

BEOWULF | CONDOR | HEARTBEAT | PARALLEL PROGRAMMING | AJAX

LINUX JOURNAL

Since 1994: The Original Magazine of the Linux Com-

SEPTEMBER 2006 | ISSUE 149 | www.linuxjournal.com

Clustering Is Not Rocket Science

JavaScript
for the Win,
Love It or Hate It



CrossOver
Office 5.0
MAKES THE
GRADE

Doc interviews
J.P. Rangaswami
of Dresdner Kleinwort Wasserstein
(DrKW)

PLUS:

**MySQL Makes
Web-Based
Reporting Easy**

**Savage 2
Game on Its
Way to Linux**

**Marcel Gagné
Serves Up
Tips for
your Body**

AN
SSC
PUBLICATION

USA \$5.00
CAN \$6.50





High-Performance and Enterprise Computing Under Your Control

Industry Leading 2P and 4P x86 Computing

Innovative server technology with outstanding performance and unrivaled memory scalability to accelerate compute and memory intensive applications.

2-way **XtremeServer™** up to **64GB** of memory



4-way **XtremeServer™** up to **128GB** of memory



4-way **XtremeWorkstation™** up to **128GB** of memory



1U **XtremeServer™**

- Dual-Core AMD Opteron™ processors
- Up to 64GB of DDR2 533/667 memory
- Up to 1.0TB SATA or 292GB SAS
- 1 PCI-X and 1 PCI-Express x16
- Dual-port Gigabit NICs
- Hot-swappable drives
- ServerDome Management – IPMI 2.0
- Windows® or Linux OS

3U **XtremeServer™**

- Dual-Core AMD Opteron™ processors
- Up to 128GB of DDR2 533/667 memory
- Up to 3.0TB SATA or 876GB SAS
- 3 PCI-X and 2 PCI Express x16
- Dual-port Gigabit NICs
- Redundant power supplies and fans
- Hot-swappable drives
- ServerDome Management – IPMI 2.0
- Windows® or Linux OS

XtremeWorkstation™

- Dual-Core AMD Opteron™ processors
- Up to 128GB of DDR2 533/667 memory
- Up to 2.0TB SATA or 584GB SAS
- 3 PCI-X and 2 PCI Express x16
- Two NVIDIA 4500/5500 graphics cards
- Redundant power supplies and fans
- Hot-swappable drives
- Windows® or Linux OS



Get the facts...

Performance Analysis and Value Proposition White Paper
Go to <http://www.appro.com>

AMD Opteron™ Processors: - AMD 64 Dual-Core Technology improve system throughput for faster networking connectivity
- Best performance per-watt, helping to reduce electricity costs while maximizing IT budget dollars

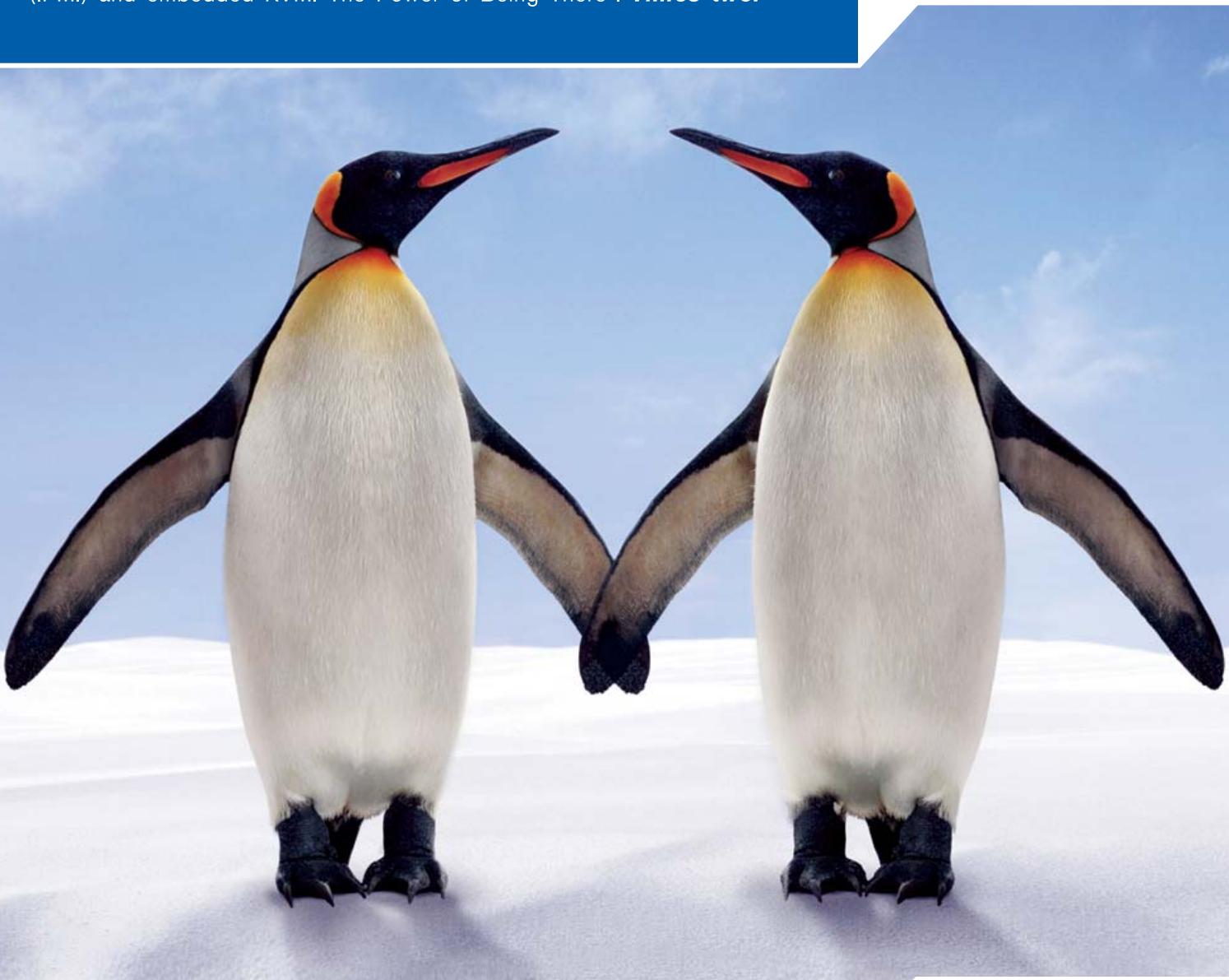
The Power of Being There.[®]

Times two.

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



An Avocent[®] Company

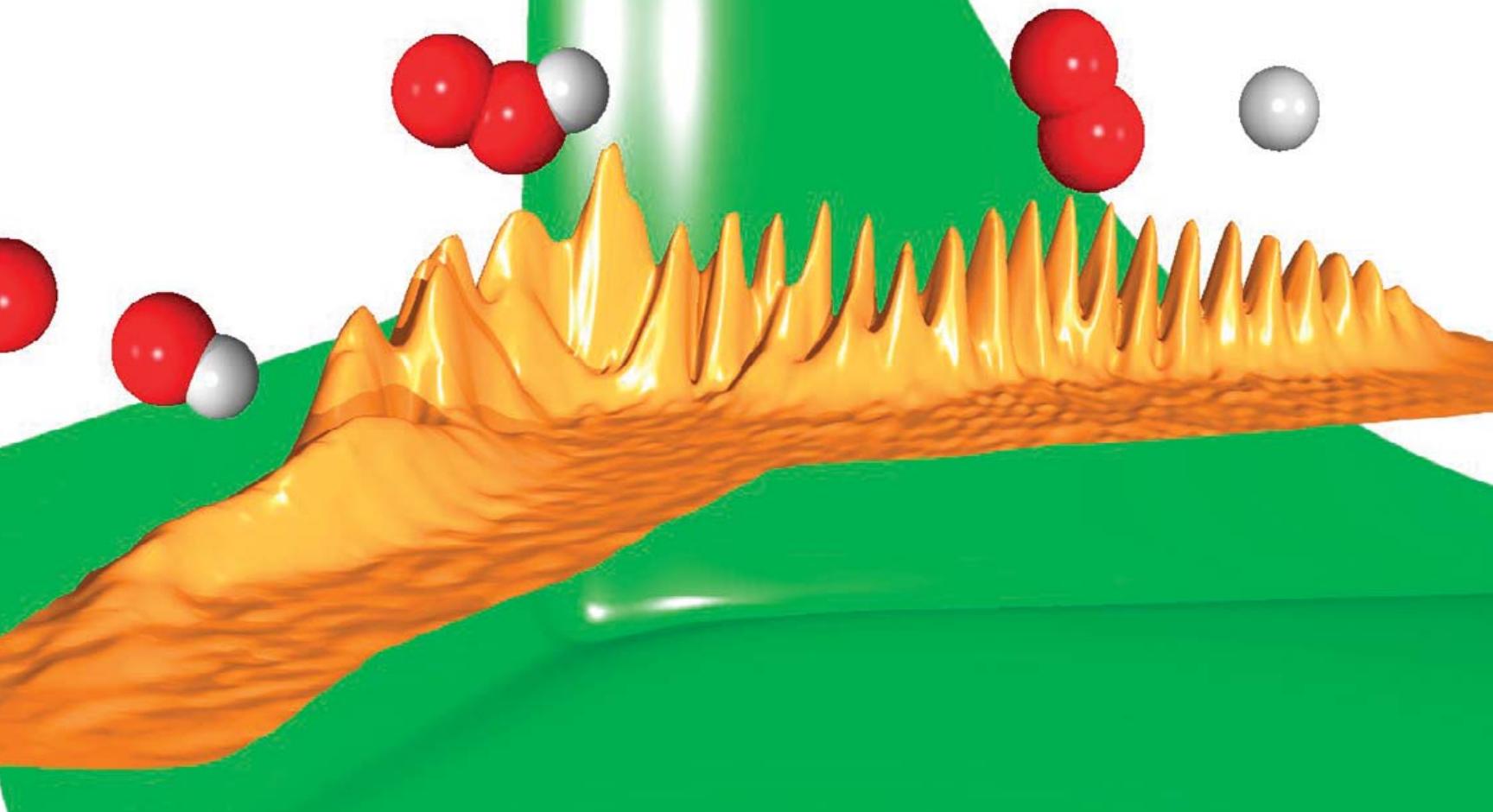


Visit www.avocent.com/powerx2

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

CONTENTS

SEPTEMBER 2006
Issue 149



FEATURES

56 CLUSTERING IS NOT ROCKET SCIENCE

Want to compute rocket science without having to be a rocket scientist?

Rowan Gollan, Andrew Denman and Marlies Hankel

62 GETTING STARTED WITH CONDOR

Computers of different feathers can still flock together with Condor.

Irfan Habib

66 DRBD IN A HEARTBEAT

Build a good redundant system to prevent downtime.

Pedro Pla

72 MAINSTREAM PARALLEL PROGRAMMING

Get a Beowulf cluster running without having to fight off Grendel.

Michael-Jon Ainsley Hore

ON THE COVER

- Clustering Is Not Rocket Science, p. 56
- JavaScript for the Win, Love It or Hate It, p. 22
- CrossOver Office 5.0, p. 48
- Doc Interviews J.P. Rangaswami, p. 44
- MySQL Makes Web-Based Reporting Easy, p. 86
- Savage 2 Game on Its Way to Linux, p. 40
- Marcel Gagné Serves Up Tips for your Body, p. 28

COVER PHOTO BY DRESDNER KLEINWORT

The competition doesn't stand a chance.



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

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

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

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



CONTENTS

SEPTEMBER 2006

Issue 149

COLUMNS

22 REUVEN M. LERNER'S
AT THE FORGE
JavaScript

28 MARCEL GAGNÉ'S
COOKING WITH LINUX
Operating Your Body at Peak Performance

34 DAVE TAYLOR'S
WORK THE SHELL
When Is "Good Enough" Good Enough?

36 MICK BAUER'S
PARANOID PENGUIN
How to Worry about Linux Security

40 DEE-ANN LEBLANC'S
GET YOUR GAME ON
S2 Games

42 JON "MADDOG" HALL'S
BEACHHEAD
Pirates and Pollywogs

44 DOC SEARLS'
LINUX FOR SUITS
An Interview with J.P. Rangaswami



96 NICHOLAS PETRELEY'S
/VAR/OPINION
Parallel Is Coming into Its Own

IN EVERY ISSUE

- 12** LETTERS
- 16** UPFRONT
- 46** NEW PRODUCTS
- 81** ADVERTISERS INDEX

INDEPTH

78 MILLE-XTERM AND LTSP
If you thought Network Computing was dead, wait until you read this.
Francis Giraldeau, Jean-Michel Dault and Benoit des Ligneris

82 64-BIT JMP FOR LINUX
64-Bit Linux and JMP statistics software are made for each other.
Erin Vang

86 WEB REPORTING WITH
MYSQL, CSS AND PERL
Some nifty features in MySQL make Web reporting a breeze.

Paul Barry

92 ECRASH: DEBUGGING
WITHOUT CORE DUMPS
You don't have to take a core dump to debug your programs.

David Frascone

REVIEWS

48 CROSSOVER OFFICE 5.0
Jes Hall

52 PATHSCALE INFINIPATH
INTERCONNECT
Logan G. Harbaugh



48 CROSSOVER OFFICE 5.0

Next Month

DIGITAL LIFESTYLE

Next month, we'll tell you how one person used MisterHouse to set up a zoned home heating system, share a fascinating experience with do-it-yourself robotics, relate how F-Spot saved the day for photo management, and detail how Mike Diehl used Linux for everything from MythTV to VoIP. We'll open up Jon "maddog" Hall's backpack and have a look at all his gadgets too. And, Marcel Gagné will explain how to tell your chat buddies what music you're listening to and how to share it with them.

That's not all. We have an exclusive sample chapter from the Apress book *Beginning Ubuntu*. Want to learn how to customize your Ubuntu desktop? It's all right here.

USPS LINUX JOURNAL (ISSN 1075-3583) is published monthly by SSC Media Corp., 2825 NW Market Street #208, Seattle, WA 98107. Periodicals postage paid at Seattle, Washington and at additional mailing offices. Cover price is \$5 US. Subscription rate is \$25/year in the United States, \$32 in Canada and Mexico, \$62 elsewhere. POSTMASTER: Please send address changes to *Linux Journal*, PO Box 980985, Houston, TX 77253-3587. Subscriptions start with the next issue.

TYAN



Power of 4 + 4 TYAN Matchless 8-Processor Server

**M4881**

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

**Thunder S4881**

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

**Transport VX50 B4881****8-Processor AMD Opteron™ 5U Server**

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

TYAN COMPUTER CORP.

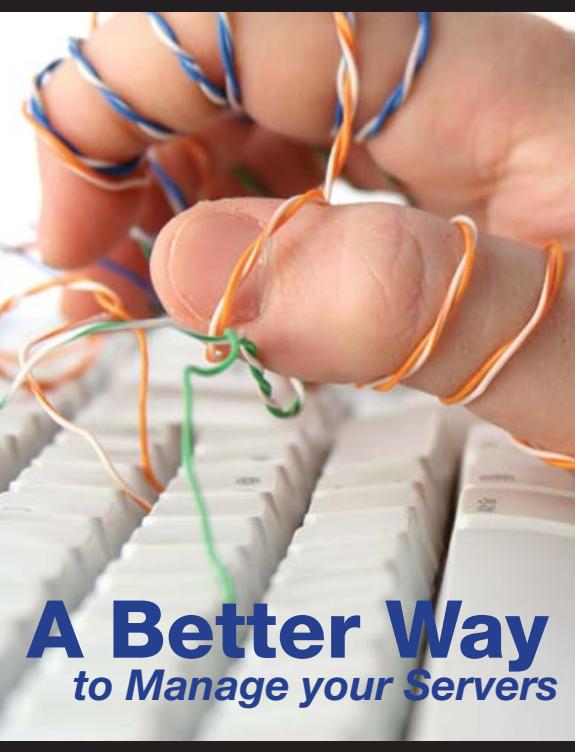
Tyan Computer USA

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

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

www.tyan.com

HOT WIRED FOR SERVICE



A Better Way to Manage your Servers

**Virtual Private Servers
Starting at \$19.95/mo.**

- No Setup Fees, No Contracts
- Free http and icmp monitoring
- Free incident response service
- Zervex ServerCP Online Control Panel Included



Real People Real Support™
-----Since 1998-----

www.jtl.net/lj
1-877-765-2300

LINUX JOURNAL

Editor in Chief

Nick Petreley, ljeditor@ssc.com

Executive Editor

Jill Franklin

jill@ssc.com

Senior Editor

Doc Searls

doc@ssc.com

Art Director

Garrick Antikajian

garrick@ssc.com

Products Editor

James Gray

newproducts@ssc.com

Editor Emeritus

Don Marti

dmarti@ssc.com

Technical Editor

Michael Baxter

mab@cruzio.com

Senior Columnist

Reuven Lerner

reuven@lerner.co.il

Chef Français

Marcel Gagné

mgagnie@salmar.com

Security Editor

Mick Bauer

mick@visi.com

Contributing Editors

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

Proofreader

Geri Gale

VP of Sales and Marketing

Carlie Fairchild

carlie@ssc.com

Marketing Manager

Rebecca Cassity

rebecca@ssc.com

International Market Analyst

James Gray

jgray@ssc.com

Sales Coordinator

Lana Newlander

ads@ssc.com

Regional Advertising Sales

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

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

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

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

Advertising Inquiries

ads@ssc.com

Publisher

Phil Hughes

phil@ssc.com

Accountant

Candy Beauchamp

acct@ssc.com

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

2825 NW Market Street #208, Seattle, WA 98107 USA

Editorial Advisory Board

Daniel Frye, Director, IBM Linux Technology Center

Jon "maddog" Hall, President, Linux International

Lawrence Lessig, Professor of Law, Stanford University

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

Church of Jesus Christ of Latter-day Saints

Sam Ockman, CEO, Penguin Computing

Bruce Perens

Bdale Garbee, Linux CTO, HP

Danese Cooper, Open Source Diva, Intel Corporation

Subscriptions

E-MAIL: subs@ssc.com

URL: www.linuxjournal.com

PHONE: +1 713-589-3503

FAX: +1 713-589-2677

TOLL-FREE: 1-888-66-LINUX

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

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

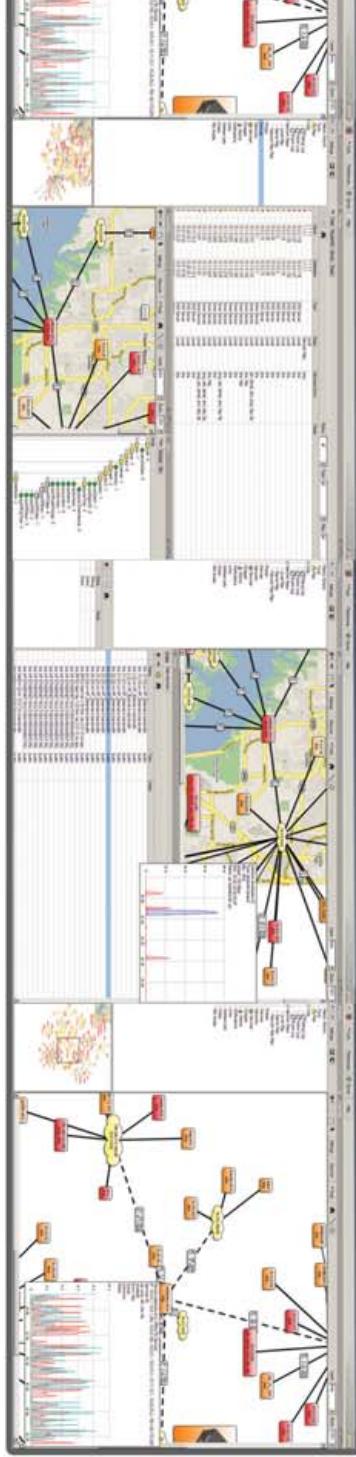
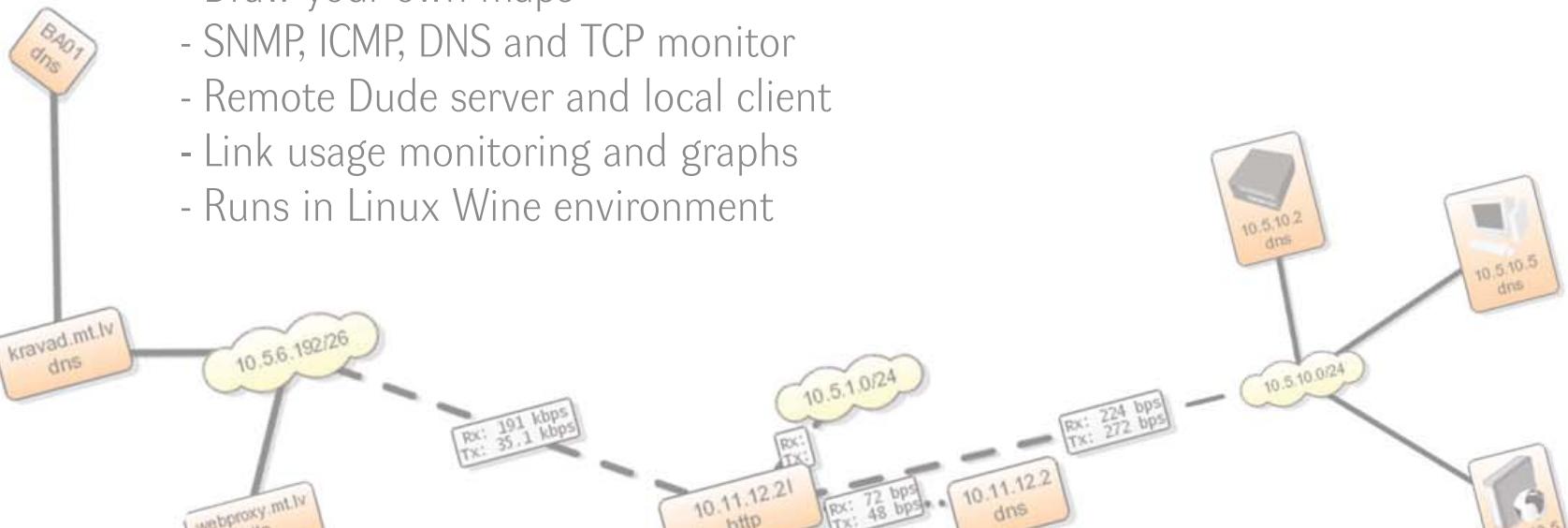
PRINTED IN USA

LINUX is a registered trademark of Linus Torvalds.

The Dude network monitor software will dramatically improve the way you manage your network environment. It will scan devices within specified subnets, draw and layout a map of your networks, monitor services of your devices, and alert you in case a service has problems. And it's free.

www.mikrotik.com/dude/

- Free of charge
- Network discovery and layout
- Discovers all kinds of devices
- Add new custom devices
- Device, Link monitoring, alerts
- SVG icons for devices
- Draw your own maps
- SNMP, ICMP, DNS and TCP monitor
- Remote Dude server and local client
- Link usage monitoring and graphs
- Runs in Linux Wine environment





DO MORE WITH LESS

For powerful servers that let you do more work on fewer systems, choose the **Dual-Core Intel® Xeon® Processor** in your **NEXXUS 4000®** Personal Cluster.

This Dual-Core Intel® Xeon® Processor based Personal Cluster server is the ideal platform for clients who have technical or mission-critical applications that require high processor count and fast interconnect. Based on a desk-side design, the NEXXUS 4000® is an alternative to traditional datacenter centric solution and the perfect system for IT or Research departments looking to leverage the benefits of Intel® standardization.



Visit us

For all your computer needs

Give us a call

CIARA-TECH.COM

1-866-789-7225

TECH

PERSONAL CLUSTER 8 PROCESSORS 16 CORES

\$19,995

8 Dual-Core Intel® Xeon® Processors 5148
32GB FBD ECC/Reg DDR2 533MHz
1TB (4 x 250GB) of Storage
Built-in Gigabit/KVM/USB Switch

Upgrade Your System

Additional 32GB Memory	\$6,599
Additional 1TB of Storage	\$1,099
InfiniBand Interconnect (including PCI/Express HCA, Switch and Cables)	\$3,455

- 150 Gigaflops¹ computing power.
- Up to 128GB² FBD DDR2 533/667MHz.
- Built in 8 or 16 Port Gigabit Switch.
(Optional 8 Port InfiniBand Switch)
- Up to 8 Serial ATAII HDD.
- Integrated KVM/USB Switch.
- Plugs into one single 110/220V outlet.
- Convenient desk-side form factor.

Colon, Colonno Inside, Centino, Centino Logo, Com Inside, Intel, Intel Logo, Intel Core, Intel Inside, Intel Inside Logo, Intel SpeedStep, Intel Viiv, Itanium, Itanium Inside, Pentium, Pentium Inside, Xeon and Xeon Inside are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. (1) Important Information. All prices, specifications and promotional offers are subject to change without notice. Clara cannot be responsible for typography errors, photographs errors, pricing errors. All pricing in US dollar. Shipping and applicable taxes are not included. (1) Based on peak performance with Dual-Core Intel® Xeon® Processor 5148. Available with 4GB HSiMMMS.





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



nero
7 ULTRA EDITION

with an upgrade
to AMD Athlon™ 64!



Get Nero 7 Ultra, a Monarch shirt or hat, a Livestrong™ Wristband, and a FREE GIFT when you upgrade to Athlon 64/FX/X2/Opteron 100 (939) or any Socket AM2 bundle!

FREE INSTALLATION
SETUP & TESTING
CERTIFIED TECHS

DFI NF4 Ultra Infinity
w/ AMD Opteron™
processor 146
(939 pin)

Only
\$229

Asus A8N-VM CSM
GeForce 6150
w/AMD Opteron™
processor 170
(939 dual-core)

Only
\$929



FREE TECH SUPPORT!

AMD Motherboard Combos

Asus M2NPV-VM
GeForce6150
with AMD Athlon™ 64 X2
processor 4200+
(Socket AM2)

Only
\$395

MSI K9N
SLI Platinum
w/AMD Athlon™ 64
processor FX-64
(Socket AM2)

Only
\$1,145

Supermicro (H8DAR-8)
w/ 2 x AMD Opteron™
processor 248
(940 dual-core)

Only
\$799

Tyan S2892G3NR
Thunder K8SE
w/ 2 x AMD Opteron™
processor 270
(940 dual-core)

Only
\$1,319

Mainboard - Processors - Heatsink and Fan with Memory Options - FREE INSTALLATION AND TESTING
LASTEST BIOS loaded for easy upgrades - AMD Sempron™ Athlon™ 64, Athlon™ 64 X2, Athlon™ 64 FX, and Opteron™ Combos Available

AMD Barebone Systems



Enermax CA-3030-BS ATX 1.3 Midtower
w/400W PS

Gigabyte GA-M51GM-S2G
AMD Sempron™ processor 3600+
(Socket AM2)

Starting @ **\$299**

Monarch Hornet Pro (Indigo)
w/270W PS

Asus M2NPV-VM GeForce 6150
AMD Athlon™ 64 processor 3500+
(Socket AM2)

Starting @ **\$685**

Go to www.monarchcomputer.com. select Barebones from the menu.
Choose AMD Sempron™, AMD Athlon™ 64 or AMD Opteron™. Then
configure your barebones online or call 1-800-611-0875.

Enermax CS-10182-BA E-ATX Full-Tower
w/550W PS

Supermicro (H8DAR-I) MB
2 x AMD Opteron™ processor 275 (940 Dual-Core)

Starting @ **\$1,925**

Tyan (Barebone) Transport GX28
w/400W PS

Thunder K8S Pro (S2882) Motherboard
2 x AMD Opteron™ processor 248 (940 Dual-Core)

Starting @ **\$1,289**



***AMD Athlon 64 and Athlon 64 FX are the first Windows®-compatible 64-bit PC processors

www.monarchcomputer.com

Components and Upgrades

1000s of In-Stock PC Parts and Accessories



100347
Cooler Master
Cavalier 3
CAV-T03-UW ATX
Mid-Tower,
No PS (Silver)

\$58.85



150939
Western Digital
Caviar SE
250 GB SATA3G
16MB Cache 7200 RPM
(WD2500KS)

\$79.99



190156
Visiontek (ATI)
Radeon X1300
PCI-E SFF
256MB GDDR2
DVI/TV-Out Retail

\$119.69



140123
512 MB DDR (533)
PC2-4200 Corsair
(VS512MB533D)

\$38.50



101258
Monarch/Coilmax
CTG-1000 1KW
ATX 2.2/EPS/SLI/RoHS
Power Supply

\$449.69



270270
Corsair 2 GB
CMFUSB202GB
Flash Voyager
USB 2.0 Drive

\$56.62



150061
3ware 9590SE-8ML
PCI-E x4
half-length
8-Ports

\$545.00



140803
1 GB (2 pcs 512)
DDR (400)
PC-3200
Corsair XMS
(TwinX1024-3200C2PT)

\$102.00

NEW! Section 508
Documentation
Available for our PCs!

Educational and
Government
POs Welcome.

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



Commercial leasing available for purchases as low as \$1000.

Please note that prices are subject to constant fluctuation. Remember to consult monarchcomputer.com for the most current pricing. Pricing is based on the date of issue in which this publication appears. Monarch Computer is not responsible for typographical or pricing errors. AMD, the AMD Arrow logo, AMD Athlon, and combinations thereof, are trademarks of Advanced Micro Devices, Inc. All brands and product names are trademarks or registered trademarks of their respective companies. AMD model numbers indicate relative software performance among this AMD processor family. *Enhanced Virus Protection (EVP) is only enabled by certain operating systems including the current versions of Microsoft® Windows®, Linux, Solaris, and BSD Unix. After properly installing the application/operating system release, users must enable the protection of their applications and associated files from buffer overrun attacks. Consult your OS documentation for information on enabling EVP. Contact your application software vendor for information regarding use of the application in conjunction with EVP. AMD and its partners strongly recommend that users continue to use a third party anti-virus software as part of their security strategy. **Please see monarchcomputer.com for important contest terms and conditions. No purchase required for entry.

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



Monarch's NEW EMPRO® 2 line of systems now features Next-Generation AMD Opteron™ processors with DDR2 and AMD Virtualization™!

Improving Direct Connect Architecture

- > For continued success in the enterprise
- Advancing Performance-per-Watt leadership**
- > Low-power, high-performing DDR2 memory
- > Consistent power roadmap with low-power options

Extending the Lead in x86 Virtualization

- > Founded on Direct Connect Architecture
- > AMD Virtualization is designed to improve business functionality and flexibility

Reducing Total Cost of Ownership (TCO)

- > One transition to your next stable platform
- > Seamless Dual-Core to Quad-Core upgrade in same thermal infrastructure
- > Improved Memory RAS and cost savings from DDR2



The AMD Opteron™ processor with Direct Connect Architecture provides industry-leading performance-per-watt and price/performance-per-watt.

"BOTTOM LINE: MUST BUY"



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

**Jason Perlow
 Linux Magazine
 April 2005**

**Check out MonarchComputer.com's AMD Store for more information on
 Next-Generation AMD Opteron™ 1000/2000/8000 Series processors**

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



LOOK FOR THIS SYMBOL
 ON LINUX COMPATIBLE
 MONARCH SYSTEMS!

You can be confident in knowing that AMD has longstanding partnerships with industry leaders –Microsoft and major Linux vendors.



Monarch Empro® 4-Way Tower Server

Ask for Part #: 80809



The AMD Opteron™ processor with Direct Connect Architecture scales from 1P/2-core up to 8P/16-core across a single industry-standard platform without external logic, allowing for maximum versatility and lowering overall system cost.

Starts @ Only
\$2,685!

SELECTED COMPONENTS:

Cooler Master RC-810 CM Stacker 810 (Silver or Black)
 Tagan TurboJet TG900-U95 ATX 2.0/EPS 2.9 900W SLI Power Supply
 Tyan S4882UG2NR 8131 Motherboard
 Video/GB-LAN/USB/SATA/SCSI/DDR-400 E-ATX Dual-Core Quad-Opteron
 Featuring 2 AMD Opteron™ Processors 870 16 GB DDR (400) PC-3200 Patriot REG ECC Signature (PSD2G40036ERB)
 4 x Seagate Cheetah 146 GB 15K RPM Ultra 320 68pin SCSI w/RAID 5 w/Hot Spare Adaptec 2200S Ultra 320 SCSI RAID Controller Plextor PX-755SA/SW-BL Dual Layer 16x8x16x DVD±RW 8x-DVD+R DL Drive Mitsumi 1.44MB 3.5" (Black) Floppy Drive Choice of Sound Card Choice of PCI Video Card Choice or Linux or Microsoft OS Choice of Network Options Industry Standard Upgradable Up to 3 year warranties available 24/7 on-site service available

This config **\$10,281!**

Monarch Empro® Custom 3U Rackmount SAS Server

Ask for Part #: 80661



AVAILABLE COMPONENTS:

Choice of AIC RMC3E-95R-SAS or AIC RMC3E-95R-SAD 3U Rackmount chassis with Hotswap SAS, Rail Kit & (2+1) Redundant 950 Watt EPS Power Supply
 Choice of Tyan Thunder K6S or Thunder K8SE E-ATX Dual Core ready Motherboard Up to 2 AMD Opteron™ 200 Series Processors (Single or Dual-Core)
 Up to 32 GB DDR (400) PC-3200
 Up to 4.8 Terabytes of storage capacity when using SAS Hard Drives (8 Terabytes with SATA drives) with options for RAID 0,1,5,6 or 10 Setup Up to 2 Optional RAID Controllers
 Optional Optical Drives w/Software
 Choice or Linux or Microsoft OS
 Choice of Network Options
 Optional Rail Kit
 Industry Standard Upgradable
 Up to 3 year warranties available
 24/7 on-site service available

Custom configs Starting @ ONLY
\$4,022!

GET QUICK QUOTES

Online or by phone:

www.monarchcomputer.com

1-800-611-0875

Paypal - Visa - Mastercard - Discover - AMEX

1 Configuration review

Our senior technical staff review every configuration to eliminate hardware and software incompatibilities.



2 Production scheduling

We allocate all parts for your system, and chart your system's assembly path through our production facility.



3 Board test

We assemble your motherboard, processor and memory and test these core components extensively. We also load the latest BIOS.



4 Build stage

All components for your system are assembled into your chassis. All cables are tied off and tucked away to increase airflow and cooling.



5 Burn in diagnostic

We combine hands-on diagnostics with a battery of automated burn-in testing to ensure all your components are operating properly together.



6 Software load

We load your OS onto your hard drive along with all factory tested updates and the most recent hardware drivers.



7 Quality control

Our QC experts put your system through a rigorous 62 point inspection to verify the system is in working order before shipping.



8 Packing & shipping

We expertly pack your system for secure and safe shipping using customized packaging and double boxing.

letters



Xoops! Not a Security Hole

In response to Cameron Spitzer's letter [June 2006, "Xoops! A Security Hole"], Juan Marcelo Rodriguez may have forgotten to mention that Xoops needs the permissions of the uploads, cache and templates_c directories set to 777 just for the install. You can change them back! When you log in to the application for the first time, if they are still set to 777, the application will ask you to change them.

--
Mohammed

Trolling for Ranting Ranters

I like how the magazine has changed. I think it's exciting. It's like watching it grow (literally, too). The design upgrade brings it closer to the style of *Network Computing*, *IT Architect* or other glossy CMP periodicals I also read, and nicely not so "pagemaker crayola" as *Wired* looks. I think this look reaches out to those who would flip thru the magazine rack and pick up a *Wired* or *PC Gamer*. That must have provoked an argument or two, because that's also an editorial and policy change for the whole magazine. It hasn't lost the hard tech edge, though. If I ever want to read about how to write a device driver, I don't have to look very far.

Boy, this has been fun reading the Letters column. Look at all the people getting riled up about Nick's trolls! I thought I saw Nick sitting under a big bridge somewhere, waiting to point out the obvious with his thick finger, just to watch all the readers squirm. I think it's brilliant! (Who among us has not enjoyed reading through a trolling thread on Slashdot?) This makes the magazine *not boring*, and that will keep plenty of us around.

Mod me down for my casual comments but don't unsubscribe me. I think it's important that Nick point out the obvious issues in the Linux community. It's like big group therapy. "Uncle Nick" is going to make fun of distros that are lagging behind! Of course he appreciates how hard they all work, that's given. But he really doesn't want any distro to lag behind, because that doesn't do service to Linux as a whole.

Maybe if "Uncle Nick" points out some of these obvious things, we'll wake up and help out. Or maybe we'll listen to our own trolls and hear ourselves. When *Linux Journal* had the "nekkid piano player" cover, the Letters column was filled with the laments from the stuffed-shirts—that was fun! I see more of that, but now it's even more cogent to the discussion: by pointing out these obvious things in an opinionated way, he's poking us in the ribs to at least join in the conversation! (And maybe we'll become involved.)

--
Jed Reynolds

Simple Is Good

The "Google Maps" installment of Reuben Lerner's At the Forge column [June 2006] was simple enough to get lots of noobs excited—that's great! However, creating dynamically named variables like myMarker\$count is bad style that I'd rather keep away from the kids. JavaScript has locally scoped variables, so there's no need for that kind of 1970s hackery.

--
Mike Schilli

Change Is Good Too

I can't even make it off page 10 of the July 2006 issue that I just got today. For the last several months, I have read for the most part how "concerned" or "upset" people are for the changes in *LJ*. And how they are going to quit subscribing to the magazine, unless things are turned back around again soon. Why is it that people are so afraid of change? Change is good folks, really, it is!

Anyways, I love the new dimensions of the magazine. I really enjoy the /etc/rants too. I find a good chuckle in them, and they're informative too. I also like the in-depth coverage, even though some readers may feel that it's a waste. I am sure that there are other readers that feel the same as I do. Kudos to *LJ* for making it fun to read again and get the feeling of a child again, waiting for Christmas day to come, when the new magazine is supposed to be coming in the mail any day now.

--
Ryan Chiles

Causes and Effects

In response to Doc Searls' "Causes and Effects" article [July 2006], I just wanted to say thanks. I really enjoy reading articles like this. Being a supporter of Net Neutrality, I feel that we need to get the word out more. We need to do what we can to make sure people are aware of proposals like Ted Stevens' Telecom bill because like most things, people aren't going to take notice until it is directly affecting them, and usually by then it is too late. Once the door is closed, it takes a lot for people to get that door opened back up.

--
Nick Baronian



» Enterprise Solutions Built on *OPEN STANDARDS*

>> We can help you migrate to Open Source Solutions in no time. We have the Enterprise server and storage solutions you are looking for, at a price you never thought possible. By using open standard components and software, we can pass significant cost savings to you. And with our new state-of-the-art, fully automated production facility, we can build and **SHIP** hundreds of machines daily, further saving you money by not having to wait weeks, or even months for your mission critical systems to arrive. All solutions are built to your specifications, loaded with your choice of software, and backed by a industry leading 3 year warranty.

Open Source Storage is now Open Source Systems !

Come visit us at www.OpenSourceSystems.com



866-664-STOR
Sales@OpenSourceSystems.com
www.OpenSourceSystems.com
1195 Borregas Avenue, Sunnyvale, CA 94089



What Is with the Pro-KDE Slant?

I don't understand this. Since that copy of *LJ* with the rant on KDE vs. GNOME [April 2006], I've been reading a lot of support for KDE in *LJ* (for the last two issues, no less), while scarcely any for the GNOME side. I find Mr Soulier's "Even Einstein Agrees" [June 2006, Letters] rather cute, but I think he attempted to apply Einstein's quote backwards.

My school's labs all use KDE, and I'm almost tempted to configure .xsession or .xinitrc to use twm instead. I think the following analogy makes the most sense. KDE is like Gentoo, you could configure it to a very fine level of detail, but you can hardly do any work until you have tinkered with it. GNOME is like Ubuntu, you need only very little customization to make it work the way that you want and expect. Things in both Ubuntu and GNOME are well integrated. I'm not sure I'd use the same description for KDE or Gentoo. I'm thankful that Linus Torvalds started the Linux movement, but he couldn't be more wrong with the KDE vs. GNOME issue. And I know more than enough people (many of whom are far smarter than I am, and from whom I seek advice and respect their opinions) in my Computing Science Student Society who couldn't think of a single good thing to say about KDE.

With such a severe lack of balance on this issue, sometimes I'm glad that I don't subscribe to *LJ*. I used to have more respect, when it did create dialog. Now it almost seems like discussion has degenerated to fan-boyism, and dissent is not tolerated.

--
Benton Lam

I fail to see any pro-KDE slant in Linux Journal content. Outside of those (including myself) who express their opinions one way or another, we take great pains to make sure our content reflects a neutral view.—Ed.

Damn Small Is Damn Good

Thank you very much for publishing the "USB Pendrives and Distributions for Them" article in *LJ* [June 2006]. I have toyed with DSL (Damn Small Linux) and even purchased one of their pendrives to contribute to their project. In the past, however, I wasn't able to follow through with creating my own pendrive, and Juan Marcelo Rodriguez's article came as a godsend. I downloaded DSL-N, which has the 2.6 kernel. Despite the fact that this project still has some bugs, I sense lots of potential in it and will try to help them by reporting bugs and testing. I felt very happy about making my own pendrive (I have some troubles with the wireless card but I haven't done enough research on the matter to be able to ask smart questions), and I look forward to testing other live distros.

--
Diego A. Acosta

Maddog's Compaq History Rears Its Head

I read, with not inconsiderable interest, parts of the article by Jon "maddog" Hall entitled "Sinking of the USS Proprietary", *Linux Journal*, July 2006, wherein appeared: "USS Compact". Well, now, I must really wonder if he instead meant "USS Compaq". I just hope you can now concur! "And ye shall know the truth, and the truth shall make you free" (ref.: John 8:32, KJV, 1611). *Tempus fugit et ad augusta per angusta.*

--
Joseph Roy D. North

For Shame

Shame on you for using proprietary software. Of all people, an old OS/2 veteran such as yourself should surely be aware of the fundamental evils of proprietary software.

Remember TrueSpectra Photo-Graphics? It was a wonderful OS/2 application that seamlessly combined raster and vector graphics. Where did it go? Down the proprietary rat hole. All the progress that TrueSpectra made in image processing has effectively vanished as if it had never existed. That is an unconscionable waste, and we are still waiting for someone to reinvent the wheel.

While I formerly tolerated proprietary software, experience has pushed me toward a different view. I now stand in full agreement with Hans Reiser, who says, "Equal Source Code Access Is A Civil Right".

--
Carl Brown, Another old OS/2 veteran

New Life in LJ

After being a subscriber to *Linux Journal* from the very first issue, I must confess that the articles started to lose interest for me around the year 2000. I assumed the journal had, to use the TV metaphor, "jumped the shark", given the popularity of everything with the word Linux at the time. It was almost simply out of loyalty that I have continued to resubscribe the last several years. That is, until about a few issues ago. Suddenly, the articles are much more relevant and useful. Whatever you have done, please keep it up!

