIA GPU, just use the standard Linux software md driver to access the data. Sure, the performance will be lower, but you still will have immediate access to your data.

Of course, CUDA-enabled GPUs can run multiple applications at the same time by time sharing—just as Linux does. CUDA devices have a very efficient hardware scheduler. It’s fun to “wow” people by running a floating-point intensive program while simultaneously watching one or more graphics-intensive applications render at a high frame rate on the screen.

CUDA is free, so it’s easy to see what I mean. If you already have a CUDA-enabled GPU in your system (see www.nvidia.com/object/cuda_learn_products.html for compatible models), simply download CUDA from the NVIDIA Web site (www.nvidia.com/cuda), and install it. NVIDIA

provides the source code for a number of working examples. These examples are built by simply typing `make`.

Check out the gigaflop performance of your GPU by running one of the floating-point-intensive applications. Start a graphics-intensive application, such as the glxgears (an OpenGL demo application), in another window and re-run the floating-point application to see how well the GPU simultaneously handles both applications. I was really surprised by how well my GPUs handled this workload.

Don’t have a CUDA-enabled GPU? No problem, because CUDA comes with an emulator. Of course, the emulator will not provide the same level of performance as a CUDA-enabled GPU, but you still can build and run both the examples and your own applications. Building software for the emulator is as simple as typing `make emu=1`.

The reason the emulator cannot provide the same level of performance—even on high-end multicore SMP systems—is because the NVIDIA GPUs can run hundreds of simultaneous threads. The current NVIDIA Tesla 10-series GPUs and select models of the GeForce 200-series have 240 hardware thread processors, while even the highest-end AMD and Intel processors currently have only four cores per CPU.

I have seen one to two orders of magnitude increase in floating-point performance over general-purpose processors when fully utilizing my NVIDIA graphics processors—while solving real problems. Other people have published similar results. The NVIDIA site provides links to examples where researchers have reported 100x performance increases (www.nvidia.com/cuda). A recent paper, dated July 12, 2008, on the NVIDIA CUDA Zone Web page, reports a 270x performance increase by using GPUs for the efficient calculation of sum products.

Happily, these performance levels don’t require a big cash outlay. Check out your favorite vendor. The low-end, yet still excellent performing, CUDA-enabled GPUs can be purchased for less than \$120.

Why are the prices so good? The simple answer is competition. In the very competitive graphics processor market, both vendors and manufacturers work hard to deliver ever higher levels of performance at various price points to entice customers to buy or upgrade to the latest generation of graphics technology.

So, how can NVIDIA offer hundreds of thread processors while the rest of the industry can deliver only dual- and quad-core processors?

The answer is that NVIDIA designed its processors from the start for massive parallelism. Thread scalability was designed in from the very beginning. Essentially, the NVIDIA designers used a common architectural building block, called a multiprocessor, that can be replicated as many times as required to provide a large number of processing cores (or thread processors) on a GPU board for a given price point. This is, of course, ideal for graphics applications, because more thread processors translate into increased graphics performance (and a more heart-pounding customer experience). Low price-point GPUs can be designed with fewer multiprocessors (and, hence, fewer thread processors) than the higher-priced, high-end models.

CUDA was born after a key realization was made: the GPU thread processors can provide tremendous computing power if

the problem of programming tens to hundreds of them (and potentially thousands) can be solved easily.

A few years ago, pioneering programmers discovered that GPUs could be harnessed for tasks other than graphics—and they got great performance! However, their improvised programming model was clumsy, and the programmable pixel shaders on the chips (the precursors to the thread processors) weren't the ideal engines for general-purpose computing. At that time, writing software for a GPU meant programming in the language of the GPU. A friend once described this as a process similar to pulling data out of your elbow in order to get it to where you could look at it with your eyes.

NVIDIA seized upon this opportunity to create a better programming model and to improve the hardware shaders. As a result, NVIDIA ended up creating the Compute Unified Device Architecture. CUDA and hardware thread processors were born. Now, developers are able work with familiar C and C++ programming concepts while developing software for GPUs—and happily, it works very well in practice. One friend, with more than 20 years' experience programming HPC and massively parallel computers, remarked that he got more work done in one day with CUDA than he did in a year of programming the Cell broadband engine (BE) processor. CUDA also avoids the performance overhead of graphics-layer APIs by compiling your software directly to

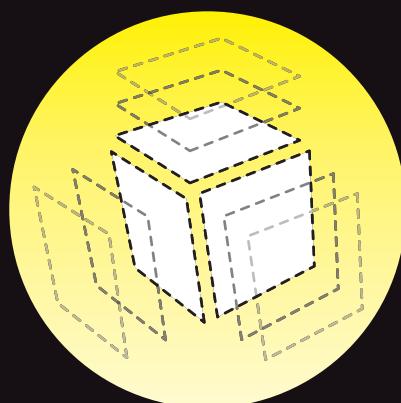
the hardware (for example, GPU assembly language), which results in excellent performance.

As a Linux developer, I especially like the CUDA framework, because I just use my favorite development tools (such as make, vi, emacs and so forth). The only real difference is that nvcc is used to compile the CUDA portions of the code instead of gcc, and the debugger and profiler are different. Optimized libraries also are available, such as math.h, FFT, BLAS and others. A big plus is that CUDA software also can be used within Python, Matlab and other higher-level languages and software.

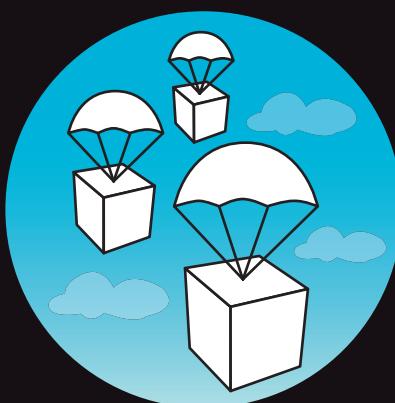
CUDA and CUDA-enabled devices are best utilized for computational kernels. A kernel is a computationally intensive portion of a program that can be offloaded from the host system to other devices, such as a GPU, to get increased performance.

CUDA is based on a hardware abstraction that allows applications to achieve high performance while simultaneously allowing NVIDIA to adapt the GPU architecture as needed without requiring massive application rewriting. Thus, NVIDIA can incorporate new features and technology into each generation of its products as needed or when the technology becomes cost-effective. I recently upgraded to the 10-series processors and was delighted to see my applications run 2x faster.

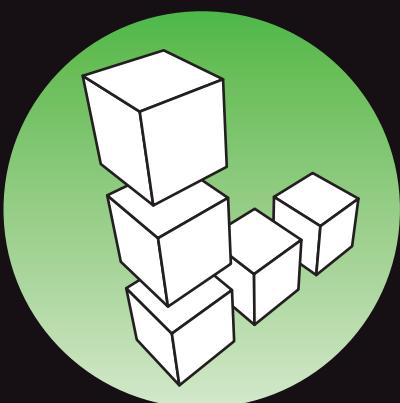
What is actually involved in writing CUDA? A number of



Develop.



Deploy.



Scale.

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

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

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



linode.com

excellent resources are available on the Internet to help you. The CUDA Zone forums are an excellent place for all things CUDA, and they have the benefit of letting you post questions and receive answers. I also recommend my *DDJ* columns, "CUDA: Supercomputing for the Masses" on the *Doctor Dobbs* Web site as well as the excellent tutorials at courses.ece.uiuc.edu/ece498/al1/Syllabus.html. Many more CUDA tutorials are listed on CUDA Zone.

Writing CUDA is greatly simplified, because CUDA automatically manages threads for the programmer. This wonderful behavior happens as a result of how concurrent threads of execution are expressed in CUDA—coupled with a hardware thread manager that runs on the CUDA-enabled device. In practice, the hardware thread manager performs amazingly well to keep the multiprocessors active and running efficiently even when confronted with non-obvious resource limitations (for example, number of thread processors, memory, memory bandwidth, latency and so on) and other issues.

Instead of creating threads explicitly (as one would do with *pthreads*, for instance), the developer simply needs to specify an execution configuration when calling the CUDA kernel. The execution configuration provides all the information the hardware thread scheduler needs to queue and asynchronously start CUDA kernels on one or more GPUs.

Syntactically, the call to a CUDA kernel looks like a C language subroutine call—except an execution configuration is added between triple angle brackets (<<< and >>>). The first

Writing CUDA is greatly simplified, because CUDA automatically manages threads for the programmer.

two parameters define the number of threads within a thread block and the number of thread blocks. (The key feature of a thread block is that threads within a thread block can communicate with each other, but not with threads outside the thread block). The total number of simultaneously running threads for a given kernel is the product of those two parameters. Other parameters in the execution configuration define a grid of threads (the 2-D or 3-D topology of the thread blocks) as well as some resource specifications.

Other than the addition of an execution configuration, a CUDA kernel call looks syntactically just like any other C subroutine call, including the passing of parameters and structures. There are no function calls to CUDA, because of the asynchronous nature of the kernel invocations. More information can be found in the tutorials or in "The CUDA Programming Guide" in the documentation.

By changing the execution configuration, the programmer easily can specify a few or many thousands of threads. In fact, the NVIDIA documentation recommends using a large number of threads (on the order of thousands) to future-proof your code and maintain high performance on future generations of GPU products. The hardware scheduler manages all the complexity in keeping as many threads active as is possible. What a wonderful problem not to need to worry about when programming!

From the programmer's point of view, the CUDA kernel

acts as if it were contained within the scope of a loop over the total number of threads—except each loop iteration runs in a separate thread of execution. Within each thread, the programmer has all the information needed to distinguish each thread from all other threads (such as thread ID within a thread block, the number of thread blocks and coordinates within the execution configuration grid). With this information, the developer then can program each thread to perform the appropriate work and with the relevant data. Once all the threads have run to completion (which looks to the programmer like the thread reached a return statement), the CUDA kernel exits. However, control returns to the serial portion of the code running on the host computer only after all scheduled CUDA kernels also have run to completion. (Recall that CUDA kernel calls are asynchronous, so many calls can be queued before a result is required from the CUDA devices.) The programmer decides where in the host code to wait for completion.

Unfortunately, this article cannot delve too deeply into any advanced topics, such as how threads communicate amongst themselves (only within a thread block), how to manage multiple asynchronous kernels (via a streams abstraction), and transferring data to and from the GPU(s), plus many other capabilities that allow CUDA to be a fully functional platform for massively parallel computing.

Although CUDA automates thread management, it doesn't eliminate the need for developers to think about threads. Rather, it helps the programmer focus on the important issues in understanding the data and the computation that must be performed, so the problem can be decomposed into the best data layout on the CUDA device. In this way, the computational power of all those threads can be brought to bear, and those wonderful 100x performance increases over commodity processors can be achieved.

One of the most important performance challenges facing CUDA developers is how to most effectively re-use data within the local multiprocessor memory resources. CUDA-enabled GPUs have a large amount of global memory (many have gigabytes of RAM). This memory is provided so the thread processors will not have to access the system memory of the host Linux system constantly. (Such a requirement would introduce a number of performance and scalability problems.) Even though global memory on the current generation of GPUs has very high bandwidth (more than 100GB/s on the new 10-series GPUs), there is unfortunately just not enough bandwidth to keep 240 hardware thread processors running efficiently without data re-use within the multiprocessors.

The CUDA software and hardware designers have done some wonderful work to hide the performance limitations of global memory, but high performance still requires data re-use within the multiprocessors. Check out the tutorials and CUDA Zone for more information. Pay special attention to the "CUDA Occupancy Calculator", which is an indispensable Excel spreadsheet to use when designing CUDA software and understanding local multiprocessor resource capabilities and restrictions.

CUDA and CUDA-enabled devices are evolving and improving with each new generation. As developers, we really want large amounts of local multiprocessor resources, because it

makes our job much easier and software more efficient. The CUDA-enabled hardware designer, on the other hand, has to focus on the conflicting goal of delivering hardware at a low price point, and unfortunately, large amounts of local multiprocessor memory is expensive. We all agree that inexpensive CUDA hardware is wonderful, so the compromise is to market CUDA-enabled hardware with different capabilities at various price points. The market then selects the best price vs. capability trade-offs.

Letting the market decide is actually a very good solution, because GPU technology is evolving quickly. Each new generation of CUDA-enabled devices is more powerful than the previous generation and contains ever greater numbers of higher-performance components, such as local multiprocessor memory, at the same price points as the previous generation.

Is CUDA appropriate for all problems? No, but it is a valuable tool that will allow you to do things on your computer that you could not do before.

One issue to be aware of is that threads on each multiprocessor execute according to an SIMD (Single Instruction Multiple Data) model. The SIMD model is efficient and cost-effective from a hardware standpoint, but from a software standpoint, it unfortunately serializes conditional operations (for example, both branches of each conditional must be evaluated one after the other). Be aware that conditional operations can have profound effects on the

runtime of your kernels. With care, this is generally a manageable problem, but it can be problematic for some issues. Advanced programmers can exploit the MIMD (Multiple Instruction Multiple Data) capability of multiple-thread processors. For more detail on Flynn's taxonomy of computer architectures, I recommend the Wikipedia article en.wikipedia.org/wiki/Flynn%27s_taxonomy.

I claim that the one or two orders of magnitude performance increase that GPUs provide over existing technology is a disruptive change that can alter some aspects of computing dramatically. For example, computational tasks that previously would have taken a year now can complete in a few days; hour-long computations suddenly become interactive, because they can be completed in seconds, and previously intractable real-time processing tasks now become possible.

Lucrative opportunities can present themselves to consultants and engineers with the right skill set and capabilities who are able to write highly threaded (and massively parallel) software.

What about you? Can the addition of CUDA to your programming skills benefit your career, applications or real-time processing needs? ■

Rob Farber is a senior scientist at Pacific Northwest National Laboratory. He has worked in massively parallel computing at several national laboratories and as co-founder of several startups. He can be reached at rmfarber@gmail.com.

SMALL, EFFICIENT COMPUTERS WITH PRE-INSTALLED UBUNTU.

GS-Lo8 Fanless Pico-ITX System

Ultra-Compact, Full-Featured Computer
Excellent for Industrial Applications



DISCOVER THE ADVANTAGE OF MINI-ITX.

Selecting a complete, dedicated platform from us is simple: Pre-configured systems perfect for both business & desktop use, Linux development services, and a wealth of online resources.



