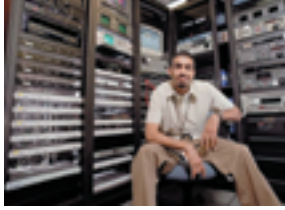
SYS-CON

Security
The Open Source Advantage in Secure Application Development

Enterprise
Selling Software in China: Better Than You Think

Ruby
Bringing Ruby to the Enterprise



NEW LOOK!
Our Name has Changed!
FORMERLY LINUXWORLD MAGAZINE

ENTERPRISE OpenSource MAGAZINE

Visit us at opensource.sys-con.com

THE LEADING MAGAZINE FOR ENTERPRISE AND IT MANAGEMENT

JUNE 2006 VOLUME 4 ISSUE 6

A Closer Look at Damn Small LINUX

The "take-along" operating system!

+PLUS

New Model, New Rules:
10 Rules of Open Source Marketing

JBoss Interview:
Open Source and Middleware

Linux Processes:
Structure, Hangs, and Core Dumps

OPENINSIGHT for LINUX

VERSION 7.2.1

MySQL NETWORK CERTIFIED PARTNER

OpenInsight and MySQL Make Beautiful Music Together

With OpenInsight for Linux (OIL) version 7.2.1 from Revelation Software you can develop professional looking applications on Linux platforms with ease.

OIL provides application developers with the freedom to be platform independent while creating professional looking, scalable, extensible, robust applications in a fraction of the time it takes to create them using other development tools, and deploy single user runtime versions at no extra cost.

New to version 7.2.1 is OIL's DataSet Object, which is the key to unlocking client/server connectivity. The DataSet Object is implemented within the product's visual Client/Server Workspace – an environment in which developers create and manage database components using a collection of visual tools. These tools include a query window, a SQL script-building assistant, a database connection builder, and a DataSet Object definition tool. The database access and manipulation components are separate from other application components and are reusable across multiple applications.

With this new DataSet Object, developers using OIL now have connectivity to MySQL and any ODBC data source.

For more information about OIL, our Solution Partner Program, or OpenInsight training classes at Revelation Headquarters or one of our Authorized Training Centers, go to revelation.com/linuxworld/lwhome or give us a call at 800-262-4747.

Revelation Offices – Global

Revelation Software, Inc.
Corporate Headquarters
99 Kinderkamack Road, 1st Floor
Westwood, NJ 07675
Toll Free: (800) 262-4747
Phone: (201) 594-1422
Fax: (201) 722-9815
Email: info@revelation.com
Web: www.revelation.com

Revelation Software Ltd.
45 St. Mary's Road
3rd Floor
Ealing
London, W5 5RG, UK
Phone: +44(0)208 912 1000
Fax: +44(0)208 912 1001
Email: info@revsoft.co.uk
Web: www.revsoft.co.uk

Revelation Software Australia Pty Ltd.
Suite 105
20 Dale Street
Brookvale, NSW 2100
Australia
Phone: +61-2-9939-6399
Fax: +61-2-9939-6366
Email: info@revelationsoftware.com.au
Web: www.revelationsoftware.com.au

REVELATION SOFTWARE

What's Inside...

JUNE 2006 VOL.4 ISSUE 6

[18]

A Closer Look at Damn Small Linux
by Russ Ethington

[8]

10 Rules of Open Source Marketing
by Dr. Ian Howells

[12]

Bringing Ruby to the Enterprise
by Brad Banister

HP ProLiant BL35p BLADE SERVER

with ProLiant Essentials Management Software

- Up to 2 Dual-Core AMD Opteron™ 200 Series processors
- High density: Up to 96 servers per rack
- Flexible/Open: Integrates with existing infrastructure
- HP Systems Insight Manager™: Web-based networked management through a single console
- Rapid Deployment Pack: For ease of deployment and ongoing provisioning and reprovisioning
- Integrated Cisco or Nortel switch options

Save up to $450 on select AMD based Blade Servers.[1]

HP StorageWorks MSA1500cs

with StorageWorks Essentials Management Software

- Up to 24TB of capacity (96 250GB SATA drives)
- Up to 16TB of capacity (56 300GB SCSI drives)
- Ability to mix SCSI and Serial ATA enclosures for greater flexibility
- 2GB/1GB Fibre connections to host

Get 2TB of storage free ($2,008.80 value)[2]

Chaos, now under your control.

HP BladeSystem servers offer tools to help you keep pace with fluctuating demands. The HP ProLiant BL35p Blade Server is designed to relieve some of the stress. Its AMD Opteron™ processors offer dual-processor power with breakthrough efficiency. With management features like the Rapid Deployment Pack that lets you deploy and redeploy blades without missing a beat, and a single-view, graphical user interface that streamlines monitoring and configuration, HP BladeSystem servers work with you so you don't have to work so hard. And, bundled with the StorageWorks MSA1500cs, you can reduce the cost and complexity of deploying a storage area network, giving you a better return on investment.

Save up to $450 on select AMD based Blade Servers.[1]

SMART ADVICE > SMART TECHNOLOGY > SMART SERVICES

AMD 64 Opteron™

Call 1-888-220-7138
Click hp.com/go/bladesmag48
Visit your local reseller

hp invent

1. Save up to $450 on select AMD based Blade Servers. Offer valid through 7/31/06. 2. Receive up to 2TB of storage free with purchase of HP StorageWorks Modular Smart Array 1500cs devices. Offer valid through 7/31/06. All offers available from HP Direct and participating resellers. Prices shown are HP Direct prices, are subject to change and do not include applicable state and local sales tax or shipping to recipient's destination. Reseller prices may vary. See Web site for full details. For hard drives, 1GB = 1 billion bytes. Actual formatted capacity is less. Photography may not accurately represent exact configurations priced. Associated values represent HP published list price. AMD, the AMD Arrow Logo, AMD Opteron and combinations thereof are trademarks of Advanced Micro Devices, Inc.

ENTERPRISE> OPENSOURCE MAGAZINE

www.OpenSource.SYS-CON.com

OFFICES

SYS-CON MEDIA
135 Chestnut Ridge Rd. • Montvale, NJ 07645
Telephone: 201 802-3000 • Fax: 201 782-9600
LinuxWorld Magazine (ISSN #PENDING)
is published monthly (12 times a year)
by SYS-CON Publications, Inc.

Postmaster send address changes to:
ENTERPRISE OPEN SOURCE MAGAZINE
SYS-CON MEDIA • 135 Chestnut Ridge Rd. • Montvale, NJ 07645

WorldWide Newsstand Distribution
Curtis Circulation Company, New Milford, NJ

FOR LIST RENTAL INFORMATION:
Kevin Collopy: 845 731-2684, kevin.collopy@edithroman.com
Frank Cipolla: 845 731-3832, frank.cipolla@epostdirect.com

SYS-CON MEDIA | www.SYS-CON.com

The Times They Are a Changing

By Mark R. Hinkle

In the 1964 Bob Dylan song The Time's They are A-Changin' it says, *"Come gather 'round people, wherever you roam and admit that the waters around you have grown, And accept it that soon you'll be drenched to the bone. If your time to you is worth savin' then you better start swimmin' Or you'll sink like a stone. For the times they are a-changin."*

When the gifted songwriter first sang those words Linux didn't exist. For that matter neither did Linux kernel creator Linux Torvalds. However, now halfway through 2006 the Open Source ranks are growing by leaps and bounds, and if you don't consider this new breed of Open Source solutions in your IT decision-making process then you'll surely sink like a stone.

The year Dylan wrote his famous tune computers were rare. It was the year IBM launched the 360, which included free software *with source code*. With computers expensive and scarce it didn't matter about the software partly because the computer was the limiting factor. But it wouldn't stay that way for long. See, it was around then that Intel co-founder Gordon Moore suggested that integrated circuits would double in complexity every year though the price would stay the same. Today the One Laptop per Child (www.laptop.org) initiative is taking flight to produce and distribute low-cost laptops running Linux and Open Source applications for less than $100. Who would have thought this was possible 10 years ago when laptops were in the thousands of dollars.

As hardware became more widely available commercial software started to become an issue. In the 1970s there were concerns about whether or not software should be copyrightable since some software vendors started to press the issue. Volume II, Issue 1 of the Homebrew Computer Club Newsletter reprinted "An Open Letter to Hobbyists" letter dated February 3, 1976 from Bill Gates that said, "Hardware must be paid for, but software is something to share. Who cares if the people who worked on it get paid?" That may have been the beginning of the debate that grows every day especially since hardware prices have dropped like a stone yet the cost of the commercial operating system hadn't. The question is will Open Source help commoditize software and if so is that a good thing?

I'm writing this article with a free word processor (OpenOffice.org) that rivals those of the commercial ilk. I can run it on either my free Linux operating system or my commercial Windows operating system. I send copy to my editor via a free mail server (Sendmail) where it's probably laid out with a commercial editing package on a commercial operating system spawned from a FreeBSD licensed operating system on proprietary hardware (Apple). Which brings me to my point. The new world of computing has a lot to do with Open Source of which Linux is just one of many successful examples.

In 2003 when we started publishing LinuxWorld Magazine we highlighted Linux because it was noticeably the most successful Open Source project. However, what we would have cited as the most popular Open Source projects then are a bit different today. For example, the popular Firefox browser is a raging success used by Windows, Solaris, and Linux users alike. So is OpenOffice.org, with its 2.0 version, which has started to demonstrate real maturity as an office productivity suite. Robust J2EE applications are built using Eclipse and rich Web sites are deployed under free content management systems like Joomla!, Drupal, and WordPress.

So as the last chapter on LinuxWorld Magazine has been closed we start a new and equally exciting chapter of the Open Source story that not only influences the Linux user but those running on other platforms. As we move forward with Enterprise Open Source Magazine I hope to be able to highlight technologies and solutions like Linux but also other Open Source solutions that might appeal to enterprise users on other platforms. You see Linux was the first shot across the bow, and I believe that more and more users will continue to mandate open products.

In the coming months I expect to showcase an powerful and evolving cast of high-quality open software and success stories on how they're changing the IT landscape. As I close our inaugural editorial for ***Enterprise Open Source Magazine*** I have to wonder if the ending of Dylan's song will ring as true for Open Source:
The line it is drawn, The curse it is cast. The slow one now, Will later be fast...
The order is rapidly fadin'. And the first one now will later be last. For the times they are a-changin'.

About the Author
Mark R. Hinkle is the vice president of strategy and corporate development for Emu Software as well as the editor-in-chief of Enterprise OpenSource Magazine. *Mark served on the Formation Board of The Desktop Linux Consortium and is also the author of the upcoming book* Linux Business Desktop Migration for Windows Users *from Charles River Press.*
mrhinkle@encoreopus.com

Reclaim lost time

Levanta Intrepid M™

The world's first Linux management appliance

The Intrepid M™ appliance performs the most important Linux management tasks in a fraction of the time you now spend. You'll save both time and money with a single point of control.

The Intrepid M does these tasks and more:

- **Quickly provision servers** — in minutes—even from bare metal
- **Easily deploy software and patches** – eliminating lengthy installations
- **Efficiently migrate software** — from one piece of hardware to another
- **Instantly rollback** — undo changes in minutes, even to the distant past

"Levanta reduced an eight-hour process down to five minutes."

-Arty Ecock, Manager of VM Enterprise Systems, City University of New York (CUNY), Computing and Information Systems

Download the Illuminata Analyst Report

Levanta Marries Provisioning with Data Virtualization

www.levanta.com/lwm0406

Find out why the Intrepid M is the cutting edge in Linux server management.

LEVANTA®
www.levanta.com
1.877.LEVANTA

WINNER
LinuxWorld
Most Innovative Hardware Solution

ENTERPRISE> OPENSOURCE MAGAZINE

www.OpenSource.SYS-CON.com

President & CEO
Fuat Kircaali fuat@sys-con.com

Group Publisher
Jeremy Geelan jeremy@sys-con.com

ADVERTISING

Senior VP, Sales & Marketing
Carmen Gonzalez carmen@sys-con.com

VP, Sales & Marketing
Miles Silverman miles@sys-con.com

Advertising Director
Robyn Forma robyn@sys-con.com

Advertising Sales Manager
Megan Mussa megan@sys-con.com

Associate Sales Managers
Kerry Mealia kerry@sys-con.com
Lauren Orsi lauren@sys-con.com

PRODUCTION

Art Director
Alex Botero alex@sys-con.com

Associate Art Directors
Abraham Addo abraham@sys-con.com
Louis F. Cuffari louis@sys-con.com
Tami Lima tami@sys-con.com

SYS-CON.TV
Ryan Palmeri ryan@sys-con.com

CUSTOMER RELATIONS

Circulation Service Coordinator
Edna Earle Russell edna@sys-con.com

SYS-CON.COM

Information Systems Consultant
Robert Diamond robert@sys-con.com

Web Designers
Stephen Kilmurray stephen@sys-con.com
Wayne Uffleman wayne@sys-con.com

ACCOUNTING

Financial Analyst
Joan LaRose joan@sys-con.com

Accounts Payable
Betty White betty@sys-con.com

Accounts Receivable
Gail Naples gailn@sys-con.com

SUBSCRIPTIONS

TOLL FREE 888-303-5282
201-802-3012
subscribe@sys-con.com
For subscriptions and requests for bulk orders, please send your letters to Subscription Department
Cover Price: $5.99/issue
Domestic: $49.99/yr (12 issues)
Canada/Mexico: $79.99/yr
all other countries $99.99/yr
(U.S. Banks or Money Orders)
Back issues: $12 U.S. $15 all others

SYS-CON MEDIA | www.SYS-CON.com

Selling Software in China

Better Than You Think

by Bud Michael

Technology companies have been taking a beating lately because of their efforts in China. Google and Yahoo are torn between the evil of government censorship and the prospect of doing business with the world's fastest-growing economy. Microsoft, Adobe, and content providers like the American movie studios face serious problems with piracy. A recent report by the China-Britain Business Council stated that only one in four people in China has purchased software legally.

Imagine my surprise then when I took over a San Jose-based enterprise software company with virtually no sales to learn that our main sales effort was in China. It's a big world, I thought, and surely we could find a safer, less controversial market. My skepticism didn't prevent me from investigating our possibilities there, however, and I'm glad it didn't. I have been pleasantly surprised by the receptive business climate, the ability of clients and other contacts to separate government ideology from business issues, and, most importantly, the size and receptivity of the market.

With an initial software licensing deal under our belts, and the prospect of more to come, I thought I could provide a few insights for others who might find themselves in my shoes.

Business, Not Politics

First of all, business people have to be able to separate political/social views from business issues. The Chinese business people I've dealt with certainly do. This doesn't mean leaving your conscience at the border, but it does mean realizing that not everybody looks at the world the same way we do. Silicon Valley is a leading source of technological innovation, and if we're going to export that technology to the world at large we must learn how to operate in different environments.

The government controls almost all aspects of Chinese society; we revel in relative freedom from government interference. Good for us, not so good for them. But my advice is simple: deal with it. I tip my hat to the U.S. tech leaders who have been grilled by Congress and often vilified in the U.S. press for "selling out" in China. To me, they are pioneers in this brave new world of capitalism co-existing with communism, and I appreciate their willingness to take their lumps in the cause of market expansion.

Intellectual Property Safety

My company's product is software, that most fragile of business products. We pour all our ingenuity and hopes for the future into bits and bytes encased in a disk or downloaded from the Internet. The very act of licensing someone else to use that software carries the implicit hope that they will respect the intellectual capital it represents and use it ethically. Is it reasonable

—continued on page 46

About the Author

Bud Michael is CEO of San Jose, CA high-availability software company Availigent (www.availigent.com).

10 Rules of Open Source Marketing

New model, new rules

by Dr. Ian Howells

I was lucky enough to be Documentum's first employee in Europe in 1993. While there, I worked closely with Geoffrey Moore and got "religion" about understanding not just the so-called "chasm" but the whole marketing model and its implications for strategy, marketing, product, and operational behaviour. I started working with John Newton in the late '80s and we recently discussed marketing models and their relevance to Open Source as well as Geoffrey Moore's new thinking in Darwin and the Demon. This conversation was the root of my thoughts on rules for Open Source marketing – new model, new rules (and some old ones).

Many people are familiar with Moore's Technology Adoption LifeCycle (TALC) where everyone dreams of hitting the "Tornado."

- ***Early Market*** – Discontinuous innovation. Attractive to technology enthusiasts and visionaries
- ***Chasm*** – A pause in market interest when there is no natural customer group
- ***Bowling Alley*** – Adoption by a specific customer segment for a specific problem. Attractive to pragmatists due to specific benefits not available in current technology
- ***Tornado*** – Hypergrowth. Remaining pragmatists adopt en masse choosing the market leader as a new infrastructure
- ***Main Street*** – Conservative buyers who choose, not for competitive advantage, but not to be left behind

Almost 10 years later, in Darwin and the Demon Moore described his new ideas as:

- ***Main Street Mature*** – Growth flattens, commoditization increases followed by consolidation, with the market leaders focusing on top-line growth through M&A. Customers take the category for granted
- ***Main Street Declining*** – Market dominators are unresponsive to customer needs

He then says, "The market is ripe for some form of disruption." In this extended model, disruption is a form of "marketing innovation" and "business model innovation." Open Source is that kind of disruption.

About the Author
Dr. Ian Howells is chief marketing officer of Alfresco and has more than 20 years of enterprise software marketing experience in the fields of content management, service-oriented architectures, and relational database systems. Ian earned a PhD in distributed databases from University College Cardiff. He has long been on the forefront of technology and marketing, holding early positions at Ingres, Documentum and SeeBeyond. You can read Howell's thoughts on open source marketing at http://blogs.alfresco.com/ianh

STRATEGY

Rule 1: Know Your Target Customer

Open Source is about mass use by the underserved customer with similarities in other industries. In the airline industry, for every business person that "lived" on a plane, there were thousands of people who flew once every few years. Low-cost airlines were a marketing and business model innovation that targeted not the businessperson, but the massive underserved market. They have now become some of the most profitable airlines in the world. In content management, for every user of an expensive ECM system there are thousands of users of shared drives.

Rule 2: Know Who You're Competing Against

Geoffrey Moore in Crossing the Chasm pointed out that when you start in a new market you compete and compare yourself to the old technical approach. In this new model the competition is the old closed vendors with old marketing and business models. The competition is not other Open Source vendors.

Rule 3: Attack the Weakness in their Strength

Ries and Trout promoted marketing warfare, not just attacking a weakness in the competition (since that can be fixed) but the weakness in its strength (since that is almost impossible to change). The strength of traditional enterprise software companies is their size – their sales force, marketing budget, and customer base. This is also their weakness. It's a very high-cost infrastructure, with a very high cost-of-sale, that needs a high price for software acquisition and maintenance. Just as you can't simply make a big V8 engine fuel-efficient, you can't make this sales process cost-efficient. Second, people tend not to trust big-budget advertising or pushy quota-driven salespeople. Open Source is about trust – being open about what you have and letting people to try it freely. Hence, Open Source marketing is trust-based marketing.

Rule 4: Discontinuity is King

Open Source is about marketing and business model innovation. However, what's also needed is a discontinuity in the infrastructure to deliver Open Source and the tools to support trust-based marketing versus big-budget marketing.

Rule 5: What is the Barrier To Entry?

Geoffrey Moore organized a way to analyse barriers to entry, discussing "compelling reasons to buy," "whole products," and "competitive strength at the point of attack." The question must be asked, why can't closed enterprise software companies become Open Source overnight? In disruptive markets the new specialist players tend to become leaders – minicomputers, PCs, RDBMSes, application servers all bear this out. There's a reason for this: the old leaders want to keep the status quo. Change is the last thing they want, since it disturbs their happy lifestyle. However, they'll follow a familiar pattern when they start to compete against Open Source. They will:

1. Ignore the technology;
2. Dismiss it as technically inferior;
3. Start to get fewer requests for information and find out about deals after they've already lost them;
4. Try to compete and not understand why they're losing; and
5. Finally, the superpowers change.

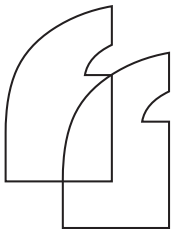
I remember a Digital salesman telling me VMS was superior to Unix and that every Ultrix (Digital's Unix) deal was taking money out of his pocket! On the other side of the coin, I remember in the 1980s relational databases being dismissed as toys. Then the complaint was they can't do transaction processing. The rest is history!

In Open Source the old closed vendors will struggle to move to Open Source for many reasons. Their strength is their weakness and they can't move overnight to change their business model – like downsize/remove their salesforce, which is equivalent to turning a V8 engine into a fuel-efficient engine. They can't effectively use new Open Source technology – and rip apart their proprietary unpluggable code and architecture. They often don't understand the model and think that making their code Open Source or free is the solution. Free and old is not attractive or interesting for a community to form around. This is not unique to software. In the airline industry many airlines tried to have a low-cost brand. The problem was that their brand couldn't be high-cost and low-cost and their infrastructure couldn't be high-cost one minute and low-cost the next minute.

MARKETING

Rule 6: Users Have a New Discovery and Acquisition Process

The way that users choose enterprise software has changed. Users go through a "discover," "research," "try/download," "join community," "buy - support, training, consulting" process. The software is discovered through the Web, SourceForge, Blogs, keyword search, forums, RSS, podcasts, webinars, trials, downloads, traditional media, and word of mouth. Rule 4 discussed "What is also needed is a discontinuity in the infrastructure to deliver Open Source and the tools to support trust-based marketing versus big-budget marketing." These are those tools. Information is no longer power since it's widely available. Trusted sources are widely available. These new tools are also very time-sensitive. Major companies spend weeks fretting about a sentence in a press release. That top-down, time-intensive, review model is now a major weakness since the model doesn't fit the new medium. Open Source companies must become masters of the new medium and masters of speed in the new medium. The strength of large companies again becomes their weakness.



ultimately the traditional software model can't compete **with the marketing and business model disruption and distribution model of Open Source**

MARKETING (continued)

Rule 7: Open Source Companies Have Different Key Performance Indicators

The new discovery and acquisition process means new leading key performance indicators for the company and marketing in particular.

