

LINUX JOURNAL

Since 1994: The Original Magazine of the Linux Community

JULY 2014 | ISSUE 243 | www.linuxjournal.com

TINY CORE LINUX
FOR FIREFIGHTERS

HOW TO USE
MACROS—A
TIME-SAVING
FEATURE OF VIM

MOBILE

TIPS
FOR USING
OLD DEVICES

ANDROID APPS
FOR REMOTE
SYSTEM
ADMINISTRATION

GADGETS
FOR A
HEALTHY
LIFESTYLE

TRY FLASK,
A POWERFUL
MICRO-
FRAMEWORK



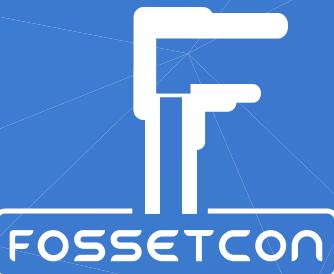
+
**A LOOK
AT PRIVATE
CLOUDS**
**OPEN-
SOURCE
NASA
SOFTWARE**



WATCH:

ISSUE OVERVIEW





FREE AND OPEN SOURCE SOFTWARE EXPO AND TECHNOLOGY CONFERENCE

FOSSETCON

2 0 1 4

Come out and participate in the First Annual Fossetcon 2014
Florida's Only Free and Open Source Conference. Within
10 minutes of Disney Land, Universal Studios and Epcot Center.

DAY 0

FOOD, TRAINING,
WORKSHOPS AND CERTIFICATIONS

DAY 1

FOOD, KEYNOTES, EXPO HALL,
SPEAKER TRACKS

DAY 2

FOOD, KEYNOTES, EXPO HALL,
SPEAKER TRACKS

BSD
Friendly

- FREE FOOD,
- TRAINING,
- CERTIFICATIONS
- AND GIVEAWAYS!!!

SEPT 11 - SEPT 13
ROSEN PLAZA HOTEL ORLANDO, FL

Fossetcon 2014: The Gateway To The Open Source Community

powered by:

CITRIX
Open@Citrix

ORACLE

MySQL

Omnibond
Oranges

ubuntu
Supported by Canonical

NVIDIA

conservancy

More info at
www.fossetcon.org



Are you considering software-defined storage?

zStax StorCore ZFS Unified Storage from Silicon Mechanics is truly software defined storage.

From modest data storage needs to a multi-tiered production storage environment, the **zStax StorCore** ZFS unified storage appliances have the right mix of performance, capacity, and reliability to fit your needs.

zStax StorCore 64



zStax StorCore 104



The **zStax StorCore 64** is your Tier 2 and 3 storage solution. While still leveraging all of the features inherent to the zStax platform, the StorCore 64 model offers an easily deployable and intuitively managed enterprise storage appliance. From backup and archival, to departmental file shares and streaming video, the zStax StorCore 64 has your needs covered.

The **zStax StorCore 104** is your system for highly available Tier 1 storage environments. Offering levels of redundancy, the StorCore 104 keeps your critical data available when competitors waiver. Finally, the StorCore 104 delivers a multi-tiered environment under one pane of management so you can eliminate the need for multiple vendors to satisfy your tiered data requirements.

CONTENTS

JULY 2014
ISSUE 243

MOBILE



Cover Image: © Can Stock Photo Inc. / janno028

FEATURES

54 Remote System Administration with Android

Work as sysadmin from afar, thanks to some Android apps.

Federico Kereki

74 New Uses for Old Devices

Got a drawer full of old phones? Check out what you could be doing with them.

Bill Childers

ON THE COVER

- Tiny Core Linux for Firefighters, p. 84
- How to Use Macros—a Time-Saving Feature of vim, p. 38
- Tips for Using Old Devices, p. 74
- Android Apps for Remote System Administration, p. 54
- Gadgets for a Healthy Lifestyle, p. 42
- Try Flask, a Powerful Microframework, p. 26
- A Look at Private Clouds, p. 98
- Open-Source NASA Software, p. 16

INDEPTH

84 Practical Tiny Core in the Fire Service

Tiny Core Linux Redux: Linux helping first responders.

Wilfredo Crespo

98 Is the Private Cloud a Real Cloud?

According to Gartner research, by 2015, the majority of private cloud computing services will evolve to leverage public cloud services in a hybrid model.

Mitesh Soni

COLUMNS

26 Reuven M. Lerner's At the Forge

Flask

34 Dave Taylor's Work the Shell

Days Between Dates?

38 Kyle Rankin's Hack and /

The Only Mac I Use

42 Shawn Powers' The Open-Source Classroom

Healthy Robots

112 Doc Searls' EOF

A Pain in the Person

IN EVERY ISSUE

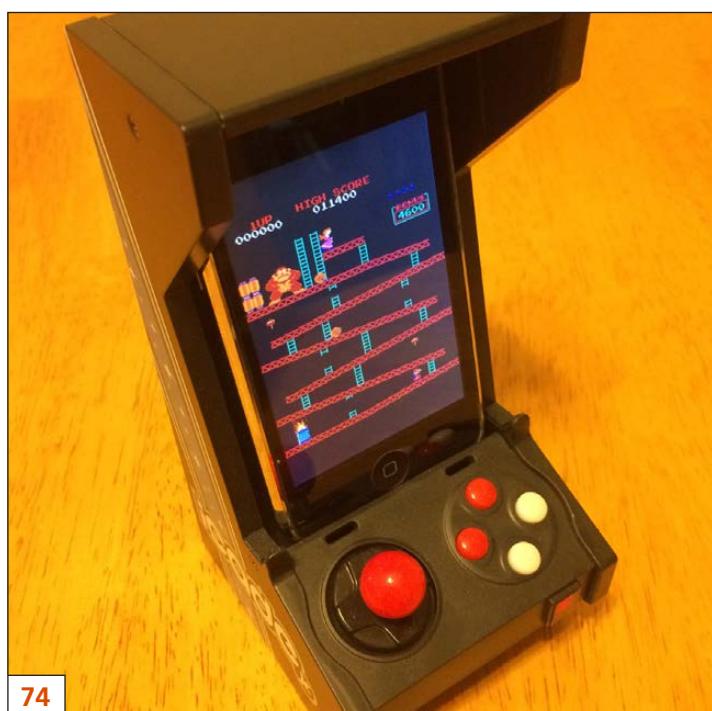
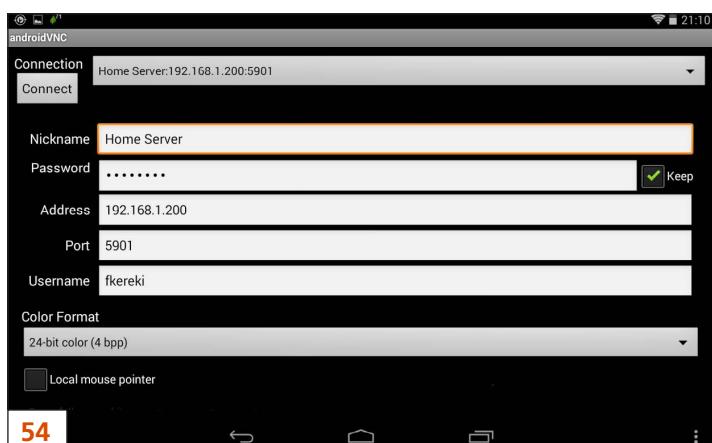
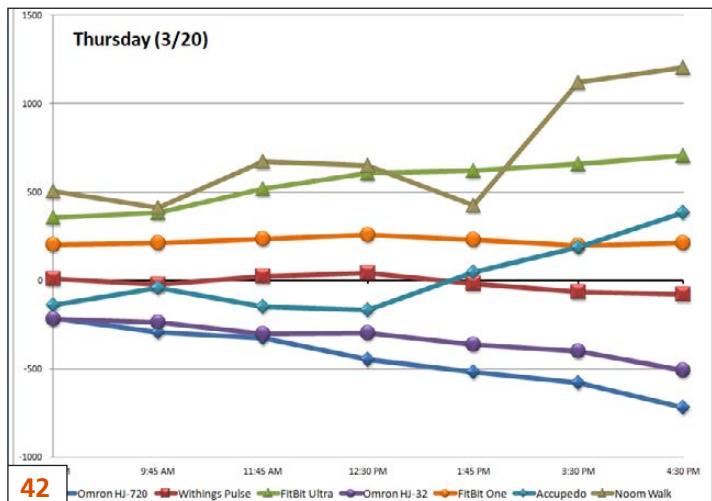
8 Current Issue.tar.gz

10 UPFRONT

24 Editors' Choice

50 New Products

115 Advertisers Index



LINUX JOURNAL™

**Subscribe to
Linux Journal
Digital Edition
for only
\$2.45 an issue.**



ENJOY:

Timely delivery

Off-line reading

Easy navigation

**Phrase search
and highlighting**

**Ability to save, clip
and share articles**

Embedded videos

**Android & iOS apps,
desktop and
e-Reader versions**

SUBSCRIBE TODAY!

LINUX JOURNAL

Executive Editor	Jill Franklin jill@linuxjournal.com
Senior Editor	Doc Searls doc@linuxjournal.com
Associate Editor	Shawn Powers shawn@linuxjournal.com
Art Director	Garrick Antikajian garrick@linuxjournal.com
Products Editor	James Gray newproducts@linuxjournal.com
Editor Emeritus	Don Marti dmarti@linuxjournal.com
Technical Editor	Michael Baxter mab@cruzio.com
Senior Columnist	Reuven Lerner reuven@lerner.co.il
Security Editor	Mick Bauer mick@visi.com
Hack Editor	Kyle Rankin lj@greenfly.net
Virtual Editor	Bill Childers bill.childers@linuxjournal.com

Contributing Editors

Ibrahim Haddad • Robert Love • Zack Brown • Dave Phillips • Marco Fioretti • Ludovic Marcotte
Paul Barry • Paul McKenney • Dave Taylor • Dirk Elmendorf • Justin Ryan • Adam Monsen

Publisher	Carlie Fairchild publisher@linuxjournal.com
Director of Sales	John Grogan john@linuxjournal.com
Associate Publisher	Mark Irgang mark@linuxjournal.com
Webmistress	Katherine Druckman webmistress@linuxjournal.com
Accountant	Candy Beauchamp acct@linuxjournal.com

***Linux Journal* is published by, and is a registered trade name of,
Belltown Media, Inc.**

PO Box 980985, Houston, TX 77098 USA

Editorial Advisory Panel

Brad Abram Baillio • Nick Baronian • Hari Boukis • Steve Case
Kalyana Krishna Chadalavada • Brian Conner • Caleb S. Cullen
Keir Davis • Michael Eager • Nick Falts • Dennis Franklin Frey
Victor Gregorio • Philip Jacob • Jay Kruizinga • David A. Lane
Steve Marquez • Dave McAllister • Carson McDonald • Craig Oda
Jeffrey D. Parent • Charnell Pugsley • Thomas Quinlan • Mike Roberts
Kristin Shoemaker • Chris D. Stark • Patrick Swartz • James Walker

Advertising

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

Subscriptions

E-MAIL: subs@linuxjournal.com
URL: www.linuxjournal.com/subscribe
MAIL: PO Box 980985, Houston, TX 77098 USA

LINUX is a registered trademark of Linus Torvalds.

LINUX JOURNAL DevOps

With deep focus on **Collaborative Development, Continuous Testing and Release & Deployment**, we offer here the DEFINITIVE **DevOps for Dummies, a mobile Application Development Primer** plus advice and help from expert sources like:

- Forrester
- Gartner
- IDC
- *Linux Journal*

Plus a host of other eBooks, videos, podcasts and more.

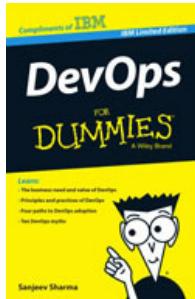
REGISTER NOW
and receive
unlimited access to
all site content and
downloads, plus
alerts when new
assets are made
available.

Have projects in development that need help?

Have a great development operation in place that can **ALWAYS be better?**

Regardless of where you are in your DevOps process, *Linux Journal* can help!

DevOps for Dummies

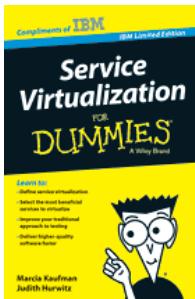


Free eBook

Today's fast-moving world makes DevOps essential for any business aspiring to be agile and lean in order to respond rapidly to changing customer and marketplace demands. This book helps you understand DevOps and how your organization can gain real business benefits from it.

You'll also discover how a holistic view of DevOps that encompasses the entire software delivery life cycle - from ideation and the conception of new business capabilities to implementation in production - can bring competitive advantage in a continuous delivery world.

Service Virtualization for Dummies Book



Free eBook

Discover service virtualization and how it fits into the big picture of software quality. In this book, *Service Virtualization For Dummies*, IBM Limited Edition, written by industry analysts Marcia Kaufman and Judith Hurwitz, learn how to deliver higher quality software by increasing the efficiency and effectiveness of your testing processes while reducing testing downtime and testing cost.

REGISTER NOW

<http://devops.linuxjournal.com>



SHAWN POWERS

Is That a Penguin in Your Pocket?

We're getting to the point in technological evolution that "mobile technology" no longer will be a term. It's becoming just "technology", as the stationary kind is less and less common. Perhaps I should trademark "Stationary Tech", in case the idea takes off! As Linux users, we've spent our lives expanding our install base to every device we can, so Linux on mobile devices is a pretty simple shift.

As a people group, developers have had a bigger learning curve when developing for low-power, tiny-sized mobile devices. It's forced devs to streamline their code and focus on UI more than ever before. Our resident developer, Reuven M. Lerner, starts this issue off by showing us the lean and powerful Flask framework that balances ease of use, a small core, oodles of features and a distinctly Python-like feeling when developing sites in it.



VIDEO:

Shawn Powers runs through the latest issue.

Dave Taylor walks us through creating the logic for a script that counts days gone by. Thanks to leap years, counting backward in days can be a confusing endeavor. Combining math and ingenuity, Dave shows how it's done. Kyle Rankin follows Dave and teaches us all how to use a mac—more specifically, a macRO for the vim editor. Kyle gets far more use out of vim than I do, even though it's the editor I use on the command line as well. We should all learn to be a little more effective on the console if we heed Kyle's wisdom. This month, as usual, he has wisdom aplenty.

I decided to take a shift this month and talk about health. Since I'm a nerd, when I talk about health, I talk about how technology can help promote that health. Specifically, I talk about how technology can aid you in living a healthy lifestyle. Whether you want to run a marathon or just make cool graphs of your weight loss/gain, my column should have something interesting for everyone. If you ever hang out with Kyle Rankin,

you should ask him about managing downtime in his data center from atop a ski lift—or maybe it was in a mountain café. I can't really remember. While he was years before his time, modern sysadmins are required to manage their server rooms at a moment's notice, and Federico Kereki gives us some great tools for doing real work on our Android mobile devices. I might not always have my laptop with me, but I don't even go to the bathroom without my phone. Federico helps make sure we get the most out of our mobile devices.

Bill Childers addresses a topic near and dear to my heart this month, when he talks about getting new use out of old mobile devices. A few months back, I shared my (continuing) adventures with BirdCam. Bill goes about 20 steps further and gives us tons of cool projects and ideas for our outdated phones, tablets and other devices. Some of them already might have occurred to you, but some will be new and interesting. Open up your bottom drawer and grab those old phones!

We also get an in-depth look at Tiny Core Linux from Wilfredo Crespo. Wilfredo takes us through the process of customizing the minimalist distribution to fit any particular set of requirements. In his case, the need is for a Web kiosk application to display call information in a fire department. I'm definitely in favor of

Linux helping save lives, and this month, we get firsthand instructions on how it's happening. Finally, in this issue, we hear from Mitesh Soni about cloud computing. If a company decides not to trust public cloud vendors with its data, and so designs a private cloud infrastructure with their own hardware, is that still a cloud? Mitesh teaches us about the nuances of the private cloud, how the concept can coexist with public offerings, and what it means for the future of cloud computing. Anyone developing applications for the cloud won't want to miss it.

Does mobile computing mean the end of the desktop is near? As someone with multiple jumbo monitors on his desk, I can say I don't think that day is coming any time soon. I can say with certainty that the mobile technology market is only going to keep growing, however. Much like data centers and cloud computing, Linux is a huge piece of the push for mobile technology. This month's issue is proof. Plus, it's just a really fun issue to read. We hope you enjoy it as much as we enjoyed putting it together! ■

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

diff -u

WHAT'S NEW IN KERNEL DEVELOPMENT

Once in a while someone points out a **POSIX violation** in Linux. Often the answer is to fix the violation, but sometimes **Linus Torvalds** decides that the POSIX behavior is broken, in which case they keep the Linux behavior, but they might build an additional POSIX compatibility layer, even if that layer is slower and less efficient.

This time, **Michael Kerrisk** reported a POSIX violation that affected file operations. Apparently, reading and writing to files during multithreaded operations could hit race conditions and overwrite each other's changes.

There was some discussion over whether this was really a violation of POSIX, but ultimately, who cares? Data clobbering is bad. After Michael posted some code to reproduce the problem, the conversation focused on what to do to fix it. But Michael did make an argument that "Linux isn't consistent with UNIX since early times. (E.g., page 191 of the 1992 edition of Stevens APUE discusses the sharing of the file offset between the parent and child after fork().) Although Stevens didn't explicitly spell out the atomicity

guarantee, the discussion there would be a bit nonsensical without the presumption of that guarantee.)"

AI Viro joined Linus in trying to come up with a fix. Linus tried introducing a simple mutex to lock files so that write operations couldn't clobber each other, and AI offered his own refinements that improved on Linus' patch.

At one point, Linus explained the history of the bug itself. Apparently, once upon a time the file pointer, which told the system where to write into the file, had been locked in a semaphore so only one process could do anything to it at a time. But, they took it out of the semaphore in order to accommodate device files and other non-regular files that ran into race conditions when users were barred from writing to them whenever they pleased.

That was what introduced the bug. At the time, it slipped through undetected, because that actual reading and writing to regular files was still handled atomically by the kernel. It was only the file pointer itself that could get out of sync. And, because high-speed threaded file operations are a pretty

At Your Service

rare need, it took a long time for anyone to run into the problem and report it.

An interesting little detail is that, while Linus and Al were hunting for a fix, Al at one point complained that the approach Linus was taking wouldn't support certain architectures, including **ARM** and **PowerPC**. Linus' response was, "I doubt it's worth caring about. [...] If the ARM/PPC people end up caring, they could add the struct-return support to gcc."

It's always interesting to see how corner cases crop up and get dealt with. In some cases, part of the fix has to happen in the kernel, part in GCC and part elsewhere. In this particular instance, Al felt the whole thing could be done in the kernel, and he was inspired to write his own version of the patch, which Linus accepted.

Andi Kleen wanted to add low-level CPU event support to **perf**. The problem was that there could be tons of low-level events, and it varied widely from CPU to CPU. Even storing the possible events in memory for all CPUs would significantly increase the kernel's running size. So, hard-coding this information into the kernel would be problematic.

He pointed out that the **OProfile** tool relied on publicly available lists of these events, though he said the OProfile developers didn't always keep their lists up to date with the latest available versions.

To solve these issues, Andi submitted a patch that allowed perf to identify which event-list was needed for the particular CPU on the given system, and automatically download the latest version of that list from its home location. Then perf could interpret the list and analyze the events, without

SUBSCRIPTIONS: *Linux Journal* is available in a variety of digital formats, including PDF, .epub, .mobi and an on-line digital edition, as well as apps for iOS and Android devices. Renewing your subscription, changing your e-mail address for issue delivery, paying your invoice, viewing your account details or other subscription inquiries can be done instantly on-line: <http://www.linuxjournal.com/subs>. E-mail us at subs@linuxjournal.com or reach us via postal mail at *Linux Journal*, PO Box 980985, Houston, TX 77098 USA. Please remember to include your complete name and address when contacting us.

ACCESSING THE DIGITAL ARCHIVE: Your monthly download notifications will have links to the various formats and to the digital archive. To access the digital archive at any time, log in at <http://www.linuxjournal.com/digital>.

LETTERS TO THE EDITOR: We welcome your letters and encourage you to submit them at <http://www.linuxjournal.com/contact> or mail them to *Linux Journal*, PO Box 980985, Houston, TX 77098 USA. Letters may be edited for space and clarity.

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

FREE e-NEWSLETTERS: *Linux Journal* editors publish newsletters on both a weekly and monthly basis. Receive late-breaking news, technical tips and tricks, an inside look at upcoming issues and links to in-depth stories featured on <http://www.linuxjournal.com>. Subscribe for free today: <http://www.linuxjournal.com/enewsletters>.

ADVERTISING: *Linux Journal* is a great resource for readers and advertisers alike. Request a media kit, view our current editorial calendar and advertising due dates, or learn more about other advertising and marketing opportunities by visiting us on-line: <http://www.linuxjournal.com/advertising>. Contact us directly for further information: ads@linuxjournal.com or +1 713-344-1956 ext. 2.

They Said It

Take the attitude of a student, never be too big to ask questions, never know too much to learn something new.

—Og Mandino

It is curious that physical courage should be so common in the world and moral courage so rare.

—Mark Twain

Observe your enemies, for they first find out your faults.

—Antisthenes

We act as though comfort and luxury were the chief requirements of life, when all that we need to make us happy is something to be enthusiastic about.

—Charles Kingsley

You don't become great by trying to be great. You become great by wanting to do something, and then doing it so hard that you become great in the process.

—Randall Munroe

overburdening the kernel.

There was various feedback to Andi's code, mostly to do with which directory should house the event-lists, and what the filenames should be called. The behavior of the code itself seemed to get a good reception. One detail that may turn out to be more controversial than the others was Andi's decision to download the lists to a subdirectory of the user's own home directory. Andi said that otherwise users might be encouraged to download the event-lists as the root user, which would be bad security practice.

Sasha Levin recently posted a script to translate the **hexadecimal offsets** from stack dumps into meaningful line numbers that pointed into the kernel's source files. So something like "fffffffff811f0ec8" might be translated into "fs/proc/generic.c:445".

However, it turned out that Linus Torvalds was planning to remove the hex offsets from the stack dumps for exactly the reason that they were unreadable. So Sasha's code was about to go out of date.

They went back and forth a bit on it. At first Sasha decided to rely on data stored in the **System.map** file to compensate, but Linus pointed out that some people, including him, didn't keep their System.map file around. Linus recommended using /usr/bin/nm to extract the symbols from the compiled kernel files.

So, it seems as though Sasha's script may actually provide meaningful file and line numbers for debugging stack dumps, assuming the stack dumps provide enough information to do the calculations.—ZACK BROWN

Adminer—Better Than Awesome!

The screenshot shows the Adminer 3.1.0 login page. At the top left, there is a language selection dropdown set to "English". Below it, the Adminer logo and version (3.1.0) are displayed. On the right, a large "Login" button is centered above a form. The form contains four input fields: "System" (set to "MySQL"), "Server" (set to "localhost"), "Username" (empty), and "Password" (empty). Below the form is a "Login" button and a checked "Permanent login" checkbox.

System	MySQL
Server	localhost
Username	
Password	

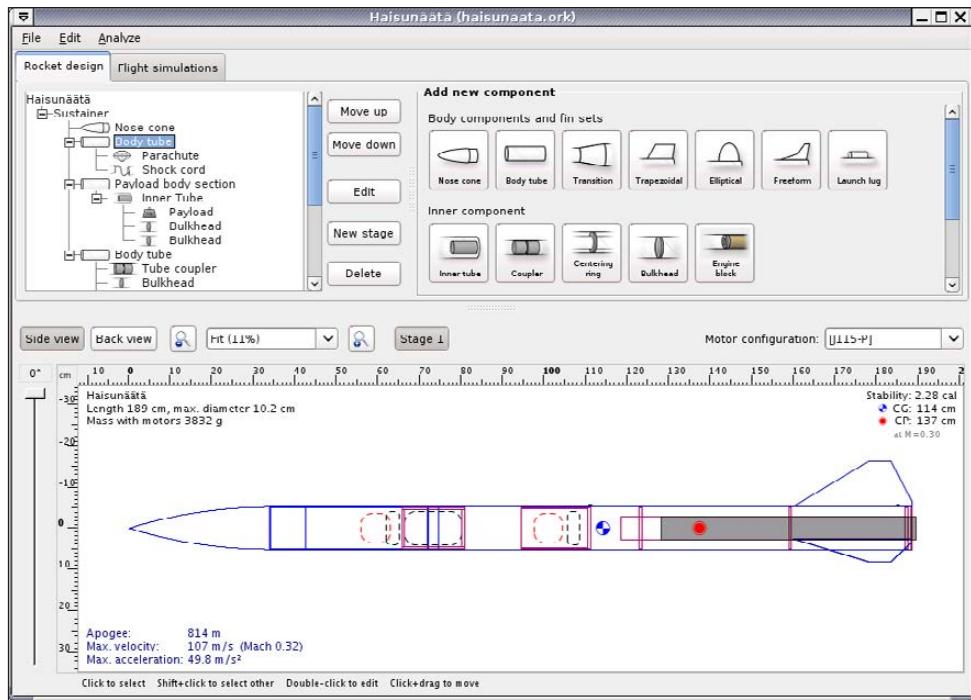
Permanent login

I've always loved PHPMyAdmin for managing MySQL databases. It's Web-based, fairly robust and as powerful as I've ever needed. Basically, it's awesome. Today, however, I discovered something better than awesome: Adminer. Although it is conceptually identical to PHPMyAdmin, it is far simpler and far more powerful. How can it be both? The Adminer Web site has a great feature comparison: <http://www.adminer.org/en/phpmyadmin>.

For me, the interface is basic, no-nonsense and intuitive. I like that installation is a single PHP file, and I also like that it supports alternate database systems like Postgres. If you are someone who prefers to use a Web interface over the command line, don't be ashamed. Heck, I recently managed an entire database department at a university, and I still prefer a Web-based interface. Anyway, if you're like me, you'll love Adminer. Get your copy today at <http://www.adminer.org>.

—SHAWN POWERS

It Actually Is Rocket Science



Screenshot from the OpenRocket Web Site (<http://openrocket.sourceforge.net>) Depicting a 2-D View of the Rocket Design Process

I've never once made a model rocket. I've always wanted to, but apart from "tube with explodey rocket part", I really didn't know where to start with designing. I recently found an open-source application that should help me with my lack of rocket science know-how: OpenRocket.

The aspect of actually designing a rocket appeals to me, because not only will I have a better chance of

launching a rocket successfully, but I'll also be able to compare expected results with actual results. If my carefully designed rocket veers into the neighbor's yard and blows up the dog house, I want to be able to figure out why!

If you've always wanted to launch a model rocket, but never had that really cool middle-school science

teacher that showed everyone how, check out OpenRocket. Even if you did launch rockets in school, with OpenRocket, you should be able to design a far more complex (and more awesome!) design on your computer. If you have any success with your pre-designed rocket, I'd love to see a video! Send a YouTube link to shawn@linuxjournal.com.

—SHAWN POWERS

Great Scott! It's Version 13!



No matter how much I love Plex, there's still nothing that comes close to XBMC for usability when it comes to watching your network media on a television. I've probably written a dozen articles on Plex during the last few years, so you know that's tough for me to admit. Still, no matter how many Plex-enabled devices I might buy (Roku, Amazon Fire TV, phones, tablets, Web browsers), I run XBMC on all my televisions. The interface, when coupled with a back-end MySQL database, is just unbeatable.

My ultimate dream is that XBMC and Plex would somehow merge together into an incredible living room experience that also kicks butt on a mobile device. Until that day of

convergence, I'll keep supporting two platforms. And, the XBMC platform recently got a significant upgrade. Version 13, code-named "Gotham" was released in May 2014. By the time you read this, 13.1 should be out, which fixes some bugs.

I'm most happy to see continued improvements with the Live TV and PVR features. Add to that further optimization for Android and Raspberry Pi devices, and XBMC is by no means out of the game. I'm excited to see XBMC continue along at a steady development pace. So, my weekend project once Saturday rolls around? Upgrading all my televisions to Gotham. Get a copy today at <http://www.xbmc.org>. —**SHAWN POWERS**

Open-Source Space

As I write this, NASA has just passed another milestone in releasing its work to the Open Source community. A press release came out announcing the release on April 10, 2014, of a new catalog of NASA software that is available as open source. This new catalog includes both older software

that was previously available, along with new software being released for the first time. The kinds of items available include project management systems, design tools, data handling and image processing. In this article, I take a quick look at some of the cool code available.

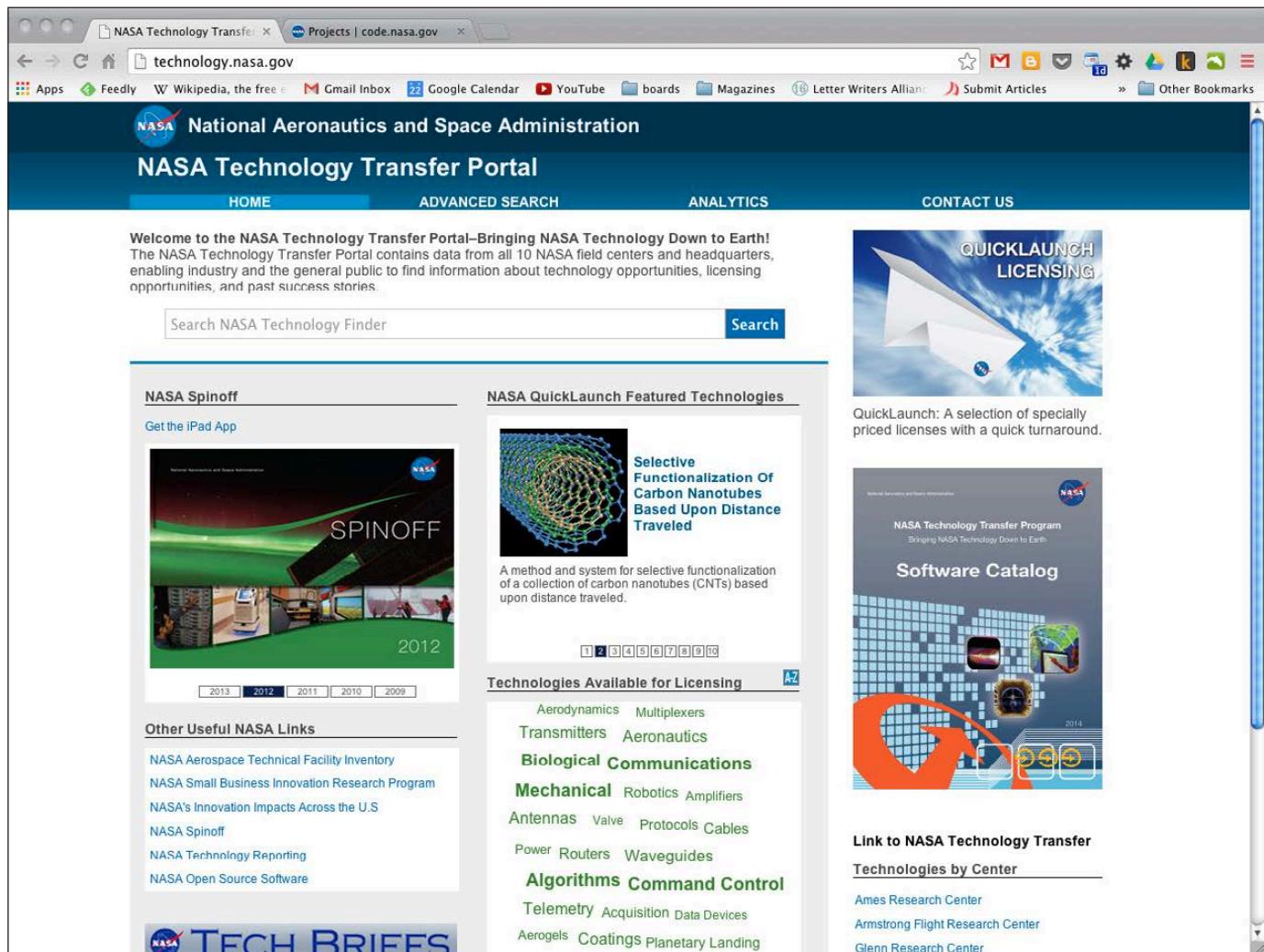


Figure 1. The main technology transfer site is a portal to provide access to everything NASA has to offer.