LOGIC
SUPPLY
www.logicsupply.com

Increase Performance, Reliability and Capacity with Software RAID

Linux software RAID provides a flexible software alternative to hardware RAID with excellent performance.

Will Reese

In the late 1980s, processing power and memory performance were increasing by more than 40% each year. However, due to mechanical limitations, hard drive performance was not able to keep up. To prepare for a "pending I/O crisis", some researchers at Berkeley proposed a solution called "Redundant Arrays of Inexpensive Disks". The basic idea was to combine several drives so they appear as one larger, faster and/or more-reliable drive. RAID was, and still is, an effective

way for working around the limitations of individual disk drives. Although RAID is typically implemented using hardware, Linux software RAID has become an excellent alternative. It does not require any expensive hardware or special drivers, and its performance is on par with high-end RAID controllers. Software RAID will work with any block device and supports nearly all levels of RAID, including its own unique RAID level (see the RAID Levels sidebar).

RAID Levels

RAID is extremely versatile and uses a variety of techniques to increase capacity, performance and reliability as well as reduce cost. Unfortunately, you can't quite have all those at once, which is why there are many different implementations of RAID. Each implementation, or level, is designed for different needs.

Mirroring (RAID 1):

Mirroring creates an identical copy of the data on each disk in the array. The array has the capacity of only a single disk, but the array will remain functional as long as at least one drive is still good. Read and write performance is the same or slightly slower than a single drive. However, read performance will increase if there are multiple read requests at the same time, because each drive in the array can handle a read request (parallel reads). Mirroring offers the highest level of reliability.

Striping (RAID 0):

Striping spreads the data evenly over all disks. This makes reads and writes very fast, because it uses all the disks at the same time, and the capacity of the array is equal to the total capacity of all the drives in the array. However, striping does not offer any redundancy, and the array will fail if it loses a single drive. Striping provides the best performance and capacity.

Striping with Parity (RAID 4, 5, 6):

In addition to striping, these levels add information that is used to restore any missing data after a drive failure. The additional information is called parity, and it is computed from the actual data. RAID 4 stores all the parity on a single disk; RAID 5 stripes the parity across all the disks, and RAID 6 stripes two different types of parity. Each copy of parity uses one disk worth of capacity in the array, so RAID 4 and 5 have the capacity of N-1 drives, and RAID 6 has the capacity of N-2 drives. RAID 4 and 5 require at least three drives and can lose any single disk in the array; whereas RAID 6 requires at least four drives and can withstand losing up to two disks. Write performance is relatively slow because of the parity calculations, but read performance is usually very fast as the data is striped across all of the drives. These RAID levels provide a nice blend of capacity, reliability and performance.

Nesting RAID Levels (RAID 10):

It is possible to create an array where the "drives" actually are other arrays. One common example is RAID 10, which is a RAID 0 array created from multiple RAID 1 arrays. Like RAID 5, it offers a nice blend of capacity, reliability and performance. However, RAID 10 gives up some capacity for increased performance and reliability, and it requires a minimum of four drives. The capacity of RAID 10 is

half the total capacity of all the drives, because it uses a mirror for each "drive" in the stripe. It offers more reliability, because you can lose up to half the disks as long as you don't lose both drives in one of the mirrors. Read performance is excellent, because it uses striping and parallel reads. Write performance also is very good, because it uses striping and does not have to calculate parity like RAID 5. This RAID level typically is used for database servers because of its performance and reliability.

MD RAID 10:

This is a special RAID implementation that is unique to Linux software RAID. It combines striping and mirroring and is very similar to RAID 10 with regard to capacity, performance and reliability. However, it will work with two or more drives (including odd numbers of drives) and is managed as a single array. In addition, it has a mode (raid10,f2) that offers RAID 0 performance for reads and very fast random writes.

Spanning (Linear or JBOD):

Although spanning is not technically RAID, it is available on nearly every RAID card and software RAID implementation. Spanning appends disks together, so the data fills up one disk then moves on to the next. It does not increase reliability or performance, but it does increase capacity, and it typically is used to create a larger drive out of a bunch of smaller drives of varying sizes.

Getting Started

Most Linux distributions have built-in support for software RAID. This article uses the server edition of Ubuntu 8.04 (Hardy). Run the following commands as root to install the software RAID management tool (mdadm) and load the RAID kernel module:

```
# apt-get install mdadm  
# cat /proc/mdstat
```

Once you create an array, /proc/mdstat will show you many details about your software RAID configuration. Right now, you just want to make sure it exists to confirm that everything is working.

Creating an Array

Many people like to add a couple drives to their computer for extra file storage, and mirroring (RAID 1) is an excellent way to protect that data. Here, you are going to create a RAID 1 array using two additional disks, /dev/sda and /dev/sdb.

Before you can create your first RAID array, you need to partition your disks. Use fdisk to create one partition on /dev/sda, and set its type to "Linux RAID autodetect". If you are just testing RAID, you can create a smaller partition, so the creation process does not take as long:

```
# fdisk /dev/sda  
> n
```

```
> p  
> 1  
> <RETURN>  
> <RETURN>  
> t  
> fd  
> w
```

Now, you need to create an identical partition on /dev/sdb. You could create the partition manually using fdisk, but it's easier to copy it using sfdisk. This is especially true if you are creating an array using more than two disks. Use sfdisk to copy the partition table from /dev/sda to /dev/sdb, then verify that the partition tables are identical:

```
# sfdisk -d /dev/sda | sfdisk /dev/sdb  
# fdisk -l
```

Now, you can use your newly created partitions (/dev/sda1 and /dev/sdb1) to create a RAID 1 array called /dev/md0. The md stands for multiple disks, and /dev/mdX is the standard naming convention for software RAID devices. Use the following command to create the /dev/md0 array from /dev/sda1 and /dev/sdb1:

```
# mdadm -C /dev/md0 -l1 -n2 /dev/sda1 /dev/sdb1
```

When an array is first created, it automatically will begin initializing (also known as rebuilding). In your case, that means

It does not require any expensive hardware or special drivers, and its performance is on par with high-end RAID controllers.

making sure the two drives are in sync. This process takes a long time, and it varies based on the size and type of array you created. The /proc/mdstat file will show you the progress and provide an estimated time of completion. Use the following command to verify that the array was created and to monitor its progress:

```
# watch cat /proc/mdstat    # ctrl-c to exit
```

It is safe to use the array while it's rebuilding, so go ahead and create the filesystem and mount the drive. Don't forget to add /dev/md0 to your /etc/fstab file, so the array will be mounted automatically when the system boots:

```
# mkfs.ext3 /dev/md0  
# mkdir /mnt/md0  
# mount /dev/md0 /mnt/md0  
# echo "/dev/md0 /mnt/md0 ext3 defaults 0 2" >> /etc/fstab
```

Once the array is finished rebuilding, you need to add it to the mdadm configuration file. This will make it easier to manage the array in the future. Each time you create or

modify an array, update the mdadm configuration file using the following command:

```
# mdadm --detail --scan >> /etc/mdadm/mdadm.conf  
# cat /etc/mdadm/mdadm.conf
```

That's it. You successfully created a two-disk RAID 1 array using software RAID.

Replacing a Failed Disk

The entire point of a RAID 1 array is to protect against a drive failure, so you are going to simulate a drive failure for /dev/sdb and rebuild the array. To do this, mark the drive as failed, and then remove it from the array. If the drive actually failed, the kernel automatically would mark the drive as failed. However, it is up to you to remove the disk from the array before replacing it. Run the following commands to fail and remove the drive:

```
# mdadm /dev/md0 -f /dev/sdb1  
# cat /proc/mdstat  
# mdadm /dev/md0 -r /dev/sdb1  
# cat /proc/mdstat
```

Notice how /dev/sdb is no longer part of the array, yet the array is functional and all your data is still there. It is safe to continue using the array as long as /dev/sda does not fail. You now are free to shut down the system and replace /dev/sdb when it's convenient. In this case, pretend you did just that. Now that your new drive is in the system, format it and add it to the array:

```
# sfdisk -d /dev/sda | sfdisk /dev/sdb  
# mdadm /dev/md0 -a /dev/sdb1  
# watch cat /proc/mdstat
```

The array automatically will begin rebuilding itself, and /proc/mdstat should indicate how long that process will take.

Managing Arrays

In addition to creating and rebuilding arrays, you need to be familiar with a few other tasks. It is important to understand how to start and stop arrays. Run the following commands to unmount and stop the RAID 1 array you created earlier:

```
# umount /dev/md0  
# mdadm -S /dev/md0  
# cat /proc/mdstat
```

As you can see, the array no longer is listed in /proc/mdstat. In order to start your array again, you need to assemble it (there isn't a start command). Run the following commands to assemble and remount your array:

```
# mdadm -As /dev/md0  
# mount /dev/md0  
# cat /proc/mdstat
```

Sometimes it's useful to place an array in read-only mode. Before you do this, you need to unmount the filesystem (you

can't just remount as read-only). If you try to place an array in read-only mode while it is mounted, mdadm will complain that the device is busy:

```
# umount /dev/md0
# mdadm -o /dev/md0
# cat /proc/mdstat
# mount /dev/md0
```

Notice how /dev/md0 is now read-only, and the filesystem was mounted as read-only automatically. Run the following commands to change the array and filesystem back to read-write mode:

```
# mdadm -w /dev/md0
# mount -o remount,rw /dev/md0
```

mdadm can be configured to send e-mail notifications regarding the status of your RAID arrays. Ubuntu automatically starts mdadm in monitoring mode for you; however, it currently is configured to send e-mail to the root user on the local system. To change this, edit /etc/mdadm/mdadm.conf and set MAILADDR to your e-mail address, then restart the mdadm daemon:

```
# vim /etc/mdadm/mdadm.conf
```

Set MAILADDR to <your e-mail address>, and then do:

```
# /etc/init.d/mdadm restart
```

Run the following command to test your e-mail notification setup:

```
# mdadm --monitor --scan -t -1
```

Converting a Server to RAID 1

If you are building a new server, you can use the Ubuntu Alternate install CD to set up your system on a software RAID array. If you don't have the luxury of performing a clean install, you can use the following process to convert your entire server to a RAID 1 array remotely. This requires your server to have an additional drive that is the same size or larger than the first disk. These instructions also assume you are using the server edition of Ubuntu 8.04 (Hardy), but the process is similar in other Linux distributions. You always should test a procedure like this before performing it on a production server.

Install mdadm and verify that the software RAID kernel module was loaded properly:

```
# apt-get install mdadm
# cat /proc/mdstat
```

Copy the partition table from your first drive to your second drive, and set the partition types to "Linux RAID autodetect":

```
# sfdisk -d /dev/sda | sfdisk /dev/sdb
# fdisk /dev/sdb
> t
```

```
> 1
> fd
> t
> 2
> fd
> w
```

Create the RAID 1 arrays for the root and swap partitions, and update the mdadm configuration file. This time, specify that the first drive is "missing", which will delay the rebuild until you add the first drive to the array. You don't want to mess with the first drive until you are sure the RAID configuration is working correctly:

```
# mdadm -C /dev/md0 -n2 -l1 missing /dev/sdb1 # root
# mdadm -C /dev/md1 -n2 -l1 missing /dev/sdb2 # swap
# cat /proc/mdstat
# mdadm --detail --scan >> /etc/mdadm/mdadm.conf
```

Modify /boot/grub/menu.lst so your server boots from the array:

```
# vim /boot/grub/menu.lst
```

Then:

- Add fallback 1 to a new line after default 0.
- Change the kopt line to # kopt=root=/dev/md0 ro.
- Copy the first kernel entry and change (hd0,0) to (hd1,0).
- Change root=xxx to root=/dev/md0 in the new kernel entry.

When your server is booting up, it needs to be able to load the RAID kernel modules and start your array. Use the following command to update your initrd file:

```
# update-initramfs -u
```

At this point, you can create and mount the filesystem, then copy your data to the additional drive. Make sure all of your applications are shut down and the server is idle; otherwise, you run the risk of losing any data modified after you run the rsync command:

```
# mkswap /dev/md1
# mkfs.ext3 /dev/md0
# mkdir /mnt/md0
# mount /dev/md0 /mnt/md0
# rsync -avx / /mnt/md0
```

To mount the RAID arrays automatically when your server reboots, modify /mnt/md0/etc/fstab and replace /dev/sda1 with /dev/md0, and replace /dev/sda2 with /dev/md1. You do this only on the second drive, in case you need to fall back to the old setup if something goes wrong:

```
# vim /mnt/md0/etc/fstab
```

FEATURE Software RAID

Then:

- Replace /dev/sda1 with /dev/md0.
- Replace /dev/sda2 with /dev/md1.

Make sure GRUB is installed properly on both disks, and reboot the server:

```
# grub  
> device (hd0) /dev/sda  
> root (hd0,0)  
> setup (hd0)  
> device (hd0) /dev/sdb  
> root (hd0,0)  
> setup (hd0)  
> quit  
  
# reboot
```

When your server comes back on-line, it will be running on a RAID 1 array with only one drive in the array. To complete the process, you need to repartition the first drive, add it to the array, and make a few changes to GRUB. Make sure your server is functioning normally and all your data is intact before proceeding. If not, you run the risk of losing data when you repartition your disk or rebuild the array.

Use sfdisk to repartition the first drive to match the second drive. The --no-reread option is needed; otherwise, sfdisk will complain about not being able to reload the partition table and fail to run:

```
# sfdisk -d /dev/sdb | sfdisk /dev/sda --no-reread
```

Now that your first drive has the correct partition types, add it to both arrays. The arrays will start the rebuild process automatically, which you can monitor with /proc/mdstat:

```
# mdadm /dev/md0 -a /dev/sda1  
# mdadm /dev/md1 -a /dev/sda2
```

```
# watch cat /proc/mdstat
```

Once the arrays have completed rebuilding, you safely can reconfigure GRUB to boot from the first drive. Although it is not required, you can reboot to make sure your server still will boot from the first drive:

```
# vim /boot/grub/menu.lst
```

Next, copy first kernel entry and change (hd1,0) to (hd0,0). Then:

```
# reboot
```

That completes the process. Your server should be running on a RAID 1 array protecting your data from a drive failure.