- ***Discovery/Mindshare:*** Press, analyst, blog coverage, web site hits, keyword ranking, number of trained partners
- ***Research:*** Podcast subscriptions, RSS subscriptions, blog subscriptions, wiki hits
- ***Trials and Downloads*** = Number of trials, number of downloads
- ***Community:*** Number of members of the community, number of forge projects

Collaborative Development, testing, translation, support, and knowledge sharing is how to beat large, slow, closed software companies. Collaborative marketing, championed by people who love your software, is also how to beat large, slow, closed software companies. The financial services industry used to live on access to information before other people. Now most information is wide open to most people. The world has changed. The same is true for Open Source software. There are no barriers for people to convey what they dislike/like about closed proprietary companies and what they like/dislike about Open Source companies.

Rule 8: The Value Proposition is Simple

I remember talking to a senior banker at Lehman Brothers and a little thing he said made a lasting impression on me – "The best return on investments (ROIs) I've ever seen I have been able to write on the back of a postage stamp!" The point he was making was that when people make great (career-making) decisions, the ROI is often so obvious it's a no-brainer. The ROI behind Open Source is a no-brainer.

An analogy that someone said to me was that if you have a party you could buy one $1,000 bottle of wine and share it between two people or you could buy 10 $50 bottles of wine for everybody. Most people can't tell the difference and you still have money left over for food. The same is true for enterprise software. Buying expensive closed software has major implications for your budget. Because of the upfront fees you have to go through an expensive and lengthy evaluation process with costly external advice. The software is then so expensive it's typically rolled out only for a department or particular process. Then there's little budget left for training and support. The result is that most people don't benefit from the new technology.

Open Source changes all of that providing a cost-effective rollout for the whole company. Closed enterprise companies will become the expensive boutiques of the software industry.

OPERATIONAL

Rule 9: Your Software Infrastructure is Key

Dell transformed the PC industry not by selling cheap PCs but transforming the whole value chain and supply chain for PC production. From an operational perspective Open Source isn't about cheap software but about transforming the whole value chain for software across development, testing, translation, product management, marketing, sales and support.

The number of people downloading your software, asking questions, accessing your Web site, accessing demonstrations, trialing the product, discussing in forums, updating the wiki ... is massive compared to a traditional software start-up company. The extended infrastructure has to be able to support contributions, bug reports, and fixes from other individuals/companies, take feedback from forums and surveys, and be able to support hundreds of thousands people downloading your software. In amongst this, you have to be able to identify those who want to buy support, patches, and updates for a mission-critical environment and those who want to use the open source as part of the community. Open Source companies have to be masters of the whole Open Source software value chain to support the massive growth potential.

Rule 10: We May Not Be in a Traditional Tornado But the Principles Are Similar

Geoffrey Moore wrote about key behaviours in a tornado and how they changed in the transition from the bowling alley. Open Source is a similar disruption – marketing and business model disruption instead of raw technology disruption. But many of the tornado principles for products still hold true. The product should be simple to install, simple to use, simple to scale out, and simple to develop applications on. It should be standardized as much as possible, reducing complexity, reducing time to deployment, reducing the services required, and making it compatible with industry standards – ultimately preparing it for commodity status and ubiquity.

Summary

After seven years at Documentum there were many things I knew in 2000 that I wish I'd known when I started. The Open Source world is moving so fast that there's a lot I know now I wish I'd known when I started at Alfresco. One of the great things about the Open Source community is that not only are people ready to share code they're ready to share ideas and marketing strategies. What we all believe is that ultimately the traditional software model can't compete with the marketing and business model disruption and distribution model of Open Source. You can reach people, geographies, and companies that are impossible in any other way. Your best salespeople become your users using the new medium to champion you. Open Source is not just about Linux or Eclipse. It's a new parallel universe for software stretching from the operating system to the RDBMS, application server, content management system, business intelligence systems and the office, CRM, and ERP applications built on top of this new infrastructure.

The Power of Being There.®

Times two.

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

Avocent®

cyclades

An Avocent® Company

Visit www.avocent.com/powerx2

Bringing Ruby to the Enterprise

Is Ruby Ready?

by Brad Banister

The Ruby language is generating a great deal of buzz in the software community these days. Developers are becoming interested in Ruby for a various reasons such as its promise of increased productivity, the power of the language itself, or simply its ease of use. At the same time, many who are new to Ruby wonder if it's capable enough to be used in enterprise software development.

This article is written for developers and IT staff who are considering using Ruby in an enterprise environment. It gives a broad overview of the features of the language from a developer's perspective and highlights the key advantages of choosing Ruby in the enterprise.

Introduction to Ruby

Ruby is a modern object-oriented language that shares many of the features of Smalltalk, Perl, and Python, while being incredibly flexible. One of its strengths is its very simple syntax, which makes it highly readable and maintainable. Ruby is also widely considered fun to program in. Much of this relates to the "principle of least surprise." That is, the language was designed to be very natural, to minimize inconsistencies (e.g., everything is an object), and to be concise. These features contribute to its ease of use and enhanced productivity over other platforms. Ruby also has good support for testing, an important facet of agile development methods.

While interest in Ruby has only taken off recently, it's a very mature language that's been around for over 10 years. It's highly portable and available on a wide range of operating system platforms.

The Case for Enterprise Ruby

Enterprise software can be defined as software that processes large amounts of business-critical data. Even a short-term enterprise software outage can cost a business large sums of money. As a result, enterprise-grade software is usually seen in terms of non-functional requirements such as scalability, availability, and reliability.

Applications in an enterprise are often initially created in isolation, whether out of business urgency, expedience, or the difficulty of coordining business units. As a result, an enterprise can find itself with lots of data silos that need processing and correlation. These heterogeneous distributed systems often need to be integrated later in their lifecycle though they may not have been designed with integration in mind.

The recent success of the Ruby platform in the rapid development of database-backed Web sites has brought the Ruby language into the spotlight. It's also brought to the forefront the question of whether Ruby is suitable for enterprise development.

Though the Ruby language has been around for a long time, it's seen limited adoption in enterprise environments. Many still consider it to be a leading-edge technology that lacks support for some common enterprise integration technologies. There's also a lack of "best practice patterns" for implementing an enterprise solution with Ruby.

It spite of this, there are many reasons why it's desirable to use Ruby for enterprise software development. Ruby began as a scripting language, and as such, it's well suited to acting as "glue code" in integrating applications. Scripting languages got their start as a way to coordinate tasks rapidly and flexibly between processes. It follows that Ruby should be capable as a technology for integrating components and services in the enterprise.

Ruby is also a good platform for data manipulation, an important element of enterprise software. Many of Ruby's features in that area build on the features in Perl, a language with excellent support for generating reports. XML data manipulation is another place where Ruby shines.

Productivity also tends to be higher with Ruby than traditional enterprise platforms. Dynamic languages like Ruby can be good tools for rapid prototyping. With a scripting language, the development process follows a fast write-run-test cycle that's a natural fit for the iterative approach of agile development

About the Author
Brad Banister is a principal consultant in the service-oriented architecture practice at MomentumSI. His current research is in applying agile methods and technologies in the implementation of service-oriented architectures.

Comparing Enterprise Java & Ruby

As an aid in determining Ruby's enterprise capabilities, the matrix below compares the enterprise features of Java and Ruby. Note that many of the Ruby frameworks are still maturing. While they still provide valuable functionality, they may not be as full-featured as the corresponding Java framework.

Enterprise Features	Java	Ruby
Web Applications	Struts, Tapestry, and JSF are major Java Web frameworks. Java provides flexibility in the choice of Web framework, flexibility in the use of supporting frameworks (like ORM), and flexibility in the use of each framework at the cost of added complexity. AJAX support is ad hoc.	Ruby on Rails dominates the Ruby Web app framework market. Rails is a full-stack framework including a database access layer. Rails is less flexible than other solutions, since it targets simplicity. Nitro is a Web framework, currently in beta, that offers greater flexibility than Rails. Apps can be written in the MVC style of Rails and major Java frameworks, or in the server pages style of ASP/JSP. Both frameworks have strong AJAX support, integrating with the prototype AJAX library.
Scalability and Availability	Enterprise app servers provide clustering and fail-over services for properly written apps at the cost of added complexity. Alternately, stateless Java apps can be scaled with load balancers just like LAMP (Linux, Apache, MySQL, Perl/Python/PHP) apps.	State-of-the-art for Rails apps is to follow the LAMP pattern, keep the application servers stateless, and push the state to the database. Then use load balancers between tiers to add/remove any number of app servers. Rails provides simplified clustering via a high-performance distributed memory cache.
Object/Relational Mapping	Hibernate is the most popular Java ORM with the flexibility to map to most database designs.	ActiveRecord is the Rails ORM solution. It follows the ActiveRecord pattern of mapping an object to a database table row. This is a simpler design and better suited for mapping to schemas that are similar to the domain model. Lafcadio is an alternative ORM with more sophisticated mapping support for legacy schemas.
Database Connectivity	Enterprise app servers provide sophisticated connection handling, including automatic recovery of leaked connections and connection reservation handling.	ActiveRecord provides simple connection pooling and dropped connection restarting.

methods. Ruby code also tends to be far more concise than similar code developed in other languages. At the same time, the intent of the code is typically clearer. Faster development and easier maintenance are features that directly impact the bottom line.

Ruby is a platform that can be used on any application tier. The Ruby on Rails Web application framework provides a robust servlet container on the presentation tier. For pure server applications, object-relational mapping tools are available to layer on top of drivers for database access. Ruby has frameworks to produce and consume Web Services. There are also options available for creating messaging solutions for enterprise application integration.

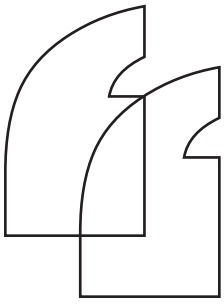
A Guide to the Ruby Platform

A programming language is nothing without a platform of supporting frameworks and libraries and Ruby is no exception. This section provides an overview of Ruby frameworks for building enterprise Web applications and Web Services, and enterprise applications that require database and application integration support.

Ruby on Rails is a full-stack Web application framework that prefers convention over configuration files, putting ease of use in common situations ahead of ultimate flexibility in all situations. Conventions increase productivity by freeing the developer from explicitly having to spell out every intention in configuration files.

Comparing Enterprise Java & Ruby (continued)

Enterprise Features	Java	Ruby
Transaction Support	Containers like Spring provide lightweight transaction services. App servers supporting JTA provide distributed transaction support.	Standard transaction support. No support for distributed transactions though ActiveRecord supports persisting to multiple databases.
Dependency Injection	Spring is the standard dependency injection container for Java.	Needle is a dependency injection container for Ruby. While DI is not as necessary in Ruby because of its dynamic nature, Needle offers service management and AOP-like advice.
Packaging and Deployment	J2EE has the WAR file and EAR file standards.	Rails has a unified packaging and deployment framework called Capistrano, with features beyond WARs and EARs like automated parallel deployment to multiple servers and easy rollback.
Enterprise Application Integration	A variety of JMS messaging products like ActiveMQ are available as are ESBs like ServiceMix. Most any legacy app can be bridged using the Java Connector Architecture.	Ruby can be bridged to JMS implementations like ActiveMQ (and thus ESBs like ServiceMix) that support STOMP (Streaming Text Orientated Messaging Protocol). Ruby Reliable Messaging is a Ruby-only messaging service. Ruby lacks the kind of legacy app connectivity provided by the Java Connector Architecture.
Web Services	J2EE provides many APIs for Web Services support.	Rails has good support for simple Web services via ActionWebService and other libraries like XML/RPC, SOAP 1.1, WSDL, REST.
XML Documents	Fully supported.	Support for parsing and building, and XPath expressions. Lacks good support for XSLT and schema validation.
Distributed Objects	Support for CORBA and Java RMI.	Ruby has an RMI-like facility for distributed objects called Distributed Ruby. CORBA support is poor.
Management	Application management via JMX. A variety of SNMP stacks are available.	No standard app management API or framework. Simple SNMP library is available.
Directory Services	JNDI for Java services and LDAP for others.	LDAP support from a beta library.
Build Systems	Ant and Maven provide full project support, and continuous integration tools like CruiseControl are available.	Rake provides ANT-like functionality. RubyGems provides automated library download support. DamageControl provides continuous integration.
Internationalization	Excellent support.	While Ruby supports Unicode encodings internally, its own string libraries don't. There are libraries that augment i18n string support, and many libraries for specific tasks support i18n, but there's no unified support.
Authentication	Support via JAAS, support for Kerberos.	No standard authentication libraries, or standard single sign-on support.
Platform support	Runs on a wide variety of platforms, OSes, etc.	Runs best on Unix systems, potential issues on Windows platforms.



Ruby isn't ready to be used in every aspect of the enterprise, **but it can fit into many areas of an application strategy**

A primary selling point of Rails is the claim that its users gain a big productivity spike over standard Java in developing Web applications. This fact is attributed to both the simplicity of using the Rails framework, as well as the vastly reduced line-of-code count between Ruby and Java code when doing identical functions. Rails is also more productive than other frameworks because its templating language is itself Ruby, which obviates the need to learn a specialized framework language.

Scalability with Rails can be addressed in two main ways. First, since Rails can run in multiple processes on a single machine via FastCGI fronted by a Web server like Apache, it scales fairly linearly with increases in computing power on a single machine. Rails also scales well across machines using the standard Web server infrastructure techniques of keeping the application servers stateless and using load balancers between the Web server and app server tiers.

Rails comes bundled with ActiveRecord, an object-relational mapping (ORM) library, for database access. ActiveRecord can also be used outside of Rails. ActiveRecord's strength is in mapping database schemas that are either created from scratch or are similar to the kinds of schema that it works best with. In these instances, ActiveRecord is among the simplest ORMs to use. It includes a powerful data validation system and impressive support for automating database schema migrations.

There are several options for database access where legacy database schemas are too complex for ActiveRecord to handle. The first is a mature ORM called Lafcadio. One of its design goals is to offer strong support for legacy schemas. Another option is to fall back from ORMs to direct database connections. Ruby includes a database abstraction layer called DBI, which is similar to JDBC in Java, and allows for database independence. Of course, another option would be to use a database driver directly. Note that ActiveRecord provides access to these other layers, so variations from ActiveRecord's preferred style of schema design can be handled without losing ActiveRecord's benefits.

Note that for working with Rails, the presentation tier doesn't require either ActiveRecord or any database for that matter. Rails can easily be used as a front-end to a Service Oriented Architecture (SOA) or any external service-based API that Ruby can connect to. In this tiered approach, there's a clean separation between the Rails presentation layer and the service and data layers. Different teams can develop the different layers using different technologies.

Ruby offers a number of options for integration with enterprise services. Ruby can be bridged to existing JMS messaging systems that support STOMP (the Streaming Text Orientated Messaging Protocol). STOMP is an open messaging protocol that provides publish/subscribe services, as well as receipt and transaction services. Ruby has good SOAP support, and Ruby classes can be generated from WSDL. It also has a very complete XML-RPC library.

Ruby has excellent base HTTP client support for custom HTTP messaging integration. It offers a WEBrick embedded Web server/servlet container for message exchange. WEBrick can also be easily integrated with existing CGI scripts and provides HTTPS/SSL support.

There are several areas where direct Ruby support for enterprise technologies is lacking. There's no standard Ruby service comparable to JMX for application management. There's also no support for a standard authentication/authorization service. Distributed transactions aren't available through any Ruby transaction manager. There's also no real CORBA support in Ruby. Finally, internationalization support isn't unified like it is in Java. Although there are libraries available to bridge some of the gaps, custom i18n development may be required, depending on the application.

In these areas where Ruby lacks native integration support, there are Ruby-Java bridges available for direct integration with existing Java code, or for bridging to services that Java's better at.

MomentumSI Recommendations

It can be a challenge to decide whether Ruby makes sense for a particular enterprise project. MomentumSI offers guidance on using Ruby on projects according to various enterprise application categories:

- ***Mission critical/non-stop processing:*** Ruby's not ready for an environment that includes integrating with transaction monitors or CICS/IMS interfaces.
- ***Complex transactional:*** Ruby's suitability in a complex transactional environment depends on the nature of the transactions. While Ruby doesn't offer native XA support natively, it can be integrated into transactional messaging systems. This opens up the possibility of using Ruby for enterprise application integration.
- ***SOA/Web Services:*** Ruby can be used as both a producer and a consumer of SOAP-based Web Services, enabling it to be used in Service Oriented Architectures.
- ***Data-intensive:*** Ruby has been proven in data-intensive applications. Examples include space shuttle simulations done by NASA and meteorological number crunching done by NOAA.
- ***Web applications:*** Ruby on Rails is a productivity leader in database-backed Web applications. It scales well and has excellent developer support.
- ***Workgroup software:*** Software in the enterprise targeted at workgroups usually requires rapid prototyping, which is a Ruby strength.

Conclusion

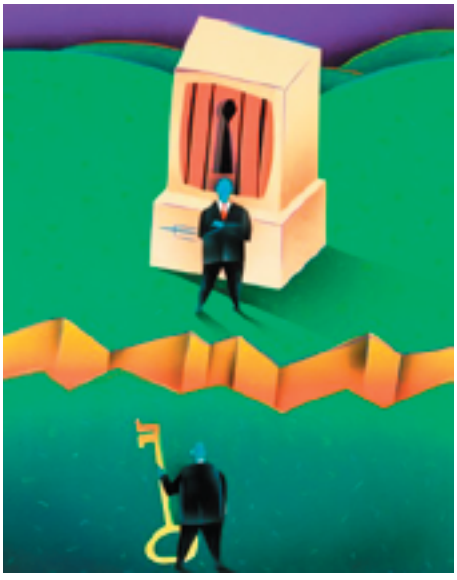
MomentumSI recommends that those businesses looking for a competitive advantage consider using Ruby in an enterprise environment. While Ruby isn't ready to be used in every aspect of the enterprise, it can fit into many areas of an application strategy. By doing so a business can reap the productivity gains from the agility that the Ruby platform offers.

The Open Source Advantage in Secure Application Development

Building a knowledge base

by Christopher M. Frenz

The security benefits and risks of Open Source code is one of the most debated topics in information security today. The views of proponents of the Open Source model are typified by Eric Raymond's argument that Open Source software is intrinsically more secure since its open nature lets a greater number of programmers view the source code and uncover potential security threats before they're released to the wild.



Fewer people see closed source software, on the other hand, and so the odds of uncovering a potential security threat before a system cracker finds it is diminished. Opponents of this model argue that the source code availability of Open Source software lets crackers search the code for potential exploits and provides them a useful way to design attacks. They argue that this makes closed source software intrinsically more secure by way of the principle of security through obscurity.

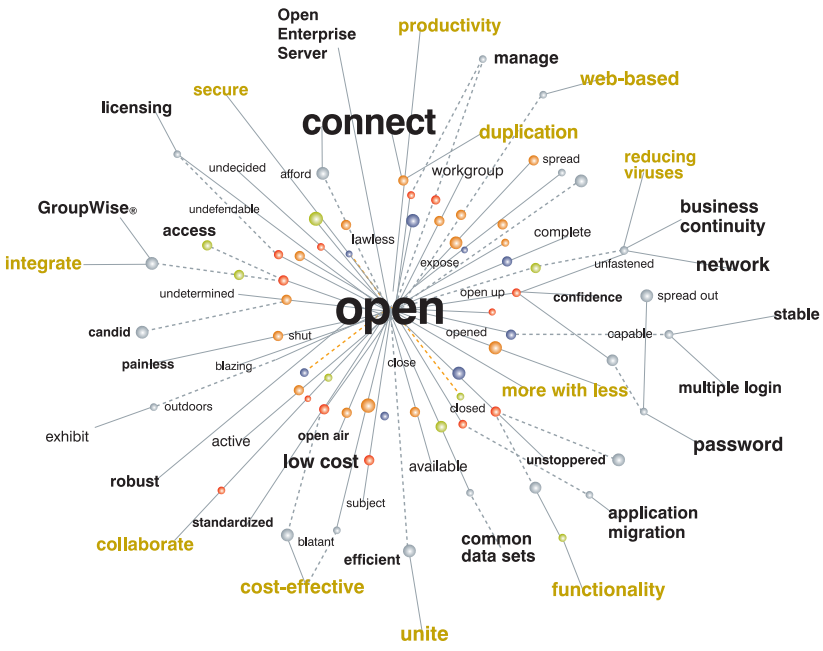
Part of the reason that no clear consensus will be reached on this issue any time soon is that both arguments have elements of truth to them. Knowledge is always a double-edged sword in that it can be applied for either good or evil purposes. In this case, the pro-Open Source argument that accessible source code uncovers vulnerabilities and produces fixes is good, while the pro-closed source argument of crackers being able to use the code for hacking is bad. As with most issues in the world, this duality of good and evil applications means the answer won't be found in arguments at either extreme. Rather the answer lies in weighing the risks and benefits offered by both sides, and as it turns out, this set of arguments isn't unique to computer security.

A similar debate has been going on for some time in biology and its intensity has escalated with the realization that there could be a bioterrorist attack. The controversy lies in scientists publishing, in publicly accessible repositories like Genbank, the DNA sequences of organisms that are known pathogens and dangerous to humans and other animal species. Using molecular biological methods these sequences can, in theory, be used to reconstruct the pathogen and potentially aid in the engineering a more virulent form of the organism. This controversy hit the news with the October 2005 publication of the 1918 influenza virus genome. The 1918 flu virus is estimated to have caused the death of 50 million people. In essence the computer security debate and the debate over the potential misuse of biological data are one in the same. DNA can basically be thought of as an information storage medium whose sequence contains all of the instruction sets necessary for an organism to develop and survive. In essence, DNA sequences are really the code of life, and the issue is whether or not such code should be open sourced.

What should interest the Open Source community is that, while dissenting opinions exist, scientists have generally decided that disseminating information is better than not. Knowledge is a valuable resource because it can generate even more knowledge, which means it can further our ability to cure diseases such as the flu or it can enhance our ability to secure computer applications. For example, scientists are trying to unlock the functional mechanisms that made the 1918 flu so virulent in hopes of developing treatments for the modern bird flu. A big part of this kind of research lies in comparing the virulent 1918 strain against more benign strains trying to pinpoint what's different.

The same principle could apply to computer security, if programmers consider the knowledge contained in the differences between pre- and post-versions of security vulnerability fixed code. Each of these can, in effect, become a case study for other programmers on how to or how not to program something to avoid a given type of vulnerability. Having this code open sourced could form a tremendous security knowledge base on which future programming choices could be made and, in the end, result in the enhanced security of computer operating systems and applications. To facilitate such learning it's imperative that developers thoroughly document their changes and rationale for making those particular changes. In contrast, in a closed source system, when the group that maintains the code fixes a certain kind of security vulnerability, that group may learn how to eliminate that type of problem, but other groups won't benefit from their experience. It's this principle that I think in the end makes Open Source more valuable from a security perspective. The true security benefit of Open Source isn't in a securer today, but in its ability to empower a more secure tomorrow.