The main Web site is at <http://technology.nasa.gov>. This main page is a central portal for accessing all of the technology available to be transferred to the public. This includes patents, as well as software.

As a quick start, there is a subject cloud in the lower central region of the page that can do a search on several different keywords for you. Unfortunately, this is only a catalog

of all the offerings, and it's not quite complete yet in terms of detailed information. So, for example, if you click on Command Control, you will be taken to a results page that includes items like Rendezvous and Proximity Operations Program (RPOP). If you click on that, you will be taken to a details page that is essentially unpopulated. The assumption is that this will be filled in as time allows in

The screenshot shows a web browser window with the URL technology.nasa.gov/?t=techfinder. The page is titled "NASA Technology Transfer Portal". The main content area is titled "Command Control" and displays a list of search results. The results are paginated from 1 to 7. Each result entry includes the title, a brief description, the reference type (e.g., SOFTWARE or PATENT), and the date of publication. The results are categorized by NASA center: Johnson Space Center, Stennis Space Center, Glenn Research Center, and Marshall Space Flight Center. The results are also dated from 2011 to 2013.

Title	Description	Reference Type	Date	Center
Rendezvous And Proximity Operations Program (RPOP)	Reference: SOFTWARE MSC-24473-1	Software	2013-07-26	Johnson Space Center
Cursor Control Device Test Suite	Reference: SOFTWARE MSC-25214-1	Software	2013-07-26	Johnson Space Center
Valve Health Monitoring System Utilizing Smart Instrumentation for Real Time and ...	Reference: PATENT SSC-00247	Patent	2013-07-08	Stennis Space Center
Wireless Controlled Chaotropic Nanoionic Radio Frequency Switch	Reference: PATENT LEW-18919-1	Patent	2013-01-18	Glenn Research Center
Novel Winding and Control Schemes for Bearingless Motors	Reference: PATENT LEW-18895-1	Patent	2013-01-18	Glenn Research Center
A Comprehensive C++ Controller For A Magnetically Supported Vertical Rotor: Versi...	Reference: SOFTWARE LEW-17293-1	Software	2012-07-20	Glenn Research Center
Pulsed ultrasonic stir welding method	Reference: PATENT 8,393,523	Patent	2012-06-14	Marshall Space Flight Center
Future ATM (Air Traffic Management) Concepts Evaluation Tool (FACET)	Reference: SOFTWARE ARC-14653-1	Software	2012-04-23	Ames Research Center
Systems and methods for peak-seeking control	Reference: PATENT 8447443	Patent	2012-02-07	Armstrong Flight Research Center
Orientation control method and system for object in motion	Reference: PATENT MFS-32651-1	Patent	2012-01-03	Marshall Space Flight Center
High power RF solid state power amplifier system	Reference: PATENT MFS-32438-1	Patent	2011-08-23	Marshall Space Flight Center
Movable ground based recovery system for reusable space flight hardware	Reference: PATENT 8,498,756	Patent	2011-08-18	Ames Research Center
High-efficiency power module	Reference: PATENT 8,476,979	Patent	2011-07-07	Glenn Research Center

Figure 2. The results page on a search will give you a list of software and patents that are available from NASA.

[UPFRONT]

the future. It does give you a list of what is available though, which is half the battle.

Staying on the result list page, you should notice that there is the name of a NASA center on the right-hand side of each line. This is the actual source for the given patent or software entry. Once you find something of interest, you can go to the individual center's Web site to find more details about it. On the

lower-right section of the main page of the NASA technology site, you can find direct links to the technology sections for each of the individual centers. The amount of information available at each of these centers varies, but you should be able to find out more details. Some of the sites have direct download links, so you can get the software that interests you. In other cases, sites provide only the contact details for a person you'll

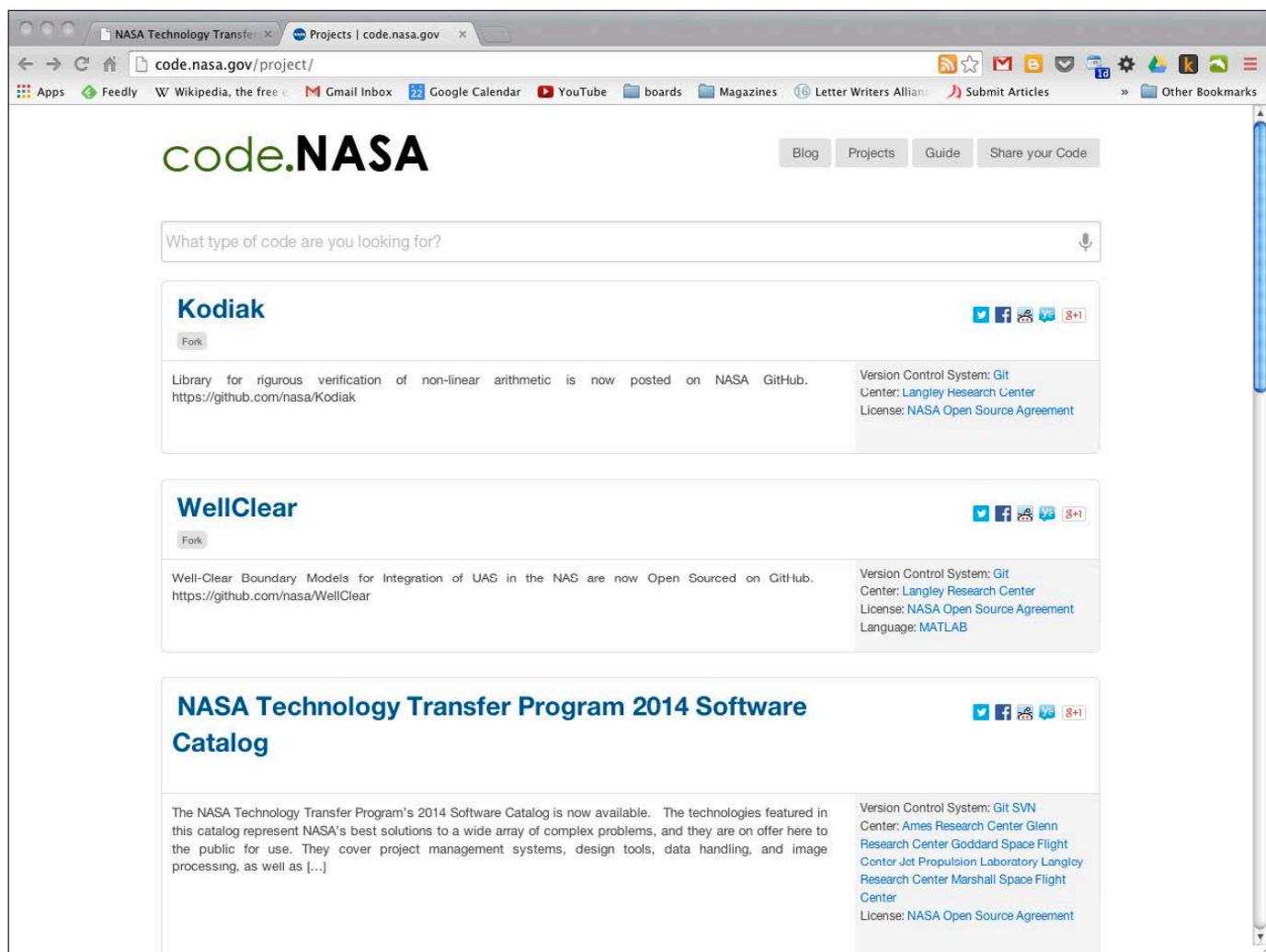


Figure 3. Software released under open source is available at this blog.

need to talk to in order to get copies of the software in question. A PDF catalog also is available on the front page of the main technology site. Here, you can get a 172-page catalog of all of the available software, broken down into 15 categories, for off-line access.

One issue that will become evident right away is that not everyone can access all of the available software. Some of the released software is

available only to US residents, and some is even more restricted to only parts of the US government. So, is there an easier option for the international community? On the front page, there is a set of other useful NASA links on the lower-left side. The one labeled NASA Open Source Software (<http://code.nasa.gov>) will take you to a sister site that provides access to a more centralized repository of software released as open source.

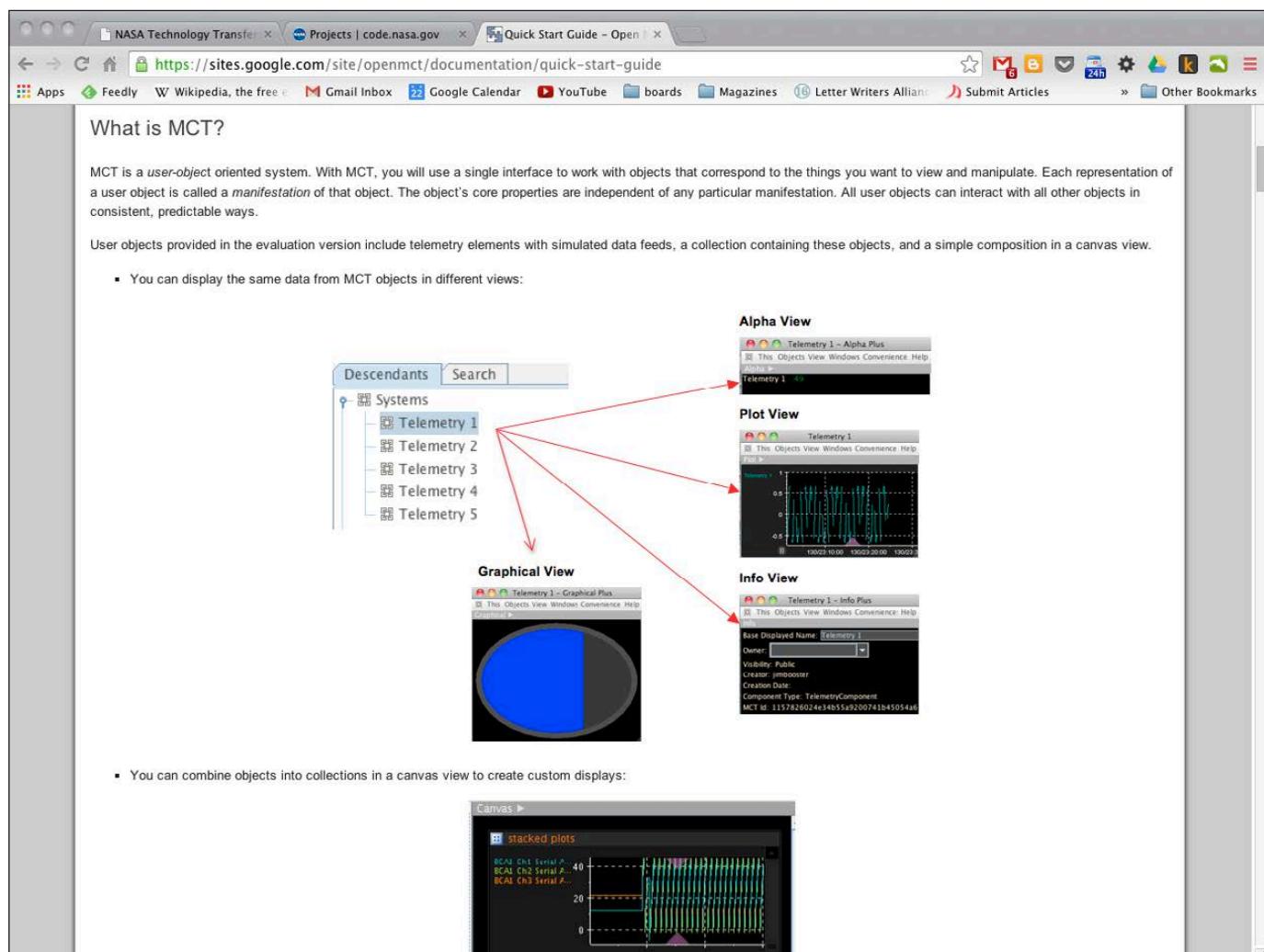


Figure 4. With the MCT, you can build your own application to monitor almost anything.

It is laid out as a list of available code within a WordPress blog, and it looks like it's being updated regularly. So, it's worth keeping an eye on this site for future releases.

So far, I haven't yet looked at what kind of software is available from the technology exchange at NASA, and there is a rather broad collection to play with. The first one I look at here is the Mission Control Technologies (MCT). This package, hosted

on GitHub, provides a real-time monitoring and visualizing platform that was developed at the Ames Research Center for use in spaceflight mission operations. It is based on configurable components, so you can use this to build your own application to monitor pretty much anything.

If you want to build your own spacecraft to monitor, you will need some way of controlling its flight. Enter the Core Flight Executive (cFE),

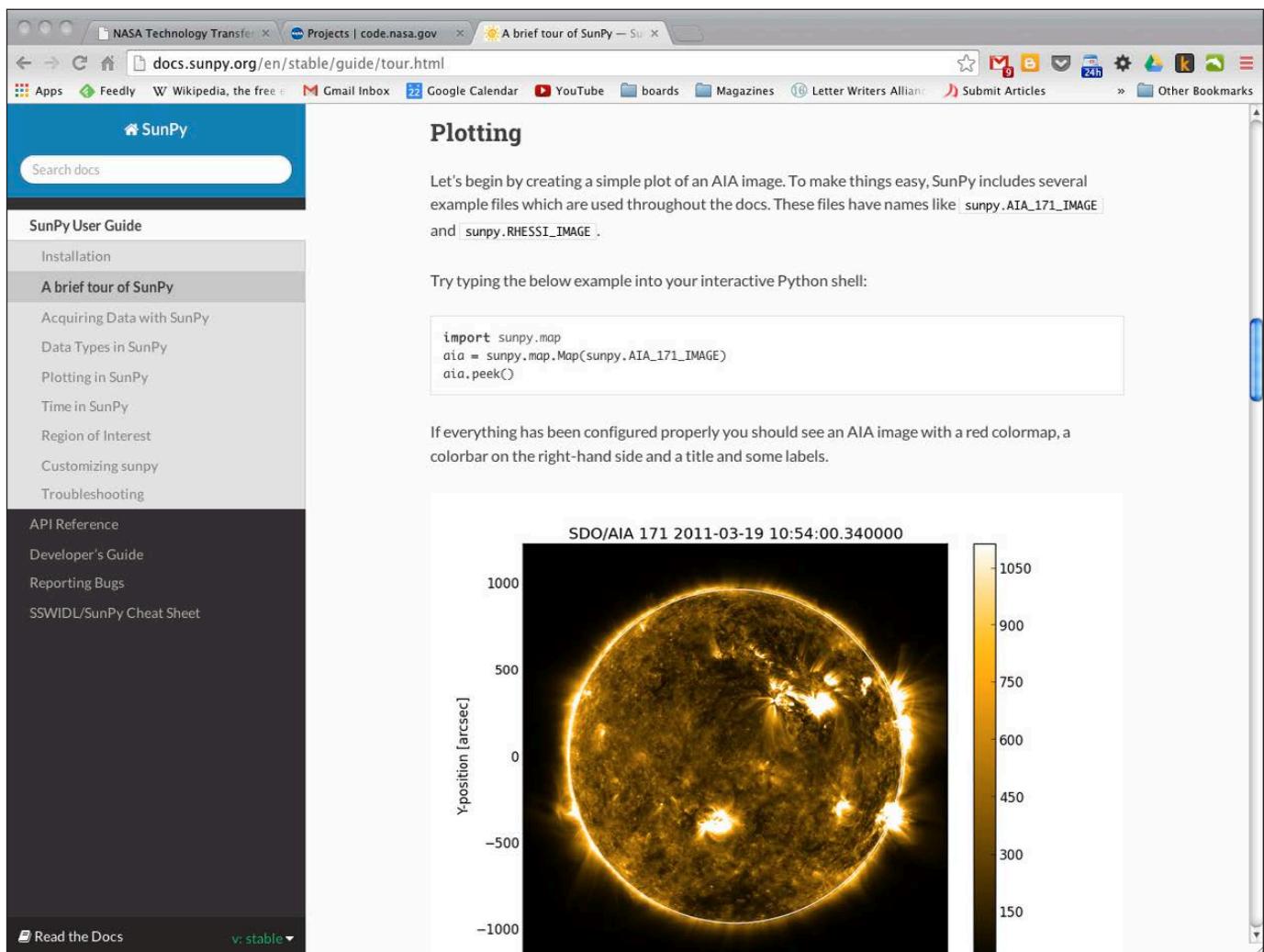


Figure 5. You can do all kinds of solar science with SunPy.

a portable, platform-independent embedded system framework developed at the Goddard Space Flight Center. It is used for flight software for satellite data systems and instruments, but you can use it for other embedded systems. It is built from subsystems including executive services, time services, event services, table services and a software bus. Python programmers can download

SunPy, a library to handle several tasks you run into when doing solar science.

For many scientific applications, you need to use clusters of machines. NASA is no exception to this. To handle the complexities, several software packages are available. For dealing with files, there are the Multi-Thread Multi-Node Utilities (Mutil). Mutil provides mcp and msum, which allow for parallelized access to files

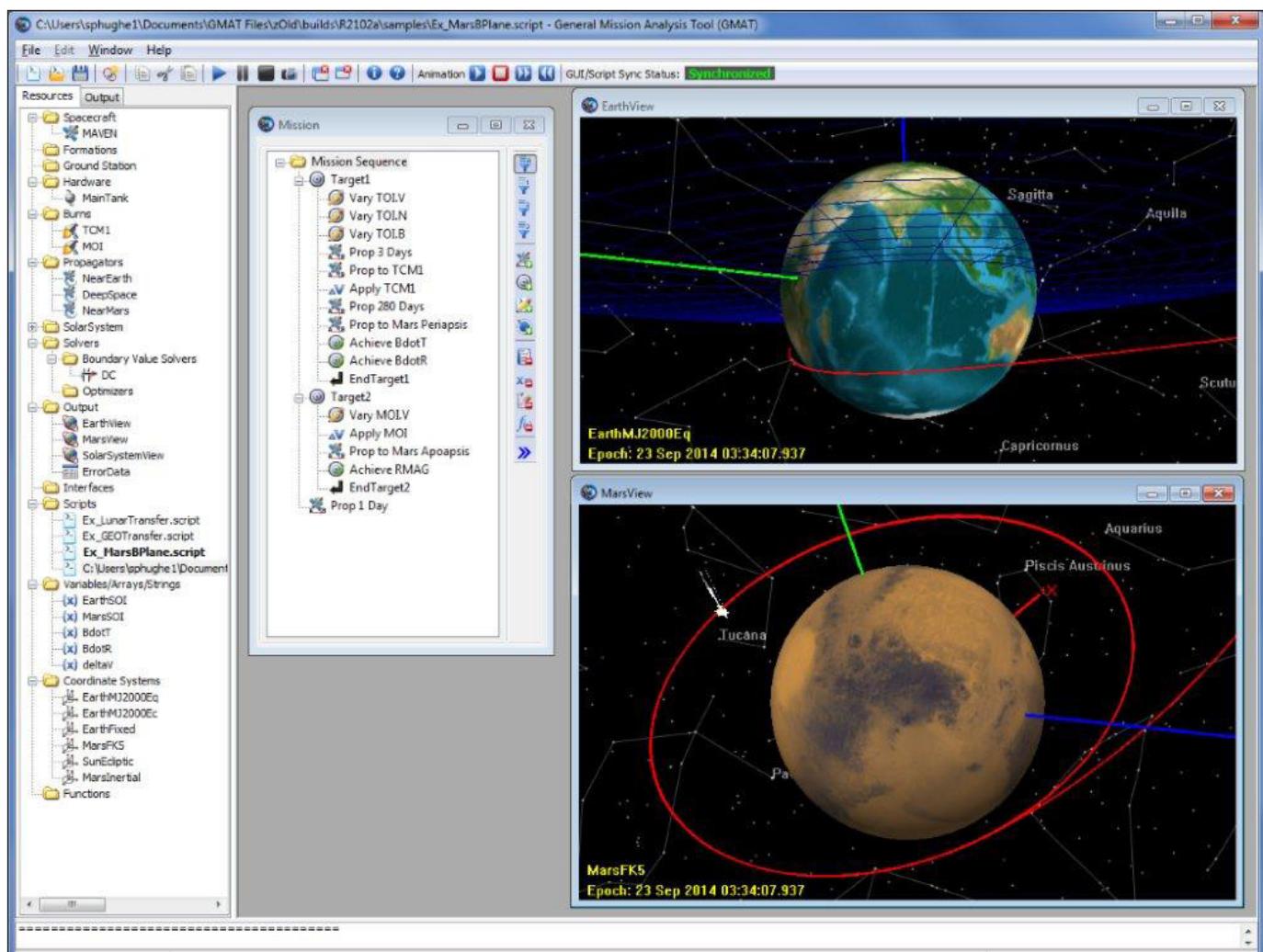


Figure 6. GMAT can help you plan out your next deep space mission from the comfort of your own living room.

for moving around a cluster. If you have a cluster of machines available over SSH, you can use them with Mesh (Middleware Using Existing SSH Hosts). Mesh provides a lightweight grid middleware that can group your cluster hosts into execution units. You then can issue a command, and Mesh will handle going to one of the available hosts in your group and running this command. If you need an interactive session, there is Ballast (Balancing Load Access Systems). With Ballast, when you try to SSH in to your cluster, you actually end up being shunted onto an available host within your cluster automatically.

The last package I want to look at is mission analysis. There is the General Mission Analysis Tool (GMAT), which is designed to help you plan your next trip to Mars. You can use GMAT to model, optimize and estimate spacecraft trajectories. You can create physical resources

required for the trip, like the spacecraft, thruster, tank, ground station and so on, and model how the trip will play itself out. There also are analysis model resources, including differential correctors, propagators and optimizers to define the details of the model. The user guide describes the multitude of available options. There also is a series of tutorials, including simulating an orbit, doing simple orbit transfers or even planning an optimal lunar flyby using multiple shooting that walks you through how to use GMAT in greater detail.

Now that you've looked at some of the newly released code from NASA, hopefully your interest is piqued enough to go exploring through the more than 1,000 other pieces of code available there. You never know what you may find. If you find something really interesting, please share it with the rest of us!—**JOEY BERNARD**

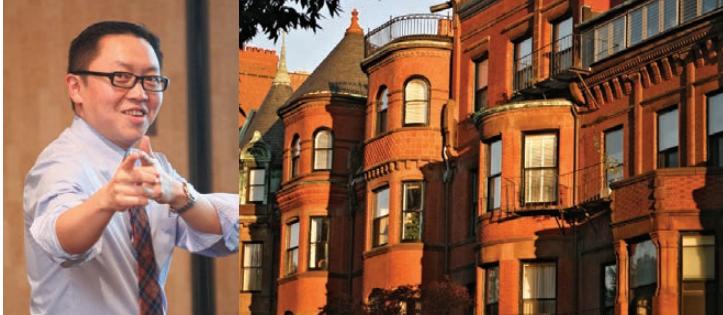
LETTERS TO THE EDITOR

We're moving Letters to the Editor to <http://www.linuxjournal.com/letters> to provide faster feedback and allow readers to comment. Please continue to send comments and feedback as usual via <http://www.linuxjournal.com/contact> or e-mail ljeditor@linuxjournal.com. We look forward to hearing from you!



The Best SharePoint Training in the World returns to Boston!

Choose from more than 80 classes and tutorials!



"I really enjoyed it. I can hardly wait to get back to work and start using what I learned. I will encourage employees and co-workers to attend future SPTechCons. The conference had great speakers with relevant subjects, and the whole thing was well organized."

—Greg Long, Infrastructure Development Manager, ITG, Inc.

"I prefer SPTechCon over Microsoft's SharePoint Conference in Vegas. I'm definitely going to tell others to go."

—Ray Ranson, Senior Architect, RSUI



September 16-19, 2014

The Boston Park Plaza Hotel & Towers

**Bolster your career by
becoming a SharePoint Master!**

- Learn from SharePoint experts, including dozens of SharePoint MVPs and Certified SharePoint Professionals
- Master document management
- Study SharePoint governance
- Find out about SharePoint 2013
- Learn how to create applications for SharePoint that solve real business problems
- Exchange SharePoint tips and tricks with colleagues
- Test-drive SharePoint solutions in the Exhibit Hall

If you or your team needs Microsoft SharePoint training, come to SPTechCon Boston!

Register Early and SAVE!



www.spotechcon.com

A **BZ Media** Event



@SPTechCon



SPTechCon™ is a trademark of BZ Media LLC. SharePoint® is a registered trademark of Microsoft.



Android Candy: Repix, Not Just Another Photo App

Apps like Instagram have made photo filters commonplace. I actually don't mind the vintage look for quick

cell-phone snapshots, but a filter can do only so much. At first glance, Repix is another one of those "make



your photo cool" apps that does little more than add a border and change saturation levels. It is more than that, however, taking photo modification to the next level and making it art.

The photo here, for instance, is from the Repix Flickr stream. It's obviously been filtered, but you're sure to notice there's a lot more going on. I'm not a terribly visually artistic person, but Repix allows a few simple touches to make a beautiful difference. If you're looking for a simple way to make your cat

photos a little more exciting, but don't want to have to transfer photos to a desktop application, check out Repix. The standard features are free, but with an in-app purchase, you can get more packages to play with.

Due to its ability to help a luddite like myself create artsy photographs, Repix gets this month's Editors' Choice award. If you like to take photos, but don't have an artistic bone in your body, I urge you to check it out. It's in the Play Store, but the Web site is <http://www.repix.it>.—**SHAWN POWERS**



Powerful: Rhino

- Rhino M4800/M6800**
- Dell Precision M6800 w/ Core i7 Quad (8 core)
 - 15.6"-17.3" QHD+ LED w/ **X@3200x1800**
 - NVidia Quadro K5100M
 - 750 GB - **1 TB** hard drive
 - Up to 32 GB RAM (1866 MHz)
 - DVD±RW or Blu-ray
 - 802.11a/b/g/n
 - Starts at \$1375
 - E6230, E6330, E6440, E6540 also available

- High performance NVidia 3-D on an QHD+ RGB/LED
- High performance Core i7 Quad CPUs, 32 GB RAM
- Ultimate configurability — choose your laptop's features
- One year Linux tech support — phone and email
- Three year manufacturer's on-site warranty
- Choice of pre-installed Linux distribution:



Tablet: Raven

- Raven X240**
- ThinkPad X240 by Lenovo
 - 12.5" FHD LED w/ X@1920x1080
 - 2.6-2.9 GHz Core i7
 - Up to 16 GB RAM
 - 180-256 GB SSD
 - Starts at \$1910
 - W540, T440, T540 also available



Rugged: Tarantula

- Tarantula CF-31**
- Panasonic Toughbook CF-31
 - Fully rugged **MIL-SPEC-810G** tested: drops, dust, moisture & more
 - 13.1" XGA TouchScreen
 - 2.4-2.8 GHz Core i5
 - Up to 16 GB RAM
 - 320-750 GB hard drive / 512 GB SSD
 - CF-19, CF-52, CF-H2, FZ-G1 available

EmperorLinux
...where Linux & laptops converge

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

Model specifications and availability may vary.





REUVEN M.
LERNER

Flask

Love Python, but don't want the overhead of a large Web framework? Try Flask, a lean, powerful microframework.

Let's face it, the Web has gotten big and complicated. No longer is it really possible for someone to be the "Webmaster", as we used to say back in the olden days of Web development. Today, we have front-end developers, back-end developers, system administrators, graphic designers, writers and any number of other jobs associated with the Web. Those of us fortunate enough to know a few of these things call ourselves "full-stack Web developers", but even full-stack developers need other people, with other talents, in order to get a Web application up and running.

As the Web has become more complex, so have the frameworks we use to develop applications. Once, we could put up a simple application in a matter of minutes by writing a CGI program. Later, it was enough to slap together a few pages of PHP or perhaps even a template that mixed HTML with a higher-level language. But then came the frameworks—first the big ones, from the Java and .NET

worlds, and then the open-source ones, particularly Rails (for Ruby) and Django (for Python).

These frameworks are totally amazing, and they do just about everything you ever would want from a Web-development framework. But over time, these frameworks—developed in order to get away from large, do-everything frameworks from the world of enterprise software—have become big. I won't use the term "bloated", because the fact is that I believe most framework maintainers are doing a good job of balancing the core needs and functionality with optional extras.

However, there are times when you want the best of all worlds—the ease and speed of creating something without a big framework, while still enjoying the benefits that a framework can provide. This is where "micro-frameworks" can suit your needs perfectly. For example, when creating the site that powers my consulting Web site (<http://lerner.co.il>), I

wanted there to be some dynamic content and also to be able to program things. But, I wasn't about to fire up a full instance of Rails or Django just for that.

One of the first, and best-known, microframeworks is Sinatra, which I covered in this column several years ago. Sinatra is written in Ruby, which makes it a great alternative to Rails for smaller projects. But if you're a Python developer, and particularly if you want to make use of the terrific Python infrastructure and community, you actually have several options from which to choose.

Perhaps the best known and most fully featured microframework for Python is Flask, written by Armin Ronacher and other members of the international "Pocoo" team of Python developers. There are other microframeworks for Python, such as Bottle, but Flask seems to do a good job of balancing ease of use, a small core, oodles of features, a distinctly Python-like feeling when developing sites in it and a large array of extensions that make it easy to add all sorts of functionality without writing it yourself.

So in this article, I take a brief tour of Flask and show how it can make life quite easy for Web developers. I've already incorporated Flask into the curriculum of some of my Python

courses, not only because it allows us to get up and running quickly, but also because I find that the design reinforces the coding style Python developers should aim to attain.

Starting with Flask

Assuming that you have pip (the modern Python installation program) on your computer, you can install Flask with:

```
pip install flask
```

(Depending on the permissions of your computer, you might need to install the above as root.)