Conclusion

As you can see, Linux software RAID is very flexible and easy to use. It can protect your data, increase server performance and provide additional capacity for storing data. Software RAID is a high-performance, low-cost alternative to hardware RAID and is a viable option for both desktops and servers.■

Will Reese has worked with Linux for the past ten years, primarily scaling Web applications running on Apache, Python and PostgreSQL. He currently is a developer at HushLabs working on www.natuba.com.

Resources

Original RAID Paper: www.eecs.berkeley.edu/Pubs/TechRpts/1987/CSD-87-391.pdf

Linux RAID: linux-RAID.osdl.org/index.php/Linux_Raid

Why Software RAID?: linux.yyz.us/why-software-RAID.html

MD RAID 10: cgi.cse.unsw.edu.au/~neilb/01093607424

TECH TIP Copying a Filesystem between Computers

If you need to transfer an entire filesystem from one machine to another, for example, when you get a new computer, do the following steps.

1) Boot both PCs with any Linux live CD (for example, Knoppix), and make sure they can access each other via the network.

2) On the source machine, mount the partition containing the filesystem to be copied, and start the transfer using netcat and tar:

```
cd /mnt/sda1  
tar -czpsf - . | pv -b | nc -l 3333
```

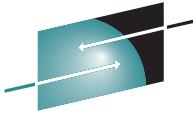
3) On the destination machine, mount the partition to receive the filesystem, and start the process:

```
cd /mnt/sda1  
nc 192.168.10.101 3333 | pv -b | tar -xzpsf -
```

The nc (netcat) command is used for any kind of TCP connections between two hosts. The pv (progress viewer) command is used to display the progress of the transfer. tar is used to archive the files on the source machine and un-archive them on the destination.

—DASHAMIR HOXHA

Technology • Connections • Results



S H A R E

Technology • Connections • Results

Discover the Possibilities with SHARE

Visit the new SHARE Web site!

SHARE proudly announces the successful launch of its newly redesigned Web site, www.share.org. The site features a new look along with upgrades to enhance your user experiences.

- Discover improved navigation
- Utilize the enhanced search functionality
- Take part in the new discussion forums
- And much more

The goal of SHARE's Web site makeover is to provide enterprise technology professionals a resource for continuous industry updates, education and networking opportunities. Stop by the new site today.



Save the Date SHARE in Austin March 1-6, 2009

Conference & Technology Exchange Expo
Austin Convention Center ■ Austin, Texas

Mark your calendar for SHARE in Austin, where you will gain maximum exposure to the industry's leading technical issues in order to solve business problems, build a network of fellow enterprise IT professionals and enhance your professional development.



Visit austin.share.org for event details as they become available.

OVERCOMING THE CHALLENGES OF DEVELOPING PROGRAMS FOR THE CELL PROCESSOR

» What has to be done to provide developers with a debugging environment that correctly represents what is happening as a program runs on a Cell processor?

CHRIS GOTTBARTH

In June 2008, Los Alamos National Lab announced the achievement of a numerical goal to which computational scientists have aspired for years—its newest Linux-powered supercomputer, named Roadrunner, had reached a measured performance of just over one petaflop. In doing so, it doubled the performance achieved by the world's second fastest supercomputer, the Blue Gene/L at Lawrence Livermore National Lab, which also runs Linux. [See James Gray's "The Roadrunner Supercomputer: a Petaflop's No Problem" on page 56 of this issue.]

The competition for prestigious spots on the list of the world's fastest supercomputers (www.top500.org) is serious business for those involved and an intriguing showcase for new technologies that may show up in your data center, on your desk or even in your living room.

One of the key technologies enabling Roadrunner's remarkable performance is the Cell processor, collaboratively designed and produced by IBM, Sony and Toshiba. Like the more-familiar multicore processors from Intel and others, the Cell incorporates multiple cores so that multiple streams of computation can proceed simultaneously

within the processor. But unlike multicore, general-purpose CPUs, which typically include a set of four, eight or more cores that behave essentially like previous-generation processors in the same family, the Cell has two different kinds of processing cores. One is a general-purpose processor, referred to as the Power Processing Element (PPE), and the remaining (eight in the current configurations) are highly optimized for performing intensive single-precision and double-precision floating-point calculations. These eight cores, called Synergistic Processing Elements (SPEs), are capable of performing about 100 billion double-precision floating-point operations per second (100 Gflops).

The Cell processor also powers the Sony PlayStation 3 and is available from IBM in blade format for clusters and the data center. With a wide range of systems using the Cell, it is interesting to speculate on the importance it will have on the market. Although a number of significant factors beyond flop ratings will contribute to its success or failure, traditional considerations, such as price, availability and reliability, still will be important. Power efficiency and the difficulty or ease of adapting applications to take advantage of the architecture

also are critical factors. If only a slim subset of applications can take advantage of a processor, or if application adaptation requires an investment of time and energy disproportionate to its benefits, how widely will it be adopted?

Recognizing the significant interest in the Cell and the accompanying concern about programmability, TotalView Technologies recently introduced a version of the TotalView debugger specifically adapted for debugging on the Cell. Doing so required significant changes to the way that the debugger models what takes place within threads and processes in order to provide computational scientists and developers with a clear representation of what is happening in a Cell program.

This article outlines the challenges faced in writing or porting applications to the Cell, some of the elements of the Cell environment and typical Cell programs that make understanding what is happening especially difficult, and specific actions users can take to develop and troubleshoot Cell programs effectively. Many of the same issues encountered in debugging Cell applications also arise with other techniques, such as general-purpose computing on graphics processing units (or GPGPU programming).

The Challenges of the Cell Architecture

The architecture of the Cell presents two general challenges software developers must solve when writing new software or porting existing software to the Cell. The first challenge is breaking the key components of a program into small chunks that can execute on the eight SPEs. In numerical programs, this often means dividing the data into small independent units. In other cases, individual tasks or pipeline stages may be delegated to specific SPEs. This issue of problem decomposition is similar to that of adapting an application to a distributed memory cluster environment, although the granularity is different due to the limited memory space directly available to each SPE.

Each of the eight SPE cores that is part of the Cell processor has its own independent registers and a small amount of local memory (256KB) used for storing instructions and data. This memory acts similarly to a cache in a general-purpose processor, in that it has a limited size and can be read and written very quickly. Unlike a cache, however, its contents are managed directly by the application. The code units running on the SPE elements are allowed to initiate direct memory access operations that

can copy chunks of data from the main memory into the SPE local store or back to main memory from the local store. The same memory is used for the machine instructions and global memory, heap memory and stack. The heap and the stack change size over time, and the programmer must take care to avoid collisions between these different memory ranges, because there is no memory protection. This explicit management of memory is the second major challenge in designing Cell implementations. Because each processor has a completely separate local store, the structure of a program for the Cell is very different from the structure of a traditional multi-threaded application, where all the threads share the same memory.

Developers who want to write code for the Cell processor will need to find ways to address these two issues in their programs. Architecture abstraction layers, like the one provided by RapidMind, may allow developers to write code that will run on a range of processors, including the Cell. If developers are going to obtain the best possible performance, however, they need to make use of the control obtained by programming for the SPEs themselves, at the expense of longer and more complex code.

The Cell provides exceptional performance by making it possible to achieve and sustain a high level of concurrent execution. Each SPE element can execute simultaneously and asynchronously, and has 128-bit wide registers and support for vectorized instructions that can apply a mathematical operation synchronously to more than one number at a time. Performance gains do come with trade-offs, however, including increased complexity and unpredictability. When working

with a serial application in which all computations happen in a single sequence, the system tends to be highly predictable. Given a set of input conditions, a serial program tends to behave the same way every time, even if it is malfunctioning.

Concurrent applications, especially malfunctioning concurrent applications, may behave in different ways if the sequence of operations in the various threads differs from run to run. These differences lead to elusive race conditions when only some sequences provide the correct behavior. In order to eliminate race conditions, developers can use synchronization constructs, such as barriers, to constrain the set of execution sequences whenever two or more threads need to read or write the same data. Synchronization needs to be applied very carefully, as its misuse will result in reduced performance or deadlocks.

The Experience of Adapting TotalView to the Cell

TotalView is a highly interactive graphical source code debugger that provides developers working in C, C++ or FORTRAN with a way to explore and control their programs. Originally designed to debug one of the first distributed memory architectures, the BBN Butterfly, TotalView has since been enhanced continually with a focus on multiprocess and multithreaded applications. Today, TotalView is used to develop, troubleshoot and maintain applications in a wide range of situations. One group uses it to develop Linux-based commercial embedded computing applications consisting of a single process with a hundred or more separate threads that simultaneously interact with a graphical user interface (GUI), sensors, on-line databases and network services. Other users troubleshoot sophisticated mathematical models of complex physical systems in astrophysics, geophysics, climate modeling and other areas. These models typically consist of thousands of separate processes that run on large-scale high-performance computing (HPC) clusters on the top 500 list. In both cases, TotalView provides users with a debugging session in which they can examine in detail any one of the many threads or processes that make up the application, control each thread or process singly or as a part of various predefined and user-defined groups,

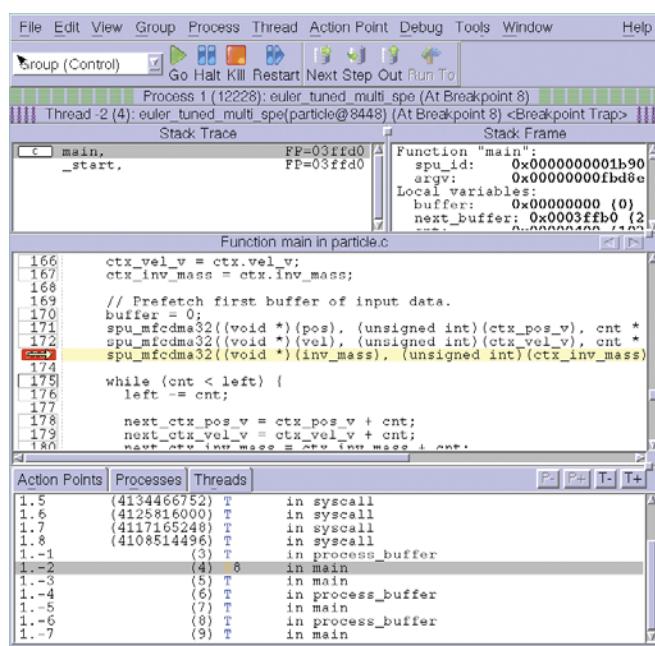


Figure 1. A simple display of code running on a Cell processor. In this example, one of the SPE threads is requesting data from main memory in preparation for a computation. Other SPE and PPE threads are stopped in a variety of states.

and display multiple types of data and state information from across many processes and threads.

The version of TotalView for the Cell applies these same user-interface techniques and capabilities to the multiple threads of execution that are running on the PPE and SPEs within each Cell processor. It also extends to Cell applications that run as distributed memory parallel applications on clusters of Cell BE blades. In order to extend this interface, however, it was necessary to extend significantly the definitions and models of processes and threads that the debugger uses internally to represent what is happening.

TotalView's traditional model for programs has three tiers: threads, processes and groups of processes. A thread is a single execution context, with its own stack (and, therefore, its own copy of any automatic or local variables) and its own state (either running or stopped). A thread also may have additional characteristics, such as a handle to "thread private" memory, an identity related to a specific OpenMP construct in another thread that is part of the application, or an operating system-assigned thread ID. A thread shares a single virtual memory address space, including both text (instructions) and data (global variables, heap memory and stack memory) sections, with the other threads that may make up a process.

A process is made up of one or more threads that are all executing a single executable "image" that represents a program. The image involves the base program and a list of library images that are loaded either at launch or during runtime. A process also has a number of resources attached to it that are managed by the operating system, including a virtual memory address space (shared with its threads), file handles, parent and/or child relationship with other processes and an operating system-assigned process ID.

Groups are made up of one or many processes running on one or more machines. These processes can be all executing copies of the same program image or, more rarely, even a different image. TotalView notes which processes in a group share the same image and shares and/or reuses information wherever possible. When the user sets a breakpoint on a line of code within a

The screenshot shows the TotalView debugger interface. The menu bar includes File, Edit, View, Tools, Window, and Help. The title bar says "1.6". The main window has tabs for Expression, Slice, and Type. The Expression tab is active, showing "argv" in the Address field (0x0003ffc0) and "unsigned long long" in the Type field. Below this is a table with columns Thread, Address, and Value. The table lists eight threads (1.7 to 1.1) with their respective addresses and values. All threads have the same address (0x0003ffc0) but different values, demonstrating local storage for each thread.

Thread	Address	Value
1.7 (4117165248)	0xf566e500	<Has no matching call frame>
1.8 (4108514496)	0xf4e2e500	<Has no matching call frame>
1.-1 (3)	0x0003ffc0	0x00000000fa64e3e0 (4200915936)
1.-2 (4)	0x0003ffc0	0x00000000fa64e410 (4200915984)
1.-3 (5)	0x0003ffc0	0x00000000fa64e440 (4200916032)
1.-4 (6)	0x0003ffc0	0x00000000fa64e470 (4200916080)
1.-5 (7)	0x0003ffc0	0x00000000fa64e4a0 (4200916128)
1.-6 (8)	0x0003ffc0	0x00000000fa64e4d0 (4200916176)
1.-7 (9)	0x0003ffc0	0x00000000fa64e500 (4200916224)

Figure 2. A simple display of variable data as seen by the SPE threads of a Cell program. In this example, there are eight SPE threads executing the same bit of code. All eight have a variable named args at 0x3ffc0, but because each SPE has its own local store, the value of this variable can be different for each thread.

group, TotalView ensures that the breakpoint appears at the right place in each process that shares the same image, even if those images have different offsets (which can happen due to randomized library load addresses across the many nodes of a cluster).

In order to support the Cell processor correctly, several changes to this model were necessary. Most significant, the SPE threads that make up a Cell process have a more distinct identity than threads created on a traditional processor. The SPE core uses a different instruction and register set from the PPE core, so the Cell version of TotalView can handle both, transparently switching between them depending on the kind of thread under operation. The SPE thread actually executes in a separate local memory space, so all the characteristics of memory space traditionally associated with processes are now associated in TotalView with threads on the Cell. As a result, looking up a variable or function in any specific SPE thread or in the PPE thread will give very different results, depending on the variables and functions that exist within the memory local to that thread.