About the Author
Christopher Frenz is the author of Visual Basic and Visual Basic .NET for Scientists and Engineers *and Pro Perl Parsing (Apress), and is pursuing his doctorate in biocomputing. In his current research Christopher uses neural networks to model biological systems and develops Web-based scientific applications.*
cfrenz@gmail.com



DEFINE **YOUR** OPEN ENTERPRISE™

In my Open Enterprise, productivity is up because workgroup solutions actually work the way groups want them to.

People are more productive when they have the tools and support they need to work more efficiently. Workgroup solutions from Novell® unite infrastructure, services and tools with unmatched security and reliability. Regardless of location or device. Our full suite of networking, communication and collaboration services support more users on a single server, simplifying administration and significantly reducing costs. So you can communicate and collaborate the way you want, wherever you want—for a lot less than you'd think.

Workgroup solutions from Novell.
This is the way to connect your Open Enterprise.

Novell®

This is Your Open Enterprise.™
www.novell.com/connect

A Closer Look at Damn Small Linux

The take-along operating system

by Russ Ethington

In this article you will learn how to turn a blank CD and an inexpensive USB keydrive into a powerful, portable, take-along operating system complete with modern applications like Firefox, a Web server, and multimedia tools. All this can be done using free Open Source Linux software.

About the Author
Russ Ethington is an enterprise software architect and developer with 15 years of experience in numerous programming languages on Unix and other platforms. His technical interests include distributed computing, programming languages, and music synthesis.
russ.ethington@gmail.com

We'll start with the latest version of a distribution called Damn Small Linux (hereafter referred to as DSL) and work through the steps of getting and "burning" an image, booting it, setting up networking and applications, and saving customizations and files to the USB media. You can do all of this even with an old PC from yesteryear, turning it into something snappy and new. One possible outcome is what you see in Figure 1.

The resulting bootable CD and USB keydrive storage work great without having to install, partition, reformat, or even modify the hard drive on the host system. This highly portable computer system based on DSL will enable you to walk up to almost any PC and boot into your very own familiar Linux desktop, get on the Internet, send e-mail, blog, play games, even listen to Internet radio, and save your work, shut it down, and take it all of it with you without leaving a trace. It will be lightweight, fast to boot, and can breath new life into almost any computer, old or new. You can even use this technique to create low-cost Internet kiosks and Web servers, as an alternative to carrying a laptop, or just to impress your friends and colleagues.

The DSL distribution of Linux has long set the standard for fitting maximum functionality into a small 50MB footprint. The discipline of keeping the entire distribution to this size has resulted in a compact and versatile Linux (derived from the venerable Knoppix, itself based on Debian Linux) that can run efficiently even on older PCs while recognizing a wide assortment of hardware. Through the innovative "MyDSL" subsystem, nearly unlimited customization and system expansion are possible.

There's a lot to look at inside this unusual little operating system, not all of it beautiful. This article will help you find the best of DSL and ignore the rest. Best of all, you'll discover how to shape and customize DSL into something useful and appealing, perhaps even something unique.

Getting Started: Obtain, Burn, Boot

Get the Latest DSL Image

Downloading the latest stable version of DSL is free and easy. If you have a high-speed Internet connection it can be done in minutes. Simply follow these instructions:

1) Visit the URL http://www.damnsmalllinux.org/download.html
2) Click on one of the mirror sites under "where to download"
3) Look for a directory called "current" and click on that to see its contents
4) Look past most of what you see and download the file "dsl-<version>.iso"

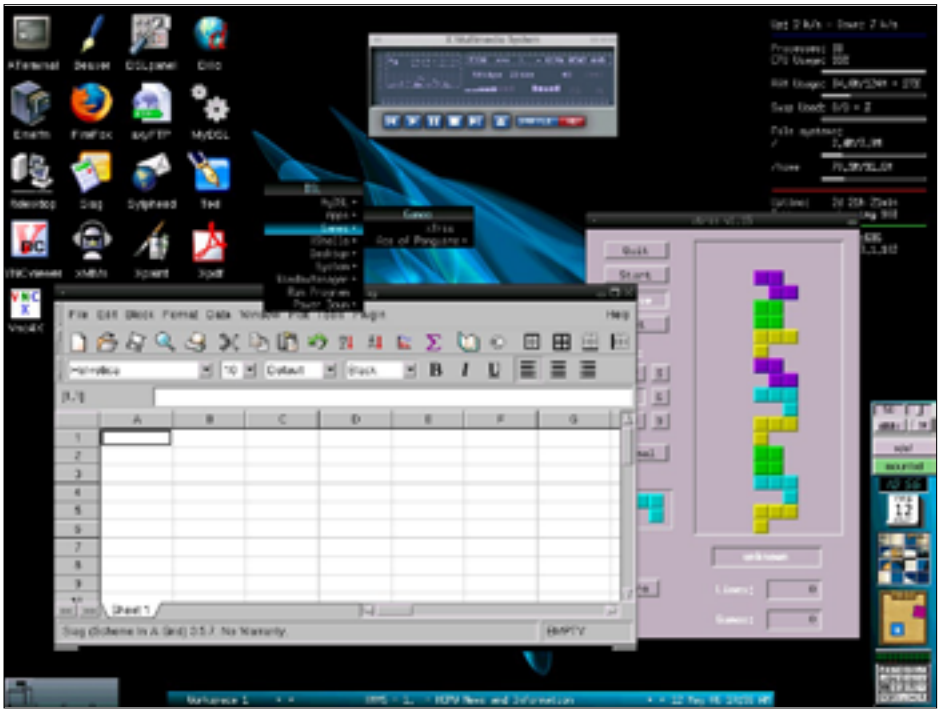


Figure 1: DSL in action with one of many possible customizations

Where I've written <version> above you'll see something like "2.4" depending on when you read this article. The current stable version is 2.3. Save your download in a place where you can find it as you move to the next step.

Burn the ISO onto CD

The file you just downloaded has an "iso" extension after the ISO standard format used on CD-ROMs. Some operating systems — such as OS X — can mount an image file like this and show you its contents. We don't really have to do that but if you're curious go ahead and see if your computer can mount the disk. You'll need a writable CD-R disk and a CD burner for the next step. Use your favorite CD-burning software to create a disk from this image.

One important note here. The DSL disk image is much smaller than the CD's capacity and that's expected. At only 50MB you could fit about 15 copies of DSL on a standard CD-R. DSL was originally configured to fit on the smaller 50MB "business card" CDs that were briefly popular. This size constraint now offers great advantages for this little operating system. As you'll see, DSL is so small that the entire operating system can boot up into the memory on just about any PC without even accessing the hard drive.

When you burn your CD remember that you're dealing with a bootable ISO image. Rather than just burning a copy of this file onto a CD (which won't boot properly in the next step), you'll want to instruct your CD burner to recognize the ISO and burn a disk image.

Boot from CD (or Use a Helper Floppy)

This step could be as simple as putting your newly minted DSL CD into the drive on your computer and rebooting. You should try that first before reading any further. By the way, many people get nervous about doing anything experimental with a working computer and rightly so. In my experience I've seen no risk in booting a PC into DSL. This little operating system will detect much of the hardware attached to your computer including your network, but it will completely ignore your hard drive and data until you tell DSL to do differently. If you're still not sure, just dig out that old PC in the closet and try DSL on that first. An old machine running DSL may surprise you as it comes back to life. Now let's get back to the boot. If your PC recognizes the CD as bootable and begins a boot sequence, skip to the next section.

Most PCs can boot from a CD (the CD booting standard has been out for over 10 years), but they may not be configured to do so. To find out you'll have to get into the BIOS settings at boot time and check. As your machine starts up, look for a message that says some-

> it will seem like you can take your computer everywhere **without really having to take your computer anywhere**

thing like "Press F-10 for BIOS Settings" and use whatever key you are instructed to press.

Next, while in the text-based BIOS menu system, look for a section on boot devices and boot order. Make sure that the CD-ROM is somewhere in the order of boot devices ahead of the hard drive (usually it goes in the order of floppy, CD, then hard drive). Find a way to put the CD-ROM in the boot sequence, save your changes then try a reboot.

At this point I should mention that one of the machines I used to prepare this article (and even write some of it) is a 10-year-old Pentium 150MHz with only 96MB memory. It can't boot from a CD so I used a helper floppy that can recognize and boot the CD. If you need such a thing, you can create one by following these steps:

1) Download the boot floppy boot image file for DSL
   Download: bootfloppy.img
   From Site: ftp://ibiblio.org/pub/Linux/distributions/damnsmall/current
   Save As: boot.img
2) Create a boot floppy using a blank 1.44MB floppy disk
   If running Windows
   Download: rawrite.exe (or other suitable variant for your OS)
   From Site: http://www.fdos.org/ripcord/rawrite/
   Run Command: rawrite -f boot.img -d A
   If already running a version of Linux:
   Run Command: dd if=boot.img of=/dev/fd0 bs=1440k

Once your computer is booting from the CD (or a combination of floppy/CD) you're ready to begin exploring the world of DSL, customizing it to your liking, and creating a configuration that you can easily return to on your next boot.

Advanced Topic: Bootable USB Keydrive

(aka Pendrive, Flashdrive, Etc.)

You may have heard about bootable USB keydrives or know someone who has one. With DSL there are at least two ways to get your USB-capable computer to boot from a keydrive. The first works without having to reformat the keydrive. The second approach will dedicate the entire USB keydrive to running DSL by creating a boot partition and a data partition.

You can skip this section entirely or return to it later if you decide that booting from a keydrive is for you. Regardless of how you end up booting, the rest of this article will assume that you're simply booting from a read-only device and intend to use a keydrive for take-along storage. This way everyone can come along with or without a bootable USB drive.

Quick and Easy Install to Keydrive

(Requires Helper Floppy)

The simplest and least disruptive approach to booting from a keydrive is just to make the keydrive look like the CD you used to get started. I made this work by copying all of the files from the DSL CD (mounted as /mnt/cdrom) to the root directory of a keydrive (it will have to be mounted as /mnt/sda1, as shown below) already formatted with the MS-DOS file system. Of course you'll need at least a free 50MB on the USB keydrive before you copy the files.

If you don't know how to copy files in Linux, first bring up a terminal window by clicking on "ATerminal" on the DSL desktop. Next type the following:

```
$ sudo mount /dev/sda1
$ cp -r /mnt/cdrom/* /mnt/sda1
```

If you're allergic to the command line, these two steps can be carried out intuitively using the "Emelfm" file manager also available on the DSL desktop.

This approach to booting will preserve the data you already have on the keydrive. It won't make the drive into a genuine Linux boot disk (we'll do this in the next section), so you will need a helper floppy to get the machine booting from this USB keydrive. I made a USB boot floppy while running DSL itself. This is simpler than the manual approach described above but note that you'll need a different boot image (bootfloppy-usb.img) than we used before.

You can create a USB boot floppy with the pre-installed DSL floppy disk tools. Right-click in the DSL desktop to bring up the main DSL menu. Under "Apps | Tools | Make Boot Floppy" select "USB Boot Floppy" and follow the instructions using a blank 1.44MB floppy disk. Once you've created the floppy, leave it in the drive and reboot with the keydrive plugged in.

This new floppy will find the keydrive you've plugged into the USB port and try to boot DSL from it.

Genuine Install to Keydrive

(Still May Require Helper Floppy)

For this approach you have to be prepared to erase and reformat the contents of a keydrive for dedicated DSL use. A 256MB keydrive, which will do nicely, runs less than $20 new. To create a dedicated bootable copy of DSL on a keydrive, read on!

To get this to work you'll also have to know which USB boot standard is implemented in the BIOS of your computer. For computers that can boot from USB drives, the BIOS will support one of two formats. Bring up the BIOS setup utility again (as described above) to find out which, if any, your computer will support. With the setup utility try to put USB-HDD or USB-ZIP before CD-ROM and HDD in the boot order or your keydrive will be ignored when you reboot.

To proceed with the installation you'll need to boot back into DSL from the CD. With DSL running and your keydrive plugged in but not mounted (that just means don't do anything special with /mnt/sda1 after plugging it in) right-click in the DSL desktop to bring up the DSL main menu. Under "Apps | Tools | Install to USB Pendrive" you'll find two choices. These are "For USB-ZIP Pendrive" and "For USB-HDD Pendrive."

Choose whichever standard your computer recognizes and follow the instructions. When prompted for the device name for the installation use "sda." The program will proceed to partition and format the drive as /mnt/sda1 (with all of DSL) and /mnt/sda2 (an empty partition for your files using the remaining

IT'S THE END OF THE LINE FOR SPAM.

PUT IT IN THE CAN.

**Ads, fraud schemes, porn, viruses, phishing attempts…
if it's offending e-mail, Roaring Penguin Software's CanIt™
solutions will stop it before it reaches your mail server.**

- Simple administration
- Automatic spam and virus updates
- Compatible with all e-mail servers
- Available as software or hardware appliance
- Flexible end-user controls

**Throw the book at spam today!
Request a free 20-day CanIt evaluation
and stop spam on YOUR terms!**

ROARING PENGUIN SOFTWARE INC.

www.roaringpenguin.com +1 (613) 231-6599 sales@roaringpenguin.com

storage available on the keydrive).

In the case of my older test PC neither of the boot standards was supported by the BIOS so I created a USB-HDD install and combined that with a helper floppy. Many people have found getting a USB keydrive booting tricky even with a BIOS claiming to support it. If you get stuck making the keydrive boot on its own you can easily create a helper floppy using the instructions in the previous section. This helper floppy works for both forms of USB keydrive installation described in this article. Besides, having that helper floppy will all but guarantee an easy boot on an unfamiliar computer.

Getting on the Internet: Browse, Blog, E-Mail, Chat

By all accounts, DSL seems to do well with most LAN cards and adapters on desktops, many PCMCIA LAN cards on laptops, and with a smaller but growing number of PCMCIA wireless network cards. Both my test machines have PCMCIA adapters (one is a Gateway Tower desktop, the other an IBM Thinkpad laptop). With the Netgear FA411 10/100 Mbps network adapter card I used, DSL was able to recognize the card, configure it, and acquire an IP address automatically via DHCP through my router. I was on the Internet before I even realized it.

Your best bet getting started with DSL is to walk up to a PC that is already wired to a working LAN configured to give out IP addresses using DHCP. If the machine can boot into some version of Windows and get on the Internet, chances are good that DSL will boot and do the same. For those of you who have a different setup (maybe dial-up or a wireless setup with an unrecognized adapter) I have included some helpful configuration resources below.

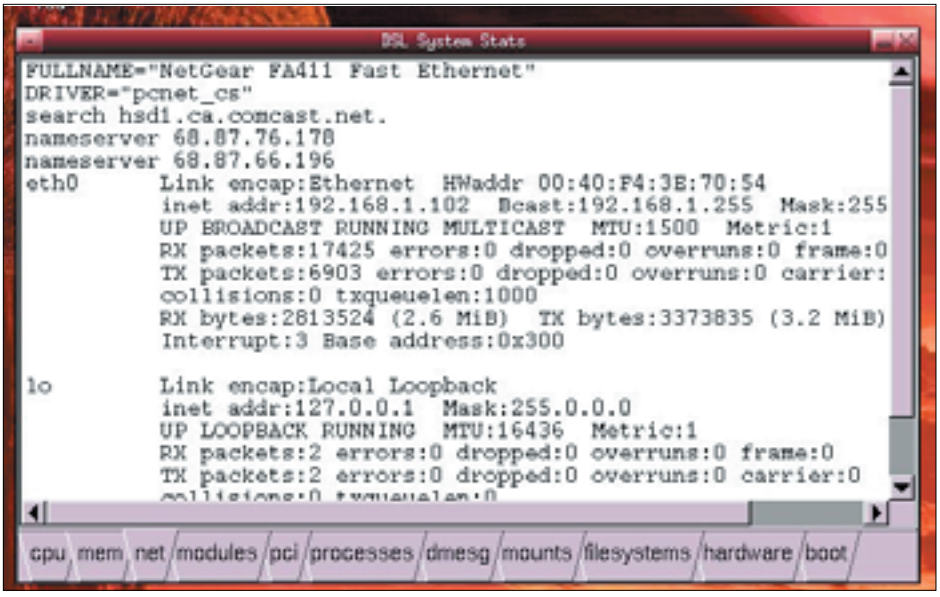


Figure 2: Network statistics for a working network card

Figure 3: Another customization showing off Firefox and Google Maps

Check Your Network Connection

If your machine boots right up and jumps onto a network with a direct "line" to the Internet you'll be able to browse right away using the "Dillo" browser. As a test, try to access http://www.google.com.

Assuming that works you can skip to the next section or read on and learn about some useful diagnostic tools included with DSL. If your first attempt to browse with Dillo didn't work, it could just mean that you're on a network with a proxy server. Check with an administrator or a working computer nearby to see if the browser needs to be configured with proxy settings. In Dillo you do this by clicking the "V" menu, select "Options," and click the "Network" tab to fill in your proxy settings.

To see if your machine can really communicate on a network right-click on the desktop to bring up the main DSL menu. Select System | Control Panel. Click the "System Stats" button and then click on the "Net" tab (see Figure 2). What you see here will be cryptic but informative. There should be a card and driver name followed by one or more "nameserver" entries, one per line. Name servers enable your machine to resolve names like "www.google.com" into real network addresses on the Internet. Further down you should see a section "eth0" containing your computer address listed under "inet addr" as a four-part number like "192.168.1.102." Further along in the same section of text you should find some statistics on the number of bytes transferred by the card since it was booted. This is further evidence of a working network.

Figure 4a: The simple but powerful MyDSL application installer

On the other hand, you may not find the information described above. In that case your next stop is the "Hardware" tab of the "System Stats" window. This is where your network card, if it was recognized, will be listed. On my test laptop, for example, I see two entries for the hardware:

```
NETCARD_FULLNAME=''NetGear FA411 Fast Ethernet''
NETCARD_DRIVER=''pcnet_cs''
```

If you see your card listed but it's not reported on the "Net" tab, you can try using either the "Netcardconfig" tool for a wired network card, or the "Wlanconfig" tool for a wireless networking card. Both are available from the Control Panel. Troubleshooting through these tools is beyond the scope of this article but see Resources for more information:

Fire Up Firefox

Having made a successful test of the Dillo browser (to http://www.google.com), try doing a couple of searches to convince yourself that you're really on the Internet. Before long you'll be ready for a more powerful browser. And you're in luck. Try clicking on the "Firefox" icon on the DSL desktop to bring up Firefox.

Figure 3 shows a screenshot of my DSL desktop running Firefox complete with RSS feeds and a favorite Web site.

You can see that I've been busy customizing the DSL "fluxbox" window manager with styles, applets, a background image, and extra applications. In the next section you'll learn how to do this, but — unless you're already too busy blogging — first why not check out a few more built-in applications for the Internet?

E-mail and Chat Programs

It seems like most people do e-mail with a browser at least part of the time. I use http://gmail.google.com, which has a rich user interface. This application is a good test since it really gives Firefox a workout. Try your favorite e-mail Web site. Or you can bring up and configure the built-in "Sylpheed" e-mail client program.

DSL includes text-based applications for doing text messaging. These support three popular protocols: AIM, IRC, and ICQ. Other more full-featured chat applications are available in the MyDSL application repositories.

Access the MyDSL Application Directory

The MyDSL package system for DSL really sets this little Linux distro apart. It comes to you packed with as many applications as will fit in 50MB. With MyDSL you can download and install (or set aside for later) as many applications as you want or need. These can be saved to a keydrive and brought along wherever your travels take you.

Access MyDSL by clicking the "MyDSL" icon on the DSL desktop. You're presented with a barebones window of buttons, each representing an application category (see Figure 4a). Click a button to get a list of applications in that category (see Figure 4b). Click an application to get a description. Click "Download" to put the application into a temporary (for now) directory and automatically install it in the "MyDSL" menu under the DSL main menu.

In the next section you'll learn how to keep these applications on a keydrive and even have them installed and ready to go the next time you boot up.

Saving for Next Time: Settings, Applications, Data

Accessing that Keydrive

When you insert a USB keydrive into a computer running DSL its contents will appear under the directory /mnt/sda1. The drive must be mounted before you can see your files so don't panic if /mnt/sda1 appears to be empty. Just use the mount tool on the DSL desktop. Click the button marked "fd0" until you see "sda1" and then click the red "unmounted" button once. It should turn green and read "mounted." That's all there is to it. Your keydrive is mounted read/write and ready to use.

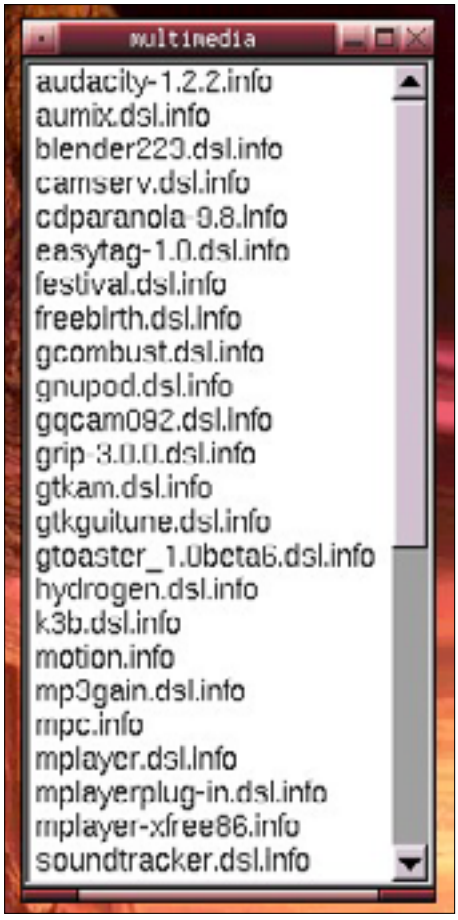


Figure 4b: A list of multimedia applications available with MyDSL

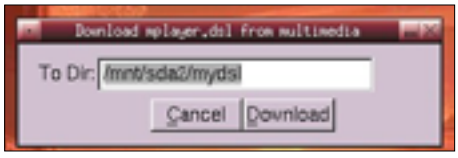


Figure 4c: Downloading and installing an application to the keydrive

Note: if you did create a bootable keydrive, yours will have /mnt/sda1 and /mnt/sda2. Use /mnt/sda2 to save your work and MyDSL extensions since it's created for that purpose.