With that package (and its dependencies) in place, you're ready to start developing. Create a new Python program that contains the following:

```
#!/usr/bin/env python

from flask import Flask
app = Flask(__name__)

@app.route("/")
def foo():
    return "Hello, world...!"

app.run(debug=True)
```

Let's go through this program (which I have called simple.py), line by line, to see what it does before

Listing 1. simple.py

```
#!/usr/bin/env python

from flask import Flask, render_template, request
app = Flask(__name__)

@app.route("/blah")
@app.route("/")
def foo():
    return render_template('foo.html')

@app.route("/submit", methods=["POST"])
def submit():
    username = request.form['name']
    return "Thank you for submitting a form, %s." % username

app.run(debug=True)
```

Listing 2. foo.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>Hello!</title>
    </head>
    <body>
        <h1>Hello!</h1>
        <p>Test paragraph</p>

        <form method="POST" action="/submit">
            <p>Name: <input type="text" name="name" /></p>
            <p><input type="submit" value="Send your name" /></p>
        </form>

    </body>
</html>
```

you run it. The first line is actually not surprising to anyone with Python experience; it simply means that you want to open the “flask” module or package and put the “Flask” class defined there in the current namespace. With that done, you now can create an instance of Flask, which you do using the built-in `__name__`, which, of course, will be the string `__main__` on all programs run directly from the command line.

Now the real magic begins, with the line starting with `@app.route`. The `@` at the start of the lines indicates that this is a decorator. This isn’t the appropriate place to go into detail on what decorators do,

but suffice it to say that this allows `app.route` to execute both before the `foo()` function is defined, and also each time it is executed. Routes are just one place in Flask that use decorators in this way; you also can use them to ensure that certain actions happen before or after others.

You also can use multiple decorators on a single function definition. Thus, if you (for whatever reason) want two different URLs to invoke the same function, just stack the decorators:

```
@app.route("/foo")
@app.route("/")
def foo():
    return "Hello, world!"
```

Next, you can see that the function “`foo`” takes no parameters and is a normal Python function. The only unusual thing about this function is that you don’t invoke it directly. Rather, the Flask framework invokes your function for you, based on the URL to which the user has navigated. So when the user goes to “`/`”, which has been registered with `app.route`, Flask knows to invoke this function, and it does so. The string that is returned by the function is then returned to the user’s browser.

Finally, you tell your application,

which you created with the call to `Flask(__name__)` at the top of the file, that it should run. You could do that just by invoking `app.run()`, but by passing `debug=True` as a keyword parameter, you gain a number of things, including automatic reloading of files without restarting the server, and a browser-based debugger and console if and when things go wrong.

If you now invoke `simple.py`:

```
$ ./simple.py
* Running on http://127.0.0.1:5000/
* Restarting with reloader
```

the server is waiting on port 5000. You can point your browser to `http://localhost:5000` and immediately see “Hello, world!” It’s not the fanciest of responses, but not a bad one.

I’m probably jumping the gun a bit, but the Web-based console is pretty snazzy, and I think it really can help in debugging. Instead of the “`return`” line shown above, in which you return a string, replace it with the line:

```
return "a" + 5
```

No, this line isn’t valid Python, but that’s just the point. You’re deliberately forcing an error. Now, go back to the “`/`” URL in your browser,

and you should see a long stack trace with error messages. So far, that's not too exciting. But if you move your mouse cursor to the right side of the darker lines (that is, those containing source code), you'll see two icons appear. The rightmost icon (of a text file) will display the Python source code that was executing when the error occurred.

Far more powerful is what happens if you click on the darker (terminal-like) icon. A full-fledged Python interpreter and console opens up, containing the variables and functions that were defined at that point in the stack. You can open a console for any point in the stack trace, explore the stack frame and its variables, and figure out what went wrong in your program.

Simple Templates

Returning strings of text, or even HTML, from within a function is good for a simple demo, but quickly becomes tedious. Flask's JinJa2 templates support not only HTML, but also a Python-like syntax that you can embed inside the templates. For now though, let's ignore JinJa2's capabilities and create a simple form you can submit.

First and foremost, if you are going to use JinJa2 templates, you

need to create a subdirectory called "templates". This subdirectory should exist in the same directory as simple.py. Inside this templates directory, let's create a very simple HTML file:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello!</title>
  </head>
  <body>
    <h1>Hello!</h1>
    <p>Test paragraph</p>
  </body>
</html>
```

As you can see, this template won't do much. But the fact that it's a full-fledged HTML file means you suddenly can have all of the text, CSS, links and JavaScript that you want, without having to stuff it inside a triple-quoted string in your function.

As things currently stand, the template (in templates/foo.html) isn't going to be called by your "foo" function. You need to change the function such that it invokes the template. You do this with:

```
@app.route("/foo")
@app.route("/")
def foo():
    return render_template('index.html')
```

The `render_template` function is defined in the `flask` module, which means you'll need to change your import statement at the top of the program too:

```
from flask import Flask, render_template
```

Once you've done that, you can reload the page, and—*voilà!*—the template is rendered, as you hoped.

Handling Forms

It's pretty typical for sites to have one or more HTML forms. You've already seen just about everything you need in order for Flask to process forms, believe it or not. All you need to do now is modify your template so that it contains an HTML form, write a function that is invoked with POST, and then grab the submitted form element and do something with it. Each of these three steps is fairly easy with Flask.

First, add an HTML form to your template:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello!</title>
  </head>
  <body>
    <h1>Hello!</h1>
```

```
<p>Test paragraph</p>
```

```
<form method="POST" action="/submit">
  <p>Name: <input type="text" name="name" /></p>
  <p><input type="submit" value="Send your name" /></p>
</form>

</body>
</html>
```

This indicates that the form will be submitted using the POST method to the “/submit” URL. In addition to a “submit” button, the form consists of a single text field, called “name”. If you click on the “submit” button, you get an error message:

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

That's not a surprise, given that you haven't defined a route for it, let alone a function. Let's add that to `simple.py`:

```
@app.route("/submit")
def submit():
    return "Thank you for submitting a form."
```

If you go back to “/” and enter your name, then click on “submit”, you get

the following error:

Method Not Allowed

The method is not allowed for the requested URL.

Notice that the problem here isn't that Flask doesn't recognize the route. Rather, the route doesn't know how to handle a POST request. That's because routes in Flask are assumed to handle GET, unless you specify otherwise. You can do that by passing the "methods" parameter to your route, specifying a list of methods

(as strings) that are acceptable:

```
@app.route("/submit", methods=["POST"])
def submit():
    return "Thank you for submitting a form."
```

Sure enough, if you submit the form, you get the text back. But this text is rather generic. It would be nice to acknowledge the user's name, given that he or she went through the trouble of providing it. You can grab the user's name, as well as any other form parameters, via the "request"

LINUX JOURNAL

on your
Android device

Download the app now
in the **Android Marketplace**



www.linuxjournal.com/android

For more information about advertising opportunities within *Linux Journal* iPhone, iPad and Android apps, contact John Grogan at +1-713-344-1956 x2 or ads@linuxjournal.com.

object that Flask makes available to you. `request.form` is a dictionary-like object that lets you query the form via key names (as strings). You need to import “`request`” from the “`flask`” package:

```
from flask import Flask, render_template, request
```

And, then you can do this:

```
@app.route("/submit", methods=["POST"])

def submit():
    username = request.form['name']

    return "Thank you for submitting a form, %s." % username
```

If you’re thinking this all seems very simple—well, that’s precisely the point. Flask is there to let you run ahead quickly using the Python you already know to create simple but interesting Web applications.

Conclusion

If you are familiar with Python, want to create Web applications and don’t want the overhead of a large framework like Django, you might well want to consider Flask. The core framework is (as you saw here) easy to get up and running, and the extensions make it extremely flexible and powerful. ■

Reuven M. Lerner, a longtime Web developer, consultant and trainer, is completing his PhD in learning sciences at Northwestern University. You can learn about his on-line programming courses, subscribe to his newsletter or contact him at <http://lerner.co.il>.



Send comments or feedback via
<http://www.linuxjournal.com/contact>
or to ljeditor@linuxjournal.com.

Resources

The Flask home page is at <http://flask.pocoo.org>. This includes links to the code, documentation, tutorials, examples and an official list of Flask extensions. The documentation is excellent, and it should provide anyone with even a bit of Python knowledge with good understanding of Flask.

Miguel Grinberg, who recently authored a book about Flask for O’Reilly (which I haven’t yet had a chance to read or review), has written an excellent Flask “mega-tutorial” that complements the official one very nicely:

<http://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world>.

A video of Miguel’s tutorial at PyCon 2014 is available at
<https://www.youtube.com/watch?v=FGrlvBDQLPg>.



DAVE TAYLOR

Days Between Dates?

Date math—it's not so easy, but Dave jumps in to create the utility `DaysAgo.sh`.

Alert readers will know that I'm working on a major revision to my popular *Wicked Cool Shell Scripts* book to come out later this year. Although most of the scripts in this now ten-year-old book still are current and valuable, a few definitely are obsolete or have been supplanted by new technology or utilities. No worries—that's why I'm doing the update.

One script I'll be adding is a complicated one that I'm going to develop here in my *Linux Journal* column: daysago. The script will take a specified date in the past and tell you how many days have elapsed between that date and the current day and time.

You might be thinking that's fairly complicated, and it is, but not in the way you might be thinking. The actual calculation is really easy because of how Linux systems store and manipulate dates. The challenge is in parsing the input.

The first part of the book includes a library of useful scripting utilities, however, and one just so happens to be what we want—no coincidence that!

Valid Date?

The easiest way to deal with something as complicated as a date is to force the work onto the user. There are a couple different strategies for that, but let's be lazy for now and prompt the user for the month, then day, then year, requiring numeric values. Then, we'll need to check whether it's valid.

Validating a user-specified date is pretty straightforward until we get to the issue of leap years. We're used to thinking that every four years is a leap year, but the formula is quite a bit more complicated than that, and it can be summarized with this set of rules:

- Years divisible by four are leap

years, unless...

- The year also is divisible by 100, except if...
- The year is divisible by 400, in which case it is.

Is that complicated enough? Of course, if we're just looking at leap years in the last few decades, it's not a very big deal, but it's inevitable that someone will try something like Feb 29, 1776, in which case we need to know whether it's valid.

Or, we can be lazy.

Since I like the lazy solution to things (remember, I'm not writing production code here, I'm demonstrating concepts), let's cheat by using the Linux `cal` command. Because it lets us specify month/year, we can hand off the question of whether there's a February 29 in the year 1776 by just asking for a display of 2/1776:

```
$ cal 2 1776
  February 1776
Su Mo Tu We Th Fr Sa
                  1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29
```

It looks like 1776 was indeed a leap year. No wonder they had time to draft the Constitution before summer came along and made Philly too darn hot for anyone to work!

To turn this command into a script, a simple `grep` and test for nonzero results does the trick:

```
mon=$1; day=$2; year=$3

if [ $mon -eq 2 -a $day -eq 29 ] ; then
    echo checking for feb 29 : was $3 a leap year?
    leapyear=$(cal 2 $year | grep '29')
    if [ ! -z "$leapyear" ] ; then
        echo "Yes, $year was a leapyear, so February 29, $year \
is a valid date."
    else
        echo "Oops, $year wasn't a leapyear, so February only \
had 28 days."
    fi
fi
```

Let's run a few quick tests to see what happens:

```
$ sh valid-date.sh 2 29 1777
checking for feb 29 : was 1777 a leap year?
Oops, 1777 wasn't a leapyear, so February only had 28 days.

$ sh valid-date.sh 2 29 1776
checking for feb 29 : was 1776 a leap year?
Yes, 1776 was a leapyear, so February 29, 1776 is a valid date.
```

That makes sense, and it's sure easy to use `cal` for this particular test.

GNU date

If you have GNU date on your system (try `date --version`), the latter part of this scripting project becomes crazy easy, because you easily can calculate the number of seconds between Jan 1, 1970 and the specified date subsequent. For example:

```
date '+%s' -d 2011-11-04
```

It's easy to subtract one date from another and divide by 86400 to convert seconds to days—for dates after Jan 1, 1970.

We still need to encapsulate the “30 days have September, April, June and November” information too, and that's easily done with a rather compact case statement:

```
case $mon in
  1|3|5|7|8|10|12 ) dim=31 ;# most common value
  4|6|9|11      ) dim=30 ;;
  2                  ) dim=29 ;# possible leap year?
  *                  ) dim=-1 ;# unknown month
esac
```

In this case, the variable we're setting is “days in month” or `dim` (not a reference to *A Clockwork Orange*, my cineophile readers). This makes it easy to check all but Feb 29 as a possible date, as demonstrated in this simple conditional:

```
if [ $day -lt 0 -o $day -gt $dim ] ; then
  echo "Invalid date: Month #$mon has $dim days, so day \
$day is impossible."
fi
```

There are a bunch of different ways to do this, of course, but because most months have 31 days, again, I'm looking for the shortcut!

Mixed together and slightly tweaking the output, we now can test the validity of any date specified in the correct month, day, year format:

```
$ sh valid-date.sh 2 29 2013
The date you specified -- 2-29-2013 -- is valid. Continuing...
$ sh valid-date.sh 1 33 2013
Invalid date: Month #1 has 31 days, so day 33 is impossible.
$ sh valid-date.sh 2 29 2013
2013 wasn't a leapyear, so February only had 28 days.
```

Ahh, that all works just fine. We started out by deciding that all the date formatting issues were going to be pushed to the user, but we still need to do some rudimentary tests, at least this one:

```

if [ $# -ne 3 ] ; then
    echo "Usage: $(basename $0) mon day year"
    echo " with just numerical values (ex: 7 7 1776)"
    exit 1
fi

```

Yes, this month the script isn't glamorous—such is the life of a scripter.

With a valid date, there's a tendency to use something like GNU date to do the math (see the GNU date sidebar), but that has some inherent limitations, not the least of which is that it won't work with any dates prior to 1970.

I'll stop here for this month, but next month, we'll take the date we've validated and see if there's a formula to count the number of days quickly from then to now! ■

Dave Taylor has been hacking shell scripts for more than 30 years. Really. He's the author of the popular *Wicked Cool Shell Scripts* and can be found on Twitter as @DaveTaylor and more generally at his tech site <http://www.AskDaveTaylor.com>.

||||||||||||||||||||||||||||||
Send comments or feedback via
[**http://www.linuxjournal.com/contact**](http://www.linuxjournal.com/contact)
or to [**ljeditor@linuxjournal.com**](mailto:ljeditor@linuxjournal.com).

LINUX JOURNAL

now available
for the iPad and
iPhone at the
App Store.



Available on the
App Store

linuxjournal.com/ios



For more information about advertising opportunities within *Linux Journal* iPhone, iPad and Android apps, contact John Grogan at +1-713-344-1956 x2 or ads@linuxjournal.com.



KYLE RANKIN

The Only Mac I Use

Mac? Only if it's a vim macro.

Okay, so the title is a bit of a troll.

Although people are, of course, free to use whatever computers they want, I've personally never liked Macs. I've always found it strange how many Linux advocates rail against Microsoft, but hold their tongues when Apple does the same things. In any case, this isn't an article about that—it's actually about vim macros, because a vim macro is about as close as I'll get to a Mac—or Emacs, for that matter. Hey, that makes two holy wars in the first paragraph—not bad.

It's no secret to frequent readers of my column that I use vim as my editor. I've even written a series of columns in the past about applications that use vi-style key bindings for navigation. In this article though, I'm going to highlight one of my favorite time-saving features of vim since I discovered you could use Ctrl-[instead of Esc: macros.

Like with a lot of things I've learned through the years, it took having an

almost insurmountable problem for me to learn how to use vim macros effectively and commit the vim macro syntax to memory. Quite a few years ago, I was an administrator for a DNS system that served thousands of zones. We were in the middle of a project to migrate to a new data center, so that meant that tens of changes needed to be made to thousands of DNS zone files to point them to the new site. Fortunately, we had time to stage the changes so this wasn't an all-or-nothing endeavor. The trick though was that the zone files had been created through the years by different administrators who each had their own sense of style. This meant I couldn't just write a script to do generic search and replace for me. Beyond that, the stakes were high enough that I couldn't afford a bug in a script wreaking subtle, silent destruction on zone files.

At first, I just hit this massive assignment head-on, by hand, and

Essentially, a vim macro allows you to record keystrokes, assign them to a particular key on the keyboard, and then play them back one or more times later.

we decided on 20-zone batches. I'd open a zone file in vim, make my changes and then save them. What I noticed after a few rounds of this was that in many cases I was doing the same exact keystrokes in the files. I'd go to the top of the file, update the serial number, then perform a few basic search-and-replace commands on some IPs and on some record names. Although all the zones weren't uniform, I discovered that there were only maybe three or four different variations on my commands. At this point, I decided to research how to use vim macros and discovered I could save only a handful of macros and reduce my editing time from 30 seconds or a minute per zone file with many keystrokes to a few seconds and only a few keystrokes.

Essentially, a vim macro allows you to record keystrokes, assign them to a particular key on the keyboard, and then play them back one or more times later. If you already are efficient

with your vim keystrokes (and by that I mean reducing your reliance on hjkl for navigation and using things like ^ and \$ to go to the beginning and end of a line, w and b to skip forward and backward a word and searching for words to move to them), once you realize you are making a lot of repetitive keystrokes to edit a file, you'll know it's time to record a quick macro and save yourself time.

Personally, I've found macros particularly useful when editing DNS zone files because in the past I've needed to add, say, 50–100 new A records to a zone, with the only difference being that the hostname and IP address kept incrementing by one. It's this kind of problem I'm going to use as an example here to show you how to use macros. Let's say you have a zone file and need to add 50 A records starting with worker1 pointing to 10.9.0.15 and ultimately ending with worker50 pointing to 10.9.0.64. By hand, this is the kind of

mundane task that would drive you crazy and burn an hour, but with vim macros, it takes only a few seconds.

The great thing about this particular problem that makes it well suited for vim macros is the fact that each line simply copies the previous line and increments two numbers. First, edit the file and add the first line by hand:

worker1 IN A 10.9.0.15

If you were to add the worker2 record, you might go one of two ways about it. Although you could just type everything by hand, you probably would type yyp to yank the worker1 line and paste it below. Next you might press e to go to the end of worker1 word, press r2 to replace the 1 with a 2, then type \$ to go to the end of the line and type r6 to replace the 5 with a 6. That works, but it's difficult to repeat over and over, because the number changes each time. A better approach is to type yyp to copy and paste the worker1 line, and because the cursor is now over the worker1 word on the second line, press Ctrl-a, which in vim increments a number. Then, type \$ to go to the end of the line and type Ctrl-a again to increment the 15 to a 16.

This approach that uses Ctrl-a is how you will tackle the macro. So

with only the single worker1 line present and your cursor anywhere on that line, type q to trigger macro recording mode. Then press a key you want to assign this macro to, such as the a key. Your vim screen now will say the word recording in the bottom-left corner. Now that you are in recording mode, any keystrokes you make will be recorded until you press the q key again. Once in recording mode, type yyp to copy and paste the first line. The cursor is actually in a good position here at the beginning of the line, but if you aren't certain that an operation will always put your cursor in the right spot, it never hurts to throw in a ^ key to ensure that the cursor is at the beginning of the line. Next press Ctrl-a to increment worker1 so it becomes worker2. Now type \$ to go to the end of the line, and press Ctrl-a again to increment the 15 to a 16. Finally, type q to exit recording mode.

Now you have a macro assigned to the a key. To replay your macro, make sure your cursor is now somewhere on the worker2 line, then press @a. The @ key tells vim to replay the macro assigned to whatever key you press next, in this case, a. You should see a third line below worker2 for the worker3 A record. This section of the file should now look like this:

```
worker1 IN A 10.9.0.15
worker2 IN A 10.9.0.16
worker3 IN A 10.9.0.17
```

Although you could just press @a 47 more times, vim allows you to preface just about any command with a number, and it will perform that command that many times. So, to finish up this file, make sure your cursor is somewhere on the worker3 line and press 47@a to see the remaining lines appear.

That's it. Now that you have a basic example of the power of macros, the next time you find yourself performing

a repetitive task in a text file, see if you can assign it to a temporary macro. This article just scratches the surface of macros, so in a future column, I'll follow up with more macro tips including how and why to nest macros. ■

Kyle Rankin is a Sr. Systems Administrator in the San Francisco Bay Area and the author of a number of books, including *The Official Ubuntu Server Book*, *Knoppix Hacks* and *Ubuntu Hacks*. He is currently the president of the North Bay Linux Users' Group.

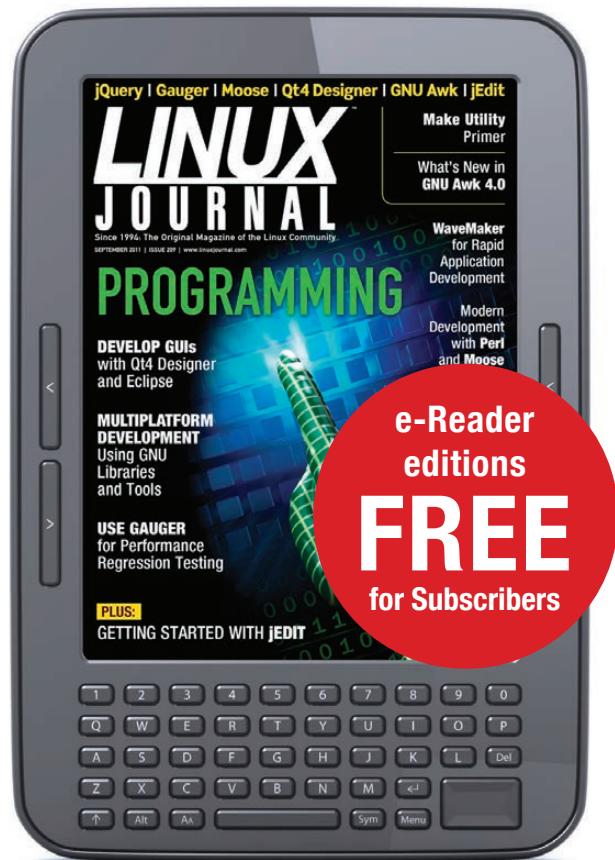
||||||||||||||||||||||||||||||
Send comments or feedback via
<http://www.linuxjournal.com/contact>
or to ljeditor@linuxjournal.com.

LINUX JOURNAL

on your e-Reader

Customized
Kindle and Nook
editions
now available

LEARN MORE





SHAWN POWERS

Healthy Robots

Getting healthy isn't much fun, but being a healthy cyborg? Awesome.

Contrary to many people's beliefs, I'm not really a robot. Based on my constant use of technology, however, it might be arguable that I'm a cyborg—or at least pre-cyborg, if that's a thing. In this article, I'm going to talk about the health-related technology I use as a Linux user. Some of the things are still hard to do in a completely Linux environment, but most work fine regardless of what's running under the keyboard. Why focus on health? I'm glad you asked.

The Single Paragraph of Why

First off, summer is here for the Northern Hemisphere, and that usually means we all want a sexy beach body—or at the very least we are willing to crawl out of our insulated caves and get outside. For me, that's especially true after the winter we've had here in the Midwest. On a personal note, however, I recently found out I have some serious health problems (kidney- and

liver-related), so a healthy lifestyle is crucial. I plan to get very, very old someday. If I'm going to be healthy, my best chance of success is if I can do it with lots of gadgets. So, here we go!

Hardware: Weight for It...

Although there are plenty of aspects to a healthy lifestyle, I'm going to focus on diet and exercise. Basically, what I put into my body, what I do to make my body sweat and the precise measurement of gravitational attraction between my body and the Earth. That gravitational attraction (or "weight" for the mundane) certainly isn't the only measurement of health, but whether we like to admit it or not, a healthy weight is extremely beneficial.

I like data, but I hate recording it. That's one of the reasons I purchased a Withings Wi-Fi-enabled digital scale. There currently are two companies selling self-uploading scales that I know of: Withings and Fitbit. I can't compare

A quick Google search tells me there is now a native Linux setup program for the Withings scale, which makes me even more happy to recommend it!

the actual brands, as I've used only the Withings, but I can tell you firsthand that what seems frivolous and silly (namely a Wi-Fi-enabled bathroom scale) is incredibly convenient and useful. Both scales automatically upload and graph your weight over time. The Withings scale supports multiple users, which it detects based on weight, and it has been an investment I've never



Figure 1. Stop looking at my oddly-tanned feet! The scale underneath is the focus of the article. It tracks weight, BMI and fat percentage.

regretted. (Note: The Fitbit Aria scale may support multiple users as well, but I don't have one, so I can't say.)

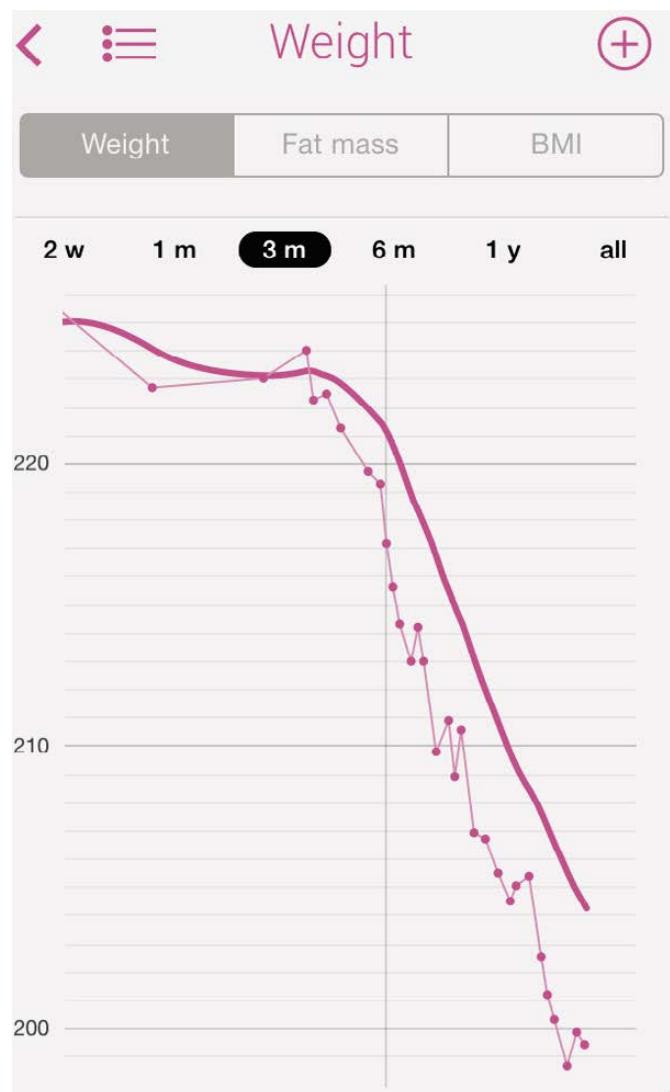


Figure 2. It's easy to tell on the graph when I started dieting. Go, go, go!

The only gotcha with the Withings scale is that when I purchased mine years ago, there was no Linux software for configuring the Wi-Fi settings. The settings are stored in the scale, even across battery changes, so I've never had to worry about it since that initial setup. (Yes, my Wi-Fi password is the same as it's always been. If you visit my house, it likely still will be "bigredchicken"—you're welcome.) A quick Google search tells me there is now a native Linux setup program for the Withings scale, which makes me even more happy to recommend it! The graphing is all done on-line with a free account (see Figures 1 and 2), but for the brave and adventurous, there is an API to connect to the Withings site. If you're a fan of keeping your own data, it's possible, albeit complicated, to arrange such a thing.

Hardware: Hut! 2, 3, 4...

The other main bit of hardware I use is a pedometer. Wearing a pedometer won't make you skinny, but knowing exactly how much moving around you do in a day is invaluable. I've found that days I thought I was pretty good about moving around the house were actually pathetically low in step count. And then days I hadn't considered particularly active have tracked miles and miles of mindless walking I hadn't

even realized I'd done. The point is, a reliable pedometer offers incredibly useful insight on your daily activity. But, who wants to keep track of all that data? Again, I want the data, but I don't want to write it down. Enter: Fitbit and the like.

I've had a half-dozen or so Fitbit pedometers during the past few years. It's not necessarily the most affordable, or even most accurate pedometer, but it's very popular. The on-line component alone (see the software section later in this article) might make it worth the expense. Thankfully, I have a dear friend, Michelle Klishis, who actually *does* like to enter data. She has done a very extensive test of pedometers, and if there's anyone whose data I trust, it's Michelle's. If you don't have a pedometer, I urge you to read her extensive review of the various options: <http://klishis.com/notreally/archives/11476> (see Figure 3 for one of many graphs).

Long story short, I currently have a Fitbit One. It slips in that useless little pocket in my jeans (what is that pocket for anyway?) and requires only weekly charging. The biggest danger with using a Fitbit is that it's easy to forget it's in your pocket. They don't do well in washing machines. The folks at company, however, do

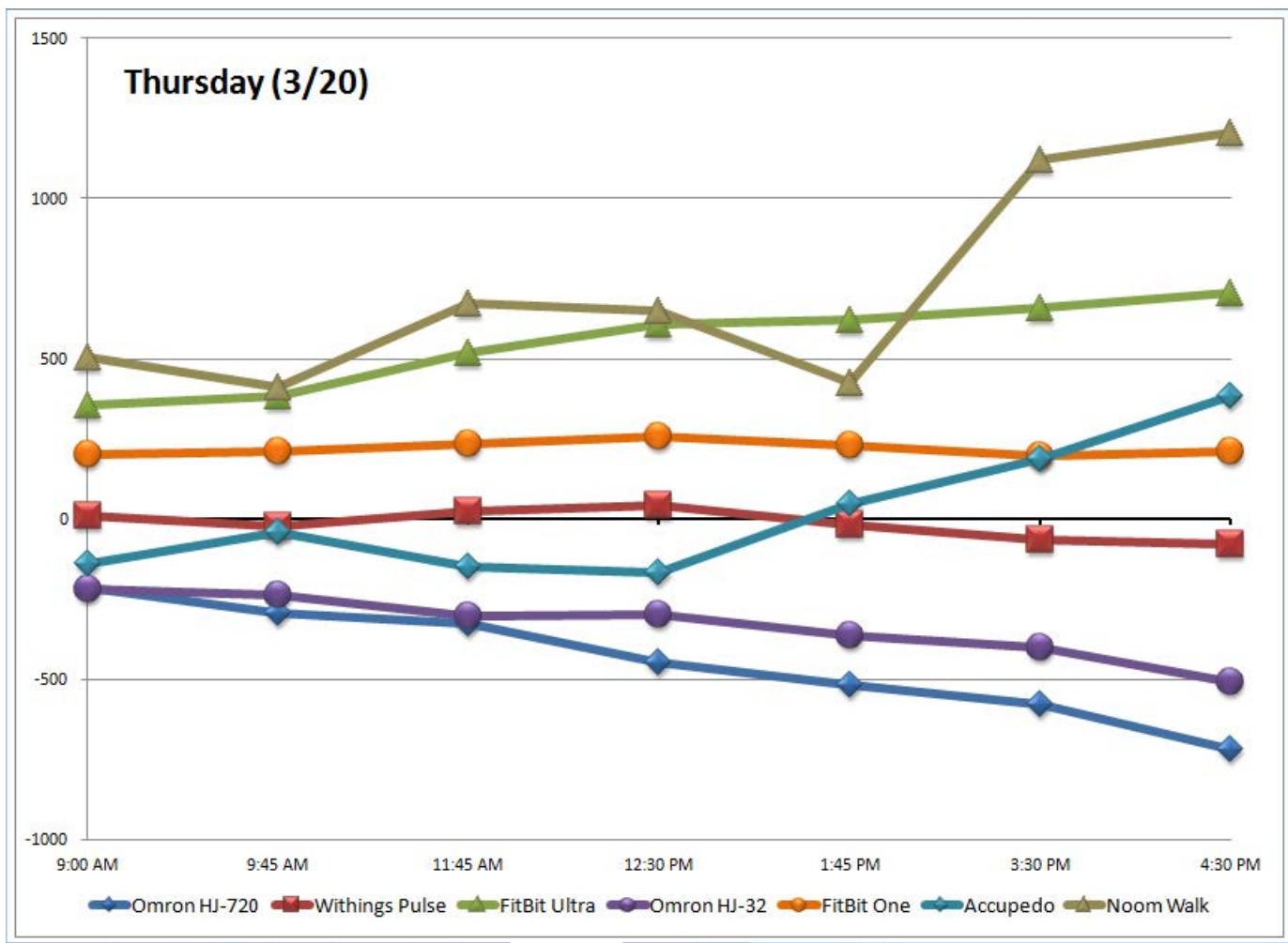


Figure 3. Deviation from average steps per hourly measurements? Yes. If you want pedometer data, check out Michelle's extensive reviews.

amazing customer service. I lost a Fitbit Ultra tracker in the park, and they sent me a free replacement, even though it was completely my fault. Really, it's a great company. BUT.

Up until the release of the Fitbit One (the newest pedometer at the time of this writing), the only way to sync the data was to use a USB dock that worked only with Windows and OS X. Feature-wise, the Fitbit is

amazing as it tracks steps, calories, miles and even altitude. On the Linux front, however, it's not all that great. I'm happy to say, with the newer tracking devices, syncing is done over Bluetooth to an Android (or iOS) device. Windows or OS X are no longer required! I shamefully used to run an old install of Windows XP in my office for the sole purpose of syncing my Fitbit. Now, that's not

an issue anymore. Plus, with phone-syncing, the data is transmitted whether or not you're at home near your computer. I love my Fitbit, I really do.

Hardware: Alternatives

I'm telling you specifically what I use because I know it works well, and it fits my particular lifestyle. Please don't take that as an endorsement of one product over another. Check out Michelle's pedometer information. Read reviews of alternate digital scales. Heck, go check them out at the store and see which one you like better!

In fact, newer cell phones are getting to the point that they can be remarkably accurate pedometers on their own. One of the pedometer apps available in the Google Play store might suit your needs just fine. The point is to count your steps and keep track of your weight—not to get caught up in brand preference or feature creep. That said, we're geeks, and digital bells and whistles tend to be attractive!

Also, although I don't have one, I really like the concept of the Withings blood pressure monitor. It syncs up over Bluetooth and records blood pressure measurements and dates automatically. Yes, it's on my

wish list. For now, I use an Omron blood pressure monitor and enter the data into the Withings site manually—like a barbarian.

Software

The software is really where the magic happens with connected-health. It's great to have the Withings site track your weight and the Fitbit site track your steps, but without data integration, it's not a whole lot better than pencil and paper (or more realistically, a spreadsheet). Thankfully, most of the popular health-related Web sites allow for third-party integration. They don't all share everything with each other, but it seems like more and more sites are offering integration. It's pretty awesome. The following are some of the sites and apps I use regularly.

MyFitnessPal: This is the app/Web site I use the most. It has a social aspect for connecting with friends and getting encouragement, but for me, the best feature is the integration with Fitbit. MyFitnessPal (MFP for short) has an incredible food database for tracking what you eat during the day. If you happen to go over your calories for the day, its integration with Fitbit means you can go for a long walk and earn more calories. Basically, after a minimal expected number of steps,

MFP will give you more calories to consume based on how far you've walked. Because it's all synced and shared automatically, you just need to check the MFP app periodically on your phone or Web site and see how many calories you have left to consume. If you know you're going to have a huge dessert after dinner, you can take a really long walk early in the day and "stock up" on calories! It's truly amazing, and it's a healthy way to make sure you don't overeat. MFP also allows you to enter other exercise activities, along with the time you performed them, and it will grant you the appropriate earned-back calories for those activities. Since you enter a time, it makes sure not to "double dip" by also giving you calories for Fitbit activity during that time frame. So if you take a two-hour bike ride, you can tell MFP about it and earn some serious calories, even though a pedometer doesn't do a great job at tracking cycling. I love MyFitnessPal. Oh, and weight? Yeah, MFP syncs with Withings to keep track of your weight loss automatically as well. It's truly an amazing site, and it's completely free: <http://www.myfitnesspal.com>.

RunKeeper: Due to my recent kidney problems, I'm not able to do any strenuous exercise. That said, I can walk. RunKeeper is an incredible

GPS-enabled app for tracking running, walking, biking and so on. It will track your progress, pace and any number of activity-related data items. Much like the Fitbit, RunKeeper will integrate with MyFitnessPal. I haven't been able to try the integration yet, but if you do, be sure to check for conflicts. It would be a shame to "double dip" and get both Fitbit and RunKeeper calorie credit. If I'm able to start running again in the future, this will be something I'll have to watch closely to see how MFP handles such things. When it comes to tracking exercise, however, it's hard to beat RunKeeper. There's a free and paid version in the Google Play Store, along with a robust free Web site.

Withings Health: I normally log in to MyFitnessPal only on a daily basis. I do this to enter food I eat and also to check on my weight loss progress. Unfortunately, at the time of this writing, MFP doesn't have a great way to track blood pressure. Remember earlier I mentioned how much I want the Withings blood pressure monitor? Well, until that day comes, I enter the information manually into the Withings site. It does a great job of creating graphs, and since blood pressure is such a critical issue for me (kidney problems), that graph is invaluable

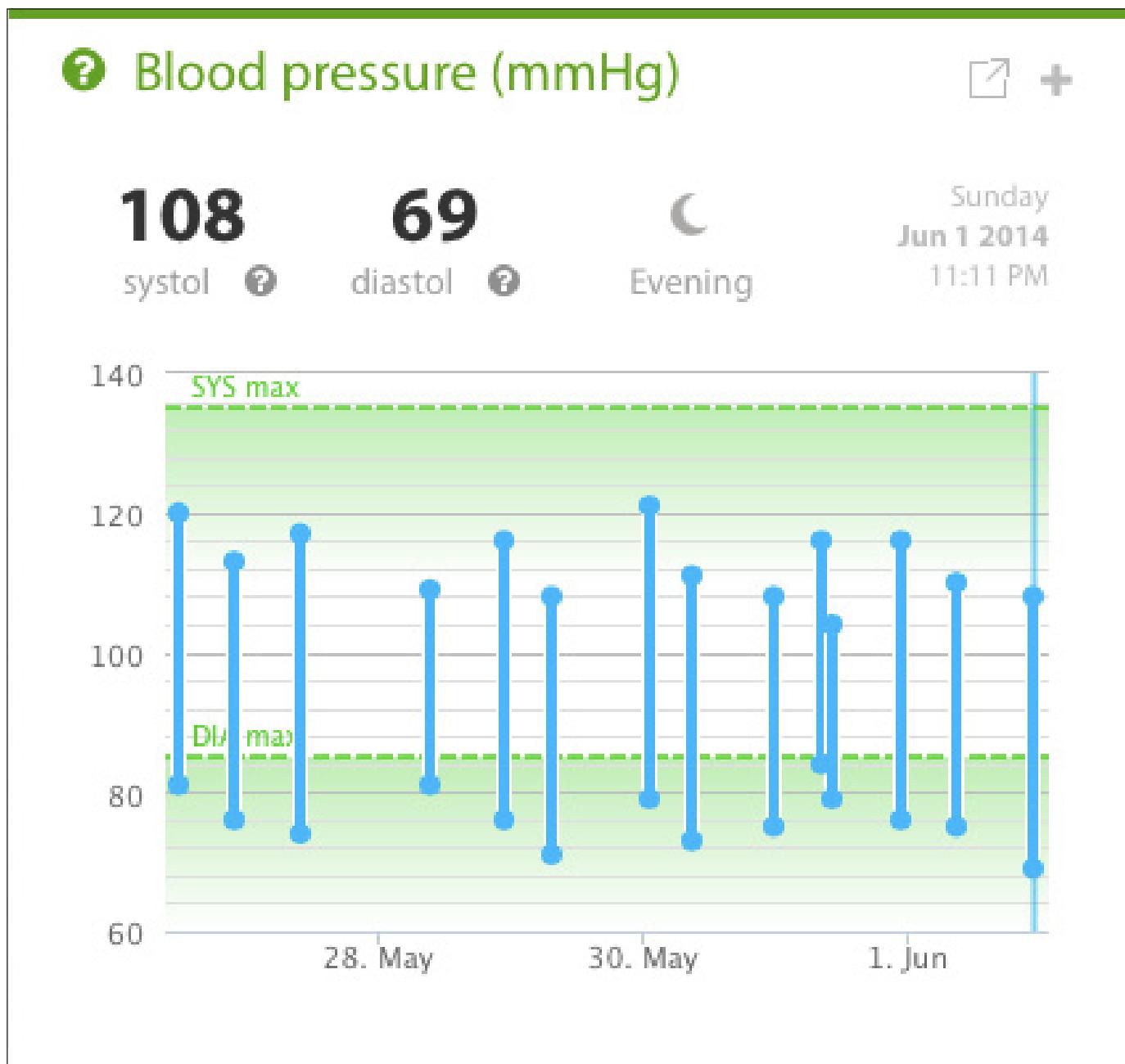


Figure 4. Green is good. My blood pressure has been pretty stable, and if anything, is getting a bit low. Soon I'll need to quit taking my blood pressure medicine! Woot!

(Figure 4). The weight-loss chart is nice too, but honestly, I really don't look at it much. MFP shows current weight lost when you log in to the site, so that's what I tend to look at.

Fitbit Site: I love the Fitbit, I really do, but I don't very often log in to the Web site. This is unfortunate, because it has a great social feature that shows you and your groups

and friends in comparison. If you're the competitive sort, it's easy to see which of your friends is "beating" you and up the pace in order to win. I'm not particularly competitive, so there's no motivation for me there, but it's nice to keep an eye on your friends and send encouragement if they seem to be stagnant. Fitbit does offer a premium membership that offers some features in addition to the free stuff, but I haven't ever been tempted to upgrade, because any features I want come with the free account. (I don't feel bad about this, because like I said, I've purchased at least a half-dozen Fitbit trackers during the past few years for myself and family members.) If you're not into tracking food intake or weight, however, the Fitbit site would make a nice entry point for tracking stuff. I think it has the nicest social/friend system, but since I want integration with all my devices, it's not the site I default to. Check it out at <http://www.fitbit.com>.

Oh the Places We'll Go

Why did I share all that information on the health stuff I do? One, I want to share the incredible ways technology can help in living a healthy lifestyle. Two, I'd love some accountability. Hit

me up on MyFitnessPal or the Fitbit site. Perhaps we can start a *Linux Journal* group on the Fitbit site for anyone interested in such a thing. The unfortunate fact is that many of us have very sedentary jobs, and we tend to eat poorly. At least, that has been the case for me. I've been forced to take a hard look at my health, and I figure as geeks, we might all benefit!

I'm going to start a Web-based "Lifestyle" column on the *Linux Journal* Web site. I hope to update everyone on new gadgets, features, accomplishments, ideas and perhaps even geeky recipes for healthy eating. If you're interested in taking this adventure with me, be sure to watch <http://www.linuxjournal.com> for my future posts. Like I said earlier, I want to live to get really, really old someday, and if I get to be part cyborg too? Bonus! ■

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

|||||
Send comments or feedback via
[**http://www.linuxjournal.com/contact**](http://www.linuxjournal.com/contact)
or to ljeditor@linuxjournal.com.

CloudGenix's Software-Defined Enterprise WANs

The fledgling company CloudGenix recently emerged from “stealth mode” to reveal Software-Defined Enterprise WAN (SDEwan), a new category of solutions for wide area networking (WAN) and remote office infrastructure for the hybrid cloud and mobile era. SDEwan virtualizes enterprise networks and securely brings together users and high-performance cloud and data-center applications while radically reducing remote office infrastructure requirements. CloudGenix says that its raison d'être is based on the fact that organizations are seeking to manage the complexity that has arisen with the rapid adoption of cloud applications and services, as well as hybrid networks within the enterprise. Enterprises need solutions that not only simplify application access, but also provide innovative ways of delivering network services to distributed enterprise IT assets and infrastructure, adds the company.

<http://www.cloudgenix.com>

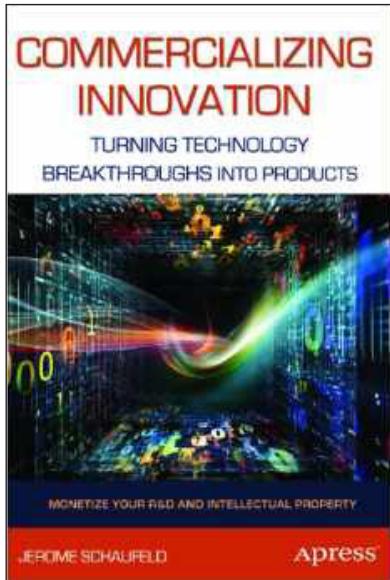
cloudgenix [SDEwan]

WIND RIVER

Wind River's Security Profile for Wind River Linux

While security-related vulnerabilities grow as more devices become connected and intelligent, Wind River is releasing targeted responses in the form of the new Security Profile for Wind River Linux. Wind River describes Security Profile as a high value add-on software profile for Wind River Linux that delivers new security capabilities to help developers combat against future threats arising from the growth of the Internet of Things. The software offering is certifiable to the Common Criteria General Purpose Operating System (GPOS) Protection Profile up to Evaluation Assurance Level 4 (EAL 4). Key features—including a hardened kernel, secure boot, a security-focused user space and the Yocto Project Compatible Wind River Linux base—seamlessly integrate with validation tools, documentation and hardware support.

<http://www.windriver.com>