--
M. Marlowe

LINUX JOURNAL**At Your Service****MAGAZINE**

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

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

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

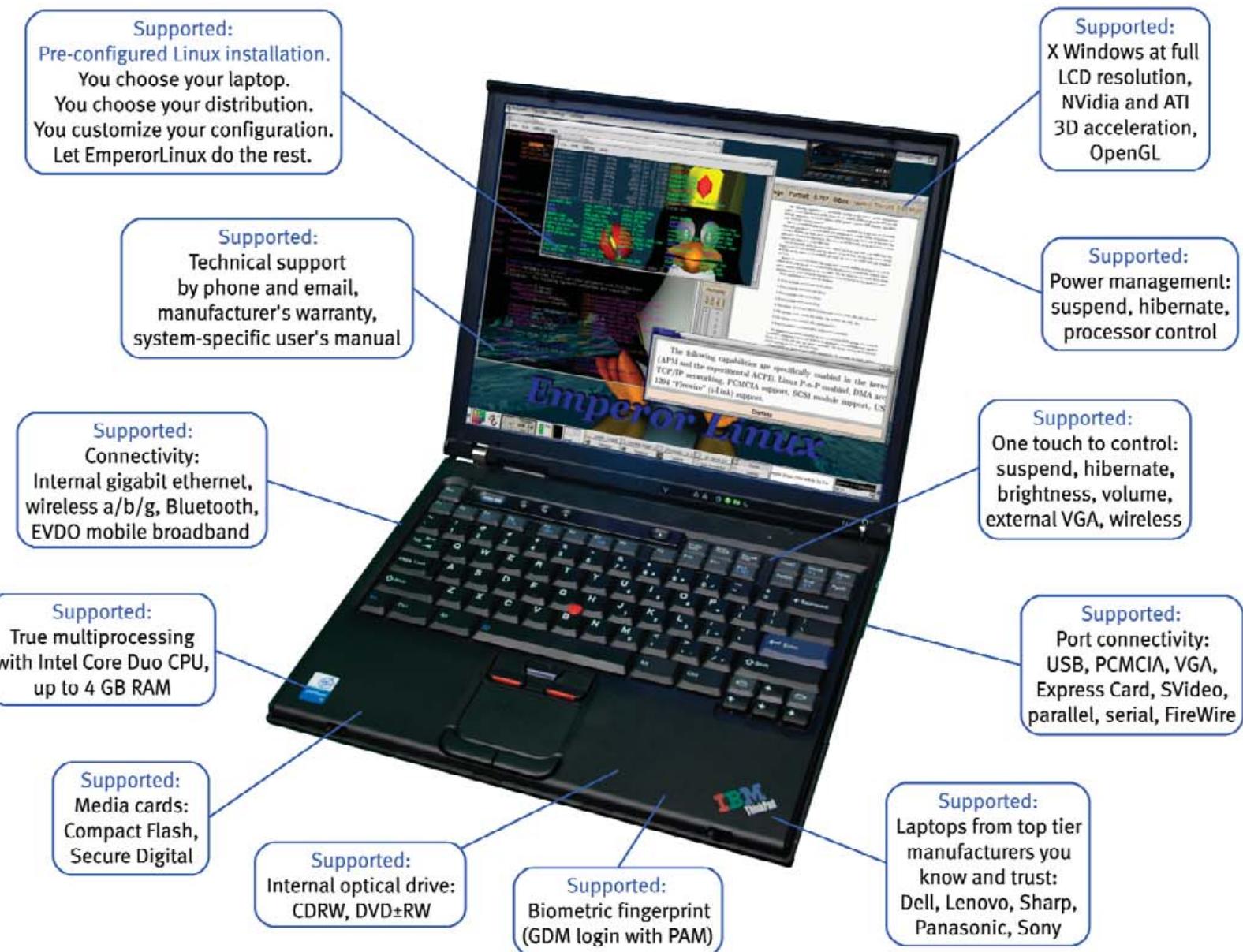
ADVERTISING: *Linux Journal* is a great resource for readers and advertisers alike. Request a media kit, view our current editorial calendar and advertising due dates, or learn more about other advertising and marketing opportunities by visiting us on-line, www.linuxjournal.com/advertising. Contact us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

ON-LINE

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

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

Linux laptops. Supported.



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

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

EmperorLinux

...where Linux & laptops converge

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

Model specifications and availability may vary.



diff -u

WHAT'S NEW IN KERNEL DEVELOPMENT

"The kernel has *one* mission in life, and one mission only. Guess what that is? It's to be the buffer between user space and shared resources. That's it. NOTHING ELSE MATTERS." —Linus Torvalds

Greg Kroah-Hartman

has produced a shiny new **device driver developer's kit**, which he hopes will rival those available for other operating systems. Distributed as a GPLed CD image, the kit contains a variety of source code and documentation, including a full copy of the third edition of *Linux Device Drivers*. Future versions will add a searchable index of all included docs. There also is talk of incorporating the kernel news sections from lwn.net into the mix. A lot of kernel folks are interested in seeing this kit grow into something very powerful, and many of them have offered ideas and suggestions. Undoubtedly, this project will accrue large numbers of contributors and fill what has been a very large gap.

Chris Wedgwood ran into a problem when attempting to unmark a few features as "experimental" recently. The whole point of tagging a feature experimental is, on the one hand, to protect users from inadvertently using dangerous, unfinished features, and on the other hand, it is to allow them to experiment with those features if they so choose. But with no clear definition of when something should or should not be considered experimental, kernel folks have been marking their projects as experimental under all kinds of different circumstances. The result of this was that so much com-

pletely usable code had been marked experimental, many users just had to enable experimental by default in all their builds, thus defeating the whole point of the feature. Typically what would happen now is that folks would argue over which definition of experimental would be most appropriate as the official definition.

After a lengthy, bitter debate, **Linus Torvalds** would come up with something that puts a whole different face on the problem, and this would be refined in practice over months and years. Will that happen in this case? Perhaps, but we just don't know.

Linux has apparently been out of compliance with **POSIX hostname lengths**, and **Randy Dunlap** sent in a patch to correct this by raising the hard-coded limit from 64 characters to 255, not counting the terminating null. Although Linus Torvalds always has insisted that POSIX should not be blindly complied with, and that Linux should first of all do the right thing regardless of official standards, it does seem as though the hostname issue was a genuine oversight, and Randy's patch will go into the 2.6.17 kernel.

The current limit of five **nested symbolic links** also is likely to be raised to nine for the 2.6.17 release. **Alexander Viro** had alerted folks back in February 2006 that he would do this, and he submitted a patch to accomplish it when the subject came up again more recently. **Andrew Morton** gave his reluctant approval to this, pointing out that this would not be a backward-compatible change. Any application extended to rely on having more than five nested links

would be unable to run on older kernels with the smaller limit. Given that it really was high time the five-nest limit was raised, however, Andrew affirmed that 2.6.17 would be the right time to do it. And, as some folks pointed out during the discussion, distributions like **Red Hat** had raised the limit to nine some time ago on their own.

Jared Hulbert has been working on a filesystem targeted at embedded devices, specifically phones. **AXFS** is short for Advanced XIP Filesystem, and XIP is short for eXecute In Place. AXFS launches and runs programs directly from their non-volatile storage location, instead of first loading them into RAM. There are several benefits to this, and some developers, such as **Mark Lord**, already have expressed serious interest in using AXFS on real-live systems. One benefit of AXFS is that program initiation may be faster without the overhead of copying the program code to RAM. Another benefit derives from the fact that RAM-based systems typically store their programs compressed in Flash. On execution, the programs are uncompressed before being copied to RAM. With AXFS, programs would be stored uncompressed, requiring more Flash for storage, but would do away with RAM entirely, which could lower the overall cost of producing the device. AXFS is not yet ready for prime time, as Jared has been quick to point out. His main purpose in sharing his work so far has been to get opinions and guidance from the kernel folks.

—Zack Brown

Google Earth Now Supports Penguins

In his CES keynote in January 2006, Google cofounder Larry Page gave the consumer electronics industry some well-deserved heat for producing exclusive, incompatible and non-interoperable gear. Also, at the same show, Google promoted a package of software products that ran only on Windows. In the Q&A, I pointed out this irony and asked what Google would do to fix its own platform-exclusivity problems. Larry admitted that this was "a problem", and said they were working on fixing it.

We reported on one result—Picasa for Linux—last month. This month, we can report that Google Earth is also

available for Linux as well. The first iteration, just released at this writing (in May 2006), is Google Earth 4 beta, also released simultaneously on Windows XP and Mac OS X.

Unlike Picasa's Linux implementation, which was built with Wine, Google Earth for Linux is a native application, built with the Qt development kit. It requires 2.4 or later kernel versions, and it has been tested on modern versions of Ubuntu, SUSE, Fedora Core, Linspire, Gentoo, Debian and Red Hat. Download the free application at earth.google.com.

—Doc Searls

top - 13:24:28 up 2 min, 2 users, load average: 1.33, 0.69, 0.27
Tasks: 154 total, 2 running, 152 sleeping, 0 stopped, 0 zombie
Cpu0 0.0% ni, 99.6% id, 0.0% wa, 0.0% si
Cpu1 0.0% ni, 100.0% id, 0.0% wa, 0.0% si
Cpu2 0.0% ni, 99.6% id, 0.0% wa, 0.0% si
Cpu3 0.0% ni, 99.6% id, 0.0% wa, 0.0% si
Cpu4 0.0% ni, 99.6% id, 0.0% wa, 0.0% si
Cpu5 0.0% ni, 99.6% id, 0.0% wa, 0.0% si
Cpu6 0.0% ni, 99.6% id, 0.0% wa, 0.0% si
Cpu7 0.0% ni, 99.6% id, 0.0% wa, 0.0% si
Cpu8 0.0% us, 0.0% ni, 99.6% id, 0.0% wa, 0.0% si
Cpu9 0.0% us, 0.0% ni, 99.6% id, 0.0% wa, 0.0% si
Cpu10 0.0% us, 0.0% ni, 99.6% id, 0.0% wa, 0.0% si
Cpu11 0.7% us, 0.0% ni, 98.3% id, 0.0% wa, 0.0% si
Cpu12 0.0% us, 0.0% ni, 99.6% id, 0.0% wa, 0.0% si
Cpu13 0.0% us, 0.0% ni, 99.6% id, 0.0% wa, 0.0% si
Cpu14 0.0% us, 0.0% ni, 99.6% id, 0.0% wa, 0.0% si
Cpu15 0.0% us, 0.0% ni, 99.6% id, 0.0% wa, 0.0% si
Mem: 0.0% us, 0.0% ni, 0.0% id, 0.0% wa, 0.0% si

What could you do with 16 cores?



THINKMATE 8-WAY WORKSTATION FEATURING AMD OPTERON™ PROCESSORS

This system offers nothing less than the flexibility and power to meet or exceed your computing requirements. Need a 5U rackmount? Need a full-scale tower? It's convertible to both form factors!

Better Efficiency, Greater Productivity, and Enhanced Scalability

Operating System

Microsoft Windows, Red Hat Enterprise Linux, and Sun Solaris
Operating System Configurations

System Memory

Supports up to 128GB PC2700 DDR ECC Registered Memory

Video

PCI Express x16 or Integrated ATI Rage XL 8MB

Storage

Up to 8 Removable Serial ATA or SCSI Hard Drives

Power Supply

1350W 3+1 Redundant Power Supplies

Thinkmate Warranty

Thinkmate systems are warranted against defects in materials and workmanship for three years following the date of purchase.

- 3-Year Advanced Replacement of Defective Components
- 3-Year Technical Support
- Optional 3-Year Onsite with IBM Global Services
- Optional 4 Hour Response

- 8P/16C system with Dual-Core AMD Opteron™ processors Model 885 2.6GHz 2x1MB Cache
- 128GB (32 x 4GB) PC2700 DDR ECC Reg. Memory
- PNY nVidia Quadro FX 5500 SDI 1GB Graphics Adapter with Next-Generation Vertex and Pixel Programmability
- 8 x 300GB Seagate Ultra320 SCSI 10,000 RPM Hard Drives with Adaptec 2230SLP Ultra320 SCSI RAID Controller Card

16 Cores

\$62,999

- 8P/16C system with Dual-Core AMD Opteron™ processors Model 885 1.8GHz 2x1MB Cache
- 8GB (16 x 512MB) PC3200 DDR ECC Reg. Memory
- PNY nVidia Quadro FX 4500 512MB Graphics Adapter with High Precision Dynamic Range Imaging (HDPR) Technology
- 4 x 400GB Seagate Serial ATA 7,200 RPM Hard Drives with 3ware 9550SX-8LP Serial ATA RAID Controller Card

16 Cores

\$17,999

THINKMATE

[800] 371-1212
WWW.THINKMATE.COM

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

LJ Index, September 2006

1. Number of hours per day wasted on misuse of office technology by the average British worker: **1.17**
2. Number of British office hours wasted per day by communications technologies not being used to good effect: **1.63**
3. Number of British office worker minutes per day spent chasing responses to urgent e-mails: **42**
4. Minimum Linux percentage of Dell enterprise hardware sales: **25**
5. Minimum number of UNIX-to-Linux migrations managed by Dell Service: **500**
6. Minimum percentage of Red Hat service calls to Dell handled without involving Red Hat: **90**
7. Thousands of desktops migrated to Linux (SUSE, KDE) by the Regional Tax Office in Lower Saxony, Germany: **12**
8. Peak number of desktops migrated per day by the Lower Saxony tax office: **300**
9. Minimum number of desktops switched to Linux in Schwäbisch Hall, Germany: **400**
10. Number of desktops planned for switching to Linux in Mannheim, Germany: **3,700**
11. Number of servers planned for switching to Linux in Mannheim: **110**
12. Percentage of local German authorities using open-source software: **90**
13. Millions of Internet users in India in 2005: **38.5 million**
14. Thousands of Internet cafés in India: **105**
15. Millions of US households promised connection by symmetrical fiber-based 45MB/s broadband by 2006: **86**
16. Billions of dollars collected by carriers in higher phone rates and tax perks, toward promised fiber broadband deployment: **200**
17. Number of operating local or regional fiber-to-the-home deployments in the US, as of April 2006: **936**
18. Millions of US homes served by those deployments as of April 2006: **4**
19. Number of non-RBOC (phone company) fiber-to-the-home deployments: **580**
20. Number of RBOC (phone company) fiber-to-the-home deployments: **376**

Sources: 1–3: ntl Incorporated | 4–6: CNet | 7–12: ZDNet UK | 13: *Hindustan Times* via NeonCarrot.co.uk | 14: ConSu, Nov 2005, via NeonCarrot.co.uk | 15, 16: TeleTruth.org | 17–20: RVallic.com and broadbandproperties.com

—Doc Searls

They Said It

The core fallacy of the idea of progress is the notion that it is possible to optimize everything at once.

—Philip Slater, radicalcentrism.org/pipermail/centroids_radicalcentrism.com/2005-August/001503.html

Premature optimization is the root of all evil.

—Hoare's Dictum, by Tony Hoare, jeff-kubina.blogspot.com/2006/03/etech-2006-session-scaling-fast-and.html

I don't know the key to success, but the key to failure is trying to please everybody.

—Bill Cosby, radicalcentrism.org/pipermail/centroids_radicalcentrism.com/2005-August/001503.html

In short order, the \$100 Laptop will debut in the developing world—running on the aforementioned free operating system, Linux. WiMaxx networks will blanket this world, just as cell networks now blanket Kenya and other parts of Africa. (Almost everyone I run across in Kenya has a cell phone—including people who live in Kibera, the largest slum in East Africa.) The developing world will be connected at a level unimaginable two years ago. Millions of new voices will join the conversation. Issues and problems will be revealed, discussed and solved in those very conversations.

Governments will fall, corruption be revealed, new ideas explode and lives be radically changed as the Generous Web weaves its magic throughout the planet.

—Bill Kinnon, www.kinnon.tv/2006/06/the_globalimpac.html

If you repeal the DMCA and the EUCD, makers of competing MP3 players will reverse-engineer FairPlay and add the capability to play iTunes songs. No further government oversight will be required. But if you pass additional regulations, we'll have to come back in another decade to figure out how to deal with the unintended consequences of those regulations.

—Tim Lee, www.techliberation.com/archives/039480.php

From each flavor of Linux

to bite-size browsing on Nintendo DS™ -
Opera goes where you go.

Follow the standards.

Break the rules.

Test in Opera first.

Download the **free**
Opera browser -

www.opera.com



Park Freely

When GoDaddy.com moved 4.5 million domain names from Apache to Microsoft's IIS on Windows 2003 (and paid to by Microsoft, reports said), it caused a 5% sever share drop for Apache on the next month's Netcraft report. In response, Bruce Perens created OpenSourceParking.com, a no-cost parking lot for undeveloped domain names. In addition to helping restore Apache's pre-eminent position (which has been between 65% and 70% for a while now), it helps Linux as well by telling Netcraft that all the parked sites are running Apache on Linux. Also, revenue from advertising on the parked domains will fund efforts on behalf of open-source and free software.

To park your undeveloped domain there, set your domain servers to ns1.opensourceparking.com and ns2.opensourceparking.com. DNS operators can set their undeveloped domain names to be a CNAME of opensourceparking.com. Domain resellers not tempted by Microsoft candy can set parking site names to opensourceparking.com.

—Doc Searls



Driving to Domination

Linux is a tortoise to the proprietary OS hares of the world. We all know how the story goes, and we've watched it play out in servers and embedded devices. (Still waiting on cell phones, but the turtle is gaining there too.)

Desktops and laptops are another matter. In no race has the going been slower for the tortoise, or its ultimate victory more inevitable, than in the race to make personal computers ready to drive any device that comes along. Lack of device drivers has always been blamed for slowing Linux's march to world domination in desktops and laptops.

But Linux will win that one too. Why? Because Linux likes its drivers in the kernel tree. There's no need to annoy the hacker and confuse the user with a CD full of binary code and promotional lockware. Instead, you write an LKM. Here's how the HOWTO (www.tldp.org/HOWTO/Module-HOWTO/x68.html) explains it:

Some people think of LKMs as outside of the kernel. They speak of LKMs communicating with the kernel. This is a mistake; LKMs (when loaded) are very much part of the kernel. The correct term for the part of the kernel that is bound into the image that you boot, i.e., all of the kernel except the LKMs, is "base kernel"; LKMs communicate with the base kernel....

There is a tendency to think of LKMs like user-space programs. They do share a lot of their properties, but LKMs definitely are not user-space programs. They are part of the kernel.

The HOWTO goes on to explain the many advantages of LKMs. Some big hardware vendors have discovered those advantages too. At the Linux Desktop Summit earlier this year, HP showed off how well its printers "just work" with a Linux box.

Ah, but what if you're not HP? What's to stop you from developing a device driver of your own?

Not much anymore. Greg Kroah-Hartman, Linux kernel subsystem maintainer, author of the LKM HOWTO and co-author of *Linux Device Drivers*, 3rd edition (O'Reilly, 2005), released a Device Driver Kit at FreedomHEC. "Have you felt left out of the crowd when looking at the 36 CD-ROM package of documentation and example source code that other operating systems provide for their developers? Well feel ashamed no longer!" he wrote. His version, in the Linux tradition, is comparatively minimal: one CD image with what Greg says is "everything that a Linux device driver author would need in order to create Linux drivers". That includes a full copy of Greg's *Linux Device Drivers* book and pre-built copies of all in-kernel docbook documentation. "It even has a copy of the Linux source code that you can directly build external kernel modules against", Greg adds. The kit is based on the 2.6.16.18 kernel release.

Anybody can download a copy for free here: www.kernel.org/pub/linux/kernel/people/gregkh/ddk.

—Doc Searls



The Industry Leader for Server Appliances



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

From design to deployment, we handle it all.

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

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

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



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



REUVEN M. LERNER

JavaScript

Like the language or hate it, JavaScript and Ajax finally give life to the Web.

About 18 months ago, Web developers started talking about Ajax. No, we weren't suddenly interested in scrubbing our kitchen countertops; rather, we were looking to take advantage of the latest paradigm in Web development. The technology itself wasn't all that new, but the idea that it was a general paradigm Web developers could adopt was new—and when it was given the snappy term Ajax (short for asynchronous JavaScript + XML) by James Paul Garrett, it was only a matter of time before everyone started demanding Ajax Web applications.

The theory behind Ajax is a relatively simple one. JavaScript excels at dynamically changing the HTML pages that Web browsers display. Modern versions of JavaScript are able to make asynchronous HTTP requests to a server. If we combine these two features, we can create desktop-like applications within the browser. Suddenly, the Web is a stable platform for all sorts of new applications. Google Maps was just the beginning; get set for Ajax word processors, spreadsheets, network-administration programs, drawing programs—you name it.

And indeed, we have seen an explosion of Ajax applications in the last year or so. Startups established to create Ajax versions of existing applications already have been bought by companies such as Google. Existing Web sites are scrambling to include Ajax functionality. Book publishers are printing Ajax-related books like they're going out of style. I probably know of at least six toolkits for adding Ajax to applications, and new ones are being released all the time.

Much of the excitement behind Ajax is the freedom it gives

Rather, we should learn JavaScript, understand its good and bad aspects, and then use it as wisely as possible in our applications.

designers and developers. Before Ajax, Web applications could be beautiful to look at, but their page-based interfaces were reminiscent of old mainframes, whose applications ran on a page model. What you want to create an application that is updated incrementally? Sorry, the HTTP/HTML combo means that you either got a new page, and got to enjoy the functionality that it offered, or you stayed on the current one. Every page update had to be accompanied by an HTTP request, and vice versa.

There is no doubt that Ajax applications have a cleaner look and feel to them than old-style Web applications. They feel more natural and responsive, and it's easy to imagine all Web applications looking like this within a few years. This is probably a good thing overall, and I'm looking forward to what the future will bring. In fact, I would guess that within a

few years, saying you're an "Ajax developer" will sound as funny as saying you're a "cookie developer", or a "DOM developer" or even a "database developer". Just as understanding each of these technologies is now an expected part of being a Web developer, the same is true for Ajax. Yes, this means that Web developers have yet another set of technologies to learn if they want to keep up.

Starting this month, I begin looking at Ajax, with an emphasis on the core of what you'll need to know in order to adjust to this new paradigm. I hope to cover the entire stack, starting with the underlying infrastructure on which Ajax depends—JavaScript, dynamic HTML and XML—moving up to the Ajax techniques themselves, and finally looking at libraries and frameworks that provide prepackaged Ajax functionality.

This month's column begins with the JavaScript language, which sits inside of every major Web browser. By the end of this column, you should have a good idea how to write basic functions in JavaScript and how to ensure that they are called automatically under a variety of different circumstances.

JavaScript

As a language, JavaScript has a long and checkered history. It began as a language called LiveScript, which was put inside of Netscape's browsers so as to differentiate them from the competition. The name was changed to JavaScript when Sun Microsystems unveiled its Java language, including applets that sit inside of browsers. The overwhelming excitement over Java applets convinced Netscape to change the name to JavaScript, causing a great deal of confusion. Microsoft then released a largely compatible version of JavaScript in Internet Explorer, called JScript.

After several years in which Web developers dealt with incompatible versions, Netscape had the language standardized by the European organization ECMA. Officially, the language is now known as ECMAScript, but no one really calls it that. The versions in Internet Explorer and Mozilla are now largely compatible with the standard, although there are still differences and issues to work around.

Although JavaScript occasionally has been used in other places, it is overwhelmingly a language that sits inside a Web browser. Its strength is that it allows Web developers to turn static pages of HTML into interactive applications. JavaScript can control every aspect of a Web page, including the contents, forms and design. JavaScript accesses this information via the DOM (document object model, a standard of the World-Wide Web Consortium for describing HTML and XML documents) and the BOM (browser object model, an unstandardized way of getting information about the browser and its objects, such as windows).

I should be up front right now and indicate that I don't particularly like the JavaScript language. Although it has improved substantially in every way since it was first released, a



**Just because your IT equipment goes dark
doesn't mean you have to go blind.**



SecureLinx SLC

Reach your IT equipment from anywhere as easily as changing a light bulb.

SecureLinx™ SLC secure console managers from Lantronix provide consolidated access so you can control, manage and repair your IT equipment from anywhere, at anytime.

For your free Console management white paper visit lantronix.com/slcpw/

Network down?
No sweat.
On the road?
No problem.

With SecureLinx you can finally achieve true "lights out," out-of-band data center management. And SecureLinx SLC has your back with the highest level of security available. For more information, check out the specs at lantronix.com/slc/ or call us at (800) 422-7055.



LANTRONIX®
Network anything. Network everything.™

© Lantronix, 2006. Lantronix is a registered trademark, and SecureLinx is a trademark of Lantronix, Inc.



Lantronix's SecureLinx SLC1
Awarded Network Computing's
Editor's Choice Award



number of aspects really bother me, including the object model and the multiple ways in which undefined values can be represented. That said, disliking JavaScript doesn't change the fact that it sits at the core of Ajax development. Web developers who ignore JavaScript, or who hope that it will simply go away or even be replaced by a different language, are being unrealistic. Rather, we should learn JavaScript, understand its good and bad aspects, and then use it as wisely as possible in our applications.

JavaScript is normally placed inside of the `<script>` tag, inside of an HTML document. In theory, we can put JavaScript inside of any type of HTML document, ignoring the standards. In practice, it is wise to use XHTML, the XML-compliant version of HTML, when working with JavaScript and other Ajax technologies in any serious way. This increases our ability to predict whether a page will work correctly with JavaScript, as well as how it will be rendered by the user's browser. Thus, a simple page containing JavaScript might look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <script type="text/javascript">
        function helloWorld() {
            alert("Hello, world!");
        }
    </script>
    <title>Test JavaScript page</title>
</head>
<body>
    <p>Hi there!</p>
</body>
</html>
```

The HTML part of the page is presumably familiar to you. Between the `<script>` and `</script>` tags, we have defined a JavaScript function, `helloWorld`, which pops up a modal dialog box containing our text. The above page works just fine, but the JavaScript itself is never executed. How can we get it to fire?

Perhaps the simplest way is to create a button—an `<input>` type meant for exactly this task—and have that button then execute our code. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <script type="text/javascript">
        function helloWorld() {
            alert("Hello, world!");
        }
    </script>
    <title>Test JavaScript page</title>
</head>
<body>
    <p>Hi there!</p>

    <p><input type="button" value="Click me!">
        onclick="helloWorld();"
    </p>

```

```
</body>
</html>
```

There are several things to notice about the previous code. To begin, we managed to connect the function `helloWorld` to our button by means of an event handler. This is the way most functions are executed in JavaScript, and it is one of the parts of the language that I most like. You define your functions in the `<script>` section and then indicate when they should fire using appropriate event handlers.

There is a variety of handlers, all of which begin with the letters "on", so you can execute a function when someone clicks (`onclick`), when something is changed (`onchange`), when an element gets the mouse focus (`onfocus`) or loses it (`onblur`), and a number of other possibilities. (Because we're using XHTML, all attributes must be in lowercase. So although it might be tempting to make the event handler more legible by writing `onClick`, that will invalidate the page.)

We can make this a bit more interesting by personalizing the message:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <script type="text/javascript">
        function helloWorld() {

            var username = document.getElementById("username").value;

            if (username == "") {
                alert("Please enter your name in the text field.");
            } else {
                alert("Hello, " + username + "!");
            }
        }
    </script>
    <title>Test JavaScript page</title>
</head>
<body>
    <p>Hi there!</p>

    <p>Enter your name: <input type="text" id="username" /></p>
    <p><input type="button" value="Click me!">
        onclick="helloWorld();"
    </p>
</body>
</html>
```

In this latest version, we added a text field (an `<input>` tag of type `text`) to the document. The text field has an `id` attribute, which uniquely identifies it on the page of HTML. Our JavaScript function, meanwhile, has gotten somewhat more complicated. We define a `username` variable, indicating with the `var` keyword that it is a local variable, rather than a global one. We then ask JavaScript to give us the value of the `username` element in the current document. Because `id` attributes are unique, we

know there will be at most one element on the page with an ID of username. However, we might misspell or otherwise make a mistake with the username. Or users might not yet have entered their name into the text field, thinking that they should click the button right away. Regardless, we need to test the value of the username variable, so that we won't foolishly say, "Hello, !" to the user. We thus use an if statement, which works similarly to if in many other C-like languages, putting up a warning note if someone tries to say "hello" without entering a username.

Modifying the Page

So far, we have seen that JavaScript makes it easy to read values from the page. But part of the magic of JavaScript, and the reason why it sits at the core of Ajax, is that we can modify the contents of the page almost as easily as we can read them. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
```

```
<script type="text/javascript">
function helloWorld() {

    var username = document.getElementById("username").value;

    if (username == "") {
        alert("Please enter your name in the text field.");
    }
    else
    {
        alert("Hello, " + username + "!");
    }

    var result = document.getElementById("result")
    var status_text =
        document.createTextNode("Thanks for clicking!");
    result.appendChild(status_text);
}

</script>
<title>Test JavaScript page</title>
</head>
<body>
<p>Hi there!</p>
```

Hurricane Electric Internet Services...Speed, Reliability, and 24/7 Support...You Can Finally Vacation Worry Free!

Flat Rate Gigabit Ethernet

1,000 Mbps of IP

\$13,000/month*

Full 100 Mbps Port

Full Duplex

\$2,000/month

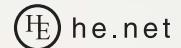
Colocation Full Cabinet

Holds up to 42 1U servers

\$400/month

Order Today!

email sales@he.net or call 510.580.4190



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

```
<p>Enter your name: <input type="text" id="username" /></p>

<p><input type="button" value="Click me!"  
onclick="helloWorld();" /></p>

<p id="result"></p>

</body>
</html>
```

The difference between this version and the previous one is in the else clause of the if statement. After displaying the "hello" alert box, we look for an element whose ID is result. This belongs to a new `<p>` element on the page, whose contents are currently empty, and which sits just below the button. Note the difference between our two invocations of `getElementById`. In the case of the `username`, we want the text that the user has entered, and thus invoke the `value` method on the returned node. In the case of `result`, we keep the node itself in the variable, without grabbing any value or other property. That's because we want to modify the `result` node, adding a text node containing our message.

Suddenly, the Web is a stable platform for all sorts of new applications.

And indeed, we create the new text node by invoking `document.createTextNode`, passing it an argument of the text we want to display. Creating the text node does not display it; in order for us to put it on the page, we must attach it to a node that is already being displayed. We will attach it to our `result` node, using the `appendChild` method.

Now, each time we click on the click me! button, we not only get an alert, but modify the contents of the page as well.

There is a problem with the above code, however. Because we used the `appendChild` method to add our text node, we end up creating and appending a new text node once for each click of the button. After clicking the button ten times, for example, we will see our "thanks for clicking" message displayed ten times on the page.

The easiest way to avoid this is to ask the `result` node if it has any children. If it does, we can assume we have already displayed our `result`. If no children exist, we safely can go ahead with adding the new text node:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<script type="text/javascript">
function helloWorld() {
    var username = document.getElementById("username").value;
    if (username == "") {
        alert("Please enter your name in the text field.");
    }
}
```

```
else
{
    alert("Hello, " + username + "!");
}

var result = document.getElementById("result")

if (result.childNodes.length > 0)
{
    alert("Already has children; not adding to the message");
}
else
{
    var status_text =
    document.createTextNode("Thanks for clicking!");
    result.appendChild(status_text);
}
}

</script>
<title>Test JavaScript page</title>
</head>
<body>
<p>Hi there!</p>

<p>Enter your name: <input type="text" id="username" /></p>

<p><input type="button" value="Click me!"  
onclick="helloWorld();" /></p>

<p id="result"></p>

</body>
</html>
```

The above, and final, version of our HTML page adds a new if statement after retrieving the `result` node. Just as we can add new nodes to `result`, we also can query it to find out if it already has children. When it is first downloaded to the user's browser, the `result` node has no children. It is empty. So, we can invoke `result.childNodes`, which returns a list of all children of the `result` node.

In this case, when we simply want to check to see if any children exist, we invoke the `length` method on the returned node list. If any number of children exist, we alert the user to the fact that we're not going to add to the message. We will modify the HTML page once, adding our "thanks for clicking" message, but if the user clicks again, JavaScript will detect the existing text node and not make the changes.

Conclusion

JavaScript sits at the heart of the shift toward Ajax applications on the Web, so it is an essential technology for Web developers to understand. JavaScript makes it possible to write functions that read and write the HTML returned by the server. These functions are then invoked by attaching them to predefined events. Next month, we will see how JavaScript allows us to work with stylesheets, which affect the design of a page. ■

Reuven M. Lerner, a longtime Web/database consultant, is a PhD candidate in Learning Sciences at Northwestern University in Evanston, Illinois. He currently lives with his wife and three children in Skokie, Illinois. You can read his Weblog at altneuland.lerner.co.il.



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

AoE is your answer!

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

www.coraid.com



EtherDrive® SR1520

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



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

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



1.706.548.7200

The Linux Storage People

www.coraid.com



MARCEL GAGNÉ

Operating Your Body at Peak Performance

High-performance systems and high-performance software are often easier to come by than high-performance users. Perhaps it's because we all work too hard. It's time to take a break, because peak performance, strangely enough, requires a little rest and relaxation.

François, *mon ami*, you look dreadful. How long have you been sitting in front of that computer? *Quoi?* You were in the restaurant six hours before it opened! Whatever for? You were trying to come up with something for today's High-Performance menu, and you became distracted. By what? Ah, I see, still trying to become the world *PlanetPenguin Racer* champion, are you? François, by now, even Tux needs a break, and he's just a character on a video game. Besides, how can you expect to perform at peak performance if you don't take a break? Plus, our guests will be here any moment, and you have a lot of work to do before they arrive.

Too late—our guests are here. Time to fetch the wine, François. Something Chilean for tonight, I think. Bring back the 2001 Casa Lapostolle Clos Apalta, and hurry. Or at least, as quickly as you can hurry, hunched over like that. And, do a few stretches before you come back. You must look your best for our guests.

Welcome, everyone! It is my exquisite pleasure to see you

all here at *Chez Marcel*, home of superb Linux software and spectacular wines. Please sit and make yourselves comfortable. I've sent François to fetch the wine, but I ask for your patience as he is moving a little slowly right now. François is a victim of sitting too long in front of his computer without taking a break. I wish I could say he was working, but he was playing video games.

My faithful waiter isn't the only one who suffers from this. All of us here spend far too long with our eyes fixed on the screen before us, occasionally moving the mouse, typing endlessly, clicking here and there. It sounds like we should all be accomplishing great things, but mostly we achieve tired eyes, sore muscles, carpal tunnel syndrome and worse. Luckily, we can use the same computer that puts us in this predicament to provide a solution, by using some great open-source software to force us to take a break. These handy little programs sit quietly in the background as you work. Then, at some regularly programmed

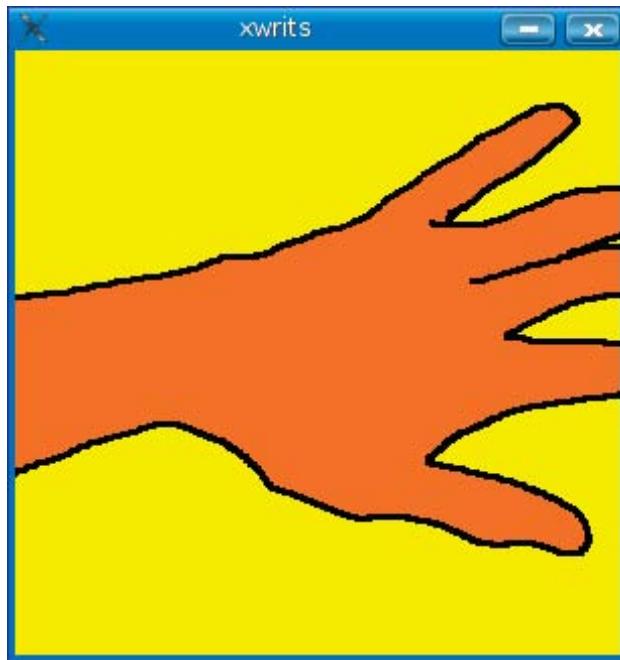


Figure 1. Xwrits displays a hand opening and closing "in pain" to alert you that it's time to take a break.



Figure 2. Click on the image to start your break. There, don't you feel better?

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

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

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



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

It sounds like we should all be accomplishing great things, but mostly we achieve tired eyes, sore muscles, carpal tunnel syndrome and worse.

intervals, they pop up a gentle reminder to take a break.

One such program has been around for quite some time. It's Eddie Kohler's Xwrits. You can pick up Xwrits from its Web site (see the on-line Resources). Optionally, check the contrib sites for your distribution of choice. When Xwrits runs, it monitors your keyboard activity and runs a timer in the background. When you have reached the maximum typing time, it pops up a warning window with brightly colored cartoon-like images of (presumably) Eddie's wrist flexing as though it needs a break.

To activate the countdown timer for your break, either stop what you are doing or click on the Xwrits window. The cartoon hand exclaims "Whew", and your break timer starts. When your break is done, another image appears giving you the thumbs up to continue. Simply click the window again and it vanishes.

The amount of time you can type before being forced to take a break is configurable. For instance, if you want to specify a time frame of 15 minutes before a break is forced, type:

```
xwrits finger top typetime=15 break=2
```

Let me explain those options. The `finger` option specifies that I will get a rude warning (the finger) telling me to take a break. When you add `top` to the options, the program tries to keep the Xwrits window on top of other windows you may have open (very useful). The `break=` option defines how long I must break before control is returned to the screen. Many other options exist that you

Figure 3. RSIBreak hides neatly out of the way in the KDE system tray. A right-click provides access to its functions.



should look at, including `lock`, which locks the keyboard, forcing the issue of whether you should take a break. Use the `breakclock` option to display a timer showing the amount of time left on your break. Xwrits even can monitor other workstations or sessions by using the `display` option. Make sure you check the included manual page or read it on the Xwrits Web site.

Ah, François, you finally have returned. Nice to have you back, *mon ami*. Please, pour for our guests.

Xwrits is a nice, light application, but those looking for a little more flavor may want to take a look at Tom Albers and Bram Schoenmakers' RSIBreak, available from its Web site (see Resources). RSIBreak is designed for KDE and integrates very nicely into the desktop. It is highly configurable, provides statistics on your compliance and can display a slideshow of pictures on your desktop. Choose wisely, and these pictures can be part of the relaxing experience that RSIBreak provides.

The program starts with a message directing you to the little clock icon on your system tray. This message pops up every time you start RSIBreak, but you can turn it off by checking the `Do not show this message again` box. A number of defaults already have been set for you, but most of them can be changed. Possibly the most important change will be the length and frequency of breaks that RSIBreak encourages. Tiny breaks are encouraged every ten minutes with a 20-second duration—basically just long enough to take your fingers away from the keyboard and rotate or massage your wrists before getting back into it. The big breaks occur every 60 minutes and last one minute. This is the change I tend to make. To make these changes, right-click on the RSIBreak system tray icon and select `Configure RSIBreak` (Figure 3).

The RSIBreak configuration window appears with a left-hand sidebar categorizing changes under General, Timings, Popup and Maximized. The General category is auto-selected, and there is one change you might want to make right here. Click the check box to have RSIBreak start every time you log in to your KDE desktop. Now, click Timings, and let us have a look at those default break times (Figure 4). The tiny break is fine for me, and a big break every hour seems like a good idea as well, but one minute just doesn't seem like enough. Let's change this to five minutes.

The final setting on this window defines how long slideshow images are displayed before moving on to the next image. To finish the slideshow setup, make sure you click on the Maximized icon. By default, your home folder is used to display images, but you can set an alternate location here. There's also a check box to allow the program to search the selected image folder recursively. At any given time, you can pause your mouse cursor over the system tray icon to get a report on the amount of time left before you are encouraged to take another break (Figure 5).

When you are satisfied with your settings, click OK to close the configuration dialog. Now, right-click on the system tray icon one more time, and you'll see an option labeled `Usage Statistics`. Click here for a report of your compliance with the suggested breaks. The report lists percentage of activity, number of breaks, time spent, breaks skipped and more.

While my faithful waiter, François, refills your glasses, let me direct your attention to the final item on tonight's menu, Rob Caelers' Workrave. This is another great helper for

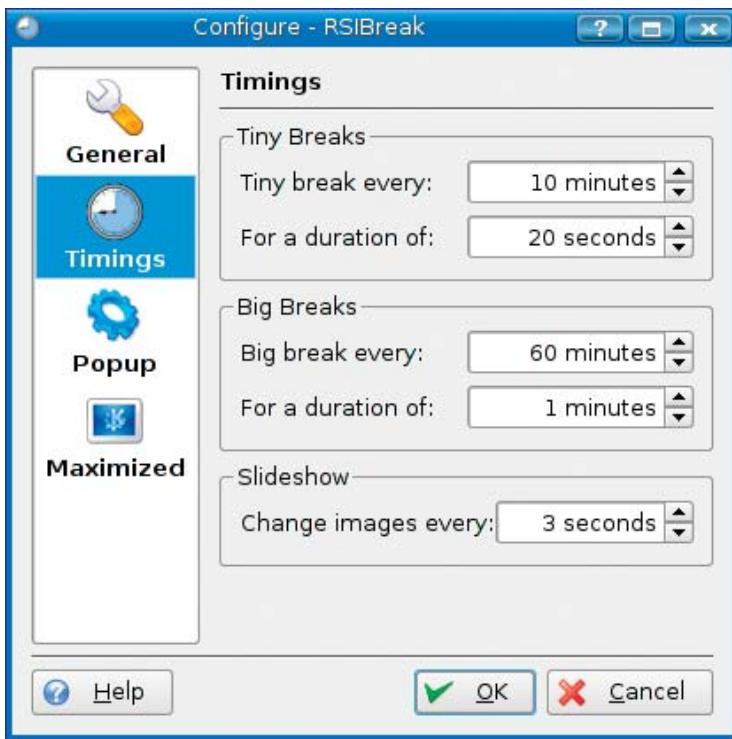


Figure 4. Short and long breaks in RSIBreak can be configured along with many other options.

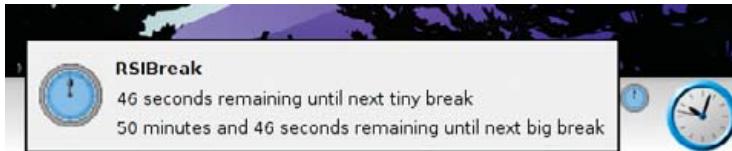


Figure 5. Pause your mouse cursor over RSIBreak's system tray icon to see how far away your next break is.

taking breaks, and you can get your copy at its Web site (see Resources). Like the other programs we've looked at, Workrave sits in the background, waiting to remind you that it's time to break. The length of break varies from micro breaks that last a few seconds to rest breaks lasting a few minutes, and finally to a message that it's time to log out and go home for the day. As you might expect, these breaks are configurable, and yes, it is possible to skip or postpone a break if you really, really, have to keep working. What sets Workrave apart from the others is that the program also suggests exercises you can do at your desk during the longer rest breaks (Figure 6).

To start using Workrave, run the program either from the command line (command name: `workrave`) or by pressing Alt-F2 and entering the program name in the run box. The first thing you will see is a small window with three icons on the left and bars on the right (Figure 7). The icons represent a micro break (the hand), a rest break (the coffee cup) and your time to leave (an open door). The bars are both graphical and numeric countdown timers for the various breaks. A panel applet provides a quick graphical display of the time left until your next break. Should you wish to bring back the small timer window, right-click on the applet, and select Open from the pop-up (or pop-down) menu.



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



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



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



Figure 6. Workrave suggests a number of exercises you may want to do during breaks.



Figure 7. The Default Workrave Status Window

Workrave works equally well regardless of whether you use KDE or GNOME for your desktop environment. KDE users will see a small sheep icon in their system tray. Pause your mouse cursor over the icon, and the countdown timers appear in a tooltip box. GNOME users will see one of the three bars from the timer window, cycling slowly through the various break types. When the time comes for a micro-break, a small pop-up window appears letting you know that a break is imminent. Just for fun, try clicking on the pop-up at this point, and you'll notice that it runs away from your mouse cursor. The idea is for you to take a break, and Workrave tries to make this as clear as possible. As with RSIBreak, the length of breaks is

totally configurable. I don't think of myself as a workaholic, but I do think that three minutes between micro-breaks is a tad low, and I immediately changed this one to ten minutes. To make those changes, right-click on the panel icon (or the Workrave window), and select Preferences. A three-tabbed window appears with three icons in the left-hand sidebar (Figure 8).

The first icon controls the timers for your micro-break and rest breaks as well as the daily limit. Aside from the time between breaks, you also can change the break duration and the postpone time. For the person who just can't take a break, there's even an option to turn off the Skip and Postpone buttons. Under the rest break tab, you also can decide how many exercises you will be shown.

Before I wrap up here, look back to the left-hand sidebar icons, and click on the User interface icon. Once again, this is a three-tabbed view with a number of options. I'm going to let you explore most of these, but I do want to point out one item under the General tab. There's a drop-down list

labeled Block mode. The default is to Block input. Change this option to Block input and screen, and not only is your keyboard locked, but the whole screen blanks out as well. That means no sneaky reading of e-mail or Web news while you are supposed to be taking a break.

If you are wondering how well you are doing with your new regimen of regular breaks, right-click on the Workrave applet (or status window) and select Statistics. Workrave keeps extensive statistics on your activity, the number of breaks you have taken and even keeps a day-to-day history.

None of the programs I've shown you here will do anything to hasten the arrival of the oft-predicted four-day work week, nor will any of them lead you to a life of leisure, but they might just be the solution for your neck pain, aching back and sore wrists. On that note, *mes amis*, it is time to take a much longer break. Closing time, it appears, is almost upon us. We don't want anyone leaving tense or tired, so take your fingers off those keyboards and relax. The best thing for your hands to be doing at this point is lifting a wineglass to your lips. On that note, I'm sure François is sufficiently well rested and will be more than happy to refill your glasses.

Please raise your glasses, *mes amis*, and let us all drink to one another's health. *A votre santé! Bon appétit!* ■

Resources for this article: www.linuxjournal.com/article/9140

Marcel Gagné is an award-winning writer living in Mississauga, Ontario. He is the author of the all-new *Moving to Ubuntu Linux*, his fifth book from Addison-Wesley. He also makes regular television appearances as Call for Help's Linux guy. Marcel is also a pilot, a past Top-40 disc jockey, writes science fiction and fantasy, and folds a mean Origami T-Rex. He can be reached via e-mail at mggagne@salmar.com. You can discover lots of other things (including great Wine links) from his Web site at www.marcelgagne.com.

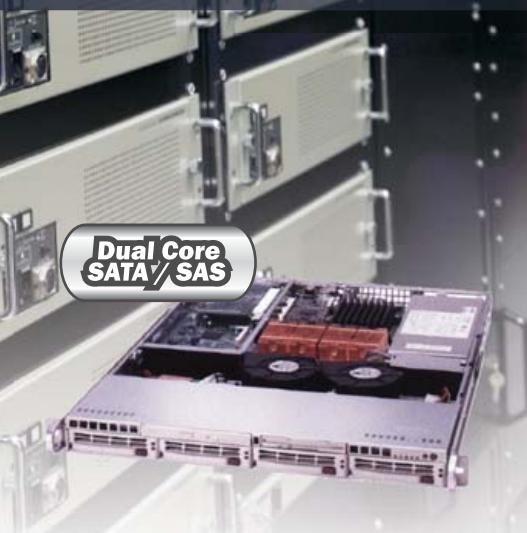
Figure 8. Workrave is extremely flexible. Make sure you take note of all the options.

Small business never stands still.

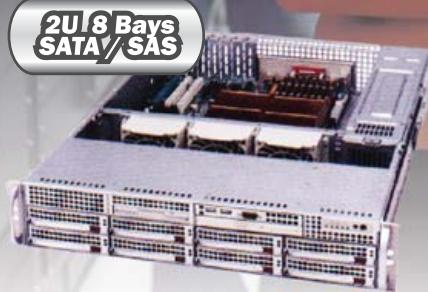
With the Dual-Core Intel® Xeon® Processor inside your ZT DC X9000 series Systems, your small business has the flexibility to grow and change. Great dual-core performance lets you run new applications and support more employees as business needs change. To keep your business on the move.



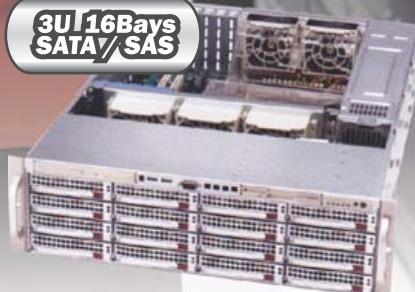
**Dual Core
SATA/SAS**



**2U 8 Bays
SATA/SAS**



**3U 16Bays
SATA/SAS**



ZT Dual Core 1U Server DC X9011

- 2 x Dual-Core Intel® Xeon® Processors 5050 (2x2MB L2 Cache, 3 GHz, 667MHz FSB, Intel® HT, EM64T)
- Intel® 5000V (Blackford VS) Chipset
- 1GB DD2 533 Fully Buffered DIMM(Upto 16GB)
- 4 x 250GB SATAII 16MB Cache Enterprise Hard Drive
- 4 x 1" Hot-swap SAS / SATA Drive Bays
- Slim DVD-ROM and Floppy Drive
- Intel® ESB2 (6)SATA 3.0Gbps Controller (RAID 0, 1, 5, 10 support)
- Intel® (ESB2/Gilgal) 82563EB Dual-port Gigabit Ethernet Controller
- 1U Rackmount Chassis w/ 520W Cold-Swap Power Supply
- SuperDoctor III Server Management Software
- 3-Year Limited Warranty + 1-Year On-site Service

\$1,999

- Highest quality server for smooth-running applications
- Enterprise storage solutions for system reliability and stability
- Flexibility from SATA to SAS drives

ZT Dual Core 2U Server DC X9012

- Dual-Core Intel® Xeon® Processor 5050 (2x2MB L2 Cache, 3 GHz 667MHz FSB, Intel® HT, EM64T)
- Intel® 5000V (Blackford VS) Chipset
- 1GB DD2 533 Fully Buffered DIMM(Upto 16GB)
- 4 x 250GB SATAII 16MB Cache Enterprise Hard Drive
- 8 x 1" Hot-swap SAS / SATA Drive Bays
- Slim DVD-ROM and Floppy Drive
- Intel® ESB2 (6)SATA 3.0Gbps Controller (RAID 0, 1, 5, 10 support)
- Intel® (ESB2/Gilgal) 82563EB Dual-port Gigabit Ethernet Controller
- 2U Rackmount Chassis w/ 700W Redundant Power Supply
- SuperDoctor III Server Management Software
- 3-Year Limited Warranty + 1-Year On-site Service

\$2,299

ZT Dual Core 3U Server DC X9013

- 2 x Dual-Core Intel® Xeon® Processors 5060 (2x2MB L2 Cache, 3.20 GHz, 1066MHz FSB, Intel® EM64T)
- Intel® 5000V (Blackford VS) Chipset
- 2GB DD2 533 Fully Buffered DIMM(Upto 16GB)
- 4 x Seagate® 500GB SATAII 16MB Cache Enterprise Hard Drive
- 16 x 1" Hot-swappable SAS / SATA Drive Bays
- Slim DVD-ROM and Floppy Drive
- Intel® ESB2 (6)SATA 3.0Gbps Controller (RAID 0, 1, 5, 10 support)
- Intel® (ESB2/Gilgal) 82563EB Dual-port Gigabit Ethernet Controller
- 3U Rackmount Server Chassis w/ 800W Redundant Power Supply
- SuperDoctor III Server Management Software
- 3-Year Limited Warranty + 1-Year On-site Service

\$3,899

Go to

ztgroup.com/go/linuxjournal

Call

866-ZTGROUP (866-984-7687)

Promote code : Lj0906

Purchaser is responsible for all freight costs on all returns of merchandise. Full credit will not be given for incomplete or damaged returns. Absolutely no refunds for merchandise returned after 30 days. All prices and configurations are subject to change without notice and obligation. Opened software is non-refundable. All returns have to be accompanied with an RMA number and must be in re-sellable condition including all original packaging. System's picture may include some equipments and/or accessories, which are not standard features. Not responsible for errors in typography and/or photography. All rights reserved. All brands and product names, trademarks or registered trademarks are property of their respective companies. Celeron, Celeron Inside, Centrino, Centrino Logo, Core Inside, Intel, Intel Logo, Intel Core, Intel Inside, Intel Inside Logo, Intel Inside Logo, Intel SpeedStep, Intel Viiv, Itanium, Itanium Inside, Pentium, Pentium Inside, Xeon and Xeon Inside are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.





DAVE TAYLOR

When Is “Good Enough” Good Enough?

Shell scripting and obsessive-compulsive perfectionism is almost a contradiction in terms.

Last month marked the end of my series about writing a *Blackjack* game as a shell script, and I don't know about you, but I had a good time with the development process and have even learned a bit more about the game itself. I received a number of fun e-mail messages from readers about the column, but I also received one that was most thought provoking.

The author criticized me for using less-than-optimal algorithms for things like my shuffle routine, for using poor scripting style and generally questioned how much I really knew about shell script programming in the first place.

Another Day, Another Flame

Having lived on the Internet for almost 30 years now, I'm quite familiar with flames and hostile e-mail, with people nit-picking, focusing on the molecules of the leaf without ever even knowing there's a forest ahead, but this message still got me thinking about the practice of scripting and of programming in general.

To quote a colleague of mine, Ken McCarthy, what's a better strategy, imperfect action or perfect inaction?

Although the Bourne Again Shell is remarkably capable and certainly has all the basic programmatic structures of more sophisticated programming languages, I think it's nonetheless

until it's 3am two weeks after your deadline. That's perfect inaction, right? The zeal to wait and wait and tweak and prod until it really is perfect, by which time you've missed your deadlines and goals.

So, when I received the criticism from this reader that I hadn't chosen the best possible algorithms and wasn't using what he thought was an optimal scripting technique, I was glad to see how he thought things should be done. But I also was unsurprised, as one of the greatest challenges I believe facing software developers is learning that in many cases and situations “good enough” really is, well, good enough.

Writing a Shell Script Isn't Developing Software

Am I advocating that the next time you're writing the firmware for the in-flight controller on the Boeing 777 you should cut corners, skip testing and write crummy code so you can ship on time? Of course not. But you know what? If you're writing a testing framework cron script that will simply log the start of the test, invoke a series of MySQL queries and log the end of the test, well, yeah, in that case, relatively crummy code, code that works well enough, might just be exactly what's required.

Am I advocating that the next time you're writing the firmware for the in-flight controller on the Boeing 777 you should cut corners, skip testing and write crummy code so you can ship on time?

fair to say that it's a lightweight, even throw-away programming environment. You don't write large, complex or mission-critical applications as shell scripts, do you?

That's how I have always approached shell script writing—basically as a fast prototyping environment. You want to know how many lines are in a file? Use something like:

```
lines=$(wc -l < $filename)
```

Is that the most elegant and efficient solution? Probably not. Indeed, if you're doing that to test whether the file has nonzero content, you should be using the `test` command instead, but here's my point: it doesn't really matter.

That's what I mean by imperfect action. It's far, far better to get going with your script, to build a sloppy prototype, to get it done, than to tune, clean up, tweak, rewrite and optimize

I see this same perfectionist attitude with letters I occasionally receive from people who have bought my *Wicked Cool Shell Scripts* book. They haven't realized that writing a shell script is inherently an exercise in rough prototyping (hence the absence of sophisticated shell script development environments and testing frameworks), not a programming world where perfect code is the digital holy grail.

The “Not Good Enough” Developer Crowd

I also think that these rather snobbish software developers who worship elegance and dismiss software developed by people who are still stumbling their way through programming are doing a significant disservice to the world of computing.

Don't get me wrong, I appreciate an ingenious algorithm and snappy implementation when I read code, but most

of the real innovations in software and applications come from the bubble gum and bailing wire crowd anyway, from the rough prototypes and the "barely beta" software that works well enough to demonstrate concepts and get the community to start experimenting anyway.

So I Should Be Writing Sloppy Code?

Some of you doubtless are a bit confused by this point in my column—puzzled that someone who is supposed to be teaching you tips and tricks of shell script programming is actually advocating sloppy coding and quick "throw-away" scripts. I ask instead why you're surprised at the idea that getting something out the door, solving the problem quickly and reasonably well, is often better than wasting—yes, wasting—time writing "the perfect script"?

This reminds me of a computer programming course I took many years ago at UC San Diego. Our challenge was to figure out the optimal sorting algorithm for a given situation and write a program that implemented it. Just about everyone in the class thrashed about, but I pulled out Knuth's *Art of Computer Programming*, picked out the algorithm he recommended, and typed it in, with an appropriate citation. I was penalized for "cheating" and had to argue adamantly that there was, in fact, no better way to learn how to choose the best sorting algorithm than to refer to the definitive work on the subject. Finally, the professor relented and gave me full credit.

Shell script programming is the same: shortcuts are always a good thing, efficiency is a measure of how fast you can solve a problem, and although tuning and tweaking can be rewarding as an intellectual exercise, 90% of the time it just doesn't matter at the end of the day.

Think about that. And ask yourself how you're working toward imperfect action rather than being trapped trying to achieve perfect inaction.

Next month, we'll go back to the nuts and bolts of shell scripting. But, please, don't expect me to write perfect little scripts or use the absolute best algorithm in the world for a given task. Indeed, sometimes my code will be inefficient, will spawn more subshells or child processes than entirely necessary, or might even have unnecessary loops and conditionals. Maybe, just maybe, that's okay? ■

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.

www.faircom.com/go/?track

Your packages get

tracked & delivered

on time by FedEx.

FairCom database
technology makes
it possible.

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



FairCom®



MICK BAUER

How to Worry about Linux Security

Worrying is good when you worry about the right things and act accordingly.

Although most of us manage to muddle through without major breaches occurring terribly frequently, running a networked Linux system is nevertheless a worrisome undertaking. Well, in my opinion, worrying is good! Worrying, if it's the constructive kind that holds complacency at bay, keeps us on our toes. The trick is to worry about the *right things* and to act on our worries.

Uh-oh, you may be thinking, Mick's setting aside his pocket-protector this month in favor of the soapbox. Guilty as charged. I'm convinced that my technical articles will be much more useful to you if you periodically think about the larger context into which security technologies fit. So this month in the Paranoid Penguin, it's time to discuss what to worry about in Linux security and what to do about it.

Folks You Should Worry About

When you're thinking about defense, it's good to work your way outward, but because the bad guys are usually looking at you from the opposite direction, it's instructive to consider their viewpoint now and then. Therefore, I begin by talking about who we need to worry about attacking our systems. Next, I discuss their tools (attack-vectors) and finish by examining common system vulnerabilities and ways we can mitigate them.

So who are these attackers? They follow a spectrum from the unskilled, often reckless punks, to careful, highly skilled professionals. I don't even pretend to know the full spectrum myself—these aren't the circles I move in—but I can describe some common types:

- Identity thieves harvest user name/password combinations, credit-card and bank account numbers, e-commerce credentials and other information they can either sell to other crooks or use themselves to perpetrate various kinds of fraud.
- Resource thieves are less interested in your data than your computing resources (your computer or the network to which it's attached), which they want to use to send spam, commit distributed denial-of-service (DDoS) attacks, mask the origin of their attacks on other systems, trade pirated software (warez) or trade pornography. You'd be surprised just how many different ways it might be useful for people to make their network activities appear to originate from your system!
- Malicious code is both a tool and an attacker. Although many worms, trojans and viruses are used to conduct specific types of attacks, others exist for their own sake (to make mischief) and operate autonomously. We need to worry, therefore, both about resource thieves and identity thieves who use malicious code in semitargeted ways and also

about self-propagating malicious code.

■ Vandals want to deface your Web site, disrupt your network traffic and generally inconvenience you. In most cases, it's more precise to say that they want to impress their friends at your expense, though some cyber-vandals do in fact choose their targets carefully (for example, so-called hacktivists, who deface the Web sites of groups or governments they feel are evil).

■ Corporate spies want to steal your organization's proprietary data: the lab notes of your pharmaceutical researchers, the merger and acquisition plans of your board of directors and so forth. This is a much more focused type of attacker than the identity thief or resource thief, and not all of us really need to worry about corporate spies, but believe me, they do exist, and they do cause financial loss.

■ Stalkers want you to see only how very, very much they love you, even though the true depth and beauty of their passion can't be understood by any humorless judge or meddling psychiatrist. (Sorry, I'm being sarcastic, but I do detest stalkers!)

Everyone needs to worry about identity thieves, resource thieves and vandals, because these attackers tend to be highly indiscriminate in choosing their targets. The old security cliché, "I don't have anything anyone would want to steal", may hold up when discussing corporate spies, but falls completely flat here. In fact, resource thieves in particular *prefer* low-profile, innocuous-looking targets, because avoiding attention is such an important part of their operations. With all three of these groups (identity thieves, resource thieves and vandals), the attacker simply couldn't care less what you use your computer for.

Note that identity thieves generally don't care about your computer at all. They care about your identity, and to get at it, they're more likely to try to hack *you*—for example, by tricking you into entering your Internet banking credentials at an impostor Web site—than to hack your actual computer. (Although, it's only a matter of time before someone writes a worm that looks for identity information stored on its victims' hard drives. More on malicious code later.)

Also, not all attackers are remote. Corporate spies and identity thieves, in particular, who are frequently "insiders" at the organizations they attack, often have or seek local/console access to the systems they attack. (It's a lot easier to get at the data on a hard drive you've just stolen than to hack your way through firewalls, intrusion detection systems, application access controls and so on.) Resource thieves and vandals, however, are usually remote attackers.



T20DC-1U

**Custom
Chassis**



Tested with Dual Core Opterons



Fan Assembly for Maximum Performance



1U Density for Scalability



Easy Access to I/O and Power

**Socket F
Opteron™
Tested!**

Highlights

- Custom 1U Density Design
- Dual-Core Opteron Tested
- "Socket F" ready
- Top rated Thermals
- Fits standard 19" Rack
- Power & Reset buttons
- Power & HDD LED
- Front USB
- 4x Front 40mm Fans
- 6x Middle 40mm Fans
- 2x Rear 40mm Fans
- Supports SATA, SAS and SCSI HDD
- Includes 1x 5.25" Bay
- Support for PCI-Express, HTX and PCI-X Risers
- 500W Power Supply
- Convenient Front Handles
- Slide Rails Included

The 1st InfiniPath
Interconnect Cluster



UC-Davis – based on
the Dual-core Opteron™

TeamHPC
A division of
M&A Technology, Inc.
1-866-TEAMHPC

www.teamhpc.com

Weapons You Should Worry About

So, how do these various attackers achieve their nefarious goals? What, in other words, are their weapons?

When I first became a network engineer in the mid-1990s, the attack paradigm we usually worried about was a human being sitting at a computer, interacting more or less directly with her "victim" systems in real time. In other words, Matthew Broderick's character in the movie *WarGames* was the sort of attacker we assumed (rightly or wrongly) to be most common.

Today, however, it's safe to say that the vast majority of probes and attacks conducted against networked systems are carried out by automated software processes—that is, by viruses, trojans and worms. People are still behind these types of attacks; make no mistake about it—someone has to write, adapt and deploy all that malicious code—but most actual attacks happen by proxy nowadays.

For example, one of the fastest-growing tools of the resource-theft trades (spamming, porn-peddling, DDoS and so on) is the botnet. A bot is a computer that has been infected with a worm or virus that surreptitiously contacts the person who released it; a botnet is an entire network of bots awaiting instruction.

Botnets are part of a strange, complicated and shady economy. Botnet operators who distribute spam under the pay of spammers are commonly paid per distribution node. Most of the merchants who pay them intentionally turn a blind eye to the fact that these nodes are probably not legitimate e-mail servers, but illegally hijacked systems (that is, bots). When spam-botnet operators are caught, the actual source of both their income and their spam invariably claims they had no idea their spam was being distributed by illegally compromised systems.

Resource thieves aren't the only ones who use botnets. DDoSers use them to conduct highly distributed network bombardments that are difficult both to trace and stop. Identity thieves often carry out phishing attacks, in which spam e-mail purporting to be from a bank or e-commerce site is used to lure people into entering their logon credentials at an impostor Web site. Phishers, even more than garden-variety pharmaceutical and porn spammers, have a strong motivation to hide their tracks, so botnets are especially useful for phishing-spam distribution.

The fully interactive system attacker is still very much with us; not all attackers cast as wide a net as spammers or phishers. Corporate spies, vandals and stalkers are all likely to make use of one-on-one attacks in which attackers focus their attention on one system, and conduct their attacks more or less in real time. Some of these attackers, especially in the corporate-espionage space, are highly skilled and creative experts who are able to crack even carefully secured systems, often by writing customized attack software.

Conventional wisdom, however, is that many if not most Web site defacers and other vandal types are script kiddies—less-skilled attackers who rely on tools they download from the Internet or obtain from friends. Such attackers are much more easily thwarted than the pros, because they tend to be not nearly as adaptable. If the attack scripts they run against a given system fail, they're far more likely to give up and seek a softer target than they are to fine-tune their scripts or write a new script altogether. And, a given script may work only against one version of one application running on one particular architecture (for example, Apache 2.0.1 on Intel x86 platforms).

In summary, the good news is that most attacks are indiscriminate, automated and not very adaptable. Highly focused, human-operated and creative attacks are much less common. The bad news is that the sheer volume and variety of automated

attacks (including spam, phishing, malicious code and script-kiddies' tools) makes them a force to be reckoned with. These attacks cost people and organizations everywhere millions of dollars annually in lost productivity and fraudulent transactions. Furthermore, just because skilled/human attackers may not seem like a likely threat in a given scenario doesn't mean you can disregard them altogether.

Vulnerabilities You Should Worry About

So, there are attackers, and these attackers have tools. What are the nuts and bolts that these tools manipulate?

Consider this simple formula from threat-modeling parlance: a threat equals an attacker plus some vulnerability. A vulnerability is some characteristic of the attacker's target that presents an opening. What the threat equation tells us is that if a given vulnerability can't be exploited by an attacker (for example, because the system isn't networked and resides in a locked room), it doesn't constitute a risk. Conversely, a system with no vulnerabilities isn't at risk regardless of how many attackers target it.

Obviously enough, there's no such thing as a completely invulnerable system. There are, however, many ways to deal with vulnerabilities that decrease their possibility of being exploited by attackers.

Common types of vulnerabilities include:

- Bugs in user-space software (applications).
- Bugs in system software (kernel, drivers/modules and so forth).
- Default (insecure) application or system settings.
- Extraneous user accounts.
- Extraneous software (with bugs or sloppy/default settings).
- Unused security features in applications.
- Unused security features in the operating system.
- Gullible users.

The remedies to these tend to be straightforward. Bugs can be patched, default/insecure settings can be changed, extraneous accounts and applications can be removed, security features can be leveraged and users can be educated. "Straightforward" doesn't necessarily mean "easy" (or "quick" or "cheap"), however.

The patch rat race, as I've said many times in this space, is futile—you can't write a patch without first discovering the bug, and what are the odds of the good guys discovering every major bug before the bad guys do? Still, we're stuck with this cycle; patch we must.

The outlook for tightening system and application settings, leveraging application/OS security features, educating users and applying additional security techniques and tools is considerably brighter.

How to Channel the Ph33r

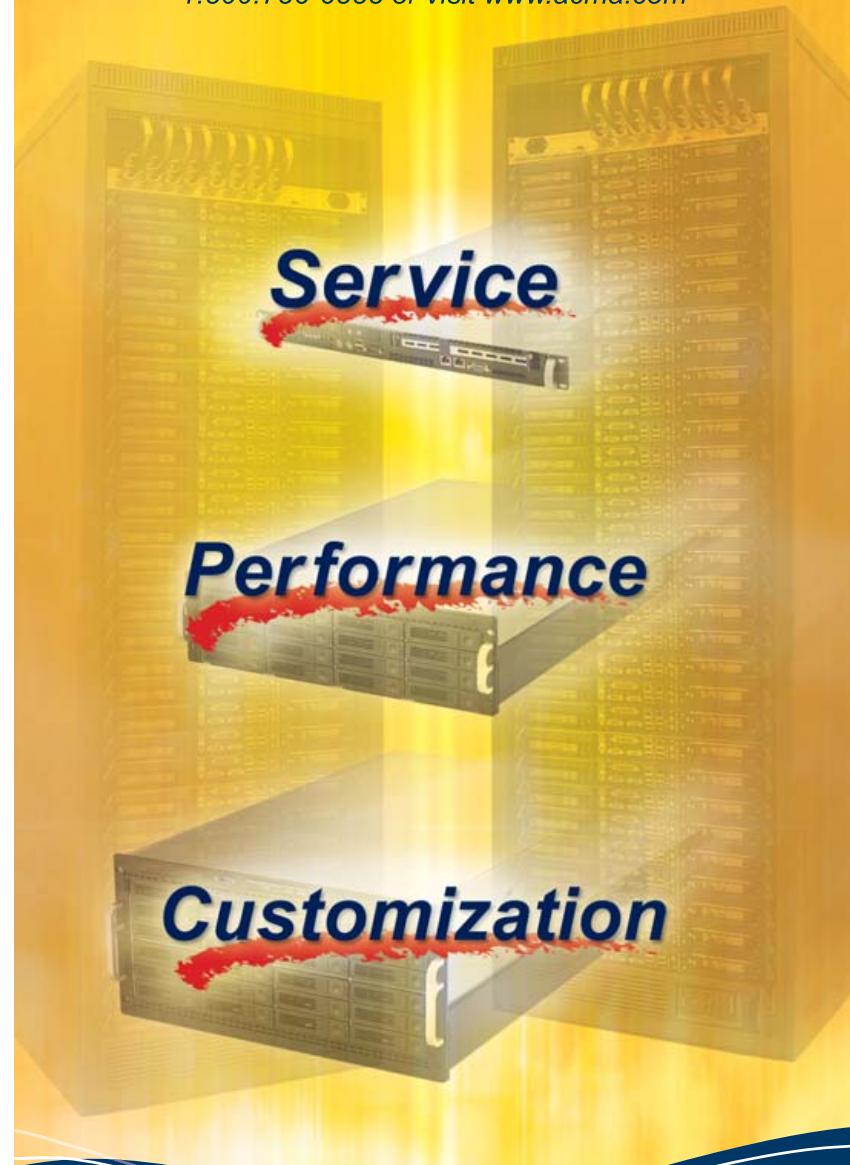
This brings us to the positive, forward-looking part of my little editorial: how to channel all these worries into constructive action. I've already begun talking about security techniques and tools, and I'm not going to go much further with those

Why ADD Value, When You Can Multiply?

ACMA is a leading provider of custom manufacturing solutions. With world-class engineering, ISO 9001 certified manufacturing, nationwide logistics, and 24/7 technical support, ACMA will not just ADD value to your business.

ACMA will multiply your business success!

Call now for a personal consultation
1.800.786-6888 or visit www.acma.com



right now—that's what my technical articles are for. Rather, I'd like to conclude here with my recipe for converting worry into action.

Step one in securing any computer system or network is to define its function. What will it do? What data will it store, process and serve to users? Who are the intended users? How will those users interact with the system? As any fan of Sun Tzu can see, this is none other than the ancient technique of analyzing the terrain you need to defend.

Step two is to identify, in general terms, the types of attackers you need to defend against. Start with all possible, but prioritize the most likely.

Step three is to think of what data or resources these attackers are most likely to attack on your system, and what sorts of vulnerabilities they might try to exploit in conducting those attacks. Once you've considered the most likely attacks, be sure also to consider less likely but still-plausible attacks, particularly those that are easy to mitigate.

Let's walk through this process with an example scenario. Suppose I've just built a Debian-based Web server. In step one, I determine that the server's function will be to host a simple "brochureware" Web page containing information about my accordion shop (hours of operation, address and so on). The Web server will have only a handful of local user accounts. Most access will be by anonymous Web clients. No sensitive data will be housed on it. No other network services (SMTP, DNS, IRC and so on) will be provided by this particular box, although I will need to administer it via Secure Shell (SSH).

In step two, I decide that because there will be no nonpublic data on the server, it will not be a likely target of corporate spies. I've got a small operation, and in fact, I'm friendly with my town's only other accordion dealer. She's highly unlikely to hire a professional system cracker to hack my Web site. My main worry, therefore, will be less-discriminate attackers: resource thieves, malicious code in general and vandals. It's also possible, though, that some more focused type of attacker may attack my system in order to use it to attack other systems, so I won't focus exclusively on automated attack vectors.

In step three, I decide that my system itself is the most likely attack target, and that automated attacks are the likeliest vector, and that software bugs and configuration settings are the likeliest vulnerabilities to be targeted. What I appear to have before me is a very generic system-hardening exercise. I need to apply all current software/system patches (maybe including running apt-get as a cron job), identify and remove extraneous software and user accounts, configure Apache as securely as possible and at least consider using some add-on tools like Tripwire, Snort and syslog-ng to help monitor my system's security status.

I may decide not to bother running Apache in a chroot jail. The whole purpose of this box is to run Apache, so if someone compromises Apache, the system is for all intents and purposes hosed. On the other hand, if the attacker manages to escalate an Apache compromise into a full system (root) compromise, he or she may have a much easier time using my Web server to attack other systems, so maybe a chroot jail (for example, using mod_security) would be worth the extra hassle after all.

Conclusion

The point of this exercise is that I've thought in an organized way about my defense goals, about the attackers that worry me most and about how best to prioritize the time and effort I spend on securing my system. Sure, I may get some or even all of it wrong, but such a mistake is apt to be much less calamitous than applying security controls in a haphazard or reactive manner.

Besides, knowing me, I'm going to worry regardless. Isn't it better to worry in an organized, constructive fashion? ■

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".

Acma[®]
We Manufacture For You!

Toll Free: 800.786.6888 WWW.ACMA.COM

© 2006 Acma Computers Inc, 1565 Reliance Way, Fremont, CA 94539. All rights reserved.



DEE-ANN LEBLANC

S2 Games

Savage 2 will ship on Linux despite the company's decision to go with DirectX as its primary graphics interface.

S2 Games (www.s2games.com) is the independent game company that made a huge splash in the summer of 2003 with the release of *Savage: The Battle for Newerth*. It wasn't the graphics or the engine that particularly made this game notable. It was the merging of two distinct genres—real-time strategy games, such as *Warcraft*, with first-person shooters, such as *Quake*—into what S2 Games calls a “real-time strategy shooter”. This is not a Massively Multiplayer Online Role-Playing Game (MMORPG), but instead it is run like a team-based first-person shooter, where a group runs its own server instance, and then all of the players join that instance.

Savage Game Play

In some ways, it's easier to talk about the *Savage* games in terms of other games with which you may be more familiar. There is a thorough breakdown of the game's general rhythm available at savage2.s2games.com/gp_overview.php, so I won't repeat that here. First, *Savage* is nothing new with having two teams (the humans versus the “beasts”) trying to destroy each other on a dedicated server. It's also not new to have the option of playing combat characters or support characters. For example, *Tribes* allows you to take on the role of a repair engineer to keep the equipment in repair. However, in *Tribes* and most other first-person shooter games, all of the characters spawn as fairly identical units—it's the equipment that allows for specialization. Some, such as the Team Fortress mod for *Quake*, did allow people to select a character class, such as engineer, that remains throughout the game.

In the *Savage* games, one player takes on the role of the Commander, and then there are three additional classes: builder/conjurer, scout/shape shifter and savage/summoner. (The name of your class depends on which side you have taken in the battle.) Other first-person shooters that allow creation of Commanders have used them only for strategic purposes, such as coordinating the action players. Building, however, is new to first-person shooters, which brings in a feature from the real-time strategy side.

Another difference is that one phase of the game involves sending your action players out to hunt NPCs for both experience and gold, much as you would in an MMORPG. Typically, there are no NPCs in such team-oriented first-person shooter games. While the players hunt, the Commander researches technology and builds weapons, siege engines and more, which is another real-time strategy and MMORPG-style feature. Through the remaining early phases of the game, things continue to work in a real-time strategy mode as costs for building upkeep come into play. So, gold mines and more need to be owned and maintained to help keep everything in good shape.

Although first-person shooters are almost entirely modern day or science fiction, the *Savage* games are held in a fantasy setting. In many team-oriented first-person shooters, to gain access to more powerful equipment, groups often need to

complete some kind of task, such as capturing a certain number of flags or destroying something valuable to the enemy, such as its generator. In *Savage* games, you have to create possessed/Hellbourne pits in order to gain certain powerful fighters. Once this goal is achieved, along with some others, the actual combat between the two teams commences. Soon after this point, the Commander switches from building and researching to tactical command, directing the forces. Both siege and battle units are important for winning. Perhaps the closest parallel to this type of play is the Battlegrounds in *World of Warcraft*.

With this level of complexity, the *Savage* games go far beyond merely “shooting each other dead”.

The Method behind the Madness

In the Linux world, S2 Games made another splash as well by offering not only a server for Linux users, but also a native client. At the time, it seemed a no-brainer for the company, because its Silverback game engine is written for OpenGL (www.opengl.org), the cross-platform graphics standard typically used by those who want to develop for more than one operating system simultaneously. An advantage of hosting its servers on Linux is that the operating system has so much less overhead than Windows that it is possible to host multiple server instances on the same Linux machine without losing performance. This is not the case with Windows.

The Linux community responded with enthusiasm, accounting for approximately 15% (7,500 units) of the sales for *Savage: The Battle for Newerth* according to Mark DeForest, lead designer and cofounder of S2 Games. Due to the strong Linux community support and the fact that S2 Games' collective personalities find a kinship with some of the Linux community's attitudes, when it came time to build *Savage 2: A Tortured Soul*, the company wanted to continue providing a Linux client. However, the decision was made to use DirectX for the next engine, K2, because DirectX is still easier for programmers to work with, and there is far more middleware (software that game developers can purchase instead of having to write every tiny piece themselves) available for DirectX than for OpenGL. DirectX is also easier to tie into Microsoft Windows and the related APIs (programming interfaces allowing programs to talk to one another).

Rather than abandoning Linux, S2 Games decided to go ahead and make an OpenGL version of *Savage 2* as well. As S2 Games wanted to make a lower-end renderer for those who have older video cards, the company folded this functionality into the OpenGL engine as well. Doing so means that both Linux and Windows users have the option of playing in both high-end and low-end graphics settings, keeping those with lower-end hardware able to play the game. Ultimately, including support for Linux adds more expense, frustration in terms of getting things working, project management and support



Figure 1. Characters and Combat in *Savage 1*



Figure 2. A Character in *Savage 2*

efforts and costs, which makes the fact that none of the S2 Games staff are actually Linux users particularly interesting.

If *Savage 1* broke ground and created a new genre, *Savage 2* refines it. DeForest says that the game-play mechanics will not change. There will be new role-playing game (RPG) elements added, but otherwise, most differences are going to be a matter of polish. *Savage 2* will simply "look and feel better", as the company takes the game to the next level. For *Savage 1*, S2 Games didn't have time to polish the product fully. This time, the developers promised themselves that *Savage 2* truly would be a step above, rather than resting on their laurels, as no one else has seriously entered the genre they created. If you look at the pictures from both games, you will see just how far they have come (see Figures 1 and 2 to compare). Many more images are available on the S2 Games site.

Beta testing for *Savage 2: A Tortured Soul* begins July 31, 2006 for both Windows and Linux, and the game is scheduled for release in November 2006, as of this writing. This game is available for pre-order on-line, for those who are interested, and the Linux client launches at the same time as the Windows client. ■

Dee-Ann LeBlanc (dee-ann.blog-city.com) is an award-winning technical writer and journalist specializing in Linux and miniature huskies. She welcomes comments sent to dee@renaisssoft.com.



ASA COMPUTERS

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

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

ASA Low Budget Low Voltage-Starting @ \$1025
1 of 2 Intel Xeon LV 1.67GHZ Dual Core
1 GB DDR2 PC2-3200 ECC REG
1 of 160 GB SATA Drive
Dual Gigabit LAN, IPMI Card, CD



ASA 14" Deep Appliance Server-Starting @ \$995
Intel Xeon LV 1.67GHZ Dual Core
1 GB DDR2 PC2-3200 ECC REG
40GB SATA Drive, One GigE
CD, FD, Second HD, Your Logo



ASA 1U 4 Drives SCSI Server- Starting @ \$1491
1 of 2 Intel Xeon LV 1.67GHZ Dual Core
1 GB DDR2 PC2-3200
1 of 4 36GB SCSI Drive
Dual Gigabit LAN, IPMI Card, CD



ASA 6 Hot Swap SAS/SATA/Dual Xeon Dual Core
2 Intel Xeon 2.0GHZ Dual Core
Supports up to 16GB DDR2 ECC REG
Supports SAS/SATA
CD, FD, IPMI Card, 2 Gigabit Ethernet
Call for Pricing & Customization



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



ASA Collocation \$75 per month for 1U Rack - 325 GB/month	ASA Collocation Special First month of collocation free.*
Storage Solutions IDE, SCSI, Fiber Raid solutions NAS, DAS, iSCSI, SATA, SAS 3Ware, Promise, Adaptec JMR, Kingston/Storcase solutions	
Clusters Rackmount and Desktop nodes HP, Intel, 3Com, Cisco switches KVM or Cyclades Terminal Server APC or Generic racks	

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



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



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

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



JON "MADDOD" HALL

Pirates and Pollywogs

Exploring proprietary software and piracy.

"maddog, maddog", came the voices. Of course, I knew what this meant; it was the weekly meeting of the Pollywogs, a group of young friends who wanted to learn more about the history of computers. They came to the small restaurant where I had dinner on Saturday evenings, the *Aldeia dos Piratas*, to ask questions and receive the best answers that I could give.

"Boa noite", I said (being most of the Portuguese I know), "what can I do for you tonight?" "Tell us about pirates", they answered in unison. "Pirates?" I questioned, "this seems a little off subject...you normally are interested in computer information, why this interest in pirates?" "Software pirates", they answered, and now I saw the connection.

"What most people call software piracy is a complex issue", I said. I continued:

In the earliest days of computers, both hardware and software were incredibly expensive. The people who could afford the hardware of the computers also usually could afford the cost of the software. Likewise, software was typically created as a service for the customer. Manufacturers supplied the operating system software for their own computer systems, so their customers could use their hardware, and applications were written specifically for a particular customer as a service. When customers "bought" software, usually the license did not specify how many computers they could run it on, and customers "owned" the software they had purchased.

As time went on and computers became manufactured in larger quantities, the price of hardware dropped considerably, but software remained expensive. So, people thought about manufacturing software the same way they manufactured hardware. This software typically would be delivered only in binary form, at a fraction of the cost, but you would not own the software, just the right to use it under terms of a license. The creator of the software still "owned" it. This is what most people are used to in today's market.

Although this was adequate for a lot of the world's market, it had some artificial limitations. Depending on the license, restrictions were made as to whether the software could be re-sold or re-used. Although older hardware might be re-sold to people who could not afford to buy new hardware, the old software could not be re-sold. Ironically, repopulating an old computer system with fresh copies of the software it originally came with might cost more than buying a new computer with the software already "bundled".

André gave me a knowing look. He told the group

about his grandmother who was given a used computer, but it had no software on it. The person giving it to her had stripped the software off, to be "legal". The software alone would have cost her \$2,000 US to replace, just to do simple word processing and some manipulation of her digital camera pictures. His grandmother was now a "software pirate".

I continued:

As the prices of the hardware dropped, the prices of the commercial software necessary to "do business on the Internet" stayed the same or even went up. While new computer systems with high-speed CPUs, lots of memory and disk space dropped in price to hundreds of US dollars, the software to run them stayed the same or rose in price to thousands of dollars.

It is interesting to note that commerce has moved in time from viewing the computer as a luxury in business to a necessity in business. It is difficult, if not impossible, for a business to exist without using computers and/or being tied to the Internet. Yet for people whose monthly income was measured in tens of dollars, this meant they could not afford the computers they were told they needed to do business today.

Enter software piracy. When caught between the proverbial rock and the hard place, people started to copy software and distribute it to people who needed it. People realized it did not cost \$600 US to make a shiny plastic disk and that they could get the equivalent disk at their local CD store for \$2 US. I mentioned that one time at a conference and a voice from the back of the room said, "One US Dollar!", a tough negotiator.

Nevertheless, this copying and selling of the software is illegal, and it is against the wishes expressed in the license as well as (in most cases) against the wishes of the software creator.

"It is as wrong as stealing a bicycle or a car, or photocopying a book for re-sale. It teaches a disrespect for the law that filters down to other issues", said Pablo. Pablo is studying to be a doctor, but is also active in his Software Libre! club. He went on to say, "In addition, stolen software does not typically allow for the thief to go back to the creator of the software to ask for patches, extinctions or training. And there is no pride in stealing software."

"Why don't companies do more to protect their software?" asked one of the members of the group.

"Ironically, companies who scream the loudest about software piracy often admit to encouraging it", I said, and continued:

A South American product manager of a large software company based in Redmond, Washington, once admitted to me that they "help out" people that they know have pirated their software, providing patches and information to them. "It keeps these people from using alternatives", he told me. Yet, in the same country, this company is an active member of the Business Software Alliance, which often audits companies to see whether they are using unlicensed software and oversees the legal actions against them.

The group agreed that this seemed more than a little hypocritical. Cesar told us that this same Redmond-based company often provides "free" distribution for "Education" or "Digital Divide" reasons. The latest edition of this software can run only three applications at a time, and other similar restrictions makes their software even more useless. "The motivation behind this is so transparent, it is enough to make a person laugh or cry. They hope to keep the buyer of the computer from trying a truly free operating system and software", he said.

Pablo pointed out that free software cannot be stolen. "You cannot steal something that is licensed to you freely. There may be a request with the free software that you acknowledge the person or company that wrote it, but this is typically a low- or no-cost request, easily done."

Finally, I said:

Free (free as in freedom) software, allows people with little money to help contribute to the software in many ways. They can contribute to the development, to the documentation, to the testing or to the promotion of the software. They even can sell support and training, making a living off of it, instead of a drain. They do not have to worry about the Business Software Alliance. This means that people who have little money still can contribute to the advancement of the software they use. They can join the software community with pride, not as a handout, and not as a thief.

Although you think I may be talking only about so-called third-world countries, how many businesses anywhere have "enough" money?

Marlon told us about his uncle who needed several pieces of very expensive software, available only from the United States and available only in English to run his business. The software would have cost him more than \$2 million US, but more important, it would have forced him to teach his employees enough English to use the software. Instead, his uncle found a free software programmer who duplicated exactly what the uncle's business needed using MySQL, Perl, FreeGIS, Apache and other free software products. His uncle paid the software programmer money to create the software needed, and the software worked in Portuguese, not English. The money also stayed locally, buying local food, local housing and paying local taxes.

"Enough!", I said, "Enough talk about software pirates and software piracy. If you do not start your football game now, it will be too dark!"

And while the younger Pollywogs went off to play their game on the sand, the older ones and I put on our jackets, watching the sun drop down behind the trees and the stars come out over the water.■

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



GENSTOR™

Company With a Vision

Great Minds Great Solutions

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

Customized Solutions for

SERVERS :: STORAGE :: APPLIANCES

Linux - FreeBSD - x86 Solaris - MS etc.

SERVERS

LOW POWER – BUDGET – HIGH DENSITY

(1U, 2U, 3U and above - Xeon, P4, Opteron, VIA)



2U Dual Xeon/Opteron Server
Upto 24GB RAM

Upt o 8 SATA/SCSI HDD

Starting @ \$ 2000.00

Call for your custom needs

SERVER MANAGEMENT & MORE

Genstor offers Data Center-proven Linux Management Technology - which gets you :



Provisioning
Deployment
Change Management for Linux Environments large and small
Track Changes
Instant Rollback and much more

Setup your racks with heterogeneous Linux environments in minutes
Contact sales@genstor.com

STORAGE

SATA- NAS-DAS-iSCSI -SAS Storage Solutions



5U Dual Xeon/
Opteron SATA Storage
Upto 24GB RAM
Upto 26 SATA HDD
Upto 18TB Storage
Starting @ \$ 3930.00
Call for your custom needs

CUSTOM APPLIANCE SOLUTIONS

Prototype – Certifications - Logo Screen Printing



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

FACING POWER PROBLEMS AT DATA CENTER PL. CALL FOR SOLUTIONS

Contact Genstor For Your Hardware Needs

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

GENSTOR SYSTEMS, INC.

780 Montague Exp. # 604, San Jose, CA 95131

Phone: 1-877-25 SERVER or 1-408-383-0120

Email: sales@genstor.com

Fax: 1-408-383-0121

Prices, Products and Availability subject to change without notice.



DOC SEARLS

An Interview with J.P. Rangaswami

The world's best suit rolls up new sleeves.



J.P. Rangaswami won't be defending his title as CIO of the Year, because that's no longer his job at Dresdner Kleinwort Wasserstein, one of the world's leading investment banks. But he still works there—harder than ever, from what I could tell when I ran into him at the latest Reboot conference in Copenhagen. (He lives in the UK, when he isn't traveling the world, looking for ways to reconcile the leading edge with sturdy infrastructure.)

At Reboot, J.P. gave a knockout keynote address, in which he volleyed back a number of ideas I have also been lobbing over the Net and here in *Linux Journal*. As always, he enlarged those ideas considerably, while adding many new ones of his own. Here are a few:

People are not assembly-line things. What makes us different is that we are different. And to try to create standardized norms of us is where evil starts stepping into organizations. It's building hospitals where you centralize ill people. Very clever, right? Why would people do that? You know many people die of exposure to illness while they are ill? The last place you want to spend any time is in a central hospital. That place is a production of industrial thought, not human thought....

This capacity to bond was denied us because of the divide-and-conquer garbage that we were exposed to, probably for the last four or five hundred years—and made a lot more sophisticated through the any-color-you-want-as-long-as-it's-black assembly-line-production-factory approach to humanity. And it doesn't scale. It doesn't work....

If you go to many Eastern countries, *identity* is not about who I am, because the I is very unusual. Identity is about *what I belong to*. Identity is a statement of the community that I belong to. Identity is a statement of things that I will associate with me, that empower me as a result of belonging to something....

There are many things you can use identity for. But if you understand that even a passport, three hundred or

four hundred years ago, used to say, "I know this person. Will you please look out for him." Not, "These are his permissions and these are the things he can do and this is what he needs a visa for." None of that stuff. It was elective to say, "Since I know you and I know this person, look after this guy. He's a good guy." It was a trusted domain-type statement. And then also to say, "If he needs anything, if he's ill, if he needs some food, if he needs shelter, help him out." That's what the passport used to be. Look what it's become. And we've allowed this to happen. We've allowed things to be taken away from us, rather than saying, "Let's build new things." I only want us to get back what we had....

The first death is device lock-in....Can you imagine me telling people, "You cannot use your own pen"? Or, "You can't use your own voice."

Afterwards, I thought it would be fun to continue the volley in the form of an interview. Here it is.

DOC: Are (or were) there any other CIOs like you? Meaning, ones that embrace open source, DIY-IT, re-usability and disruptions you know are going to be more constructive and stable in the long run than the technologies being disrupted?

JP: I think there are many others, but I had some advantages. One, given our size and shape, innovation was an imperative rather than a nice-to-have. Two, we have some very talented people working in IT at DrKW. Three, we had management commitment and buy-in, which also goes a long way. So, it was easier for me to do this than for many of my peers. Other CIOs that come to mind are Al-Noor Ramji at BT (who, incidentally, hired me at DrKW!), Tom Sanzone at Credit Suisse and David Yu at Betfair. David is now the CEO at Betfair, which is going some.

DOC: You are widely regarded as the best CIO in the world (and have won awards for that). Yet you moved on to another job at DrKW. How come? And what does that say for the CIO job in general?

JP: Well, I'd done five long years, and I needed a fresh challenge. My lifeblood is in innovation, so it made sense for me to join our Digital Markets operation—a bunch of very tech-literate and tech-friendly people, who have done wonders in actually using Web 2.0 tools rather than talking about them.

DOC: What is your new job and what is it about?

JP: I'm the Head of Alternative Market Models in our Digital Markets unit. Simply put, my role is to take our learning in Web 2.0 from the enterprise to our customer base. There's a high involvement in technology as a result, so I can continue to work with my erstwhile colleagues. And it's refreshing.

DOC: What are less-recognized but important trends you see happening with (or coming from) open source? Or with Linux in particular, since we center our focus on that platform?

JP: I see three things against a common background.

First, the background. People have finally come to grips with the fact that open source is about commoditized infrastructure and not about communism. In fact, open source is more secure than proprietary architectures, and that really helps us take advantage of Moore and Metcalfe effects. It helps us get away from the traditional full-of-holes and insecure arguments and gives us a sound business basis for doing it.

One is a shift in scope and breadth: Linux is about ecosystems and not vertical stacks, which changes the way we manage our portfolio.

Two is a shift in time: beta now means live. We have learned to bring iteration into our processes—something that we needed to discover was the value of democratized innovation.

Three is a shift in workforce location: adoption of Linux and open-source principles really helps us work better with a geographically dispersed workforce; Linux is ubiquitous, it engenders faster and cheaper localization and is easier to support on a global basis than any proprietary architecture.

By the way, I'm also intrigued by where NAND RAM is going, Robson and so forth, and watching very carefully what that means for us in years to come.

DOC: You've talked about technologies or practices that you consider dead or terminal. DRM, for example. Device lock-in. Yet, major vendors such as Apple, in spite of their embracing of open source, remain very committed to both DRM and device lock-in. The iPod, while a brilliant technological and marketing achievement, may now be the world's largest and most successful silo, combining both DRM and device lock-in. Is there any hope that these companies will get clues from leaders on the customer side like yourself?

JP: As people recognize the sheer cost and inefficiency of DRM and device lock-in, I think this will change. Right now, everyone's used to levels of lock-in at the device, but as I said at Reboot, things are changing. Generation M will resist and the market will drive vendor behavior. Generation M is predisposed against lock-in, unlike our generation. And things will change. Like the phone example. First, you rent a black phone from the phone company; then you buy a color phone, but only from the phone company; then you buy any phone from anywhere. If iPod and iTunes want to emulate Polaroid, that's their call, but I have faith that they will see the light. And it won't take 100 years like with the phone companies. It will take maybe three years.

When generation M goes to work, those people will have their pens and their computers and their phones and their cameras. Not company-issued ones. And we have a job to do to pave the way for them, which is where identity and authentication and permissioning and walled-garden arguments begin and end.

DOC: I've talked and written about the "because of" principle, which says you make more money because of infrastructure (such as the Net, Linux and the LAMP stack) than you do with infrastructure. I discovered that principle by following open-source development, by the way. You recently responded on your blog (confusedofcalcutta.com/2006/06/09/four-pillars-because-of-rather%0a-than-with-a-very-provisional-post), framing the "because of" principle in terms of chronology. For example, you say companies shift over time from "with" to "because" strategies. So here are two observations I'd like to get your thoughts on.

First, there are lots of "because" companies that seem unaware of that fact. The infrastructural code they use is just cheap or free building material. The big Web-based companies, from Amazon to Google to Yahoo to Akamai, are "because of" companies. They make money because of Linux and Apache, not with those things.

JP: Yes. "Because of" companies tend to avoid understanding what they are largely as a result of misconceptions: "Because" doesn't have buzz, "because" doesn't have innovation and so on. "Because" has a bad name. In the UK we have a phrase: "Where there's muck there's brass", which roughly translates to, "There's money to be made in cleaning things up, in providing basic services."

Open source is all about commodity and infrastructure, so we understand that. I think you will find that Warren Buffett invests primarily in "Because" companies. Someone should do research on the split between "because" and "with" in Berkshire Hathaway's portfolio.

Follow the money. "Because" companies will make money because "with" companies need them. "Because" is actually lower risk, smooths out kinks, creates annuity returns. "Because" is safe. It just takes time for us to understand that.

As you say, many of today's technology leaders would not exist if it weren't for "because" companies. I don't see Linux or Apache screaming like the telcos for their share of the dough; nor do I see FedEx or Wal-Mart scream. There is good money in "because".

I also think that there is no such thing as a "because" vacuum. If no one invests in a "because" space, and people are scared of the upfront investment, then the "because" community will innovate to find a new way. That's what happened to fixed-line and mobile in the developing world.

This year, even before he took up his new job at DrKW, J.P. took up blogging. You'll find his posts—often long, always thoughtful—at confusedofcalcutta.com.■

Doc Searls is Senior Editor of *Linux Journal*.

Is your network healthy ??

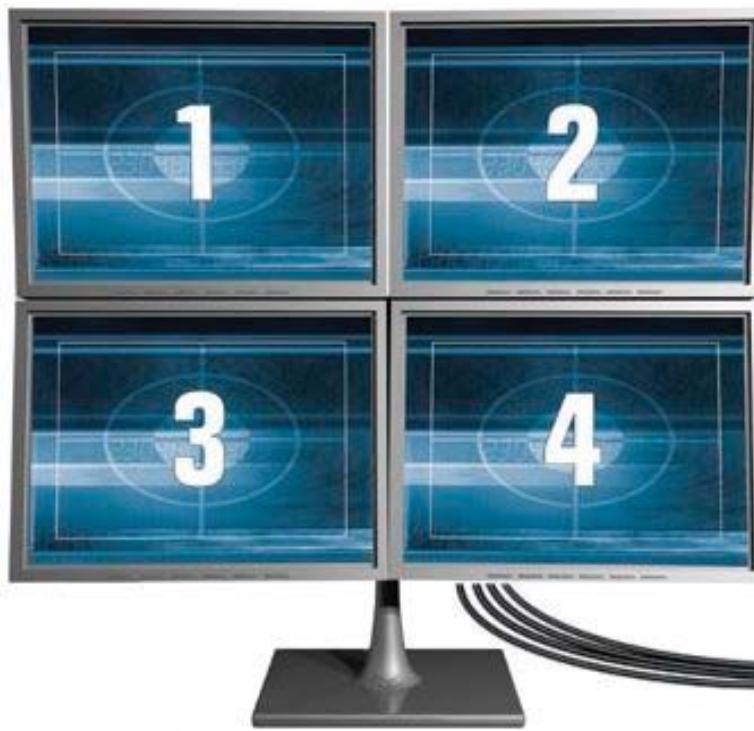
High-performance, open-source solutions to measure and monitor network vital signs.



From \$4999



www.npulsenetworks.com

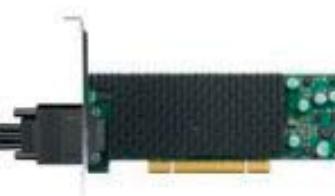


Matrox Graphics' EpicA Graphics Cards

The team at Matrox Graphics recently released its EpicA series of dual- and quad-display PCI graphics cards intended for thin clients and other mission-critical systems. According to Matrox, the products offer "innovative, new, server-based software...to manage multi-display configurations in remote sessions".

Supported protocols include Linux desktop remote connection software, Microsoft RDP and Citrix ICA for Windows. Other features include passive cooling, small form factor, support for digital and analog monitors, low power consumption and independent and "stretched" modes. The TC-2 and TC-2 Lite models support two monitors concurrently, while the TC-4 model supports four.

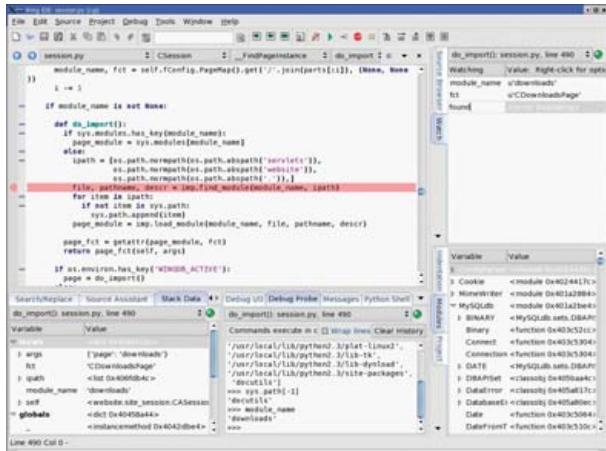
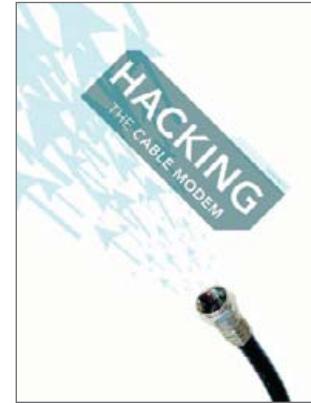
www.matrox.com



DerEngel's Hacking the Cable Modem (No Starch Press)

No Starch Press is a publisher with a penchant for finding geeky niches that other publishers won't touch, and DerEngel's *Hacking the Cable Modem* is a fine case in point. This book "reveals secrets of many popular cable modems, including products from Motorola, RCA, WebSTAR, D-Link and more", sayeth the No Starchers. It is also a guide to hacking a cable modem, installing new firmware, unblocking ports and unlocking hidden features. One net benefit of these efforts, besides pure hacking enjoyment of course, is an increase in bandwidth up to 20-fold. In addition, who wouldn't be curious to know what the author, whose alias is DerEngel ("the angel" in German) and has been tagged as "the underground Prometheus of super-broadband", has lurking in his brain. The product will be on real and virtual bookshelves in August 2006.

www.nostarch.com

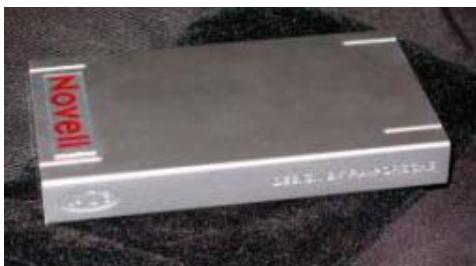


Wingware's WingIDE 2.1

Wingware is now shipping release 2.1 of WingIDE, its development environment for Guido van Rossum's masterpiece, the Python programming language. WingIDE's purpose is to provide "powerful debugging, editing, code intelligence and search capabilities that reduce development and debugging time, cut down on coding errors, and make it easier to understand and navigate Python code". New features in version 2.1 include Visual Studio, VI/Vim and brief key bindings, Subversion and Perforce support, improved Windows performance, named bookmarks, breakpoint manager and call stack as list, file evaluation or selection in the Python Shell and support for Macs on Intel. Supported platforms include recent Intel Linux systems, Windows 2000 and later and OS X 10.3.9 or later (with X11 installed). Solaris, *BSD and other Posix platforms are supported for those willing to compile from source code. A free trial is available on Wingware's Web site.

www.wingware.com

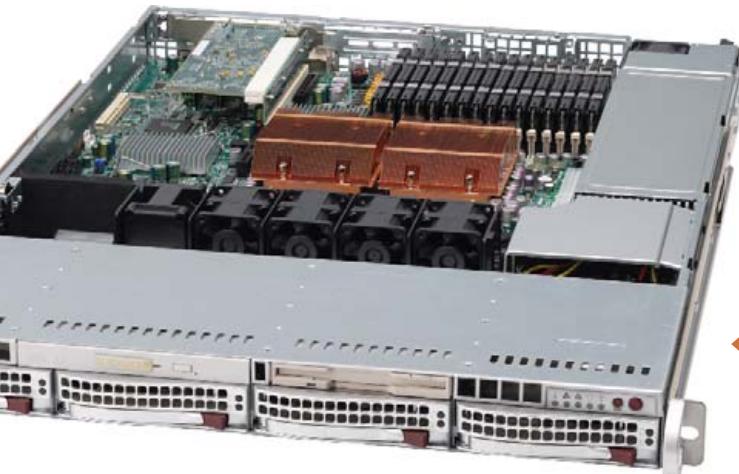




Novell's SUSE Linux Enterprise 10 First Class Course

First there was toast on a stick (vintage David Letterman reference), and now Novell offers training on a stick too! SUSE Linux Enterprise 10 First Class is a new, self-contained course on a USB drive that allows users to test the new server and desktop products on their own. Included on the multiplatform (Linux, Windows) device are both the learning content and an installed example of the server and desktop on a virtual machine. Novell states that this approach "gives the student a unique environment to study the lecture and then gain hands-on experience using the virtual machine to do exercises". Students can utilize the course themselves or use it as part of an instructor-led two-day course at various training centers. The course/device hybrid is available for purchase at Novell's Web site.

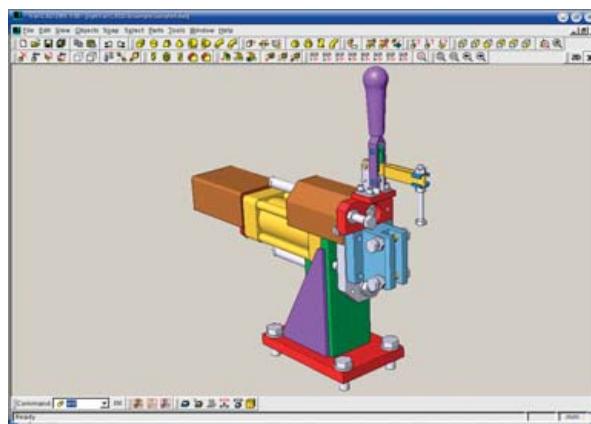
www.shopnovell.com



VariCAD 2005 3.0

VariCAD has significantly rebuilt its eponymous 3-D/2-D mechanical CAD system—which is no stranger to the Linux platform—with release 3.0. At its heart, says VariCAD, the multiplatform (Linux, Windows) product is a fully loaded package that offers not only "powerful tools for 3-D modeling and 2-D drafting and dimensioning", but also "libraries of standard mechanical parts (ANSI, DIN), calculations of standard mechanical components and tools for working with bills of materials and blocks". Innovations in the new version include "improvements in the STEP interface allowing input and output of 3-D objects, new high-resolution bitmap output from 3-D, new user-defined default settings, improved file opening and dimensioning" and others. A Linux-specific improvement is reduced dependence on system files, allowing the software to run on more distros. A free trial version (Debian or RPM) is available for download from VariCAD's Web site.

www.varicad.com



Supermicro Computer's X7 Series Server Platforms

Like other vendors in the diverse hardware arena, Supermicro has upgraded its product line to take advantage of Intel's new Dual-Core Xeon 5100 series processors, aka Woodcrest. The new platform is slated to improve system performance and memory capacity, as well as reduce energy consumption and operating temperatures. Supermicro has leveraged Intel's advances to improve its own systems, or SuperServers, that are based on its X7 series of motherboards. Supermicro claims that the combination of its high-efficiency power supplies and the Woodcrest processors result in "5% or greater efficiency than competitors' systems", providing energy savings of "up to \$200 per server over three years".

www.supermicro.com

Arcom's SBC-GX533 Development Kit

Arcom recently released its SBC-GX533 Development Kit for developing embedded devices in a Linux environment. The target applications, says Arcom, are "deeply embedded, remote or unattended installations demanding reasonable processing power", typically industrial RTUs and data acquisition modules, as well as networking and communications devices. The SBC-GX533 is a low-profile, fanless, RoHS-compliant, EBX-form-factor board with a 400MHz AMD Geode GX533 1.1W processor, 512MB of DDR DRAM and 32MB of Flash, of which only 13MB are occupied by the preloaded Linux image. The Linux OS is kernel 2.6 with a Compressed Journaled Flash File System (JFFS2) for reliability and recovery from power interruptions. One of the SBC-GX533's key advantages is the preconfiguration of the Linux image, which preempts the need to build it from scratch. A few of the optional features are TFT, analog touch-screen and Java Technology.

www.arcom.com



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

SOFTWARE

CrossOver Office 5.0

Those who need to run legacy Windows applications on Linux should consider CrossOver Office 5.0. JES HALL

CodeWeavers' CrossOver Office 5.0 is a commercial application based on Wine that allows you to run many popular Microsoft Windows-based office and productivity applications under Linux, as well as a few multimedia and Internet applications. For those who are tied to Microsoft Office in particular, it can provide a means to migrate to Linux.

Supported applications include:

- Acrobat Reader 5
- FrameMaker 7.1
- Photoshop 6 and 7
- DreamWeaver MX
- Endnote 8
- Flash MX
- iTunes
- Lotus Notes 5 and 6.5.1
- Internet Explorer 6 Service Pack1
- MS Office 97/2000/XP/2003
- MS Project
- MS Visio
- QuickBooks
- Quicken
- QuickTime
- Various MS Office document viewers

For people researching CrossOver Office with a mind to deploying the product in their workplace or as a single-user home installation, the CodeWeavers' Web site is an excellent resource (www.codeweavers.com). Clear and well-written information is grouped in a logical manner. Case studies illustrate how effectively the product copes in real-world situations, and questions about Microsoft licensing are addressed.

CodeWeavers offers a time-limited demonstration version of its products. For 30 days,

you can try the full capabilities of the product without being limited in functionality. This is an excellent way to let customers test the compatibility and performance of the product for themselves before committing to purchasing a license.

CodeWeavers' CrossOver Office comes in three main flavours. The Standard Edition, which we tested here, is a standalone application licensed for a single user with limited support and upgrade period. If any other users on the machine want to use CodeWeavers' CrossOver Office, they need to purchase their own copy. The Standard Edition is available only as an 11MB download.

CrossOver Office Professional is also standalone, with support for multiple local users. It can be purchased on CD as well as downloaded, and it includes 12 months of a higher level of support. Bulk and educational discounts are available.

CrossOver Office Server Edition provides a centrally managed way of distributing Windows productivity applications to Linux thin clients, with a premium level of support designed for large-scale deployments.

Installation

CrossOver Office is distributed as a large bourne shell script. On invoking the script, a graphical installer is launched that takes you through the install and configuration process. We found the installer to be simple to use. Once the installer was complete, CrossOver Office then offered us a dialog through which to install various Microsoft Windows software. The installation process automatically created entries in our KDE and GNOME menus, a



Figure 1. CrossOver Office Install Software Dialog

CrossOver submenu with access to various CrossOver Office tools and a Windows Applications menu for our Windows application launchers once installed.

The installation process was quick, painless and easy to follow. We give CrossOver Office a thumbs up for installation.

Ease of Use

One installed, we immediately tested installing an application from the Internet by selecting the Internet Explorer browser. CrossOver Office gave us excellent feedback, letting us know what it was doing at each stage of the process. The familiar Windows installer dialog for Internet Explorer was launched, and after clicking through the wizard, we were returned to CrossOver Office while it simulated a Windows reboot. Once this was complete, an Internet Explorer launcher could be found in the KDE menu under Windows Applications, making finding and executing the newly installed program a breeze.

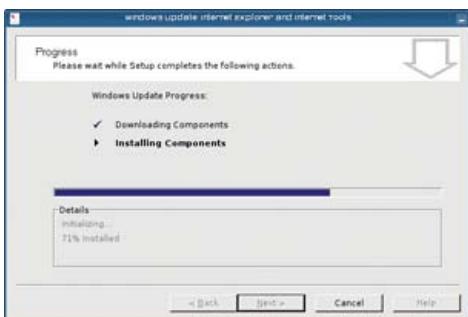


Figure 2. Installing Internet Explorer

We tested installing software from CD using Microsoft Office 2000. The application installer managed to detect the correct CD-ROM drive and install from it.

Documentation

The user guide for CrossOver Office is distributed as HTML files on disk suitable for off-line reading. The documentation is clear and easy to read, with helpful screenshots. It covers an exhaustive range of issues from installing CrossOver Office and Windows applications, to more advanced topics, such as the configuration of Windows emulation options.

The troubleshooting section was a little confusing, with a long cluttered list of frequently asked questions. The FAQ does cover a wide range of questions, and we recommend reading it for handy tips. We give CrossOver Office a thumbs up for documentation.

Compatibility

CrossOver Office installed and ran without a hitch on SUSE, Ubuntu and Slackware Linux. The list of supported distributions

THE PENGUIN

Skip the distal prototype phase and get your designs off the ground faster. With Penguin, it's plane and simple. Penguin Computing® Clusters combine the economy of Linux with the ease of Scyld. Unique, centrally-managed Scyld ClusterWare HPC™ makes large pools

of Linux servers act like a single virtual system. So you get supercomputer power, manageability and scalability, without the supercomputer price. So upgrade your design cycle. Let the simulations fly. And whatever you do, don't eat the fish.

FLIES FIRST



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

PLEASE VISIT US AT SC06 AT BOOTH 435,
Tampa Convention Center, November 14-16, 2006.
www.penguincomputing.com



REALLINUX

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

HOW WE TESTED

We tested by installing CrossOver Office on three machines. Waste is a Pentium 4 1.7GHz with 512MB memory running Ubuntu, and Holly is a Pentium III 700MHz with 256MB of memory running SUSE. To check how CrossOver Office installs on unsupported distributions, we also tried running various applications on Hal, a 2.4GHz with 1,024MB of memory and Slackware Linux.

We installed Internet Explorer, Apple's iTunes music player, Adobe Photoshop 7.0 and Microsoft Office 2000 on both machines. To test the stability of these applications, we performed a range of fairly standard tasks with them—playing MP3s from disk and streamed from music shares on a Macintosh with iTunes, touching up photographs with Adobe Photoshop, using Internet Explorer for Webmail and on-line banking and opening and editing various Word and Excel files with MS Office.

currently includes:

- Red Hat 9
- Fedora Core 4
- Red Hat Enterprise Linux Workstation 3 and 4
- SUSE 10.0
- Novell Linux Desktop
- Mandriva 10.x
- Debian Stable
- Ubuntu 5.04
- Ubuntu 5.10

Given the reasonable range of modern distributions supported and that CrossOver Office worked very well even on unsupported distributions we tried, we give CrossOver Office a thumbs up for compatibility.

SUMMARY

INSTALLATION: 

EASE OF USE: 

DOCUMENTATION: 

COMPATIBILITY: 

CAPABILITY: 

PRICE: 

FINAL SCORE: **Very Good**

CrossOver Office 5.0 is a great solution for those who must run some popular legacy Windows applications under Linux. It has some glitches, and you need good hardware to get the best out of the product, but it does the job it is designed to do quite well.

Support

Support for the standalone editions is available in two levels. CrossOver Office Standard includes six months of Level 3 support. This level of support includes fixes for installation issues and only the most grievous problems with supported applications on tested distributions: "...you should not expect a rapid or complete response to any but the most serious problems."

CrossOver Office Professional includes 12 months of Level 2 support. This level of support promises problem resolution for any problem found within any supported application. Any Level 2 problem will be triaged and logged. Problems encountered on non-tested distributions will be considered.

The archives of the support ticketing system used by all levels of support can be found on the CodeWeavers' Web site and can be accessed by unregistered users, allowing all users of the software to search the database to find solutions to their problems. For lesser support levels, CodeWeavers provides a medium-volume mailing list and an IRC channel.

We went to the IRC channel to find support for various minor problems we encountered. The person who helped us was courteous and friendly and very well informed. We give CrossOver Office a thumbs up for support.

Capability

Microsoft Office, Photoshop and IE performed well on Waste and tolerably on Holly. Given Waste's modest specifications, as compared to even the cheapest of new machines today, we think this is pretty reasonable. Even running natively under Windows, Photoshop 7 isn't terribly fast on a Pentium III. The applications were not completely stable and crashed a few times during testing. All applications suffered from occasional rendering glitches.

iTunes and QuickTime required considerably more CPU power, and Holly's 700MHz was simply not up to the task. Audio crackled and was jerky, and video completely refused to play. Even on Waste, iTunes was unbearably slow to use and respond to user interaction. No amount of coaxing could convince it to play music smoothly. We tried various suggestions from support without success.

Due to some issues with stability and drastic problems with multimedia over



Figure 3. Adobe Photoshop Running under CrossOver Office

multiple distributions, we give CrossOver Office a flat hand for capability.

Price

At \$39.95 US, CrossOver Standard is relatively inexpensive—certainly cheaper than purchasing a licensed copy of Windows. Most professional users would consider \$69.95 US to be a reasonable price to pay for the higher level of support. We give CrossOver Office a thumbs up for price.

Overall Rating

In summary, CodeWeavers CrossOver Office is a promising product, and we give it a final score of very good. We recommend strongly that you run CrossOver Office on a decent machine with some good processing power and lots of RAM. On anything less, CrossOver Office would score satisfactory at best. With the right hardware though, it easily could have earned a score of excellent if it weren't for the fact that it has minor stability problems and difficulty getting multimedia applications working properly. This brings down the overall score of a product that has excellent documentation, support and an easy-to-use interface. Hopefully, future versions of CrossOver Office will address these points, making it a more viable professional product for hardware that is less than the latest and greatest. ■

Jes Hall is a KDE developer from New Zealand who is passionate about helping open-source software bring life-changing information and tools to those who would otherwise not have them. She welcomes comments sent to jhall@kde.org.

Accelerate your competitive advantage

Choose ServersDirect Systems powered by the innovative technology of the 64-bit Intel® Xeon® Processor with dual-processor functionality. Enjoy excellent performance and headroom for today's 32-bit applications. And protect your investments as you transition to 64-bit computing.

SDR-1500T

Designed for optimal performance while supporting maximum frequency dual-core Xeon® 5000/5100 sequence processors in a high-density 1U form-factor



\$2,499

2U Entry Level Server

SDR-2500T

Optimize performance with dual-core Xeon® 5000/5100 sequence processors and FB-DIMM memory



- Intel® 64-bit Xeon™ 3.0/2x2M/1066FSB (Dual Processor Option)
- 2U rack designed with 550W Power Supply
- Intel® 5000P (Blackford) Chipset
- Kingston 1024MB 667MHz DDR2 ECC FB-DIMM (2x512MB)
- 1pc x Western Digital WD4000YR, SATA 7200RPM HD
- 6 x 1" Hot-swap SATA Drive Bays
- ATI ES1000 Graphics with 16MB video memory
- Intel® (ESB2/Gilgal) 82563EB Dual-port Gigabit Ethernet Controller
- RAID 0, 1, 5, 10 support

\$1,999

3U Application Server

SDR-3500T

64-bit + dual-core = power efficiency. This equation is ideal for intense computing environments and business-critical applications



- Intel® 64-bit Xeon™ 3.2/2x2M/1066FSB (Dual Processor Option)
- 3U Chassis with 650W Redundant Power Supply
- Intel® 5000P (Blackford) Chipset
- Kingston 1024MB 667MHz DDR2 ECC FB-DIMM (2x512MB)
- 1pc x 3Ware 9550SX-12 port RAID Controller Card
- 12pcs x Western Digital 250GB SATA RAID Drive
- ATI ES1000 Graphics with 16MB video memory
- Intel® (ESB2/Gilgal) 82563EB Dual-port Gigabit Ethernet Controller
- RAID 0, 1, 5, 10 support

\$4,799

Features

- 1U Rackmount Chassis with 700W Power Supply
- Intel® 64-bit Xeon™ 3.0/2x2M/667FSB (Dual Processor Option)
- Intel® 5000P (Blackford) Chipset
- Kingston 1024MB 667MHz DDR2 ECC FB-DIMM (2pcs x 512MB)
- 4pcs x WD4000YR, SATA 7200RPM hard drive
- 4 x 1" Hot-swap SATA Drive Bays
- RAID 0, 1, 5, 10 support
- Intel® (ESB2/Gilgal) 82563EB Dual-port Gigabit Ethernet Controller

3U Database Server

SDR-3500T

Outstanding performance with dual-core Xeon® 5000/5100. Ideal solution for storage and business applications.



- Intel® 64-bit Xeon™ 3.2/2x2M/1066FSB (Dual Processor Option)
- 3U Chassis with 650W Redundant Power Supply
- Intel® 5000P (Blackford) Chipset
- Kingston 1024MB 667MHz DDR2 ECC FB-DIMM (2x512MB)
- 2pcs x 3Ware 9550SX-8 port
- 16pcs x Seagate 300GB SATA-II Drive
- ATI ES1000 Graphics with 16MB video memory
- Intel® (ESB2/Gilgal) 82563EB Dual-port Gigabit Ethernet Controller
- RAID 0, 1, 5, 10 support

\$5,899

5U Advanced Storage Server

SDR-5500T

Powered by the latest dual-core Xeon 5000/5100® sequence processors, this system offers the best storage capacity available in a 5U format



- Intel® 64-bit Xeon™ 3.2/2x2M/1066FSB (Dual Processor Option)
- 5U Chassis, 24 hot-swap bays & 950W redundant power supply
- Intel® 5000P (Blackford) Chipset
- Kingston 1024MB 667MHz DDR2 ECC FB-DIMM (2x512MB)
- 1pc x 3Ware 9550SX-12 port RAID Controller Card
- 12pcs x Western Digital 400GB SATA RAID Drive
- ATI ES1000 Graphics with 16MB video memory
- Intel® (ESB2/Gilgal) 82563EB Dual-port Gigabit Ethernet Controller
- RAID 0, 1, 5, 10 support

\$6,799

CONTACT US TODAY!

Our flexible online configurators allow you to create custom solutions. If you prefer, call us and our expert staff will help you assemble the perfect system. Servers Direct - your 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.



HARDWARE

PathScale InfiniPath Interconnect

**InfiniBand and AMD HyperTransport
are made for each other just like soup
and...something that goes with soup.**

LOGAN G. HARBAUGH

As the use of large clusters gains ground in academia and moves from the scientific world to the business world, many administrators are looking for ways to increase performance without significantly increasing the cost per node. Some may focus on CPU power/speed or the amount of RAM per node, relatively expensive components, to increase their horsepower. PathScale (recently acquired by QLogic) is taking a different approach, instead focusing on unleashing the computational power already contained in the cluster as a whole by allowing the "thoroughbred" processors built by Intel and AMD to move all the messages they are capable of generating.

By focusing on dramatically increasing the message traffic between nodes and by reducing the latency of those messages, applications running on clusters are able to run faster and scale higher than previously possible. And, the increased performance is achieved with the combination of inexpensive x86 servers with standard InfiniBand adapters and switches.

The InfiniPath InfiniBand cluster interconnect is available in two flavors: PCI Express for ubiquitous deployments with any motherboard and any processor, and directly connected to the HyperTransport bus for the absolute lowest latency. This article deals with the InfiniPath HyperTransport (or HTX) product line. Servers with motherboards that support InfiniPath HTX are available from more than 25 different system vendors, including Linux Networx, Angstrom, Microway, Verari and Western Scientific. In the near future, servers with HTX slots could be available from the larger tier-one computer system suppliers. Motherboards with HTX slots are currently shipping from Iwill (the DK8-HTX) and Supermicro (H8QC8-HTe), with additional offerings from Arima, ASUS, MSI and others coming soon. InfiniPath adapters, which can be used with just about any flavor of Linux, can be connected to any InfiniBand switch from any vendor. Also, for mixed deployments with InfiniBand adapters from other vendors, InfiniPath supports the OpenFabrics (formerly OpenIB) software stack (downloadable from the PathScale Web site).

What the InfiniPath HTX adapter does better than any other cluster interconnect is accept the millions of messages generated every second by fast, multicore processors and gets them to the receiving processor. Part of the secret is removing all the delays associated with bridge chips and the PCI bus, because traffic is routed over the much faster HyperTransport bus. In real-world testing, this produces a two- to three-times improvement in latency, and in real-world clustered applications, an increase in messages per second of ten times or more.

Message transmission rate is the unsung hero in the interconnect world, and by completely re-architecting its adapter, InfiniPath beats the next-best by more than ten times. Where the rest of the industry builds off-load engines, miniature versions of host servers with an embedded processor and separate memory, InfiniPath is based on a very simple, elegant design that does not duplicate the efforts of the host processor. Embedded processors on interconnect adapter cards are only about one-tenth the speed of host processors so they can't keep up with the number of messages those processors generate. By keeping things

**Listing 1.
b_eff output**

```
The effective bandwidth is b_eff = 5149.154 MByte/s on 16 processes
( = 321.822 MByte/s * 16 processes)