Next, bring up Emelfm (or a terminal window) and create the following directories under /mnt/sda1 (or /mnt/sda2):

- ***mydsl*** – DSL will look here automatically for MyDSL apps.
- ***myfiles*** – I use this as a good place for other files I want to keep.

Installing Themes and Applications

To get interesting desktop and menu themes, you can visit the "Themes" area of MyDSL. When prompted for a place to save your theme use the full path to your "mydsl" directory you created above. Themes you download will become available under the menu: Desktop | Styles. Figure 4c shows a theme download to the writable storage of a bootable keydrive.

Installing applications works the same way only these will appear under the menu called MyDSL. If you want to download some applications but don't want them installed every time you reboot you can create a subdirectory under /mydsl called /optional. When you save MyDSL applications to the "optional" directory they appear in a separate sub-menu named "Install Optional Extensions" ready to install.

Saving Other Data

Other files that you download or create should be handled with care. You'll want to make sure to save on the permanent storage of your keydrive rather than a transient directory elsewhere. This is where your /myfiles subdirectory comes in handy. Save to this directory and you can count on finding your files again on the next reboot, even if you boot up on a different computer next time.

Taking it Along: Everything-To-Go with MyDSL

Restoring Customization and Saved Applications

Everything you tucked away on your keydrive will be available next time you boot up. This happens in part because DSL makes a backup of some important system files and directories when you shut down. To make sure this happens properly it's a good idea to do a manual back up before you shut down the computer.

To back up, choose the following from the DSL main menu: System | Backup / Restore. Next enter just the device name where you want your backup saved. If you booted from a CD and have a keydrive inserted, use: sda1. If you have a bootable keydrive with a second partition for data use: sda2. Click "Backup" and wait for the program to finish before shutting down. This creates a compressed archive file called "backup.tar.gz" in the root directory of the filesystem.

When you boot up next time, DSL will search for and discover its backup archive on the keydrive and restore the files and settings back to the way you last saw them. Now any computer you boot up with your take-along CD/floppy/keydrive will seem like your very own.

Other Uses for DSL

Internet Kiosk

Let's say you want an inexpensive way to convert some old PCs into a collection of Internet kiosks for an art gallery, business directory, or school project. Running DSL with Firefox would make this easy.

First, you can make Firefox start up automatically at boot time. Just edit the file called ~/.xinitrc and insert this command:

```
/usr/local/bin/firefox http://your.home.page &
```

For a kiosk it would be good to disable extraneous menus and applications and limit the scope of the browser. For more information on building Linux-based kiosks, there are a number of how-tos available (Google: "linux kiosk"). Once you've got one kiosk working, it would be easy to duplicate the contents of its keydrive onto others to create more kiosks.

Web Server
(System | Control Panel | MonkeyWeb)

For fun I turned my bootable keydrive into a portable library of e-books hosted on a Web server. It runs a MonkeyWeb server — included with DSL — to serve out electronic books that I downloaded for free from Project Gutenberg (www.gutenberg.org). First I copied a number of classic e-texts into the /dsl/Share directory. Then I started up the MonkeyWeb Web server (included with DSL) from the System | Control Panel menu. Now users can access these books from any computer on the network by browsing to http://192.168.1.102/Share. With a few changes this machine can be configured to automatically serve out e-texts from a better-named URL every time it boots.

Remote Desktops

Another handy trick is to make DSL available remotely from another computer. To do this I downloaded VNC (called "vncserver") from MyDSL. VNC is a remote desktop program. Many client programs exist for Windows and Mac. I use the popular "Chicken of the VNC" (Google: "vnc chicken") for OS X to access my "headless" DSL (i.e., no monitor). Once you've installed "vncserver" from MyDSL, select MyDSL | vncserver | vnc4x0 and connect to your DSL machine using its DNS address, e.g. 192.168.1.102. When running full screen, the PowerBook appears to be running DSL.

In summary, I've found this take-along operating system to be a great companion for doing most of what I need from a computer. Once your keydrive is configured the way you want, it will seem like you can take your computer everywhere without really having to take your computer anywhere.

Resources

- *DSL Forum on Networking* http://damnsmalllinux.org/cgi-bin/forums/ikonboard.cgi?act=SF;f=7
- *DSL Wiki on Verified Wireless Cards* http://damnsmalllinux.org/wiki/index.php/Verified_Wireless_Cards
- *List of Working Netwoork Cards (using ndiswrapper):* http://ndiswrapper.sourceforge.net/mediawiki/index.php/List
- *Installing and Using Windows Drivers (with ndiswrapper)* http://ndiswrapper.sourceforge.net/mediawiki/index.php/Installation
- *More Tips on Using "ndiswrapper"* http://ndiswrapper.sourceforge.net/medi-

Open Source Storage®

Data Center Solutions Built on OPEN STANDARDS

OSS-1000 1U OPEN Series

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

OSS-2000 2U OPEN Series

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

OSS-3000 3U OPEN Series

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

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

AMD 64 Opteron
The AMD64 Platform

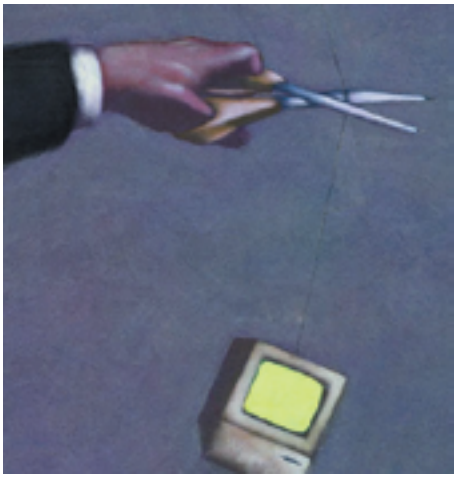
:: 1195 Borregas Avenue,Sunnyvale, CA 94089 ::

The High Cost of Independence

Putting the independent in independent software vendor

by Billy Marshall

The acronym ISV stands for Independent Software Vendor. Historically, independence was important to protect customers from the proprietary lock-in associated with third-party components such as hardware or system software. A greater choice of interoperable components gave customers greater flexibility to procure and assemble a system that met their needs. Microsoft alleviated some of this concern with the Windows platform because customers could always choose multiple hardware providers when selecting applications that ran on Windows. Of course, an application that only runs on Windows isn't exactly an "independent" application, but customers seem to accept hardware independence as sufficient freedom. (More on Microsoft and Windows later.)



About the Author
Prior to founding rPath, Billy served as Red Hat's vice president of North America Sales from 2001 until 2005. He conceived and oversaw the launch of Red Hat Network, the platform that enabled Red Hat's subscription revenue model. Billy also worked in IBM Global Services where he worked with global leaders such as Boeing, Ford, Eaton, Mercedes Benz, and Raytheon.

Unfortunately, independence has a high cost these days. Customers are burdened with the expense of assembling and maintaining components that are sub-optimal because they are not engineered as integrated solutions. Application providers are burdened with the engineering and customer service expense of delivering multiple implementations of their software for multiple operating systems. The costs associated with engineering, assembling, and maintaining the application across multiple operating system environments do not add any value for the customer.

The market is responding to the inefficiency of this legacy software approach by rewarding vendors who remove the burdens of assembly and maintenance from the customer by engineering an integrated solution. The amazing popularity of on-demand software as a service (SaaS) solutions (e.g., salesforce.com) and integrated hardware appliances (e.g., google mini) can be largely attributed to the simplicity and ease of use these solutions offer to customers. By sacrificing their independence from the operating system and embracing Linux and open source, software providers such as salesforce.com and google can offer their valuable applications to customers without the legacy hassles of assembly and maintenance.

Ten years ago, the first-generation on-demand SaaS providers were launching their products into the robust software market created by the Y2K frenzy. Typically, they were re-hosting third-party applications from vendors such as PeopleSoft or SAP or Oracle on proprietary Unix platforms such as Sun Solaris or HP-UX. None of these first-generation SaaS companies was a success because they couldn't profitably offer customers a lower price. Some may argue that they failed because the solution's performance was poor because of an immature Internet infrastructure and an application architecture that was not Internet-friendly, but I think they would have failed anyway. Customers still paid for the software licenses, hardware, and system administration, but they paid for it monthly. Also, the SaaS provider deployed the application exactly as the customer would have deployed it in his data center. These early SaaS companies didn't provide any economies of scale associated with higher system utilization because each customer had dedicated systems with dedicated licenses. The only resources they shared were the network infrastructure, data center infrastructure (power, cooling, etc.), and system administrators.

By contrast, the current crop of on-demand SaaS companies is very successful because they've changed the economics of software.

How are these new SaaS providers so different from the failed SaaS companies of the late '90s?

For starters, they own their application code, so they're not simply passing along a license from a third party in the form of a lease. Also, by owning the code, they can architect the application for high utilization of capital assets. Not only do these applications universally run on Linux, they also leverage other open source infrastructure such as Apache and Tomcat. In addition, they use low-cost industry standard hardware and they have a multi-tenant architecture so that multiple customers can share the same hardware assets. The result is an infrastructure that's tuned for high performance and high utilization of assets.

But there's even more value for customers in this model. Because these application providers are not "independent" from the infrastructure, customers benefit from a lower cost of engineering and customer service as well. The typical ISV will spend between 25% and 40% of engineering and customer service expense on "context" issues as opposed to "core" application features. "Context" issues are things such as installers, multi-platform and multi-version operating system ports, and system configuration. The on-demand providers don't have any of this "context" overhead, so they can focus exclusively on the "core" issues of application performance and features. Customers get a better application with fewer integration hassles at a lower cost. The SaaS provider gets a business model superior to the historical ISVs. They can focus their resources on the features that add the most value for the end user.

Also consider the case of the hardware appliance vendors that market integrated systems for tasks such as network security, authentication, data storage, and spam control. These aren't low-margin undifferentiated hardware products. Their gross margins look much more like those of a successful software franchise because the value they provide is the application functionality they deliver. Yet rather than be "independent" and foist the integration problem onto their customers, they choose to deliver an integrated solution that's easy for the customer to deploy and manage. Like the on-demand SaaS providers, most of these vendors choose Linux or FreeBSD as the operating system of their "appliance" to gain the benefit of rock solid system services and industry standard hardware compatibility. Linux and FreeBSD are also flexible so that the platform can be tuned to the application's needs. Finally, Linux and FreeBSD are free from the distribution restrictions that might challenge the "independence" of these integrated solution providers. The freedom of open sourcde provides true independence.

The final nail in the coffin of independence for software vendors may be the availability of high-performing virtualization technology for industry standard hardware. Virtualization technology such as that provided by VMware and the free software project Xen not only addresses the issue of asset utilization by allowing multiple "systems" to run on a single server, it also addresses the issue of interoperability by allowing those "systems" to be different without creating incompatibilities. "Certification" will simply mean that the application comes in a system form that's compatible with the underlying virtualization layer, thereby allowing it to interoperate over sockets with all other applications on the system. With virtualization as the standard for application integration, customers can simply require that their application providers deliver an integrated virtual software appliance that's compatible with their virtualization layer standard. The only "context" engineering the application providers have to manage is the interface between their system and the customer's virtualization standard.

Given the success of these two models in the market, it's amazing that most ISVs simply treat Linux like "yet another OS" that they must support based on customer demand. Another port means more engineering expense, and perhaps a negative impact on revenue for certain CPU-based licensing models. Since Linux runs on processors that are often two to three times faster than proprietary Unix platforms, a customer might only need half to a third the number of application licenses for a given workload. For these vendors, Linux is a miserable combination of lower revenue and higher costs because they do not consider how it might be used to completely change their business.

What is it about Linux and open source that these vendors can use to create a better business? For starters, Linux is free. Not free as in "free beer," but free as in "free speech." No one owns Linux, which is good if you want to use it to change your business. You don't want your business beholden to any third party that you don't control, or you may find your business held hostage by an erstwhile "partner" in the future. Second, Linux is flexible. It can be tailored to maximize the performance of your application. There are also an almost unlimited number of free software utilities that can be readily added to Linux to enhance the value of your application. Apache, Tomcat, and Struts are just a few good examples. Third, Linux runs on industry standard hardware, so there's no threat of customer lock-in from proprietary hardware vendors.

Given these lessons from the market, it still is not obvious how independent software vendors might win their freedom. There are legacy issues to consider. It's impractical to discontinue support for existing deployments on "independent" systems. And there's the issue of revenue recognition. Moving to a subscription model probably means deferring revenue, which in turn creates a mismatch between revenue and operating expenses on an income statement. Finally, not all applications can be delivered as multi-tenant, on-demand services over the network, and it does not make sense for all software vendors to consider delivering a hardware appliance solution because their customers do not want to add another hardware platform to receive the value of the application.

One transition approach for software vendors to consider to lose their independence and gain their freedom is the software appliance concept. Think of a software appliance as a hybrid of the on-demand application

—continued on page 46

rPath

rPath provides rBuilder and rPath Linux, the first platform for creating and maintaining software appliances. Software appliances bring the simplicity and value of Software-as-a-Service (SaaS) to on-premise application deployments. Using rBuilder, application providers can migrate to a high-value software subscription model while reducing their development and QA costs.

rPath was founded in 2005 by Erik Troan, former VP of engineering at Red Hat, and Billy Marshall, former VP of North America sales at Red Hat, based on their deep expertise in Linux and related Open Source technology. The company was commercially launched in January 2006 with corporate headquarters in Raleigh, North Carolina.

rPath provides a platform for creating and maintaining software appliances. Think of a software appliance as a hybrid of the on-demand application model and the network appliance model. The customer gets an integrated solution in the form of an installable CD or a virtual machine file that readily installs and runs on industry standard hardware. Maintenance for the entire solution comes from the application provider via a simple Web user interface. Since maintenance comes from the application provider, it's pre-tested and certified in the exact environment the customer has deployed, so there are no more mismatched maintenance streams from various vendors.

Linux Processes: Structure, Hangs and Core Dumps

Efficient and effective resolution practices

by James Kirkland, David Carmichael, Chris Tinker, and Greg Tinker

About the Book

Product Details:
ISBN: 0131855158
Format: Paperback, 624pp
Pub. Date: May 2006
Publisher: Prentice Hall Professional Technical Reference

Troubleshooting a Linux process follows the same general methodology as that used with traditional UNIX systems. In both systems, for process hangs, we identify the system resources being used by the process and attempt to identify the cause for the process to stop responding. With application core dumps, we must identify the signal for which the process terminated and proceed with acquiring a stack trace to identify system calls made by the process at the time it died. There exists neither a "golden" troubleshooting path nor a set of instructions that can be applied for all cases. Some conditions are much easier to solve than others, but with a good understanding of the fundamentals, a solution is not far from reach.

This article is an excerpt from Linux Troubleshooting for System Administrators and Power Users *by James Kirkland, David Carmichael, Christopher Tinker and Gregory Tinker, published by Prentice Hall.*

About the Authors

James Kirkland is a Senior Consultant for Racemi. He was previously a senior systems administrator at Hewlett-Packard. He has been working with Unix variants for more than 10 years. James is a Red Hat Certified engineer, Linux LPIC level one certified, and an HP-UX certified system administrator. He has been working with Linux for seven years and HP-UX for eight years. He has been a participant at HP World, LinuxWorld, and numerous internal HP forums.

David Carmichael works for Hewlett-Packard as a technical problem manager in Alpharetta, Georgia. He earned a bachelors degree in computer science from West Virginia University in 1987 and has been helping customers resolve their IT problems ever since. David has written articles for HP's IT Resource Center (http://itrc.hp.com) and presented at HP World 2003.

Chris and Greg Tinker are twin brothers. Chris began his career in computers while working as a Unix System Administrator for Lockheed Martin in Marietta, Georgia. Greg began his career while at Bellsouth in Atlanta, Georgia. Both Chris and Greg joined Hewlett-Packard in 1999. Chris's primary role at HP is as a senior software business recovery specialist and Greg's primary role is as a storage business recovery specialist. Both Chris and Greg have participated in HP World, taught several classes in Unix/Linux and Disk Array technology, and obtained various certifications including certifications in Advanced Clusters, SAN, and Linux.

This excerpt explains various facets of Linux processes. We begin by examining the structure of a process and its life cycle from creation to termination. This is followed by a discussion of Linux threads. The aforementioned establish a basis for proceeding with a discussion of process hangs and core dumps.

Process Structure and Life Cycle

This section begins with an overview of process concepts and terms, noting the similarities and differences between UNIX and Linux. We then move on to discuss process relationships, process creation, and process termination.

Process/Task Overview

It is helpful to begin with a general comparison of processes in UNIX and in Linux. Both operating systems use processes; however, the terminology employed by each differs slightly. Both use the term "process" to refer to process structure. Linux also uses the term "task" to refer to its processes. Therefore, in Linux, the terms "process" and "task" are used interchangeably, and this chapter also so uses them. Note that UNIX does not use the term "task."

The process structures of the two operating systems differ more dramatically, which is easily recognized when observing a multithreaded program in action. The thread is the actual workhorse of the process and is sometimes referred to as a lightweight process (LWP). In Linux, every thread is a task or process; however, this is not the case with UNIX.

As described previously, the UNIX process model places its threads within the process structure. This structure contains the process's state, process ID (PID), parent process ID (PPID), file table, signal table, thread(s), scheduling, and other information. Thus, there is only one PID for a process that can have many threads. However, when a process calls the pthread_create() subroutine in Linux, it creates another task/PID, which just happens to share the same address space. Figure 1 depicts this fundamental difference.

Figure 1 Comparison of UNIX and Linux processes

Unlike UNIX, Linux does not have a kernel object that represents the process structure; instead, it uses a task structure. Each task has a unique ID just like a UNIX PID. However, the Linux task model only represents a single thread of execution. In this way, a task can be thought of as a single UNIX thread. Just like the UNIX process structure, the Linux task structure contains the task's state, PID, PPID, file table, address space, signals, scheduling, and so on. In addition, it contains the Task Group ID (tgid), which we elaborate on later in this chapter.

Process Relationships

When troubleshooting a process, it is crucial to identify all related tasks/processes, and there are several approaches to doing so. A task could hang or dump core because a resource it requires is in use by another process, or a parent could mask a signal that the child needs to execute properly. When it comes to identifying a process's relationship to others, you could use the /proc/<pid>/ directory to manually search out a process's information and its relationship to others. Relationships can also be determined by the use of commands such as ps, pstree, and top, among others, which make use of this pseudo filesystem. These tools make short work of obtaining a picture of a process's state and its relationship to others.

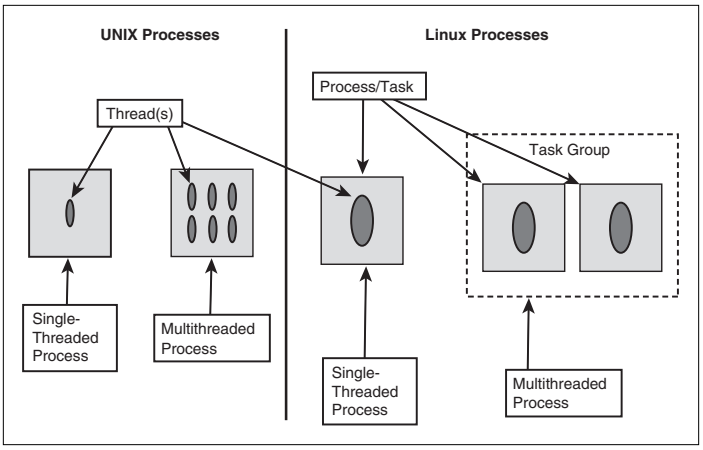


Figure 1: Comparison of UNIX and Linux processes

Linux Process Creation

An understanding of process creation is necessary for troubleshooting a process. Processes are created in Linux in much the same way as they are created in UNIX. When executing a new command, the fork() system call sets up the child's context to reference the parent's context and creates a new stack. This referencing of the parent's context (essentially a pointer to the parent's task_struct() structure) increases overall OS performance. The child's context references the parent's context until modification is required, at which point the parent's address space is copied and modified. This is achieved by the copy-on-write (COW) design.

Shortly after fork() has set up the new task for execution, the exec system call is made. This is where the copy-on-write does its magic. The parent's structure is no longer just referenced; rather, it is copied into a new virtual location. Next, the object file (command) is copied into this location, overwriting the copied pages. Now the new task's context is set up, and the new process is running.

There are some differences between how processes are created in UNIX and how they are created in Linux. For example, some flavors of UNIX perform a copy-on-access, for which the fork() copies the context of the parent to a new virtual memory address with no references pointing back to the parent's context. One is no better than the other because in a majority of instances, the referenced pages must be modified, causing the COW method to copy the pages anyway.

An Example of Linux Process Creation

In this section, we demonstrate the fork() system call by tracing the parent process. In this example, we use the ls command to list a file. Because the ls program is the child of its local shell, we need to trace the shell from which the ll (ls -al alias) command is executed. Two shell windows are required to perform this test.

1. Window one: Determine the pseudo terminal and PID of shell.

```
# echo $$
16935
```

The parent shell's PID is 16935. Now we must start the trace in a second window.

2. Window two: Start trace of shell process.

```
# strace -o /tmp/ll.strace -f -p 16935
```

Now that the trace is running in window two, we need to issue the ll command in window one.