And, this is important to represent clearly what actually is happening in the program. Depending on the Cell executable's construction, the SPE threads may look like separate programs (which happen to do all of their IO via DMA

calls) with their own main() routine. Much of the key information about the code running in the SPE threads is different from the PPE process that contains it. User-defined data types, functions and global variables that accidentally or intentionally have the same name in both the SPE and PPE probably refer to distinct things, and the code may be different between two different SPE threads that are part of the same process.

A single Cell program may load many distinct images at the same time to run on different SPEs, and a breakpoint set on a specific line of a specific SPE thread may or may not need to be duplicated across other threads (the user can choose). It certainly shouldn't be set at that same address in threads that don't execute that line of code there. Because the threads are executing and issuing DMA memory read and write requests and synchronization operations around those requests, there is a significant opportunity for race conditions and deadlocks. TotalView provides a set of thread control commands that allow developers to approach these problems systematically by exploring specific sequences of execution among the various SPE and PPE threads.

The intensive explicit memory management required for each unit of data moved in and out of the SPE local memory means that developers are concerned with low-level issues of data

representation, layout and alignment. As they are moving units of memory from the PPE to the SPE and back again, it is quite common to want to examine the data on both sides of the transfer. TotalView provides the ability to explore the data in any thread in exactly the same terms as it is being used within that thread. This is a significant benefit. One computational scientist noted that before TotalView was available on the Cell, he had to perform offset calculations with a calculator every time he wanted to look at a variable on the SPE.

To understand the data that has been transferred to or from the SPE threads, scientists and programmers can navigate data structures and large arrays, following pointers as if they were HTML links in a browser, and sort, filter and graph the data they find.

The TotalView process group model applies to the version on the Cell with very little modification. Users can work freely with large groups of processes running on one or many blades. The only change is a more nuanced model that allows breakpoints to be shared between threads that are all executing the same image, rather than between processes executing the same image.

The changes made to TotalView in order to provide for clear troubleshooting on the Cell processor highlight the architectural distinctions that are uniquely challenging for scientists and programmers writing software for the Cell. The capabilities that were developed in response to these distinctions make the process of adapting software to the Cell a bit less daunting for those scientists and programmers.

These lessons may apply in other situations. For example, a program written for the Cell somewhat resembles a program written to take advantage of graphics co-processors as computational accelerators. This technique, called GPGPU (General-Purpose Graphics Processing Unit) programming, also involves writing small bits of code, called kernels, which will execute independently on specialized hardware using a separate device memory for local store. GPGPU programming introduces the additional notion of extremely lightweight context swapping and encourages developers to think of creating thousands of threads, where each might operate on a single element of a large array (a stream, in

GPGPU context). Many of the same programming and debugging challenges exist for this model, and future debuggers for GPGPU should give developers clear ways of examining the stream data in both main memory and the GPU store by accurately representing what is happening in the GPU, and displaying the state of thousands of threads to the user.

Developers who work with multi-threaded applications on other platforms can thank their caches that they don't

need to deal with explicitly loading data into and out of each thread. They can concentrate instead on the challenge of finding the right level of concurrency for their application, their processor and its cache. ■

Chris Gottbrath is Product Manager for the TotalView debugger and MemoryScape product lines at TotalView Technologies, where he focuses on making it easier for programmers, scientists and engineers to solve even the most complex bugs and get back to work. He welcomes your comments at chris.gottbrath@totalviewtech.com.

ServerBeach by geeks, for geeks™



When **YouTube** first started to experience its exponential growth and our hosting needs changed, ServerBeach offered us great flexibility. They continually redesigned our streaming architecture for optimum performance while keeping our hosting costs in check.

STEVE CHEN Founder | **YouTube** 

ValuePack *(always included)*

- 24/7 live customer service
- 24/7 ticketing system
- Personal account manager
- Lots of bandwidth
- Free OS reloads
- Free Rapid Reboot
- Free Rapid Rescue
- Super fast PEER 1 network
- Rock-solid IT infrastructure
- 100% uptime guarantee
- Choose your data center - East Coast, West Coast and Central

serverbeach.com **1.800.741.9939**

A PEER 1 COMPANY

Cory Doctorow—Linux Guru?

Cory Doctorow has “open sourced” every one of his books, so to us, he seems like the perfect candidate to use Linux. He agrees, and tells us about it. DAN SAWYER

Cory Doctorow’s history and credentials would take the space of this entire article to list properly, so I’m going to do a decidedly improper job, and I hope you’ll bear with me.

An internationally sought-after activist on copyright and DRM issues, Doctorow is an ardent proponent of the Creative Commons, releasing all his own work under CC licenses on the Internet, regardless of their primary publishing format. A lifelong Mac devotee, a bit more than a year ago, he switched to Linux. I caught up with him on his *Little Brother* book tour and arranged an interview to talk about his experience with Linux. What follows is an excerpt of the conversation.

Note: to those of you who have never met him or who plan to interview him on your podcast, be aware that I caught him on an off day. When we talked, he was jet-lagged and exhausted, and he still spoke at a consistent 148 words per minute. And for him, that’s slow!

DS: What’s the deal with your Linux switch? You used to be a Mac guy, right?

CD: Yup, I mean, obviously, always Linux on the server, yeah, but I was always Mac on the desktop for a long time, starting with Apple II Plusses in 1979, and then Macs all my life—one or two a year minimum over the years.

DS: What prompted the switch, and what do you think so far?

CD: Well, there are a couple things—the DRM stuff keeps getting worse and worse. It seems like every time I turned around, Apple is doing something with its OS to add more bullshit to it. More DRM, more controls on how users use it....They’re anti-features. There’s no customer who woke up this morning and said, “Gosh, I wish there was a way I could do less with my music this morning—I hope there’s an iTunes update waiting for me.”

So, it just seemed to me, increasingly, that Apple wasn’t making computers to suit my needs; they were making computers to suit the needs of some theoretical entertainment giant. And, you know, I think that’s their business if they want to do it, but they’re not a charity, so if they don’t want to make the stuff I want to buy, I don’t have to buy it. Which is exactly what I did—I stopped buying it.

DS: And this despite the noises that Steve Jobs keeps making about removing DRM from the iTunes store?

CD: ...they keep saying that. Meanwhile...Audible...has the exclusive contract to deliver audiobooks to the iTunes store—and is now actually the largest audiobook seller in the world by far (they’re owned by Amazon now), and Audible won’t turn the DRM off on their audiobooks, even when the authors and the publishers ask them to. And, it’s not just when it’s a



Cory Doctorow

weird little indie. With my latest book *Little Brother*, the audio edition is published by Random House audio—they’re the largest publisher in the world; they’re part of Bertlesmann....So, you know, if Steve Jobs really felt like he didn’t want DRM, he would be setting things up so there were competitors of Audible in the iTunes store who were offering stuff without DRM.

...Meanwhile you have Steve Jobs running around saying—at one point he made this big speech—how you should never allow your videos to be available in HD if you’re a movie studio executive, unless you are assured that no one is going to make an HD burner or ripper that will be capable of making your DVDs....So it seems awfully mealy-mouthed, basically.

So [the switch to Linux] just seemed like the right idea. And then the other thing that happened was that I wanted a computer in a “color other than black”....There is a certain elegance to going into an Apple store and saying, “well, I want a Pro machine”, and they say, “well, this is the Pro machine and you can get it in fast, faster or fastest”. But at the same time, it was awfully refreshing to walk into the EmperorLinux Web store and have this incredible variety of CPUs to choose from. And Emperor’s been pretty good to me. I’m on my second machine from them now. They have very, very responsive service. Every once in a while, I run into some problems that I think are endemic to Linux—where, I think in hindsight I made a mistake. I bought the next model up from the tiny, little ThinkPad I’ve been loving. I had an x50 and decided I wanted an x51 for the additional speed and power. And, frankly, the drivers weren’t there and still aren’t entirely there.

DS: Other than driver issues, what other problems or good points have you run into so far?

CD: Well, for the good stuff, it's all around Synaptic for me—and it's not just Synaptic as a package manager, but it's just the idea that there is a suite of tools that are guaranteed to work with your machine (more or less), where the cost of pulling the wrong tool is zero, and where they're open—so, if there are problems you can get them fixed. So that's really nice.

Let's say I need vector graphics—well, let's just go into Synaptic and type "vector" and see what comes up. All right, there's nine packages—let's try them all! Okay, this one's good.

That's pretty cool—that's actually really, really cool.

The downside has been that, especially for larger projects, the process by which you submit bugs—especially hard bugs—is way beyond my ability as a user to fiddle with....

...If I want to submit a bug to OpenOffice.org, and it forces me to do an e-mail loop to the Bugzilla, and the e-mail loop success takes 14 hours to go through, and the e-mail loop confirmation message goes to my junk-mail folder, so I have to spend 14 hours checking my junk-mail folder, remembering over and over again that I have to check it. And then when I'm through it, actually filing the bug against Bugzilla is crazy. It's just not an easy procedure.

DS: I've gotten taken to task a lot with reader feedback for the reviews I write, because I don't always submit bugs when I bitch about things, but it's for exactly this reason. I'm not a coder.

CD: Yeah. And I totally understand why they want a membrane between submitters and programmers. I have a membrane between me and people who want to submit stuff to BoingBoing, which is the "submit a link" file. The fact is that the "submit a link" page—when we've turned that off and we've given people easier ways to send us feedback and send us posts—the proportion of good stuff we got barely changed, but the proportion of bad stuff we got went way up. And by "bad stuff", I mean spam, stupid suggestions and so on....So I think it's kind of a truism that "fools and wreckers look for easy targets", whereas people of good will tend to hunt around. The problem is that it's very frustrating when you're experiencing a bug, and it's often nearly impossible to really get to it.

...And The GIMP too. I use The GIMP every day, all day long, and there's a persistent bug...that shows up when if I open a file, edit the file, and then delete the file, The GIMP crashes. And, that's my actual standard work flow, which is, take a screenshot of a Web page that I'm going to put on BoingBoing, open it in The GIMP to edit it. Save it. Upload it to my server, then delete it. And then The GIMP crashes. So basically, every time, I have to wait for The GIMP to load up.

It's kind of a dumb bug, and my guess is that it has to do with the "recent files" menu, but it's just a general pain in the ass, and I've never figured out how to submit bugs up to The GIMP. So yeah, it's that kind of thing I find very, very frustrating, but I understand why it's there. And I also have problems with drivers or proprietary hardware—which I understand is

not Linux's fault—but that lack of driver stuff is very, very frustrating at times. Like I have a Wi-Fi bug with my Wi-Fi card that's apparently well known....So it doesn't work properly with B networks. I was staying at a hotel all last weekend in Los Angeles that had a B network, and...after about five minutes, the machine would lose the ability to route packets until I rebooted it....And then, if I tried to open too many sockets at once—like to get my RSS reader to run—then it would just crash entirely. The machine would hard-freeze.

So this was unbelievably frustrating. I've encountered bugs of that gravity in the Mac OSes. At least with this stuff, I was able to call up someone I know at Canonical and say I had this problem, and he said, "Yes, that's a well-known problem and we don't know how to fix it." But at least it was a well-known problem...

DS: And you know not to spend your whole day banging your head against the wall trying to fix it.

CD: Yeah, that's it exactly.

DS: You got into Linux because Apple squeezed you out with the DRM shit. Beyond just giving us an operating system that's not governed by that kind of crappy politics, what other significance do you think Linux has in the copyright and patent and IP wars?

CD: ...In every DRM negotiation...there's always, at some point, some coercive mechanism by which the manufacturers and the consortium get together and require of the implementers that they implement in a way that is resistant to user modification. It's kind of easy to see why you would want DRM to be resistant to user modification, because the point of DRM is that you're designing a computer that's adverse to its user. So, if you're going to be adverse to the user, it doesn't do to have the user be able to modify the system, because the user is the attacker in that model.

This is the opposite of FOSS....There's no such thing as open-source DRM for that reason—and to the extent that there is, it involves things like code signing, which is really outside the spirit of open-source software—it violates the Four Freedoms if not the letter of the license....So wherever you have a DRM consortium you have a conspiracy to fight open source, and wherever that happens, you have a really good, chewy policy argument, because open source is generally considered by most IT ministries and policy-makers and so on to be of really important value to national economies, national autonomy, national security and all of that stuff. So, creating a mandate, as they tried to do with the Broadcast Flag, requires that the government would require of hardware designers that they design their hardware to resist user modification is such a nonstarter when put in those terms. When put in the terms that "well, you realize that this is a prohibition on FOSS", that really gives us a lot of power to derail those DRM mandates. DRM always involves some kind of mandates....

DS: There's no market demand on it.

CD: Right. Some manufacturers might have an incentive to do it because they'll be offered some kind of special privilege by

the entertainment industry...but when that happens, it *has* to be everybody. It has to be all the manufacturers that go along with it; otherwise, you wind up with a situation like you had a couple years ago where the big "legit" manufacturers were abiding by the Region Controls in DVDs and all the little guys weren't, and the little guys were clobbering the big guys because everyone wants region-free DVD players.

DS: Yup. I literally walked down to Wal-Mart and paid \$30 for a region-free player.

CD: Yeah, and then what you end up with is that the big guys turn around and say, "Look, I know we agreed that we'd implement this region coding stuff, but we're not going to implement this region coding stuff", because tiny, little two-man outfits in garages in Asia are kicking Sony's ass and Sony can't be having that.

So, you end up with this kind of Prisoner's Dilemma.

DS: Particularly in a situation where you have Sony owning movie studios as well, you've got an internal fight going on at the vertically integrated companies.

CD: Absolutely. So Sony or whoever needs an assurance that everyone's going to play ball, and without a mandate, that doesn't happen. So that means that wherever the mandate is arriving, if you can show the policy-makers who will be making the mandate stick that they're about to ban FOSS, it can often sway the debate, so that's kind of a big way in which DRM and FOSS kind of interact with each other, and in which FOSS is so vital to that debate.

I'm sure there are other ways. Obviously, between CC and FOSS there's the kind of QED—the demonstration that you can

"But at the same time, it was awfully refreshing to walk into the EmperorLinux Web store and have this incredible variety of CPUs to choose from."

do it a different way, that there isn't just one way of doing it. And so wherever people say, "we need higher fences and stronger laws, otherwise no one will invest and no innovation will take place, and there will be no good equipment—no good software, no good hardware and so on", then to the extent that you have FOSS in the marketplace that eschews that and CC licenses that eschew that, you've got something very powerful as well.