Jerome Schaufeld's *Commercializing Innovation* (Apress)

Commercializing technology is not easy. There are many complicated decisions to make. Which ideas do you pursue? Whom do you hire? Where do you manufacture? Where do you get the capital? Help in answering these and myriad others can be found in the new book *Commercializing Innovation: Turning Technology Breakthroughs into Products* by Jerome Schaufeld. In the book, Schaufeld—a successful technology entrepreneur and professor of entrepreneurship—

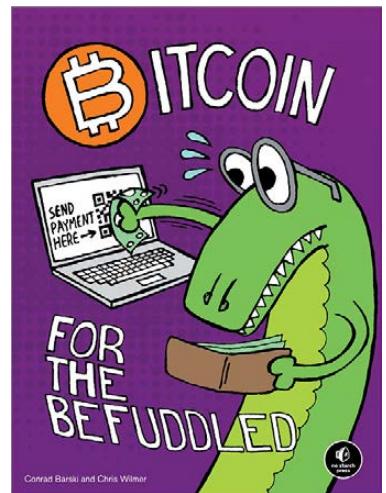
offers a step-by-step commercialization process that begins with assessing technology from a variety of sources and ends with taking viable products into the market. Schaufeld's systematic approach, complemented by numerous case studies and models, combines both the theoretical and practical aspects of bringing products to market.

<http://www.apress.com>

Conrad Barski and Chris Wilmer's *Bitcoin for the Befuddled* (No Starch Press)

When you see a Craigslist "roommate wanted" listing that says "must be into Bitcoin", you know it's one of our tribe. In that geek pad, you also will be likely to encounter Conrad Barski and Chris Wilmer's book *Bitcoin for the Befuddled*, a new, fun and patient introduction to the topic of Bitcoin. The title explores Bitcoin's origins, as well as what it is, how it works, and how to acquire, store and spend Bitcoins safely and securely. Other topics include Bitcoin's underlying cryptographic principles, how new Bitcoins are created, the currency's potential future impact on trade and commerce, the blockchain public ledger, choosing a Bitcoin wallet program, accepting Bitcoins as a payment method, as well as Bitcoin cryptography, mining and programming.

<http://www.nostarch.com>



Optek Music Systems, Inc.'s SDK for the Fretlight Guitar

The Fretlight Guitar from Optek Music Systems, Inc., is similar to other high-quality guitars you'll find at a music store, only with one important difference. Each is equipped with built-in LED lights that indicate the fingering positions for chords, scales, songs, riffs and even tablature right under one's fingertips. What's more, the instrument can be connected to a computer for an enhanced experience. The news about the Fretlight that will truly set the geek heart aflutter, however, is the new software development kit, which enables the creation of new, third-party desktop applications that enhance the musical experience. With the new SDK, Fretlight is opening up its interface code so that developers working in C and C++ can write software applications, royalty free, for Linux, Mac or Windows. Optek opines that this new development will lead to a wide range of new consumer applications. With the addition of the Linux Community, the chances are certainly improved.

<http://www.fretlight.com/developers>



NetComm Wireless Ltd.'s NTC-6200 Router



Most of the world's mobile operators offer Machine-to-Machine (M2M) services. As the focus in this sector moves beyond simple connectivity to solutions that improve business productivity and efficiency, firms like NetComm Wireless Ltd. are releasing products like the NTC-6200 Router

3G M2M Router Plus. The NTC-6200 series devices, says NetComm, are an array of versatile, intelligent and reliable M2M devices that address universal applications in areas like asset management, security and surveillance, building automation, POS, healthcare and smart city initiatives. The new NTC-6200 series was designed to deliver reliable connectivity and remote management capabilities to ensure that businesses can keep operational costs to a minimum and run their assets efficiently. The embedded Linux OS and SDK facilitate the development of custom software applications, giving users the freedom to create their own custom software applications.

<http://www.netcommwireless.com>



VectorWerks' Vector Cupholder

Think of the new Vector Cupholder from VectorWerks as a low-cost insurance plan for your device collection. This new product neatly solves the problem of where to put a beverage safely when most of your workspace is occupied by your computing equipment, papers and books. The Vector Cupholder solves this dilemma by taking the beverage off the workspace altogether. The Vector Cupholder securely grips table edges as thick as 1.5" (3.8cm) and holds a wide assortment of containers—from a huge, liter-sized tumbler to a small airplane cup; from Thermos bottles to soda cans; from water bottles to mugs up to 3" (8cm) in diameter. By moving beverages off the table, the Vector Cupholder also delivers the added benefit of freeing up “real estate” on cramped work surfaces like coffeehouse tables and airplane trays. The investment in the Vector Cupholder is certainly less than the cost and hassle of getting cappuccino out from under a keyboard.

<http://www.vectorwerks.com>

Plug-up International's Klije Authentikator for FIDO

With its new Klije Authentikator for FIDO UAF USB token solution, Plug-up International seeks to demonstrate how high-security USB consumer products that are compliant with the FIDO Alliance standards can be built for minimal added cost. The FIDO Alliance seeks to change the nature of on-line authentication through mechanisms that reduce reliance on passwords. Plug-up says that the Authentikator's unique USB Smartcard form factor “provides stronger security to desktop, laptop and portable devices in a universal, fun and portable way”. Plug-up is a member of the FIDO Alliance.

<http://www.plug-up.com>



Please send information about releases of Linux-related products to newproducts@linuxjournal.com or New Products c/o *Linux Journal*, PO Box 980985, Houston, TX 77098. Submissions are edited for length and content.

Remote System Administration with **Android**

**A look at some Android apps
to help you get your job done.**

Federico Kereki

Being a system administrator isn't simple, and being able to detect problems and quickly solve them from afar is quite an advantage. Server problems used to mean receiving an urgent phone call and then making a trip to a data center to fix whatever was needed. However, today's Android-powered phones ("phablets") and tablets provide a vast array of tools to help you do away with most of the bother. You can monitor things remotely, receive alerts instantly and solve problems from wherever you are. In fact, given the sizes, CPUs and raw power of today's devices, it wouldn't be too far wide off the mark to say you can do anything for which you might otherwise require a desktop or laptop.

In this article, I first cover some tools for diagnostics and for server and services monitoring. Then I describe some other tools for remote access, so you can patch up things without going anywhere. I don't include tasks that are performed easily with common apps (say, Chrome for Webmin or PhpMyAdmin, Gmail for getting e-mail alerts and so on). Instead, I focus on more-specific, less-known Android apps.

I ran all the examples here on my home network, which includes a server, a couple Wi-Fi routers,

several PCs, laptops and what not. I even added a virtual server running a couple services, and I set up Nagios monitoring for the servers. For extra variety, I also decided to keep an eye on an arbitrary external Web service (Open Weather Map, at <http://api.openweathermap.org>), whose public API I possibly might want to use for some application. All the tools I use will work equally well with public or private networks, so this setup won't affect your tests.

Are Servers Running?

For starters, let's consider how to keep an eye on Web pages, services or servers. For Web pages, consider signing up for an on-line monitoring service (see Resources for a short list), which you can set up to be warned about failures and crashes by text message or e-mail. As a bonus, some of these companies even provide their own Android apps to check logs, run tests and more from your device. These services are quite simple to set up, so you can use them in addition to the other tools I consider here—more eyes on your systems can't hurt!

To check servers and services directly, I used to use Decaf Monitor, but nowadays I prefer Server Monitor, because I find its interface to be clearer, and it's easier to notice what

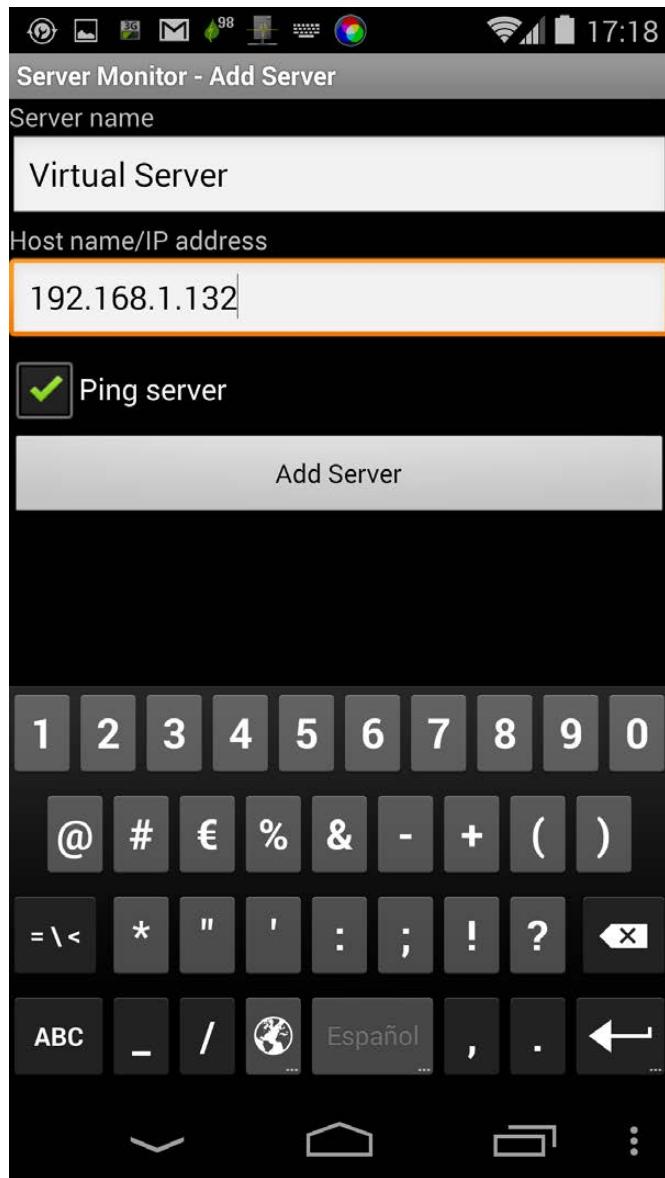


Figure 1. Adding a New Server for Monitoring in Server Monitor

isn't working right. (Of course, feel free to test both apps; they are rather similar in concept.)

First, define what servers you want to monitor and give each a descriptive name, plus a hostname or IP address. Then, long-tap the server name and select "Add Component" to specify

Server Monitor	
Main Server	Online
Web server	Online
ssh	Online
MySQL	Online
Virtual Server	Offline
Web server	Offline
ssh	Offline
Router #1	Online
Router #2	Online
Open Weather Map	Online
web service	Online

Figure 2. A problem is detected: the SSH service isn't running on a server.

a port/service to check. (Tests can be done directly via TCP or over SSH.) Specify the component name (any description) and the port number. The main display will show your server, and below it, you'll see all of its components in green ("on-line") or in red ("off-line").

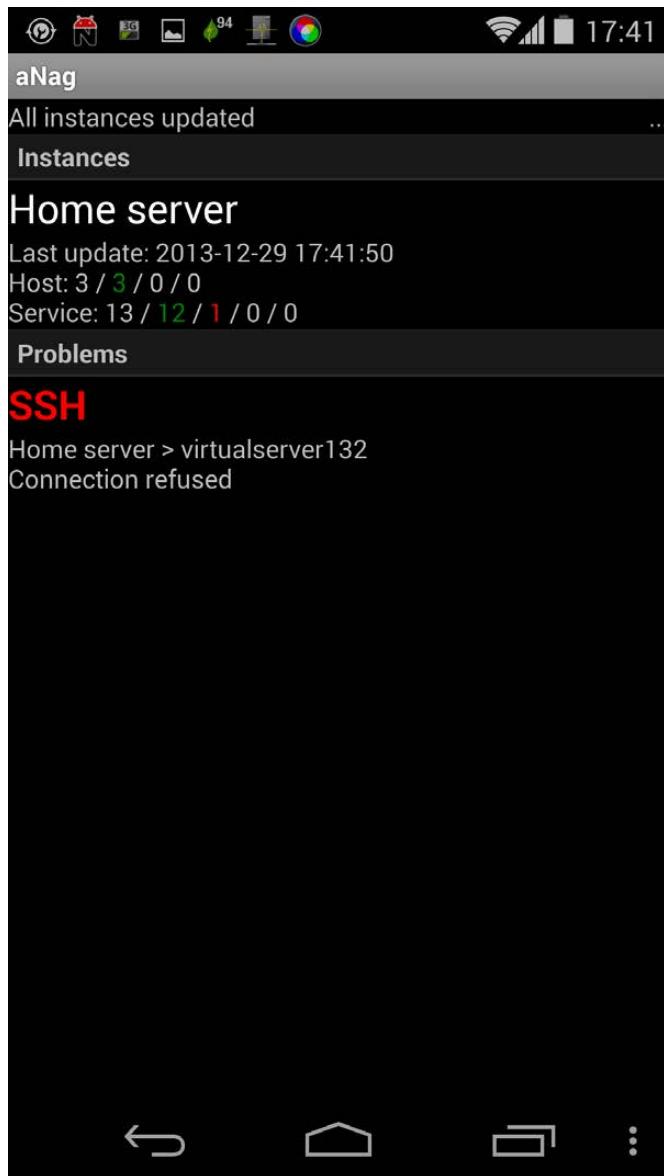


Figure 3. aNag shows that SSH isn't accepting connections on a server.

Finally, go to "Settings", and you can define whether Server Monitor will start at boot, choose the polling frequency and whether it will use sound and/or vibration to let you know if any monitored component isn't working. Server Monitor does its job simply and well, and it lets you

keep an eye on servers from afar, so it's a worthy tool for your Android-monitoring bag of tricks.

For more specific and complex server setups, you probably will use a monitoring tool, such as Nagios. (There are other options, but that's what I use.) You can verify the status of all monitored servers and services over the Web, but aNag goes a bit further and does these verifications on its own and then presents a summary of results that you can analyze at a glance.

You even can opt for a widget, which will sit on your screen, updating itself periodically and warning you in the event of problems. Tapping on the widget produces a more detailed explanation. For more clarity, "T" means "Total", "U" means "Up", "D" is "down", "O" is "okay", "C" is "Critical", "W" is "Warning", and "U" is "Unknown". For more details on those, check Nagios' documentation (see Resources).

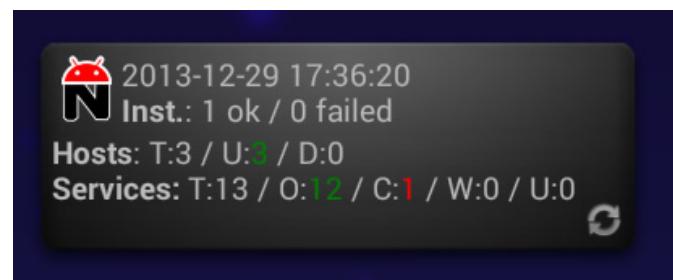


Figure 4. The aNag widget shows a critical problem with a service.

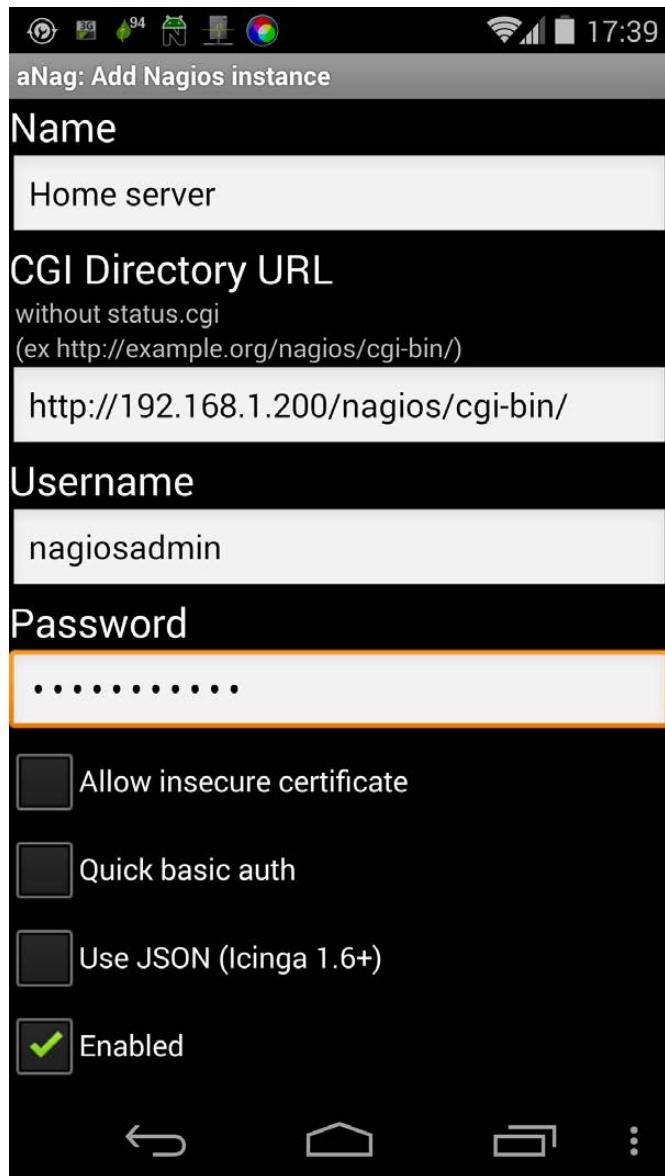


Figure 5. To set up aNag, provide the details of all available Nagios instances it should watch.

Basically, aNag works by connecting to your Nagios instances, scraping their Web pages, and summarizing and consolidating the information within in order to present it more compactly. To configure it, go to its settings, and define one or more

Nagios instances. You'll need to specify its name, the URL for the CGI Web access excluding the final "status.cgi" (this parameter usually looks like `http://your.own.site/nagios/cgi-bin/`, but check your installation), user name and password for Nagios access, or possibly some extra parameters for custom authentication schemes.

While you are at the settings page, examine the multiple options in the Notification tab, which include not only several methods (vibration, sound, LED), but also an "Auto Silence" configuration (to avoid extra warnings after the first one) and "Quiet Hours" (periods of time during which you want no updates and no notifications). Additionally, you should set the "Auto update" delay (how often the test should run), and also take a look at all the other available configuration possibilities (there are too many to include here).

Is the Network Okay?

Another common problem is diagnosing a possibly faulty network, and for this kind of job, I use Fing Network Tools. (Its developer, Overlook Soft, also offers Fingbox for remote monitoring of your network by means of a locally installed sentinel, which is able not only to warn



Figure 6. Fing's network discovery function provides a clear view of your network.

about network discoveries and events, but also to do services monitoring, server rebooting and more.) If you are curious about the name, it's a combination of two common command names, find and ping—just don't look up this word in Hungarian!

Fing's main screen provides a network discovery report, which shows all equipment (Wi-Fi, Ethernet) in your current network.

Tap on any entry to get full details, including IP address, MAC address, vendor, NetBIOS data (Domain, Name, User and Role) and more. You also can choose an appropriate

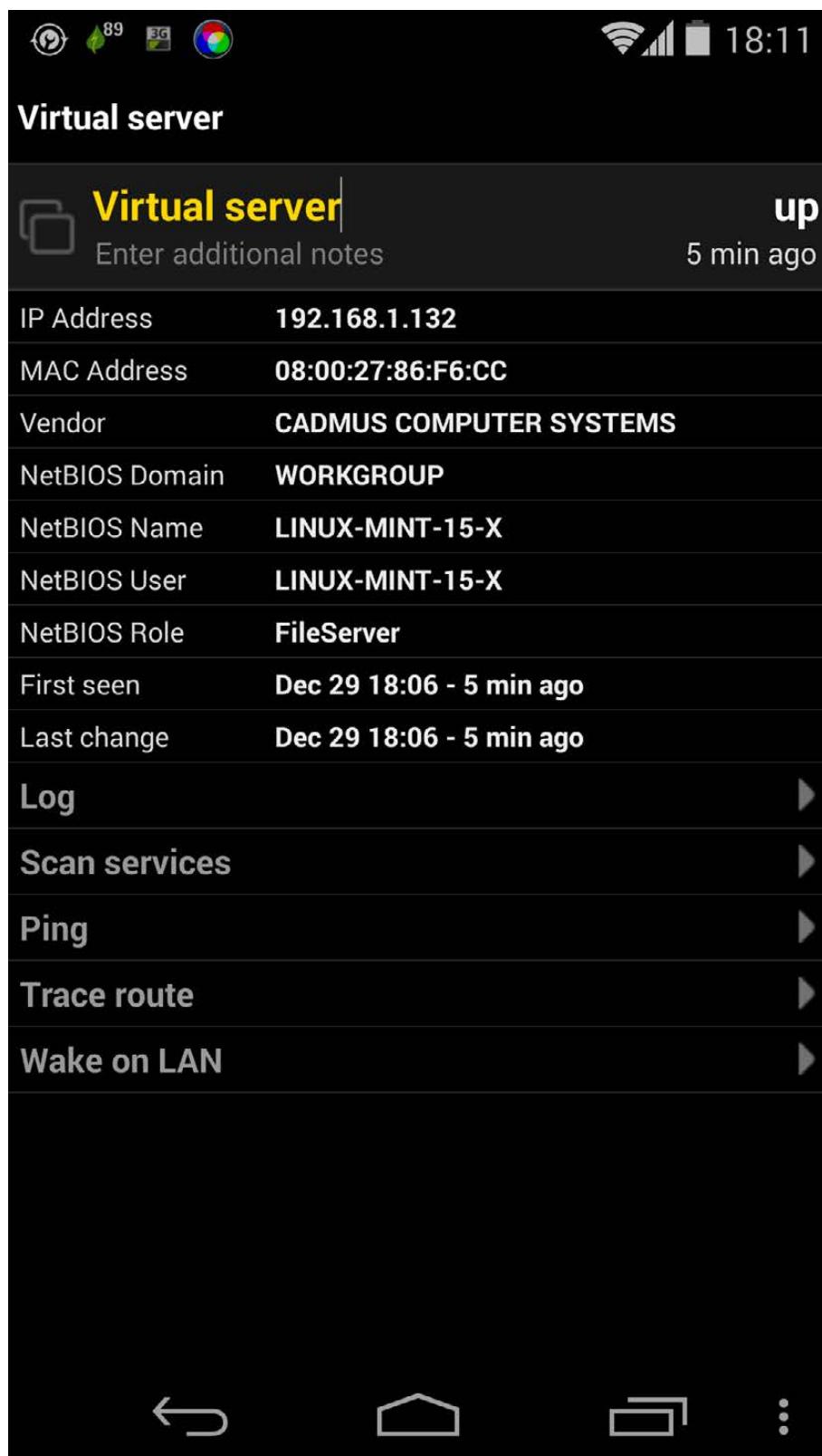


Figure 7. Tapping on an entry provides more information and access to extra functions.

icon, name and description for each piece of equipment. Plus, you can tap on “Scan Services” to determine all open ports, “Ping” and “Trace Route” (the equivalents of the command-line ping and traceroute commands), and “Wake on LAN” to wake up a computer remotely.

Clicking on the gear in the top-right corner provides access to several more tools, including Network Discovery, Host Tools (port scanning, ping, traceroute, DNS lookup, “Wake on LAN” and TCP connection testing), as well as several configuration items, backup and restore, and more. For even more tools, consider the Net Swiss Tool, which adds even more tools to Fing’s, including arp, netstat, route, whois and other functions, such as DNS testing,

port scanning and more. If there was a tool with the functionality of both these apps together, using it would be a no-brainer, but for the moment, you'll have to work with this pair!

Is Wi-Fi Working?

Configuring or debugging Wi-Fi setups is also a typical task, and for this, I use Wifi Analyzer. This app provides five useful tools: a channel graph, time graph, channel ratings, access points (AP) lists and a signal meter. The channel graph view shows all available networks (either in 2.4GHz or in 5GHz—touch the icon to the left to select which) along with their channels and signal strength.

Alternatively, you can examine the same information in listing form, by choosing the “AP list” view, which shows each available connection point, along with an icon and a bar showing the connection strength, the device brand, and the used channel and its frequency. The list can be sorted alphabetically, by channel, by signal strength, by openness, by vendor or in “natural order”, which shows your best options first in availability terms. If you want to connect to any of the shown networks, just tap on it.

The “Time Graph” view shows the different AP's strengths over

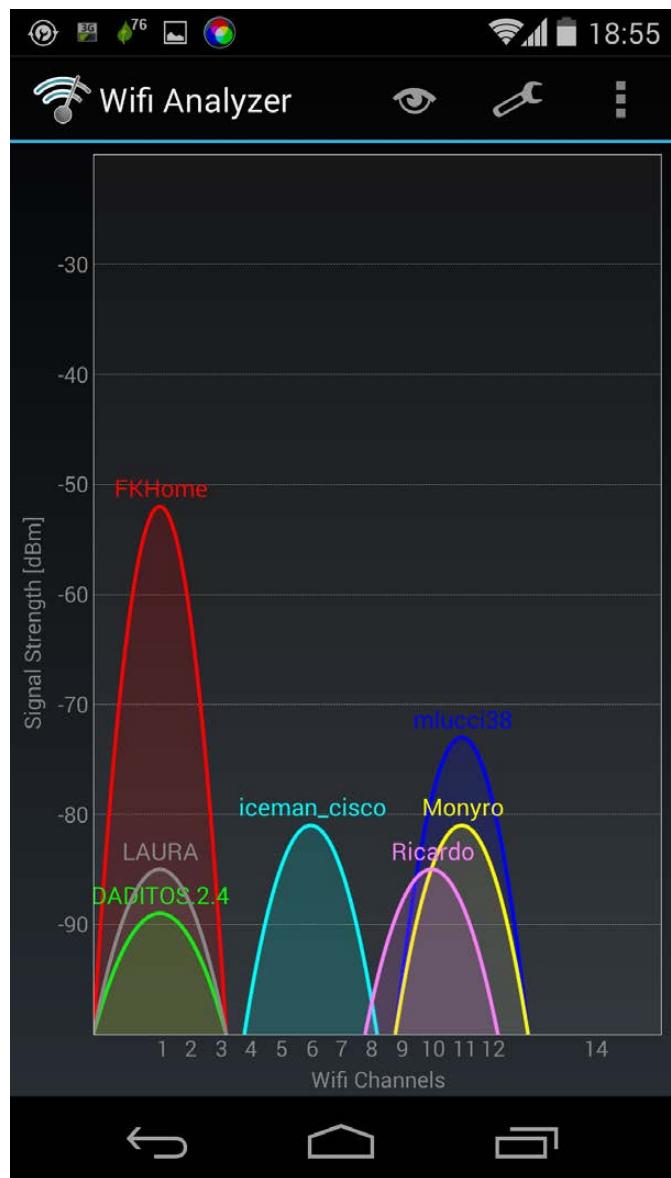


Figure 8. Tapping on an entry gives more information and provides access to extra functions.

time. Available AP's are at the top of the screen and are color-coded, and if you tap on one, its graph will be highlighted. To determine which channel to assign to an AP, use the “Channel Rating” view, which shows a horizontal bar graph, pointing out

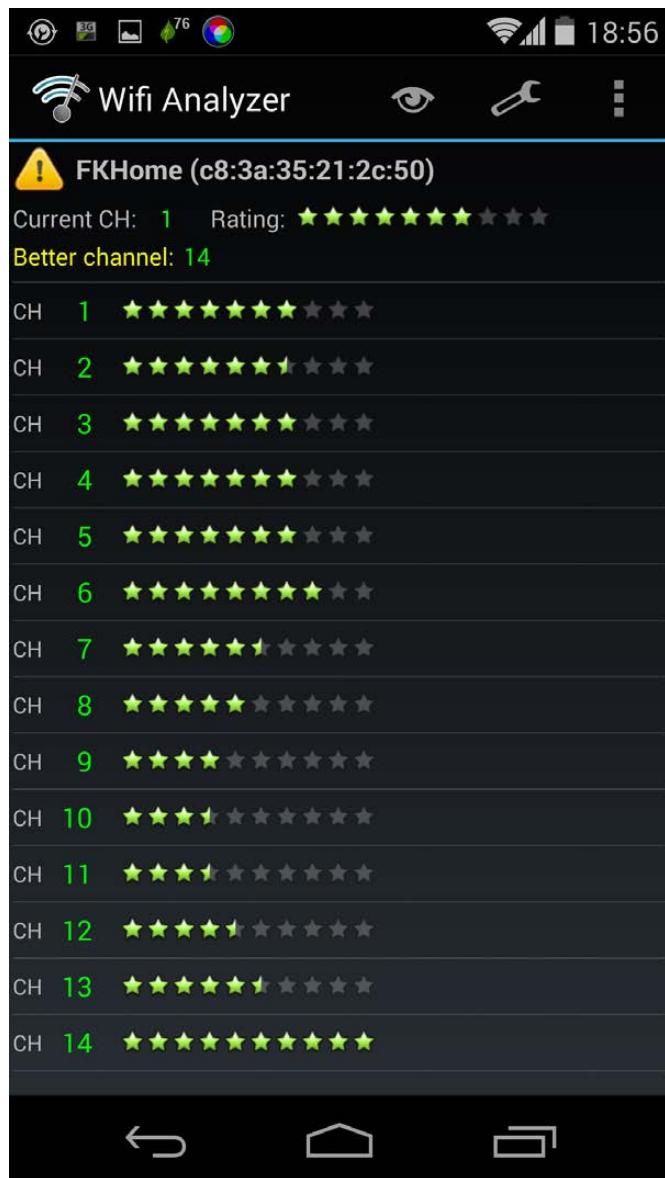


Figure 9. You also can determine the optimum channel to use for a router.

which would be the best options for the AP, in terms of interference with other nearby networks, on a scale from 0 to 10. Take care to configure the app and select which channels are available. In the US, there are channels 1 to 11, but 12, 13 and 14

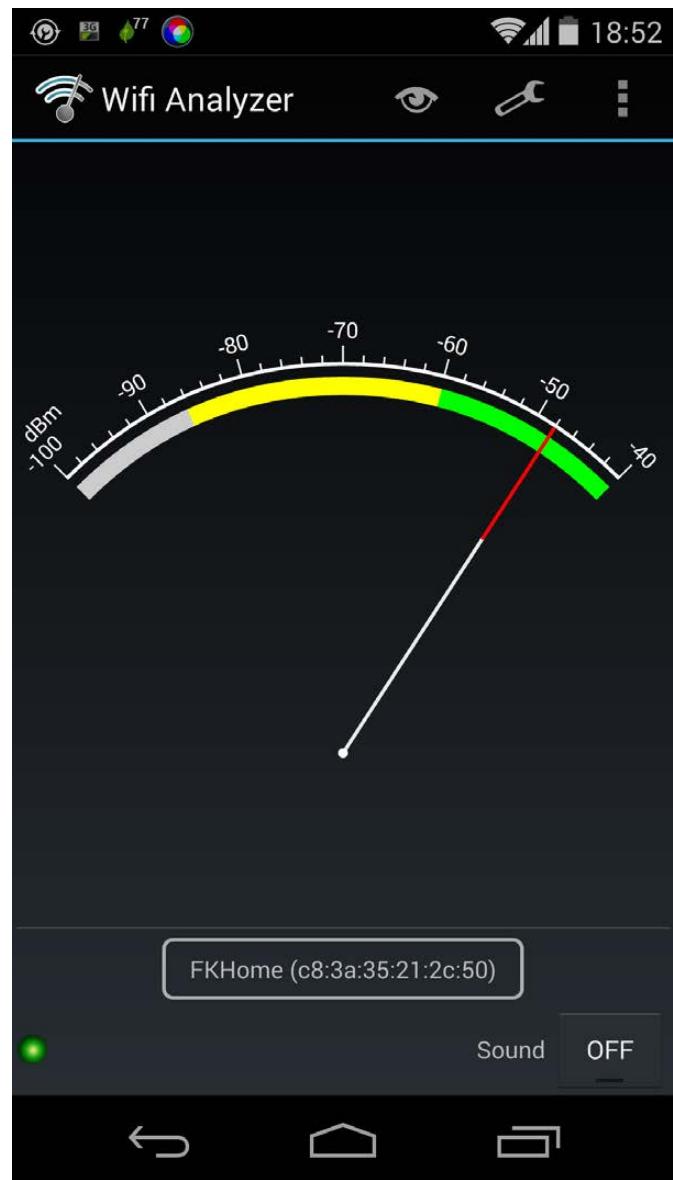


Figure 10. Finding the best place for an access point or router is easy with Wifi Analyzer.

are available in other regions. If your AP isn't using the best option, just configure it and fix the problem.

Finally, the most entertaining function (which will make you feel like a grizzled old uranium prospector, walking around with a Geiger counter

looking for ore) shows a dial with a “needle” showing the current signal strength of the AP to which you are connected, and it even adds the option of beeping in proportion to the strength. Walk around, closer or farther away from the AP, and the pinging rate will vary. This will let you determine the best place for an antenna, router or repeater, and it also lets you know if the signal strength is what you expected.

Wifi Analyzer can handle a lot of common problems, which is why it's in my bag of sysadmin tools.

Remote Access

If (when!) something happens, what you need is a remote console session, so you can work directly on fixing whatever is wrong. For this, over several terminal emulators, I prefer ConnectBot. Its stable version at the Google Play store is quite old (October 2010!), but work still goes on, and development versions, as new as November 2013, are available at the project site (see Resources).

With ConnectBot, you can use SSH or Telnet to connect to a remote server (or to your own Android device)

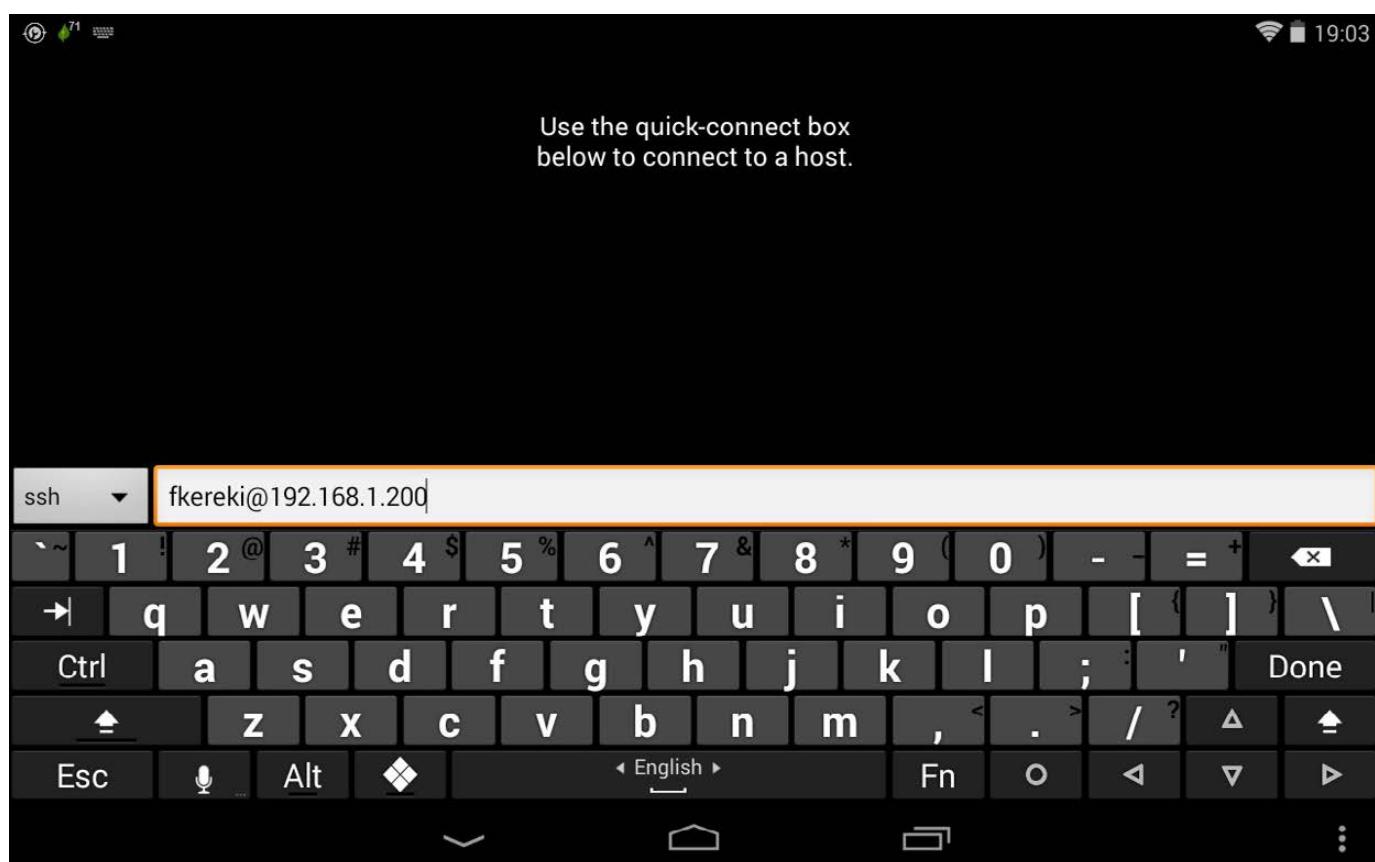


Figure 11. Connecting to a remote server is easy with ConnectBot.