3. Window one: Issue the ll command.

```
# ll test
rw-r--r--    1 chris    chris       46759 Sep  7 21:53 test
```

4. Window two: Here are the results of the stdout and stopping the trace.

```
# strace -o /tmp/ll.strace -f -p 16935
Process 16935 attached <-- Trace running on 16935
Process 17424 attached  <-- forked child process
Process 17424 detached <-- child ending returning to parent
Process 16935 detached <-- ctrl +c ending trace
```

The trace shows the fork() and execve() calls. Note that we are not showing the entire trace because so many system calls take place for each seemingly simple command.

```
...
16935 fork()                            = 17424  <-- NEW task's PID
17424 --- SIGSTOP (Stopped (signal)) @ 0 (0) ---
17424 getpid()                          = 17424
17424 rt_sigprocmask(SIG_SETMASK, [RTMIN], NULL, 8) = 0
17424 rt_sigaction(SIGTSTP, {SIG_DFL}, {SIG_IGN}, 8) = 0
17424 rt_sigaction(SIGTTIN, {SIG_DFL}, {SIG_IGN}, 8) = 0
17424 rt_sigaction(SIGTTOU, {SIG_DFL}, {SIG_IGN}, 8) = 0
17424 setpgid(17424, 17424)             = 0
17424 rt_sigprocmask(SIG_BLOCK, [CHLD TSTP TTIN TTOU], [RTMIN], 8) = 0
17424 ioctl(255, TIOCSPGRP, [17424])    = 0
17424 rt_sigprocmask(SIG_SETMASK, [RTMIN], NULL, 8) = 0
17424 rt_sigaction(SIGINT, {SIG_DFL}, {0x8087030, [], SA_RESTORER, \ 0x4005aca8},
8) = 0
17424 rt_sigaction(SIGQUIT, {SIG_DFL}, {SIG_IGN}, 8) = 0
17424 rt_sigaction(SIGTERM, {SIG_DFL}, {SIG_IGN}, 8) = 0
17424 rt_sigaction(SIGCHLD, {SIG_DFL}, {0x80776a0, [], SA_RESTORER, \ 0x4005aca8},
8) = 0
17424 execve("/bin/ls", ["ls", "-F", "--color=auto", "-l", "test"], \
[/* 56 vars */]) = 0
```

Summary of Process Creation

The fork() call creates a new task and assigns a PID, and this step is soon followed by the execve() call, executing the command along with its arguments. In this case, we see that the ll test command is actually ls -F --color=auto -l test.

Linux Process Termination

An understanding of process termination is useful for troubleshooting a process. As with process creation, the termination or exiting of a process is like that of any other UNIX flavor. If signal handling is implemented, the parent can be notified when its children terminate irregularly. Additionally, the parent process can also wait for the child to exit with some variation of wait(). When a process terminates or calls exit(), it returns its exit code to the caller (parent). At this point, the process is in a zombie or defunct state, waiting for the parent to reap the process. In some cases, the parent has long since died before the child. In these cases, the child has become orphaned, at which point init becomes the parent, and the return codes of the process are passed to init.

Linux Threads

No discussion of process fundamentals is complete without an explanation of Linux threads because an understanding of threads is crucial for troubleshooting processes. As mentioned earlier, the implementation

of threads in Linux differs from that of UNIX because Linux threads are not contained within the proc structure. However, Linux does support multithreaded applications. "Multithreading" just means two or more threads working in parallel with each other while sharing the same address space. Multithreaded applications in Linux just use more than one task. Following this logic in the source, include/linux/sched.h shows that the task_struct structure maintains a one-to-one relationship with the task's thread through the use of a pointer to the thread_info structure, and this structure just points back to the task structure.

Excerpts from the source illustrate the one-to-one relationship between a Linux task and thread.

include/linux/sched.h

```
...
struct task_struct {
    volatile long state;    /* -1 unrunnable, 0 runnable, >0 stopped */
    struct thread_info *thread_info;
...
```

To see the thread_info structure point back to the task, we review include/asm-i386/thread_info.h.

```
...
    struct thread_info {
        struct task_struct      *task;          /* main task structure */
...
```

Using multithreaded processes has its advantages. Threading allows for better processor loading and memory utilization. A drawback is that it also significantly increases the program's complexity. On a single-CPU machine, a multithreaded program for the most part performs no better than a single-threaded program. However, well-designed multithreaded applications executed on a Symmetric Multi-Processor (SMP) machine can have each thread executing in parallel, thereby significantly increasing application performance.

Threaded application performance is enhanced by the fact that threads share resources. Different types of processes share resources in different ways. The initial process is referred to as the heavyweight process (HWP), which is a prerequisite for lightweight processes. Traditionally, a thread of a process is referred to as a lightweight process (LWP), as mentioned earlier. The main difference between these two is how they share their resources. Simply stated, when an HWP forks a new process, the only thing that is shared is the parent's text. If an HWP must share information with another HWP, it uses techniques such as pipes, PF_UNIX (UNIX sockets), signals, or interprocess communication's (IPCS) shared memory, message queues, and semaphores. On the other hand, when an HWP creates an LWP, these processes share the same address space (except the LWP's private stack), thus making utilization of system resources more efficient.

Note that although several forms of threads exists, such as user space GNU Portable Threads (PTH) and DCE threads, in this chapter, we only cover the concept of POSIX threads because they are the most commonly used threads in the industry. POSIX threads are implemented by the pthread library. The use of POSIX threads ensures that programs will be compatible with other distributions, platforms, and OSs that support POSIX threads. These threads are initiated by the pthread_create() system call; however, the Linux kernel uses the clone() call to create the threads. As implied by its name, it clones the task. Just as fork() creates a separate process structure, clone() creates a new task/thread structure by cloning the parent; however, unlike fork(), flags are set that determine what structures are cloned. Only a select few flags of the many flags available are required to make the thread POSIX compliant.

The Linux kernel treats each thread as an individual task that can be displayed with the ps command. At first, this approach might seem like a large waste of system resources, given that a process could have a great number of threads, each of which would be a clone of the parent. However, it's quite trivial because most task structures are kernel objects, which enables the individual threads to just reference the address space. An example is the HWP's file descriptor table. With clone(), all threads just reference the kernel structure by using the flag CLONE_FILES.

With help from developers from around the world, the Linux kernel is developing at an extraordinary rate. A prime example is the fork() call. With the IA-64 Linux kernel, the fork() call actually calls clone2(). In addition, pthread_create() also calls clone2(). The clone2() system call adds a third argument, ustack_size. Otherwise, it is the same as clone(). With the IA-32 2.6 kernel release, the fork() call has been replaced with the clone() call. The kernel clone() call mimics fork() by adjusting clone() flags.

Detailed next are examples of tasks and threads being created on different versions and distributions of Linux:

```
• IA-32 (2.4.19) Fork call
   2970  fork()                        = 3057  <-- The PID for the new HWP
• IA-32 (2.4.19) Thread creation
   3188  clone(child_stack=0x804b8e8, flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_
SIGHAND) =                          3189 <-- LWP
• IA-32 (2.6.3) Fork call
   12383 clone(child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_
SETTID|SIGCHLD,                               child_tidptr=0x4002cba8) = 12499
<-- HWP
• IA-32 (2.6.3) Thread creation
   12440 <... clone resumed> child_stack=0x42184b08, flags=CLONE_VM|CLONE_FS|CLONE_
         FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS
   |CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID|CLONE_DETACHED, parent_
tidptr=0x42184bf8, {en                        try_number:6, base_addr:0x42184bb0,
limit:1048575, s
   eg_32bit:1, contents:0, read_exec_only:0, limit_in_pages:1, seg_not_present:0,
useable:1},                         child_tidptr=0x42184bf8) = 12444  <--LWP
• IA-64 (2.4.21) Fork call
   24195 clone2(child_stack=0, stack_size=0, flags=CLONE_CHILD_CLEARTID|CLONE_
CHILD_                              SETTID|SIGCHLD, child_tidptr=0x200000000002cdc0)
= 24324 <--HWP
• IA-64 (2.4.21) Thread creation
   24359 clone2(child_stack=0x20000000034f4000, stack_size=0x9ff240, flags=CLONE_
VM|CLONE_                           FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_
SYSV
   SEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID|CLONE_DETACHED, par-
ent_tidptr=0x                       2000000003ef3960, tls=0x2000000003ef3f60,
child_tidptr=0x2000000003ef3960) = 24365 <--LWP
```

As the previous examples show, the kernel clone() call creates threads, whereas clone2() creates threads, new processes, or both. In addition, the previous traces reveal the creation of threads and the flags needed to make them POSIX compliant, as defined in the next listing.

```
clone(child_stack=0x804b8e8, flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND)
child_stack:       Unique process stack
CLONE_VM :         Parent and child run in the same address space
CLONE_FS:          Parent and child share file system info
CLONE_FILES:       Parent and child share open file table
CLONE_SIGHAND:     Parent and child share signal handlers
```

Welcome to the Future of Video on the Web!

REGISTER NOW!
www.iTVcon.com

CALL FOR PAPERS NOW OPEN!

LIVE SIMULCAST!
AROUND THE WORLD ON SYS-CON.TV

iTVCON.COM
INTERNET TV CONFERENCE & EXPO 2006

Coming in 2006 to New York City!

"Internet TV is wide open, it is global, and in true 'Web 2.0' spirit it is a direct-to-consumer opportunity!"

For More Information, Call 201-802-3023 or Email itvcon@sys-con.com

Welcome to the Future!

Did you already purchase your ".tv" domain name? You can't afford not to add Internet TV to your Website in 2006!

2005 was the year of streaming video and the birth of **Internet TV**, the long-awaited convergence of television and the Internet. Now that broadband is available to more than 100 million households worldwide, every corporate Website and every media company must now provide video content to remain competitive, not to mention live and interactive video Webinars and on-demand Webcasts.

20 years ago the advent of desktop publishing tools opened the doors for the creation of some of today's well-known traditional print media companies as well as revolutionized corporate print communications. Today, with maturing digital video production, the advent of fully featured PVRs, and significant advances in streaming video technologies, **Internet TV** is here to stay and grow and will be a critical part of every Website and every business in the years to come.

It will also very rapidly become a huge challenge to network and cable television stations: **Internet TV** is about to change forever the $300BN television industry, too.

The Internet killed most of print media (even though many publishers don't realize it yet), Google killed traditional advertising models, and **Internet TV** will revolutionize television the way we watch it today. You need to be part of this change!

Jeremy Geelan
Conference Chair, iTVCon.com
jeremy@sys-con.com

PRODUCED BY SYS-CON EVENTS

List of Topics:

- Advertising Models for Video-on-demand (VOD)
- Internet TV Commercials
- Mastering Adobe Flash Video
- How to Harness Open Media Formats (DVB, etc)
- Multicasting
- Extending Internet TV to Windows CE-based Devices
- Live Polling During Webcasts
- Video Press Releases
- Pay-Per-View
- Screencasting
- Video Search & Search Optimization
- Syndication of Video Assets
- V-Blogs & Videoblogging
- Choosing Your PVR
- Product Placement in Video Content
- UK Perspective: BBC's "Dirac Project"
- Case Study: SuperSun, Hong Kong

Track 1	Corporate marketing, advertising, product and brand managers
Track 2	Software programmers, developers, Website owners and operators
Track 3	Advertising agencies, advertisers and video content producers
Track 4	Print and online content providers, representatives from traditional media companies, print and online magazine and newspaper publishers, network and cable television business managers

Identifying Threads

As previously discussed, the ps command lists all tasks in Linux, preventing the user from distinguishing the HWP from the LWP. At approximately the 2.4.9 kernel release, the Task Group ID (tgid) was added to fs/proc/array.c. This placed a task's tgid in the /proc/<pid>/status file. A key point is that the tgid is equal to the HWP's PID. This new feature enables users to identify threads of a multithreaded process with ease. Reviewing the source, we see:

```
# ./fs/proc/array.c
...
static inline char * task_state(struct task_struct *p, char *buffer)
{
        int g;
        read_lock(&tasklist_lock);
        buffer += sprintf(buffer,
                "State:\t%s\n"
                "Tgid:\t%d\n"
                "Pid:\t%d\n"
                "PPid:\t%d\n"
                "TracerPid:\t%d\n"
                "Uid:\t%d\t%d\t%d\t%d\n"
                "Gid:\t%d\t%d\t%d\t%d\n",
                get_task_state(p), p->tgid,
...
```

Linux commands, such as ps, were modified to make use of this new value, enabling them to display only the parent HWP task (tgid), or all threads of a task by passing the -m or -eLf flag.

In Listing 1, we have included a small example of a threaded program that demonstrates how threads appear in Linux. Note that this code makes no attempt either to lock threads with mutex locks or semaphores or to perform any special signal masking. This code just creates threads that perform sequential counts to exercise the CPU(s).

Listing 1

```
Example of a Threaded Program
#include <pthread.h> /* POSIX threads */
#include <signal.h>
#include <stdlib.h>
#include <linux/unistd.h>
#include <errno.h>

#define num_threads  8

void *print_func(void *);
void threadid(int);
void stop_thread(int sig);
/* gettid() is not portable.. if compiling on other Operating Systems, \ remove
reference to it */
_syscall0(pid_t,gettid)

int main ()
{
        int x;
        pid_t tid;
        pthread_t threadid[num_threads];

        (void) signal(SIGALRM,stop_thread); /*signal handler */

         printf("Main process has PID= %d PPID= %d and TID= %d\n", \ getpid(),
getppid(), gettid());

        /* Now to create pthreads */
        for (x=1; x <= num_threads;++x)
        pthread_create(&threadid[x], NULL, print_func, NULL );

        sleep(60); /* Let the threads warm the cpus up!!! :) */
        for (x=1; x < num_threads;++x)
                pthread_kill(threadid[x], SIGALRM);

        /*wait for termination of threads before main continues*/
        for (x=1; x < num_threads;++x)
        {
        printf("%d\n",x);
        pthread_join(threadid[x], NULL);
         printf("Main() PID %d joined with thread %d\n", getpid(), \ threadid[x]);
        }
}

void *print_func (void *arg)
{
        printf("PID %d  PPID = %d  Thread value of pthread_self = %d and \ TID=
%d\n",getpid(), getppid(), pthread_self(),gettid());
       while(1);  /* nothing but spinning */
}

void stop_thread(int sig) {
pthread_exit(NULL);
}
```

Using Listing 1, create a binary by compiling on any UNIX/Linux system that supports POSIX threads. Reference the following demonstration:

1. Compile the source.

```
# gcc -o thread_test thread_test.c -pthread
```

Next, execute thread_test and observe the tasks with pstree. Note that we have trimmed the output of pstree to save space.

2. Execute the object.

```
#./thread_test
```

3. In a different shell, execute:

```
# pstree -p

init(1)-+-apmd(1177)

~~~~~~Saving space~~~~~

        |-kdeinit(1904)-

~~~~~~Saving space~~~~~

        |             | -kdeinit(2872)-+-bash(2874)---thread_test (3194)-+-
thread_test(3195)
        |             |              |                                    |-
thread_test(3196)
```

```
        |             |             |                                 |-
thread_test(3197)
        |             |             |                                 |-
thread_test(3198)
        |             |             |                                 |-
thread_test(3199)
        |             |             |                                 |-
thread_test(3200)
        |             |             |                                 |-
thread_test(3201)
        |             |             |                                 `-
thread_test(3202)

~~~~~Saving space~~~~~

        |              |              `-bash(3204)---pstree(3250)

~~~~~Saving space~~~~~
```

4. We can display more details with the ps command. (Note that the PIDs would have matched if we had run these examples at the same time.)

```
# ps -eo pid,ppid,state,comm,time,pri,size,wchan | grep test
28807 28275 S thread_test      00:00:12  18 82272 \ schedule_timeout
Display threads with -m.
# ps -emo pid,ppid,state,comm,time,pri,size,wchan | grep test
28807 28275 S thread_test      00:00:00  18 82272 \ schedule_timeout
28808 28807 R thread_test      00:00:03  14 82272 -
28809 28807 R thread_test      00:00:03  14 82272 \ ia64_leave_kernel
28810 28807 R thread_test      00:00:03  14 82272 \ ia64_leave_kernel
28811 28807 R thread_test      00:00:03  14 82272 \ ia64_leave_kernel
28812 28807 R thread_test      00:00:03  14 82272 -
28813 28807 R thread_test      00:00:02  14 82272 \ ia64_leave_kernel
28814 28807 R thread_test      00:00:02  14 82272 \ ia64_leave_kernel
28815 28807 R thread_test      00:00:03  14 82272 -
```

Even though some UNIX distributions have modified commands such as ps or top to display a process with all its threads by including special options such as -m or -L, HPUX has not. Therefore, the HPUX ps command only shows the HWP process and not the underlying threads that build the process. On the other hand, Solaris can display the LWP of a process by using the -L option with its ps command.

Other vendors have created their own tools for displaying threads of a process. HPUX's glance is a good example. Using the same procedures as earlier, we demonstrate multithreads in HPUX to show the main difference between UNIX threads and Linux's implementation of threads.

```
HPUX 11.11:
# cc -o thread_test thread_test.c -lpthread

hpux_11.11 #glance
Process Name   PID   PPID Pri Name   ( 700% max)   CPU   IO Rate   RSS \   Cnt
------------------------------------------------------------------------------
thread_test         14689  14579 233 root     698/ 588   57.3  0.0/ 0.2 \   560kb
9
```

Thus, using HPUX's glance, we can see that the thread count is nine, with one thread representing the main HWP and eight additional threads that were created by the program as shown in the source. Each thread does not have its own PID as with Linux threads. In addition, Linux tools such as top do not show the threads of a process consuming CPU cycles. This can be tested by executing the thread_test program in one tty and the top program in another tty.

Identifying Process Hangs

Now that we have covered the building blocks of processes and threads, it is time to address process hangs and their potential causes. There is little hope that killing an offending process with -9 (sigkill) will lead to discovering the root cause of a process hang. Neither will rebooting the OS unless you are dealing with a stale file handle. Furthermore, these steps will not prevent these anomalies from reoccurring. However, by applying the knowledge of how processes are created and how resources are used, the root cause can be identified.

When a process appears hung, the first step toward a solution is determining certain critical information about the task. Using the ps command, start by determining the task's state, threads, priority, parent, and wait channel. In addition, identify the cumulative CPU time and the initial start time of the task. A holistic approach is needed because although a single ps command is a good start, it will not deliver all the needed information.

Let us first determine whether the process is hung because sometimes a process appears to be blocked when it actually is in the middle of a computation or non-blocking I/O. If cumulative CPU time constantly grows, the task's state will most likely be R. In this state, the process is on the run queue and does not have a wait channel. Monitor its cumulative CPU time. If the process remains in the run queue, it might be performing some calculation that takes a while. Even the fastest computers in the world take a while to calculate an infinite loop! Nevertheless, note that a process in the run state could be the normal operation of that program, an application "feature," or a driver problem.

If an offending process is consuming system resources at an extraordinary rate and starving production applications, killing the offending process is justified if the process can be killed. However, sometimes a process cannot be killed. When a process has exhausted its timeslice, it is put to sleep() with a given priority. When the priority of the process falls below PZERO, it is in an uninterruptible state and cannot be signaled; however, signals can be queued, and for some operations, this is normal. For others, where the program has hung and never returns, the cause is usually located in the driver or hardware. If the process has a state of D (blocked on I/O), it is uninterruptible and cannot be killed. For example, a process accessing a file over a failed hard NFS mount would be in a state of D while attempting to stat() a file or directory.

Uninterruptible processes usually take place when entering I/O calls, at which point the process has called into the kernel, which is in driver code, during which the process cannot receive signals from user space. In this state, a command cannot be signaled even by a SIGKILL (kill -9). It is important to note that signals are queued if not ignored by a sigmask and executed after the code returns from kernel space. Some signals cannot be masked; see the signal man page for more details. Here is an excerpt from the signal man page:

```
...
Using  a  signal  handler function for a signal is called "catching the signal".
The signals SIGKILL and SIGSTOP cannot be caught or ignored.
...
```

A zombie process is another process that a user cannot kill. These processes, however, should not be consuming any CPU cycles or memory resources other than the overhead of having the task structure in the

kernel's Virtual Address Space (VAS). The main goal of troubleshooting a zombie process is determining why the parent died without reaping its children. In short, you should focus on why and how the parent dies. Listed next are process state codes pulled right out of the source code.

```
./linux/fs/proc/array.c
/*
 * The task state array is a strange "bitmap" of
 * reasons to sleep. Thus "running" is zero, and
 * you can test for combinations of others with
 * simple bit tests.
 */
static const char *task_state_array[] = {
        "R (running)",          /*  0 */
        "S (sleeping)",         /*  1 */
        "D (disk sleep)",       /*  2 */
        "Z (zombie)",           /*  4 */
        "T (stopped)",          /*  8 */
        "W (paging)"            /* 16 */
};
...
```

In Scenario 1, we demonstrate an instance in which a process cannot be killed.

Scenario 1: Troubleshooting a Process That Does Not Respond to kill

A user begins rewinding a tape but realizes that the wrong tape is in the drive. The user tries to kill the job but must wait for the process to finish.

Why?

The mt command has made an ioctl call to the SCSI tape driver (st) and must wait for the driver to release the process back to user space so that use signals will be handled.

```
# mt -f /dev/st0 rewind
# ps -emo state,pid,ppid,pri,size,stime,time,comm,wchan | grep mt
D  9225  8916  24 112 20:46 00:00:00 mt             wait_for_completion

[root@atlorca2 root]# kill -9 9225
[root@atlorca2 root]# echo $?   # This produces the return code for the previous
command.  0 = success
0
[root@atlorca2 root]# ps -elf | grep 9225
0 D root     9225 8916  0  24   0    -   112 wait_f 20:46 pts/1    00:00:00 mt -f
/dev/st0
```

The mt command has entered a wait channel, and after the code returns from the driver, the signal will be processed.

Let's check the pending signals:

```
cat ../9225/status
Name:   mt
State:  D (disk sleep)
Tgid:   9225
Pid:    9225
PPid:   8916
TracerPid:      0
Uid:    0       0       0       0
Gid:    0       0       0       0
FDSize: 256
Groups: 0 1 2 3 4 6 10
VmSize:     2800 kB
VmLck:         0 kB
VmRSS:       640 kB
VmData:       96 kB
VmStk:        16 kB
VmExe:        32 kB
VmLib:      2560 kB
SigPnd: 0000000000000100  <-- SigPnd is a bit mask which indicates the value of
the pending signal.  Each byte accounts for 4 bits. In this case, the pending
signal has a value of 9, so the first bit on the 3rd byte is set.  This algorithm
is detailed in linux/fs/proc/array.c under the render_sigset_t() function. The
following table illustrates this function.

Signal    : 1 2 3 4 . 5 6 7 8 . 9 10 11 12 . 13 14 15 16
bit value : 1 2 4 8 . 1 2 4 8 . 1  2  4  8 . 1  2  4  8

kill -3 yields bit mask 0000000000000004
kill -9 yields bit mask 0000000000000100

ShdPnd: 0000000000000100
SigBlk: 0000000000000000
SigIgn: 0000000000000000
SigCgt: 0000000000000000
CapInh: 0000000000000000
CapPrm: 00000000fffffeff
CapEff: 00000000fffffeff
```

Troubleshooting the hung process involves these steps:

1. Identify all the tasks (threads) for the program.
2. Assess the hanging process. Is it easily reproducible?
3. Assess the other things going on. What else is the machine doing?

Check load and other applications' response time.

The following scenario demonstrates a way of troubleshooting a process that periodically hangs and then continues.