[Note: Doctorow then went on to tell of successes in this area, such as the defeat of Informem in Europe. The discussion is too lengthy to reproduce here.]

...So the last thing is, that where you have this stuff available at a low cost and low barrier to entry, it creates in users a set of expectations of what they can and can't do with media, so I think that it doesn't necessarily naturally occur to people that, for example, you can record a television show to a DVD.

That doesn't always naturally occur. You kind of have to see it being done and then have it taken away from you to get worked up about it....

DS: It's human nature to expect "now" to last forever, despite the fact that we're used to an incredibly rapid rate of change and development.

CD: Exactly. So FOSS tends to be more richly featured, and so as a result it creates new expectations from users about what they can and can't do with their equipment.

DS: To swap tracks again to your other big advocacy area—privacy. What privacy tools and techniques can you recommend for beginning Linux users?

CD: I've been using GPG for a while now, and I'm actually finding it very easy to use, once I got it up and running. Although, again, it's not the best integration I was hoping for. I've got GPG running with Thunderbird, and I want it to sign every e-mail I send automatically, and I actually have to mouse over and click an icon every time I want to sign an e-mail, so if I don't remember, it won't sign all my e-mails, so it's just kind of a pain in the ass, and I can't believe there isn't a switch in an obvious place for that.

...TOR was incredibly useful to me when I was in China last year. It just seemed to me like, over and over again, the sites I wanted to visit were being blocked by the firewall, so I was able to get to them that way. And, I use TOR in other ways too—I mean, there are plenty of times when I try to get on-line, and I just find myself not able to access one site or another, and TOR just fixes it, which is great. I have FoxyProxy on Firefox, which allows me to turn on or turn off TOR automatically when I need it, and my friend Seth Shoan helped me with a little script to tunnel my mail over TOR....so I'm sending SMTP over SSH over TOR, which is great.

The Script

```
alias tortunnel='ssh -o ProxyCommand="/usr/bin/connect
➥ -S localhost:9050 %h %p" -f -N -C -l username
➥ -L5002:255.255.255.25:25 -L5003:255.255.255.25:110
➥ -L5555:localhost:5555 255.255.255.255'
```

DS: One of the things I run into with GPG or with encrypted IMs or SELinux, which ships with almost every distribution now, is that they're all commonly available, most of them are easy to use, and the vast majority of people don't. Why?

CD: Yeah. We undervalue our privacy because the cost of losing it is so far in the future, and again very speculative, so we tend to assume that because it doesn't cost us anything to lose our privacy today, it won't cost us anything to lose our privacy tomorrow, and that's generally a bad bet. So we don't worry about encrypting our hard drives until we lose our laptops—oh, and that's the other thing I do. I encrypt my hard drive, and I also just figured out how to use Cryptix with SD cards as well.

Advertiser Index

For advertising information, please contact our sales department at 1-713-344-1956 ext. 2 or ads@linuxjournal.com. www.linuxjournal.com/advertising

DS: So, tell us a bit about *Little Brother*. What's it about, why the title, and how does it tie in to your other advocacy?

CD: *Little Brother* is a novel about hacker kids in the Bay Area who, after a terrorist attack that blows up the Bay Bridge, decide that there are worse things than terrorist attacks, which, after all, end. Those things include the authoritarian responses to terrorists, which have no end, which only expand and expand. When you're fighting a threat as big and nebulous as terrorism, there's virtually no security measure that can't be justified. And so they find themselves caught inside an ever-tightening noose of control and surveillance, and they decide that they're going to fight back. They do so by doing three things: they use technology to take control of their technology, so they jailbreak all of their tools and use them to build free, encrypted wireless networks that they can communicate in secrecy with. The second thing they do is get better at understanding the statistics of rare occurrences so that they can control the debate. So they start to investigate how, when you try to stop a very rare occurrence with a security measure, the majority of things you end up stopping won't be the rare occurrence because the rare occurrence happens so rarely. So they start to show how automated surveillance and automated systems of suspicion and control disproportionately punish innocent people and rarely if ever catch guilty people.

DS: Yeah, you're actually having this problem in London now, aren't you?

CD: Oh, well, absolutely. We've got massive surveillance networks here, but it's in the US as well. You've got the hundreds of pages of no-fly-list names. People who are so dangerous that they can't be allowed to get on an airplane but so innocent that we can't think of anything to charge them with....And then, finally, they get involved in electoral politics, because no change endures unless it can be cemented into place and shellacked over with law. You might be able to convert this year's government to the cause, but...in order to make it endure, you have to make it into a law that every government that comes afterward has to abide by. And so for these three measures, they end up changing society and changing the whole world.

The novel is very explicitly didactic. Every chapter has instructions and information necessary to build technology that can help you fight the war on the war on terror. So, from setting up your own TOR node, to building a pinhole camera detector, to disabling an RFID tag, it's in the book. We did a series of "instructables"—little how-tos for building this stuff with kids that can be used as science-fair projects or home projects, and people have taken some of this stuff to heart. There's a notional Linux distro in the book called Paranoid Linux that's kind of an amalgam of all the different security-conscious Linux distros out there, and there are people trying to build a Linux distro based on Paranoid Linux, which is pretty exciting.

DS: Thank you very much for the interview Cory.■

Dan Sawyer is the founder of ArtisticWhispers Productions (www.artisticwhispers.com), a small audio/video studio in the San Francisco Bay Area. He has been an enthusiastic advocate for free and open-source software since the late 1990s, when he founded the Blenderwars filmmaking community (www.blenderwars.com). He currently is the host of "The Polyschizmatic Reprobates Hour", a cultural commentary podcast, and "Sculpting God", a science-fiction anthology podcast. Author contact information is available at www.jdsawyer.net.

Advertiser	Page #	Advertiser	Page #
181 INTERNET INC.	10, 11	MIKROTIK	C2
www.oneandone.com		www.routerboard.com	
ABERDEEN, LLC	15	OPENGEAR	21
www.aberdeencinc.com		www.opengear.com	
ASA COMPUTERS	27	POLYWELL COMPUTERS, INC.	5
www.asacomputers.com		www.polywell.com	
CARINET	87	THE PORTLAND GROUP	93
www.cari.net		www.pgroup.com	
CORAID, INC.	7	RACKSPACE MANAGED HOSTING	C3
www.coraid.com		www.rackspace.com	
EMAC, INC.	53	ROBODEVELOPMENT	89
www.emacinc.com		www.robodevelopment.com	
EMPERORLINUX	23	SERVERBEACH	77
www.emperorlinux.com		serverbeach.com	
GENSTOR SYSTEMS, INC.	39	SERVERS DIRECT	9
www.genstor.com		www.serversdirect.com	
INTEL	1	SHARE.ORG	73
www.intel.com		www.share.org	
IRON SYSTEMS	3	SILICON MECHANICS	13, 91
www.ironsystems.com		www.siliconmechanics.com	
LINODE.COM	65	SOFTWARE BUSINESS ONLINE	55
www.linode.com		www.softwarebusinessonline.com	
LINUX CERTIFIED	47	TECHLOGIC SYSTEMS	43
www.linuxcertified.com		www.embeddedx86.com	
LOGIC SUPPLY, INC.	67	VERSALOGIC CORPORATION	19
www.logicsupply.com		www.versalogic.com	
LULLABOT	45	ZT GROUP INTERNATIONAL	29
www.lullabot.com		www.ztgroup.com	
MICROWAY, INC.	C4, 61		
www.microway.com			

How We Should Program GPGPUs

Advanced compilers can simplify GPU and accelerator programming. MICHAEL WOLFE

Using GPUs for general-purpose computing is attractive because of the high-compute bandwidth available, yet programming them still is a costly task. This article makes the claim that current GPU-oriented languages, like CUDA or Brook, require programmers to do a lot of busy work and keep track of a lot of details that would be better left to a compiler. I argue that host-based, accelerator-enabled C, C++ and FORTRAN compilers are feasible with current technology and discuss the issues and difficulties with automatic compilation for GPUs and how to address them.

Programming GPUs and Accelerators

Today's programmable GPUs are the latest instantiation of programmable accelerators. Going back to the 1970s, the Floating Point Systems AP-120 and FPS-164 attached processors provided mainframe computational speeds at minicomputer cost. Many other vendors built similar products until the advent of the full mini-supercomputer. Today, we have the Clearspeed board, selling into much the same market. In the embedded world, music players and digital phones use programmable digital signal processors (DSPs) to encode and decode audio. Typically, they use dedicated hardware blocks to handle the video to minimize power requirements. In each case, the accelerator is designed to improve the performance for a particular family of applications, such as scientific computing or signal processing. GPUs are themselves (obviously) designed to speed up graphic shaders.

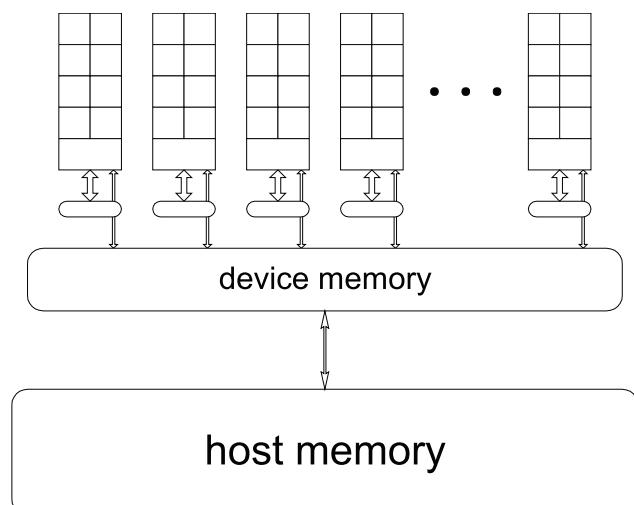


Figure 1. NVIDIA Architecture Block Diagram

Accelerators are made programmable so as to be as flexible as possible. DSPs can be reprogrammed to handle new encoding standards or to use more efficient algorithms, for example. The programming strategy always is a trade-off between performance and convenience, and efficiency and productivity. Designing a language that is close to the hardware allows a user to write as efficient a program as possible, yet such a program likely will have to be rewritten for each such accelerator. It even may need re-tuning for different generations of accelerator from the same vendor and may bear no resemblance to the corresponding CPU program from which it was ported. On the other hand, using a higher-level language puts performance in the hands of a compiler, which may leave a significant fraction of the potential on the table.

Some arguments in favor of a lower-level language are that many programmers want that level of control; compiler development is expensive and time consuming; hardware moves so quickly that compilers are out of date by the time they are tuned properly; and programmers don't trust compilers to deliver good performance.

There are similarities between these arguments and those used 50 years ago during the development of the first high-level languages. When developing the first FORTRAN compiler, the state of the art of programming was machine or assembler language. The developers at IBM realized they had to overcome a very high-acceptance barrier before anyone would be willing to adopt the language. Where would we be today had they not forged ahead? Although FORTRAN is out of favor with most Linux programmers, I submit that all subsequent imperative languages are descendants of, and owe a debt to, that first FORTRAN effort, which formalized named procedures and variables, and introduced the if keyword and the assignment statement (and what languages have no assignment statement?).

Now, having had some years of experience programming GPUs in various ways, we should look at whether we can use less-specific (and less-arcane), more general-purpose languages, open up GPU programming to a wider audience, and still achieve high performance.

Comparison to Vector Computers

Vector computing was first introduced in the 1960s, but was commercially successful only with the introduction of the Cray 1 in 1976. It may be difficult for less-experienced developers to believe, but the Cray 1's 80MHz clock (12.5ns) was the fastest on the planet, and it was the fastest scalar machine available at the time. In addition,



Figure 2. Cray 1 Supercomputer at the Computer History Museum. Photograph by Ed Toton, April 15, 2007

the Cray 1 had a vector instruction set that could deliver floating-point results six to ten times faster than the corresponding scalar code. Because the payoff was so high, programmers were quite willing to invest nontrivial effort to make sure their programs ran as fast as possible.

As with GPUs and accelerators today, the parallelism exploited on a Cray covered a wide range of applications, but it was not universal; it tended to self-select the applications that were ported to it. Moving a program to the Cray may have required changing an algorithm to one that was more amenable to vector processing, changing the data layout and, even with appropriate algorithms and data, may have required recoding the program to expose the parallelism. When it first was introduced, many users decried the effort they had to expend. In response, many optimized library kernels (such as the BLAS) were introduced, which could be written in assembly and presumably were highly optimized—programs that used these kernels for computationally intensive regions could expect good performance.

Some of the Cray 1's contemporaries and competitors introduced new languages or language extensions. The

Control Data Cyber 205 exposed its vector instruction set in its version of FORTRAN.

Writing $a(1:n)$ meant a vector starting at $a(1)$ with a length of n . Two vectors could be added, as $a(1:n) + b(2:n)$. Nontrivial operations were handled by a large set of Q8 intrinsics, which generated inline code. The expression $q8ssum(a(1:n))$ would generate the vector instruction to compute the single-precision summation of the vector $a(1:n)$. Such an approach made programming the Cyber 205 costly, and programs optimized for it were not portable.

The Texas Instruments' Advanced Scientific Computer (TI-ASC) came with an aggressive automatic vectorizing compiler, though the machine was not a commercial success. The Cray Fortran Translator (CFT) was the first widely used vectorizing compiler, and it became the dominant method by which the Cray vector instruction set was used. Many tuned libraries still were written in assembly, but the vectorizing compiler was good enough that most programmers could use it effectively.

Although CFT did an effective job of vectorization, the key to its success was something else. It was one of the first compilers that gave performance feedback to the user. The compiler produced a program listing that included a vectorization table. It would tell which loops would run in vector mode and which would not. Moreover, if a loop failed to vectorize, the compiler gave very specific information as to why not, down to which variable or array reference in which statement in the loop prevented vectorization. This had two effects; the first was that programmers knew what to change to make this loop run faster.

The second effect was that programmers became trained to write vector loops: take out procedure calls, don't use conditionals, use stride-1 array references and so on. The next program they wrote was more likely to vectorize from the start. After a few years, programmers were quite comfortable using the "vectorizable subset" of the language, and they could achieve predictable high performance. An important factor in this success was that the style of programming the compiler encouraged was portable, in that it gave predictably good performance.

across a wide range of vector computers, from Cray, IBM, NEC, Fujitsu, Convex and many others.