```
Last login: Sun Dec 29 00:07:36 2013 from console
Have a lot of fun...
fkereki@fkereki-desktop:~> cd /
fkereki@fkereki-desktop:/> uptime
19:04pm up 3 days 19:01, 5 users, load average: 0.11, 0.35, 0.48
fkereki@fkereki-desktop:/> ls
bin boot dev home lost+found mnt proc run selinux sys usr
bck data etc lib media opt root sbin srv tmp var
fkereki@fkereki-desktop:/> █
```



Figure 12. Working remotely is best with a landscape tablet and Hacker's Keyboard.

for command-line work. You must specify the host and port to which to connect, and you need to provide either your user name and password or public key certificates (see <http://michaelchelen.net/articles/android-connectbot-ssh-key-auth-howto.html> for this) to gain access. ConnectBot stores your connection details to simplify reconnecting to known servers.

You can be connected to several servers simultaneously and change from one to another. Select the “Disconnect” menu option, use

the exit command or the key combination Ctrl-D at a console to disconnect fully. The back button returns you to the main menu, where opened connections will show a green icon; disconnected ones show it in red. Tap on an open connection to go back to that session.

You can use ConnectBot either in portrait or in landscape mode, but the latter is best, especially for using the “Hacker’s Keyboard” and getting all the usual PC keys. The small console text can be hard on your eyes, so you can use the volume up and down keys



Figure 13. Defining a connection to VNC is simple.

to increase or reduce the font size. For even more screen space, you can remove the status line in Settings.

For better history, set the scrollback size to 1000 or more lines. So connections won't be dropped when they are in the background, check the box "Persist Connections". Finally, set "Keep Screen Awake", so your device won't turn off while working, and set "Keep WiFi active", so your connection won't be dropped.

Doing It Graphically

Working with a remote server over

SSH is probably the most common way of fixing sysadmin problems, but if you want to connect to a machine running the X Window System (not quite usual on servers, but far more likely on terminals), there are ways to access that server from your tablet. (Why a tablet? Scrolling and panning a full-size display on a phone's small window is just about murder!) There are two options you should consider: android-vnc-viewer and TeamViewer. The first is totally free, and the second is free too, but only for non-commercial work (it



Figure 14. Yes, the remote desktop can be tiny! Zooming and panning are mandatory.

requires a license otherwise).

On opening android-vnc-viewer, you can choose an earlier defined connection or create a new one. Speed for such connections usually isn't great, so you may want to investigate different color formats—24 bits of color depth look good, but fewer colors mean higher speed. When a connection is established, you can work remotely as with a browser. Your finger is the mouse, a single tap is a left-click, and a double tap is a right-click. (For other needs, try Menu→Send Keys.) Long-tapping the

screen provides buttons for enlarging or reducing the display, as well as a virtual keyboard. This app can be a bit harder to configure, because you'll probably have to open ports and so on in order to access an internal machine over the Web. If this is a problem, check out the second option.

If you have someone over at the remote machine, and the security rules at your site allow for it, TeamViewer is easier to use, because it doesn't require configuring any firewalls or forwarding any ports. If a machine can connect to the Web, it can connect via

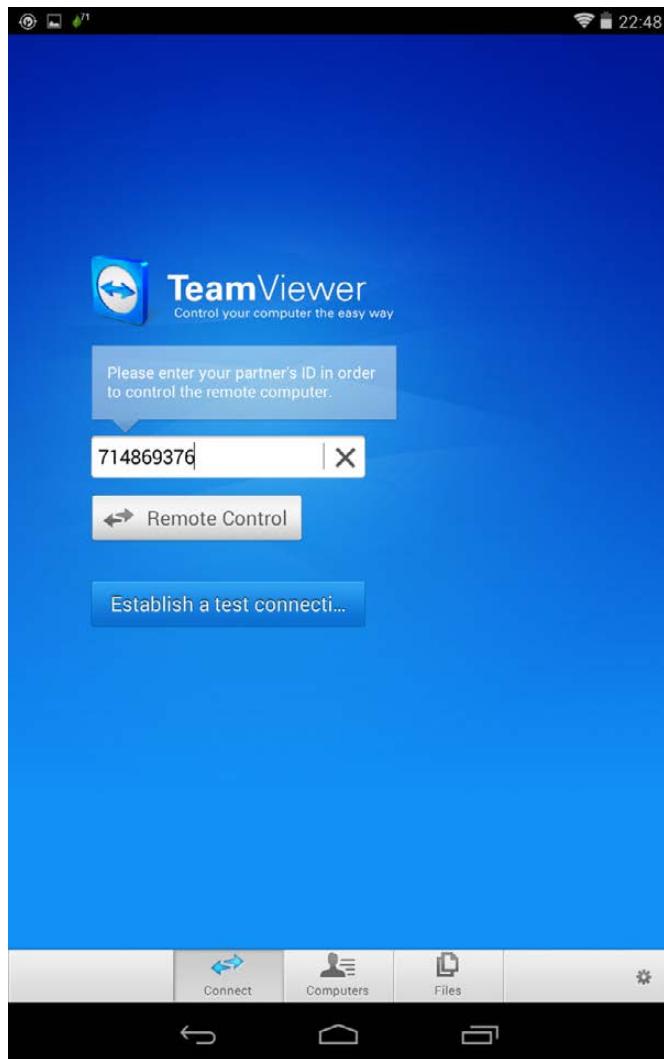


Figure 15. With the help of a user, you can connect to any machine with Internet access.

TeamViewer to another machine, since the TeamViewer servers mediate the connection. You'll need a (human!) helper at the remote machine to start a session and provide you with an ID and password.

While the session is open, you not only can work remotely, but you also even can perform special actions,

DOES SIZE MATTER?

These days, I'm completely Google-fied and going for stock Android: my phone is a Nexus 5 and my tablet is a Nexus 7. (The tablet is Wi-Fi-only, but I can make do by tethering the phone as a portable hotspot.) As a matter of course, I unlocked and rooted both devices, and I also added a Hacker's Keyboard to them to get keys like Alt, Ctrl, Esc and Tab.

If I had to choose to work with only one of the Nexus devices, it would be a hard decision to make. The (small-ish) phone is more comfortable to use and carry, lets me run quick server tests and receives alarms on the go, all of which are a plus. But, the larger tablet is far better for working with remote SSH or VNC connections, which require a lot of screen space. So, I usually carry both devices, and I even work with both at the same time, fixing things with the tablet and checking the results on the phone—dual-screen sysadmin work!

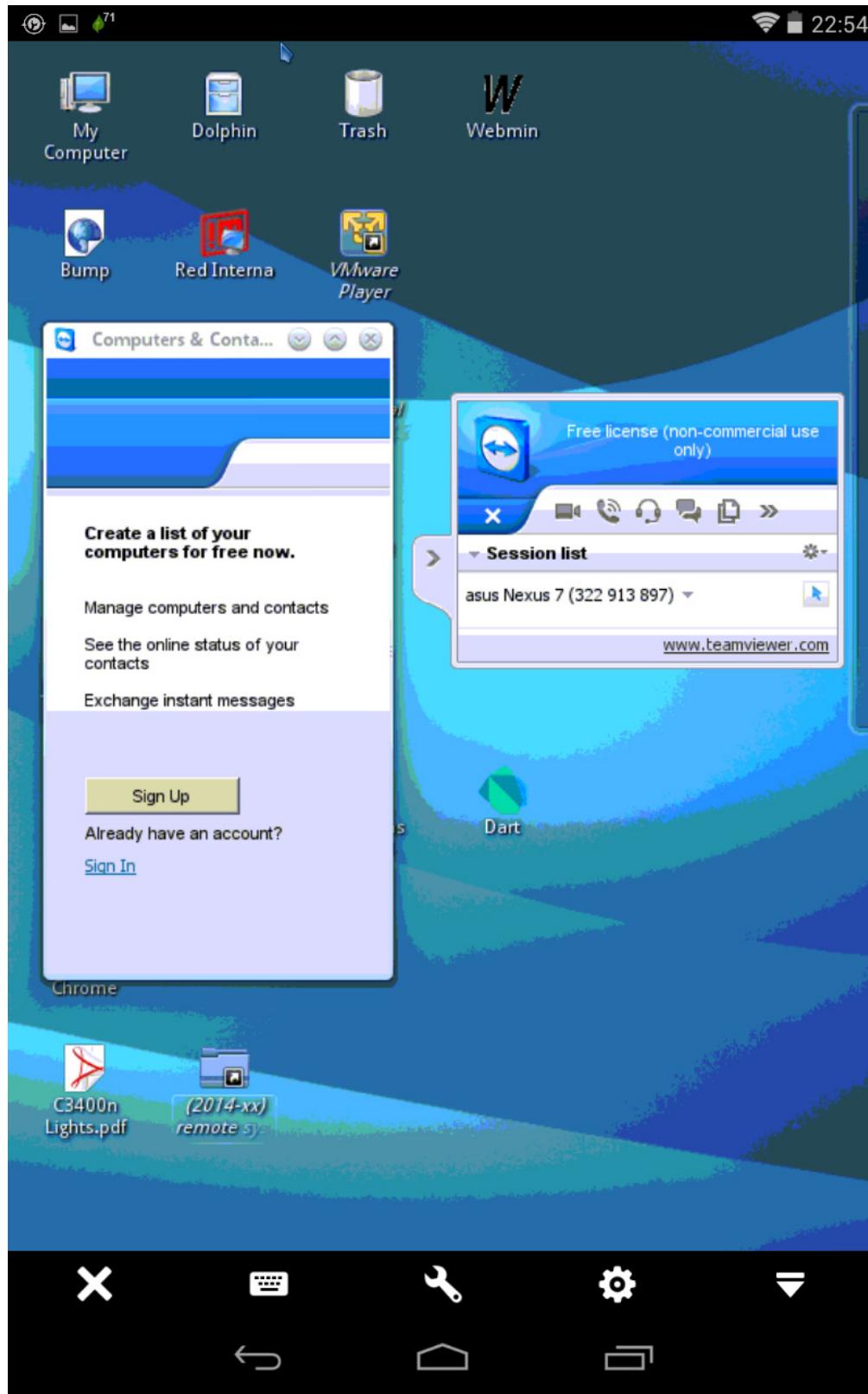


Figure 16. Working in low resolution, speeds can be acceptable even over not-so-good connections.

such as rebooting the machine or locking your helper out, so only you will be able to work. You won't need a Hacker's Keyboard with this app, as the app can show a virtual overlaid keyboard, which lets you simulate all keys, such as Ctrl, Alt and the Meta (or Windows) key.

There are some display method settings (for quality or for speed) that can change the connection speed. Performance can be lower than with VNC, but given that connection requirements are minimal (Web access), this app may help you out in difficult cases.

Working with Files

Apart from directly connecting to a server, you may need to upload and download files. (Of course, for just editing a file, a remote access solution does the job, but you might prefer to download the file to your tablet, edit it there with a better editor such as Jota+, and then upload it again.) Out of many available file browsers, I find ES File Explorer to be the most useful. You not only can browse local files (even the “/” directory, on rooted devices), but you also can use several Web protocols (SMB, FTP, FTPS, SFTP and WebDAV) and

cloud services (Box, DropBox, Google Drive, S3, Skydrive, SugarSync, Yandex and Ubuntu One), working with remote files as if they were local to your device.

To connect to a remote server, choose “Network” in the left-hand list, then “FTP”. Next, define its parameters: protocol (FTP, FTPS, SFTP or WebDAV), server, port, user name and password (leave these empty for anonymous access), data encoding and display name. For secure connections, the remote server will require the **PasswordAuthentication SSH** parameter to be set to Yes. (Note that

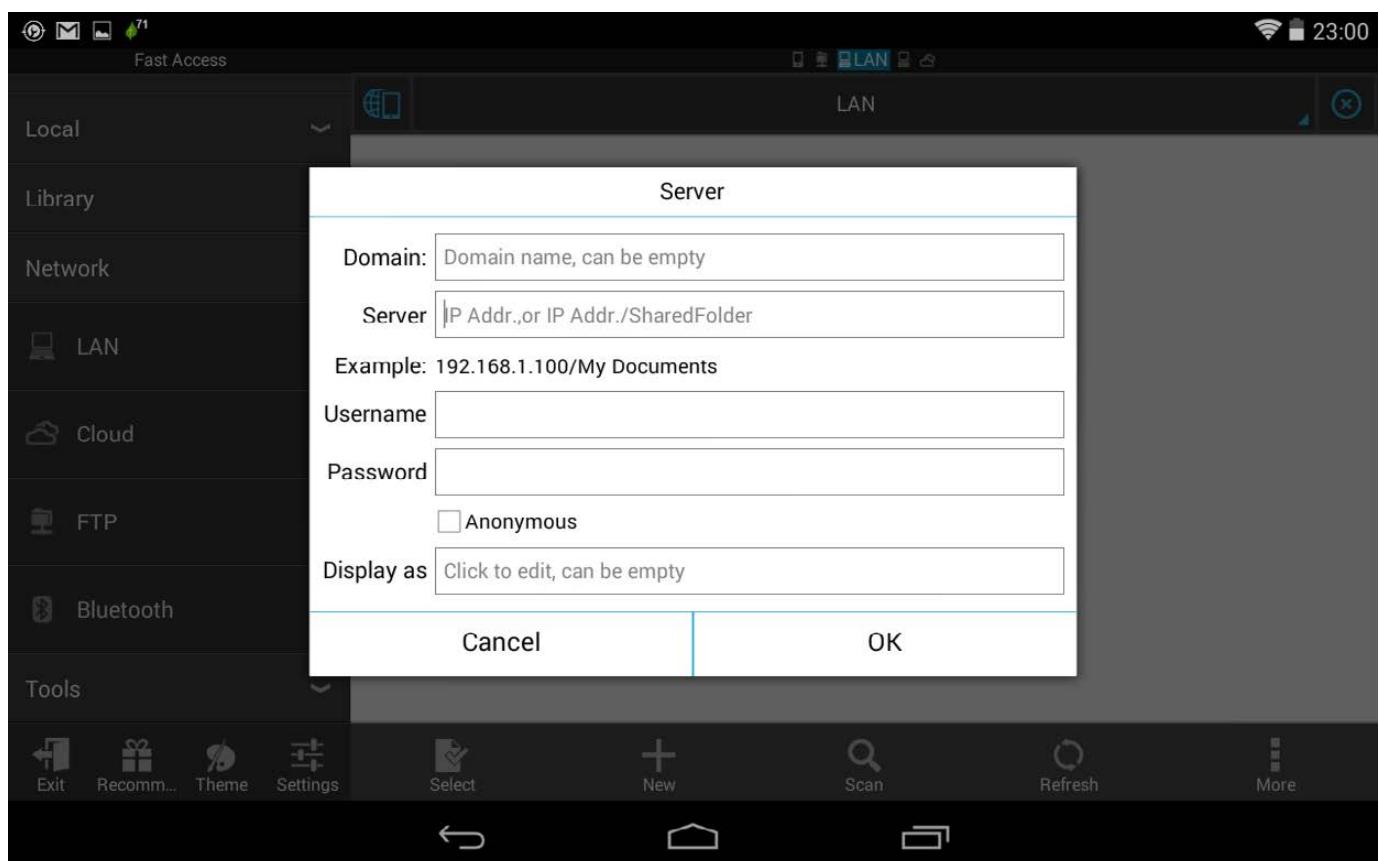


Figure 17. With ES File Explorer, you can connect to servers and cloud servers.

MAKE YOUR CHOICES

How do I choose apps? First, I prefer to get everything off the Google Play store, so I can take into account the ratings, number of installs and user reviews. To detect “abandonware”, the version number and, more important, the date of the last app update are relevant. I also visit the developer’s Web site and look around. Does it look professional? Is documentation provided for the app? Is there a way to communicate with the developer directly for questions or bug reports? If so, does the developer answer quickly?

For security reasons, I try to be very careful with the permissions required by the app: are they logical or do they look fishy?

As for pricing, I (of course) like free apps better, but even most paid apps also have a free version to try before updating to the paid one, which works well for everybody.

Finally, since there usually are several overlapping apps for each function, I generally try my hand at many of them before settling on one. I also periodically check for new apps, because updates and new functions are usually frequent, and apps previously left aside may become a new favorite. I have changed tools several times and probably will continue doing so.

this is also a requirement for other file managers that are capable of SFTP access.) If you can’t connect to the server, edit its /etc/sshd/sshd_config file (the path and name may vary depending on your distribution), change the appropriate line, and restart the service with a sudo /etc/init.d/sshd reload command.

Long-tap a defined server to edit

it. Tap a server, and it will show its directories and files, as if they were local. Long-tap a file, and you can work with it (open, copy, cut, delete, rename, share, send and more). If you need to make only a simple fix, opening a file as text lets you edit it with a (rather basic) editor. Finally, you can select several files at a time for group operations.

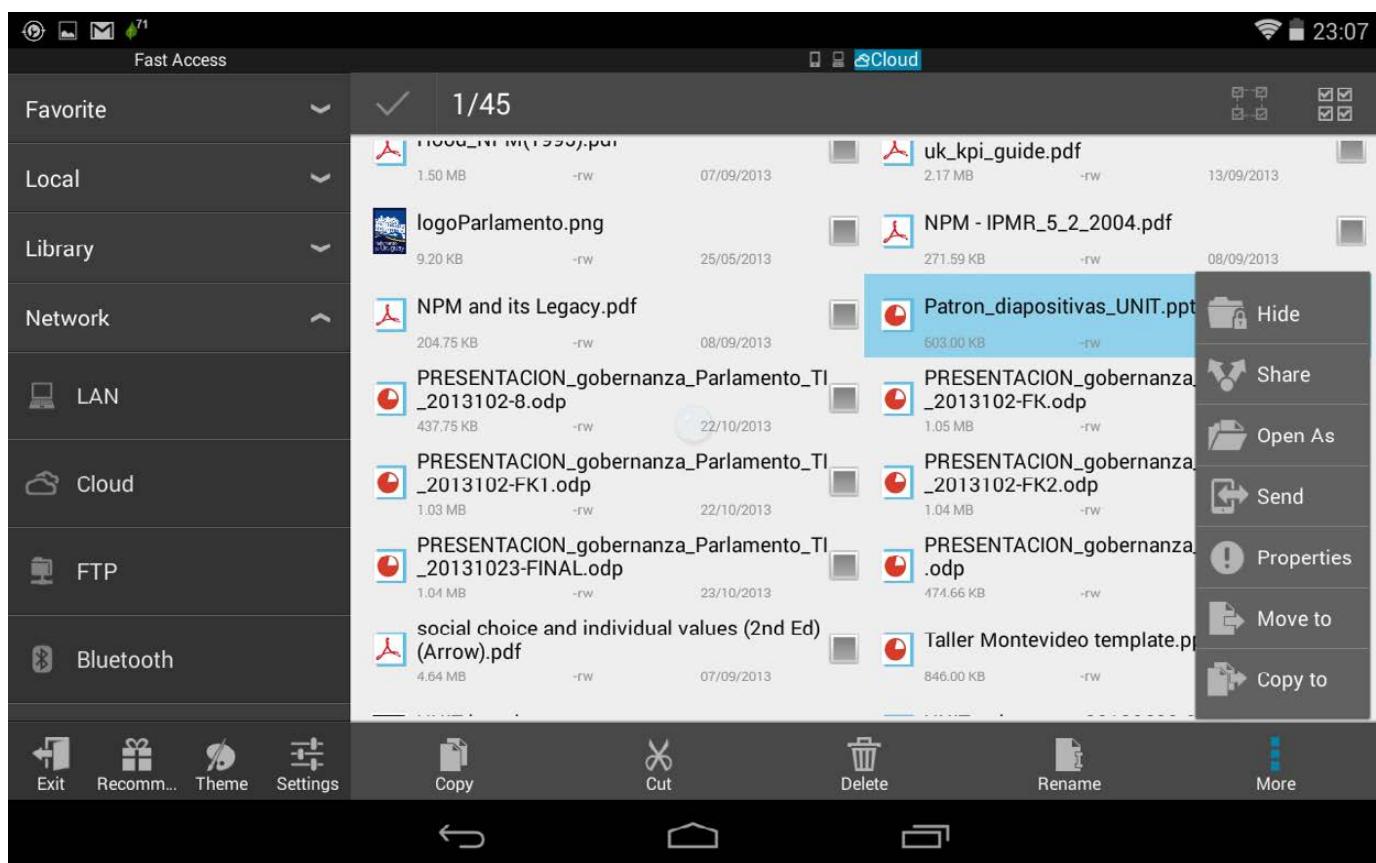


Figure 18. You can work with remote files as if they resided on your device.

File browsers are widely available, well known and similar in function. However, I find ES File Explorer to be a bit ahead of the others because of the simple way it shows and works with remote directories and files as if they were local, making work easier and faster.

Conclusion

Sysadmin work is never going to be really easy, but with the Android apps I've shown here, you will be able to analyze networks, monitor servers and services, detect problems as they happen, and connect to servers to get

things running again. Hopefully, your life will be a tad easier than before. ■

Federico Kereki is a Uruguayan systems engineer with more than 20 years of experience developing systems, doing consulting work and teaching at universities. He wrote the *Essential GWT* book, and he currently is working with a good jumble of acronyms: SOA, GWT, Ajax, PHP and more. All his computing machines run either Linux or Android, and he finds the mixture quite appealing to his FLOSS interests! You can reach Federico at fkereki@gmail.com.

|||||
Send comments or feedback via
[**http://www.linuxjournal.com/contact**](http://www.linuxjournal.com/contact)
or to ljeditor@linuxjournal.com.

Resources

Here's a list of the apps I'm using at the time of this writing:

- aNag: <https://play.google.com/store/apps/details?id=info.degois.damien.android.aNag>; version: December, 2013; 10,000–50,000 installs.
- Button Savior: <https://play.google.com/store/apps/details?id=com.smart.swkey>; version: December, 2013; 100,000–500,000 installs.
- ConnectBot: <https://play.google.com/store/apps/details?id=org.connectbot>; version: October, 2010 (!!); 1–5 million installs. For newer versions, check the developers' site at <https://code.google.com/p/connectbot>.
- Decaf Monitor: <https://play.google.com/store/apps/details?id=net.nineapps.monitor>; version: May 2012; 5,000–10,000 installs.
- ES File Explorer File Manager: <https://play.google.com/store/apps/details?id=com.estrong.android.pop>; version: December 2013; 10–50 million installs.
- Fing: <https://play.google.com/store/apps/details?id=com.overlook.android.fing>; version: December 2013; 1–5 million installs.
- Hacker's Keyboard: <https://play.google.com/store/apps/details?id=org.pocketworkstation.pckeyboard>; version: January, 2013; 1–5 million installs.
- Jota+: <https://play.google.com/store/apps/details?id=jp.sblo.pandora.jota.plus>; version: December 2013; 100,000–500,000 installs.
- Net Swiss Tools (with ads): <https://play.google.com/store/apps/details?id=com.sanctuaire.netswisstool>; version: December 2013; 100,000–500,000 installs.
- Wifi Analyzer: <https://play.google.com/store/apps/details?id=com.farproc.wifi.analyzer>; version: November, 2013; 10–50 million installs.

Just a few of many available on-line monitoring services are:

- Are my sites up?: <http://www.aremysitesup.com>
- Basic State: <http://basicstate.com>
- HostTracker: <http://host-tracker.com>
- HyperSpin: <http://www.hyperspin.com>
- Internet Seer: <http://www.internetseer.com>
- Monitor.us: <http://www.monitor.us>
- Montastic: <http://www.montastic.com>
- Pingdom: <http://www.pingdom.com>
- Site 24x7: <http://www.site24x7.com>
- Site Probe: <http://www.siteprobe.com>
- Site Uptime: <http://siteuptime.com>
- Uptime Robot: <http://uptimerobot.com>

Nagios is at <http://www.nagios.org>, and documentation is at <http://www.nagios.org/documentation>.

Attend Big Data TechCon!

The how-to technical conference for professionals implementing Big Data



"Great conference. I took a lot away from all of the talks I attended."

—David Hollis, Consultant, Raybeam Inc.

"You will great insights and the speakers will put you on the fast track."

—Chandrashekhar Vyas, Solution Architect, Diaspark

"Big Data TechCon offers great technology depth."

—Rahul Gupte, Associate Director, Deloitte

BigData TECHCON San Francisco

October 27-29, 2014

www.BigDataTechCon.com

Come to Big Data TechCon to learn the best ways to:

- Process real-time data pouring into your organization
- Master Big Data tools and technologies like Hadoop, Map/Reduce, hbase, Cassandra, NoSQL databases and more!
- Learn how to integrate data collection technologies with data analytics and predictive analysis tools to produce the kind of workable information and reports your organization needs!
- Collect, sort and store massive quantities of structured and unstructured data.
- Looking for Hadoop training? We have several Hadoop tutorials and dozens of Hadoop classes to get you started — or advanced classes to take you to the next level!
- Understand HOW to leverage Big Data to help your organization today

NEW USES for OLD DEVICES

**GIVE NEW LIFE
TO OLD PHONES
AND TABLETS
WITH THESE TIPS!**

BILL CHILDERS

If you're like me (and you're a *Linux Journal* reader, so you may actually be like me), you probably rotate through your cell phones and/or tablets every couple years. These little devices are so convenient and have been consistently dropping in price, while their power continues to go up, so you may have a few older devices sitting in a drawer. Thank Moore's Law for that—but what can you do with your old devices? There are a couple obvious things to do with old devices: some phone carriers allow you to "trade up", or you can hand them down to your kids or friends. However, there are quite a few uses for old devices aside from just pawning them off on your friends or disposing of them. This article is geared primarily at Android devices. Some of you may have older Apple iOS devices (it's okay if you do, I do too), and I'll drop hints on how to re-use those devices as well. With that, let's get started!

Media Player

This may be an obvious choice, so let's lead with it. Re-using an old device as a dedicated media player is easy and doesn't require much work. Just load up the device with your media of choice, and rock out! If you really want to get obsessive though, install the Plex Media Server running on a computer at your home, get the Plex Media Client for your mobile device (it's available for both Android and iOS) and turn your old smartphone into a media-streaming machine!

Digital Picture Frame

This is another obvious choice, but surprisingly easy to forget about. Simply upload any pictures you'd like to display, and then set the gallery/photo program to display them in a looping slideshow.

Security Camera/Baby Monitor

Smartphones generally have the ability to record video. Why not utilize that ability to have your own little wireless security camera? You can use the app IP Webcam if you're on Android (or Presence for iOS), and turn your old phone into an effective way to monitor your home. Alternatively, if you've got a small child, set it up in his or her room to act as a video-equipped baby monitor. I wish this

would have been possible when our daughter was little. It would have saved my wife and I a lot of tiptoeing into the kid's room to check on her... without waking her up, which was always a trick.

Kitchen Timer/Internet Recipe Lookup

When I saw someone do this, I thought, "Why didn't I think of that?" It's just that simple. Put an old phone or tablet on the counter, use its timer while cooking recipes, and have its Web browser up and running to display recipes while whipping up food in the kitchen. There's another trick that a smart device can do for you in the kitchen too. You can use its calculator to help do the math for doubling up ingredients for recipes. Who cares if you accidentally get the old phone full of frosting or flour? Just make sure it doesn't find its way into the oven—lithium-ion batteries don't take well to baking at 350°F for a half-hour.

Dedicated VoIP/Skype Handset

If you're a heavy Skype or VoIP user, why not re-use your old smartphone as a wireless handset for those services? Even an HTC Dream or iPhone 3GS has enough CPU to run VoIP applications, if it's not forced to do anything else. Bonus points if you

pair that with a Bluetooth headset for double the wireless action.

IRC/IM Terminal

If you're *not* a VoIP user, and prefer to use old-school messaging services like IRC, just drop a copy of Andchat on your Android phone (or Colloquy for iOS), configure it up, and have fun chatting like it's 1998. Optionally, you can set up a crazy Irssi proxy and bouncer to keep you connected 24 hours a day, seven days a week (see Resources for a link to the *Linux Journal* article on this).

Miniature Arcade System

This may be one of my favorite uses for an old smartphone or tablet—so much so, that I have one of these on my desk at work. There are a lot of emulators available for Android for your favorite arcade or console systems, although though you may have to scour the Google Play store to

find them. This also assumes you have a library of ROMs as well. It's possible to do this on iOS devices too, but you may have to jailbreak the device. When you're all done, you wind up with a really cool little arcade system, especially if you pair it with an iCade or iCade Jr.



Figure 1. Old iPhone with iCade Jr.

E-Book Reader

Why buy a Kindle, if you have an old device in a drawer? There are Kindle apps for both Android and iOS that are very functional. I actually have a couple old iPod Touches around the house that do nothing but this—I have one by the couch and another by my bedside. Since the Kindle application syncs reading positions via the Amazon cloud, it provides a seamless way for me just to grab a device and read for a while. It may be a bit overkill, even lazy, but it keeps the old devices out of the desk drawer.

Internet Radio

Much like the media player option, you can dedicate a device to being a Pandora, Spotify or Rdio streamer. If you've got a stereo receiver in your house, you even could wire it in semi-permanently and never use physical media again.

Build Your Own Car DashCam

Dashboard-mounted cameras aren't only for people in Russia to post videos to YouTube. They can be beneficial for lots of other reasons too. I wish I had one to capture the antics of my daily commute. With an app called Daily Roads Voyager and a suction-cup dash mount, any Android device can become a dashcam. It'll

loop the video so you don't have to worry about memory management, and it will correlate the video with a position on a GPS map. It could be handy in the event of an accident while driving.

Build an Android-Powered Laptop

This tip really applies only if you've got a Motorola Android phone, like a Droid RAZR or Atrix, but at one point, Motorola made a Lapdock, which was a clamshell docking station for its handsets that included a large battery built in to the base, along with a USB connection for the built-in keyboard and touchpad, and a mini-HDMI connector for video. These Lapdock units originally sold for \$500 or so when they were introduced, but now they can be found on eBay and Amazon.com for as little as \$50. Depending on your Motorola device, the Lapdock may work out of the box, or you may have to modify the docking connector slightly. (I had to flip the orientation of the HDMI connector to get it to work with a Droid RAZR, for instance.) A bonus of having a Lapdock on hand—if you have a Raspberry Pi, the Lapdock can be modified to work with that as well. *Linux Journal* Editor Kyle Rankin's got one of these

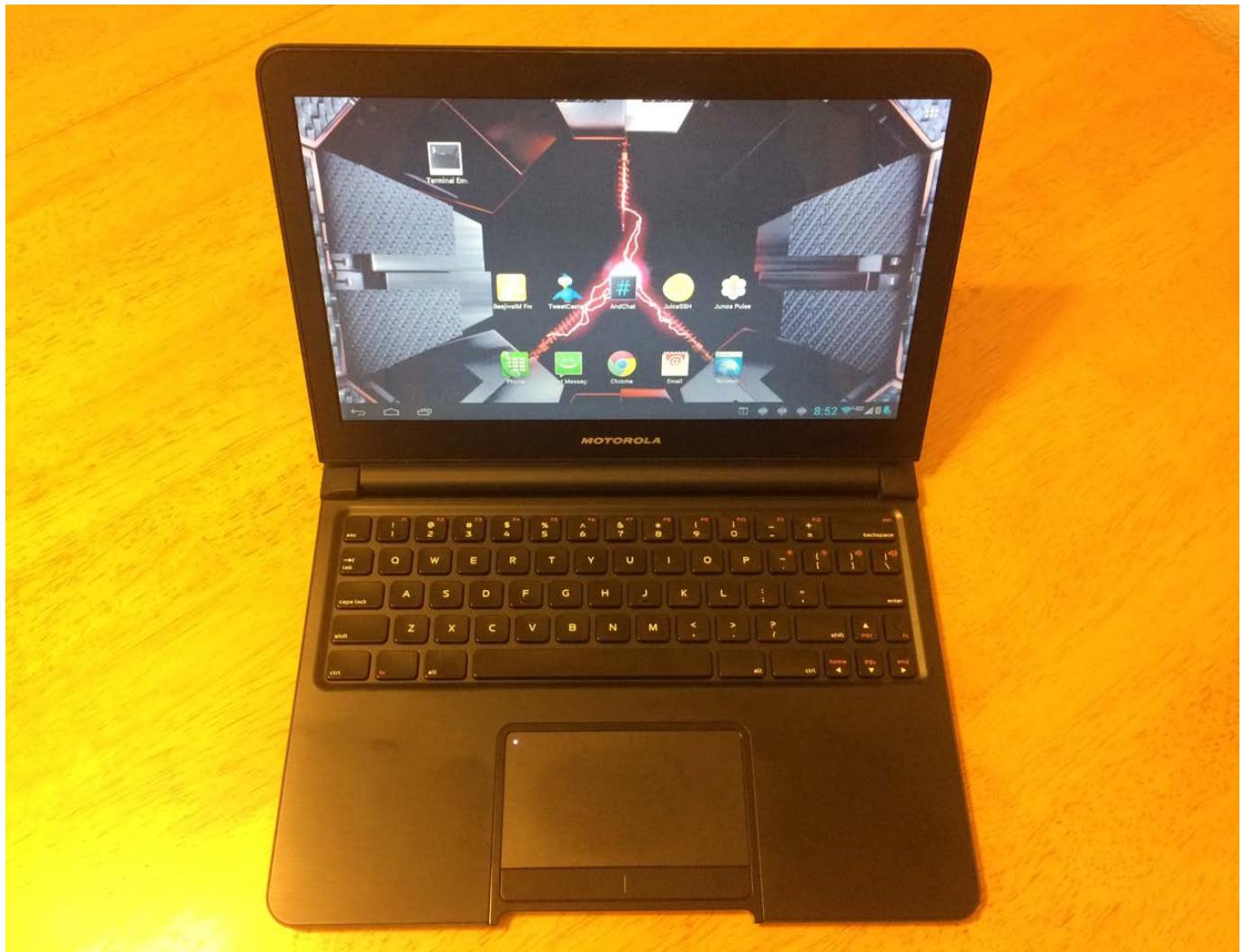


Figure 2. Android Laptop

homemade creations and has said that it works surprisingly well.

Experiment with Alternate ROMs

Only the curious need apply here, but it's a fun place to be. Due to the open-source nature of the Android operating system, lots of alternative ROMs have been released for various devices. The most prolific of these is the Cyanogenmod project. The

Cyanogenmod folks have released ROMs for just about any Android device released to date. The Cyanogenmod project aims to release a basic, no-frills, yet stable Android product, unlike some versions of Android that device makers ship. The Cyanogenmod team's products also work on some of the oldest devices out there, like the T-Mobile G1. If you're in the mood to coax new life out of an old Android

device, Cyanogenmod should be your first stop. If you're interested in digging deeper into the guts of your device, the XDA-Developers forum is a great place to find more esoteric ROMs and other information about your device. Many members of that community port newer ROMs to older devices, so be sure to look at the subforum that matches your hardware and see if you can find something cool to try out.