Scenario 2: Troubleshooting a Hanging Web Browser

A user complains that her machine is working great except for her Web browsing application. When accessing Web sites, the browser hangs for a few minutes every so often. The user has installed several different versions and tried other browsers to no avail.

You ask her the following question: Has it ever worked? The reply is "Yes . . . several days ago it was fine."

For the sake of simplicity, we attempt to find the problem with a light Web browser with little overhead.

```
# strace -f -F -r -T -o /tmp/lynx.strace_2 lynx http://www.hp.com
```

Using the vi editor and greping for network calls, such as poll(), we can identify what seems to be a problem right away:

```
:g/poll/p
3660       0.000085 poll([{fd=0, events=POLLIN}], 1, 0) = 0 <0.000020>
3660       0.000186 poll([{fd=0, events=POLLIN}], 1, 0) = 0 <0.000008>
3660       0.000049 poll([{fd=3, events=POLLIN}], 1, 5000) = 0 <5.005154>
3660       0.000043 poll([{fd=3, events=POLLIN}], 1, 5000) = 0 <5.009763>
3660       0.000042 poll([{fd=3, events=POLLIN}], 1, 5000) = 0 <5.008875>
3660       0.000043 poll([{fd=3, events=POLLIN}], 1, 5000) = 0 <5.009264>
3660       0.000042 poll([{fd=3, events=POLLIN}], 1, 5000) = 0 <5.009216>
3660        0.000043 poll([{fd=3, events=POLLIN, revents=POLLIN}], 1, \ 5000) = 1
<0.001146>
3660       0.000081 poll([{fd=0, events=POLLIN}], 1, 0) = 0 <0.000017>
3660       0.000088 poll([{fd=0, events=POLLIN}], 1, 0) = 0 <0.000008>
```

ENTERPRISECLASS	**ADVANCED**DESIGN	**OPEN**SOURCE	**CUSTOMER**MANAGEMENT
Robust security. Fully scalable.	Best-of-breed CRM modules.	Source code freely available.	Manage sales pipeline end-to-end.
Java/J2EE environment.	Runs on all standard-based platforms.	Flexible license.	Leads management. Sophisticated
Supports leading SQL databases.	VoIP integration.	Freely extend product capabilities.	Help Desk. Custom surveys and
In use today by Fortune 500.	Customizable workflows.	Integrate with existing systems.	marketing communication.

No Matter the Weather,

The Weather Channel® Is **Open**

The Weather Channel is open for business 24/7. When they decided to make their IT systems open as well, they turned to **Centric CRM**, the most advanced Open Source CRM system available. **Centric CRM** gave The Weather Channel's sales and support teams access to data from their patented technologies housed in 10,000 cable headends across the country. The result? Reduced total cost of ownership (TCO), improved customer view, and the elimination of multiple proprietary applications. Regardless of the conditions outside, **Centric CRM** helps The Weather Channel stay open.

Find out how Centric CRM can help your business get open:
Visit www.centriccrm.com or Call 757.627.8977
Set up a free demo system / Become a Centric CRM Partner

"Centric CRM is a key part of The Weather Channel's move to Open Source software, allowing us to mothball expensive, proprietary applications. Centric CRM is reliable, low cost and easy to use, adding feature rich functionality and flexibility to the way critical data is managed."

Because Great Minds Are Open

CENTRICCRM
Built for the Enterprise...Open to All

```
3660       0.000088 poll([{fd=0, events=POLLIN}], 1, 0) = 0 <0.000022>
```

We see that some poll() calls took over five seconds each. That would explain the Web browser hanging and taking a long time to browse sites.

Focusing on the trace, we see the following:

```
3660     0.000254 socket(PF_INET, SOCK_DGRAM, IPPROTO_IP) = 3 <0.000044>
3660      0.000095 connect(3, {sa_family=AF_INET, sin_port=htons(53), \
sin_addr=inet_addr("15.50.74.40")}, 28) = 0 <0.000017>
3660      0.000108 send(3, "\245`\1\0\0\1\0\0\0\0\0\0\3www\2hp\3com\0\0\
34\0\1", 28, 0) = 28    <0.000404>
3660     0.000476 gettimeofday({1097369839, 928119}, NULL) = 0 <0.000005>
3660     0.000049 poll([{fd=3, events=POLLIN}], 1, 5000) = 0 <5.005154>
3660      5.005262 send(3, "\245`\1\0\0\1\0\0\0\0\0\0\3www\2hp\3com\0\0\
34\0\1", 28, 0) = 28 <0.000426>
```

Checking the man page on poll, we see that it is waiting for an event to take place on a file descriptor.

```
# ls -al /proc/3660/fd/3 -> socket:[52013]
```

This confirms a network issue. After reviewing the /etc/resolv.conf file, we see that 15.50.64.40 is the nameserver.

The user contacted her IT department and found that the nameserver configuration for her network had changed. Switching to a different nameserver in the resolv.conf file alleviated the five-second poll() call and resolved the problem.

Commands commonly used to troubleshoot a hung process include the following:

- ***ps***—Concentrate on pid, ppid, state, comm, time, pri, size, and wchan flags.
- ***lsof***—Determine the open files on the system.
- ***pstree***—Focus on processes and how they relate to each other.
- ***strace***—Flags most commonly used: -f -F -r -T -o <outfile>.
- ***man pages***—Believe it or not, the manual helps.
- ***source code***—Use it to determine what the application is doing.
- ***/proc filesystem***—It offers a wealth of information.

In Scenario 3, we show a process that appears to be hung but is not.

Scenario 3: Troubleshooting an Apparent Process Hang

In this scenario, a user's goal is to create a file that takes data, automatically compresses it, and sends it to a new file. To perform this task, the user creates a named pipe and issues gzip, redirecting input from the pipe to a new file. The odd part is that the gzip process seems to hang, and the user cannot find the gzip process when searching ps -ef. So you devise an action plan: Re-create the event and trace the process involved.

1. Create a named pipe.

```
$ mknod /tmp/named_pipe p
$ ll /tmp/named_pipe
prw-r--r--    1 chris    chris           0 Oct  9 16:53 /tmp/named_pipe|
```

2. Acquire the current process ID.

```
$ echo $$   # note that the current PID = the shell
5032
```

3. From the same shell window, start the gzip process on the named pipe.

```
$ gzip < /tmp/named_pipe > /tmp/pipe.out.gz
```

4. Find the process with a parent of 5032.

```
$ ps -emo pid,ppid,state,comm,time,pri,size,wchan |grep 5032
5236  5032 S bash            00:00:00  30 1040 pipe_wait
```

Notice that the command name is bash, and it is in the sleep state, sleeping on wait chan nel pipe_wait. Yet gzip was the command executed.

5. In another shell window, start a trace on the parent before executing the gzip command.

```
$ strace -o /tmp/pipe.strace -f -F -r -T -v -p 5032
Process 5032 attached - interrupt to quit ........Parent shell \
process
Process 5236 attached ..................................The gzip \
process being forked
Process 5032 suspended
```

As mentioned earlier, fork() essentially creates a process structure by copying the parent. Until execve() executes the binary, the new executable is not loaded into memory, so ps -ef |grep gzip does not show the process. In this case, the gzip process waits for something to be sent to the pipe before executing gzip.

6. A review of the trace explains why the ps -ef | grep gzip command does not show the process.

```
PID        Time              call()
...
5032       0.000079 fork()   = 5236 <0.000252>..........."GZIP was
executed at command line"
5032       0.000678 setpgid(5236, 5236) = 0 <0.000008>
5032       0.000130 rt_sigprocmask(SIG_SETMASK, [RTMIN], NULL, 8) = 0 <0.000007>
...
5032       0.000074 waitpid(-1,  <unfinished ...>..........."man
waitpid: -1 means wait on child"
5236       0.000322 --- SIGSTOP (Stopped (signal)) @ 0 (0) ---
5236       0.000078 getpid() = 5236 <0.000006>
5236       0.000050 rt_sigprocmask(SIG_SETMASK, [RTMIN], NULL, 8) = 0 <0.000007>
5236       0.000067 rt_sigaction(SIGTSTP, {SIG_DFL}, {SIG_IGN}, 8) = 0 <0.000009>
5236       0.000060 rt_sigaction(SIGTTIN, {SIG_DFL}, {SIG_IGN}, 8) = 0 <0.000007>
5236       0.000057 rt_sigaction(SIGTTOU, {SIG_DFL}, {SIG_IGN}, 8) = 0 <0.000007>
5236       0.000055 setpgid(5236, 5236) = 0 <0.000008>
5236       0.000044 rt_sigprocmask(SIG_BLOCK, [CHLD TSTP TTIN TTOU], [RTMIN], 8)
= 0 <0.000007>
5236       0.000071 ioctl(255, TIOCSPGRP, [5236]) = 0 <0.000058>
5236       0.000102 rt_sigprocmask(SIG_SETMASK, [RTMIN], NULL, 8) = 0 <0.000007>
5236       0.000060 rt_sigaction(SIGINT, {SIG_DFL}, {0x8087030, [], SA_RESTORER,
0x4005aca8}, 8) = 0 <0.000007>
5236       0.000075 rt_sigaction(SIGQUIT, {SIG_DFL}, {SIG_IGN}, 8) = 0 <0.000007>
5236       0.000057 rt_sigaction(SIGTERM, {SIG_DFL}, {SIG_IGN}, 8) = 0 <0.000007>
5236       0.000058 rt_sigaction(SIGCHLD, {SIG_DFL}, {0x80776a0, [], SA_RESTORER,
0x4005aca8}, 8) = 0 <0.000007>
5236       0.000262 open("/tmp/named_pipe", O_RDONLY|O_LARGEFILE) = 3 <141.798572>
5236     141.798719 dup2(3, 0) = 0 <0.000008>
5236       0.000051 close(3) = 0 <0.000008>
5236       0.000167 open("/tmp/pipe.out.gz", O_WRONLY|O_CREAT|O_TRUNC|O_LARGEFILE,
0666) = 3 <0.000329>
5236       0.000394 dup2(3, 1) = 1 <0.000007>
5236       0.000042 close(3) = 0 <0.000008>
5236       0.000127 execve("/usr//bin/gzip", ["gzip"]
```

So 141.79 seconds after opening, the named pipe data was received, evecve() executed gzip, and the data was compressed and redirected to the file /tmp/pipe.out.gz. Only at this point would the gzip process show up in the ps listing. So what was initially thought to be a hung process is simply a sleeping process waiting on data.

7. Now ps -ef | grep gzip works.

```
$ ps -ef | grep 5236
```

```
chris     5236  5032  0 17:01 pts/4    00:00:00 gzip
```

Process Cores

Now that we have sufficiently covered structure and hangs as they pertain to Linux processes, let us move on to process core dumps. A core dump enables the user to visually inspect a process's last steps. This section details how cores are created and how to best use them.

Signals

Process core dumps are initiated by the process receiving a signal. Signals are similar to hardware interrupts. As with interrupts, a signal causes a task to branch from its normal execution, handling a routine and returning to the point of interruption. Normal executing threads encounter signals throughout their life cycles. However, there are a finite number of signal types that result in a core dump, whereas other signal types result in process termination.

A process can receive a signal from three sources: the user, the process, or the kernel.

From the User

A user can send a signal in two ways: either using an external command such as kill or within a controlling tty, typing Ctrl+c to send a sigint as defined by stty -a. (Note that by definition, daemons do not have a controlling tty and therefore cannot be signaled in this manner.)

```
# stty -a
speed 9600 baud; rows 41; columns 110; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>; eol2 =
<undef>; start = ^Q; stop = ^S;
```

From the Program

From a program, you can perform the raise() or alarm() system call, allowing a program to signal itself. Consider this example: a ten-second sleep without using the sleep call.

```
main()
{
alarm(10);
pause()
}
```

From the Kernel

The kernel can send a signal, such as SIGSEGV, to a process when it attempts an illegal action, such as accessing memory that it does not own or that is outside of its address range.

Linux supports two types of signals: standard and real-time. A complete overview of signals is outside the scope of this chapter; however, there are a few key differences to note. Standard signals have predefined meanings, whereas real-time signals are defined by the programmer. Additionally, only one standard signal of each type can be queued per process, whereas real-time signals can build up. An example of this was shown earlier in this chapter when a process was blocked on I/O. A kill -9 sigkill was sent to the process and placed in SigPnd.

```
SigPnd: 0000000000000100  <– A signal waiting to be processed, in this case
sigkill
```

In troubleshooting a process, a user might want to force a process to dump core. As stated, this is accomplished by sending the appropriate signal to the process. Sometimes after this step is taken, the dump does not follow because the process has not returned from an interrupt due to some other issue. The result is a pending signal that needs to be processed. Because the signals that result in a core are standard signals, sending the same signal multiple times does not work because subsequent signals are ignored until the pending signal has been processed. The pending signals are processed after the program returns from the interrupt but before proceeding to user space. This fact is illustrated in the entry.S source file, as shown in the following:

arch/i386/kernel/entry.S

```
...
ret_from_intr()
...
_reschedule()
...
_signal_return()
...
        jsr     do_signal       ; arch/cris/kernel/signal.c
...
```

It is also possible to have difficulty achieving the dump because signals are being blocked (masked), caught, or ignored. An application might have signal handlers that catch the signal and perform their own actions. Signal blocking prevents the delivery of the signal to the process. Ignoring a signal just means that the process throws it away upon delivery. Additionally, the signal structure of a process is like any other structure in that the child inherits the parent's configuration. That being stated, if a signal is blocked for the parent, the child of that process has the same signals blocked or masked. However, some signals cannot be masked or ignored, as detailed in the man page on signal. Two such signals are sigkill and sigstop.

The user can obtain a list of signals from the kill command. This yields a list of signals that the user can send to a process. Possible signals include the following (note that this is not a complete list):

```
$ kill -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL
 5) SIGTRAP      6) SIGABRT      7) SIGBUS       8) SIGFPE
 9) SIGKILL     10) SIGUSR1     11) SIGSEGV     12) SIGUSR2
13) SIGPIPE     14) SIGALRM     15) SIGTERM     17) SIGCHLD
...
```

As mentioned earlier and illustrated next, the man page on signal details the signals that produce a core file.

$ man 7 signal

```
...
Signal     Value    Action    Comment
----------------------------------------------------------------------
       SIGHUP       1      Term    Hangup detected on controlling terminal
                                   or death of controlling process
       SIGINT       2      Term    Interrupt from keyboard
       SIGQUIT      3      Core    Quit from keyboard
       SIGILL       4      Core    Illegal Instruction
       SIGABRT      6      Core    Abort signal from abort(3)
       SIGFPE       8      Core    Floating point exception
...
The source code on signal also provides this list as illustrated next:
linux/kernel/signal.c
...
#define SIG_KERNEL_COREDUMP_MASK (\
        M(SIGQUIT)   |  M(SIGILL)    |  M(SIGTRAP)   |  M(SIGABRT)   | \
        M(SIGFPE)    |  M(SIGSEGV)   |  M(SIGBUS)    |  M(SIGSYS)    | \
        M(SIGXCPU)   |  M(SIGXFSZ)   |  M_SIGEMT                     )
```

```
...
```

Limits

By default, most Linux distributions disable the creation of process core dumps; however, the user can enable this capability. The capability to create or not create core dumps is accomplished by the use of resource limits and the setting of a core file size. Users can display and modify their resource limits by using the ulimit command.
In this listing, we depict core dumps being disabled by displaying the user soft limits:

```
$ ulimit -a
core file size        (blocks, -c) 0  <– COREs have been disabled
data seg size          (kbytes, -d) unlimited
file size             (blocks, -f) unlimited
max locked memory      (kbytes, -l) unlimited
max memory size        (kbytes, -m) unlimited
open files                    (-n) 1024
pipe size          (512 bytes, -p) 8
stack size             (kbytes, -s) 8192
cpu time              (seconds, -t) unlimited
max user processes             (-u) 4095
virtual memory         (kbytes, -v) unlimited
```

There are two limits for each resource: a soft limit (shown previously) and a hard limit. The two limits differ in how they can be modified. The hard limit can be thought of as a ceiling that defines the maximum value of a soft limit. Users can change their hard limit only once, whereas they can change their soft limits to any values at any time as long as they do not exceed the hard limit.

Rerunning the ulimit command with the -Ha option as shown below, we see the hard limits for each resource.

```
$ ulimit -Ha
core file size        (blocks, -c) unlimited
data seg size          (kbytes, -d) unlimited
file size             (blocks, -f) unlimited
max locked memory      (kbytes, -l) unlimited
max memory size        (kbytes, -m) unlimited
open files                    (-n) 1024
pipe size          (512 bytes, -p) 8
stack size             (kbytes, -s) unlimited
cpu time              (seconds, -t) unlimited
max user processes             (-u) 4095
virtual memory         (kbytes, -v) unlimited
```

A user can set a hard or soft limit to unlimited, as in the previous example. unlimited just means that the process does not have an artificial limit imposed by setrlimit. However, the kernel must represent "unlimited" with a value so that it has a manageable range. The program is limited by what the kernel can address or the physical limits of the machine, whichever comes first. Thus, even when set to unlimited, a limit exists. The 32-bit representation of unlimited (denoted "infinity") is defined in sys_ia32.c as indicated next:

```
...
#define RLIM_INFINITY32 0xffffffff   <--  Equals  4294967295 bytes ~ 4Gig
#define RESOURCE32(x) ((x > RLIM_INFINITY32) ? RLIM_INFINITY32 : x)

struct rlimit32 {
        unsigned        rlim_cur;     <-- soft limit
        unsigned        rlim_max;     <-- hard limit
};
...
```

Anytime a process dumps core and the resource limit core file size is anything other than zero, the kernel writes the core image. There are times, however, when user limits are set to low, resulting in a corrupt or unusable core image. If the core file resource limit is not adequate to accommodate the process's core image, the kernel either does not produce a dump, truncates the dump, or attempts to save only the stack portion of the process's context.

What occurs if the kernel is unable to create the dump depends on the type of executing process. Linux supports a multitude of executable formats. Originally, the a.out binary was used, which contains a magic number in its header. Traditionally, this magic number was used to characterize the binary type—for example, exec magic, demand magic, shared_mem magic, and so on. However, it was decided early on that the Executable and Linking Format (ELF) would be Linux's default binary format because of its flexibility. Although AT&T defined the original ELF-32 binary format, UNIX System Laboratories performed the original development of this format. Later HP and INTEL defined the ELF-64 binary format. Today's Linux systems contain very few, if any, a.out binaries, and support has been removed from the main kernel and placed into a module called binfmt_aout.o, which must be loaded before executing one of these binaries.

Referencing the binfmt source for each format details what action is taken in the event of a process attempting to produce a core file, as illustrated next.

The following snippet is from fs/binfmt_aout.c.

```
...
/* If the size of the dump file exceeds the rlimit, then see what would happen
   if we wrote the stack, but not the data area.  */
...
The next snippet is from fs/binfmt_elf.c.
...
/*
 * Actual dumper
 *
 * This is a two-pass process; first we find the offsets of the bits,
 * and then they are actually written out.  If we run out of core limit
 * we just truncate.
 */
```

The Core File

After the core file is generated, we can use it to determine the reason for the core dump. First, we must identify the process that created the core and the signal that caused the process to die. The most common way of determining this information is through the file command. Next, we determine whether the program in question has had its symbols stripped. This information can be determined by executing the file command against the binary. As mentioned earlier, the core file is the process's context, which includes the magic number or type of executable that created the core file. The file command uses a data file to keep track of file types, which by default is located in /etc/magic.

In Scenario 4, we show an example of a program with an easily reproducible hang. We can use tools such as gdb and other GNU debuggers/wrappers such as gstack to solve the problem.

Scenario 4: Using GDB to Evaluate a Process That Hangs

We use gdb to evaluate a core file created when a program was termi-

nated because it hangs.

```
$ ll gmoo*
-rwxr-xr-x    1 chris    chris     310460 Jan  2 20:25 gmoo.stripped*
-rwxr-xr-x    1 chris    chris     321486 Jan  2 22:25 gmoo.not.stripped*
```

The file command informs us of the type of executable (defined in /etc/magic). In the previous example, we have one binary that hangs when executing.

It is helpful to determine the type of binary, as in the following example:

```
$ file gmoo.*
gmoo.not.stripped: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for
GNU/Linux 2.2.5, dynamically linked (uses shared libs), not stripped
gmoo.stripped:      ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for
GNU/Linux 2.2.5, dynamically linked (uses shared libs), stripped
```

When the command is hung, we send a kill -11 (SIGSEGV) to the program, causing the program to exit and dump core. An example of such a resulting core file follows:

```
$ file core.6753
core.6753: ELF 32-bit LSB core file Intel 80386, version 1 (SYSV), SVR4-style,
SVR4-style, from 'gmoo.stripped'
Using the GNU Project Debugger (GDB), we get the following:
$ gdb -q ./gmoo.striped ./core.6753
...
Core was generated by './gmoo.striped'.
Program terminated with signal 11, Segmentation fault.
Reading symbols from /usr/lib/libgtk-1.2.so.0...(no debugging symbols found)...
done.
Loaded symbols for /usr/lib/libgtk-1.2.so.0
Reading symbols from /usr/lib/libgdk-1.2.so.0...(no debugging symbols found)...
done.
Loaded symbols for /usr/lib/libgdk-1.2.so.0
Reading symbols from /usr/lib/libgmodule-1.2.so.0...(no debugging symbols
found)...done.
Loaded symbols for /usr/lib/libgmodule-1.2.so.0
Reading symbols from /usr/lib/libglib-1.2.so.0...(no debugging symbols found)...
done.
...
(gdb) backtrace
#0  0x4046b8e6 in connect () from /lib/i686/libpthread.so.0
#1  0x0806bef1 in gm_net_connect ()
#2  0x080853e1 in gm_world_connect ()
#3  0x0806c7cf in gm_notebook_try_add_world ()
#4  0x0806cd8c in gm_notebook_try_restore_status ()
#5  0x08061eab in main ()
#6  0x404c5c57 in __libc_start_main () from /lib/i686/libc.so.6
(gdb) list
No symbol table is loaded.  Use the "file" command.
(gdb)
```

Without the source, we have gone about as far as we can. We must use other tools, such as strace, in combination with gdb. Other tool suites such as valgrind can also prove useful.