More recently, SSE registers with packed arithmetic instructions were added to the X86 architecture with the Pentium III in 1999. When first introduced, compilers were enhanced with a large set of intrinsics that operated on 128-bit wide data; `_mm_add_ss(m,n)` would do a packed four-wide single-precision floating-point add on two `_m128` operands, which presumably would be allocated to XMM registers.

PGI applied classical vectorization technology in its version 3.1 C and FORTRAN compilers to generate the packed arithmetic instructions automatically from loops. The other x86 compilers soon followed, and this is now the dominant method by which the SSE instructions on the x86 are used. Again, there are library routines written in assembly, but the compilers are good enough to be used effectively. As with the Cray compilers, performance feedback is crucial to effective use. If performance is critical, programmers will look at the compiler messages and rewrite loops that don't vectorize.

Programming GPUs—a Better Way

Current approaches to programming GPUs still are relatively immature. It's much better than it was a few years ago, when programmers had to cast their algorithms into

Accelerators are made programmable so as to be as flexible as possible.

OpenGL or something similar, but it's still unnecessarily difficult.

Programmers must manage (allocate and deallocate) the device, deal with the separate host and device memories, allocate and free device memory, and move data from and to the host, manage (upload) the kernel code, pack up the arguments, initiate the kernel, wait for completion and free the code space when done. All this is in addition to writing the kernel in the first place, exposing the parallelism, optimizing the data access patterns and a host of other machine-specific items, testing and tuning. A matrix multiplication takes a few lines in FORTRAN or C. Converting this to

Listing 1. Simplified Matrix Multiplication in CUDA, Using Tiled Algorithm

```

__global__ void
matmulKernel( float* C, float* A, float* B, int N2, int N3 ){
    int bx = blockIdx.x, by = blockIdx.y;
    int tx = threadIdx.x, ty = threadIdx.y;
    int aFirst = 16 * by * N2;
    int bFirst = 16 * bx;
    float Csub = 0;

    for( int j = 0; j < N2; j += 16 ) {
        __shared__ float Atile[16][16], Btile[16][16];
        Atile[ty][tx] = A[aFirst + j + N2 * ty + tx];
        Btile[ty][tx] = B[bFirst + j*N3 + b + N3 * ty + tx];

        __syncthreads();

        for( int k = 0; k < 16; ++k )
            Csub += Atile[ty][k] * Btile[k][tx];

        __syncthreads();
    }

    int c = N3 * 16 * by + 16 * bx;
    C[c + N3 * ty + tx] = Csub;
}

void
matmul( float* A, float* B, float* C,
        size_t N1, size_t N2, size_t N3 ){
    void *devA, *devB, *devC;
    cudaSetDevice(0);

    cudaMalloc( &devA, N1*N2*sizeof(float) );
    cudaMalloc( &devB, N2*N3*sizeof(float) );
    cudaMalloc( &devC, N1*N3*sizeof(float) );

    cudaMemcpy( devA, A, N1*N2*sizeof(float), cudaMemcpyHostToDevice );
    cudaMemcpy( devB, B, N2*N3*sizeof(float), cudaMemcpyHostToDevice );

    dim3 threads( 16, 16 );
    dim3 grid( N1 / threads.x, N3 / threads.y );

    matmulKernel<<< grid, threads >>>( devC, devA, devB, N2, N3 );

    cudaMemcpy( C, devC, N1*N3*sizeof(float), cudaMemcpyDeviceToHost );
    cudaFree( devA );
    cudaFree( devB );
    cudaFree( devC );
}

```

CUDA or Brook takes a page or more of code, even when making simplifying assumptions. One might question whether there is a better way.

Compilers are good at keeping track of details and

Listing 2. Simplified Matrix Multiplication in Brook

```
kernel void
matmulKernel( float N2, float A[][], float B[][],
               out float result<> ){
    float2 ik = indexof(result).xy;
    float4 ijk = float4( ik.x, 0.0f, 0.0f, ik.y );
    float4 jp1 = float4( 0.0f, 1.0f, 1.0f, 0.0f );

    float C = 0.0f;
    float n2 = N2;

    while( n2 > 0 ) {
        C += A[ijk.zw]*B[ijk.xy];
        ijk += jp1;
        n2 -= 1.0f;
    }
    result = C;
}

void
matmul( float* A, float* B, float* C,
        size_t N1, size_t N2, size_t N3 ){
    float Astream<N1, N2>;
    float Bstream<N2, N3>;
    float Cstream<N1, N3>;

    streamRead( Astream, A );
    streamRead( Bstream, B );
    matmulKernel( (float)N2, Astream, Bstream, Cstream );

    streamWrite( Cstream, C );
}
```

should be taken advantage of for that as much as possible. Is there anything specific to a GPU that makes it a more difficult compiler target than vector computers or attached processors, both of which had very successful, aggressive compilers? Could a compiler be created that would generate both host and GPU or accelerator code from a single source file, using standard C or FORTRAN, without language extensions?

I think it's feasible (though nontrivial), and a good idea. Here, I discuss what such a compiler might look like and what steps it would have to take. Two overriding goals are that the compiler operates just like a host compiler, except perhaps with a command-line flag to enable or disable the GPU code generator, and that no changes are needed to the other system tools (such as a linker and library archiver).

A significant difference between such a compiler and one for a vector computer has to do with the cost of failure. If a compiler fails to vectorize a specific loop, the performance cost can be a factor of five or ten, which is enough that a programmer will pay attention to messages

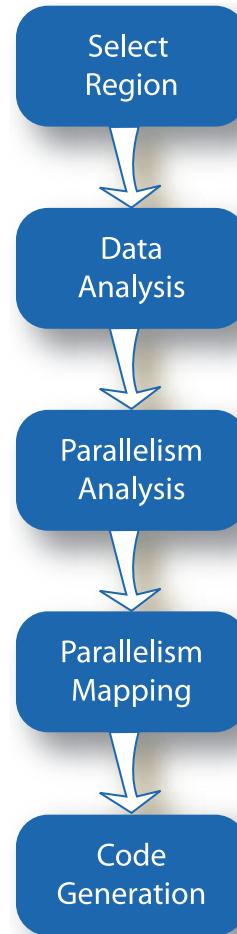


Figure 3. Five-Step Flowchart

from the compiler. If a compiler does a bad job of code generation for a GPU, the cost can be a slowdown (relative to host code) of a factor of ten or 100. This is enough that a fully hands-off, automatic approach just isn't feasible, at least not yet. At all steps, a programmer must be able to understand what the compiler has done and, if necessary, to override it.

The first, and perhaps most important step is to select what part or parts of the program should be converted to a kernel. Currently, that is done explicitly by a programmer who rips out that part of the program, replaces it with the code to manage the GPU, writes a kernel to execute on the GPU and combines it all into a single program.

Abstractly, we can use compute intensity to determine the parts of the program that are attractive for GPU acceleration. Compute intensity for a function, loop or block of code is the ratio of the number of operations to the amount of data that

needs to be moved. For GPU computing, we are most limited by the host-GPU bandwidth, so the critical ratio is the amount of data that needs to be moved from the host to GPU and back, divided into the number of operations that the GPU will execute. If the ratio is high enough, it's worth the cost of the data movement to get the high compute bandwidth of the GPU, assuming the computation has enough parallelism.

Although a compiler may be able to determine or estimate compute intensity, there are enough issues with GPU computing that it's better to leave this step to the programmer. Let's suppose a programmer can add a pragma or directive to the program, telling the compiler that a particular routine or loop or region of code should be compiled for the GPU.

The second step is data analysis on the region: what data needs to be allocated on the device memory and what needs to be copied from the host and back to the host afterward? This is within the scope of current compiler technology, though peculiar coding styles can defeat the analysis. In such cases, the compiler reports usage patterns with strange

boundary conditions; usually, it's easy to determine where this comes from and adjust the program to avoid it. In many cases, it arises from a potential bug lurking in the code, such as a hard-coded constant in one place instead of the symbolic value used everywhere else. Nonetheless, the compiler must have a mechanism to report the data analysis results, and the user must be able to override those results, in cases where the compiler is being too conservative (and moving too much data, for example).

Could a compiler be created that would generate both host and GPU or accelerator code from a single source file, using standard C or FORTRAN, without language extensions?

The third step is parallelism analysis on the loops in the region. The GPU's speed comes from structured parallelism, so parallelism must be rampant for the translation to succeed, whether translated automatically or manually. Traditional vectorizing and parallelizing compiler techniques are mature enough to apply here. Although vectorizing compilers were quite successful, both practically and commercially, automatic parallelization for multiprocessors has been less so. Much of that failure has been due to over-aggressive expectations. Compilers aren't magic; they can't find parallelism that isn't there and may not find parallelism that's been cleverly hidden or disguised by such tricks as pointer arithmetic.

Yet, parallelism analysis for GPUs has three advantages. First, the application domain is likely to be self-selected to include those with lots of rampant, structured parallelism. Second, structured parallelism is exactly the domain where

the classical compiler techniques apply. And finally, the payoff for success is high enough that even when automatic parallelization fails, if the compiler reports that failure specifically enough, the programmer can rewrite that part of the code to enable the compiler to proceed.

The fourth step is to map the program parallelism onto the machine. Today's GPUs have two or three levels of parallelism. For instance, the NVIDIA G80 architecture has multiprocessor (MIMD) parallelism across the 16 processors. It also has SIMD parallelism within each processor, and it uses another level of parallelism to enable multithreading within a processor to tolerate the long global memory latencies. The loop-level program parallelism must map onto the machine in such a way as to optimize, as much as possible, the performance features of the machine. On the NVIDIA, this means mapping a loop with stride-1 memory accesses to the SIMD-level parallelism and mapping a loop that requires synchronization to the multi-thread-level parallelism. This step is likely very specific to each GPU or accelerator.

The fifth step is to generate the GPU code. This is more difficult than code generation for a CPU only because the GPU is less general. Otherwise, this uses standard code-generation technology. A single GPU region may generate several GPU kernels to be invoked in order from the host. Some of the code-generation goals can be different from that of a CPU. For instance, a CPU has a fixed number of registers; compilers often will use an extra register if it allows them to schedule instructions more advantageously. A GPU has a large number of registers, but it has to share them among the simultaneously active threads. We want a lot of active threads, so when one thread is busy with a global memory access, the GPU has other work to keep it busy. Using extra registers may give a better schedule for each thread, but if it reduces the number of active threads, the total performance may suffer.

The final step is to replace the kernel region on the host with device and kernel management code. Most of this will

TECH TIP List Open Files

If you try to unmount a partition and get a message like this:

```
# umount /media/usbdisk/
umount: /media/usbdisk: device is busy
```

use the lsof command to find out what programs are using what files:

```
# lsof /media/usbdisk/
COMMAND PID USER FD TYPE DEVICE SIZE NODE NAME
bash   6925 root cwd DIR  8,17 4096    1 /media/usbdisk/
xmms  6979 root cwd DIR  8,17 4096    1 /media/usbdisk/
xmms  6979 root  8r REG  8,17 2713101 377 /media/usbdisk/a.mp3
```

This shows that the programs bash and xmms are using the device. For an even clearer picture, use the device name rather than the mountpoint:

```
# lsof /dev/sdb1
COMMAND PID USER FD TYPE DEVICE SIZE NODE NAME
bash   6925 root cwd DIR  8,17 4096    1 /media/usbdisk/
xmms  6979 root cwd DIR  8,17 4096    1 /media/usbdisk/
xmms  6979 root  8r REG  8,17 2713101 377 /media/usbdisk/a.mp3
```

You either can wait until those processes exit or terminate them manually.

—JAGADISH KAVUTURU

Listing 3. Fortran Matrix Multiplication Loop, Tagged to Be Compiled for the Accelerator

```
!$acc begin

do i = 1,n1
  do k = 1,n3
    c(i,k) = 0.0
    do j = 1,n2
      c(i,k) = c(i,k) + a(i,j) + b(j,k)
    enddo
  enddo
enddo

!$acc end
```

turn into library calls, allocating memory, moving data and invoking kernels.

These five steps are the same that a programmer has to perform when moving a program from a host to CUDA or Brook or other GPU-specific language. At least four of them can be mostly or fully automated, which would simplify programming greatly. Perhaps OpenCL, recently submitted by Apple to the Khronos Group for standardization, will address some of these issues.

There are some other issues that still have to be addressed. One is a policy issue. Can a user grab the GPU and hold onto it as a dedicated device? In many cases, there is only one user, so sharing the device is unimportant, but in a computing center, this issue will arise. Another issue has to do with the fixed size, nonvirtual GPU device memory. Whose job is it to split up the computation so it fits onto the GPU? A compiler can apply strip-mining to the loops in the GPU region, processing chunks of data at a time. The compiler also can use this strategy to overlap communication with computation by sending data for the next chunk while the GPU is processing the current chunk.

There are other issues that aren't addressed in this article, such as allocating data on the GPU and leaving it there for the life of a program, or managing multiple GPUs from a single host. These can all be solved in the same framework, all without requiring language extensions or wholesale program rewrites.

Bright Future

The future for GPU programming is getting brighter; these devices will become more convenient to program. There is no magic bullet; only appropriate algorithms written in a transparent style can be compiled for GPUs; users must understand and accept their advantages and limitations. These are not standard processor cores.

The industry can expect additional development of programmable accelerators, targeting different application markets. The cost of entering the accelerator market is much lower than for the CPU market, making a niche target market potentially attractive. The compiler method described here is robust enough to provide a consistent interface for a wide range of accelerators. ■

Michael Wolfe has been a compiler engineer at The Portland Group since joining in 1996, where his responsibilities and interests include deep compiler analysis and optimizations ranging from improving power consumption for embedded microcores to improving the efficiency of FORTRAN on parallel clusters. He has a PhD in Computer Science from the University of Illinois and authored *High Performance Compilers for Parallel Computing, Optimizing Supercompilers for Supercomputers* and many technical papers.

Lowest Prices on QUADCORE servers

Quad Core	Quad Core	2x Quad Core
Kentsfield Xeon 3200	Harpertown Xeon 5410	Harpertown Xeon 5410
\$100/mo	\$140/mo	\$180/mo

1 GB RAM
500/250 GB SATA 2
1300 GB/mo Included
100 Mbps Dedicated Port

intel® Xeon® inside™

carinet Better Servers. Better Service.