Create a Super-Low-Power Server

Another fun little project that you can do with an older Android device is to repurpose it to be a low-power server. It's possible to install packages like Apache on Android if you know what you're doing, but there's an easier way. There's a package on the Google Play store called Servers Ultimate, and it enables all manner of servers, from httpd/Web, to Samba/CIFS, even VNC and SSH. There are optional "packs" of servers sold as add-ons to the main application, so you even can host your

own Git server on an old phone, if you chose to do so.

Wireless Touchpad

Install the free Gmote application on your device and its companion software on your computer, and you can turn your old phone or tablet into a wireless touchpad for your desktop PC. If you pair it with a laptop, you could use an old phone as a wireless presentation remote too!

Use Tasker to Automate

Mundane Things

Tasker is an automation engine for Android devices. If you've ever tried the IFTTT Internet service, Tasker is similar to that, without the service, as it runs solely on your device. Tasker's built to run on your primary device, either a handset or tablet, but it's got some value for use on a secondary device. Imagine an alarm clock that would wake you up with a random song from your favorite playlist, and then do text-to-speech

Alternate ROMs for Old iOS Devices

iOS devices can present their own challenges, but if you have one lying around, you still can sideload an alternate ROM onto it—assuming it's a supported model. Right now, Whited00r is the only option you've got, and you have to be on either an old iPhone 2G, 3G or iPod Touch 1G or 2G. It tries to backport a lot of Apple's iOS 7 look and feel as well as features to older hardware, and it succeeds...sort of. If you've got an older iOS device just sitting in a drawer though, and you're curious, it's worth a test drive.

conversion on your incoming e-mail and read you your messages while you were waking up. That's the kind of thing that's possible with Tasker.

Conclusion

Now that I've shown you a few ideas of what's possible with old devices you may have sitting in a drawer, maybe you'll think of a few ideas of your own. Get those devices out and start putting them to use! These little guys have as much processing power as full-fledged

computers did just a few years ago, so they're still able to get some real work (or play) done. Feel free to experiment, and see what you can come up with! ■

Bill Childers is an IT Manager in Silicon Valley, where he lives with his wife and two children. He enjoys Linux far too much, and probably should get more sun from time to time.



Send comments or feedback via <http://www.linuxjournal.com/contact> or to ljeditor@linuxjournal.com.

Resources

Plex Media Server: <http://plex.tv>

IP Webcam: <https://play.google.com/store/apps/details?id=com.pas.webcam>

Presence (iOS): <https://itunes.apple.com/us/app/presence-by-people-power/id618598211?mt=8>

Andchat: <http://www.andchat.net>

Colloquy (iOS): <http://colloquy.mobi>

"Seamlessly Extending IRC to Mobile Devices" by Bill Childers, *Linux Journal*, March 2012:
<http://www.linuxjournal.com/content/seamlessly-extending-irc-mobile-devices>

iCade: <http://www.ionaudio.com/products/details/icade>

iCade Jr.: <http://www.ionaudio.com/products/details/icadejr>

Daily Roads Voyager: <https://play.google.com/store/apps/details?id=com.dailyroads.v>

Motorola Lapdock: <http://www.motorola.com/us/Motorola-Lapdock-100/73945.html>

Cyanogenmod: <http://www.cyanogenmod.org>

XDA-Developers: <http://forum.xda-developers.com>

Servers Ultimate: <https://play.google.com/store/apps/details?id=com.icecoldapps.serversultimate>

Gmote: <https://play.google.com/store/apps/details?id=org.gmote.client.android>

Tasker: <http://tasker.dinglisch.net>

Whited00r for iOS: <http://www.whited00r.com>

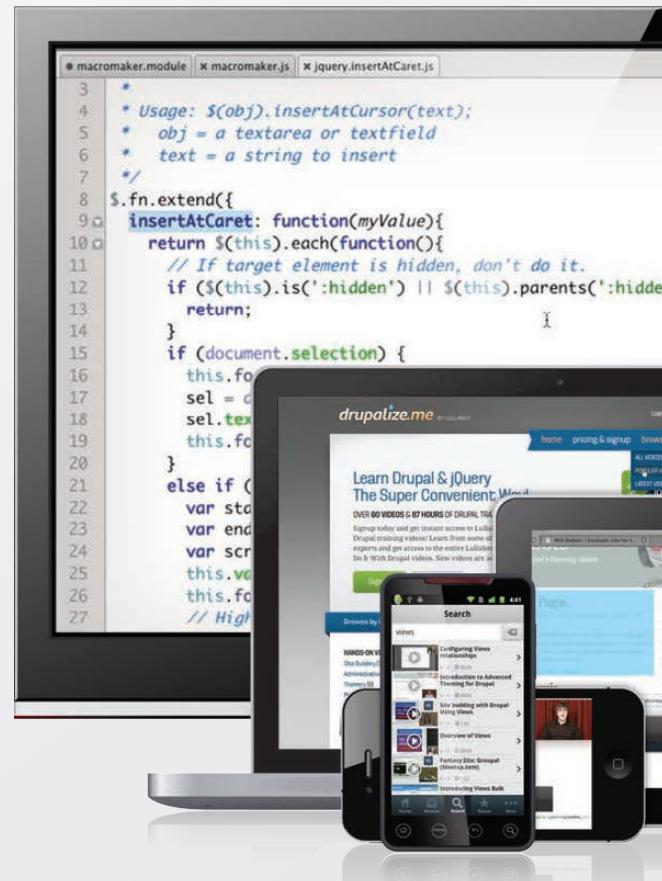


Instant Access to Premium Online Drupal Training

- ✓ Instant access to hundreds of hours of Drupal training with new videos added every week!
- ✓ Learn from industry experts with real world experience building high profile sites
- ✓ Learn on the go wherever you are with apps for iOS, Android & Roku
- ✓ We also offer group accounts. Give your whole team access at a discounted rate!

Learn about our latest video releases and offers first by following us on Facebook and Twitter (@drupalizeme)!

Go to <http://drupalize.me> and get Drupalized today!



WEBCASTS



Learn the 5 Critical Success Factors to Accelerate IT Service Delivery in a Cloud-Enabled Data Center

Today's organizations face an unparalleled rate of change. Cloud-enabled data centers are increasingly seen as a way to accelerate IT service delivery and increase utilization of resources while reducing operating expenses. Building a cloud starts with virtualizing your IT environment, but an end-to-end cloud orchestration solution is key to optimizing the cloud to drive real productivity gains.

> <http://lnxjr.nl/IBM5factors>



Modernizing SAP Environments with Minimum Risk—a Path to Big Data

Sponsor: SAP | **Topic:** Big Data

Is the data explosion in today's world a liability or a competitive advantage for your business? Exploiting massive amounts of data to make sound business decisions is a business imperative for success and a high priority for many firms. With rapid advances in x86 processing power and storage, enterprise application and database workloads are increasingly being moved from UNIX to Linux as part of IT modernization efforts. Modernizing application environments has numerous TCO and ROI benefits but the transformation needs to be managed carefully and performed with minimal downtime. Join this webinar to hear from top IDC analyst, Richard Villars, about the path you can start taking now to enable your organization to get the benefits of turning data into actionable insights with exciting x86 technology.

> <http://lnxjr.nl/modsap>

WHITE PAPERS



White Paper: JBoss Enterprise Application Platform for OpenShift Enterprise

Sponsor: DLT Solutions

Red Hat's® JBoss Enterprise Application Platform for OpenShift Enterprise offering provides IT organizations with a simple and straightforward way to deploy and manage Java applications. This optional OpenShift Enterprise component further extends the developer and manageability benefits inherent in JBoss Enterprise Application Platform for on-premise cloud environments.

Unlike other multi-product offerings, this is not a bundling of two separate products. JBoss Enterprise Middleware has been hosted on the OpenShift public offering for more than 18 months. And many capabilities and features of JBoss Enterprise Application Platform 6 and JBoss Developer Studio 5 (which is also included in this offering) are based upon that experience.

This real-world understanding of how application servers operate and function in cloud environments is now available in this single on-premise offering, JBoss Enterprise Application Platform for OpenShift Enterprise, for enterprises looking for cloud benefits within their own datacenters.

> <http://lnxjr.nl/jbossapp>

WHITE PAPERS



Linux Management with Red Hat Satellite: Measuring Business Impact and ROI

Sponsor: Red Hat | **Topic:** Linux Management

Linux has become a key foundation for supporting today's rapidly growing IT environments. Linux is being used to deploy business applications and databases, trading on its reputation as a low-cost operating environment. For many IT organizations, Linux is a mainstay for deploying Web servers and has evolved from handling basic file, print, and utility workloads to running mission-critical applications and databases, physically, virtually, and in the cloud. As Linux grows in importance in terms of value to the business, managing Linux environments to high standards of service quality — availability, security, and performance — becomes an essential requirement for business success.

> <http://lnxjr.nl/RHS-ROI>



Standardized Operating Environments for IT Efficiency

Sponsor: Red Hat

The Red Hat® Standard Operating Environment SOE helps you define, deploy, and maintain Red Hat Enterprise Linux® and third-party applications as an SOE. The SOE is fully aligned with your requirements as an effective and managed process, and fully integrated with your IT environment and processes.

Benefits of an SOE:

SOE is a specification for a tested, standard selection of computer hardware, software, and their configuration for use on computers within an organization. The modular nature of the Red Hat SOE lets you select the most appropriate solutions to address your business' IT needs.

SOE leads to:

- Dramatically reduced deployment time.
- Software deployed and configured in a standardized manner.
- Simplified maintenance due to standardization.
- Increased stability and reduced support and management costs.
- There are many benefits to having an SOE within larger environments, such as:
 - Less total cost of ownership (TCO) for the IT environment.
 - More effective support.
 - Faster deployment times.
 - Standardization.

> <http://lnxjr.nl/RH-SOE>

Practical Tiny Core in the Fire Service

Linux in the Firehouse. Tiny Core helping firefighters save time, money and lives.

WILFREDO CRESPO

I'm sure many of you have at least heard of Tiny Core Linux—legends of how small it is, how little it takes it to run a system with it and even now how it's been ported to run on Raspberry Pi. It's an esoteric minimalist distribution. There was a very good September 2011 write up about it in *Linux Journal* by Joey Bernard (<http://www.linuxjournal.com/article/11023>).

I came to Tiny Core Linux after a protracted quest to find a good solution to a problem I've had. I am a Firefighter and EMT with the Bushkill Fire Company in Pennsylvania. Being an all-volunteer fire department while also being the primary service provider for our entire coverage area poses some

unique challenges. When a dispatch comes through, fire engines and rescue apparatuses are expected to get out the door and on the road, quick.

Part of the logistical acrobatics that we perform every call is figuring out who is even coming. Different personnel bring different skills, skill levels, responsibilities and capabilities. Some are specialists in operating particular and specialized pieces of equipment, while others may bring specific types of know-how in techniques. Depending on the call, officers like to know who they have arriving as they plan out the best approach to tackle the given emergency. It's nice to know, down right imperative to know, that your

best vehicle extrication technicians are responding to a motor vehicle accident involving two cars with a possible entrapment.

Technology has been at the forefront of this ongoing struggle in emergency services to solve a plethora of problems. A few companies have released Web apps designed for the fire service to tackle the “who’s coming” problem. These systems consist of a Web page that acts as a dashboard and is displayed in the station. Each responder dials a number that registers his or her name as responding to the call. Officers then can assess what their man power is like and decide quickly whether they need more resources, simply by looking up at the monitor in the station. These apps do more than that, but this is the crux of what they are designed for.

It sounds fantastic, and indeed it is an enormous help and resource. Being the go-to IT guy at my firehouse, I had fallen in the enviable position of making this system work for us. The trivial solution is, of course, just to fire up Windows with IE and let the monitor sit there—in fact, the vendor suggested this to me. I suppose when you’re trying to sell something as easy to use, that’s what you do. That solution, needless to say, was

unsatisfactory. From a budgetary standpoint, I was encouraged to keep costs down. The first decision and the easiest decision was to use Linux. I just shaved off the cost of the Windows license.

The initial requirements and constraints became as follows:

1. Have a low-power computer to run the Web browser—the smaller the better.
2. The monitor must be 32" or larger.
3. The computer and monitor are in different rooms.
4. The building is on a generator backup, and the system must be able to endure the minimal power loss between the time the main power goes out and the generator activates.
5. The monitor needs to be mounted 10–12 feet up so it is easily visible by everyone entering the building.

I went ahead and purchased a mini PC for about \$160 (a Zotac Mag in my case). Given the distance requirement, with the monitor being in a different room from the computer, I decided to use an HDMI connection with an

active range extender (Cat 5 Ethernet to HDMI extender). The monitor is a 32" 720p LCD TV.

My first instinct was to grab the latest Ubuntu and install it, and that's exactly what I did. At the time, perhaps two or three years ago, it was Ubuntu 11. I naively installed it and set up the wireless and made Firefox load on startup. I also set Firefox to save the session so it always booted with the same session open. Making sure to turn off screen blanking was important as well. Everything was working great, or so I thought. As time went by, I started encountering problems I hadn't thought of. Let me go through a few of them here.

The first most glaring problem is what happens when the Internet connection goes down. Ubuntu's Wi-Fi management is built for a desktop environment. It does a great job at performing in that environment. However, for what is essentially a kiosk, it has some drawbacks. It still requires some degree of interaction, as with most things designed for a desktop. The most succinct way to put it is that it is just too much operating system for my needs. I don't need Unity; I don't need a compositing windowing system; I don't need to be badgered about updates; I don't need a fancy packaging system, and

I don't need to hunt down where each setting I want to change is. My approach was just wrong. I found myself solving problems by dismantling the operating system. I'm sure many of you have done this as well. Finally, it was corruptible. The setup could change if you wanted it to. This is fantastic for a desktop, because it means I can customize it to whatever I want. Conversely, this is terrible for a kiosk, because it means everyone who gets their hands on it, even with good intent, can change settings and give you that much more of a headache when you need it to go back to exactly how you originally set it up. This too happened. A fellow firefighter would come in and wonder if the page was refreshing correctly, grab the keyboard and mouse, and the next thing I knew, the browser was set to start on yahoo.com with a sports ticker. This is simply the natural consequence of having a publicly-facing system.

After constantly fixing these small issues, I'd had it. I decided there must be a better way. I started from scratch and jumped on Distrowatch.com and started my search for a better-suited distro. The few I'd considered were Vector Linux, Puppy Linux, Damn Small and SliTaz. Each of these is an amazing distro, and this is by no

means an exposition on what each is capable of. This is just my account of what I did.

I finally settled on the reality that I will make a distro that does what I need to do—no more and no less. In other words, I wanted Just Enough Operating System (JeOS). Eventually, I settled on Tiny Core Linux. It lets me do just that.

I probably should regress for a second, lest I offend the Tiny Core experts out there. Tiny Core Linux should not be thought of as a distribution. It should be thought of as a set of tools for building your Linux system however you see fit. I needed just enough to get this particular job done.

Tiny Core is available on its Web site in three flavors. Core, the smallest of the three, is just 9MB. Core provides a command-line interface. TinyCore provides a basic FLTK/FLWM GUI, and finally, CorePlus provides a choice of seven different window managers, Wi-Fi support, remastering tools and support for non-US keyboards.

I installed CorePlus onto a USB drive and fired it up. I got to understand how Tiny Core works, and make no mistake, there is a pretty steep learning curve. For my purposes, it was more than worth it. Besides, who doesn't like to learn new things? One

of the most important things about Tiny Core Linux is that it is non-corruptible. I can set it up exactly how I see fit, and it always will boot to that state. Nothing is saved. Tiny Core boots and runs entirely in a RAM disk. It opens the image file you create, loads it into memory and runs. Whether you boot it from a hard drive or USB drive, it simply loads the image file with all the programs, settings, files and so on that you built in to it, straight into memory. Tiny Core uses the concepts of extensions to install applications. There is an excellent write up on the Tiny Core Web site explaining extensions. For more intricacies on Tiny Core, there is also the excellent *Linux Journal* article I mentioned earlier, which I suggest you look at.

With a better idea of how Tiny Core Linux worked, I decided it was the best option for me. So, I got started setting up the system how I wanted:

- Choose an X server: Tiny Core defaults to xvesa. I didn't need anything fancy but decided to go with Xorg due to ease of configuration. HDTVs come in two resolutions 1920x1080 (1080p) and 1280x720 (720p). So configuration wasn't too big an issue. My display was a 720p.

- Choose a window manager: among the choices available, I chose Joe's Window Manager. Go with whatever is comfortable and suits your needs.
 - Pick my extensions: I went with the following extensions: Firefox as my chosen browser; Ixrandr, for configuring resolutions under Xorg; wifi, includes as dependencies all the required Wi-Fi libraries and firmware Xorg, as stated previously.
 - Optional: wicd, graphical configuration for Wi-Fi.
- Really, the only way to tailor your system completely is by trial and error. You live, you learn, and you become better. Once I had the extensions I wanted, I chose to have them loaded

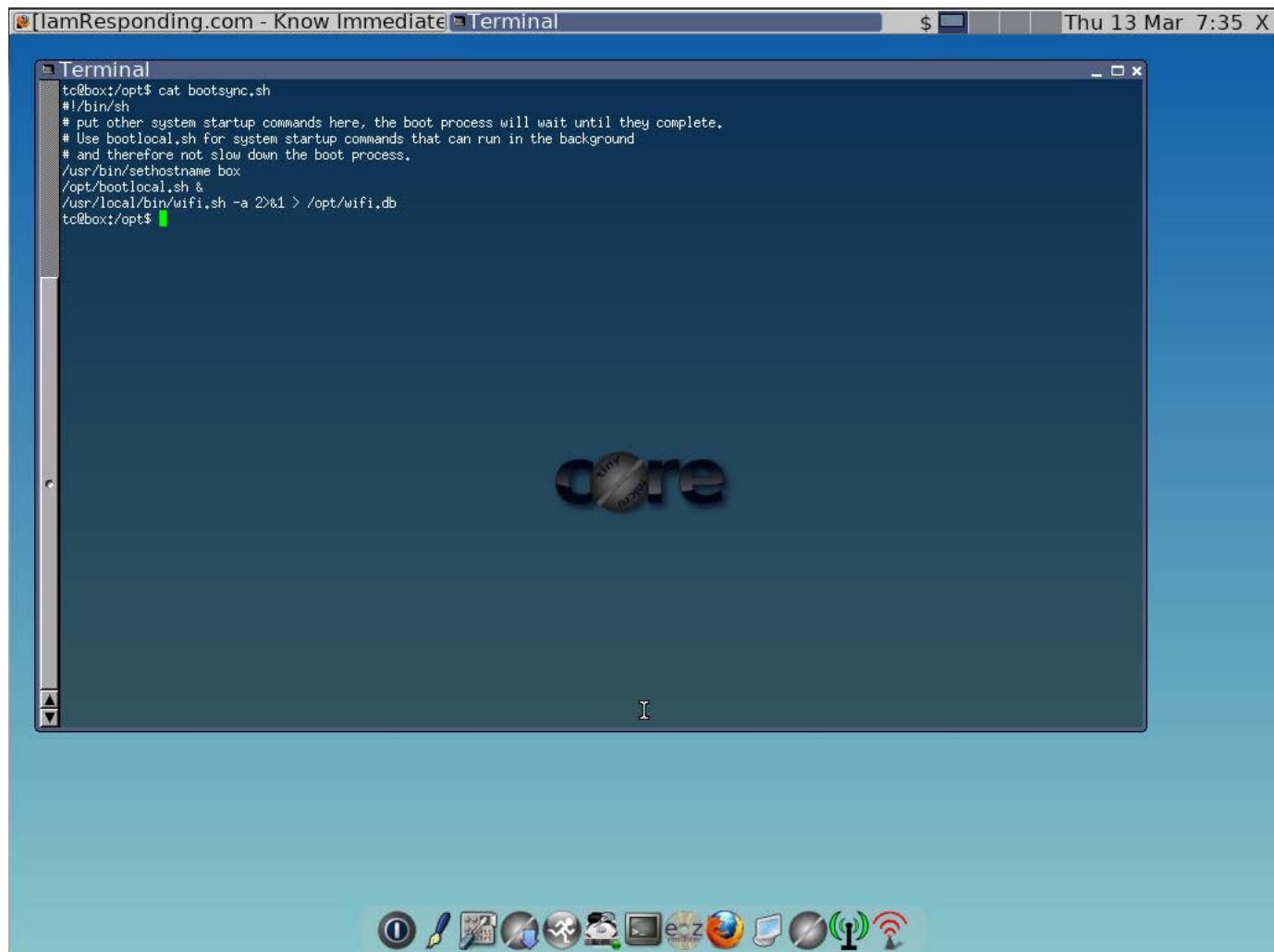


Figure 1. The bootlocal.sh is called in this file. Here I am loading the Wi-Fi settings from opt/wifi.db. After it connects, the boot process continues.

inside initrd apps on boot. The Tiny Core Wiki does a great job of explaining the differences. I chose this because there aren't too many extensions to load, and they all fit in the RAM of the machine I'm using. This also frees from me doing excessive writes to a Flash drive should the system be running one, or from even requiring a hard drive. The system I'm using has a 160GB hard drive that I installed. It also saves on power usage.

I was lucky enough to have an older laptop I could experiment with as I tailored my system. I also used VirtualBox on my Ubuntu box. With all my extensions chosen, I remastered and ended up with a system that fits in less than 70MB. You can get as pedantic as you want with this, but I needed to get it working quickly; although admittedly, I will go back one day and see how small I can get it!

The system booted and everything

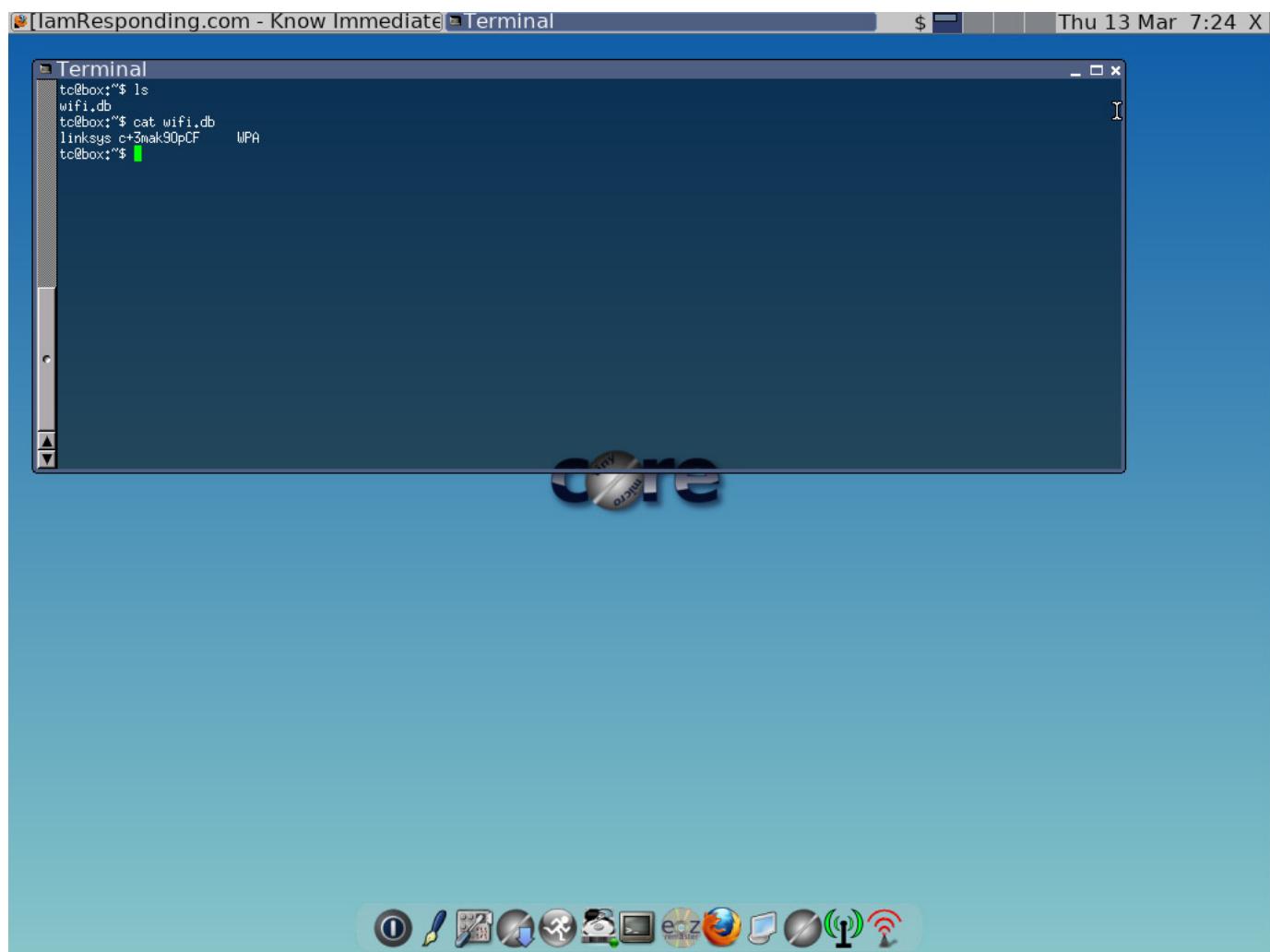


Figure 2. An example wifi.db—notice the SSID first and password, in plain text, second.

came up fine. Granted, this came after about four remasterings. Like I said, it's a learning process, and you tweak as you go. Now came the work of making it do what I wanted it to do on boot. The first issue was I needed Wi-Fi. I used the wifi utility to connect to a Wi-Fi network. The standard command-line Wi-Fi utility Tiny Core's wifi extension provides saves the SSID and password settings you provide in a wifi.db file.

Once I had saved my settings, I set Wi-Fi to connect automatically at boot time. Tiny Core has two places for loading commands and scripts upon startup, bootlocal.sh and bootsync.sh, both in /opt. There is a subtle important difference between the two. bootsync.sh, as the name might imply, isn't asynchronous; commands are run, and they block the boot process until they are finished. The bootlocal.sh is run from bootsync.sh in the background. I needed the Wi-Fi to be loaded before anything attempts to use the Internet. It would be catastrophic if the browser loaded and tried to load the page while the Wi-Fi script still was attempting to connect. One very important gotcha to note is Tiny Core's wifi.sh script will take down the WLAN interface if it does not connect. This confused me at first, making me think I had a kernel module issue. Alas, I didn't, so I saved

the proper Wi-Fi credentials, and then I was up and running once again.

The next operating system configuration specific to deal with was the DPMS and screensaver settings—after all, I couldn't have the screen blanking every 15 minutes on a station display. There are a few ways to disable auto-blank and sleep. This is what I did: I put the following into the ~/.xsession file (being they are X-specific options, that's where they belong):