Now, let us look at an example of the same hang with a non-stripped version of the binary.

```
$ gdb -q ./gmoo.not.stripped ./core.6881
...
Core was generated by './gmoo.not.stripped'.
Program terminated with signal 11, Segmentation fault.
Reading symbols from /usr/lib/libgtk-1.2.so.0...done.
Loaded symbols for /usr/lib/libgtk-1.2.so.0
Reading symbols from /usr/lib/libgdk-1.2.so.0...done.
Loaded symbols for /usr/lib/libgdk-1.2.so.0
Reading symbols from /usr/lib/libgmodule-1.2.so.0...done.
...
(gdb) backtrace
#0  0x40582516 in poll () from /lib/i686/libc.so.6
(gdb)
```

Although the stack trace appears to be different, we have identified the root cause. The program is hung on a network poll call, which, according to the man page, is a structure made up of file descriptors. Using other tools, such as lsof, strace, and so on, we can determine exactly the network IP address upon which the process is hung.

Summary

As is apparent from the discussion in this chapter, the topic of process structure, hangs, and core files is a complex one. It is crucial to understand the process structure to troubleshoot hangs and most efficiently use core files. New troubleshooting tools are always being developed, so it is important to keep up with changes in this area. Although troubleshooting process hangs can be intimidating, as you can conclude from this chapter, it simply requires a step-by-step, methodical approach that, when mastered, leads to efficient and effective resolution practices.

Aha! What Your Refrigerator Can Teach You About Selling Software

Delivering your software as an appliance

by Tom Crowley

Great ideas often come from the most commonplace occurrences. Take the story of Sir Issac Newton's "discovery" of gravity. According to the legend, he was sitting under an apple tree minding his own business when – bonk! – an apple dropped onto his head, and suddenly all the complex mathematical formulae he'd been considering became crystal clear. It's what's known as an "Aha!" moment.

There's a similar "Aha!" moment waiting for you right in your office kitchen, holding the energy drinks you suck down in mass quantities to keep you going through long, late-night hours of intensive programming. Yes, it's the simple refrigerator. Whether a little college dorm cube-type or a full-on Sub Zero stainless steel unit with a port that hooks into your PC, the refrigerator has but one mission: keep things cold.

The nice thing is you didn't have to buy the outer metal casing and then figure out how to make it keep things cold rather than cook food, grind garbage, or make coffee. You didn't have to install an operating system, and then download instructions that tell it how to maintain a temperature of 47 degrees to keep your energy drinks just the way you like them. And you didn't have to worry about it causing conflicts with the toaster oven. All you had to do was pull it out of the shipping carton, put it in place, plug it in, set the temperature dial, and load it up with goodies.

Here's the "Aha!" moment for you. Rather than delivering your Java-based software application as a box full of CDs or an electronic software download (ESD) that needs to be installed, configured, and tweaked, you can use that same appliance model to provide a simpler, more complete package for your application.

Not only does providing software as part of an appliance make installation and use easier for the customer, it also simplifies product development and makes after-sale services such as support a whole lot easier on your end. If you're using open source software as part of the application, delivering the final product as an appliance even helps get it approved by enterprise gatekeepers who have a bias against open source. It may not be the discovery of gravity, but it certainly has a big impact on your business.

What Is an Appliance?

In the technology sense, an appliance is a device that delivers the software you've created, the hardware it runs on, and the operating system that connects the two, all in a single, hardened package. When it arrives, the customer unpacks it, places it in the rack (after checking power and cooling requirements), connects the power, connects the appliance to their network, and presses, flips, or toggles the "on" button. At that point the appliance fires up, configures itself, and is ready to get to work. All of this occurs in less time than it normally takes to load the software from CDs or a downloaded executable alone.

Traditionally appliances have tended to be used more to deliver smaller, single-focused, back office applications, such as DNS management, firewall services, or video streaming. Now, though, they are also being looked at for more complex front-end applications, such as customer relationship management (CRM) and enterprise resource planning (ERP). And for good reason.

There are a number of advantages to the appliance model – some more evident than others. These advantages tend to fall into two categories: experiential and technical. Both are important, as they have a direct impact on the total customer experience. Let's look at each category to see how they affect both immediate customer satisfaction and future opportunities.

The Appliance Experience

One of the most significant impressions an appliance makes right out of the box, so to speak, is the perceived value of delivering a complete, ready-to-go product rather than a set

About the Author

Tom Crowley is president and CEO of MBX Systems (www.mbx.com), a design and manufacturing organization that helps independent software vendors take their product offerings from application to appliance. MBX offers turnkey services for standard, semi-custom, and fully custom configurations in sizes from 1U to 4U, including hardware and graphical design, technical consultation, assembly, and tech support.

tom@mbx.com

of CDs or downloads that are just the starting point. Software is somewhat ethereal, even to those who work with it every day, and Java can be intimidating to the non-expert. But a metal box you can hold in your hands that makes a loud noise if dropped – now that has substance!

An appliance also provides an opportunity for branding and visibility within the technical organization that often isn't there with software alone. Consider that with many software packages, especially for back-end functions, such as spam filtering or security applications, if all goes well it just runs in the background. Out of sight, out of mind. The only time anyone sees your company's name is when something goes wrong, which is not exactly when you want them to be thinking of you. With an appliance, though, your brand identity is very visible right in the rack. Network administrators or other IT types pass by it on their way to solve some other crisis. Or when they go for their own energy drinks. And the more appliances you sell to that customer, the more visible you are to them, which keeps you at the top of the mind for future sales.

Delivering software as an appliance solves logistics problems for the customer as well. They no longer have to make room on an existing server, or even worse have to try to coordinate software and hardware delivery from two separate sources, using two separate purchase orders. One call, fax, or e-mail does it all, with a guarantee that everything arrives at the same time.

Finally, if a problem does arise with your application, there is none of the usual "it's a hardware problem"/ "No it's a software problem"/ "I knew we shouldn't have used open source products"/ "Is there anyone else we can blame?" fingerpointing. Customers have one call to make to resolve any issues, no matter what the cause. That's a benefit that can't be overestimated, especially at the enterprise level.

Technically Speaking

Because they are hardened system – i.e., no outside software or hardware is required to run them – appliances offer certain technical advantages over software users install themselves.

An obvious one is that they are far easier to deploy. Normally, the most difficult part of deploying an appliance is pulling it out of the shipping carton. A few connections, a little self-configuration, maybe a couple of manual steps, and the customer is up and running. Depending on the nature and complexity of the software, users may need to perform certain operations from a master console. But even then, they can get to that part a lot faster if they're not first trying to load it onto a separate server or seek out additional components on a Web site.

On the development end, one of the most serious complexities is trying to anticipate the hardware platform and operating system the customer will be using. Much of the development time, in fact, is actually spent testing the software with various common configurations. Of course it never fails that some important customer has created some proprietary nightmare, and then your development team becomes your tech support for that important customer instead.

The appliance model removes the need to create versions to run on Windows, Unix, Linux, or BSD, because it has its own built-in operating system, often (but not always) based on a Linux kernel. That arrangement eliminates hardware-software conflicts. If a problem does arise, usually the outsourced appliance manufacturer or an outsourced technical group can take over the unit via the Internet and get it running properly. If the problem is mechanical, you can even ship a new, working appliance to take the place of the old one, solving the problem as quickly as the delivery method will allow, while saving you the trouble of sending a support team to the customer's location.

Easy set up also makes for more successful trial programs. You can ship the appliance to the customer, and they can put it in place to get a trial under way immediately rather than having to carve out hours to configure demo software. As an added benefit, once it's in the rack customers historically have been far more likely to purchase the product than they are a software-only demo.

Updates become easier too. In the traditional model, and especially in these security-conscious times, it's usually up to the customer to download and install any product upgrades. (See Upgrade Tuesday, subsection you-know-who.) That's taking an awful big chance. Because while you may have diligently fixed any bugs or improved on product performance, the ultimate application of those upgrades is still in the hands of someone over whom you have no control.

In the appliance model, you can push upgrades out directly to the appliance itself with no intervention from the customer. As a result, they're always running the latest, greatest, most secure version, and they don't have to tie up any resources to do it.

Once the appliance is in place, it eliminates much of the need for training since many of the operations happen automatically. You'll still need to support higher-level functionality, such as dashboard set up or data presentation. But you can eliminate the entire step of "here's how it operates."

When you lay out these technical advantages, you gain another one. Being able to deliver an appliance helps shorten the sales cycle because there just aren't as many obstacles to making the purchase.

A Universal Solution – Almost

Those are all the reasons you may want to deliver your software as an appliance. Yet, while the appliance model is a good one for nearly any type of software, there are a few exceptions, as usual.

Perhaps the most important reason not to go the appliance route is if the price point of the software won't support it. Bundling the software with its own hardware does add to the cost. If your software would normally sell for less than $100, it's probably not worth going the appliance route unless there's a huge marketplace expectation for it – and the customers are willing to pay for the convenience.

Likewise, if your software is designed for the individual consumer or the SOHO market, an appliance model may not fit. While there are some adventurous souls who have extensive home networks, or who have converted a linen closet into a wiring closet, they are definitely the minority. A rack-mounted appliance that requires a more sophisticated network just doesn't make sense in an environment where the "network" consists of a wireless router the business owner picked up at the local consumer electronics chain. Then he or she had to spend two hours on the phone with Dave the Late Night Customer Service Guy to figure out how to get it to work.

It's also important to know your customers. If all the conditions are right but the industries you're targeting for sales are appliance-averse, don't try to force it in. Educate them, inform them, and look to the future. Maybe they'll have their own "Aha!" moment someday. But make sure you do what's right to get the sale today so there is a tomorrow to work toward.

Is Your Refrigerator Running?

As a software delivery method, appliances offer many advantages over traditional CDs and ESD downloads. They are also effective marketing tools, often allowing you to differentiate your offering and build brand recognition in a very unique and visible way.

Don't wait until an apple drops on your head to have your moment of revelation. Take the initiative, run the numbers, check out the market, and discover whether your next (or first) version would be better delivered as an appliance. It just might keep you in energy drinks for a long time to come.

Rich Internet Applications: AJAX,

www.AjaxWorldExpo.com

SANTA CLARA SILICON VALLEY

SYS-CON Events is proud to announce the first-ever AjaxWorld Conference & Expo 2006!

The world-beating Conference program will provide developers and IT managers alike with comprehensive information and insight into the biggest paradigm shift in website design, development, and deployment since the invention of the World Wide Web itself a decade ago.

The terms on everyone's lips this year include "AJAX," "Web 2.0" and "Rich Internet Applications." All of these themes play an integral role at AjaxWorld. So, anyone involved with business-critical web applications that recognize the importance of the user experience needs to attend this uniquely timely conference – especially the web designers and developers building those experiences, and those who manage them.

CALL FOR PAPERS NOW OPEN!

We are interested in receiving original speaking proposals for this event from i-Technology professionals. Speakers will be chosen from the co-existing worlds of both commercial software and open source. Delegates will be interested learn about a wide range of RIA topics that can help them achieve business value.

Flash, Web 2.0 and Beyond...

REGISTER TODAY AND $AVE!

"It Was The Best AJAX Education Opportunity Anywhere in the World!" —John Hamilton

October 3-4, 2006

Santa Clara Convention Center
Hyatt Regency Silicon Valley
Santa Clara, CA

To Register
Call 201-802-3020 or
Visit www.AjaxWorldExpo.com

May 7-8, 2007

First International AjaxWorld Europe
Amsterdam, Netherlands

"Over the two information-packed days, delegates will receive four days' worth of education!"

Topics Include...

Themes:
- > *Improving Web-based Customer*
- > *Interaction*
- > *AJAX for the Enterprise*
- > *RIA Best Practices*
- > *Web 2.0 – Why Does It Matter?*
- > *Emerging Standards*
- > *Open Source RIA Libraries*
- > *Leveraging Streaming Video*

Technologies:
- > *AJAX*
- > *The Flash Platform*
- > *The Flex 2 Framework & Flex Builder 2*
- > *Microsoft's approaches: ASP.NET, Atlas, XAML with Avalon*
- > *JackBe, openLaszlo*
- > *JavaServer Faces and AJAX*
- > *Nexaweb*
- > *TIBCO General Interface*

Verticals:
- > *Education*
- > *Transport*
- > *Retail*
- > *Entertainment*
- > *Financial Sector*
- > *Homeland Security*

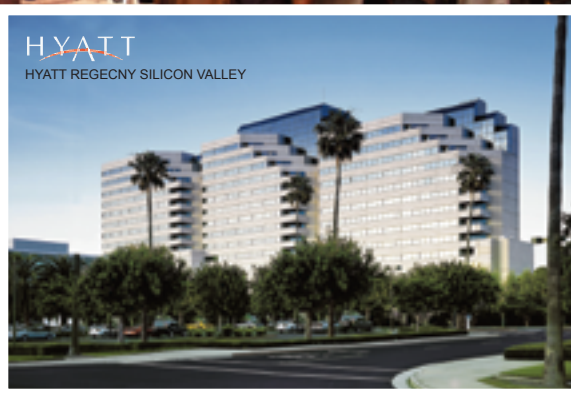
GROUP DISCOUNTS AVAILABLE:
5 Delegates from Same Company for only $995 (each)
Register a Second Person for only $1195

LIVE SIMULCAST!
AROUND THE WORLD ON SYS-CON.TV

Receive FREE WebCast Archives of Entire Conference!

The best news for this year's conference delegates is that your "Golden Pass" registration now gives you full access to all conference sessions. We will mail you the complete content from all the conference sessions in seven convenient DVDs after the live event takes place.

► **This on-demand archives set is sold separately for $995**



Early Bird*
(Register Before August 31, 2006) **$1,495****
See website or call for group discounts

Special Discounts*
(Register a Second Person) **$1,395****
See website or call for group discounts

(5 Delegates from same Company) **$1,295/ea.****
See website or call for group discounts

On-Demand Online Access
(Any Event) **$695**

*Golden Pass access includes Breakfast, Lunch and Coffee Breaks, Conference T-Shirt, Collectible Lap-Top Bag and Complete On-Demand Archives of sessions in 7 DVDs!

**OFFER SUBJECT TO CHANGE WITHOUT NOTICE, PLEASE SEE WEBSITE FOR UP-TO-DATE PRICING

SYS-CON EVENTS

For more great events visit www.EVENTS.SYS-CON.com

VISIT WWW.AJAXWORLDEXPO.COM FOR THE MOST COMPLETE UP-TO-DATE INFORMATION

Vyatta and Sangoma Partner to Drive Adoption of Open-Source Linux Routing

(*Toronto, Ontario / San Mateo, CA*) – Vyatta and Sangoma Technologies have partnered to advance the market for network routers and firewalls based on open source Linux technologies. Vyatta's Open Flexible Router (OFR) will provide the protocol software and Sangoma's WANPIPE PCI cards will supply cost-effective WAN connections.

By combining these products, customers in small and mid-sized businesses can build a WAN router using a standard PC platform and save at least 50% over traditional closed-source products. The companies have agreed to work together to ensure the two product lines are technically compatible and provide ease-of-use and deployment for its customers. Vyatta and Sangoma plan to collaborate on marketing activities that promote growth of the open source router and firewall applications among user communities.

Vyatta is developing the Open Flexible Router (OFR), a flexible, cost-effective, and secure alternative to the incumbent, closed-source solutions that dominate the market today.

They recently launched a community designed to bring together users and developers to shape the product and advance the cause of open source networking. A beta version of the Vyatta OFR is available for download. To join the Vyatta community and participate in building the open, flexible router, go to www.vyatta.com.

www.sangoma.com

Linux Networx Accelerators Expected to Drive Up to 4x Price/Performance of Existing Accelerator Solutions

(*Salt Lake City*) – Linux Networx, The Linux Supercomputing Company, is applying its supercomputing experiencee to the delivery of a series of application acceleration solutions expected to deliver up to 4x the price/performance value of current application accelerators for key applications. Featuring powerful accelerators, tight integration with the compute system, and partitioning of overhead tasks, the Linux Networx acceleration solutions can solve larger problems than traditional accelerators.

Multi-paradigm computing using application accelerators can achieve increases in computational throughput by shifting the execution of selected algorithms from a compute system's general-purpose CPU to one or more highly specialized accelerators. However, the performance of current accelerators is limited by high levels of data communications overhead and insufficient accelerator resources, squandering potential acceleration capacity. Linux Networx solves these problems by using powerful accelerators and ultra-high bandwidth, low latency interface to the host system.

Linux Networx is also initiating a developer program to optimize the performance of the accelerators for targeted applications and industries, including seismic analysis, computational fluid dynamics, and high energy physics The first commercial release of Linux Networx' new accelerators will be in the third quarter of 2006.

www.linuxnetworx.com

IBM Contributes Ajax Software Development Technology to Open Source Community

(*Armonk, NY*) – IBM has announced plans to contribute key intellectual assets to the open source community in an effort to help companies and software developers adopt and share best practices for Ajax software development.

The software contributions will foster an industry collaboration and adoption of Ajax, a technology that improves Web application responsiveness, and does so at a lower cost point and with less complexity. The technology contributions will extend the code already available in the Dojo Toolkit enabling internationalization of applications and making them fully accessible to persons with disabilities through a variety of assistive technologies, including DHTML and Accessible widgets. IBM's donation will also extend the data model already in the Dojo Toolkit and provide a foundation architecture and Web-based tools for the industry to engineer, collaborate on, share and reuse software development best practices.

Ajax – short for Asynchronous JavaScript+XML – is a type of "Rich Internet Application." The technology enables greater usability and responsiveness of browser-based applications, with the benefits of a zero-install and extremely low maintenance of Web development. Developers have the ability to update portions of a Web page without refreshing the entire screen. This can enable users to scroll through a virtual map or photographs without refreshing their browser screens. In addition, developers can allow users to drag and drop objects inside a Web browser similar to the capabilities available on computer desktops to move folders and documents around. Ajax provides the ability to develop rich buttons sets, icons, scroll bars, menus and widgets that can ease and speed navigation, as well as simplify information retrieval.

www.ibm.com

Free Version of SourceForge Enterprise Edition Available

(*Fremont, CA*) – VA Software Corporation has announced availability of a free, full-feature version of its collaborative software development platform, SourceForge Enterprise Edition (SFEE).

The download, which includes a perpetual, no-cost license for 15 users, is available at www.vasoftware.com.

Built on the Java 2 Enterprise Edition J2EE platform, SFEE provides developers with the features, scalability, reliability, and robustness they expect in an enterprise-class product. SFEE is optimized for the enterprise environment with features such as LDAP and Wiki integration, workflow tracking capabilities, role-based access controls and support for multiple Software Configuration Management (SCM) tools. With SFEE, development teams of any size can deliver products more rapidly. This download lets development teams experience the benefits of the collaborative capabilities of SFEE.

The Free Standards Group Expands Membership and Activities in Korea

(*Seoul*) – The Free Standards Group (FSG), a nonprofit organization dedicated to developing and promoting open source software standards, has announced that Korean IT governmental organizations – the Korea IT Industry Promotion Agency (KIPA) and the Electronics and Telecommunications Research Institute (ETRI) and Korea's largest software provider Haansoft – have joined the organization as silver sponsors and will participate in the FSG's Linux Standard Base workgroup. The FSG is also announcing the opening of an outreach and development office in Korea, one of the fastest growing Linux markets in the world.

KIPA is a non-profit organization established in 1998 by the Korean Ministry of Information and Communication to further the exchange and cooperation between Korean and foreign IT enterprises. KIPA will work with the FSG by serving as a gateway between the FSG and the Korean IT industry and will promote the use of FSG's open standards in Korea.

ETRI is a non-profit Korean government-funded research organization established in 1976. It has developed UNIX technologies for more than 20 years and began developing Linux technologies for Korea seven years ago. It plans to devote resources to the Linux Standard Base workgroup and use the LSB in its development efforts.

Haansoft is a Korean developer of software and is a member of Asianux, a Linux server operating system co-developed by Chinese Leading Linux vendor Red Flag Software Co., Ltd., Japanese Linux vendor Miracle Linux Corporation and Haansoft. Asianux aims to be the standard enterprise Linux platform for Asia and is currently certified to the Linux Standard Base.

Symantec to Support Linux on POWER with High Availability and Backup Solutions

(*Cupertino, CA*) – Symantec has announced an agreement with IBM that provides for the delivery of Symantec high availability, storage management, and backup products for the Linux on POWER platform by the end of 2006.

These solutions will help clients consolidate Linux applications on the IBM System p platform, leveraging the benefits of IBM's System p hardware platform and Symantec's data center infrastructure software.

The announcement is another example of Symantec's commitment to support all major UNIX, Linux, and Windows platforms in the data center, enabling customers to standardize on a consistent software infrastructure across heterogeneous environments.

The deal extends the strong engineering relationship between the two companies that began in April 2000. In addition, IBM will resell Symantec's Veritas Cluster Server and the Veritas Storage Foundation family of products with IBM System p servers for AIX and Linux on POWER in the U.S. Federal and Japan markets.

The addition of Symantec's high availability and storage management software, including Veritas Cluster Server, Veritas Storage Foundation, Veritas Storage Foundation for Oracle RAC, and Veritas Volume Replicator to the IBM resell portfolio of System p software enhances IBM's ability to offer clients integrated solutions in the key market segments of U.S. Federal and Japan. The agreement also includes a commitment by Symantec to port these products and Veritas NetBackup Enterprise backup and recovery technology to the Linux on POWER platform.
www.symantec.com

Xandros Teams with Ingram Micro Canada to Deliver End-to-End Linux Solutions to Major Canadian Resellers

(*New York, NY / Ottawa, ON*) – Xandros, a provider of Linux alternatives to Windows desktop and server products, has announced that it has teamed up with the Canadian operations of Ingram Micro Inc., a technology distributor, to deliver the Xandros line of Linux desktop and server solutions to major Canadian resellers.

This move follows the recent release of Xandros Server, an SMB alternative to Windows Server 2003, which, together with the Xandros Desktop, delivers an end-to-end Linux solution to the retail channel.

Xandros provides a full line of desktop operating systems that work right out of the box and offer unrivaled compatibility with Microsoft Windows. Xandros Desktop OS assures rock-solid stability and security, along with the freedom that an open environment provides.
www.xandros.com.
www.ingrammicro.com

Virtual Iron Joins the Distributed Management Task Force (DMTF)

(*Lowell, MA*) – Virtual Iron Software, a provider of software solutions for creating and managing virtual infrastructure in the data center, has announced that it has joined the Distributed Management Task Force (DMTF) to help lead the development of management standards and promote interoperability in the enterprise data center.