CARI.NET/LJ
888.221.5902

Use Python for Scientific Computing

As a general-purpose programming language, Python's benefits are well recognized. With the help of some add-on packages, you can use Python for scientific computing tasks as well. JOEY BERNARD

As computers become more and more powerful, scientific computing is becoming a more important part of fundamental research into how our world works. We can do more now than we could even imagine just a mere decade ago.

Most of this work has been done traditionally in more low-level languages, such as C or FORTRAN. Originally, this was done in order to maximize the efficiency of the code and to squeeze out every last bit of work from the computer. With computers now reaching multi-GHz speeds, this is no longer the bottleneck it once was. Other efficiencies come into play, with programmer efficiency being paramount. With this in mind, other languages are being considered that help make the most of a programmer's time and effort.

This article discusses one of these options: Python. Although Python is an interpreted language and suffers, unjustly, from the stigma that entails, it is growing in popularity among scientists for its clarity of style and the availability of many useful packages. The packages I look at in this article specifically are designed to provide fast, robust mathematical and scientific tools that can run nearly as fast as C or FORTRAN code.

Getting Set Up

The packages I focus on here are called numpy and scipy. They are both available from the main SciPy site (see Resources). But before we download them, what exactly are numpy and scipy?

numpy is a Python package that provides extended math capabilities. These include new data types, such as long integers of unlimited size and complex numbers. It also provides a new array data type that allows for the construction of vectors and matrices. All the basic operations that can be applied to these new data types also are included. With this we can get quite a bit of scientific work done already.

scipy is a further extension built on top of numpy. This package simplifies a lot of the more-common tasks that need to be handled, including tools such as those used to find the roots of polynomials, doing Fourier transformations, doing numerical integrals and enhanced I/O. With these functions, a user can develop very sophisticated scientific applications in relatively short order.

Now that we know what numpy and scipy are, how do we get them and start using them? Most distributions include both of these packages, making this the easy way to install

them. Simply use your distribution's package manager to do the install. For example, in Ubuntu, you would type the following in a terminal window:

```
sudo apt-get install python-scipy
```

This installs scipy and all of its dependencies.

If you want to use the latest-and-greatest version and don't want to wait for your distribution to get updated, they are available through Subversion. Simply execute the following:

```
svn co http://svn.scipy.org/svn/numpy/trunk numpy svn co http://svn.scipy.org/svn/scipy/trunk scipy
```

Building and installing is handled by a setup.py script in the source directory. For most people, building and installing simply requires:

```
python setup.py build  
python setup.py install # done as root
```

If you don't have root access, or don't want to install into the system package directory, you can install into a different directory using:

```
python setup.py install --prefix=/path/to/install/dir
```

Other options also are available, which you can find out about by using:

```
python setup.py --help-commands
```

Take time to experiment and see whether you can use any of the extra options in your specific case.

Basic Math

Now that we have scipy and numpy installed, let's begin our tour by looking at some of the basic functions that are often used in scientific calculations. One of the most common tasks is matrix mathematics. This is greatly simplified when you use numpy. The most basic code to do a multiplication of two matrices using numpy would look like this:

```

import numpy
a1=numpy.empty((500,500))
a2=numpy.empty((500,500))
a3=a1*a2

```

Contrast this to what we would need to write if we did it in C:

```

#include <stdlib.h>
int main() {
    double a1[500][500];
    double a2[500][500];
    double a3[500][500];
    int i, j, k;
    for (i=0; i<500; i++) {
        for (j=0; j<500; j++) {
            a3[i][j] = 0;
            for (k=0; k<500; k++) {
                a3[i][j] += a1[i][k] * a2[k][j];
            }
        }
    }
}

```

Table 1. Average Runtimes

Language	Average Time (seconds)
C	1.620
C (-O3)	0.010
Python	0.250

The Python code is much shorter and cleaner, and the intent of the code is much clearer. This kind of clarity in the code means that the programmer can focus much more on the algorithm rather than the gritty details of the implementation. There are C libraries, such as LAPACK, which help simplify this work in C. But, even these libraries can't match the simplicity of scipy.

"But what about efficiency?", I hear you ask. Well, let's take a look at it with some timed runs. Taking our above example, we can put some calls around the actual matrix multiplication part and see how long each one takes. See Table 1

**ROBO
DEVELOPMENT
CONFERENCE & EXPO '08**

NOV. 18-19, 2008
SANTA CLARA CONVENTION CENTER
SANTA CLARA, CA



For Information on
**Sponsorship and Exhibiting
Opportunities**, contact
Ellen Cotton at ecotton@ehpub.com or 508-663-1500 x240

**Join the International Technical Design and Development Event
for the Personal, Service & Mobile Robotics Industry**

**EXCLUSIVE OFFER:
USE PRIORITY CODE RDLXJ AND
SAVE \$300 ON A CONFERENCE PASS
WWW.ROBODEVELOPMENT.COM ■ 800-305-0634**

■ The industry's most comprehensive conference program covering these critical topics:

- Systems & Systems Engineering
- Tools & Platforms
- Enabling Technology
- Achieving Autonomy
- Design & Development

■ Valuable networking opportunities that put you in touch with peers, industry experts and up-and-coming talent:

- Evening Welcome Reception
- Speaker Meet & Greet
- Birds-of-a-Feather Discussions

■ Exposition floor featuring what's new and what's next in robotics design and development

Sebastian Thrun, Winner of the DARPA Grand Challenge; Director, Artificial Intelligence Laboratory, Stanford University

Maja J. Matarić, Founding Director, USC Center for Robotics and Embedded Systems; Director, USC Robotics Research Lab

Jeanne Dietrich, CEO, MobileRobots Inc

Michael Bruch, Section Head, Space and Naval Warfare (SPAWAR) Systems Center

Presented By:




FOR COMPLETE EVENT DETAILS VISIT WWW.ROBODEVELOPMENT.COM

Founding Sponsor

Microsoft®

Gold Sponsor

MOBILEROBOTS INC

Silver Sponsors

ADAM ROBOTICS

ADVANCED MOTION CONTROLS

MOTION BOSTON ENGINEERING

ENERGID NATIONAL INSTRUMENTS

pico Technology

Velodyne WAVE GARAGE

Corporate Sponsors

FESTO HOKUYO

JETTA COMPANY LIMITED

PNI SENTEK solutions

OceanServer TEXASHOST, INC Unigen

Analyst, Association & Academic Co-Sponsors

Carnegie Mellon SILICON VALLEY

SOVESTOR VC - Private Equity ROUND TABLE Best Connect Show

Media Co-Sponsors

CIRCUIT CELLAR Dr.Dobb's Entertainment Engineering

HINUX NUTS & VOLTS SERVO

Listing as of September 8, 2008. For a current list of participating companies, please visit www.robodevelopment.com.

for the results.

Although your mileage will vary, because these times depend on your hardware and what other programs also are running on your machine, we can see a general trend. The Python code actually was about eight times faster than the C code compiled with no command-line options. That is actually quite surprising. Once we use the optimization command-line option, we see that the C code is now faster, by a factor of approximately 25. So, we can get faster code using optimized C, but we need to realize that multiplying two matrices with 250,000 elements each in one-quarter of a second is probably fast enough.

As well, we get a certain amount of protection when we use Python. What happens if we try to multiply two matrices where such multiplication doesn't make sense mathematically? When we try to multiply two matrices of different sizes, Python gives us:

```
ValueError: shape mismatch: objects cannot be
broadcast to a single shape
```

In C, we get no error at all. This due to the fact that when we work with matrices, we actually are using pointer arithmetic. So pretty much anything we do is valid C, even if it makes no sense in the problem domain.

We also can work just as easily with complex numbers. If we wanted to create an array of 64-bit complex numbers, we would write:

```
a=zeros((500,500), dtype=complex64)
```

which would give us a 500x500 element matrix initialized with zeros. We access the real and imaginary parts of each element using:

```
a.real[0,0]=1.0 a.imag[0,0]=2.0
```

which would set the value $1+2j$ into the [0,0] element.

There also are functions to give us more complicated results. These include dot products, inner products, outer products, inverses, transposes, traces and so forth. Needless to say, we have a great deal of tools at our disposal to do a fair amount of science already. But is that enough? Of course not.

Getting Down to "Real" Science

Now that we can do some math, how do we get some "real" science done? This is where we start using the features of our second package of interest, `scipy`. With this package, we have quite a few more functions available to do some fairly sophisticated computational science. Let's look at an example of simple data analysis to show what kind of work is possible.

Let's assume you've collected some data and want to see what form this data has, whether there is any periodicity. The following code lets us do that:

```
import scipy
inFile = file('input.txt', r)
```

```
inArray = scipy.io.read_array(inFile)
outArray = fft(inArray)
outFile = file('output.txt', w)
scipy.io.write_array(outFile, outArray)
```

As you can see, reading in the data is a one-liner. In this example, we use the FFT functions to convert the signal to the frequency domain. This lets us see the spread of frequencies in the data. The equivalent C or FORTRAN code is simply too large to include here.

But, what if we want to look at this data to see whether there is anything interesting? Luckily, there is another package, called `matplotlib`, which can be used to generate graphics for this very purpose. If we generate a sine wave and pass it through an FFT, we can see what form this data has by graphing it (Figures 1 and 2).

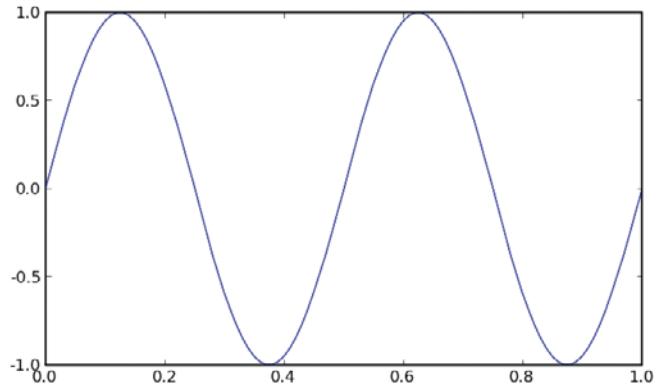


Figure 1. Sine Wave

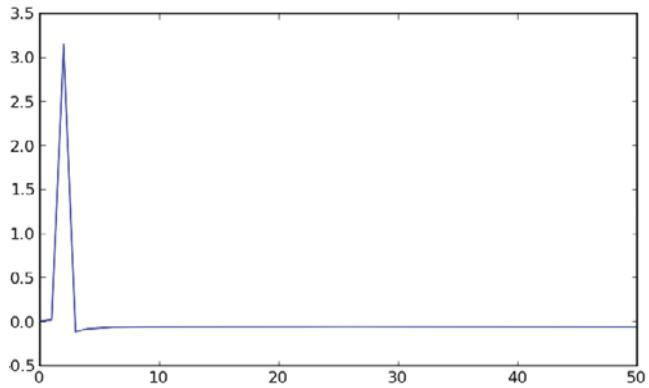


Figure 2. FFT of Sine Wave

We see that the sine wave looks regular, and the FFT confirms this by having a single peak at the frequency of the sine wave. We just did some basic data analysis.

This shows us how easy it is to do fairly sophisticated scientific programming. And, if we use an interactive Python environment, we can do this kind of scientific analysis in an exploratory way, allowing us to experiment on our data in near real time.

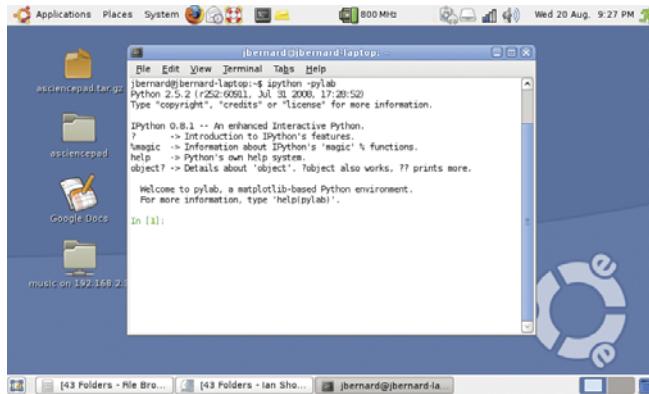


Figure 3. iPython Window

Luckily for us, the people at the SciPy Project have thought of this and have given us the program ipython. This also is available at the main SciPy site. ipython has been written to work with scipy, numpy and matplotlib in a very seamless way. To execute it with matplotlib support, type:

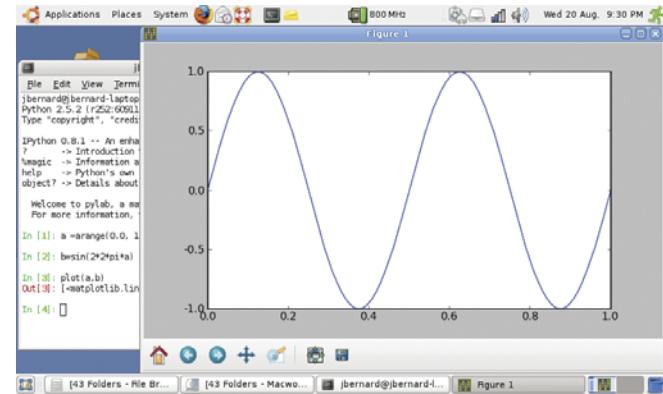


Figure 4. iPython Plotting

`ipython -pylab`

The interface is a simple ASCII one, as shown in Figure 3. If we use it to plot the sine wave from above, it simply pops up a display window to draw in the plot (Figure 4).

The plot window allows you to save your brilliant graphs

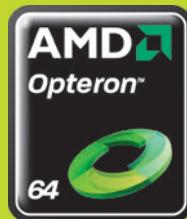
Expert included.

As a Production Manager at Silicon Mechanics, Jason stays up to date with continual refinements in server technology. He's proud that the new Silicon Mechanics Rackform nServ A421 has raised the bar for density in high-performance hardware, and he's sure our customers will be happy about that, too.