Ping-pong latency: 1.352 microsec

Ping-pong bandwidth: 923.862 MByte/s at Lmax= 1.000 MByte
(MByte/s=1e6 Byte/s) (MByte=2**20 Byte)

system parameters : 16 nodes, 128 MB/node

system name : Linux

hostname : cbc-01

OS release : 2.6.12-1.1380_FC3smp

OS version : #1 SMP Wed Oct 19 21:05:57 EDT 2005

machine : x86_64

Date of measurement: Thu Jan 12 14:20:52 2006
```

simple, InfiniPath avoids wasting CPU cycles on pinning cache and other housekeeping chores, required with off-load engines, and instead does real work for the end user. The beauty of this approach is that it not only results in lower CPU utilization per MB transferred, but it also has a lower memory footprint on host systems.

The reason a two- or three-times improvement in latency has such a large effect on the message rate (messages per second) is that low latency reduces the time that nodes spend waiting for the next communication at both ends, so all the processors substantially reduce wasted cycles spent waiting on adapters jammed with message traffic.

What does this mean for real-world applications? It will depend on the way the application uses messages, the sizes of those messages and how well optimized it is for parallel processing. In my testing, using a 17-node (16 compute nodes and one master node) cluster, I got a result of 5,149.154 MB/sec using the b_eff benchmark. This compares with results of 1,553–1,660 MB/sec for other InfiniBand clusters tested by the Daresbury Lab in 2005, and with a maximum of 2,413 MB/sec for any other cluster tested. The clusters tested all had 16 CPUs.

See Listing 1 for the results of the b_eff benchmark. The results of the Daresbury Lab study are available at www.cse.clrc.ac.uk/disco/Benchmarks/commodity.2005/mfg_commodity.2005.pdf, page 21.

Most vendors do not publish their message rate, instead putting out their peak bandwidth and latency. But bandwidth varies with the size of the message, and peak bandwidth is achieved only at message sizes much larger than most applications generate. For most clustered applications, the actual throughput of the interconnect is a fraction of peak, because few clustered applications pass large messages back and forth between nodes. Rather, applications running on clusters pass a large number of very small (8–1,024 byte) messages back and forth as nodes

2U in 1U 4+1 Add-on Cards

SuperServer 6015X-8/T

- Dual Intel® Dual-Core 64-bit Xeon™ processors support
- 1333/1066/667MHz Front-Side Bus
- 2 PCI-E x8 w/RSC-R1UEP-2E8 (full-length)
 - 1 Universal PCI-X 133MHz / PCI-E x8 (full-length)
 - 1 Universal PCI-X 100MHz / PCI-E x4 (low-profile)
 - optional 2 PCI-X 133MHz w/RSC-R1UEP-A2X (full-length)
- 1 SIM1U management slot (optional 3rd data LAN)
- Up to 32GB Fully Buffered DIMM Memory 667/533 MHz
- 700W high-efficiency power supply w/I²C management
- 5x heavy duty counter-rotating fans w/optimal fan speed control and air shroud
- 3 hot-swap SCA/SATA drive bays
- Dual Gigabit LAN ports



SUPERMICRO® Serverboard X7DBX-8/T

Supermicro's Feature Advantages

- 4 Universal Expansion cards support PCI-X and PCI-E with interchangeable configurations
- Flexible Supermicro Intelligent Management (SIM): IPMI 2.0 with Built-in Virtual Media over LAN, optional KVM over LAN and 3rd LAN port
- High Efficiency Power Supply (85% and up) reducing energy costs and customer TCO
- Optimized air-flow for efficient cooling
- SAS version available



Dual-Core Xeon 5100/5000 Series

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

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

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

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

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

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

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

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



**ASA
COMPUTERS**

www.asacomputers.com

1-800-REAL-PCS

Hardware Systems For The Open Source Community—Since 1989

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

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

AMD Opteron® Value Server-\$795
1U 14.3" Deep
AMD Opteron 140 1M Cache
1 GB DDR ECC REG PC-3200
1 of 2 4.0GB SATA Drive
2 X 10/100/1000 NIC
Options: CD, FD, or Second Drive, Raid
ADD Your Logo



iSCSI Dual AMD Opteron™ 1U to 8U, Call for Pricing
1TB to 30TB of iSCSI Storage
Dual AMD Opteron 246
1 GB DDR ECC REG PC-3200
Dual GigE LAN
Redundant PS, Hot-Swap Drives
RAID Options, RAID 5, 10, 50
More Customization is available



1U SCSI Quad AMD Opteron™-Starting @ \$2850
1of 4 AMD Opteron 848
2 GB DDR ECC REG PC-3200
1 of 3 36GB SCSI Drive
2 GigaE, CD, FD,
Remote Management Card (IPMI)



30TB AMD Opteron™ Storage Solution- Starting @ \$26,395
30TB SATA Storage in 8U
Includes all Raid Cards, Raid 5, 10
Dual AMD Opteron 246
2 GB DDR, ECC REG PC-3200
Dual GigE, FD, CD



Your Custom Appliance Solution

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

"Your Logo Here"



Custom Server, Storage, Cluster, etc. Solutions

Please Contact us for all type of Storage solutions, NAS, DAS, iSCSI, Fiber RAID, SATA, SAS.

*Free shipping on selected servers and all notebooks.



2354 Calle Del Mundo, Santa Clara, CA 95054

www.asacomputers.com

Email: sales@asacomputers.com

P: 1-800-REAL-PCS | FAX: 408-654-2910

Prices and availability subject to change without notice.

Not responsible for typographical errors. All brand names and logos are trademark of their respective companies.

REVIEWS

Table 1. The InfiniPath 1.2 release has been tested on the following Linux distributions for AMD Opteron systems (x86_64).

Linux Release	Version Tested
Red Hat Enterprise Linux 4	2.6.9
CentOS 4.0-4.2 (Rocks 4.0-4.2)	2.6.9
Red Hat Fedora Core 3	2.6.11, 2.6.12
Red Hat Fedora Core 4	2.6.12, 2.6.13, 2.6.14
SUSE Professional 9.3	2.6.11
SUSE Professional 10.0	2.6.13

begin and finish processing their small pieces of the overall task.

This means that for most applications, the number of simultaneous messages that can be passed between nodes, or message rate, will tend to limit the performance of the cluster more than the peak bandwidth of the interconnect.

As end users attempt to solve more granular problems with bigger clusters, the average message size goes down and the overall number of messages goes up. According to PathScale's testing with the WRF modeling application, the average number of messages increases from 46,303 with a 32-node application to 93,472 with a 512-node application, while the mean message size decreases from 67,219 bytes with 32 nodes to 12,037 bytes with 512 nodes. This means that the InfiniPath InfiniBand adapter will become more effective as the number of nodes increases. This is borne out in other tests with large-scale clusters running other applications.

For developers, there is little difference between developing a standard MPI application and one that supports InfiniPath. Required software is limited to some Linux drivers and the InfiniPath software stack. Table 1 shows the versions of Linux that have been tested with the InfiniPath 1.2 release. PathScale also offers the EKOPATH Compiler Suite version 2.3, which includes high-performance C, C++ and Fortran 77/90/95 compilers as well as support for OpenMP 2.0 and PathScale-specific optimizations. But the compiler suite is not required to develop InfiniPath applications because the InfiniPath software environment supports gcc, Intel and PGI compilers as well. The base software provides an environment for high-performance MPI and IP applications.

The optimized ipath_ether Ethernet driver provides high-performance networking support for existing TCP- and UDP-based applications (in addition to other protocols using Ethernet), with no modifications required to the application. The OpenIB (Open Fabrics) driver provides complete InfiniBand and OpenIB compatibility. This software stack is freely available as a download on their Web site. It currently supports IP over IB, verbs, MVAPICH and SDP (Sockets Direct Protocol).

PathScale offers a trial program—you can compile and run your application on its 32-node cluster to see what performance gains you can attain. See www.pathscales.com/cbc.php.

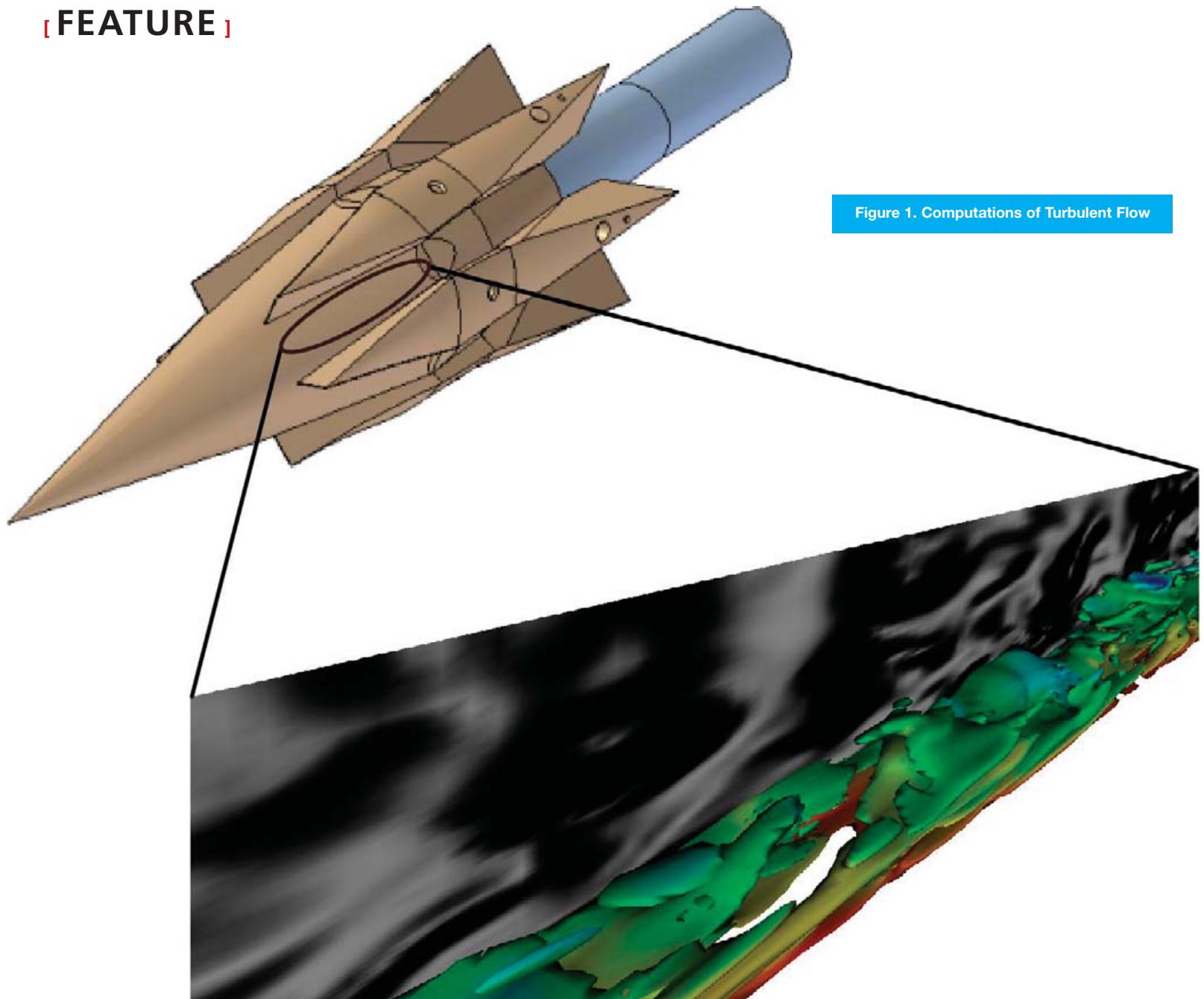
In addition, you can test your applications on the Emerald cluster at the AMD Developer Center, which offers 144 dual-socket, dual-core nodes, for a total of 576 2.2GHz Opteron CPUs connected with InfiniPath HTX adapters and a SilverStorm InfiniBand switch.

Tests performed on this cluster have shown excellent scalability at more than 500 processors, including the LS-Dyna three-vehicle collision results posted at www.topcrunch.org. See Table 2 for a listing of the top 40 results of the benchmark. Notice that the only other cluster in the top ten is the much more expensive per node Cray XD1 system.■

Logan Harbaugh is a freelance reviewer and IT consultant located in Redding, California. He has been working in IT for 20 years and has written two books on networking, as well as articles for most of the major computer publications.

Table 2. LS-Dyna Three-Vehicle Collision Results. Posted at www.topcrunch.org

Result (lower is better)	Manufacturer	Cluster Name	Processors	Nodes x CPUs x Cores
184	Cray, Inc.	CRAY XD1/RapidArray	AMD dual-core Opteron 2.2GHz	64 x 2 x 2 = 256
226	Cray, Inc.	CRAY XD1/RapidArray	AMD dual-core Opteron 2.2GHz	64 x 2 x 1 = 128
239	Cray, Inc.	CRAY XD1/RapidArray	AMD dual-core Opteron 2.2GHz	32 x 2 x 2 = 128
239	Rackable Systems/AMD Emerald/PathScale	InfiniPath/Silverstorm InfiniBand switch	AMD dual-core Opteron 2.2GHz	64 x 2 x 2 = 256
244	Cray, Inc.	CRAY XD1/RapidArray	AMD Opteron 2.4GHz	64 x 2 x 1 = 128
258	Cray, Inc.	CRAY XD1/RapidArray	AMD dual-core Opteron 2.2GHz	48 x 2 x 1 = 96
258	Rackable Systems/AMD Emerald/PathScale	InfiniPath/Silverstorm InfiniBand switch	AMD dual-core Opteron 2.2GHz	64 x 1 x 2 = 128
268	Cray, Inc.	CRAY XD1/RapidArray	AMD Opteron 2.4GHz	48 x 2 x 1 = 96
268	Rackable Systems/AMD Emerald/PathScale	InfiniPath/SilverStorm InfiniBand switch	AMD dual-core Opteron 2.2GHz	32 x 2 x 2 = 128
280	Cray, Inc.	CRAY XD1/RapidArray	AMD dual-core Opteron 2.2GHz	24 x 2 x 2 = 96
294	Rackable Systems/AMD Emerald/PathScale	InfiniPath/SilverStorm InfiniBand switch	AMD dual-core Opteron 2.2GHz	48 x 1 x 2 = 96
310	Galactic Computing (Shenzhen) Ltd.	GT4000/InfiniBand	Intel Xeon 3.6GHz	64 x 2 x 1 = 128
315	Cray, Inc.	CRAY XD1/RapidArray	AMD dual-core Opteron 2.2GHz	32 x 2 x 1 = 64
327	Cray, Inc.	CRAY XD1/RapidArray	AMD Opteron 2.4GHz	32 x 2 x 1 = 64
342	Cray, Inc.	CRAY XD1/RapidArray	AMD dual-core Opteron 2.2GHz	16 x 2 x 2 = 64
373	Rackable Systems/AMD Emerald/PathScale	InfiniPath/SilverStorm InfiniBand switch	AMD dual-core Opteron 2.2GHz	32 x 1 x 2 = 64
380	Cray, Inc.	CRAY XD1/RapidArray	AMD Opteron 2.2GHz	32 x 2 x 1 = 64
384	Cray, Inc.	CRAY XD1/RapidArray	AMD dual-core Opteron 2.2GHz	24 x 2 x 1 = 48
394	Rackable Systems/AMD Emerald/PathScale	InfiniPath/SilverStorm InfiniBand switch	AMD dual-core Opteron 2.2GHz	16 x 2 x 2 = 64
399	Cray, Inc.	CRAY XD1/RapidArray	AMD Opteron 2.4GHz	24 x 2 x 1 = 48
405	Cray, Inc.	CRAY XD1/RapidArray	AMD Opteron 2.2GHz	32 x 2 x 1 = 64
417	Cray, Inc.	CRAY XD1/RapidArray	AMD dual-core Opteron 2.2GHz	12 x 2 x 2 = 48
418	Galactic Computing (Shenzhen) Ltd.	GT4000/InfiniBand	Intel Xeon 3.6GHz	32 x 2 x 1 = 64
421	HP	Itanium 2 CP6000/InfiniBand TopSpin	Intel Itanium 2 1.5GHz	32 x 2 x 1 = 64
429	Cray, Inc.	CRAY XD1/RapidArray	AMD Opteron 2.2GHz	32 x 2 x 1 = 64
452	IBM	e326/Myrinet	AMD Opteron 2.4GHz	32 x 2 x 1 = 64
455	Cray, Inc.	CRAY XD1 RapidArray	AMD Opteron 2.2GHz	24 x 2 x 1 = 48
456	HP	Itanium 2 Cluster/InfiniBand	Intel Itanium 2 1.5GHz	32 x 2 x 1 = 64
480	PathScale, Inc.	Microway Navion/PathScale InfiniPath/SilverStorm IB switch	AMD Opteron 2.6GHz	16 x 2 x 1 = 32
492	Appro/Level 5 Networks	1122Hi-81/Level 5 Networks - 1Gb Ethernet NIC	AMD dual-core Opteron 2.2GHz	16 x 2 x 2 = 64
519	HP	Itanium 2 CP6000/InfiniBand TopSpin	Intel Itanium 2 1.5GHz	24 x 2 x 1 = 48
527	Cray, Inc.	CRAY XD1/RapidArray	AMD dual-core Opteron 2.2GHz	16 x 2 x 1 = 32
529	HP	Opteron CP4000/TopSpin InfiniBand	AMD Opteron 2.6GHz	16 x 2 x 1 = 32
541	Cray, Inc.	CRAY XD1/RapidArray	AMD Opteron 2.4GHz	16 x 2 x 1 = 32
569	Cray, Inc.	CRAY XD1/RapidArray	AMD dual-core Opteron 2.2GHz	8 x 2 x 2 = 32
570	HP	Itanium 2 Cluster/InfiniBand	Intel Itanium 2 1.5GHz	24 x 2 x 1 = 48
584	Appro/Rackable/Verari	Rackable and Verari Opteron Cluster/InfiniCon InfiniBand	AMD Opteron 2GHz	64 x 1 x 1 = 64
586	IBM	e326/Myrinet	AMD Opteron 2.4GHz	16 x 2 x 1 = 32
591	Self-made (SKIF program)/United Institute of Informatics Problems	Minsk Opteron Cluster/InfiniBand	AMD Opteron 2.2GHz (248)	35 x 1 x 1 = 35



CLUSTERING IS NOT ROCKET SCIENCE

HPC clustering for computing at hypersonic speed is not as difficult as it sounds.

ROWAN GOLLAN, ANDREW DENMAN AND MARLIES HANKEL

The rocket science involved with designing and developing supersonic combustion ramjets (scramjets) is a tricky business. High-performance Linux clusters are used to aid the study of scramjets by facilitating detailed computations of the gas flow through the scramjet engine. The computational requirements for this and other real-world problems go beyond a few PCs under a desk. Prior to the Linux-cluster age, researchers often had to scale down the problem or simplify the mechanisms being studied to the detriment of the solution accuracy. Now, for instance, entire scramjet engines can be studied at quite high resolution.

In this article, we try to serve two purposes: we describe our experiences as a research group operating a large-scale cluster, and we demonstrate how Linux and companion software has made that possible without requiring specialist HPC expertise. As HPC Linux clustering has

matured, it has become an aid to rocket science, without needing to be rocket science itself.

That last statement probably requires some clarification. When clusters were first built, they were heralded for offering unbeatable performance per dollar or "bang for buck", as the phrase goes. However, as you tried to scale up to large numbers of nodes, the operation of a large-scale cluster started to become quite complex. For a number of reasons, including lack of clustering software tools, large clusters required a full-time system administrator. We argue that this situation has changed now thanks to simple effective tools written for Linux that are aimed at cluster operation and management.

In June 2004, two research groups at the University of Queensland, the Centre for Hypersonics and the Centre for Computational Molecular Science, teamed up to purchase a cluster of 66 dual-Opteron nodes from Sun Microsystems. The people at Sun were generous enough to sponsor two-thirds of the cost of the machine. A grant from the Queensland Smart State Research Facilities Fund covered the remaining third of the machine cost. Additionally, the University of Queensland provided the infrastructure, such as the air conditioning and specially designed machine room. We suddenly faced the challenge, albeit a pleasant one, of operating a 66-node cluster that was an order of magnitude larger than our previous cluster of five or six desktops. We didn't have the resources to obtain expensive proprietary cluster control kits, nor did we have the experience or expertise in large-scale cluster management. We were, however, highly aware of the advantages Linux offered in terms of cost, scalability, flexibility and reliability.

We emphasise that the setup we arrived at is a simple but effective Linux cluster that allowed the group to get on with the business of research. In what follows, we discuss the challenges we faced as a research group scaling up to a large-scale cluster and how we leveraged open-source solutions to our advantage. What we have done is only one solution to cluster operation, but one that we feel offers flexibility and is easy for research groups to implement. We should point out that expensive cluster control kits with all the bells and whistles weren't an option for us with our limited budget. Additionally, at the time of initial deployment, the open-source Rocks cluster toolkit wasn't ready for our 64-bit Opteron hardware, so we needed to find a way of using the newest kernel that was 64-bit ready. The attraction of packaged cluster deployment kits is that they hide some of the behind-the-scene details. The disadvantages can be that the cluster builder is locked in to a very specific way of using and managing the cluster, and it can be hard to diagnose problems when things go wrong. In setting up our cluster, we've held to the UNIX maxim of "simple tools working together", and this has given us a setup that is highly configurable, easily maintained and has a transparency of operation.

LAYING THE FOUNDATIONS

When building a cluster of five nodes, our IT administrator had given us five IP addresses on the network. That was easy—our machines had an IP, and we left the details of security and firewalling to our network administrator. Now with 66 nodes plus front-end file servers and another 66 service processors each requiring an IP address, it was clear we'd have to use a private network. Basically, our IT administrator didn't want to know us and mumbled something vague about us trying a network address translation (NAT) firewall. So that's what we did; we grabbed an old PC and installed Firestarter and created a firewall for our cluster in about half an hour. Firestarter provides an intuitive interface to Linux's iptables. We created our NAT firewall and were able to forward a few ports through to the front ends allowing SSH access.

With the network topology sorted, the next challenge was installing the operating system on all 66 of the servers. Previously, we had been happy to spend a morning swapping CDs in and out of drives in order to install the OS on a handful of machines. We quickly realised we

Listing 1. Temperature Monitoring with Python Scripting

```
---- monitor_temp.py ----
SHUTOFF_TEMP = 50.0
mail_list = ['fake.email@fakedomain.com']

import os, re, smtplib

def send_mail(toaddrs, msg):

    fromaddr = 'sysadmin@fakedomain.com'
    server = smtplib.SMTP('smtp.uq.edu.au')
    server.sendmail(fromaddr, toaddrs, msg)
    server.quit()

    f = os.popen('hostname', "r")
    hostname = f.readline().split()[0]
    svc_proc = hostname[:4] + 'sp' + hostname[4:]
    f.close()

    f = os.popen("ipmitool -I lan -P password -H %s sensor | grep 'cpu[0-1].memtemp'" % svc_proc, "r")

    mail_sent = False

    for line in f.readlines():
        if mail_sent:
            break
        tokens = line.split()
        str = tokens[2]

        if str == 'NA':
            temp = 1.0
        else:
            temp = float(str)

        if temp >= SHUTOFF_TEMP:
            msg = 'Re: hot temperature initiated shutdown for %s\n' % hostname
            msg += 'The CPU memtemp for %s exceeded %.1f.\n' % (hostname, SHUTOFF_TEMP)
            msg += 'This node has been shutdown.\n'
            for address in mail_list:
                send_mail(address, msg)
            # Clean up scratch and power down
            os.system('rm -rf /scratch/*')
            os.system('/sbin/shutdown now -h')
            mail_sent = True
```

would require some kind of automated process to deal with 66 nodes. We found that the SystemImager software suite did exactly what we were looking for. Using the SystemImager suite, we needed to install the OS only on one node. After toying with the configuration of that node, we had our golden client, as they call it in SystemImager parlance, ready to go. The SystemImager tools allowed us to take an image and push out the image when required. We also required a mechanism to do OS installs over the network so that we could avoid CD swapping. One of the SystemImager scripts helped us to set up a Pre-boot Execution Environment (PXE) server. This execution environment is handed out to nodes during bootup and allows the nodes to proceed with a network install. With this minimal environment, the nodes partition their disks and then transfer the files that comprise the OS from the front-end server. For the record, we use Fedora Core 3 on the cluster. The choice was motivated by our own familiarity with that distribution and the fact that it is close enough to Red Hat Enterprise Linux that we are able to run the few commercial scientific applications that are installed on the cluster.

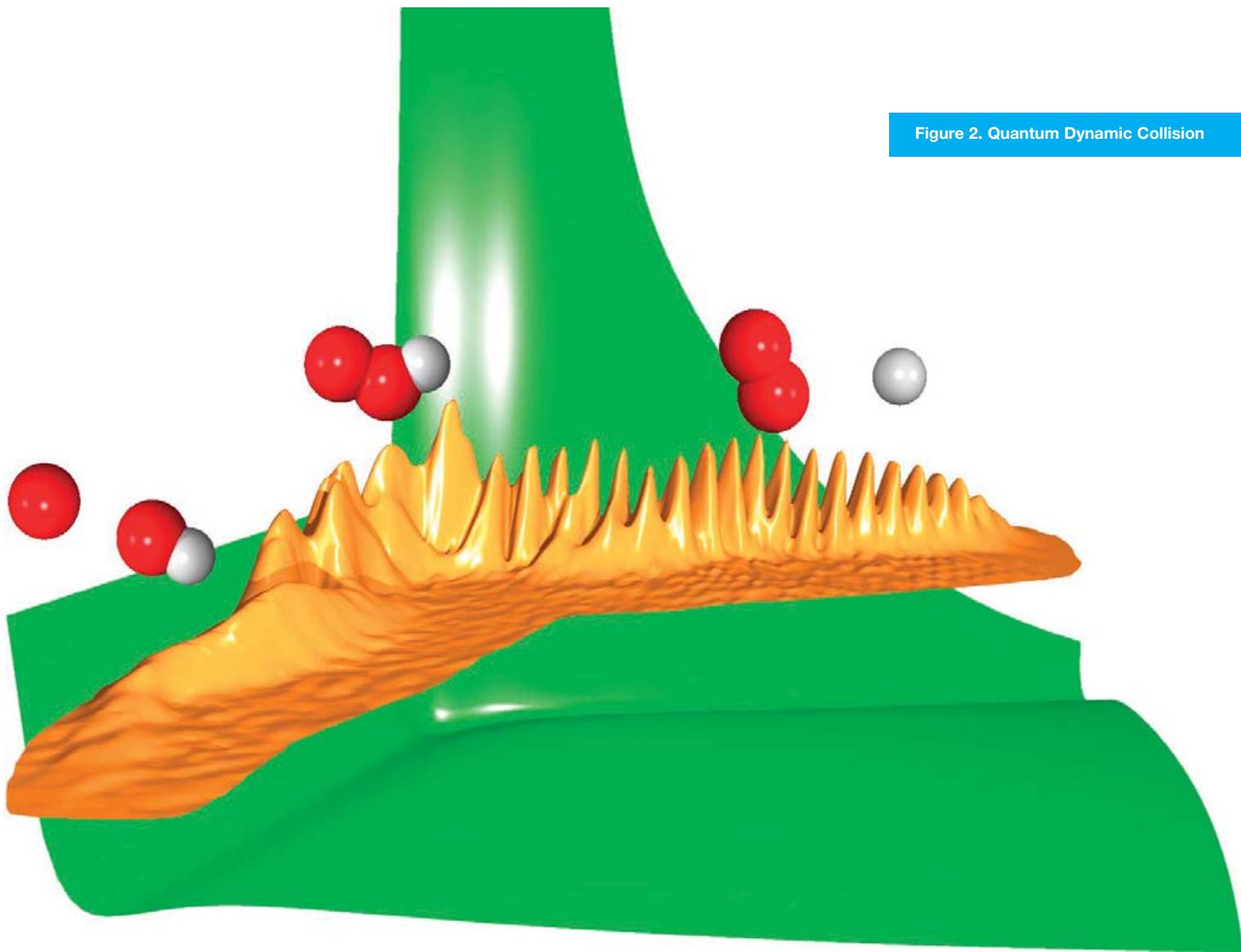


Figure 2. Quantum Dynamic Collision

A COLLECTION OF COMPUTERS BECOMES A CLUSTER

As we are interested in massively parallel computations, we needed to configure the servers to communicate with each other. We installed lam-mpi to use as a message-passing interface, and we configured the SSH service on each node to allow passwordless access between nodes by using host-based authentication. Note that lam-mpi doesn't do all the work of parallelizing your application; you still need to write or have available an MPI-aware code.

We configured an NFS (Network File System) server to provide a shared filesystem for all of the cluster compute nodes. We share the home directories of users across all nodes and some of the specialist applications we use for scientific computing. User accounts are managed by the Network Information Service (NIS) that comes standard with most Linux distributions.

DEALING WITH AN INCREASED NUMBER OF USERS

Previously, our computational group was about four people sharing time on five nodes. We had an extremely reliable job-scheduling system that involved a whiteboard and some marker pens. Clearly, this method of job scheduling would not scale as we expanded to a user base of about 40 users. We chose the Sun Grid Engine scheduling and batch processing software to install on the cluster.

The other challenge with the expanded user base was that the majority of users had limited experience with an HPC facility and little

or no experience using Linux. We decided that one of the best ways to share information about using the cluster was through the use of a wiki page. We set up a wiki page with the MediaWiki package. The wiki page has all manner of information about the cluster—from basic newbie-type information about copying files onto the cluster to advanced usage information about various compilers. The wiki page has been useful in bridging the knowledge gap between the sysadmins and the newbie users. The wiki allows for inexperienced cluster users to modify the documentation to make it simpler for other new players and also add neat tricks they may have devised themselves. The dynamic nature of a wiki page is a clear advantage when it comes to keeping documentation about the cluster facility up to date.

The second purpose of the wiki is to maintain an administrator's log of work on the cluster. As we sit in separate buildings, it was not practical to keep a traditional (physical) logbook. Instead, we use the wiki page to keep each other abreast of changes to the cluster. We actually keep this part of the Web page password-protected to ensure against any wiki vandalism.

ADMINISTERING EN MASSE

Sometimes it is necessary to issue commands on every node of the cluster or copy some files onto all nodes. Again, this wasn't a problem with five or six machines—we'd simply log in to the machines individually and do whatever was necessary. But with 66 machines, logging in to each

Polywell Server and PC Solutions

Specialize for Small to Medium Size Business

High-End Servers and Workstations

1U 4-way, 64GB DDR, 2TB RAID

- 2 x AMD® Opteron™ Dual-Core Processors 285+ with Hyper Transport Technology
- Upto 64GB 400MHz ECC DDR (16 Sockets)
- Upto 2TB 4 x 500G Swap Drive per 1U Rack
- 1 x PCI-X 133MHz or 1 x PCI-E RISER Slot
- Dual Gigabit Ethernet, ATI Graphics, 4 SATA-RAID
- CD-ROM Drive, Optional DVD-RW or CDRW
- Optional Slim Floppy Drive
- 1U 24" Depth Rack Chassis with upto 600W PS
- 4 x Swappable Drive Bays (SATA or SCSI)
- Supports Linux, FreeBSD or Windows
- Custom Configuration Available
- Please call for other Options



1114AIS-2050M, 285+, 32GB, 2TB **\$9,999**

1114AIS-2050M, 260HE, 16GB, 1TB **\$4,499**

1112ES-2200A, 265+, 2GB, 500GB **\$1,999**



4024AIS 4U 12TB SAS/SATA **\$12,500**

4024AIS 4U 6TB StorageServer **\$7,995**

AMD Dual-Core technology Enables one platform to meet the needs of multi-tasking and multi-threaded environments; providing platform longevity

Low-Cost Linux PCs

Allows users to run 32- and 64-bit applications as they desire – without sacrificing performance



626TD112-800LX
OEM Appliance starts at **\$299**

Linux Appliance PCs

- Custom Made Odd Size Chassis
- AC or DC Power Supply
- Low Power Voltage AMD Sempron™ Processor or High Performance AMD Athlon™ 64 Processor
- Diskless or Flash OS Boot Drive
- Swapable Hard Drive, CD-ROM, FDD
- Integrated Graphics, Ethernet, USB
- Optional LCD LED Control Module
- ISZ Audio, MPEG2/4 Hardware Video
- Upto 4 Ethernet Ports or 4 Serial Ports
- We have over 18 years OEM Experience in Set top Box, Digital Media Player, POS, Kios, Thin Client, Networking Appliance,

64GB DDR 4-way Workstation

- 2 x AMD® Opteron™ Dual-Core Processors 285+
- Upto 64GB ECC DDR (16 Memory Sockets)
- 2 x 133MHz PCI-X, 1 x PCI-E x16 Slots
- Dual Gigabit Ethernet, 4 x SATA-RAID Controller
- Quiet and Cool 11-Bay Tower + 1400W 80Plus P/S
- 2x300G HD, DVD-RW, Floppy, Optional Card Reader
- 8-Layer Motherboard with Special Quiet Cooling
- Optional Sound Card, 1394 firewire
- Supports 64/32-bit Linux, FreeBSD or Windows
- Special for Large Memory Intensive Applications
- Built-to-Order or Configure-to-Order

PolyStation 2050M

4-Way Two Dual-Core Processors

Up to 64GB memory, 16 Sockets

AMD64 architecture reduces I/O bottlenecks, increases bandwidth, and reduces memory latency. Critical information gets to those who need it quickly and efficiently.



64G RAM, 2x285+, QuadroFX5500 **\$22,999**

32G RAM, 2x265+, QuadroFX3450 **\$7,599**

16G RAM, 2x246+, QuadroFX1500 **\$3,750**

1U Power Saving ISP Server

- AMD Sempron™ or Opteron™ Processor
- 512M DDR 400MHz Memory
- 80GB Hard Drive
- 10/100Mbit Ethernet
- Drive Image Service
- 1U 14" Short Rack, allow 2 x 1U per Rack
- Low Power Usage, Data Center Cost Saving
- Perfect Entry Level ISP Server or Appliance System
- Disk-Less, IDE Flash Drive Boot Option
- Supports Linux, FreeBSD or Windows
- Custom Configuration is Available
- Please call us to discuss your specification



Order# VX2500SP1U-17LJ11A
starts at **\$399**

Pick the style for your application

2U 8-way, 5U 16-way Servers

- 8 or 4 AMD® Opteron™ Dual-Core Processors 865+ with Hyper Transport Technology
- Upto 128GB DDR Memory for 16-way (32 sockets)
- Upto 64GB DDR Memory for 8-way (16 sockets)
- 4x Gigabit LAN, 8x SATA RAID-5 for 16-way
- 2x Gigabit LAN, 4x SATA, U320 SCSI for 8-way
- 4 x 133/100/66MHz PCI-X Slots for 16-way
- 2 x 133, 2 x 66MHz PCI-X, 1x PCI Slots for 8-way
- On-board ATI Graphics, USB 2.0
- 5U 26" Rack 1300W 3+1 Redundant P/S 16-way
- 2U 27" Rack 700W PFC P/S for 8-way
- Supports Linux, FreeBSD or Windows
- Custom Configuration Available
- Please call for other Options



5U 16-way 865, 32GB, 2TB, 8801T5U **\$19,999**

2U 8-way 865, 8GB, 900GB, 8422C **\$6,999**

Mainstream Linux Systems

- AMD® Athlon™64 or X2 Dual-Core Processors with Hyper Transport Technology
- 512MB to 4GB Dual Channel DDR Memory
- 80GB to 500GB Hard Drive (SATA-RAID-0/1/5)
- 2 PCI, 2 PCIe, Gigabit LAN, 7.1 Audio, 1394a
- nForce 430 Chipset, GeForce 6100 Graphics
- Desktop, Tower, MiniBox or Rackmount Chassis
- Internet Server or Linux Appliance Configuration

939NV3000TD12LX starts at **\$399**



*AMD64 Investment Protection
Migrate to 64-bit platforms seamlessly.
Add 64-bit application as necessary.*

888.765.9686

www.Polywell.com/us/LJ



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



Polywell has been in business since 1987, Our professional engineers have many years of experiences helping small to mid-size business to custom make or migrate products into Linux Appliances. We also provide Linux/FreeBSD Driver Development Services.

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

Opteron, Sempron and ATHLON are trademarks of Advanced Micro Devices, Inc., Quadro, nForce and Nvidia are trademarks of NVIDIA Corporation. All other brands, names are trademarks of their respective companies.

machine individually becomes both tedious and error-prone. Our solution here was to use the C3 package developed by the group at the Oak Ridge National Laboratory. C3 stands for cluster command and control. It provides a set of Python scripts that allows for remote execution of commands across the cluster. There is also a tool to allow for copying files to groups of compute nodes. This is a Python script that uses rsync to do the transfer.

Speaking of Python scripts, we have found Python to be a useful all-purpose scripting language for cluster work. The particular attraction to Python is its sophisticated support for string manipulation. This allows us to take the text-based output from a number of standalone programs and parse it into more meaningful information. For example, the queuing system provides some detailed information about the status of the cluster, such as available processors on each node and queue availability on each node. Using Python, we can take the detailed output from such a command and provide some summary statistics that give us an indication of cluster load at a glance. Another example of Python scripts in action is our monitoring of temperatures on the compute nodes. This script is displayed in Listing 1. Python's ease of string handling and access to system services come in handy for many scripting tasks on the cluster.

The temperature monitoring script makes use of the intelligent platform management interface (IPMI). By using the IPMI specification, we had an implementation of a monitoring subsystem that permitted fully remote and customizable management of the compute nodes. Each compute node came equipped with a PowerPC service processor that communicated on a separate network from the main cluster. By combining the power of the open-source tools of Python and IPMItool, we created a totally autonomous thermal monitoring system. The system can shut down individual compute nodes if they exceed a predetermined temperature or cut the power if the server doesn't respond to a shutdown command. An e-mail is also sent, using the Python smtplib, to the admin team to advise of the situation.

SCALING UP TO MORE COMPUTATIONAL GRUNT

About 12 months after receiving our first 66 nodes, we had the opportunity to purchase 60 more dual-Opteron nodes, thanks to funding from the University of Queensland. Applying the same tools and techniques just described, we were able to integrate the additional 60 nodes into our cluster with minimal time and effort. The main technical difficulty we faced as we scaled up the compute resources was the additional load on the file server. It is well known that the present NFS version (v3) that is bundled with Linux does not scale well with increasing nodes. We have circumvented this situation by employing two file servers to share the load. The ideal situation would have been to invest in a dedicated storage area network (SAN). With 66 nodes, this would have been overkill, and due to the capricious nature of research funding in a university environment, we could never predict that we would have the money to buy an additional 60 nodes.

CLUSTER IN ACTION

Although there is a little more detail involved with the cluster setup, such as setting common time across the cluster with NTP, the collection of tools just described forms the basis of cluster operation and administration. This leaves time for research and the chance to use the cluster for some interesting science and engineering.

At the Centre for Hypersonics at the University of Queensland, there are two primary areas of research: planetary-entry vehicles and scramjets. Planetary-entry vehicles experience enormous heat loads during atmospheric entry, and this is a primary design concern for the aerospace engineer. Using the cluster, we can do large-scale parallel computations of the high-temperature gaseous flow around typical spacecraft. So far, we have studied spacecraft re-entering Earth, entering Mars and Titan, the largest moon of Saturn. In addition to computations of realistic spacecraft configurations, we also study simplified geometries like spheres and cylinders in order to better understand the fundamental flow physics at these high temperatures.

The other main focus of research at the Centre for Hypersonics is the study, design and optimization of scramjet engines. When traveling at speeds many times faster than the Concorde, scramjets suffer from large

amounts of aerodynamic drag. The drag forces experienced play a leading role in determining the performance capabilities of these engines. The cluster allows for theories of drag reduction, such as near wall hydrogen combustion, to be examined in very fine detail. Using complex three-dimensional turbulence models, we can study the very fine scales of the flow that govern the amount of drag.

Figure 1 shows an example of the results of computations inside a scramjet combustion chamber. The colored contours represent vorticity, which is an indication of mixing, and the shaded pattern shows flow density variations.

The Centre for Computational Molecular Science (CCMS) engages in interdisciplinary research in areas where molecular scale computations are involved. The areas of research are diverse and include studies of electronic structure, quantum and molecular dynamics, computational nanotechnology and biomolecular modelling. Among the current projects is the computational modelling of red fluorescent proteins found in coral reefs that have application in deep-tissue biomedical imaging. Another project is investigating materials for hydrogen storage in future fuel cell technologies.

The quantum and molecular dynamics group conducts research into the detailed dynamics and mechanisms of gas phase reactions. These reactions involving only a few atoms often play the key role in atmospheric or combustion cycles. The detailed quantum-level calculations are parallel in nature and are impractical to do serially as the memory requirements far exceed the average desktop. Of current interest is the study of the reaction of hydrogen with molecular oxygen. It is one of the most important reactions in the combustion of hydrocarbon fuels.

Figure 2 provides a graphical representation of quantum dynamic collision of an hydrogen atom and an oxygen molecule. The figure shows the wavefunction and the potential energy for the HOO system. From right to left: the hydrogen atom approaches the oxygen molecule, the HOO complex is formed (a deep well can be seen in the potential energy surface), the complex breaks apart and the products O and OH (hydroxyl radical) are formed.

CONCLUSION

In this article, we have given an oversight of the Opteron cluster setup at the University of Queensland. We have described how effective large-scale cluster computing can be managed by a few sysadmins looking over the cluster a couple of hours per week. The success of the cluster deployment has been in part due to the quality open-source Linux tools available for cluster operation, such as the SystemImager imaging suite and the C3 package for remote command execution. We believe there are significant advantages by using these simple tools rather than cluster deployment kits. Those advantages are a highly configurable and easily upgradable system. Our cluster has been extremely reliable, and the biggest source of downtime is the power interruptions we get due to storms typical of a Queensland summer.

As for the future, we may be approaching the time when we need to consider seriously the use of some type of parallel filesystem. We have been lucky so far with our NFS file server, but we had to educate our users about file staging and ask them to treat the file server with a little bit of respect. But for now, it's all systems go. ■

Resources for this article: www.linuxjournal.com/article/9133

Rowan Gollan is a PhD student at the Centre for Hypersonics, the University of Queensland, Australia. When not researching radiating flows about planetary-entry vehicles, his duties include part-time supervision of the cluster and a few departmental Linux servers.

Andrew Denman is also a PhD student at the Centre for Hypersonics. Andrew's doctorate is about the computation of turbulent compressible flows. He is also the ultimate authority for all happenings on the cluster.

Maries Hankel is a Postdoctoral Researcher at the Centre for Computational Molecular Science. Marlies represents the interests of the computational scientists and prevents them from being bullied by the engineers. Marlies' current research focus is on quantum dynamics of reactive scattering processes relevant to combustion and atmospheric chemistry.



THE #1 CHOICE OF SERVER AND STORAGE CRITICS

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

Network Computing — Aberdeen AberNAS 128

**"brute of a server housed in a tidy 3U package...
powerful enough to tackle the most cutting-edge applications"**

CRN Test Center Recommended — Aberdeen Stonehaven A381

Finalist: Best Servers

LinuxWorld Product Excellence Awards — Aberdeen Stonehaven A261

"unrivaled five-year warranty"

PC Magazine

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

PC Magazine — Aberdeen Stonehaven A261

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

*CRN Test Center Recommended
— Aberdeen AberSAN i100*



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

PC Magazine — Aberdeen XDAS

THE ABERDEEN CODE

THE SECRET IS OUT

ABERDEEN LLC PRESENTS ABERDEEN SERVERS FEATURING STIRLING STONEHAVEN BACKUP MONSTER TERASTORUS TERAZILLA
AND ABERDEEN STORAGE FEATURING ABERDEEN ABERNAS NETWORK ATTACHED STORAGE • ABERDEEN XDAS DIRECT ATTACHED STORAGE •
ABERDEEN ABERSAN STORAGE AREA NETWORKS PRODUCED BY THE ABERDEEN TEAM, CA, USA

Now PLAYING AT A FORTUNE 500™ COMPANY NEAR YOU.

FORTUNE 500 is a trademark of Time Inc. Links to above mentioned magazine articles and reviews
can be found at www.aberdeencorp.com/abcatg/reviews.htm and www.aberdeencorp.com/lj015

888-297-7409

www.aberdeencorp.com/linux



Getting Started with Condor

How to try Condor for multiplatform distributed computing.

IRFAN HABIB

Cluster computing emerged in the early 1990s when hardware prices were dropping and PCs were becoming more and more powerful. Companies were shifting from large mini-computers to small and powerful micro-computers, and many people realized

that this would lead to a large-scale waste of computing power, as computing resources were being fragmented more and more. Organizations today have hundreds to thousands of PCs in their offices. Many of them are idle most of the time. However, the same

organizations also face huge computation-intensive problems and thus require great computing power to remain competitive—hence the stable demand for supercomputing solutions that largely are built on cluster computing concepts.

Many vendors offer commercial cluster computing solutions. By using free and open-source software, it is possible to forego the purchase of these expensive commercial cluster computing solutions and set up your own cluster. This article describes such a solution, developed by University of Wisconsin, called Condor.

The idea behind Condor is simple. Install it on every machine you want to make part of the cluster. (In Condor terminology, a Condor cluster is called a pool. This article uses both terms interchangeably.) You can launch jobs from any machine, and Condor matches the requirements of the job with the capabilities offered by the idle computers currently available. Once it finds a suitable idle machine, it transfers the job to it, executes it and retrieves the results of the execution. One of the features of Condor is that it

doesn't require programs to be modified to run on the cluster.

In practice, however, Condor is more complicated. Condor is installed in different configurations on each machine. Each Condor pool has a central manager. The central manager, as the name implies, is the central manager of the cluster. It manages the detection of new idle machines and coordinates the matchmaking between job requirements and available resources. Machines in a Condor pool also can have Submit and Full Install configurations. Submit machines are those machines that can only submit jobs, but can't run any jobs; Full Install machines are machines that can do both, submit and execute.

REQUIREMENTS AND INSTALLATION

Condor does not require the addition of any new hardware to the network; the existing network itself is sufficient. Condor runs on a variety of operating systems, including Linux, Solaris, Digital Unix, AIX, HP-UX and Mac OS X as well as MS Windows 2000 and XP. It supports various architectures, including Intel x86, PowerPC, SPARC and so on. However, jobs developed on one specific architecture, such as Intel x86, will run only on Intel x86 computers. So, it is best if all the computers in a Condor pool are of a single architecture. It is possible, however, for Java applications to run on different architectures.

In this article, we cover the installation from basic tarballs on Linux, although distribution/OS-specific packages also may be available from the official site or sources. (See the Condor Project site for more details, www.cs.wisc.edu/condor/downloads.)

Download the tarball from the Project site, and uncompress it with:

```
tar -zvf condor.tar.gz
```

The condor_install script, located in the sbin directory, is all you need to run to set up Condor on a machine. Before you run this script, add a user named condor. For security reasons, Condor does not allow you to run jobs as root; thus, it is advisable to make a new user to protect the system.

One of the first questions the script asks is how many machines are you setting up to be part of the pool? This is important if you have a shared filesystem. If you do, the installation script will prompt you for the names of those machines, and the installation of Condor on those machines will be handled by the software itself. If a shared filesystem does not exist, you have to install Condor manually on each system. Also, if you want to be able to use Java support, you need to have Sun's Java virtual machine installed prior to installing Condor. The install script provides plenty of help and annotation on each question it asks, and you always can turn to Condor's comprehensive user manual and its associated mailing lists for help.

The variable \$CONDOR is used from now on to denote the root path where condor has been installed (untarred).

After the installation, start Condor by running:

```
$CONDOR/bin/condor_master
```

This command should spawn all other processes that Condor requires. On the central manager, you should be able to see five condor_ processes running after entering:

```
ps -aux | grep condor
```

On the central manager machine, you should have the following processes:

- ▶ condor_master
- ▶ condor_collector
- ▶ condor_negotiator
- ▶ condor_startd
- ▶ condor_schedd

Expert included.



Chris codes a wide variety of applications, and he expects his hardware to keep pace. He developed the Silicon Mechanics website to help customers find and configure the right servers for their needs, then developed a suite of fully integrated tools to support the entire production process, from configuration through delivery. Chris likes rack-mount servers based

on the Dual-core AMD Opteron™ processor with AMD Virtualization™ because virtualization allows him to develop and test software across multiple operating systems on a single server. In addition, the integrated memory controller, now supporting DDR2, reduces latency for fast memory reads, yielding quick computational processing for increased performance.

When you partner with Silicon Mechanics, you get more than a powerful AMD solution — you get an expert like Chris.



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

► FEATURE: GETTING STARTED WITH CONDOR

All other machines in the pool should have processes for the following:

- ▶ condor_master
- ▶ condor_startd
- ▶ condor_schedd

And, on submit-only machines you will see:

- ▶ condor_master
- ▶ condor_schedd

After that, you should be able to see the central manager machine as part of your Condor cluster when you run `condor_status`:

```
$CONDOR/bin/condor_status
Name OpSys Arch State Activity LoadAv Mem ActvtyTime Mycluster
LINUX INTEL Unclaimed Idle 0.115 3567 0+00:40:04
Machines Owner Claimed Unclaimed Matched Preempting
INTEL/LINUX 1 0 0 1 0 0
Total 1 0 0 1 0 0
```

If you now run `condor_master` on the other machines in the pool, you should see that they are added to this list within a few minutes (usually around five minutes).

LAUNCHING JOBS ON YOUR NEW CLUSTER

To test our new condor setup, let's create a simple "Hello Condor" job:

```
#include
int main()
{ printf("Hello World!\n");}
```

Compile the application with gcc.

Now, to submit a job to Condor, we need to write a submit file. A submit file describes what Condor needs to do with the job—that is, where it will get the input for the application, where to produce the output and if any errors occur, where it should store them:

```
Universe = Vanilla
Executable = hello
Output = hello.out
Input = hello.in
Error = hello.err
Log = hello.log
Queue
```

The first Universe entry defines the runtime environment under which Condor should run the job. Two Universes are noteworthy: for long jobs, such as those that will last for weeks and months, the Standard Universe is recommended, as it ensures reliability and the ability to save partial execution state and relocate the job to another machine automatically if the first machine crashes. This saves a lot of vital processing effort. However, to use the Standard Universe, the application must be "condor compiled", and the source code is required. The Vanilla Universe is for jobs that are short-lived, but long jobs also can be executed if the stability of the machines is guaranteed. Vanilla jobs can run unmodified binaries.

Other Universes in Condor include PVM, MPI and Java, for PVM, MPI and Java applications, respectively. For more detail on Condor Universes consult the documentation.

In this example, our executable file is called `hello` (the traditional "Hello Condor" program), and we're using the Vanilla Universe. The Input, Output, Error and Log directives tell Condor which files to use for stdin, stdout and stderr and to log the job's execution. Finally, the Queue directive specifies how many copies of the program to run.

After you have the submit file ready, run `condor_submit hello.sub`

to submit it to Condor. You can check on the status of your job using `condor_q`, which will tell you how many jobs are in the queue, their IDs and whether they're running or idle, along with some statistics.

Condor has many other features; so far we have covered only the basics of getting it up and running. A number of tutorials are available on-line, along with the Condor Manual (www.cs.wisc.edu/condor/manual), that will teach you the basic and advanced capabilities of Condor. When reading the Condor Manual, pay particular attention to the Standard Universe, which allows you to checkpoint your job, and the Java Universe, which allows you to run Java jobs seamlessly.

You also can add Condor to the boot sequence of your central manager and other machines. You can shut down cluster machines, and their jobs will continue or restart on a different machine (depending on whether it's a Standard Universe job or a Vanilla job). This allows for a lot of flexibility in managing a system.

BEYOND CLUSTERS

Condor is not only about clusters. An extension to Condor allows jobs submitted within one pool of machines to execute on another (separate) Condor pool. Condor calls this flocking. If a machine within the pool where a job is submitted is not available to run the job, the job makes its way to another pool. This is enabled by special configuration of the pools.

The simplest flocking configuration sets a few configuration variables in the `condor_config` file. For example, let's set up an environment where we have two clusters, A and B, and we want jobs submitted in A to be executed in B. Let's say cluster A has its central manager at `a.condor.org` and B at `b.condor.org`. Here's the sample configuration:

```
FLOCK_TO = b.condor.org
FLOCK_COLLECTOR_HOSTS = $(FLOCK_TO)
FLOCK_NEGOTIATOR_HOSTS = $(FLOCK_TO)
```

The `FLOCK_TO` variable can specify multiple pools, by entering a comma-separated list of central managers. The other two variables usually point to the same settings that `FLOCK_TO` does. The configuration macros that must be set in pool B authorize jobs from pool A to flock to pool B. The following is a sample of configuration macros that allows the flocking of jobs from A to B. As in the `FLOCK_TO` field, `FLOCK_FROM` allows users to authorize the flocking of incoming jobs from specific pools:

```
FLOCK_FROM=a.condor.org
HOSTALLOW_WRITE_COLLECTOR = $(HOSTALLOW_WRITE), $(FLOCK_FROM)
HOSTALLOW_WRITE_STARTD = $(HOSTALLOW_WRITE), $(FLOCK_FROM)
HOSTALLOW_READ_COLLECTOR = $(HOSTALLOW_READ), $(FLOCK_FROM)
HOSTALLOW_READ_STARTD = $(HOSTALLOW_READ), $(FLOCK_FROM)
```

The above settings set flocking from pool A to pool B, but not the reverse. To enable flocking in both directions, each direction needs to be considered separately. That is, in pool B you would need to set the `FLOCK_TO`, `FLOCK_COLLECTOR_HOSTS` and `FLOCK_NEGOTIATOR_HOST` to point to pool A, and set up the authorization macros in pool A for B.

Be careful with `HOSTALLOW_WRITE` and `HOSTALLOW_READ`. These settings let you define the hosts that are allowed to join your pool, or those that can view the status of your pool but are not allowed to join it, respectively.

Condor provides flexible ways to define the hosts. It is possible, for example, to allow read access only to the hosts that belong to a specific subnet, like this:

```
HOSTALLOW_READ=127.6.45.*
```

CONDOR-G

Another way to link distributed Condor pools together is by using Condor's grid computing features, which utilize the Globus Toolkit (www.globus.org). The Globus Toolkit is an open-source software toolkit used for building Grid systems and applications. It provides an infrastructure for authentication, authorization and remote job submission (including data transfer) on Grid

resources. Condor-G, an extension of Condor, provides all of Condor's job submission features, but for far-removed resources on the Grid.

Condor-G is sort of a gateway to the Grid for Condor pools. Condor-G is a program that manages both a queue of jobs and the resources from one or more sites where those jobs can execute. It communicates with these resources and transfers files to and from these resources using the Globus mechanisms. For more detail on setting up Condor-G, consult the Condor Manual mentioned previously.

A sample submit file for a job to be executed over Globus looks like this:

```
executable = mygridjob
globusscheduler = grid.sample.net/jobmanager
input=mygridi.txt
universe = globus
output = mygridjob.out
log = mygridjob.log

queue
```

As you can see, there are only two differences with Grid jobs and normal local pool jobs. The Universe is Globus, which tells Condor that this job will be scheduled to the Grid. And, we specify the globusscheduler, which points to the Globus Job manager at the remote site. The jobmanager is the Globus service that is spawned at the remote site to submit, keep track of and manage Grid I/O for jobs running on the local system there. Grid jobs can be monitored the same way as ordinary Condor jobs with condor_q.

CONCLUSION

Condor provides the unique possibility of using our current computing infrastructure and investments to target processing of jobs that are simply beyond the capabilities of our most powerful systems. Condor is easy-to-install and easy-to-use software for setting up clusters. Condor is scalable. It provides options to extend its reach from a single cluster to interconnecting clusters that can be located anywhere in the world. Condor has been fundamental software for many grid computing projects. Various success stories with Condor have been reported in the press. One of the recent ones is of Micron Technologies. Micron is one of the world's leading providers of advanced semiconductor solutions. In an interview with *GridToday* in April 2006, a senior fellow at Micron said that they had deployed 11 Condor pools consisting of 11,000 processors, located in four countries in seven different sites. Why Condor? Because it supported all the platforms Micron was interested in, and it was already widely used, well supported, and of course, it was open source. These pools have become a vital asset for Micron. They are used for everything from manufacturing, engineering, reporting and software development to security. Condor is not only a research toy, but also a piece of robust open-source software that solves real-world problems.■

Irfan Habib is an undergraduate student in software engineering at the National University of Sciences Technology Pakistan. He has been deeply interested in free and open-source software for years, and he does research in Distributed and Grid Computing. Condor combines both of his interests. He can be reached at irfan.habib@niit.edu.pk.



Don't complicate a simple task

Keep basic tasks just that with handheld, stationary and vehicle-mounted wireless data collection from AML. While others are busy reinventing the wheel, we're keeping things simple, from our products to our personal service. Visit us at www.amltd.com or call **1.800.648.4452** for a real live person.



AML
NEVER COMPLICATED™

M7100 Wireless Family



DRBD in a Heartbeat

How to build a redundant, high-availability system with DRBD and Heartbeat.

PEDRO PLA

About three years ago, I was planning a new server setup that would run our new portal as well as e-mail, databases, DNS and so forth. One of the most important goals was to create a redundant solution, so that if one of the servers failed, it wouldn't affect company operation.

I looked through a lot of the redundant solutions available for Linux at the time, and with most of them, I had trouble getting all the services we needed to run redundantly. After all, there is a very big difference in functionality between a Sendmail daemon and a PostgreSQL daemon.

In the end, though, I did find one solution that worked very well for our needs. It involves setting up a disk mirror between machines using the software DRBD and a high-availability monitor on those machines using Heartbeat.

DRBD mirrors a partition between two machines allowing only one of them to mount it at a time. Heartbeat then monitors the machines, and if it detects that one of the machines has died, it takes control by mounting the mirrored disk and starting all the services the other machine is running.

I've had this setup running for about three years now, and it has made the inevitable hardware failures unnoticeable to the company.

In this tutorial, I show you how to set up a redundant Sendmail system, because once you do that, you will be able to set up almost any service you need. We assume that your master server is called server1 and has an IP address of 192.168.1.1, and your slave server is called server2 and has an IP address of 192.168.1.2.

And, because you don't want to have to access your mail server on any of these addresses in case they are down, we will give it a virtual address of 192.168.1.5. You can, of course, change this to whatever address you want in the Heartbeat configuration that I discuss near the end of this article.

HOW IT WORKS

This high-availability solution works by replicating a disk partition in a master/slave mode. The server that is running as a master has full read/write access to that partition; whereas the server running as slave has absolutely no access to the partition but silently replicates all changes made by the master server.

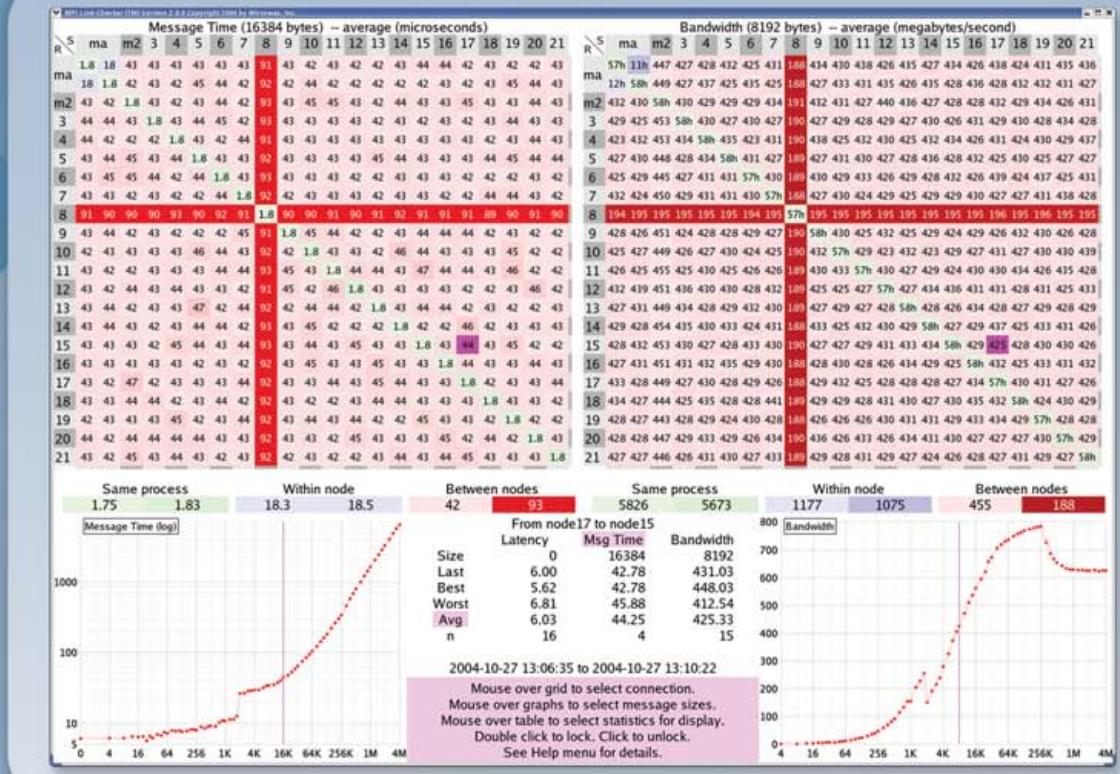
Because of this, all the processes that need to access the replicated partition must be running on the master server. If the master server fails, the Heartbeat daemon running on the slave server will tell DRBD that it is now the master, mount the replicated partition, and then start all the processes that have data stored on the replicated partition.

HOW TO GET IT RUNNING

The first step for running a redundant system is having two machines ready to try it out. They don't need to have identical specs, but they should meet the following requirements:

- ▶ Enough free space on both machines to create an equal-sized partition on each of them.
- ▶ The same versions of the daemons you want to run across both machines.
- ▶ A network card with crossover cable or a hub/switch.
- ▶ An optional serial port and serial port crossover cable for additional monitoring.

X Marks the Slow Node!



MPI Link-Checker™ to the Rescue!

A single slow node or intermittent link can cut the speed of MPI applications by half. Whether you use GigE, Myrinet, Quadrics, InfiniBand or InfiniPath HTX, there is only one choice for monitoring and debugging your cluster of SMP nodes: Microway's MPI Link-Checker™.

This unique diagnostic tool uses an end-to-end stress test to find problems with cables, processors, BIOS's, PCI buses, NIC's, switches, and even MPI itself! It provides instant details on how latency and bandwidth vary with packet size. It also provides ancillary data on inter-process and intra-CPU latency, and includes FastCheck!, which runs in CLI mode and checks up to 100 nodes per second.

A complimentary one year license for MPI Link-Checker™ is installed on every Opteron based Microway cluster purchased in 2006.

Wondering what's wrong with your cluster's performance, or need help designing your next one? Microway designs award-winning single and dual core AMD Opteron based clusters. Dual core enables users to increase computing capacity without increasing power requirements, thereby providing the best performance per watt. Configurations include 1U, 2U, and our 4U QuadPuter™ RuggedRack™—available with four or eight dual core Opterons, offering the perfect balance between performance and density.

Microway has been an innovator in HPC since 1982. We have thousands of happy customers in HPC, Energy, Enterprise and Life Science markets. Isn't it time you became one?

Call us first at 508-746-7341 for quotes and benchmarking services. Find technical information, testimonials, and newsletter at www.microway.com.



▲ Microway® Quad Opteron™ Cluster with 36 Opteron 880s, redundant power, 45 hard drives and Myrinet™ in our CoolRak™ cabinet.

Microway
23 Years of Expertise Built In



► FEATURE: DRBD IN A HEARTBEAT

You also should think carefully about which services you want running on both machines, as this will affect the amount of hard disk you will need to dedicate to replication across them and how you will store the configuration and data files of these services.

It's very important that you have enough space on this shared partition, because it will be the main data storage location for all of these services. So, if you are going to be storing a large Sendmail spool or a database, you should make sure it has more than enough space to run for a long time before having to repartition and reconfigure DRBD for a larger disk size.

SETTING UP THE BASICS ON YOUR SERVERS

Once you've made sure your machines are ready, you can go ahead and create an equal-sized partition on both machines. At this stage, you do not need to create a filesystem on that partition, because you will do that only once it is running mirrored over DRBD.

For my servers, I have one DRBD replicated drive that looks like this on my partition tables:

```
/dev/sda5      7916    8853   7534453+  83  Linux
```

Listing 1. /etc/drbd.conf

```
# Each resource is a configuration section for a
# mirrored disk.
# The drbd0 is the name we will use to refer
# to this disk when starting or stopping it.

resource drbd0 {
    protocol C;
    handlers {
        pri-on-incon-degr "echo 'DRBD: primary requested but inconsistent!' \
            >| wall; /etc/init.d/heartbeat stop"; #"/halt -f";
        pri-lost-after-sb "echo 'DRBD: primary requested but lost!' \
            >| wall; /etc/init.d/heartbeat stop"; #"/halt -f";
    }
    startup {
        degr-wfc-timeout 120;    # 2 minutes.
    }

    disk {
        on-io-error    detach;
    }
    # These are the network settings that worked best for me.
    # If you want to play around with them, go
    # ahead, but take a look in the man pages of drbd.conf
    # and drbdadm to see what each does.

    net {
        timeout 120;
        connect-int 20;
        ping-int 20;
        max-buffers     2048;
        max-epoch-size  2048;
        ko-count 30;

        # Remember to change this shared-secret on both the master
        # and slave machines.

        cram-hmac-alg "sha1";
        shared-secret "FooFunFactory";
    }

    syncer {
```

```
        rate 10M;
        al-extents 257;
    }

    # This next block defines the settings for the server
    # labeled as server1. This label should be in your
    # /etc/hosts file and point to a valid host.

    on server1 {
        # The following device will be created automatically by
        # the drbd kernel module when the DRBD
        # partition is in master mode and ready to write.
        # If you have more than one DRBD resource, name
        # this device drbd1, drbd2 and so forth.

        device      /dev/drbd0

        # Put the partition device name you've prepared here.

        disk       /dev/sda5;

        # Now put the IP address of the primary server here.
        # Note: you will need to use a unique port number for
        # each resource.

        address    192.168.1.3:7788;
        meta-disk  internal;
    }

    # This next block is identical to that of server1 but with
    # the appropriate settings of the server called
    # server2 in our /etc/hosts file.

    on server2 {
        device      /dev/drbd0;
        disk       /dev/sda5;
        address    192.168.1.2:7788;
        meta-disk  internal;
    }
```

Note: type `fdisk -l` at your command prompt to view a listing of your partitions in a format similar to that shown here. Also, in my case, the partition table is identical on both redundant machines.

The next step after partitioning is getting the packages for Heartbeat version 1.2+ and DRBD version 0.8+ installed and the DRBD kernel module compiled. If you can get these prepackaged for your distribution, it will probably be easier, but if not, you can download them from www.linux-ha.org/DownloadSoftware and www.drbd.org/download.html.

Now, go to your `/etc/hosts` file and add a couple lines, one for your primary and another for your secondary redundant server. Call one `server1`, the other `server2`, and finally, call one `mail`, and set the IP addresses appropriately. It should look something like this:

```
192.168.1.1    server1
192.168.1.2    server2
192.168.1.5    mail
```

Finally, on both your master and slave server, make a folder called `/replicated`, and add the following line to the `/etc/fstab` file:

```
/dev/drbd0    /replicated    ext3    noauto    0    0
```

Expert included.



CONFIGURING DRBD

After you've done that, you have to set up DRBD before moving forward with Heartbeat. In my setup, the configuration file is /etc/drbd.conf, but that can change depending on distribution and compile time options, so try to find the file and open it now so you can follow along. If you can't find it, simply create one called /etc/drbd.conf.

Listing 1 is my configuration file. I go over it line by line and add explanations as comments that begin with the # character. Now, let's test it by starting the DRBD driver to see if everything works as it should. On your command line on both servers type:

```
drbdadm create-md drbd0; /etc/init.d/drbd restart; cat /proc/drbd
```

If all goes well, the output of the last command should look something like this:

```
0: cs:Connected st:Secondary/Secondary ds:Inconsistent/Inconsistent r---  
ns:0 nr:0 dw:0 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0  
    resync: used:0/7 hits:0 misses:0 starving:0 dirty:0 changed:0  
    act_log: used:0/257 hits:0 misses:0 starving:0 dirty:0 changed:0
```

Note: you always can find information about the DRBD status by typing:

```
cat /proc/drbd
```

Now, type the following command on the master system:

```
drbdadm -- --overwrite-data-of-peer primary drbd0; cat /proc/drbd
```

The output should look something like this:

```
0: cs:SyncSource st:Primary/Secondary ds:UpToDate/Inconsistent r---  
ns:65216 nr:0 dw:0 dr:65408 al:0 bm:3 lo:0 pe:7 ua:6 ap:0  
    [>.....] sync'ed: 2.3% (3083548/3148572)K  
    finish: 0:04:43 speed: 10.836 (10,836) K/sec  
    resync: used:1/7 hits:4072 misses:4 starving:0 dirty:0 changed:4  
    act_log: used:0/257 hits:0 misses:0 starving:0 dirty:0 changed:0
```

This means it is syncing your disks from the master computer that is set as the primary one to the slave computer that is set as secondary.

Next, create the filesystem by typing the following on the master system:

```
mkfs.ext3 /dev/drbd0
```

Once that is done, on the master computer, go ahead and mount the drive /dev/drbd0 on the /replicated directory we created for it. We'll have to mount it manually for now until we set up Heartbeat.

PREPARING YOUR SERVICES

An important part of any redundant solution is properly preparing your services so that when the master machine fails, the slave machine can take over and run those services seamlessly. To do that, you have to move not only the data to the replicated DRBD disk, but also move the configuration files.

Let me show you how I've got Sendmail set up to handle the mail and store it on the replicated drives. I use Sendmail for this example as it is one step more complicated than the other services, because even if the machine is running in slave mode, it may need to send e-mail notifications from internal applications, and if Sendmail can't access the configuration files, it won't be able to do this.

On the master machine, first make sure Sendmail is installed but stopped. Then create an etc directory on your /replicated drive. After that, copy your /etc/mail directory into the /replicated/etc and create a symlink from /replicated/etc/mail to /etc/mail.

Next, make a var directory on the /replicated drive, and copy /var/mail, /var/spool/mqueue and any other mail data folders into that directory. Then, of course, create the appropriate symlinks so that the new folders are accessible from their previous locations.

Jon brings years of experience to finding innovative ways to meet his customers' IT challenges. He is a fan of the Rackform iServ R256 with two Intel® Dual-Core Xeon® 5100 series processors because he knows that intense computing environments demand powerful, efficient solutions. Servers based on Intel's 5100 series processors offer an unprecedented combination of power and efficiency, including Intel Virtualization Technology, Intel Active Server Manager, and Intel I/O Acceleration Technology.

When you partner with Silicon Mechanics, you get more than a breakthrough Intel solution — you get an expert like Jon.

 SILICON MECHANICS

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

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 or its subsidiaries in the United States and other countries.

► FEATURE: DRBD IN A HEARTBEAT

Your /replicated directory structure should now look something like:

```
/replicated/etc/mail  
/replicated/var/mail  
/replicated/var/spool/mqueue  
/replicated/var/spool/mqueue-client  
/replicated/var/spool/mail
```

And, on your main drive, those folders should be symlinks and look something like:

```
/etc/mail -> /replicated/etc/mail  
/var/mail -> /replicated/var/mail  
/var/spool/mqueue -> /replicated/var/spool/mqueue  
/var/spool/mqueue-client -> /replicated/var/spool/mqueue-client  
/var/spool/mail -> /replicated/var/spool/mail
```

Now, start Sendmail again and give it a try. If all is working well, you've successfully finished the first part of the setup.

The next part is to make sure it runs, even on the slave. The trick we use is copying the Sendmail binary onto the mounted /replicated drive and putting a symlink to the binary ssmtp on the unmounted /replicated folder.

First, make sure you have ssmtp installed and configured on your system. Next, make a directory /replicated/usr/sbin, and copy /usr/sbin/sendmail to that directory. Then, symlink from /usr/sbin/sendmail back to /replicated/usr/sbin/sendmail.

Once that's done, shut down Sendmail and unmount the /replicated drive. Then, on both the master and slave computers, create a folder /replicated/usr/sbin and a symlink from /usr/sbin/ssmtp to /replicated/usr/sbin/sendmail.

I've had this setup running for about three years now, and it has made the inevitable hardware failures unnoticeable to the company.

After setting up Sendmail, setting up other services like Apache and PostgreSQL will seem like a breeze. Just remember to put all their data and configuration files on the /replicated drive and to create the appropriate symlinks.

CONFIGURING HEARTBEAT

Heartbeat is designed to monitor your servers, and if your master server fails, it will start up all the services on the slave server, turning it into the master. To configure it, we need to specify which servers it should monitor and which services it should start when one fails.

Let's configure the services first. We'll take a look at the Sendmail we configured previously, because the other services are configured the same way. First, go to the directory /etc/heartbeat/resource.d. This directory holds all the startup scripts for the services Heartbeat will start up.

Now add a symlink from /etc/init.d/sendmail to /etc/heartbeat/resource.d.

Note: keep in mind that these paths may vary depending on your Linux distribution.

With that done, set up Heartbeat to start up services automatically on the master computer, and turn the slave to the master if it fails. Listing 2 shows the file that does that, and in it, you can see we have only one line, which has different resources to be started on the given server, separated by spaces.

The first command, server1, defines which server should be the default master of these services; the second one, IPAddr::192.168.1.5/24,

Listing 2. /etc/heartbeat/haresources

```
server1 IPAddr::192.168.1.5/24 datadisk::drbd0 sendmail
```

Listing 3. /etc/heartbeat/ha.cf

```
debugfile /var/log/ha-debug  
logfile /var/log/ha-log  
logfacility local0  
keepalive 2  
deadtime 120  
initdead 120  
serial /dev/ttyS1  
baud 9600  
udpport 694  
udp eth0  
nice_fallback on  
node server1  
node server2
```

tells Heartbeat to configure this as an additional IP address on the master server with the given netmask. Next, with datadisk::drbd0 we tell Heartbeat to mount this drive automatically on the master, and after this, we can enter the names of all the services we want to start up—in this case, we put sendmail.

Note: these names should be the same as the filename for their startup script in /etc/heartbeat/resource.d.

Next, let's configure the /etc/heartbeat/ha.cf file (Listing 3). The main things you would want to change in it are the hostnames of the master/slave machine at the bottom, and the deadtime and initdead. These specify how many seconds of silence should be allowed from the other machine before assuming it's dead and taking over.

If you set this too low, you might have false positives, and unless you've got a system called STONITH in place, which will kill the other machine if it thinks it's already dead, you can have all kinds of problems. I set mine at two minutes; it's what has worked best for me, but feel free to experiment.

Also keep in mind the following two points: for the serial connection to work, you need to plug in a crossover serial cable between the machines, and if you don't use a crossover network cable between the machines but instead go through a hub where you have other Heartbeat nodes, you have to change the udpport for each master/slave node set, or your log file will get filled with warning messages.

Now, all that's left to do is start your Heartbeat on both the master and slave server by typing:

```
/etc/init.d/heartbeat start
```

Once you've got that up and running, it's time to test it. You can do that by stopping Heartbeat on the master server and watching to see whether the slave server becomes the master. Then, of course, you might want to try it by completely powering down the master server or any other disconnection tests.

Congratulations on setting up your redundant server system! And, remember, Heartbeat and DRBD are fairly flexible, and you can put together some complex solutions, including having one server being a master of one DRBD partition and a slave of another. Take some time, play around with them and see what you can discover.■

Pedro Pla (pedropla@pedropla.com) is CTO of the Holiday Marketing International group of companies, and he has more than ten years of Linux experience.

Growing a World of Linux Professionals



We at the Linux Professional Institute believe the best way to spread the adoption of Linux and Open Source software is to grow a world wide supply of talented, qualified and accredited IT professionals.

We realize the importance of providing a global standard of measurement. To assist in this effort, we are launching a Regional Enablement Initiative to ensure we understand, nurture and support the needs of the enterprise, governments, educational institutions and individual contributors around the globe.

We can only achieve this through a network of local "on the ground" partner organizations. Partners who know the sector and understand the needs of the IT work force. Through this active policy of Regional Enablement we are seeking local partners and assisting them in their efforts to promote Linux and Open Source professionalism.

We encourage you to contact our new regional partners listed above.

Together we are growing a world of Linux Professionals.



Stable. Innovative. Growing.

Whether you're a scientist, graphic artist, musician or movie executive, you can benefit from the speed and price of today's high-performance Beowulf clusters.

MAINSTREAM PARALLEL PROGRAMMING

MICHAEL-JON AINSLEY HORE

When Donald Becker introduced the idea of a Beowulf cluster while working at NASA in the early 1990s, he forever changed the face of high-performance computing. Instead of institutions forking out millions of dollars for the latest supercomputer, they now could spend hundreds of thousands and get the same performance. In fact, a quick scan of the TOP500 project's list of the world's fastest supercomputers shows just how far-reaching the concept of computer clusters has become. The emergence of the Beowulf cluster—a computer cluster created from off-the-shelf components and that runs Linux—has had an unintended effect as well. It has captivated the imaginations of computer geeks everywhere, most notably, those who frequent the Slashdot news site.

Unfortunately, many people believe that Beowulfs don't lend themselves particularly well to everyday tasks. This does have a bit of truth to it. I, for one, wouldn't pay money for a version of *Quake 4* that runs on a Beowulf! On one extreme, companies such as Pixar use these computer systems to render their latest films, and on the other, scientists around the world are using them this minute to do everything from simulations of nuclear reactions to the unraveling of the human genome. The good news is that high-performance computing doesn't have to be confined to academic institutions and Hollywood studios.

Before you make your application parallel, you should consider whether it really needs to be. Parallel applications normally are written because the data they process is too large for conventional PCs or the processes involved in the program require a large amount of time. Is a one-second increase in speed worth the effort of parallelizing your code and managing a Beowulf cluster? In many cases, it is not. However, as we'll see later in this article, in a few situations, parallelizing your code can be done with a minimum amount of effort and yield sizeable performance gains.

You can apply the same methods to tackle image processing, audio processing or any other task that is easily broken up into parts. As an example of how to do this for whatever task you have at hand, I consider applying an image filter to a rather large image of your friend and mine, Tux.

A TYPICAL SETUP

The first thing we need is a set of identical computers running Linux, connected to each other through high-speed Ethernet. Gigabit Ethernet is best. The speed of the network connection is one factor that can bring a fast cluster to a crawl. We also need a shared filesystem of some sort and some clustering software/libraries. Most clusters use NFS to share the hard drive, though more exotic filesystems exist, like IBM's General Parallel Filesystem (GPFS). For clustering software, there are a few available choices. The standard these days is the Message Passing Interface (MPI), but the Parallel Virtual Machine (PVM) libraries also should work just fine. MOSIX and openMOSIX have been getting a lot of attention lately, but they

are used primarily for programs that are not specifically written to run on clusters, and they work by distributing threads in multithreaded programs to other nodes. This article assumes you have MPI installed, though the process for parallelizing an algorithm with PVM is exactly the same. If you have never installed MPI, Stan Blank and Roman Zaritski have both written nice articles on how to set up an MPI-based cluster on the *Linux Journal* Web site (see the on-line Resources).

INITIALIZING THE PROGRAM

At the beginning of each MPI program are a few calls to subroutines that initialize communication between the computers and figure out

EURO OSCON™

OPEN SOURCE CONVENTION

OPEN AND CONNECTED

EuroOSCON will be filled to the brim with mindbending demos, provocative keynotes, hands-on practical tutorials, and lots of two-way interactivity. Open technology is at the core of the conference.

Hear from the best speakers from Europe and the world as they engage with you and fellow participants from across the spectrum of technology, business, culture and government in Europe.



18 – 21 SEPTEMBER, 2006
BRUSSELS, BELGIUM

Media

Mobile

Management

Browser/Web

Services

Open Culture

Open Business

Open Source

Open Standards

Open Society

Cybernetics

Register by 7 August and Save

(Use code *euos06ljou* for 10% discount) conferences.oreilly.com/eurooscon

what “rank” each node is. The rank of a node is a number that identifies it uniquely to the other computers, and it varies from 0 to one less than the total cluster size. Node 0 typically is called the master node and is the controller for the process. After the program is finished, you need to make one additional call to finish up the process before exiting. Here’s how it’s done:

```
#include <mpi.h>
#include <stdlib.h>

int main (void) {

    int myRank, clusterSize;
    int imgHeight, lowerBoundY, upperBoundY,
        boxSize;

    // Initialize MPI
    MPI_Init((void *) 0, (void *) 0);

    // Get which node number we are.
    MPI_Comm_rank(MPI_COMM_WORLD, &myRank);

    // Get how many total nodes there are.
    MPI_Comm_size(MPI_COMM_WORLD, &clusterSize);

    // boxSize - the amount of the image each node
    //           will process
    boxSize = imgHeight / clusterSize;

    // lowerBoundY - where each node starts processing.
    lowerBoundY = myRank*boxSize;

    // upperBoundY - where each node stops processing.
    upperBoundY = lowerBoundY + boxSize;

    // Body of program goes here

    // Clean-up and exit:
    MPI_Finalize(MPI_COMM_WORLD);
    return 0;
}
```

This code runs independently on each machine that is part of the process, so the values for lowerBoundY and upperBoundY will vary on each machine. These will be used in the next section.

BREAKING UP THE IMAGE

Applying a filter to an image can take minutes or even hours to complete, depending on the complexity of the filter, the size of the image and the speed of the computer. To fix these problems, we need to get smaller chunks of the image to several computers to help speed up the task. Figure 1 shows common ways of doing this—we’ll cut our image into strips. If we have one large image file, we can do this in C/C++, like this:

```
FILE *imageFile = fopen("image_in.ppm", "rb");

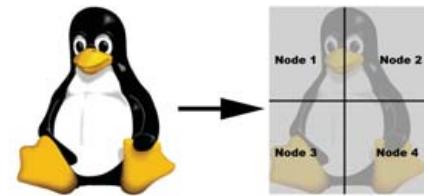
// Safety check.
if (imageFile != NULL) {

    // Read in the header.
    fread(imageHeader, sizeof(char),
          HEADER_LENGTH, imageFile);

    // fseek puts us at the point in the image
    // that this node will process.
    fseek(imageFile, lowerBoundY*WIDTH*3,
          SEEK_SET);
```

```
// Here is where we read in the colors:
// i is the current row in the image.
// j is the current column in the image.
// k is the color, 0 for red, 1 for blue,
// and 2 for green.
for (i=0; i<boxSize+1; i++) {
    for (j=0; j<WIDTH; j++) {
        for (k=0; k<3; k++) {
            fread(&byte, 1, 1, imageFile);
            pixelIndex = i*WIDTH+j+k;
            origImage[pixelIndex] = byte;
        }
    }
}
fclose(imageFile);
```

Break into grids:



Cut into strips:

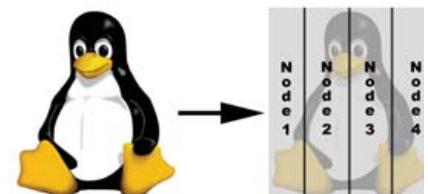


Figure 1. The process of breaking up a problem into smaller chunks is called domain decomposition, and it usually is done as shown here.

APPLYING THE FILTER

Now that we have each node storing a piece of the image for processing, we need to apply the filter to the image. The GIMP Documentation Team has done a good job of describing how to do this using a convolution matrix in the GIMP Documentation. Many image effects—such as sharpen, blur, Gaussian blur, edge detect and edge enhance—have unique matrices that provide the desired effect. The convolution matrix works by studying

each pixel of an image and changing its value based on the values of neighboring pixels. We consider the edge detect matrix in this article, shown in Figure 2.

When we apply this filter to the image, we multiply each pixel by -4 and add the values of the pixels above, below and to the left and right to that. This becomes the new value of the pixel. Because there are zeros in the corners of the

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Figure 2. This matrix represents the edge detect filter. The red square represents the pixel to be considered, and the other numbers represent the neighboring pixels.

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

matrix, we can simplify our program and get better performance by skipping those calculations. Below I've shown how this is done in practice. In Figure 3, I show what the filter does to our image of Tux.

```

for (i=0; i<boxSize; i++) {
    for (j=0; j<WIDTH; j++) {
        if (i>0 && i<(HEIGHT-1) &&
            j>0 && j<(WIDTH-1)) {

            // Now we apply the filter matrix

            // First to the current pixel.
            pixelIndex = i*WIDTH + j;
            r = origImage[pixelIndex];
            g = origImage[pixelIndex+1];
            b = origImage[pixelIndex+2];
            filter_r = -4*r;
            filter_g = -4*g;
            filter_b = -4*b;

            // Next to the left neighbor.
            pixelIndex = i*WIDTH + j - 1;
            r = origImage[pixelIndex];
            g = origImage[pixelIndex+1];
            b = origImage[pixelIndex+2];
            filter_r += 1*r;
            filter_g += 1*g;
            filter_b += 1*b;

            // Next to the right neighbor.
            pixelIndex = i*WIDTH + j + 1;
            r = origImage[pixelIndex];
            g = origImage[pixelIndex+1];
            b = origImage[pixelIndex+2];
            filter_r += 1*r;
            filter_g += 1*g;
            filter_b += 1*b;

            // The neighbor above.
            pixelIndex = (i-1)*WIDTH + j;
            r = origImage[pixelIndex];
            g = origImage[pixelIndex+1];
            b = origImage[pixelIndex+2];
            filter_r += 1*r;
            filter_g += 1*g;
            filter_b += 1*b;

            // The neighbor below.
            pixelIndex = (i+1)*WIDTH + j;
            r = origImage[pixelIndex];
            g = origImage[pixelIndex+1];
            b = origImage[pixelIndex+2];
            filter_r += 1*r;
            filter_g += 1*g;
            filter_b += 1*b;
        }