Virtual Iron delivers virtualization and management solutions that leverage industry standards and the Xen open source hypervisor. The software is designed for production-class performance and scalability and it supports hundreds of industry-standard (x86) physical servers and thousands of virtual servers. Virtual Iron includes comprehensive capabilities to manage virtual infrastructure and address a number of data center and virtualization initiatives. The software partitions servers to run multiple operating systems simultaneously and increases utilization by managing the pooling and sharing of all server, storage and network elements. This enables users to automate many manual tasks such as provisioning new virtual servers and moving capacity to handle dynamic workloads. It also monitors physical systems to provide high availability for virtual servers running on them.
www.virtualiron.com

Mercury Computer Systems Selects Wind River Carrier Grade Linux for Ensemble2 AdvancedTCA-Based Application Platform

(*Alameda, CA*) – Wind River Systems, Inc., a provider of Device Software Optimization (DSO), has announced that Mercury Computer Systems has selected the Wind River Platform for Network Equipment, Linux Edition (PNE-LE) for its Ensemble2 AdvancedTCA-based application platform. Mercury's Ensemble2 is a standards-based system that allows developers to develop, evaluate and deploy their applications on the same platform. Mercury selected Wind River's Platform for Network Equipment, Linux Edition because it provides the company with a commercial-grade quality Carrier Grade Linux (CGL) platform and enables developers to standardize on a single development suite for all device software projects.

Ensemble2 is built around the power, functionality, and scalability of serial RapidIO, Gigabit and 10 Gigabit Ethernet, AdvancedTCA (ATCA), and AdvancedMC (AMC). The platform supports a variety of I/O sources and heterogeneous processing, thereby reducing integration costs, improving efficiency, and minimizing risks in design and deployment of next-generation applications.

Based on a CGL distribution, Wind River Platform for Network Equipment, Linux Edition, integrates with many telecommunications boards, making it an ideal fit for the telecommunications and high-end data networking markets. It supports the emerging PICMG 3.x ATCA specification and various high-availability functions, and includes a standards-based interprocess communication (IPC) mechanism that connects the Linux and VxWorks editions of the platform. This tested and validated Linux platform is supported and maintained by Wind River.
www.windriver.com
www.mc.com

enterprise —continued from page 7

to hope that Chinese business people will live up to their side of that bargain?

My experience, based our initial sales success in the Chinese oil and gas industry, tells me the answer is yes. China is furiously trying to develop energy supplies for its vast and growing population, and realizes it needs help to do so. It doesn't make sense that our customer, a giant oil company whose shares are traded on the New York Stock Exchange, would cheat on someone who is helping it overcome obstacles. Is this a leap of faith? Maybe, but it's one made with the realization of the tremendous opportunities in enterprise software that exist in China.

We have made a further leap in entering into an outsourcing contract with ChinaSoft Resource Co., a branch of the China Software and Technology Service Corporation (CS&S). CS&S is an example of the efforts of the Chinese government to restructure its state-owned enterprises and bring more transparency to their operations. Through this alliance we eventually hope to outsource parts of our Quality Assurance and final test process, plus other functions. Giving us confidence is the fact that among other U.S. companies using ChinaSoft are Microsoft and IBM.

Don't Go It Alone

Although many things about doing business in China surprised us, one truism remained unchanged: you need local partners. We proudly display in our reception room a plaque certifying us as members of the Zhongguancun Software Association (ZSoft), and I'm glad to be a member of its advisory committee for American software companies. This plaque is the global economy's version of a Chamber of Commerce or Rotary Club membership citation

ZSoft is an interesting example of China's efforts to catch up with the world – particularly India – in software outsourcing. Named after the Zhongguancun region of Beijing, which is widely regarded as China's Silicon Valley, it functions as U.S. trade associations do to promote China's software industry. It recently hosted an outsourcing summit on the theme "Software Changing China" that was attended by more than 300 representatives of governments, associations, companies, and institutes.

Joining ZSoft was in my company's best interests, and also served China's goal of improving its software capabilities. China graduates two million engineers a year, but its largest software outsourcing firm employs fewer than 7,000 people. Clearly there's a gap between where they are at present and where they want to be. They see contact with U.S. software companies as an opportunity to learn what skillsets they need to develop and the direction that technology is headed. As a young company trying to get a foothold in the Chinese enterprise software market, we clearly need the kind of contacts that ZSoft offers. And if they can learn a thing or two from us, that's fine, too.

Be Opportunistic, Be Smart

My overall assessment of the Chinese opportunity for U.S. enterprise software companies is optimistic. For companies in the storage and security fields, particularly, China presents an enormous opportunity. The commoditization of high-end serving technology, which bedevils companies like Sun and SGI, has opened up a large market for companies that can offer solutions for the new-era enterprise data center. China's preference for Linux-based server architectures is a real advantage for those who can deal in the Open Source environment.

It's not a perfect world in China. Its transformation from a planned economy to a market-driven economy proceeds in fits and starts. As often as you see promising developments you run into incomprehensible regulations. Piracy and lack of protection for intellectual property remain serious problems. Finding local partners with a commitment to outflanking the bureaucracy and guaranteeing IP integrity is an absolute necessity. But, in a brave new world, only the brave will thrive.

enterprise —continued from page 27

model and the network appliance model. The customer receives an integrated solution that combines the application with a streamlined operating system. The software appliance might install natively on the hardware via an installation CD or DVD, or it could be delivered to the customer in a virtual machine format that runs atop another OS or atop a standard virtualization layer like VMware's ESX server. Maintenance for the entire solution is received from the application provider via a simple web user interface. Since maintenance comes from the application provider, it is pre-tested and certified in the exact environment the customer has deployed, so there are no more mismatched maintenance streams from various vendors.

Many will argue that Windows-based apps benefit from the concept of a "universal" platform. Not true. Microsoft doesn't allow the OS to be stripped of all components except those required to support your application (Microsoft even argued in court that a browser was an integral part of an OS), nor does it let you pass judgment on its maintenance stream before releasing it to your customer base. Microsoft has a "one size fits all" mentality that's inconsistent with the concept of a software appliance. The operating system should be secure, reliable, small, and practically invisible. It should also be free so that your application is never held hostage to the technical or economic whims of another vendor. These are not attributes that describe Windows.

Parting Thoughts

Linux, open source infrastructure, and virtualization provide application vendors the historic opportunity to lose their independence and free their customers from the hassles of assembly and maintenance of complex software solutions. The initial transition is likely to require an investment in an optimized port for Linux, and some decisions regarding when and how to transition customers from legacy platforms will certainly be difficult.

It is inevitable, however, that customers are demanding lower starting costs and faster time to value from application vendors, and the legacy method of delivering software applications is not going to be acceptable. The great news is that engineering expense that used to be spent on context issues can be redirected to investments in product features, sales, and marketing. Lower cost of entry also tends to expand the available market, opening up customer opportunities that were previously unreachable due to the expense of complex assembly and maintenance routines. It's time to stop thinking of Linux and open source as a threat or simply another application port. It's time to embrace the open source trend as a strategic opportunity to improve the economics and market position of the software application business.

JAVA J2EE-Ready Managed Dedicated Hosting Plans:

Xeon III

SAMEDAY SETUP

Dual 3.2 GHz Xeons
4GB RAM
Dual 73GB SCSI
1U Server
Firewall
Linux
Monitoring
NGASI Manager

$399 *monthly*

Pentium 4 I

SAMEDAY SETUP
FREE SETUP

2.4 GHz P4
2GB RAM
Dual 80GB ATA
1U Server
Firewall
Linux
Monitoring
NGASI Manager

$199 *monthly*
2nd month FREE

4Balance I

1 Database Server and 2 Application Servers connected to 1 dedicated load balancing device.
Dual Xeons.
High-Availability.

$1724 *monthly*

NEW! Geronimo 1.0 Hosting . Virtual Private Servers(VPS) starting at $69/month

At **WebAppCabaret** we specialize in **JAVA J2EE Hosting**, featuring **managed dedicated servers** preloaded with most open source JAVA technologies.
PRELOADED WITH:
JDK1.4 . JDK1.5 . Tomcat . Geronimo . JBoss . Struts . ANT . Spring . Hibernate
Apache . MySQL . PostgreSQL . Portals . CRM . CMS . Blogs . Frameworks
All easily manage via a web based control panel.
Details:
-All Servers installed with the latest Enterprise Linux
-Firewall Protection
-Up to 60 GB daily on site backup included at no extra charge per server.
-Database on site backup every 2 hours
-Daily off site database backup
-A spare server is always available in case one of the server goes down
-Intrusion detection.
-24x7 Server and application monitoring with automatic self healing
-The Latest Bug fixes and Security updates.
-Tier 1 Data Center. 100% Network Uptime Guarantee
-Guaranteed Reliability backed by industry-leading Service Level Agreements

Log on now at **http://www.webappcabaret.com/jdj.jsp** or call today at **1.866.256.7973**

NGASI POWERED

WebAppCabaret sm

JAVA J2EE Hosting

Open Source and Middleware

Interview with Pierre Fricke, director of product management at JBoss

Interviewed by Jeremy Geelan

JEREMY GEELAN: We're here with Pierre Fricke, director of product management at JBoss, at LinuxWorld in Boston. JBoss has made about 77 announcements and it's only day two, and there's some other stuff we must get onto. Thank you for joining us Pierre.

PIERRE FRICKE: Thank you, Jeremy. It's been exciting to be at LinuxWorld.

SYS-CON.TV

Visit SYS-CON.TV to watch this interview and many more videos!

GEELAN: You guys, announcements every half minute.

FRICKE: There's a lot going on in open source in middleware, so let me take you through a couple of the announcements we've made. Last week we had a couple of interesting announcements. We announced JBoss Messaging – and it's 1.0. It's a first release of refactoring JBoss MQ to be a high-performance messaging core that can support high-end SOA and high-end ESB scenarios.

GEELAN: That's quite a shift, quite an innovation.

FRICKE: It's interesting because it's modular. There's a messaging core and then we have protocol façades. The first protocol façade is JMS, but we can support other messaging protocols so it makes it flexible. There are some people who don't want to use JMS and they may want to use something else. We will enable those kinds of protocols. We're excited about that because this is going to be the foundation for Enterprise Service Bus later in the year. We also announced the community release for JBoss Web server.

GEELAN: Now that was intriguing.

FRICKE: That announcement was very interesting. The JBoss Web server basically is designed to – think of it as Tomcat on steroids. It basically marries Tomcat with the Web server and basically brings these things together in a more integrated fashion, for high performance Web scenarios, and we've seen more than 10,000 concurrent clients going through the Web server into the Java environment. We have situations in JBoss where we need that and customers are looking for those kinds of higher-end scenarios.

GEELAN: Yes, absolutely.

FRICKE: JBoss Web is real exciting. The other thing about it is we're going to be able to run ASP.NET and PHP.

GEELAN: That's kind of the flavor of the show here, anyway, in our interop age. But you guys have always been ready for that. Of course, one begins to wonder about the J in JBoss but let's not even go there. You know, we're going to be in trouble. Did you see any of Bill Hilf's announcements? Have you been monitoring the brouhaha around Microsoft at this particular LinuxWorld?

FRICKE: No, actually, I have not focused too much on that.

GEELAN: Okay, that awaits you then.

FRICKE: I was going to catch up on that tomorrow.

GEELAN: We can no longer extinguish so we are embracing – let me just put it as simply as that. The jury is out on where the embrace will lead. A lot is going on in general, as you're saying, with open source. Even more so though, Pierre, a lot is going on with the technology space in general.

I wonder if I cannot observe our tried and true tradition, if I may, of using you as a kind of observer, an anthropologist of technology is how I've always thought of you with your analyst background and you're completely immersed in the real-world beast of e-commerce and the Net. Let me ask you this, obviously one wants a renaissance in technology. But because of the economic cycle, one perhaps wants a renaissance, he wants a user because of usability. There are all sorts of reasons for us wanting, not just to have a VC-pumped bubble two, but something really substantive. Let me put it to you that this particular LinuxWorld, and I think you've been to like all of them, or most of them...

FRICKE: Almost all of them.

GEELAN: This particular LinuxWorld is beginning to resemble, to me, our old friend Malcolm Gladwell's The Tipping Point in that it seems to

About the Interviewer

Jeremy Geelan is group publisher and editorial director of SYS-CON Media and is responsible for the development of new titles and technology portals for the firm. He regularly represents SYS-CON at conferences and trade shows, speaking to technology audiences both in North America and overseas.

jeremy@sys-con.com

be very different from any other LinuxWorld. I'm not quite sure in which sense yet it's different and would love your help with that, but some of the theories I'm hearing are that perhaps it's quieting down in the sense of traffic and noise because the signal is stronger and we're going somewhere. A lot of people can't get here because they're just too darn busy. You don't have to make the case; that's one theory. Of course there are the doom mongers saying, No, no, it just shows that we're all lost, we're all lost. Linux has been busted. Open source, professional open source, is an oxymoron. I'm sure that you don't go into that camp. But I would like your take. This LinuxWorld or this period, this quarter, places in 2006, what's different about exactly now?

FRICKE: Well, I was joking with Larry Augustin yesterday. I said, you know, if we had this in Orlando in the winter, you would definitely double the traffic.

GEELAN: It was very wet this morning.

FRICKE: Putting that aside, I do think these things rise and then they have to mature and then they adopt a new thing, but I'm an old Uniform attender. Remember Uniform in the 1980s?

GEELAN: Okay, now we are going back.

FRICKE: Well, you said anthropologist so let's go do some digging.

GEELAN: Historian now, archeologist.

FRICKE: Let's go do some digging. Uniform used to be – in 1985, I remember that, that was the technological coming together of Unix. In fact, if you looked at LinuxWorld '00 and Uniform 1985 they were very similar.

GEELAN: They would be analogous. We're right at the starting point and what you see is a cycle.

FRICKE: Uniform basically evolved into – the suits started showing up in '87, '88. IBM made the announcements, you know, Apollo was making announcements.

GEELAN: It is analogous.

FRICKE: The same thing happened here. In 1999 we had IBM and people here were kind of putting their toes in the water. By '01 they were investing a billion dollars in Linux, and HP was investing large amounts of money in Linux in multiple dimensions that competed with and complemented what IBM did. And the Novells and the Red Hats and all these people, and it was a maturing business show in '01, '02 and '03.

GEELAN: What is the next phase? It sounds to me like you're almost saying, for example, in show terms, LinuxWorld just becomes a technology show, and what else? What about in technology terms? Open source becomes what? Will the standard – will the phrase get retired eventually?

FRICKE: Open source?

GEELAN: Yes.

FRICKE: Not in my career.

GEELAN: That's not going to happen?

FRICKE: Well, never say never.

GEELAN: Right. But in a sense, if it's no longer a contradistinction because that becomes the predominant paradigm...

FRICKE: Well, there are still a lot of proprietary environments out there and especially up the stack, that we're still – the current contesting battleground is in middleware and databases.

GEELAN: Right. But do you see that going up the stack over time?

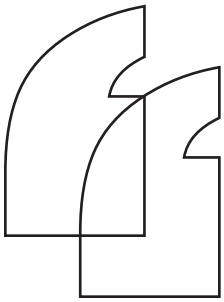
FRICKE: Yes, it's going to go up the stack. Open source will not take over the entire world, but it's going to become more prevalent at each layer of the stack, with the biggest part of the pyramid being at the operating system and Web tools. But Solaris hasn't gone away, although they had to adopt; AIX, HP-UX haven't gone away. In fact AIX is growing to some extent. Windows is growing; it hasn't gone away. But Linux is growing the fastest, right, so it's establishing its place in the sun, pun intended.

GEELAN: Now you know why we have Pierre to do this. This is good stuff.

FRICKE: It's establishing its place in the sun as a mainstream operating environment that a set of people will consume to solve business problems in a way that gives them the flexibility to do so. There are other people who are using HP-UX and AIX and Windows and they will continue to do so. Mainframes have been dead for a long time but IBM still makes a lot of money on mainframes.

GEELAN: Everything is going to keep on going.

FRICKE: So stuff doesn't die. It may mature and it may become a cash cow or it may become just a stable market. Then there are growth paradigms and Linux is in its hockey stick growth – it's maturing. When something matures, at these kinds of conferences you don't have as much new news any more.



WebSphere is really good for these intergalactic complex problems, and if you need a space shuttle that's the place to go. **We're good at the mass market, high volume, solving mainstream problems that don't require a great deal of complexity.**

GEELAN: Yes, the fizz goes out, but that's not a bad thing.

FRICKE: What it means is people are doing real work now. We're off the height curve and we're under the what we're just starting to call the plateau of productivity. That's where we are with Linux. Where we're still trying to sort things out, where there's still the battle now, is in middleware. JBoss versus WebLogic and WebSphere, and [bias QL] versus the local UB.

GEELAN: Now put on your JBoss hat because people are absolutely interested in that. Where are we on that? Within the ecosystem JBoss is high energy; you have this fantastic reputation; you are driven; you're nice. I mean, all these things are kind of weird but it's all coming together. Where are you on the great battle against those Goliaths?

FRICKE: I think we're basically where Linux was a little while ago and coming up the pike maybe in '01-'02 kind of a timeframe. Linux was really starting to climb and start to really establish itself and we're establishing ourselves now. We have at present 37 percent of the enterprise surveyed, tied with WebSphere at 37, so we've caught up with penetration with WebSphere. We're neck and neck in terms of penetration of the enterprise.

We tend to be used for different things. WebSphere is really good for these intergalactic complex problems, and if you need a space shuttle that's the place to go. We're good at the mass market, high volume, solving mainstream problems that don't require a great deal of complexity. In fact that have a great deal of synergy and the developers have a great deal of synergy with our simplicity message. You know, our developers get rewarded for developing things that are simple. At IBM you tend to get rewarded for developing things that are complex and solve complex problems, and we get rewarded and recognized in the industry for solving things in a simple way, solving problems in a simple way.

WebSphere is not going to go away, right, and BEA is not going to go away unless they get bought or something, but those things are going to live on. We're capturing a lot of the growth now.

GEELAN: Let's call a spade a spade. You all at JBoss must be anticipating one day sitting on 65% of the market. Then, do all the same things that have blighted the other players when they got a predominant market share, are they going to apply? Are we going to see a sullying of the JBoss purity because...

FRICKE: Bill Gates wrestled with this problem, and he's been focused on making sure he doesn't become the next IBM of the '80s. He knows what he did to IBM and he doesn't want to become that. Even now he pretends to be focused on that. I think Marc Fleury is very similar to Bill Gates in that respect. He's not going to let us become WebSphere and, by the way, there's another dimension to this that Microsoft doesn't have – our community will not let us become that.

GEELAN: You really believe that with a passion, that it's not just a manner of speaking, this community. The JBoss community is real and will not take any BS from...

FRICKE: No, that's one thing I've learned at JBoss, these guys know what they need to solve business problems, to solve development type problems, and they're not going to take force-fed technology to satisfy vendor needs. That's not going to work.

GEELAN: Change imposed is change opposed.

FRICKE: Right. What's going to happen is, I think you're going to see things like JBoss Seam, the new Web application framework that Gavin King has been leading, that marries EJB with JavaServer Faces, eliminates the glue code, incorporates jBPM for workflow, that's real exciting. You have this Web application environment that can do lots of different things in a very simple way including manage stateful type application environments, including workflow type scenarios in which you've have to go buy process servers at $20,000, a processor for other kinds of environments. That has been driven by the community because there's a real need to do things beyond the simple Web application framework we've seen in the past couple years. There's a need to really bring that level of simplicity to a larger scale of problem. That's what JBoss Seam is about. JBoss Seam will be pulled into the community as one of the key SOA component foundations, I think. Developers are going to love this development.

I went on the road with Gavin King and the JBoss operations network people. We did On The Road, a road show, kind of a good play on words. When Gavin got up there and presented Seam, you could hear a pin drop, guys were just –

GEELAN: – blown away.

FRICKE: : There were just smiles on the faces and excitement in the room. It was just incredible. You don't see that with these multi-headed hybrid hydra kind of consortium type of things that produce specifications, right? You just don't see that level of excitement.

GEELAN: This does kind of make my case that this particular moment of the cycle, it does seem to be enormously pregnant in a way that I haven't noticed for three or four years. Now JBoss is another use case for my great theory, so I'm going to absolutely keep a very, very close view, if I may, Pierre, and perhaps pursue this as a theme through this year, because obviously JBoss is part of the greater picture and as it becomes great, it would be nice to keep you humble and, because we want this kind of ecosystem view, we'll have to keep you honest, too.

FRICKE: Absolutely. We need that.

GEELAN: In the meantime, I want to thank Pierre Fricke for talking to SYS-CON.TV.

Open Management Consortium

Systems Management is Now Open

In May 2006 the Open Management Consortium was announced to help advance the promotion, adoption, development and integration of open source systems /network management software. The founding members of the consortium are Ayamon, Emu Software, Qlusters, Symbiot, Webmin, and Zenoss.

Specific objectives of the Open Management Consortium include:

- Create awareness of open source management tools in the market
- Provide education and resources to help end users make informed decisions regarding open source
- Establish conventions and standards that enable integration and interoperability
- Enable collaboration and coordination on common development projects
- Promote collaborative open source systems management solutions

Open source systems management replaces monolithic vendor lock-in with a modular approach. Pick what you need, customize it to your exact specifications and add to it as your needs change. Because open source products welcome contributions by users, partners and other third parties, they must be standards-based. This standards-based approach facilitates interoperability between open source solutions across the systems management life cycle.

AYAMON

EmuSoftware

Qlusters

symbiot security

Webmin

Zenoss
OPEN SOURCE ENTERPRISE MONITORING

www.open-management.org

~~Our CRM SYSTEM IS SO INFLEXIBLE.~~

~~Our CRM SYSTEM COSTS TOO MUCH.~~

Frustrated by CRM restrictions?

Take back control of your CRM solution with SugarCRM, the most powerful commercial open source CRM solution for managing sales, marketing, support, and group collaboration activities.

Learn More: www.sugarcrm.com/eosm

Complimentary white paper, "*Running a Business on Commercial Open Source*."

2006 ENTERPRISE> OPENSOURCE CONFERENCE

June 5–6 • New York, NY • USA

See **SugarCRM** demos and speak to CRM experts.

FREE CRM Solution Pack CD

Commercial Open Source Customer Relationship Management

SUGARCRM
COMMERCIAL OPEN SOURCE

www.sugarcrm.com • www.sugarforge.org

1.87SUGARCRM • +1.408.454.6941