The A421 is an enterprise-level server solution well suited to demanding, mission-critical deployments. It is equipped with 4 Quad-Core or Dual-Core AMD Opteron™ processors, up to 128 GB DDR2 memory, 6 hot-swap SAS / SATA drives, integrated SAS controller, 4 PCI expansion slots, and redundant power supply with 88% maximum efficiency. It makes an excellent choice for virtualized environments.



When you partner with Silicon Mechanics, you get more than efficient, enterprise-grade compute hardware—you get an expert like Jason.



For more information about the Rackform nServ A421
visit www.siliconmechanics.com/A421.



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



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

release date: february 2008



www.LinuxJournal.com/ArchiveCD

The 1994–2007 Archive CD,
back issues, and more!

and plots, so you can show the entire world your scientific breakthrough. All of the plots for this article actually were generated this way.

So, we've started to do some real computational science and some basic data analysis. What do we do next? Why, we go bigger, of course.

Going Parallel

So far, we have looked at relatively small data sets and relatively straightforward computations. But, what if we have really large amounts of data, or we have a much more complex analysis we would like to run? We can take advantage of parallelism and run our code on a high-performance computing cluster.

The good people at the SciPy site have written another module called mpi4py. This module provides a Python implementation of the MPI standard. With it, we can write message-passing programs. It does require some work to install, however.

The first step is to install an MPI implementation for your machine (such as MPICH, OpenMPI or LAM). Most distributions have packages for MPI, so that's the easiest way to install it. Then, you can build and install mpi4py the usual way with

ScientificPython

numpy and scipy are not the only options available to Python programmers. Another popular package is ScientificPython. It includes geometric types (such as vectors, tensors and quaternions), polynomials, basic statistics, derivatives, interpolation and more. This is the same type of functionality available in scipy. The major difference is that ScientificPython has the ability to do parallel programming built in, whereas scipy requires an extra module. This is done with a partial implementation of MPI and an implementation of the Bulk Synchronous Parallel library (BSplib).

the following:

```
python setup.py build python setup.py install
```

To test it, execute:

```
mpirun -np 5 python tests/helloworld.py
```

1

2

3

4

5

6

HPC Your Way

Intel or AMD. Ethernet or InfiniBand. Linux or Microsoft Windows HPC Server. Now you can have a uniform set of HPC compilers and tools across all of your x64 clusters. PGI CDK compilers and tools are available directly from most cluster suppliers. Take a free test drive today at www.pgroup.com/reasons

PGI CDK® Cluster Development Kit®

LAPACK and BLAS

The argument can be made that comparing the complexity of C and FORTRAN to that of Python is unfair, because we actually are using add-on packages in Python. Equivalent libraries can be used in C and FORTRAN, with LAPACK and BLAS being some of the more popular. BLAS provides basic linear algebra functions, while LAPACK builds on these to provide more complex scientific functions. Although these libraries provide optimized routines that will extract every useful cycle from your hardware and are much simpler to write than straight C or FORTRAN, they still are orders of magnitude more complex than the equivalent in Python. If you really do need to squeeze out every last tick from your machine, however, nothing will beat these types of libraries.

which will run a five-process Python and run the test script.

Now, we can write our program to divide our large data set among the available processors if that is our bottleneck. Or, if we want to do a large simulation, we can divide the

Types of Parallel Programming

Parallel programs can, in general, be broken down into two broad categories: shared memory and message passing. In shared-memory parallel programming, the code runs on one physical machine and uses multiple processors. Examples of this type of parallel programming include POSIX threads and OpenMP. This type of parallel code is restricted to the size of the machine that you can build.

To bypass this restriction, you can use message-passing parallel code. In this form, independent execution units communicate by passing messages back and forth. This means they can be on separate machines, as long as they have some means of communication. Examples of this type of parallel programming include MPICH and OpenMPI. Most scientific applications use message passing to achieve parallelism.

simulation space among all the available processors.

Unfortunately, a useful discussion of MPI programming would be another article or two on its own. But, I encourage you to get a good textbook on MPI and do some experimenting yourself.

Conclusion

Although any interpreted language will have a hard time matching the speed of a compiled, optimized language, we have seen that this is not as big a deterrent as it once was. Modern machines run fast enough to more than make up for the overhead of interpretation. This opens up the world of complex applications to using languages like Python.

This article has been able to introduce only the most basic available features. Fortunately, many very good tutorials have been written and are available from the main SciPy site. So, go out and do more science, the Python way.■

Joey Bernard has a background in both physics and computer science. Finally, his latest job with ACENet has given him the opportunity to use both degrees at the same time, helping researchers do HPC work.

Resources

Python Programming Language—Official Web Site:
www.python.org

SciPy: www.scipy.org

ScientificPython—Theoretical Biophysics, Molecular Simulation, and Numerically Intensive Computation:
dirac.cnrs-orleans.fr/plone/software/scientificpython

Statement of Ownership, Management, and Circulation		
1. Publication Title: <i>Linux Journal</i>	Business Office of Publisher:	10. Owner(s):
2. Publication Number: 1075-3583	PO Box 980985 Houston, TX 77098	Carlie Fairchild PO Box 980985 Houston, TX 77098
3. Filing Date: September 5, 2008	9. Full Names and Complete Addresses of Publisher, Editor, and Managing Editor:	Joyce Searls PO Box 980985 Houston, TX 77098
4. Issue Frequency: Monthly	Publisher: Carlie Fairchild PO Box 980985 Houston, TX 77098	Adele Soffa PO Box 980985 Houston, TX 77098
5. Number of Issues Published Annually: 12	Editor: Doc Searls PO Box 980985 Houston, TX 77098	11. Known Bondholders, Mortagees, and Other Security Holders Owning or Holding 1 Percent or More of Total Amount of Bonds, Mortages, or Other Securities: None
6. Annual Subscription Price: \$29.50	Managing Editor: Jill Franklin PO Box 980985 Houston, TX 77098	12. Tax Status: Has not Changed During Preceding 12 Months
7. Complete Mailing Address of Known Office of Publication: PO Box 980985 Houston, TX 77098	Contact Person: Mark Irgang 713-344-1956	
8. Complete Mailing Address of Headquarters of General		
13. Publication Title: <i>Linux Journal</i>	14. Issue Date: October 2008	
15. Extent and Nature of Circulation	Average No. Copies Each Issue During Preceding 12 Months	No. Copies of Single Issue Published Nearest to Filing Date
a. Total Number of Copies: (Net press run)	53,925	51,659
b. Paid and/or Requested Circulation		
(1) Paid/Requested Outside-County Mail Subscriptions on Form 3541.	18,971	18,089
(2) Paid In-County Subscriptions Stated on Form 3541	0	0
(3) Sales Through Dealers and Carriers, Street Vendors, Counter Sales, and Other Non-USPS Paid Distribution	21,447	20,319
c. Total Paid and/or Requested Circulation	40,418	38,408
d. Free Distribution Outside the Mail		
(1) Outside-County as Stated on Form 3541	747	693
(2) In-County as Stated on Form 3541	0	0
(3) Other Classes Mailed Through the USPS	0	0
e. Free Distribution Outside the Mail	4,600	4,820
f. Total Free Distribution	5,347	5,513
g. Total Distribution	45,765	43,921
h. Copies Not Distributed	8,160	7,738
i. Total	53,925	51,659
j. Percent Paid and/or Requested Circulation	88%	87%
PS Form 3526		

Do you take
"the computer doesn't do that"
as a personal challenge?

So do we.

LINUX
JOURNAL™

Since 1994: The Original Monthly Magazine of the Linux Community

Subscribe today at www.linuxjournal.com

Lincoln and Whitman's Unfinished Business

We've had democracy for a long time. Now we finally can make it work.

DOC SEARLS

During the '04 campaign, Phil Windley—then CIO for the state of Utah—said something profound about democracy, "Most people just want the roads fixed".

He went on to explain that there are two sides of democracy, and they aren't defined by parties. One side is elections. The other side is governance. Elections hog the spotlight, because they make good stories. Governance is how the work gets done. To Phil, governance is the real frontier, the side of democracy that has not met its promise.

Phil isn't alone. In the closing sentence of his Gettysburg Address, Abraham Lincoln said the battle's fallen soldiers had fought so "...that this nation...shall have a new birth of freedom", with "government of the people, by the people, for the people". Lincoln called this "unfinished work".

More than 145 years later, it's still unfinished, and will remain so until most ordinary citizens feel free to engage their governments directly and personally. Until we do that, we'll be stuck with too much government of the people, and not enough by or for the people.

Means for citizen involvement have always been there, but the threshold of engagement has always seemed too high. If you wanted action, you had to be outraged, obsessed or connected by money and other currencies of favor.

But then the Net came along. It didn't change everything overnight, but it provided a whole new environment where the rules of engagement were far different, and where privileges of the few could spread to the many.

Nobody knows or explains this better than Steve Urquhart, who represents the citizens of St. George in the Utah state legislature. The first time I talked to Steve, he told me money in politics was a red herring. What really matters, he said, are connections. Relationships. Money is often involved, but far from always. The real challenge for democracy, he said, is to get more citizens involved directly at every level. This is why Steve blogs. It's a means of engagement.

On July 22, 2008, Steve ran a post titled "Transparency and Performance". It serves as an exceptionally detailed frame among many moving pictures of changes in democracy enabled by the Net. Here it is:

When I entered the Legislature eight years ago, information was hidden from voters....

Fortunately, we quickly changed our tune, and offered the public access to lots of information. Everything I have available—in terms of access to bills, voting records, floor and committee speeches—the public also has....

Giving people direct access to information has three consequences. One, officials pay closer attention to their actions. Two, people can more readily hold officials accountable for their actions. And, three, officials can be bolder in their actions. The first two points are obvious. I'll discuss the third point.

Transparency allows officials to be bolder, because they have a closer, more-informed relationship with constituents; and, in any event, there is no place to hide. Where good information is lacking, drivel flows back and forth between constituents and their elected officials; officials can hide behind aphorisms, and the public has a difficult time digging them out with a rehash of the week's editorials. Available information, however, allows people to dig in more on their elected officials; likewise, it allows officials to cite specifics in the record, to explain actions that contradict the wishes of the editorialists.

I believe we'll see a shift in voters' concerns, at least to some degree, away from the litmus of liberal or conservative toward a litmus of "conversative" or "non-conversative", with the conversative candidates having the advantage....And, I believe that this openness signifies great things for representative democracy.

Twelve comments follow. One is by somebody named Do What Is Right. It reads: "Hey Steven, if you really wanted transparency, why do the great

GOP go behind closed doors in the capitol during the legislative sessions? You guys are such hypocrites. Explain that one Steven."

Steve replies:

Case in point on transparency. You are repeating the media's mantra. Tell me how many times the House Republican caucus went behind closed doors last session. Answer: zero. Yet the media keeps on printing it, so it must be true.

And, that's why it's still early in the story of How Democracy Works in the networked age. We not only need to be weaned from media, but from the need for mediation. In *Leaves of Grass*, Walt Whitman wrote this brief against mediation:

You shall no longer take things at second
or third hand...
nor look through the eyes of the dead,
nor feed on the spectres in books.
You shall not look through my eyes either,
nor take things from me.
You shall listen to all sides and filter them
for yourself.

Whitman also called himself a "poet of democracy" and wrote much about the subject. Here's one sample, from *Specimen Days & Collect*:

Did you, too, O friend, suppose democracy was only for elections, for politics, and for a party name? I say democracy is only of use there that it may pass on and come to its flower and fruit in manners, in the highest forms of interaction between men, and their beliefs....I submit, therefore, that the fruition of democracy, on aught like a grand scale, resides altogether in the future.

That future is here. The time has come to complete Lincoln and Whitman's unfinished business. ■



Doc Searls is Senior Editor of *Linux Journal* and a fellow with both Berkman Center for Internet and Society at Harvard University and the Center for Information Technology and Society at the University of California, Santa Barbara.

THE SIMPLE GUIDE TO HOSTING:

IF IT NEEDS TO BE ONLINE AND STAY ONLINE IT NEEDS TO BE HOSTED AT RACKSPACE®

You need the world's leader in hosting.

- The Fanatical Support Promise™
- Industry Leading Service Level Agreements and Guarantees
- Dedicated Support Team Assigned to Every Customer
- 24x7x365 Live Support – No Call Centers
- Proactive Sales and Support Consultation
- Flexible Hosting Solutions and Services
- 100% Network Uptime Guarantee
- Strategic Technology and Business Partner Relationships

TOLL FREE: 1-888-571-8976
www.rackspace.com/linuxjournal

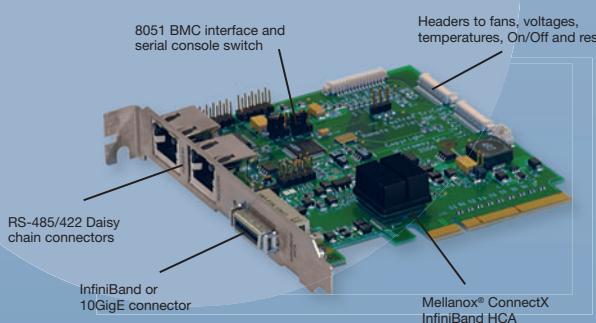
experience **fanatical support**®



Your Applications Will Run Faster With Next Generation Microway Solutions!

TriCom™ X

- QDR/DDR InfiniBand HCA
- ConnectX™ Technology
- 1μsec Latency
- Switchless Serial Console
- NodeWatch™ Remote Management



Teraflop GPU Computing

For Workstations and HPC Clusters

- NVIDIA® Tesla™ GPU with 240 Cores on One Chip
 - ▶ CUDA™ SDK
- NVIDIA® Quadro® Professional Graphics
- AMD® FireStream™ GPU
 - ▶ Stream SDK with Brook+



NumberSmasher®

Large Memory Scalable SMP Server



- Scales to 1 TB of Virtual Shared Memory
- Up to 128 CPU Cores
- 8U System Includes 32 Quad Core CPUs
- QDR 1 μsec Backplane



FasTree™ X

- Mellanox® InfiniScale™ IV Technology
- QDR/DDR InfiniBand Switches
- Modular Design
- 4 GB/sec Bandwidth per Port
- QSFP Interconnects
- InfiniScope™ Real Time Diagnostics

Call the HPC Experts at Microway to Design Your Next High-Reliability Linux Cluster or InfiniBand Fabric.

508-746-7341

Sign up for Microway's Newsletter at
www.microway.com

Microway
Technology you can count on™