        // Record the new pixel.
        pixelIndex = i*WIDTH + j;
        filterImage[pixelIndex] = filter_r;
        filterImage[pixelIndex+1] = filter_g;
        filterImage[pixelIndex+2] = filter_b;
    }
}

```

We can mimic the `readImage()` subroutine to create a `writelImage()` subroutine to write the image to disk in chunks.

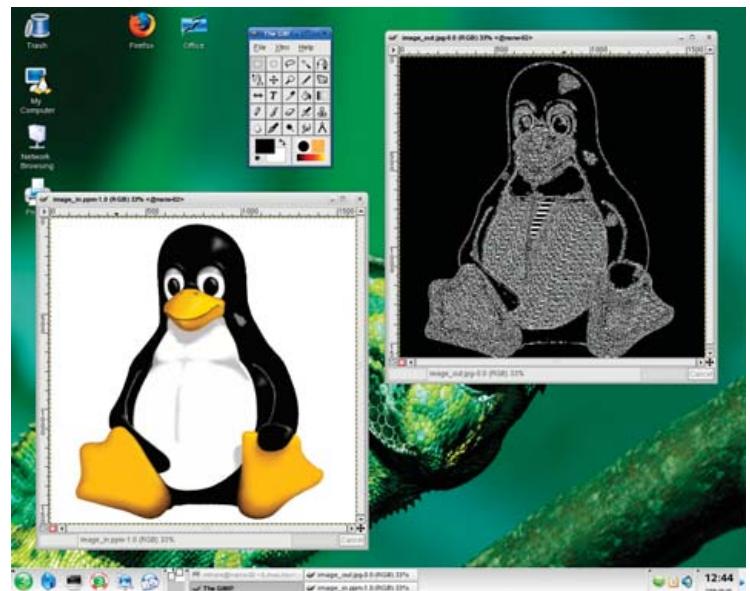


Figure 3. On the left is our original image of Tux, and on the right is the image after we apply the edge detect filter.

COMPILING AND RUNNING YOUR CODE

Both of the popular MPI distributions—LAM and MPICH—include wrapper scripts to allow users to compile their programs easily with the required MPI libraries. These wrapper scripts allow you to pass parameters to GCC like you always do:

- ▶ `mpicc`: for C programs
- ▶ `mpi++`: for C++ programs
- ▶ `mpif77`: for FORTRAN 77 programs

Use `mpirun` to execute your newly compiled program. For example, I compiled my code with the command `mpicc -O3 -o parallel parallel.c` and then executed it with `mpirun n0 ./parallel`. The `n0` signifies that the program is to run on node 0 only. To run it on additional nodes, you can specify a range like `n0-7` (for eight processors) or use `mpirun C` to signify that the program is to run on all available nodes.

PERFORMANCE GAINS

So, with only a few simple MPI calls, we have parallelized our image filter algorithm very easily, but did we gain any performance? There are a few ways that we can gain performance. The first is in terms of speed, and the second is in terms of how much work we can do. For example, on a single computer, a 16,000 x 16,000 pixel image would require an array of 768,000,000 elements! This is just too much for many computers—GCC complained to me that the array was simply too big! By breaking the image down as we did above, we can ease memory requirements for our application.

I tested the code above on a 16-node Beowulf cluster running Fedora Core 1. Each node had 1.0GB of RAM, a 3.06GHz Pentium 4 processor and was connected to the other nodes through Gigabit Ethernet. The nodes also shared a common filesystem through NFS. Figure 4 shows the amount of time required to read in the image, process it and write it back to disk.

From Figure 4, we can see that parallelizing this image filter sped things up for even moderately sized images, but the real performance gains happened for the largest images. Additionally, for images of more than 10,000 x 10,000 pixels, at least four nodes were required due to memory constraints. This figure also shows where it is a good idea to parallelize the code and where it was not. In particular, there was hardly any

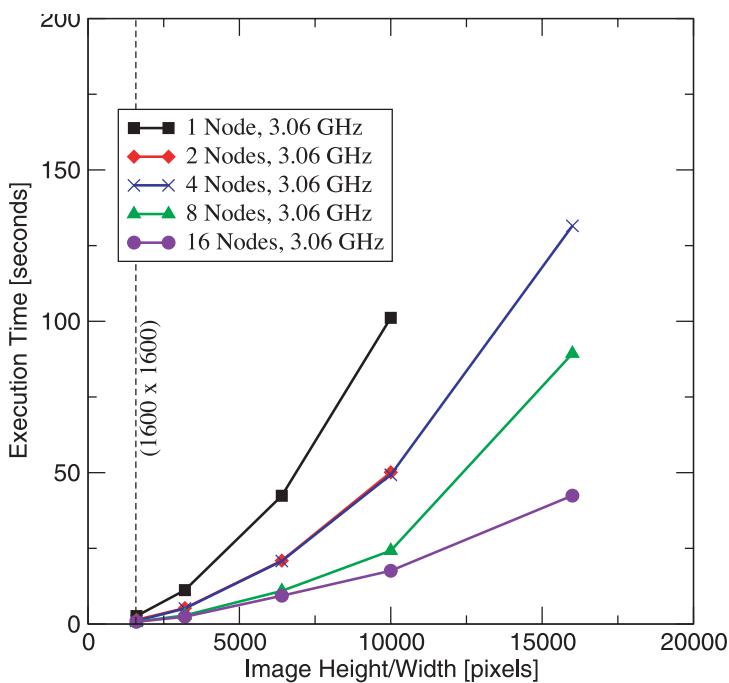


Figure 4. Shown here are the times that the program took for various image sizes and various cluster sizes. Image sizes ranged from 1,600 x 1,600 pixels to 16,000 x 16,000 pixels. A minimum of four nodes was required for the largest image.

difference in the program's performance from 1,600 x 1,600 pixel images to about 3,200 x 3,200 pixel images. In this region, the images are so small that there is also no benefit in parallelizing the code from a memory standpoint, either.

To put some numbers to the performance of our image-processing program, one 3.06GHz machine takes about 50 seconds to read, process and write a 6,400 x 6,400 image to disk, whereas 16 nodes working together perform this task in about ten seconds. Even at 16,000 x 16,000 pixels, 16 nodes working together can process an image faster than one machine processing an image 6.25 times smaller.

POSSIBLE MAINSTREAM APPLICATIONS

This article demonstrates only one possible way to take advantage of the high performance of Beowulf clusters, but the same concepts are used in virtually all parallel programs. Typically, each node reads in a fraction of the data, performs some operation on it, and either sends it back to the master node or writes it out to disk. Here are four examples of areas that I think are prime candidates for parallelization:

1. Image filters: we saw above how parallel processing can tremendously speed up image processing and also can give users the ability to process huge images. A set of plugins for applications such as The GIMP that take advantage of clustering could be very useful.
2. Audio processing: applying an effect to an audio file also can take a large amount of time. Open-source projects such as Audacity also stand to benefit from the development of parallel plugins.
3. Database operations: tasks that require processing of large amounts of records potentially could benefit from parallel processing by having each node build a query that returns only a portion of the entire set needed. Each node then processes the records as needed.
4. System security: system administrators can see just how secure their users' passwords are. Try a brute-force decoding of /etc/shadow using a Beowulf by dividing up the range of the checks across several machines. This will save you time and give you peace of mind (hopefully) that your system is secure.

FINAL REMARKS

I hope that this article has shown how parallel programming is for everybody. I've listed a few examples of what I believe to be good areas to apply the concepts presented here, but I'm sure there are many others.

There are a few keys to ensuring that you successfully parallelize whatever task you are working on. The first is to keep network communication between computers to a minimum. Sending data between nodes generally takes a relatively large amount of time compared to what happens on single nodes. In the example above, there was no communication between nodes. Some tasks may require it, however. Second, if you are reading your data from disk, read only what each node needs. This will help you keep memory usage to a bare minimum. Finally, be careful when performing tasks that require the nodes to be synchronized with each other, because the processes will not be synchronized by default. Some machines will run slightly faster than others. In the sidebar, I have included some common MPI subroutines that you can utilize, including one that will help you synchronize the nodes.

In the future, I expect computer clusters to play an even more important role in our everyday lives. I hope that this article has convinced you that it is quite easy to develop applications for these machines, and that the performance gains they demonstrate are substantial enough to use them in a variety of tasks. I would also like to thank Dr Mohamed Laradji of The University of Memphis Department of Physics for allowing me to run these applications on his group's Beowulf cluster.■

SOME ESSENTIAL AND USEFUL MPI SUBROUTINES

There are more than 200 subroutines that are a part of MPI, and they are all useful for some purpose. There are so many calls because MPI runs on a variety of systems and fills a variety of needs. Here are some calls that are most useful for the sort of algorithm demonstrated in this article:

1. `MPI_Init((void*) 0, (void*) 0)` — initializes MPI.
2. `MPI_Comm_size(MPI_COMM_WORLD, &clstSize)` — returns the size of the cluster in clstSize (integer).
3. `MPI_Comm_rank(MPI_COMM_WORLD, &myRank)` — returns the rank of the node in myRank (integer).
4. `MPI_Barrier(MPI_COMM_WORLD)` — pauses until all nodes in the cluster reach this point in the code.
5. `MPI_Wtime()` — returns the time since some undefined event in the past. Useful for timing subroutines and other processes.
6. `MPI_Finalize(MPI_COMM_WORLD)` — halts all MPI processes. Call this before your process terminates.

Resources for this article: www.linuxjournal.com/article/9135.

Michael-Jon Ainsley Hore is a student and all around nice guy at The University of Memphis working toward an MS in Physics with a concentration in Computational Physics. He will graduate in May 2007 and plans on starting work on his PhD immediately after.

MILLE-XTERM and LTSP

MILLE-XTERM provides a scalable infrastructure for massive X-terminal deployment.

FRANCIS GIRALDEAU, JEAN-MICHEL DAULT AND BENOIT DES LIGNERIS

Linux-based X terminals are well known for making computing affordable, for giving a second life to old hardware and for lightening administrative burdens. If you ever have toyed with the idea of using Linux X terminals, you are probably familiar with the Linux Terminal Server Project (LTSP), described in Figure 1. An LTSP server is perfectly suited for small workgroups or classrooms. However, in order to deploy a greater number of terminals, say thousands of them, the current LTSP model encounters scalability problems.

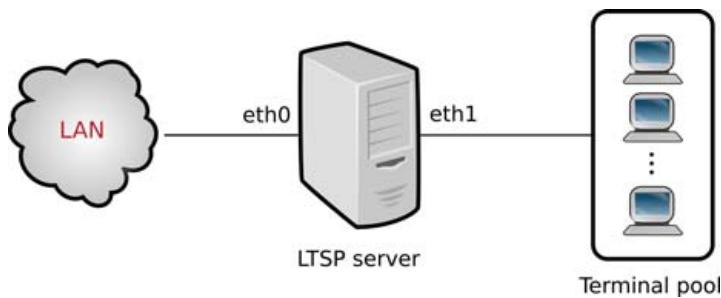


Figure 1. LTSP Simple Network Configuration

The main goals of the MILLE-XTERM Project are:

- Unlimited scalability.
- Centralized X-terminal management.
- Enhanced user experience.

The MILLE-XTERM Project applied clustering concepts to the X-terminal infrastructure to achieve these goals.

The MILLE Project is funded by Canadian public agencies and school districts in the province of Quebec. MILLE means "Free Software Infrastructure Model for Education" and is targeted at educational institutions. It is composed of four subprojects: a portal (based on uportal), an open-source middleware stack, a CD with free software for Windows/Mac and, finally, MILLE-XTERM (the object of the present article).

Overview of a MILLE-XTERM Cluster

The solution entails centralizing servers in a secure, air-conditioned computer room to form a cluster of terminal servers. The cluster has four major components, as shown in Figure 2. The first is the boot server, which provides DHCP and TFTP services and serves as a base system optimized for the terminals via NFS. Next comes the configurator, which generates the `lts.conf` configuration file from an SQL database. The terminal then queries the load balancer, which in turn seeks out the cluster's least-loaded application server. The chosen application server login screen then appears, and after a successful authentication, the user can start using the desktop, browser, office suite and other applications.

MILLE-XTERM relies on central file and authentication services that provide users with the same account and file on every application server. The open-source choice is NFS for users' home directories and

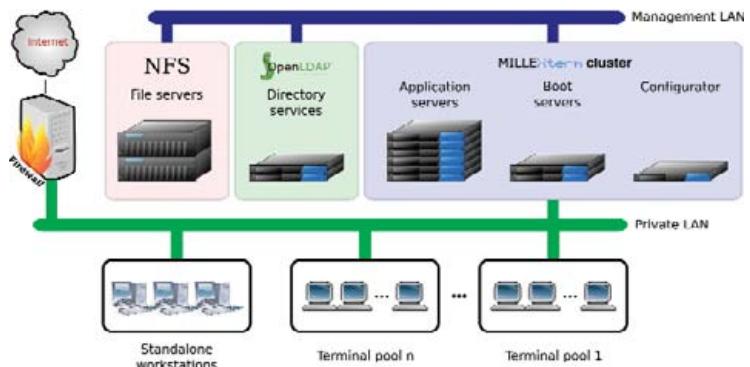


Figure 2. Clustered LTSP Network Diagram

OpenLDAP for the directory service. It also can be integrated into a Novell or Windows environment with additional configuration.

Unlike LTSP, there is no need for a separate network dedicated to terminals. They can share a LAN with other PCs. However, a reliable network infrastructure is crucial. With usual usage, each terminal generates an average of 1Mb/sec of X11 traffic. Low-end hubs should be avoided; managed switches with full-duplex capabilities really make a difference.

Unlike LTSP, each component is built from RPMs; system administrators easily can add features and local applications with standard package manager tools. The init scripts from the distribution are replaced by standard LTSP scripts. We are currently using Mandriva 2006 as the base distribution, though other distributions can be supported.

Boot Server for Terminal

The boot server is mainly a read-only NFS server. Optionally, it can provide DHCP, TFTP and NBD swap services. The `xtermroot` contains a base system and an X server. With more than one boot server, it is easy to rsync the `xtermroot` periodically on each boot server. The terminals then boot in a uniform way, whichever boot server they use.

Getting Configuration under Control

Diskless terminals need a way to store configuration data, such as screen resolution and available printers. Under LTSP, a central file called `lts.conf` stores the configuration of terminals and has to be edited manually. With thousands of terminals, you need a hierarchical database—that's the purpose of the configurator.

This component is written in PHP and has two interfaces. The first is dedicated to terminals. During the boot process, the terminal requests its configuration from the server using its MAC address as a parameter. The server generates the corresponding configuration and sends it to the terminal in the standard `lts.conf` format. A wrapper around the `getltscfg` command ensures backward compatibility with the other LTSP scripts.

The other interface lets administrators manage the configuration of the terminals via a Web browser. Administrators can organize terminals hierarchically by groups and apply configurations according to specific criteria, such as location or hardware type. But the configurator serves yet

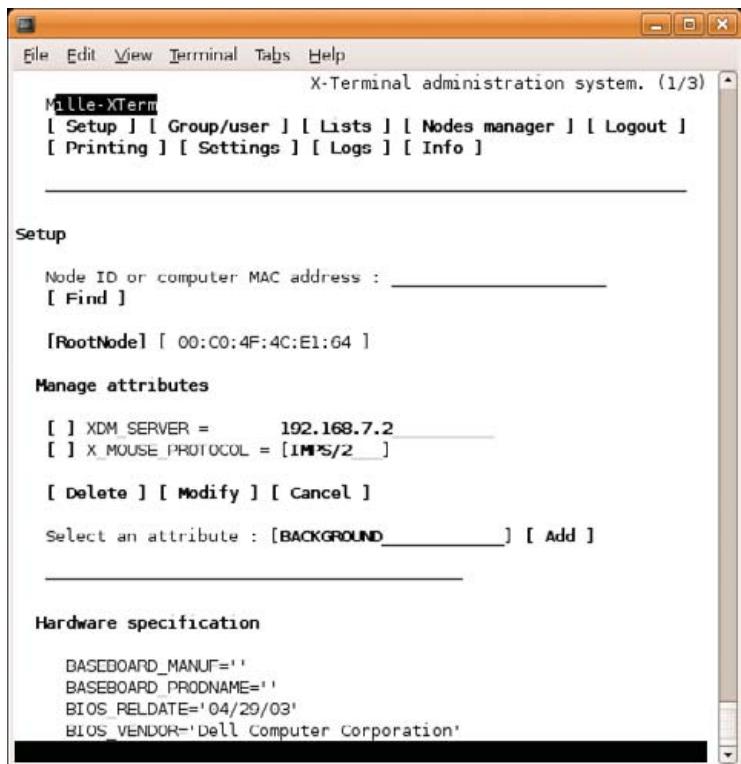


Figure 3. Xterm-Based Administration System

another function. It is designed to work with links, a console text browser, as shown in Figure 3. The terminal can boot in a special admin mode that does not require running the X server. To boot in this mode, the option mode=admin is appended to the kernel options in the bootloader configuration. Then, links is launched with the terminal configurator URL and MAC. The administrator can change the terminal settings directly. When complete, the terminal reboots and receives its new configuration.

The configurator also is useful for building terminal inventories. Hardware information is sent to the configurator during the boot process. Administrators can generate reports regarding the state of the terminals. Also, every connection to the configurator is logged and then can be analyzed to determine terminal usage, user login information and much more. You know how managers like reports!

Load Balancing between Application Servers

When a terminal boots, it requests a display from the application server. To dispatch users on available application servers, MILLE-XTERM provides a load balancer. The first version of the load balancer (proof of

BOOT METHODS

MILLE-XTERM supports different booting methods: CD-ROM, local hard drive, Flash disk, Etherboot or PXE. Each boot method has its advantages and drawbacks.

PXE and Etherboot rely on TFTP to transmit the initial file used for the boot process. It simplifies the deployment, as no configuration is stored on the terminal itself. However, simultaneously booting up hundreds of clients via TFTP can result in transmission errors and, consequently, boot problems.

An alternative is to use a 16MB IDE Flash disk that holds the kernel, the initrd and grub. The Flash disk is updated automatically as the terminal boots up. The disk is used only at startup and contains no moving parts.

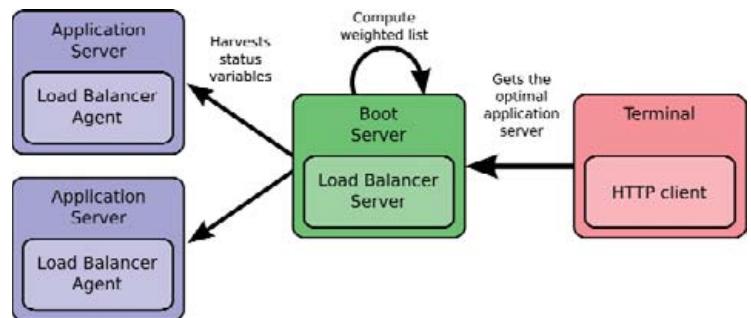


Figure 4. Load Balancing the LTSP Cluster

concept) required five lines of PHP and returned a random address from a static list of application servers. Although simple, this approach had some drawbacks. First, an off-line server should be removed from the list and not be returned to the terminals. And, to provide reliable load balancing, several factors, such as number of processors, speed and load average have to be taken into account. Therefore, a much more robust and complete Python system has replaced the initial prototype (Figure 4). The load balancer agent runs on every application server, collecting data on the state of the application server and waiting for load-balancer server requests. The balancer is also a Python script that runs on the boot server. It contacts each load-balancer agent to determine its state and computes a weight for each server. A greater weight indicates that the server is less loaded and will be selected more often statistically to accept new users. A terminal request for an application server will then prompt the load-balancer server to get a randomly

Linux Laptops

Starting at \$990

DON'T BE SQUARE! GET CUBED!



R³ Technologies™
Working Hard to Bring Technologies to Life

309.34.CUBED
shoprcubed.com

chosen application server in the weighted list.

Let's examine a concrete example: three application servers and two boot servers. Install the mille-xterm-lbagent package on each application server, and install mille-xterm-lbserver on each boot server. Make sure that the respective services are started, lbagent and lbserver. Add one node entry for each application server in the file /etc/mille-xterm/lbsconfig.xml:

```
<?xml version="1.0"?>
<lbsconfig>
<nodes>
<group default="true" name="PROD">
<node address="http://10.0.0.1:8001" name="xapp1"/>
<node address="http://10.0.0.2:8001" name="xapp2"/>
<node address="http://10.0.0.3:8001" name="xapp3"/>
</group>
</nodes>
```

Copy this file on every boot server. Fire up a browser and enter the URL of the load balancer to see it in action. By default, lbserver listens on port 8008, so don't forget to append the port to the URL: http://localhost:8008/. IP addresses of the chosen application server will be displayed. Press the refresh button to get a new IP and you're set!

Getting the Best User Experience

The MILLE-XTERM Project focuses on user experience. A MILLE-XTERM X terminal has to provide the same functions as a regular workstation. In order to achieve this goal, extensive desktop personalization is required. Here are some of the most important topics:

- Sound support: because the applications run on the server and the sound card is on the terminal, using /dev/dsp doesn't work. Solutions such as Esound, Arts, Nas and others have been developed over time. We have found that Arts can make a machine unstable (for example, TiMidity++ takes up to 99% of the CPU) and that very few applications support Nas. In our experience, the only workable solution is Esound (esd). GNOME support is native, and KDE, which uses Arts, has an option to use esd as its back end. Major applications, such as Flash Player, RealPlayer and SDL-based games, use it natively. When users first log in to the system, a script automatically creates configuration files to use esd. However, there is still work to do. For example, Audacity does not yet support esd.
- Video support: with Windows, people usually use QuickTime, RealPlayer or Windows Media Player. They expect these essential packages to work on an X terminal. This is a bit of a challenge, as most distributions do not provide these players and appropriate codecs. Those familiar with the command-line swear by MPlayer, and others swear at it. A Mozilla plugin called mplayerplug-in solves the problem and neatly embeds MPlayer in the Web browser with Play, Pause and Forward buttons. MPlayer also can use Windows DLLs optionally to play popular file formats, but as with all proprietary programs, there are many licensing issues to consider.
- Local applications: some applications simply cannot operate with the standard client/server scenario. For example, if users want to control the volume of the terminal sound card, they must run Aumix or Alsamixer locally on the terminal. Also, running the media player directly on the terminal is better than streaming 30 frames per second over the network, as this can quickly clog the network. In these instances, these applications run better when bunched locally on the terminal. To that end, a wrapper script called runlocal connects to the terminal with the user key via SSH and mounts the user's home directory via sshfs, allowing the application to access user files and settings.

■ Application configuration: most applications will create their own configuration files when first launched. In many cases, the default works well. In other cases, they need to be modified for use on an X terminal. For instance, several applications use the X-server memory as a cache memory. Although this is very efficient on a Linux workstation, it can cause an X-terminal crash when the memory used by the X server is bigger than the RAM of the terminal. An effective way to diagnose problems with X-server memory consumption is to use the xrestop tool. For KDE users, this kind of memory problem may occur when copying/pasting a large drawing with Klipper enabled. The only solution at this point is to disable Klipper. Here's a sample Firefox configuration that disables memory and disk cache:

```
// File /opt/firefox/greprefs/xterm.js
pref("general.config.obscure_value", 0);
pref("general.config.filename", "firefox.cfg");

// File /opt/firefox/firefox.cfg
lockPref("browser.cache.disk.enable", false);
lockPref("browser.cache.memory.enable", false);
lockPref("browser.sessionhistory.max_total_viewers", 0);
```

- Personalization: many projects are focusing on language issues, such as internationalization (i18n) and localization (l10n). However, still no personalization layer (p13n) provides an easy way to configure icons on the desktop, menus, browser preferences, bookmarks, backgrounds, screensavers, default applications and so forth. Work has begun in the freedesktop.org project and with Sabayon and Kiosk, for instance, but it remains far from complete and covers only KDE and GNOME. In the meantime, we use a set of homemade scripts and config files to configure the desktop according to an organization's needs.
- Global infrastructure integration: to be successful, the system must be integrated fully into the existing infrastructure (directory server, files and printing). Take printing, for example. If a school district has 50 schools and more than a thousand different printers, how do you select which ones to display in Firefox or OpenOffice.org? We solved the challenge with a simple wrapper that intercepts calls to the CUPS shared library in order to apply a filter based on the user location. The printer list and the default printer for the terminal is stored in the configurator database.

Virtualization of Components with Linux-VServers

For security and administrative considerations, making isolated MILLE-XTERM components is possible. To gain the benefits of virtualization without performance drawbacks, Linux-VServer is the perfect alternative, although a few specific configurations are needed to install MILLE-XTERM inside a Linux-VServer. When installing a boot server in a vserver, it is not feasible to generate the initrd, unless the vserver has the CAP_SYS_ADMIN property set. The solution is to use a chroot on the host. Also, a user-space NFS server is used instead of the regular kernel-based nfsd. Finally, GDM on the application server will try to launch X inside a vserver, which is not needed. To correct this, append the --no-console option to the init script and it will listen only for network requests with no local host display.

When mastered, these few tricks allow you to add or remove application servers, copy existing application servers, back up and update them, and when satisfied with the changes, put them into production and duplicate them throughout the cluster, thereby elevating manageability to a higher level.

Future Developments

MILLE-XTERM can go further in a number of ways—beginning with security, or the lack thereof, as is the case of the XDMCP protocol.

Advertiser Index

For advertising information, please contact our sales department at 206-782-7733 ext. 2 or ads@ssc.com. www.linuxjournal.com/advertising

You can try it at home. Start an X session with Xnest and capture packets with ethereal. The following filter lets you view every keystroke typed:

```
x11.eventcode == 2
```

You could solve the problem with a local secure display manager that creates an SSH tunnel to encrypt the X11 traffic. Another possibility is to use OpenVPN between the terminal and the application server.

Almost every component of the MILLE-XTERM Project should be highly available. Work is in progress for the configurator (using slony replication for the PostgreSQL database). The boot servers (as well as the load balancer) will follow in order to have transparent failover (this can be achieved easily because their main functionality is as a read-only NFS server).

Optimizing the X protocol in order to save bandwidth is another interesting development. One can then use an X terminal with a simple broadband Internet connection. The next step for Linux terminals is NX/FreeNX. Last year, *Linux Journal* devoted five articles to the topic. NX clients would run locally on terminals, which would then require them to be added to the xtermroot in order to work.

Conclusion

Currently, more than 800 terminals are deployed with MILLE-XTERM at the Laval School District (one of the founders of the MILLE Project), and the plan is to deploy more than 1,000 additional terminals yearly (up to 75% of the existing computers will become X terminals).

We strongly believe that Linux terminals are the key solution that will allow school districts to provide a low-cost/high-quality desktop experience. With a cluster of Linux terminal servers, children can access the software they need to learn, create and be part of the Linux revolution.

Acknowledgements

The authors would like to thank the founders of the MILLE Project as well as the early adopters of the MILLE-XTERM solution:

- Laval School District (www.cslaval.qc.ca): 800 terminals and still counting.
- Mille Iles School District (www.cssmi.qc.ca): 300 terminals and still counting.
- Grandes Seigneuries School District (www.csdfs.qc.ca): 100 terminals—pilot project.
- Coeur des Vallées School District (www.cscv.qc.ca): 75 terminals—pilot project.
- Affluents School District (www.csaffluents.qc.ca): pilot project. ■

Resources for this article: www.linuxjournal.com/article/9134.

Francis Giraldeau is an electrical engineer from the Université de Sherbrooke. He works for Revolution Linux while he completes his MSc degree in computer science. He has been devoting time and energy to the MILLE-XTERM Project for three years now. He can be reached at francis.giraldeau@revolutionlinux.com.

Jean-Michel Dault (jmdault@revolutionlinux.com) started his first Internet provider in 1994 using Linux. After a five-year stint at Mandriva, he is now cofounder and CTO of Revolution Linux.

Benoit des Ligneris completed his PhD in Physics at the Université de Sherbrooke where he developed his expertise in large systems (clusters) and scientific computing. He has been the chairman of the OSCAR (Open Source Cluster Application Resources) Project. He is now CEO of Revolution Linux.

Advertiser	Page #	Advertiser	Page #
ABERDEEN, LLC www.aberdeencorp.com	61	LOGIC SUPPLY INC www.logicsupply.com	87
ACMA www.acma.com	39	LPI www.lpi.org	71
AML www.amlt.com	65	MBX www.mbx.com	21
APPRO HPC SOLUTIONS appro.com	C2	MICROWAY, INC. www.microway.com	C4, 67
ASA COMPUTERS www.asacomputers.com	41, 54	MIKROTIK www.routerboard.com	7
CARI.NET www.cari.net	85	MONARCH COMPUTERS www.monarchcomputer.com	10, 11
CIARA TECHNOLOGY www.ciara-tech.com	8, 9	NPULSE NETWORKS www.npulsenetworks.com	45
CONCURRENT COMPUTER CORPORATION www.cccur.com	89	O'REILLY EURO OSCON conferences.oreilly.com/eurooscon	73
CORAID, INC. www.coraid.com	27	OPEN SOURCE STORAGE www.opensourcestorage.com	13
COYOTE POINT www.coyotepoint.com	3	OPERA SOFTWARE www.opera.com	19
CYCLADES, AN AVOCENT COMPANY www.cyclades.com	1	PENGUIN COMPUTING www.penguincomputing.com	49
EMAC, INC. www.emacinc.com	93	POLYWELL COMPUTERS, INC. www.polywell.com	59
EMPERORLINUX www.emperorlinux.com	15	THE PORTLAND GROUP www.pgroup.com	29
FAIRCOM CORPORATION www.faircom.com	35	RACKSPACE MANAGED HOSTING www.rackspace.com	C3
GENSTOR SYSTEMS, INC. www.genstor.com	43	R CUBED TECHNOLOGIES www.rcubedtech.com	79
HPC SYSTEMS, INC. www.hpcsystems.com	31	SERVERS DIRECT www.serversdirect.com	51
HURRICANE ELECTRIC www.he.net	25	SILICON MECHANICS www.siliconmechanics.com	63, 69
IRON SYSTEMS www.ironsystems.com	90	SUPERMICRO www.supermicro.com	53
JTL NETWORKS www.jtlnet.com	6	TEAMHPC www.teamhpc.com	37
LANTRONIX www.lantronix.com	23	TECHLOGIC SYSTEMS www.embeddedx86.com	91
LAYER 42 NETWORKS www.layer42.net	83	THINKMATE www.thinkmate.com	17
LINUX JOURNAL www.linuxjournal.com	75, 84, 88	TYAN COMPUTER USA www.tyan.com	5
LINUXWORLD UK www.linuxworldexpo.co.uk	95	ZT GROUP INTERNATIONAL www.ztgroup.com	33

64-Bit JMP for Linux

64-bit Linux represents a milestone in JMP statistical computing history. ERIN VANG

The world's largest privately owned software company, SAS, was cofounded in 1976 by Dr James Goodnight and John Sall. They continue to run the company as CEO and Executive Vice President. Sall is also chief architect of SAS's statistical discovery software called JMP (pronounced "jump"), which he invented for the Macintosh in the late 1980s. It is a desktop statistical analysis program using exploratory graphics to promote statistical discovery. JMP was released for Windows in 1995 and has been available for 32-bit Linux since 2003.

SAS's version 6.1 release of JMP later in 2006 harnesses the vast computational power of 64-bit Linux, which is not only exciting news for JMP and Linux, but is also a milestone in statistical computing.

To understand the importance of a 64-bit version of JMP, let us contemplate the purpose and history of statistical analysis.

Statistics Simplified

Ultimately, the purpose of statistics is to make sense out of too much information. For example, the only possible way to digest the results of the United States census data every ten years, with its dozens of measurements on 275-million people, is by reducing it to statistical conclusions, such as the average household income by county and median age by city or neighborhood. Nobody could possibly look at the raw census data and draw a meaningful conclusion beyond “the United States has a large, diverse population”.

The problem is that there are hundreds and thousands of statistical measures—in fact, SAS has already spent 30 years extending and refining its analytical capabilities and doesn’t see any end in sight. Learning what techniques to use for which real-world situations can take years, and developing the insights to proceed effectively from raw data to knowledge can take a lifetime. This is what led John Sall to develop JMP in the late 1980s. Inspired by the way the Macintosh made desktop computing accessible to a whole new audience by introducing a graphical user interface, Sall realized he could make statistics accessible to a wider audience by making the analysis process visual.

Comprehending the meaning buried in pages of statistical test results—p-values, standard deviations, error terms, degrees of freedom and on and on—is a mind-boggling task even for experts, but Sall knew that just about anyone could look at a well-drawn graph and understand things about his or her data. JMP always leads every analysis with graphs, so that researchers needn’t waste time poring over statistics when those graphs make it intuitively obvious whether they are on the right analysis path or not. JMP also groups related analyses together and presents them in the order a researcher would need them in the course of a sound data exploration process. Researchers do not have to wrack their brains to remember which procedure might be helpful next. Instead, JMP provides the tools that are appropriate at each stage. Further, all of JMP’s graphs and data tables are dynamically linked, so that users can point and click to select points in a graph or bars in a histogram and instantly see where those points are represented in all other open graphs and data tables.

A Calculating Idea

Setting aside for a moment what it takes to understand statistics, consider what it takes to calculate statistics. For a researcher to compute a

standard deviation on thousands of observations using only a pencil and paper could take weeks or months.

When he created SAS in the early 1970s, Jim Goodnight's idea was to store all that data in a file and then write procedures that could be used and reused to compute statistics on any file. It's an idea that seems ludicrously simple today, but it was revolutionary at the time. The agricultural scientists using SAS could perform calculations over and over again on new data without having to pay for computer scientists to write and rewrite programs. Instead of taking weeks, these computations took hours. Fast-forward 30 years, and modern statistical software can do these calculations on hundreds of thousands of rows, instantaneously.

When it took months to compute simple descriptive stats, researchers often didn't get much further before they'd burned through their grant money. Now that the basics take seconds, researchers can dig much deeper, and thus the science and practice of statistics have evolved along with computing power.

64-bit Linux Empowers JMP to Solve Problems of Greater Magnitude

For the last decade, desktop computing has been built on operating systems such as Windows, Linux and Mac that rely on 32-bit memory addressing. Accordingly, desktop applications have operated within the computational limits implied by this architecture. In practical terms, this meant statistical programs like JMP that load the entire dataset into RAM before performing any computations were limited to about a million rows of data. They couldn't handle the large-scale problems

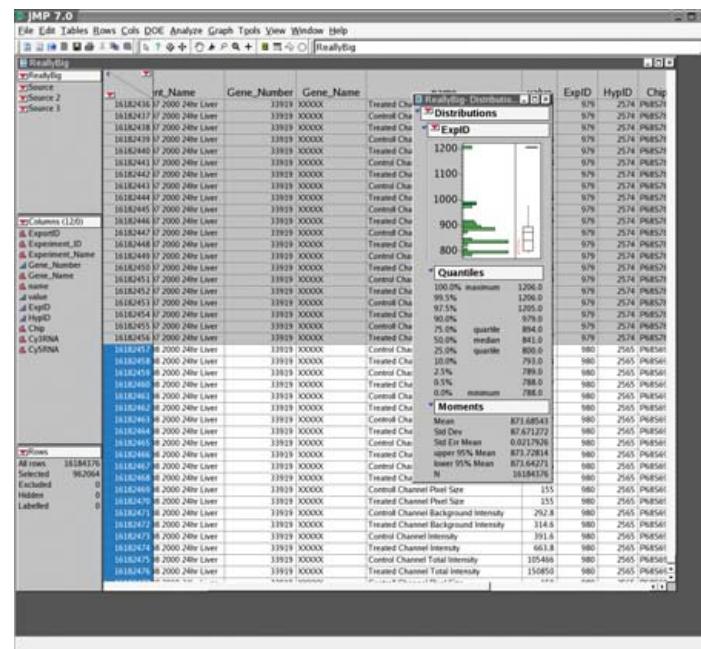


Figure 1. JMP Main Window Showing a Distribution for Genetics Data

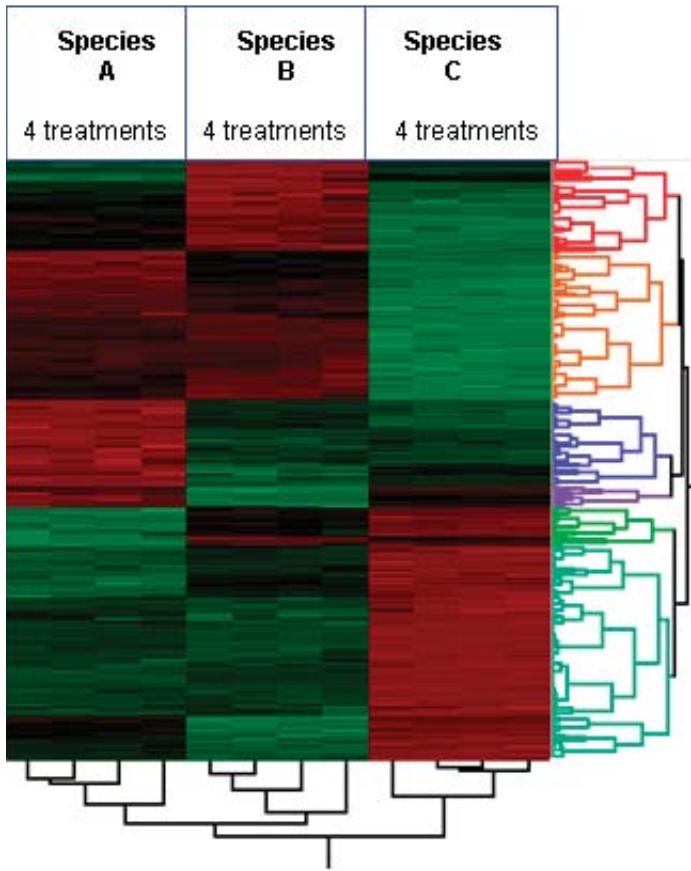


Figure 2. Hierarchical Cluster Diagram or "Heatmap" Showing Gene Expression, Protein and Metabolite Production Patterns

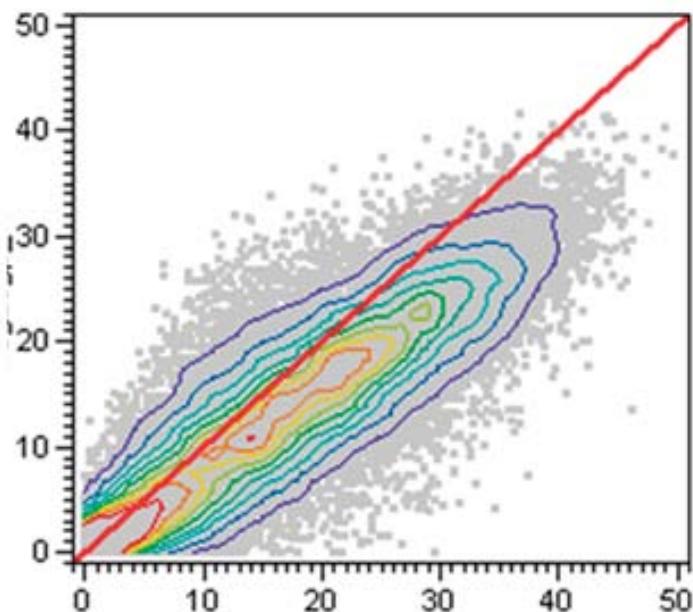


Figure 3. Bivariate Fit with Nonparametric Density Contours Representing Density Levels

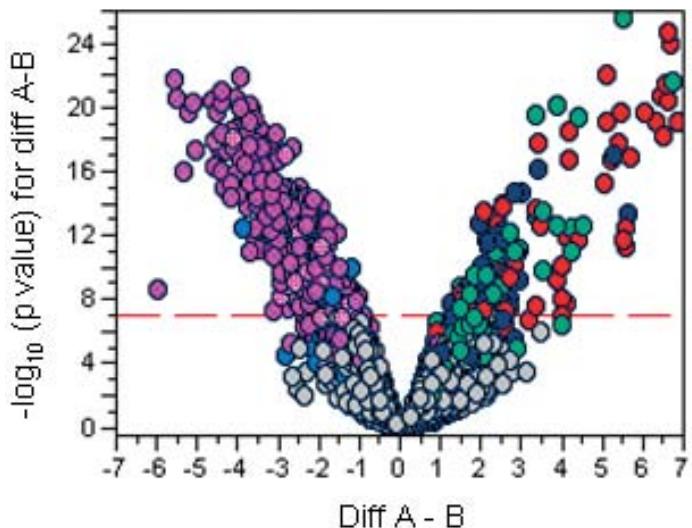


Figure 4. Bivariate Fit Volcano Plot for Comparing Gene Expression

confronting researchers today. Geneticists are probing 3-billion base pairs of DNA. Semiconductor manufacturers are squeezing millions of transistors onto ever-tinier chips. Pharmaceutical companies comb through thousands of potentially therapeutic properties on countless



LAYER 42™

- Redundant UPS and generator
- Nationwide network
- Free tech support

2U 256kbps ~80GB \$60/mo.	4U or Mid-tower 256kbps ~80GB \$80/mo.
1/4 Rack 512kbps (14U) ~165GB \$200/mo.	1/2 Rack 1mbps (28U) ~330GB \$350/mo.

www.layer42.net
 All prices include 100Mbps port, Firewall,
 24x7 Monitoring and DNS hosting

408-450-5740 2312 Walsh Ave., Santa Clara, CA 95051

known and theoretical compounds.

Dr Richard C. Potter, Director of Research and Development for JMP Product Engineering, was responsible for porting JMP from Macintosh to Windows and later from Windows to Linux, in collaboration with Paul Nelson, lead Linux System Developer. Potter says:

JMP's 64-bit Linux release lifts this limit dramatically. Now JMP can move beyond the confines of the 32-bit addressing memory limit to a theoretical limit of 16 exabytes, which would allow JMP to work on two-billion rows of data. The 64-bit Linux release of JMP is also multithreaded, and the size and complexity of the problems someone can solve using JMP is mind-boggling.

Design of Experiments Revolutionized the Science of Statistics

One particular analytical strength of JMP sets it clearly apart from all other stats packages, open source or otherwise: its capabilities for design of experiment (DOE). The idea of experimental design goes beyond the traditional statistical concept of trying to learn from data that has already been collected, to an idea of planning how best to learn more by designing and running experiments.

Suppose you make tires by cooking various ingredients into rubber, combining that rubber with steel and other materials, then forming the rubber into tire-shaped molds of various dimensions and tread patterns. You might judge the success of those tires by measuring their traction in a variety of driving conditions, the range of operating

temperatures and speeds they can endure, and the length of their usable lifetimes. You would be attempting to solve an impossibly complex problem. You would be trying to optimize at least four response measurements of varying importance and interactions that depend on an infinitely variable mixture of ingredients and countless combinations of cooking temperatures and times, molding pressures and times, tire dimensions and tread patterns. If you tried to optimize this manufacturing process using traditional methods, trying out every imaginable combination, you would be working for centuries and expending more natural resources than you could hope to collect.

Design of experiments offers a better way. By running a representative set of experiments that span the space of possibilities and using statistical modeling methods to interpolate and extrapolate those results, researchers can reduce the size of problems to something manageable.

The Next Level of Design of Experiments

The problem is that analysts always had to leaf through books searching for a design model that resembled the problem they were trying to solve. In practice, they would have to use a model that was sort of like their problem, that sort of handled their conditions and that sort of modeled the behaviors of their system. To make matters worse, most of these "canned designs" required huge numbers of runs. If a run costs a few cents and takes minutes, that's no problem, but if your experiment involves building or changing a multimillion-dollar semiconductor fabrication complex or shutting down a thousand-unit-per-minute assembly line, you cannot hope to do all the runs it would take to get meaningful results.

JMP's DOE has allowed blue-chip customers to discover million-dollar annual savings or profit opportunities in mere weeks.

FREE NEWSLETTER!

Wish you could get the latest from *LJ* more than once a month? You can—sign up today for **LJ's weekly e-mail newsletter**.

Each week the *LJ* newsletter features great tech tips, links to web-only articles, and news on the latest events in the Linux market.

**Sign up for the *LJ* e-mail newsletter now at
<http://www.linuxjournal.com/>**

JMP takes DOE to a whole new level by providing unique, powerful custom design capabilities. Researchers can describe their problem precisely and fully, and JMP can determine smaller numbers of runs that will be sufficient. JMP's unique graphical factor profilers enable researchers to explore the result space over any combination of responses and factors and ultimately maximize desirabilities for their entire system in seconds. JMP's DOE has allowed blue-chip customers to discover million-dollar annual savings or profit opportunities in mere weeks. As Bradley Jones, Senior Manager of Statistical Development and chief architect of JMP's DOE capability, says, "Of all the statistical methods invented in the last 100 years, design of experiments is the most cost-beneficial."

64-Bit Linux Powers JMP's Latest Innovations

But JMP's unique power in areas, such as design of experiments and its newly introduced restricted maximum likelihood estimation of general linear models, are computationally intensive in the extreme. With JMP's new multithreaded architecture running on a 64-bit dual processor, experimental designs that once took several days to compute can now be calculated in minutes. And, problems that were impossible only last year can now be handled in minutes with 64-bit Linux JMP.

Porting JMP to 64-Bit Addressing

"When porting a 32-bit application to 64 bits, there are certain pitfalls you are likely to encounter", says Potter. He continues:

The complexity is compounded if you support not only Linux

but Windows and Macintosh as well, as we do with JMP. The key thing to remember is that in the 64-bit Linux architecture, pointers and longs are both 64-bits wide, while an int remains 32 bits. This breaks any code that assumes a pointer can be stored in an int.

If you have source code compiled on both Linux and Windows, you must be extra careful. Although it's perfectly legal to store a pointer in a long on 64-bit Linux, converting a pointer to a long won't work on Windows, where a long is still only 32-bits wide. On Windows any 64-bit pointers will have to be converted to long longs instead.

Until now floating-point operations on PowerPC for Macintosh used a 64-bit long double, but that may change in the future. PowerPC-based Macs use the "LP64" model in 64-bit executables, meaning that longs and pointers are 64-bit, just as they are on Linux. Apple has not yet announced a 64-bit strategy for Intel-based Macs.

Finally, although graphical user interface APIs are available to 64-bit applications on Linux and Windows, the Macintosh GUIs are not. Of course, you can separate your application into a 32-bit GUI that communicates with a 64-bit kernel, as Wolfram Research did with 64-bit Mathematica.

Nelson observed that porting JMP to 64 bits went smoothly. "The only issue was locating the few places where our code made assumptions about 32-bit word and pointer sizes", says Nelson. "There were no surprises from the tools used for the port, either. Much of the ease in porting is due to the long heritage on 64-bit UNIX platforms of the open-source tools we used."

"The GNU Compiler Collection, gdb, Emacs and the rest of the toolchain performed as expected on the x86_64 architecture", says Nelson.

JMP or Open-Source Alternatives?

Besides JMP, the only statistical software available on the Linux desktop today is an open-source product called R. It has considerable analytical depth and its open-source nature allows statisticians who also have computer programming talent to extend R. However, for the vast majority of people working in statistics today, JMP is generally acknowledged to be a better choice. JMP has an intuitive graphical user interface, a broad range of deep analytical capabilities and comprehensive professionally written documentation. JMP customers have the confidence that their investment is backed by SAS's award-winning, PhD-staffed quality assurance, technical support and professional training and consulting services. SAS boasts a 30-year record of continuous growth, so JMP customers know they can count on SAS and JMP to be around for the long haul.

What's Next for Linux?

"People keep wondering if Linux will ever be a serious contender in the desktop market",

says Potter. "It's been disappointing to Linux enthusiasts that this hasn't yet happened. Now, with the availability of affordable 64-bit desktop machines, we might start to see that change", Potter says. He continues:

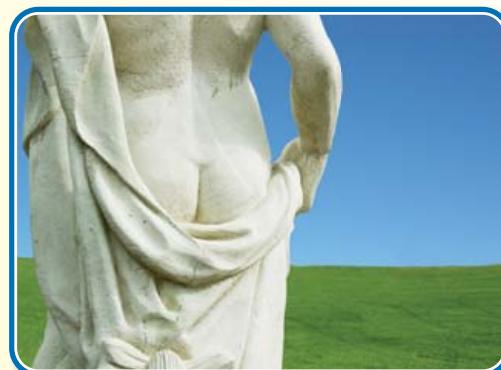
From the server perspective, the Linux operating system is generally recognized to be more reliable and secure at a lower cost of ownership than the alternatives.

Many research scientists and engineers would have liked to adopt Linux on their desktops too. They have refrained from doing so, however, because the applications they depended upon and the computing power they needed simply weren't there. Now those obstacles are gone.

As more of the applications that researchers depend upon, like JMP, become available for 64-bit Linux, its share of the desktop market can only grow. ■

Erin Vang, International Program Manager for JMP R&D at SAS, built JMP's localization and internationalization program. Previously, she was documentation and localization manager for Abacus Concepts (StatView) and technical writer and quality assurance manager for SYSTAT. She holds a B.Mus. in music performance, music history and math from St. Olaf College and an M.Mus. in horn performance from Northwestern University.

DEDICATED SERVERS ROCK BOTTOM PRICING



\$59
/mo

64bit, 2.97GHz
Celeron D
Brand-Name Hardware

w.cari.net carinet
betterServers. betterService.

Web Reporting with MySQL, CSS and Perl

Extending the Maypole Soccer Club Database System with a simple Web-based reporting mechanism. PAUL BARRY

In the March 2005 issue of *Linux Journal*, I used Maypole to create a Web-based database application in only 18 lines of Perl code. The functionality provided by Maypole is impressive except in one important area: reporting. Consequently, I started to research technologies for producing reports from my Soccer Club system. My goal was to provide a set of standard reports that could be executed from a Web interface.

Web Reporting? What to Do?

Web reports can be produced in a lot of ways using any of the many server-side programming technologies, such as PHP, JSPs, Perl scripts and the like. Standalone desktop reporting tools also are available, and it's even possible to use OpenOffice.org to report on a MySQL database. As my reporting requirements are basic, however, I wanted to keep my intellectual effort to the minimum. What I didn't mind doing was spending time crafting the SQL queries that I'd need to produce my reports. Once written, I wanted my SQL query to produce an HTML table of results.

I can do this with Perl, of course, using the DBI and DBD::mysql modules, hand-crafting program code to send the query to the database. I then could post-process the results with more code, before—ultimately—writing yet more code to create the table. For my simple requirements, this felt like too much work. What I really wanted was a quick-and-dirty solution. In the remainder of this article, I detail the Web reporting solution I designed.

MySQL to the Rescue!

While browsing Paul DuBois' excellent *MySQL Cookbook*, I discovered a command-line option for turning the results of a command-line query into an HTML table (recipe 1.23, page 33). By way of example, consider the following command line:

```
mysql -e "select name from player"           \
-u manager -ppwhere CLUB
```

which produces the following textual output when invoked:

```
+-----+
|name      |
+-----+
|Robert Plant |
|Tim Finn    |
|James Taylor|
|Bryan Adams |
|Ian Gillen  |
|Mick Jagger  |
|Neil Young   |
|Bob Dylan   |
+-----+
```

These results not only show the names of all of the players in the Soccer Club database, but they also appear to indicate that the club's players are named after some famous folk and rock singers. When re-run with the HTML creation option, like so:

```
mysql -H -e "select name from player"           \
-u manager -ppwhere CLUB
```

the above command line produces the following, which, trust me, is an HTML table:

```
<TABLE BORDER=1><TR><TH>name</TH></TR><TR><TD>
Robert Plant</TD></TR><TR><TD>Tim Finn</TD></TR>
<TR><TD>James Taylor</TD></TR><TR><TD>Bryan Adams
</TD></TR><TR><TD>Ian Gillen</TD></TR><TR><TD>
Mick Jagger</TD></TR><TR><TD>Neil Young</TD></TR>
<TR><TD>Bob Dylan</TD></TR></TABLE>
```

It is possible to put the SQL query into a file and then refer to the file on the command line. For example—and assuming the above query is in a file called name.sql—this command line produces the same HTML table:

```
mysql -H -u manager -ppwhere CLUB < name.sql
```

Knowing this much, I figured that if I could come up with a means of issuing the HTML-producing command line from a Web interface, I'd be most of the way toward providing my Web reporting solution. So, I wrote a small CGI script in Perl to execute the command line for me.

The CGI Script

The strategy employed by my simple CGI script is straightforward: after determining the name of the query to execute, a command line is constructed and then issued by the CGI script. Any results produced from executing the command line are put inside the body part of the HTML page that the CGI script produces.

After the usual Perl startup lines, the runquery.cgi script starts by defining a series of constant values:

```
#!/usr/bin/perl -w

use strict;

use constant MYSQL  => '/usr/bin/mysql';
use constant USERID => 'manager';
use constant PASSWD => 'pwhere';
use constant DBNAME => 'CLUB';
```

The location for the MySQL client on your computer may be different

from where I have mine, so change the MYSQL constant value if need be. Also, note that I'm hard-coding the values for the database user (USERID), the password (PASSWD) and the database that is to be queried against (DBNAME). Although this may not be the best practice, I am going to explain it away by saying that this is the dirty part of my quick-and-dirty solution. With the constants defined, I indicate that I'm going to use the standard interface to Perl's CGI programming technology:

```
use CGI qw( :standard );
```

Two Perl scalars then are defined, taking their value from any parameters passed from a Web interface to the CGI script. The first parameter, called query, identifies the SQL file to use, while the second, called title, provides a report title to use when displaying results:

```
my $query = param( 'query' );
my $title = param( 'title' );
```

The script then creates the command line that runs the query through the MySQL client program. Note that Perl's dot operator is used to concatenate strings:

```
my $cmdline = MYSQL .
    ' -H -u ' .
    USERID .
    ' -p' .
    PASSWD .
    ' ' .
    DBNAME .
    "< $query ";
```

The script then starts to build an HTML page. The header function generates the correct Content-Type header, and the start_html function starts to create the HTML page using the value provided for the page's title:

```
print header;
print start_html( -title => $title );
```

The next line of code uses Perl's qx operator to execute the command line and return any resulting output from its execution to a variable, called \$results:

```
my $results = qx/ $cmdline /;
```

The rest of the script adds an HTML level 3 heading to the Web page, together with the query results and an HTML link to the reports page. The end_html function finishes the HTML page generation and concludes the script:

```
print "<h3>$title</h3>";

print $results;

print p, "Return to the ".
    a( { -href => "/Club/Reports.html" },
    "List of Reports" );

print end_html;
```

Invoking the Script

To run the script, you need to do two things: put the script in a place where your Web server can find it and put an SQL query into a file. On

my Fedora Core 3 system running Apache 2, the /var/www/cgi-bin/ directory is used to hold the Web server's CGI scripts. So, I simply copy the CGI script into that location and make it executable:

```
cp runquery.cgi /var/www/cgi-bin/
chmod +x /var/www/cgi-bin/runquery.cgi
```

The above directory may not be the location used by your distribution for Web pages, so be sure to check first. As for a query, here's the contents of the file conditions.sql:

```
select player.name as 'Player',
       condition.name as 'Medical Condition'
  from player, condition
 where player.medical_condition = condition.id and
       player.medical_condition != 1;
```

The above SQL query joins the player and condition tables in order to list the names of each player together with his medical condition, assuming he has one. This query file also needs to be copied to the CGI directory on the Web server:

```
cp conditions.sql /var/www/cgi-bin/
```

To execute the query from the CGI script, type the following into your browser's address bar, substituting localhost with the name of

Logic Supply offers a full line of Mini-ITX mainboards and systems for embedded applications.



Mini- and Nano-ITX mainboards

with VIA, Pentium, or Pentium M processors

- x86 compatibility
- full-featured SBC
- off-the-shelf availability



fanless Mini-ITX systems

Utilizing heat pipe technology, these completely fanless Mini-ITX systems offer the perfect solution for applied computing and embedded environments.

LEADERS IN MINI-ITX SOLUTIONS

LOGIC
SUPPLY

LOGICSUPPLY.COM

info@logicsupply.com

CALL US AT 802 244 8302

The screenshot shows a Mozilla Firefox window with the title bar "Results - Mozilla Firefox". The address bar contains the URL "http://localhost/cgi-bin/runquery.cgi?title=Results". The main content area is titled "Results" and contains a table:

Player	Medical Condition
Robert Plant	Asthma
Ian Gillen	Asthma
Neil Young	Asthma
Tim Finn	Epilepsy
Bob Dylan	Epilepsy

Below the table are two links: "Return to the List of Reports" and "Done".

Figure 1. A Functional but Plain HTML Report

your Web server:

```
http://localhost/cgi-bin/runquery.cgi?
    title=Results&query=conditions.sql
```

This URL produces the output shown in Figure 1, which, despite being a little plain, looks okay—but it could be nicer.

Making Things Look Nicer with CSS

To produce a report with an improved look and feel, I created a small cascading style sheet (CSS), called reports.css, to improve the general appearance of the produced report:

```
body {
    font-family: sans-serif;
}

table {
    font-family: sans-serif;
    background-color: LIGHTYELLOW;
}

table th {
    background-color: LIGHTCYAN;
    font-size: 75%;
}

h3 {
    font-family: sans-serif;
    color: BLUE;
}
```

As stylesheets go, mine is pretty simple. I declare a font for the text in my main body and then I fiddle with the font and background color of any tables that I put on my HTML page. The table headings are shown at 75% of the user's normal text size with a different background color from the data in the table. I then declare that my level 3 headings are colored blue.

The CSS file needs to be copied into the Web server's root directory so that my Web pages can find it:

```
cp reports.css /var/www/html
```

To use the CSS file, I changed the print start_html line from runquery.cgi to refer to the stylesheet, as follows:

```
print start_html( -title => $title,
                  -style => { -src => "/reports.css" } );
```

The screenshot shows a Mozilla Firefox window with the title bar "Results - Mozilla Firefox". The address bar contains the URL "http://localhost/cgi-bin/runquery.cgi?title=Results". The main content area is titled "Results" and contains a table:

Player	Medical Condition
Robert Plant	Asthma
Ian Gillen	Asthma
Neil Young	Asthma
Tim Finn	Epilepsy
Bob Dylan	Epilepsy

Below the table are two links: "Return to the List of Reports" and "Done".

Figure 2. A Much Improved HTML Report

Reloading the query produces the output shown in Figure 2. It may not win me a Web design award, but it does look a whole lot better than the plain results shown in Figure 1.

Creating the Web Interface

This part of my solution was easy. All I needed was a simple Web page describing a list of reports. As with the generated reports, I use my simple stylesheet to improve the look of the reports page. Here's the HTML I used:

```
<HTML>
<HEAD>
  <TITLE>Soccer Club Reporting System</TITLE>
  <LINK rel="stylesheet" type="text/css"
        href="/reports.css" />
</HEAD>
<BODY>
<H3>Soccer Club Reporting System</H3>
Choose from one of these reports:
<OL>
  <LI>List players that have a
      <a href="/cgi-bin/runquery.cgi?
          title=Players with a Medical Condition&
          query=conditions.sql">Medical Conditions</a>
  <LI>List all players,
      <a href="/cgi-bin/runquery.cgi?
          title=Listing of all Players (Youngest First)&
          query=desc_dob.sql">youngest first</a>
</OL>
Return to the <A HREF="/Club">Soccer Club</A>
database system.
</BODY>
</HTML>
```

As shown above, each report is executed with two parameters: title, which provides a report description, and query, which identifies the SQL query file to run through MySQL. With my Web page created, I copied it into the root directory of the Soccer Club Web site:

```
cp Reports.html /var/www/html/Club/
```

When loaded into a Web browser, the reporting Web interface

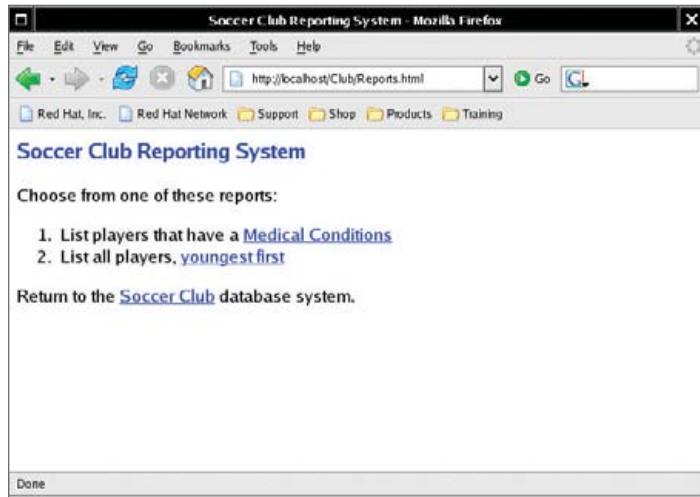


Figure 3. The Reporting Web Interface

looks like that shown in Figure 3.

At this point, I think I'm done. I have a simple Web interface to a standard report producing mechanism. If I write more queries, I can put them into their own SQL query file, copy the file to my cgi-bin directory and update my HTML reports Web page to invoke the query as required. My solution is quick-and-dirty and more than good enough.

Or is it?

The security of my solution is very, very poor. I need to worry about two things, protecting my CGI script and SQL query files from user tampering and protecting my system from the CGI script.

Security: Protection from User Tampering

When it comes to tampering with the CGI script and SQL query files, the problem is—by default—all of the files can be read by any user logged in to the system that runs the Web server; a simple cat or less command would do the trick. Any user can look inside runquery.cgi and display the user ID and password used in accessing the database, which is not good.

The User and Group directives in the Apache httpd.conf configuration file indicate which user and group the Apache Web server runs under. On my computer, this user and group is set to apache. Knowing this, I issued the following commands to ensure that the contents of my CGI script and SQL query files are owned by the apache user and that they can be read from and written to only by the same apache user. This stops any other user—except root, of

An advertisement for NightStar LX. The top half has a red background with the word "WARNING:" in large white letters. Below it, in smaller white text, is "NIGHTSTAR LX™ IS HABIT-FORMING, AND MAY CAUSE EXTREME FEELINGS OF EUPHORIA." The bottom half features a photograph of a shirtless man with a tattoo on his chest, shouting with his mouth open. The text below the photo reads: "Natural ability and ordinary debuggers can take you just so far. That's why you need NightStar LX™. An integrated suite of tools that gives you full visibility into your Linux® application. You can debug, monitor, analyze and tune at application speed, so you see real execution behavior. Plus, you'll reduce test time and lower costs. NightStar LX. Experience *real power* for a change." At the bottom, there is a logo for "NIGHTSTAR LX™" with the tagline "Serious tools for serious apps", a "FREE TRIAL" offer with the website "www.ccur.com/nightstar/LX2", and contact information: "800-666-4544 • 954-974-1700" and "Email us at nightstar@ccur.com". The Concurrent logo is also present.

course—from examining their contents:

```
cd /var/www/cgi-bin
chown apache:apache *
chmod 600 *
chmod 700 *.cgi
```

The first chmod command ensures the files in the cgi-bin directory can be read from and written to solely by the apache user. The second chmod command switches on the executable bit for any CGI scripts, but only for the owner of the file. With these simple precautions, my solution is now safe from user tampering.

Security: Protection from the CGI Script

The above chmod command lines protect the files from other users logged in to the system, but my solution still is vulnerable. Unfortunately, it is open to exploitation by any user with access to the Web server by way of any Web browser. For example, consider what happens if the following URL is sent to the CGI script:

```
http://localhost/cgi-bin/runquery.cgi?title=Ha!&query=conditions.sql | cat runquery.cgi
```

The contents of the CGI script appear in the browser, and the

issuer easily can read the database name, the user ID and password contained within the file. This is bad enough, but imagine if the cat runquery.cgi pipe in the above URL is replaced with this:

```
cat /etc/passwd
```

or the potentially disastrous:

```
rm -rf /
```

The problem with the CGI script as written is it blindly trusts the issuer not to fiddle with the URL. By simply adding the pipe symbol and any other shell command line to the URL, the issuer exploits this poorly designed CGI script, effectively executing other commands of the issuer's choosing on the Web server. By passing the query string unaltered to the operating system to execute, the CGI script makes it far too easy for such a vulnerability to be exploited.

Thankfully, Perl has a special mode of operation that can help, and it is called taint mode. Most any book on Perl describes taint mode, and the second edition of Christiansen and Torkington's *Perl Cookbook* provides a handy primer (recipe 19.4, page 767). By turning on taint mode, the Perl interpreter is instructed not to trust data that originates outside the script. As the data is not trusted—it's "tainted"—Perl won't let you use the data in an unsafe way without raising a run-time exception.

I can turn on Perl's tainting technology by changing the first line of my CGI script to include the taint mode switch:

```
#!/usr/bin/perl -wT
```

When I reload the fiddled-with URL, the resulting HTML page is empty and Apache's error_log has been appended with an insecure dependency error. This is Perl's way of telling me that the script failed due to tainting errors. Obviously, with the script failing in this way, it is no longer a security threat to the system. However, it also is no longer doing what it was designed to do, which makes it all but unusable. To make the script usable again, we need to untaint the input data using Perl's regular expression technology. The idea is straightforward: by defining a pattern representing safe data, the pattern can be applied to the tainted data and—assuming the pattern matches—any results are untainted and considered safe. With the CGI script, there are two data inputs, query and title. I added the following regular expressions to the CGI script to untaint the input data:

```
$query =~ /^([-w]+\.sql)$/;
$query = $1;

$title =~ /^([\w.:?! ]+)$/;
$title = $1;
```

The first regular expression matches on a string that has any combination of hyphens or word characters, followed by a period and the letters s, q and l. Anything else doesn't match and is considered suspect. If a match does occur, Perl remembers the match in the

Ultra Dense, Powerful, Reliable... Datacenter Management Simplified!

15" Deep, 2-Xeon/Opteron or P4 (w/RAID) options



Customized Solutions for... Linux, BSD, W2K

High Performance Networking Solutions

- Data Center Management
- Application Clustering
- Network and Storage Engines

Rackmount Server Products

- **1U Starting at \$499:** C3-1GHz, LAN, 256MB, 20GB IDE
- 2U with 16 Blades, Fast Deployment & more...



**iron
SYSTEMS™**

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

CALL: 1-800-921-IRON

NEW!

TS-7300 High Security
Linux FPGA Computer



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

 200 MHz CPU

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

 Design your solution with one of our engineers

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

 See our website for options, peripherals and x86 SBCs

 **Technologic**
SYSTEMS

We use our stuff.

visit our TS-7200 powered website at

www.embeddedARM.com

(480) 837-5200

\$1 match-variable, which then is assigned back to the now untainted \$query. With titles, the pattern allows any combination of word characters and those characters included within the square brackets of the regular expression. Again, the \$title variable is untainted when a successful match occurs.

As the CGI script executes an external executable—namely, the MySQL client—the environment's path also needs to be untainted. This is accomplished by setting the PATH variable within the environment to a safe list of directories, as follows:

```
$ENV{'PATH'} = "/usr/bin";
```

With these changes made to the runquery.cgi script, it is usable once more. As well as being quick-and-dirty and safe from user tampering, my solution is no longer a potential security threat to my system.

Integration with a Maypole Application

To link my simple reporting interface into my Soccer Club application, I changed the custom/frontpage template to include an additional list item that refers to the reporting Web page:

```
<ul>
[% FOR table = config.display_tables %]
  <li>
    <a href="[%table%]/list">Work with the
      [%table %] data</a>
  </li>
[% END %]
  <li>Work with the <a href="Reports.html">
    Reports System</a>
</ul>
```

When the application is loaded into a browser, the link appears as part of the initial Maypole menu, as shown in Figure 4. My Web-based reporting system is simple, safe and easily extended. All I need to do now is write some more SQL queries. ■

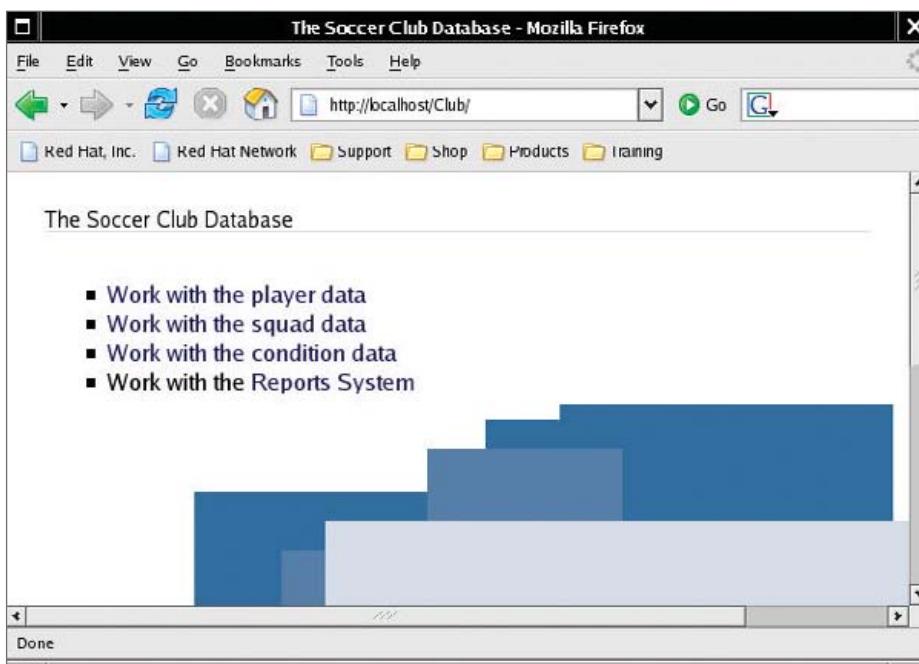


Figure 4. Integrating the Reporting Web Interface with Maypole

Paul Barry (paul.barry@itcarlow.ie) lectures at the Institute of Technology, Carlow, in Ireland. Information on the courses he teaches, in addition to the books and articles he has written, can be found on his Web site, glasnost.itcarlow.ie/~barryp.

eCrash: Debugging without Core Dumps

How to use backtrace and a custom library to debug your embedded applications. DAVID FRASONE

Embedded Linux does a good job of bridging the gap between embedded programming and high-level UNIX programming. It has a full TCP stack, good debugging tools and great library support. This makes for a feature-rich development environment. There is one downside, however. How can you debug problems that occur out in the field?

On full-featured operating systems, it's easy to use a core dump to debug a problem that occurs in the field.

On non-embedded UNIX systems, when a program encounters an exception, it outputs all of its current state to a file on the filesystem. This file is usually called core. This core file contains all the memory the program was using at the time the failure occurred. This allows for post-mortem investigation to diagnose the exception.

Typically, on embedded Linux systems, there is no (or very little) persistent disk storage. On all of the systems on which I have worked, there is more RAM than persistent storage. So, getting a core dump is impossible. This article describes some alternatives to core dumps that will allow you to perform post-mortem debugging.

Programs can fail for reasons other than exceptions. Programs can deadlock, or they can have run-away threads that use up all system resources (memory, CPU or other fixed resources). It also would be beneficial to generate some kind of persistent crash file under these situations.

Requirements

So, first we need to come up with the information we want to save. Because of memory constraints, saving all of the process' memory is not an option. If it were, you simply could use core dumps! But, there is other very useful information we can save. At the top of the list is the backtrace of the failed thread.

A backtrace is a list of the functions that were called to get to the current position in the program. Even with the absence of system memory and data, a backtrace can shed light onto what was happening at the time of failure.

Many embedded systems also have logs: lists of errors, warnings and metrics to let you know what happened. Having a post-mortem dump of the last few logs before failure is an invaluable asset in finding the root cause of a failure.

In complex, multithreaded systems, you usually have many mutexes. It could be useful, in the case of a deadlock, to show the state of all the processes' mutexes and semaphores.

Showing memory usage statistics also could help diagnose the problem.

Once we have determined the information we want to save, we still need to come up with where to save it. This will vary greatly from system to system. If your system has no persistent storage at all, perhaps you can output the crash information to a serial terminal or display it on an LCD readout. (We have serious space constraints there!) If your system has CompactFlash, you can save it to a filesystem. Or, if it has raw Flash (an MTD device), you can either save it to a jffs2 filesystem, or maybe to a raw sector or two.

If the crash was not too severe, perhaps the crash could be uploaded to a tftp server or sent to a remote syslog facility.

Now that we have a firm grasp on what we want to save, and locations

to which we can save it, let's talk about how we are going to do it!

The Backtrace

In general, getting a backtrace is not as simple as it sounds. Accessing system registers (like the stack pointer) varies from architecture to architecture. Thankfully, the FSF comes to our rescue in GNU's C Standard Library (see the on-line Resources). Libc has three functions that will aid us in retrieving backtraces: backtrace(), backtrace_symbols() and backtrace_symbols_fd().

The backtrace() function populates an array of pointers with a backtrace of the current thread. This, in general, is enough information for debugging, but it is not very pretty.

The backtrace_symbols() function takes the information populated by backtrace() and returns symbolic names (function names). The only problem with backtrace_symbols is that it is not async-signal safe. backtrace_symbols() uses malloc(). Because malloc() uses spinlocks, it is not safe to be called from a signal handler (it could cause a deadlock).

The backtrace_symbols_fd() function attempts to solve the signal issues associated with malloc and output the symbolic information directly to a file descriptor.

Working inside of a Signal Handler

Some functions inside of libc rely on signals themselves: some IO operations, memory allocation and so on. So, we are very limited in what we should do inside of a handler. In our case, we can cheat a little. Because our program already is crashing, a deadlock is not *that* big of a concern. The code in my examples makes use of several not-allowed functions, such as fwrite(), printf() and sprintf(). But, we can work to avoid some of the functions that are prone to deadlock, such as malloc() and backtrace_symbols().

In my opinion, the biggest loss we have is the loss of backtrace_symbols. But, here is where things get easier. You always can implement your own symbol table and look up the functions from the pointers themselves.

In my examples, I sometimes use backtrace_symbols(). I have not seen a deadlock yet, but it *is* possible.

A Simple Backtrace Handler

So, what does the crash handler look like? To get a backtrace, the first thing we need to do is grab our signals. Some of the common ones are SIGSEGV, SIGILL and SIGBUS. Additionally, abort() is usually called in the case of an assertion and generates a SIGABRT.

Then, when a signal occurs, we need to save our backtrace. The following snippet details a simple backtrace function that displays the backtrace to standard output when a crash happens:

```
void signal_handler(int signo)
{
    void *stack[20];
    int count, i;

    // Shouldn't use printf . . . oh well
```

```

printf("Caught signal %d\n");

count = backtrace(stack, 20);
for (i=0; i < count; i++) {
printf("Frame %2d: %p\n", i+1, stack[i]);
}
}

int main(...)

{
...
signal(SIGBUS, signal_handler);
signal(SIGILL, signal_handler);
signal(SIGSEGV, signal_handler);
signal(SIGABRT, signal_handler);
}

Caught signal 11
Frame 1: 0x401a84
Frame 2: 0x401d88
...

```

And, here is a similar signal handler, but one that uses backtrace_symbols to print out a prettier backtrace:

```

void signal_handler(int signo)
{
    void *stack[20];
    char **functions;

    int count, i;

    // Shouldn't use printf . . . oh well
    printf("Caught signal %d\n");

    count = backtrace(stack, 20);
    functions = backtrace_symbols(stack, count);
    for (i=0; i < count; i++) {
printf("Frame %2d: %s\n", i, functions[i]);
    }
    free(functions);
}

Caught signal 11
Frame 1: ./a.out [0x401a84]
Frame 2: ./a.out [0x401bfa]
...

```

eCrash—A Generic Crash Handler

As I was writing this article, I realized that this was about the fifth time I had written a crash handler. (Why can't all software be open source?) So, I decided to write a quick library to handle crash dumps and provide it for this article. I liked the little library I started with, but I found myself needing more and more features. As I kept extending it, I realized that it was a very useful library, that I wanted to be able to leverage on any future project!

I named the new library eCrash and created a SourceForge site for it (see Resources). Since then, I have been extending it, and it now supports dumping multiple threads—using only backtrace, using backtrace and backtrace_symbols, and using backtrace with a user-supplied symbol table to avoid the malloc() inside of backtrace_symbols. The rest of the examples in this article are going to be leveraging eCrash.

eCrash is relatively simple to use. You first call eCrash_Init() from your parent thread. If you have a single-threaded program, you are

already finished. A backtrace will be delivered based on your settings in the parameters structure.

If you have a multithreaded program, any thread that wants to be backtraced in a crash (other than the crashing thread) must also call eCrash_RegisterThread(). It is sometimes useful to dump the stacks of all threads when a crash occurs, not only the crashing thread's stack.

With eCrash, you specify where the output should go by setting file descriptors (async safe writes), FILE * streams (not async safe), and/or a filename of the file to output when a crash occurs. eCrash will write to *all* destinations supplied.

eCrash—Gathering Stacks from Other Threads

Obtaining the stack from a thread that did *not* crash is a bit trickier. When a thread registers, it specifies a signal that the thread does not catch or block. eCrash registers a handler for that signal (called the Backtrace Signal).

When eCrash needs to dump a thread (when some other thread has caused an exception), it sends the thread the Backtrace Signal via pthread_kill(). When that signal is caught, the thread saves its backtrace to a global area and continues on. The main exception handler can then read the stack and display it.

What we end up with is a very nice-looking crash dump, showing exactly what was happening in the system when the failure occurred.

eCrash—A Real-World Example

Enough talking—time for some meat. Now, let's take what we have

**EMBEDDED LINUX
Made Easy!**

Panel PCs PC Compatible SBCs SoM & PC/104 Modules

Custom Engineering Embedded Servers

Turn-Key Solutions!

- ECLIPSE IDE with full Library and Module Support
- Integrated GDB Source Level Debugger
- Resident Menu Driven Setup Utility
- No Charge for Tools or Standard Linux Install
- Free Integration, Setup and Test

Since 1985
OVER
21
YEARS OF
SINGLE BOARD
SOLUTIONS

EMAC, inc.

Phone 618 - 529 - 4525 Fax 618 - 457 - 0110
2390 EMAC Way, Carbondale, Illinois 62902
World Wide Web: <http://www.emacinc.com>

2.4 Kernel

discussed and put it to work. We are going to use the ecrash_test program included in eCrash. That program was designed to break in any one of its threads (generate a segmentation violation by attempting to write to a NULL pointer).

We execute the test program with the following flags:

```
ecrash_test --num_threads=5 --thread_to_crash=3
```

This causes the test program to generate five threads. All but thread number 3 will call a few functions, then go to sleep. Thread 3 will call a few functions (to make the backtrace interesting) and crash.

The crash file generated is shown in Listing 1 and backtrace_symbols() in Listing 2. Due to space constraints, all listings for this article are available on the *Linux Journal* FTP site (<ftp.ssc.com/pub/lj/issue149/8724.tgz>).

The crash file has the backtrace of our offending thread (the one that caused the segmentation violation) and the backtraces of all threads on the system.

Now it's time to debug the crash. We will debug this as if the crash happened at a remote site and the system administrator e-mailed you this crash file.

One last thing: in the real world, the executables always are stripped of debugging information. But, that is okay. As long as you keep a copy of the program *with* its debugging information, you can ship a stripped copy of the code, and everything will still work!

So, in the lab, you have your crash file and your program with debugging information. Run `gdb` on the debug version of your program. We know that we have a segmentation violation. So, starting from frame zero of the offending thread, start listing the code as shown Listing 3 (see the *LJ* FTP site).

- Frame 0 is inside of our crash handler—nothing to see here.
- Frame 1 is also inside of the crash handler.
- Frame 2 is still inside of our crash handler.
- Frame 3 shows no source file (it is inside of libc).
- Frame 4 shows the actual crash (inside of crashC).
- Frame 5 shows crashB.
- Frame 6 shows crashA.
- Frame 7 shows ecrash_test_thread.
- And, frames 8 and 9 are where the thread gets created in libc.

As you can see, there is a trick to displaying function pointers with `gdb`. Simply give it an address and dereference it in a list:

```
(gdb) list *0xWHATEVER
```

This also works with `symbolic_names` and offsets:

```
(gdb) list *main+100
```

Okay, that was our crashed thread, but what about one of the sleepers? Examine the backtrace from Thread 5, Listing 4 (see the *LJ* FTP site):

- Frame 0 is inside of our backtrace handler.
- Frame 1 still inside of the handler.

- Frame 2 is in libc.
- Frame 3 is in libc.
- Frame 4 is in libc.
- Frame 5 is inside of sleepFuncC—it is showing the for statement as the program counter, because we are outside of the `sleep()` function. This is notable because the async signal sent to tell the thread to dump its stack caused `sleep()` to exit prematurely.
- Frame 6 shows sleepFuncB.
- Frame 7 shows sleepFuncA.
- Frame 8 shows crash_test_thread.
- Frame 9 is where the thread gets created in libc (or libpthread).

So, this thread is one of the sleeping threads. Not much to see, but in some cases, this thread's information could be vital to discovering the cause of a crash.

Other Useful Crash Information

Now that we have clobbered to death what a backtrace is, how to produce one, the different methods of displaying one and how to debug a crash with one, it's time to change gears. A crash file can include a lot more information:

- States of mutexes (who is holding the locks—useful for deadlock diagnosis).
- Current error logs.
- Program statistics.
- Memory usage.
- Most recent network packets.

Some of the above items could be useful information for post-mortem debugging. There is one caveat, however. Because we have encountered an exception, something has gone terribly wrong. Our data structures could be corrupt. We could be low on (or out of) memory.

Also, some threads could be deadlocked waiting on mutexes that our crashed thread was holding.

Because some of the data we want to display might generate another exception (if it is corrupted), we want to display the most important information first, then display more and more unsafe information. Also, to prevent information loss, buffers always should be flushed on FILE* streams.

Conclusion

Diagnosing a problem on a deployed embedded system can be a difficult task. But, choosing the right data to save or display in the case of an exception can make the task much easier.

With a relatively small amount of storage, or a remote server, you can save enough post-mortem information to be able to find a failure in your system.■

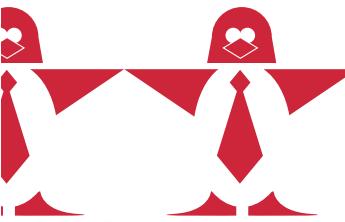
Resources for this article: www.linuxjournal.com/article/9139.

David Frascone (dave@frascone.com) works for Cisco Systems, Inc., in the Wireless Business Unit. He is currently working on Next Generation controller design.



25-26 October 2006 • Olympia 2 • London

Open. For Business.

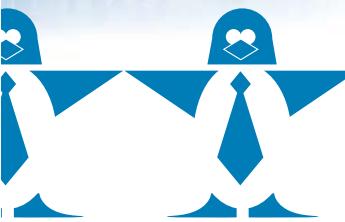


EXPO

- See new products and services from over 60 exhibitors
- Meet developers and technical specialists in the .Org Village
- Take LPI examinations at a special visitor discount

Free-to-attend sessions include:

- Business Briefings
For Corporate and Senior IT managers looking to gain real life examples of advantages, disadvantages, applications and integration of Linux and Open Source in business.
- Great Linux Debate
- Seminars and Presentations



CONFERENCE

Featuring world-leading technical speakers discussing today's hottest topics. Aimed at technical and IT professionals the 2006 conference offers:

- 3 conference/master class streams
- Over 32 hours of content
- In-depth technical sessions
- Business focused topics
- Sessions from just £95

Book your passport package today at
www.linuxworldexpo.co.uk/conference

LinuxWorld Conference & Expo is the premier event exclusively focused on Linux and open source solutions. As the world's most comprehensive marketplace for open source products and services, LinuxWorld provides business decision-makers and technical professionals with information and resources to implement Linux and open source solutions into business infrastructure and enterprise networks.

Register today for FREE queue jumping entry to the expo and save £15
www.linuxworldexpo.co.uk

EXPERT SPEAKERS INCLUDE:



Jon "maddog" Hall
Linux International



Ian Pratt
Kings' College Cambridge



Alan Cox
Red Hat



Bruce Perens
Sourcelabs



Jeremy Allison,
Novell



Chris DiBona
Google



Bill Weinberg
OSDL

Register now for updates at www.linuxworldexpo.co.uk

ORGANISED BY:

turret GROUP
International Business Publications & Exhibitions

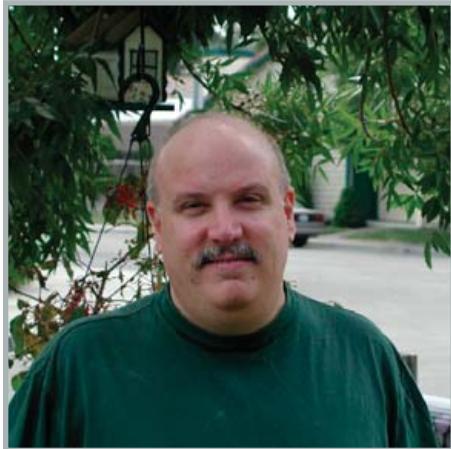
LinuxWorld Conference & Expo is owned and licensed by IDG World Expo, which is a business unit of IDG, the world's leading technology media, research and event company. Copyright © 2006 IDG World Expo Corp. All rights reserved. LinuxWorld and LinuxWorld Conference & Expo are registered trademarks of International Data Group, Inc. E&OE. Programme may be subject to change. Correct at time of press.

IDG
WORLD EXPO

IDG
INTERNATIONAL DATA GROUP

Parallel Is Coming into Its Own

Changing my mind about distributed computing made me aware of the sweet aroma of opportunity.



Nick Petreley, Editor in Chief

I started writing about computing back in the 1980s. I don't want to say which year, or do the math for how long I've been doing this. It makes me feel old.

I've made a plethora of predictions since then. Some of them left me red-faced and embarrassed. Some of them were spot-on. Some of them have not yet been fulfilled, but I still think my predictions are on target.

One of my earliest predictions was rather easy, but it was considered controversial back in the 1980s. I said that it was only a matter of time before we bumped up against the limits of Moore's Law, and the only viable answer would be parallel processing. Lo and behold, dual-core processors are now common, and it won't be long before we see quad-core processors, and the multicore cell processor in the PlayStation 3 is around the corner.

Naturally, the next logical step is clustering or other means of distributed processing. Here's where I begin to get nervous. When "grid computing" became a buzzword, my knee-jerk reaction was, "no, thanks". I don't work in a company office anymore, but if I did, I wouldn't want the company off-loading

processing to my desktop workstation unless I was certain that everything ran in a completely isolated sandbox. Put the grid processes in a chroot environment on Linux, for example. Even then, I'm not sure I'd be happy about the idea. What if I want to do something compute-intensive, and the grid process decides it wants my CPU cycles more than I do? This isn't supposed to happen, but since it's all in the hands of some administrator with his or her own agenda, why should I trust that it won't happen?

It's the lack of control and fear of security breaches that make me nervous. I've got four computers in my home that nobody ever turns off, and two more for special purposes that I turn on as needed. The two hand-me-down computers my kids use sit idle much of the time, unless my daughter is browsing the Web, or my son is playing *World of Warcraft*. I use a server as a centralized provider of resources such as printers, files and e-mail. It's a very old machine, but it never breaks a sweat given its purpose. All this represents a tremendous amount of wasted processing power. I'd love to tap in to that unused power at home. This is a safe environment, because I'm not talking about exposing my processing power to everyone on the Internet. I'm talking about distributing workloads across local machines.

In principle, however, Sun was right all along when it said, "the network is the computer". Other companies, such as IBM, worked along the same lines before Sun did, but I don't know of any company that said it better than Sun. "The network is the computer" is a powerful phrase. As long as there is adequate security built in to every aspect of distributed processing, it makes perfect sense to provide common services as remote procedure calls and distribute every conceivable workload across as many computers as you want to make available to the system. If someone could make me feel comfortable about security and control, I'd buy into distributed processing in a big way.

Here are the challenges as I see them. First, there's the problem of heterogeneous platforms. How do you distribute a workload across machines with different processors and different operating systems? ProActive is one of several good platform-agnostic distributed computing platforms (see www-sop.inria.fr/oasis/ProActive). It is 100% pure Java, so it runs on any platform that supports Java. It has a great graphical interface that lets you manage the way you distribute the load of a job. You can literally drag a process from one computer and drop it onto another.

The problem is that a tool like ProActive doesn't lend itself to the way I want to distribute computing. I want it to be as transparent as plugging a dual-core processor in to my machine. Unfortunately, you can't get this kind of transparency even if you run Linux on all your boxes. The closest thing to it that I can think of is distcc, which lets you distribute the workload when you compile programs. Even this requires you to have the same version of compiler (and perhaps some other tools) on all your boxes. If you want this to be a no-brainer, you pretty much have to install the same distro of Linux on all your machines.

The bottom line here is that I smell an opportunity for Linux. I would love to see a project that makes distributed computing on Linux brainlessly transparent and distribution-agnostic. I'm talking about the ability to start up any computation-intensive application and have it automatically distribute the work across other machines on the network configured to accept the role as yet another "processor core". You can make this transparent to the application by building it into the core user-space APIs. You manage it like you would any other network service. Is this too pie in the sky? I'd love to hear your opinions. ■

Nicholas Petreley is Editor in Chief of *Linux Journal* and a former programmer, teacher, analyst and consultant who has been working with and writing about Linux for more than ten years.



Rackspace – Managed Hosting Backed by Fanatical Support™

Fast servers, secure data centers and maximum bandwidth are all well and good. In fact, we invest a lot of money in them every year. But we believe hosting enterprise class web sites and web applications takes more than technology. It takes Fanatical Support.

Fanatical Support isn't a clever slogan, but the day to day reality our customers experience working with us. It's how we have reimaged customer service to bring unprecedented responsiveness and value to everything we do for our customers. It starts the first time you talk with us. And it never ends.

Contact us to see how Fanatical Support works for you.

1.888.571.8976 or visit www.rackspace.com



Thanks for honoring us with the
2005 Linux Journal Readers' Choice Award for
"Favorite Web-Hosting Service"



The World's Fastest InfiniBand Switch

From a Company You've Trusted for 24 Years

Microway's **FasTree™** DDR InfiniBand switches run at 5GHz, twice as fast as the competition's SDR models. FasTree's non-blocking, flow-through architecture makes it possible to create 24 to 72 port modular fabrics which have lower latency than monolithic switches. They aggregate data modulo 24 instead of 12, improving nearest neighbor latency in fine grain problems and doubling the size of the largest three hop fat tree that can be built, from 288 to 576 ports. Larger fabrics can be created linking 576 port domains together.



▲ 72 Port FasTree™ Configuration

Working with QLogic's InfiniPath InfiniBand Adapters, the number of hops required to move MPI messages between nodes is reduced, improving latency. The modular design makes them useful for SDR, DDR and future QDR InfiniBand fabrics, greatly extending their useful life. Please send email to fastree@microway.com to request our white paper entitled *Low Latency Modular Switches for InfiniBand*.

Harness the power of 16 Opteron™ cores and 128 GB in 4U

Microway's **QuadPuter®** includes four or eight AMD dual core Opteron™ processors, 1350 Watt redundant power supply, and up to 8 redundant, hot swap hard drives-all in 4U. Dual core enables users to increase computing capacity without increasing power requirements, thereby providing the best performance per watt. Constructed with stainless steel, QuadPuter's RuggedRack™ architecture is designed to keep the processors and memory running cool and efficiently. Hard drives are cooled with external air and are front-mounted along with the power supply for easy access and removal. The RuggedRack™ with an 8-way motherboard, 8 drives, and up to 128 GB of memory is an excellent platform for power- and memory-hungry SMP applications.



Call us first at 508-746-7341 for quotes
on clusters and storage solutions.
Find testimonials and a list of satisfied
customers at microway.com.

◀ QuadPuter® Navion™ with hot swap, redundant power and hard drives and four or eight dual core Opterons, offering the perfect balance between performance and density



Microway
Technology you can count on™
508.746.7341 microway.com