```
xset s off  
xset s noblank  
xset -dpms
```

The `s off` option shuts off the screensaver functionality; `noblank` tells it not to blank the screen, and finally, `-dpms` tells Xorg to disable DPMS Energy Star features.

I'm almost there. The final, albeit big, piece of this puzzle is to have the browser load on startup with the correct settings. To do so, I put a script in `~/.X.d`. I simply made a filename of `firefox` and put `firefox &` inside it. It doesn't need to be executable, so there's no need to play with its permissions. Now the browser loads on startup. I could write an entire article on configuring browsers, but here is a rundown of how it went.

I used the Session Manager extension for Firefox to save the session I wanted. Session Manager has settings for auto-loading a session; tweak those how you see fit. I also made sure Session Manager would not overwrite my session should something go awry. In addition to Session Manager, I installed the following:

- FF Fullscreen: starts browser in full screen.
- Reload Every: refreshes the page at specified intervals.
- Memory Restart: automatically restarts the browser when its memory usage reaches a threshold.

After setting up the browser just how I wanted, it was time to save my settings. I've already said that Tiny Core was incorruptible; what I mean by this is that it starts in the same state all the time. Your settings and anything else done are not saved on shutdown. Everything resides in RAM. When it boots up again, it simply decompresses its image file straight into memory. So what do you do when you need to save settings? Tiny Core lets you save any changes you make to the filesystem in the form

of a backup. The backup is simply a tarball of whatever you specify. Tiny Core then can be configured to restore this file on each boot. Tiny Core includes a backup utility. The backup utility creates the tarball file containing anything you specify in the /opt/.filetool.lst and excluding anything in the /opt/.xfiletool.lst. My file include list consists of:

- home
- opt

My home and opt directories are both included in the backup, but I don't need everything:

- Cache
- cache
- .cache
- XUL.mfasl
- XPC.mfasl
- mnt
- ./adobe/Flash_Player/AssetCache
- .macromedia/Flash_Player

- .opera/opcache
- .opera/cache4
- .Xauthority
- .wmx
- *.iso

Everything here is default, with the

exception of the *.iso. This is because the remastering process can create a bootable ISO for you, and I don't want it included in my backup. Make sure any setting you want saved is in a file included in the backup. In my case, the Firefox extensions and settings are all contained in .mozilla in the home directory. You even can do a dry run to see what files would be saved given the rules you provided. Once I

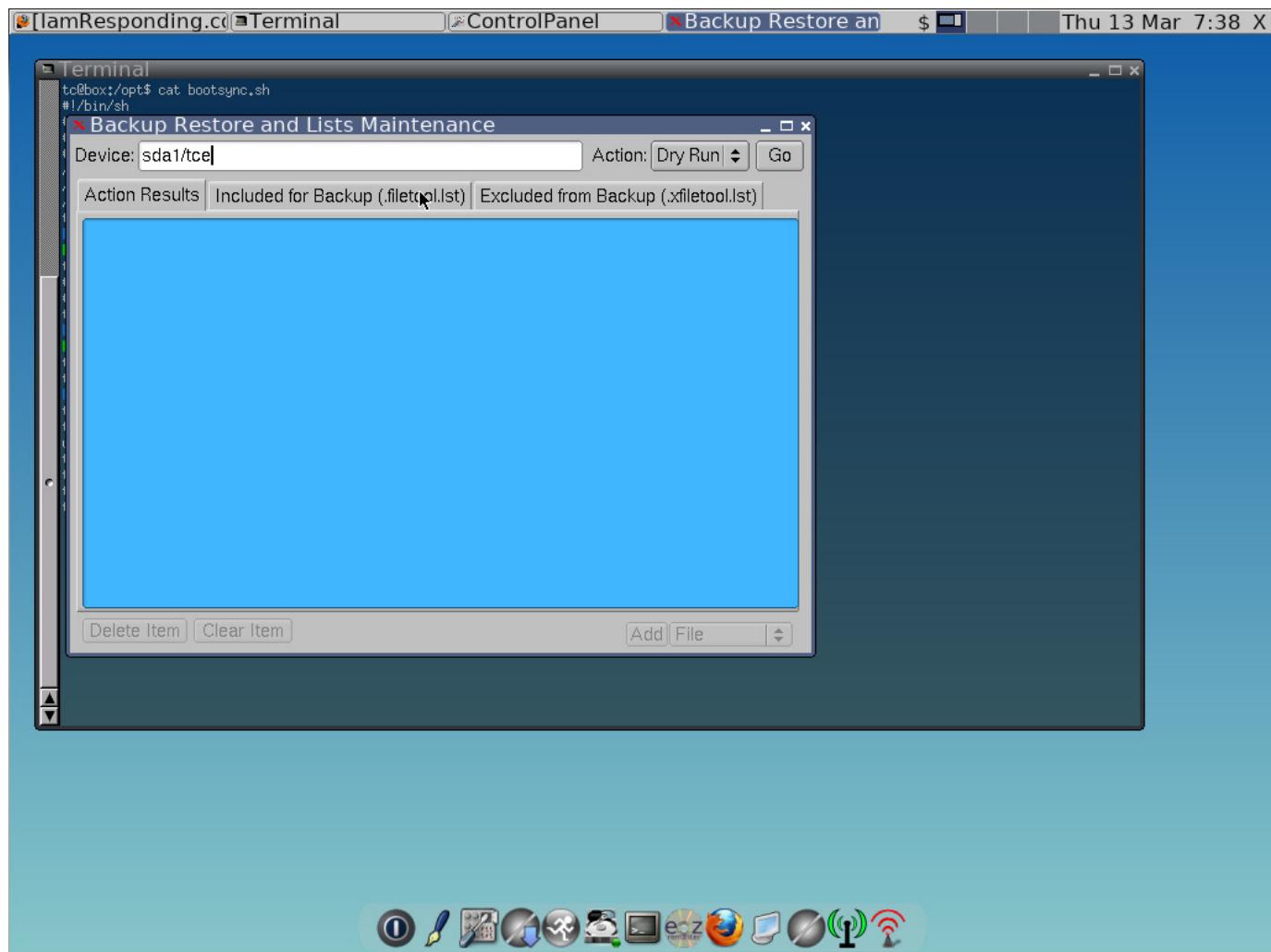


Figure 3. The backup utility is where you create your .tgz backup file. The dry run option is always a good idea so you know exactly what it's going to do.

was satisfied, I did a backup, and my resulting backup file, mydata.tar.gz, was around 800KB.

Now that I had my backup file, I was ready to remaster. I remastered using the included utility, but there are other ways to remaster (a quick trip to the Tiny Core documentation will show you them). During the remaster process, I chose my extensions (as outlined previously)

and the boot codes I wanted.

Tiny Core has a number of boot codes. Let me review some important boot codes here. You can define the location of the home directory and the opt directory. Because the systems I was installing Tiny Core on have hard drives, I specified the boot codes as follows:

`opt=sda1 home=sda1`

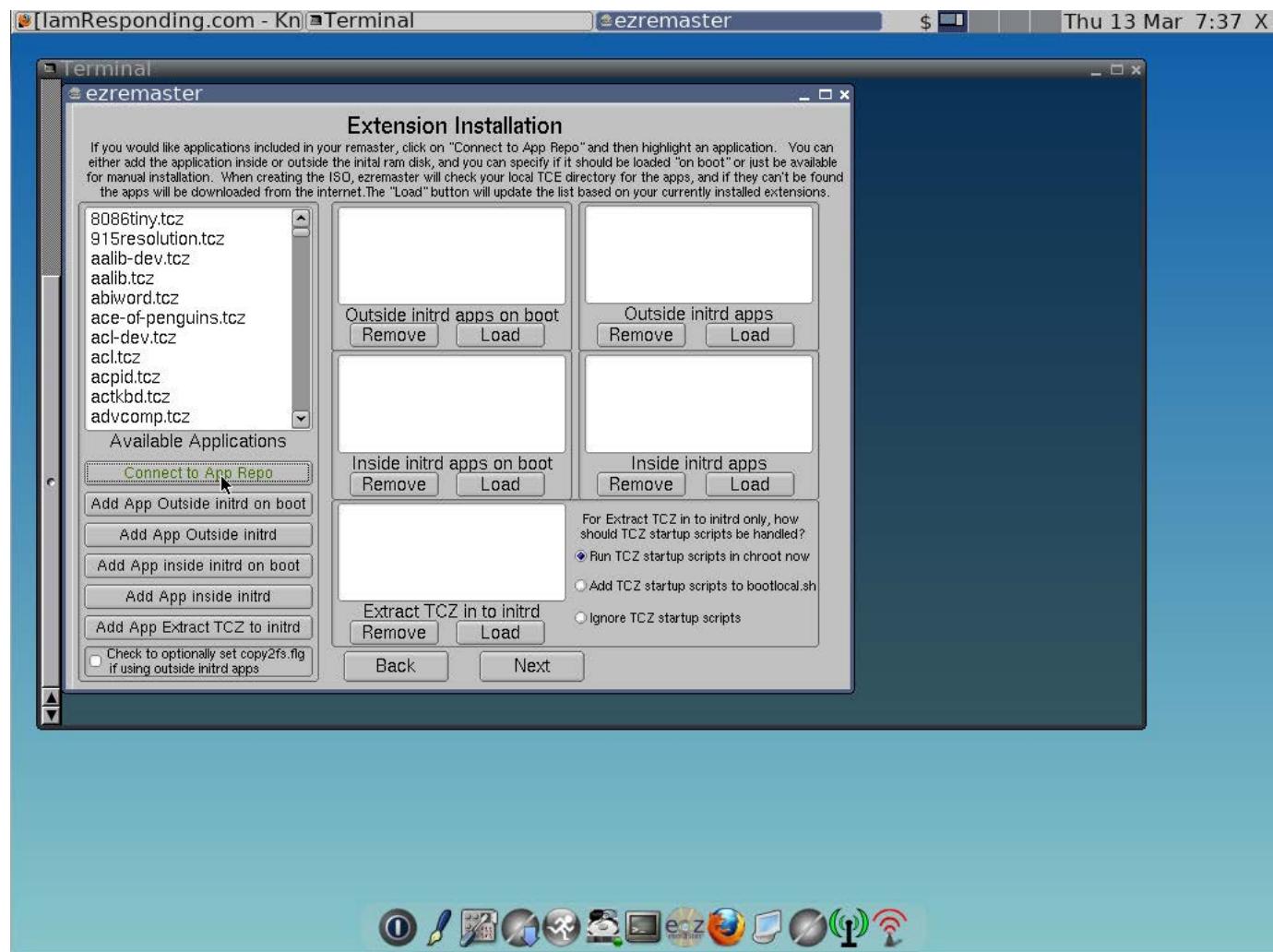


Figure 4. Choosing what to load and the best way to load it will be a big part of your project. If need be, you can create your own extensions.

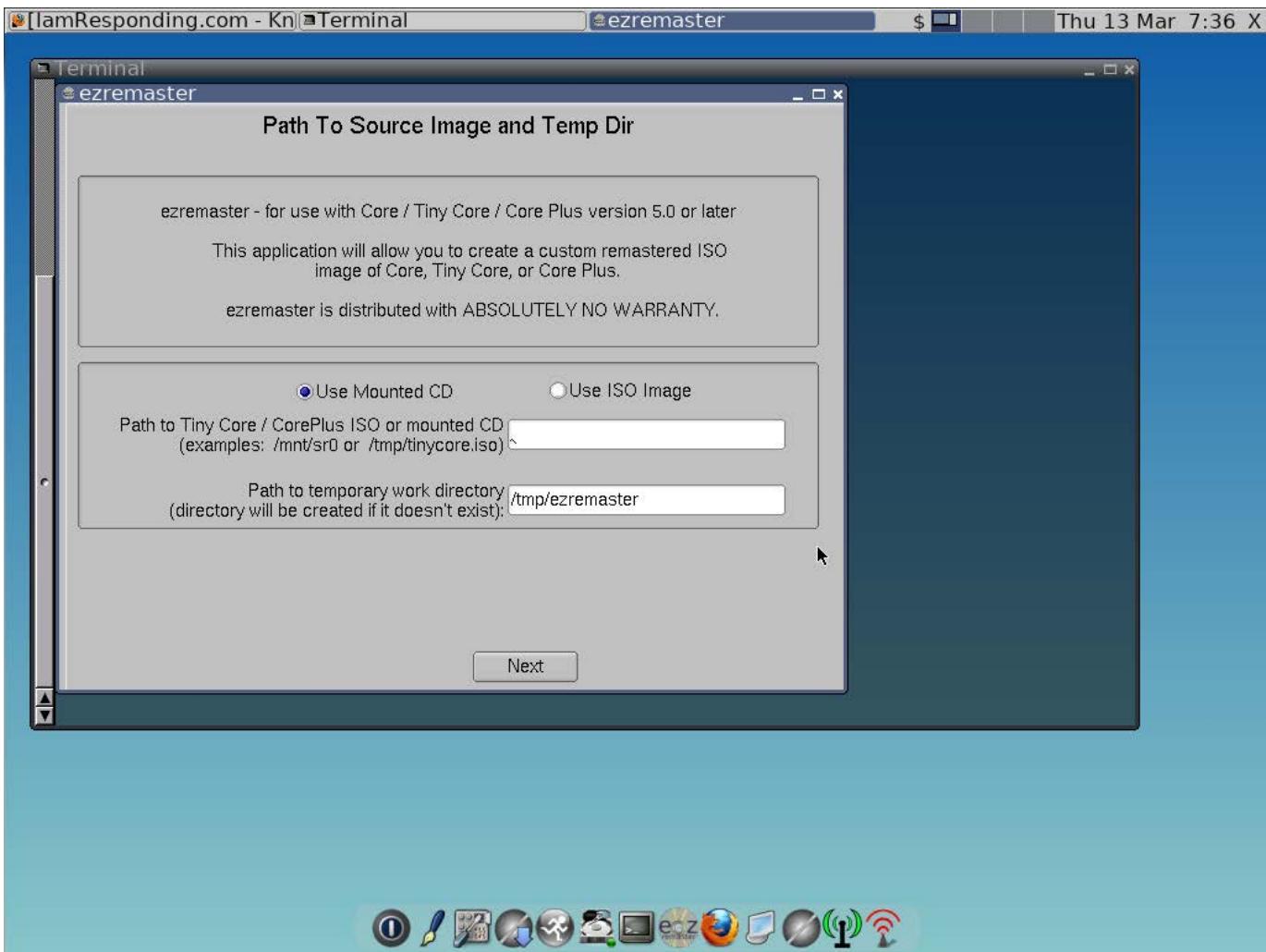


Figure 5. The Ezremaster utility makes it very easy to do your own remaster. This was a huge time-saver in my project.

There is a norestore option listed and described in the documentation that tells Tiny Core not to restore any backups. Because I wanted to have my exact settings restored on each boot, I did not use that option. During the Ezremaster process, I specified my backup file. I also told the Ezremaster process to create an ISO file for me.

With everything remastered, I loaded my new Tiny Core build onto a USB

drive. It booted perfectly. Now with a working copy of what I needed, I went ahead and installed it on our mini PC. The installation process involves running the Tiny Core installer. I didn't need any particular partitioning scheme, so I just used the entire drive. I specified sda1 as my opt and home as previously stated. Once the system was installed, I placed my backup tarball on the newly created filesystem.

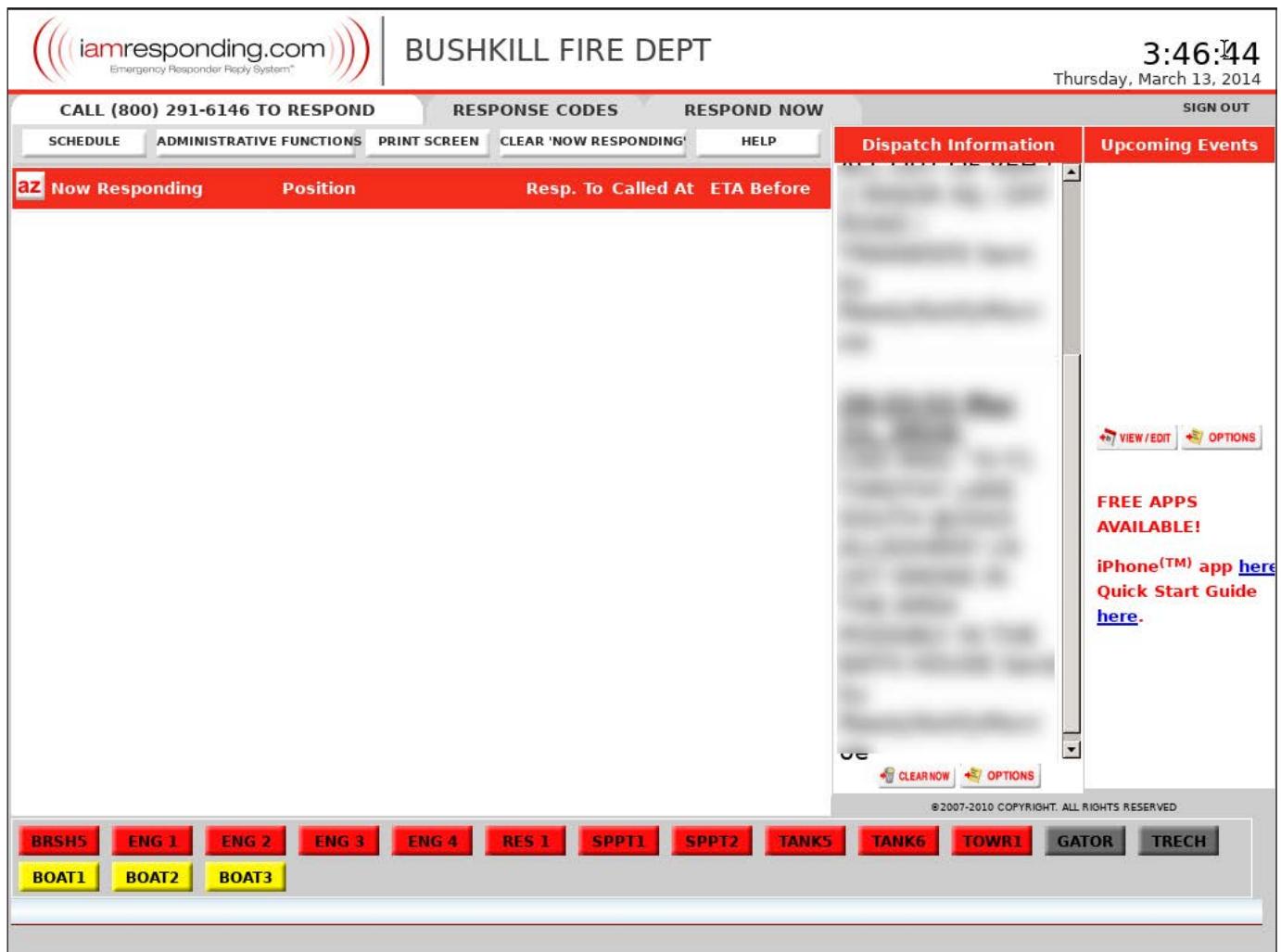


Figure 6. This our display board, showing our call, unit and arrival information

Then, I removed my USB thumbdrive and gave it a reboot. I now had my custom, Tiny Core Linux display kiosk fully working. Every reboot provided me with a clean environment tailored to my specifications.

The time in learning how Tiny Core works was well worth the outcome. There have been no problems, and the system is rock-solid. Even if the power goes out, it will start up with

our page loaded displaying dispatch information and who is arriving to the call should one come out. It needs no administration because the settings do not change. It is Just Enough Operating System for me!

Here are some things I'd like improve upon for my next version:

- Move settings over from a backup file to an extension. The Tiny Core



Figure 7. Our display at the Firehouse: LED TV with an active HDMI-to-Ethernet extender—all up and running our custom Tiny Core Linux!

documents outline this procedure, and I can't wait to try it. This should streamline the process of loading settings and user-created files.

- Create display profiles for 1080p and 720p. With two profiles available for both TV types, they can be selected easily should Xorg not be able to auto-detect the resolution of a particular display.
- Remote monitoring—install something

like monit to enable monitoring.

- Create an extension with initial setup scripts. The scripts would set information, such as Wi-Fi credentials, monitor types and so on, and handle persistence for these settings.
- Integrate with the UPS. Install the NUTS package for interfacing with the UPS, although this might be a needless complication. I'm sure



Figure 8. This is a forward view of our bays. When entering the station, crews getting into these apparatuses are able to view the display (not pictured).

someone will have a use for it.

Tiny Core Linux has saved us money, saved me some sanity and keeps our fellow firefighters informed and ready to perform the job. Firefighters must be ready 24/7, so technology should help us accomplish that goal and not get in our way. Linux has proved to more than capable, and I will be making changes and improvements as I see fit. I hope you try your hand at it

as well and let us know how it goes. ■

Wilfredo Crespo is currently a Volunteer Firefighter at Bushkill Fire Company, in Pike County, Pennsylvania. His current rank is Lieutenant, and he's been with the department for approximately six years. He's also a state-certified EMT. He worked in the finance industry as a software engineer in New York City, and then he did freelance work. Now he's launching a new Web application for the Fire Service called Gearax (<http://gearax.com>), which is an issue-tracking system for fire apparatus (trucks) maintenance—basically a fleet maintenance application. He graduated with a BA in Computer Science from New York University in 2007.

Is the Private Cloud a Real Cloud?

The private cloud is a term used for a flexible proprietary computing architecture that provides compute, storage and network resources to various business units of an organization behind a firewall.

MITESH SONI

Consider this traditional scenario: in today's competitive world, dynamic business requirements need flexible and rapid provisioning of IT resources. Along with flexibility, traditional IT environments need new resources to support the dynamic workloads of applications. It is a very likely scenario where separate business units within the same organization manage their own computing resources, such as hardware, software, storage, networking and applications, creating silos of computing infrastructure. Power, space and cooling of corporate data centers are some of the major challenges faced by organizations today. Applications running on increased-capacity hardware are

not able to utilize available resources in an efficient manner in traditional data centers. In addition to this, expert human resources are needed to maintain and manage the whole environment.

Fortunately, developments in computing have opened many possibilities for all types organizations, including education and government. Until recently, a main concern was deployment and hosting of applications considering cost or inadequate acquisition of compute, storage and network resources with a fixed capacity for managing unexpected application traffic demands. Other challenging tasks were to manage the installation,

maintenance and configuration of the whole stack that supported the respective application. In the past few years, cloud computing has been considered as a boon for organizations that were facing these issues in traditional computing environments. Application design, deployment, maintenance and hosting have become easier, agile and less costly with the use of the pay-per-use charge-back model provided by cloud providers. IT executives are considering transitions from traditional data centers to cloud infrastructures for cost reductions and more agile IT environments.

What Is Cloud Computing?

According to the National Institute of Standards and Technology (NIST), cloud computing is a model for enabling convenient, ubiquitous and on-demand network access to a shared pool of configurable computing resources, such as servers, networks, storage, applications and services, that can be provisioned and de-provisioned rapidly with minimal management effort or service-provider interaction. These cloud models support availability and are composed of five essential characteristics, three service models and four deployment models. The four deployment models

are the public cloud, private cloud, community cloud and hybrid cloud.

Cloud computing provides an agile environment, faster time to market, infinite scalability and services. Different applications, depending on their type, have different scaling and performance requirements. The cloud environment is perfectly suitable for applications where users have dynamic and competing traffic requirements. However, the five characteristics, three service models and four deployment models create a hazy situation when you want to use the cloud to deploy various kinds of applications. Public cloud services from third-party service providers are considered to be a windfall for SMBs, but for large enterprises, application security, compliance adherence and CIA (Confidentiality, Integrity and Availability of data) are fundamental, so it is difficult to convince them to adopt the public cloud.

Primarily, business data is the heart of any organization. All compliance requirements based on regulatory constraints and risks revolve around that critical and sensitive data. Security, compliance, vendor lock-in and interoperability are on the rise and are obstructing public cloud adoption.

The question one should ask is,

"If public cloud service providers can deliver services efficiently and realize all benefits, including cost benefits and efficient resource utilization, then why can't organizations do this with existing expertise, an investment in resources and the availability of private cloud products that help organizations build cloud environments under their own control?"

It seems obvious that the journey to the cloud will take place in environments where organizations have more trust and control. Essentially, by building a private cloud and then integrating it on-site, with private cloud data or applications in the public cloud, in the process, they can build a hybrid cloud by utilizing existing resources.

The private cloud refers to a flexible proprietary computing architecture that provides compute, storage and network resources to various business units of an organization behind a firewall.

The Private Cloud—Is It Really a Cloud?

The private cloud provides an in-house cloud infrastructure, absolute control over resources, more security and privacy, and manageable compliance to regulatory requirements. It also

requires a huge capital investment and the expertise for building a private cloud environment and maintaining that infrastructure. In the IT industry, thought leaders and other experts have different points of view on private clouds. Initially, the experts made their points of view public about the private cloud, saying the idea that organizations can run an internal cloud as efficiently as public cloud service providers does not mean it is a real cloud. Let's consider the various arguments based on economy of scale, cost sharing and multi-tenancy.

Organizations may not realize one of the primary benefits of cloud computing: economies of scale. Public cloud environments provide agility and scalability that organizations require to survive in this modern era of wider customer bases. However, organizations that choose to adopt private rather than public clouds may miss out on the same benefits. Hence, it often becomes a major point of discussion that the private cloud does not carry the similar value propositions as the public cloud.

One significant question raised by many public cloud supporters was regarding cost sharing. Who is sharing costs with the private cloud owner? According to them, if all the costs are managed by a single organization, it

is not a cloud. If a single organization owns the computing resources, it is not a cloud.

Multi-tenancy is fundamental to the cloud's proposition. Multi-tenancy and shared resources are two of the defining characteristics of cloud computing environments in which compute, storage and network resources are shared between multiple users. Organizations deploy the application and store data in the cloud, driving down the cost for all. Organizations shift the resource management burden and risks to the service provider. This is why the cloud is so exciting. One perspective is, if it is not multi-tenant, it fails the test, and it is not a cloud. It is just an environment that delivers remote resources, and that has been around for years in organizations.

In a nutshell, an organization that builds a private cloud will spend more money and achieve less benefits compared to the public cloud, which needs expertise to build it, but it provides greater control and better perceived security.

The private cloud has come a long way to gain momentous footing in the market. Public cloud vendors who once criticized private clouds are now providing features for interoperability with private cloud platforms. So,

it's a reasonable assumption that the private cloud is a real cloud. Still, it's better to muse over general features and aspects than to draw a firm conclusion. The public cloud has changed performance expectations for IT dramatically, but there are industries where the latency effect of any public cloud transaction will lead to intolerable service levels, loss of revenue and so on. These concerns have captured the attention of IT leadership. It is a strategic decision to build your own cloud that requires stakeholders to think meticulously and exclusively. It is the need of the hour that organizations must enable self-service and become on-demand providers of infrastructure, platforms and applications for their internal business units. Speed and agility, not cost reduction, drive private cloud implementations.

Now, let's take a step forward in the discussion by revisiting the history of Amazon's Public Cloud. How did Amazon start its operations and offerings of the public cloud? It all started with the intent to gain flexibility and agility in the traditional environment. For those same reasons, IT organizations are building private clouds today.

Many public cloud providers have announced their connection with

various private cloud vendors. Now the question is why has everyone accepted the existence of “private clouds”? Many organizations still have concerns regarding putting their intellectual property or critical applications in the public cloud or off-site or beyond the organization’s firewall. They aptly assessed the market situation and realized that they stand to gain by partnering in the private cloud arena.

It is a simple case of the first step in the direction of acceptance of the existence of private clouds and the foresight of hybrid cloud implementations.

According to a poll at a recent Gartner data-center conference, 47% of respondents want the ability to manage clouds on-premises and off-premises centrally by 2015. There will be more private cloud deployments in coming years, because OpenStack, CloudStack, Eucalyptus, VMware, BMC and HP are maturing and emerging. However, the technology is not without its challenges. It requires an up-front investment in hardware, software, training on implementation and managing the cloud.

Let’s take a close look at private cloud benefits, building blocks and open-source private cloud vendors to get more information.

Benefits of Using Private Clouds

Private cloud computing introduces a new way to use IT resources and brings automation, process changes, management changes, service standardization, culture and policies that consistently come up in polls as more difficult challenges than the technology itself. Private cloud architectures should be designed with the future use of hybrid cloud implementations in mind, providing more choices and a potential migration path to public cloud services as they mature. In the case of private clouds, private cloud resources will be used in place of dedicated servers, and thus, organizations will realize the following benefits:

- Security—security and governance capabilities specifically designed for an organization’s requirements.
- Regulatory and compliance—controls the service level of the platform, based on the organization’s needs and compliance requirements.
- Control—the organization owns, operates and controls the way services will be delivered to various departments and partners.

- Efficient utilization of resources and better resource provisioning systems—highly automated in terms of how it manages pools of resources, including everything from compute capability to storage, analytics, process management and middleware.

Given these benefits, why wouldn't everyone choose a private cloud?

Building Blocks of Private Clouds

Various cloud service providers offer the building blocks for a private cloud infrastructure virtualization (hypervisors, such as ESXi, Xen and so on), self-service, metering or chargeback, and automated workflow management. Most private cloud providers also provide

a management platform as well as security products and best practices to make the private cloud infrastructure robust and secure.

Real-World Application of a Private Cloud

Commonly, lab automation, dev-test, performance testing, sales demonstrations and so on are use cases that are best fits for a private cloud. Consistent and reproducible environments improve uniformity between development, test and production environments. A self-service option not only helps in productivity, but it also avoids many issues, such as variable workload demands. Scalability is more complicated to manage manually.

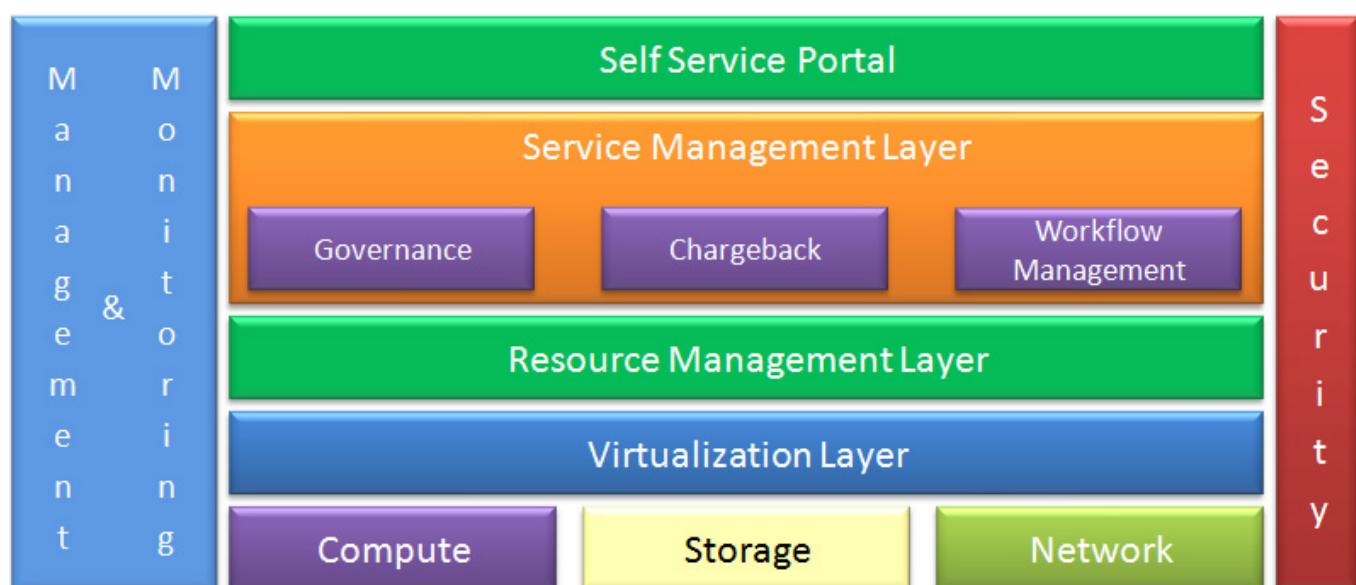


Figure 1. Building Blocks of a Private Cloud

Open-Source Private Cloud Product Vendors

OpenStack: OpenStack offers an open-source cloud computing platform to build different kind of clouds, such as public, private and hybrid clouds. Rackspace (the “Cloud Files” platform) and NASA (Nebula) started it in 2010. At present, HP, SUSE Linux, Red Hat, Cisco, Dell, IBM and others have joined the OpenStack project. It now is released under Apache license. OpenStack provides an Infrastructure-as-a-Service solution through a set of unified services, and each service offers an API that facilitates this integration.

OpenStack components:

- Compute provisioning and managing large networks of virtual machines.
- Block storage provides persistent block storage to guest virtual machines.
- Object storage stores and retrieves files.
- Networking enables network connectivity as a service and enables users to create and attach interfaces to networks.

- The dashboard enables users to interact with OpenStack services to launch an instance and to set access control.
- The identity service provides authentication and authorization for OpenStack services.
- The image service provides a registry of virtual machine images.
- The metering/monitoring service is used for billing, benchmarking and statistics purposes.
- The orchestration service orchestrates multiple composite cloud applications.

CloudStack: CloudStack is an open-source cloud computing platform to create, manage or deploy an Infrastructure-as-a-Service model in a service-provider environment or in an organizational environment. Cloud.com originally developed it in an initial phase. Cloud.com initially released most of CloudStack under the GNU General Public License, version 3. Citrix later released the remaining code under GPLv3. Citrix donated CloudStack to the Apache Software Foundation. CloudStack 4.2.0 was released on October 1, 2013, and it

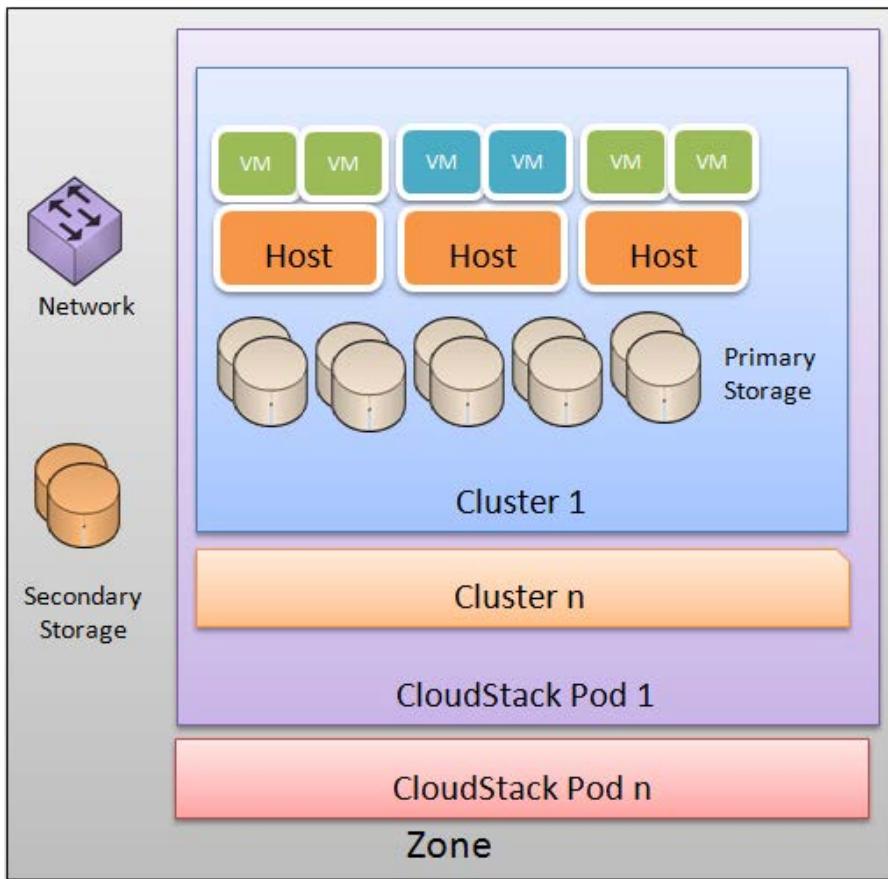


Figure 2. CloudStack

is the latest stable version available at the time of this writing. It supports the end-to-end functionality that is required for IaaS, which includes compute, storage, network, user management, user interface, RESTful API and command-line tools. It supports hypervisors, such as KVM, VMware vSphere, XenServer and more for virtualization. CloudStack implements the Amazon EC2 APIs, Amazon S3 APIs and vCloud API in addition to its own API. CloudStack installation includes two parts: 1) the management server manages the cloud

infrastructure and 2) the cloud infrastructure itself.

With CloudStack, a zone normally corresponds to a data center. There can be multiple zones in a data center. Pods are enclosed within zones. Each zone can control one or more pods. Zones can be private or public.

A CloudStack pod represents a single rack that consists of one or more clusters of hosts and one or more primary storage servers. Hosts in the same pod are available in the same

subnet. A host is a computer that provides the computing resources, such as CPU, storage, memory, networking and so on to run the virtual machines. CloudStack is hypervisor-agnostic. Multiple hypervisor-enabled servers, such as a Linux KVM-enabled server, a Citrix XenServer server or an ESXi server can be used.

Each host has a hypervisor to manage the VMs. A cluster is a collection of one or more hosts and one or more primary storage servers. It can be considered as a

set of XenServer servers or a set of KVM servers. Primary storage is coupled with a cluster that stores the disk volumes for all the VMs running on hosts in that specific cluster. Secondary storage is shared by all the pods in the zone that stores ISO images, templates and disk volume snapshots.

Eucalyptus: Eucalyptus is available in two different flavors: 1) open source and 2) a commercial private IaaS service provider. Eucalyptus is short for “Elastic Computing Architecture for Linking Your Programs to Useful Systems”. The cluster controller manages a

collection of node controllers. It has access to public and private networks. It controls execution of VMs and manages virtual networking too. Each Eucalyptus cloud will have multiple cluster controllers, and each Eucalyptus cloud will have one or more node controllers per cluster controller. It controls VM activities, such as launch instances, inspection, termination of instances and cleanup of instances. The cloud controller manages virtualized resources of the Eucalyptus-based private cloud environment. Each Eucalyptus cloud will have a single CLC.

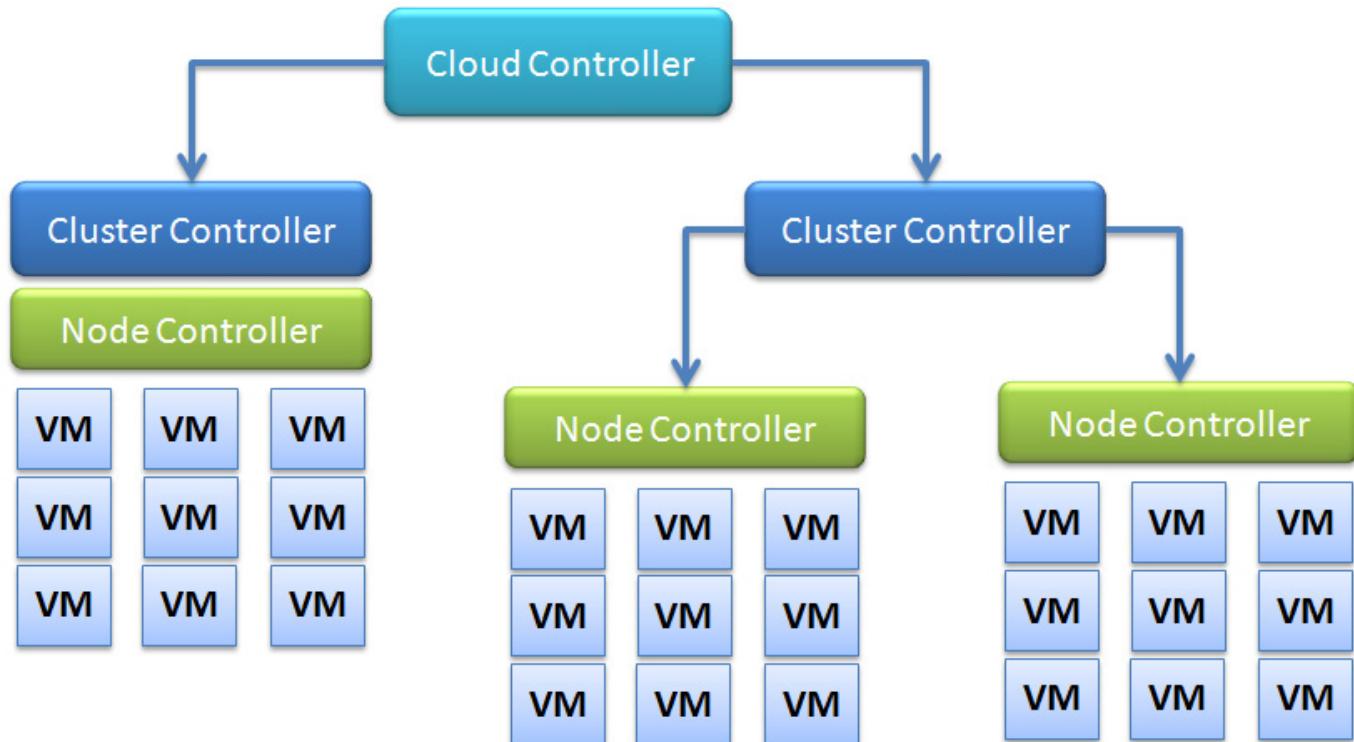


Figure 3. Eucalyptus

Eucalyptus features:

- Linux and Windows virtual machines (VMs) are supported.
- Security groups and elastic IPs are supported.
- Users and groups management, accounting reports.
- Configurable SLAs and

scheduling policies.

- Compatible with Amazon Web Services (AWS) API.
- Support for installation and deployment from source or DEB and RPM packages.

One of the leading sportswear companies is using Eucalyptus as the foundation for its cloud computing

Table 1. OpenStack vs. CloudStack vs. Eucalyptus

	OpenStack	CloudStack	Eucalyptus
Components	Nova (compute), Swift (object storage), Glance (image service), Keystone (identity management), Horizon (GUI interface)	Management Server; Hypervisor Nodes; Storage Nodes; Layers: Zone, Pod, Cluster, Host, Primary Storage, Secondary Storage	Cloud Controller (CLC): manages the virtualization resources and APIs, provides the Web interface; Walrus (S3 storage); Cluster Controller (CC): controls execution of VMs and their networking; Storage Controller (SC): provides block-level storage to VMs (EBS); Node Controller (NC): controls VMs via hypervisors
Codebase	Python	Java	Java, C
Hypervisor support	Xen, KVM, UML, LXC, VMware	Xen, KVM, VMware, Citrix XenServer	Xen, KVM, VMware
API	Yes	Yes	Yes
AWS API Compatibility	Yes	Yes	Yes
Snapshots	Yes	Yes	Yes
Self-Service User Portal	Yes	Yes	Yes

OpenNebula provides support for hybrid cloud computing with connectors for AWS.

initiative to support its elastic and dynamic requirements related to marketing campaigns. The Eucalyptus-based private cloud environment has helped the company deliver faster time to market, agility, highly available, scalable and elastic Web applications using a secure on-premises IT environment. With the Eucalyptus cloud, it is able to provide a fast, highly available e-commerce experience for its customers worldwide. It uses scaling up and down to manage variable demand effectively. One-click deployment of applications, quick provisioning and de-provisioning of compute and storage resources on secure, reliable and highly available on-premises infrastructure is extremely useful for the on-line campaign management.

OpenNebula: Ignacio M. Llorente and Ruben S. Montero initiated OpenNebula as a research project in 2005, and it's now available under the Apache license. It is written in C, C++, Java, Ruby, shell scripts, yacc and lex. It's an open-source project to build and manage virtualized enterprise data centers and IaaS clouds. It emphasizes core values, such as

openness, excellence, cooperation and innovation. OpenNebula manages a virtual infrastructure to build private, public and hybrid IaaS (Infrastructure-as-a-Service) clouds. It orchestrates storage, network, virtualization, monitoring and security.

OpenNebula provides support for hybrid cloud computing with connectors for AWS. It provides AWS EC2 and EBS APIs and a self-service portal for cloud consumers. A catalog of virtual appliances ready to run in OpenNebula environments is available in the Appliance Marketplace. It provides a powerful CLI that resembles typical UNIX command applications. OpenNebula adheres to a modular and extensible architecture, so it's easy for third-party tool integration. Best-effort community support and SLA-based commercial support directly from the developers is available for customers. OpenNebula helps enable agile, elastic and multi-tenant environments customized to offer infrastructure services.

Public Clouds vs. Private Clouds

The private cloud is recommended over the public cloud in the

Table 2. Public Clouds vs. Private Clouds

	Public Clouds	Private Clouds
Definition	A public or external cloud allows resources to be provisioned dynamically (partly on a self-service basis) over the Internet from an off-site third-party provider who shares resources and bills on usage or subscription basis.	An IT infrastructure that is under control of an organization's corporate firewall, delivered as a service to a restricted set of users usually within the organization, available as a flexible resource pool, provisioned on-demand, managed by the organization or a third party, existing on-premises or off-premises.
Control	The cloud consumer doesn't have complete control of the computing resources. A shared responsibility model exists between the cloud service provider and the cloud consumer. The responsibility may vary depending on the service model, such as IaaS, PaaS and SaaS.	The cloud consumer and cloud service provider are the same. The organization builds the private cloud internally behind the firewall to maintain complete control. The responsibility for resource management is with the organization only. (Here I am talking purely about private clouds, and not a hosted scenario.)
Resource Sharing	The infrastructure is shared between multiple organizations across geographic regions.	The infrastructure is dedicated to a single organization, and it is shared between different business units of an organization.
Scalability	Resources are seemingly infinite, hence it is very easy to scale based on application requirements while maintaining efficiency for not under-utilizing the resources or for not incurring too much cost.	Compute, storage and network resources are limited in capacity, because only the single organization needs to bear the cost of infrastructure. Most of the time, the focus of the organization and budgetary constraints play a significant role in the private cloud setup.
Control of Physical Infrastructure	The cloud consumer has no direct control of the resources, especially of the physical infrastructure where the IaaS model is provided.	The cloud consumer has a direct say on the physical resource requirements, and based on need, resources can be allocated, which is extremely flexible.
Compliance Requirements	It's difficult to meet compliance requirements, as a cloud consumer depends upon the cloud service provider for various external audits.	The organization's defined best practices can be applied in the environment, as resources are available on-premises and are under complete control.
Capital Expenditure	Huge capital expenditure is not required, as you are going to use resources based on a pay-per-use basis.	Huge capital expenditure is required to set up a private cloud environment by purchasing the physical components to install it and various software-licensing-related costs.
Usage of Cloud Features	It's advisable to use cloud features intelligently and on the basis of application requirements and with correct configurations by keeping best practices in line. Limited understanding and too many cloud features incur huge cost over time, which may not be required considering the application's characteristics.	The main benefit of the private cloud environment is the way the organization utilizes existing infrastructure and expertise to build an agile environment.

following scenarios:

- Stringent security management and regulatory compliance requirements demand tight security controls and full visibility into how the IT infrastructure is managed. It prevents moving the applications and data to an external cloud.
- High-performance requirements need the cloud to be set up and accessed over a private high-speed network.
- Need for full flexibility and control over how to operate the cloud to enable business innovation and future growth based on the cloud.
- Consolidate and standardize hardware and software resources of an organization; create a

shared, automated service platform for business users, and account them for usage.

Conclusion

The private cloud contrasts with the public cloud computing environment, where access to the resources is open to any customer who is willing to pay. There are number of variations that fit somewhere between those two, such as community cloud computing, virtual private cloud computing and hybrid cloud computing. The private cloud offers organizations a way to take the benefits of increasing simplicity of resource management with automation, flexibility and cost benefits, and provides an environment to be more competitive in the marketplace through greater control, choice, efficiency, quality of service and, most important, business agility. It is particularly well

Resources

Peter Mell and Timothy Grance, “The NIST Definition of Cloud Computing”:
<http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>

Phil Wainewright, “Private cloud discredited, part 2”:
<http://www.zdnet.com/blog/saas/private-cloud-discredited-part-2/1289>

Gartner, “Private Cloud Computing: An Essential Overview”:
<https://www.gartner.com/doc/1476032/private-cloud-computing-essential-overview>

PUMA.com Migrates to a Eucalyptus On-premises Cloud:
<https://www.eucalyptus.com/sites/all/files/cs-puma.en.pdf>

suites for organizations that are in greatly regulated sectors, such as health care, finance or banking or those that have made huge capital investments in existing data-center infrastructure, including virtualization and server consolidation.

Enterprise interest is already high, with roughly 75% of respondents in Gartner surveys saying that they plan to follow a private cloud computing strategy by 2014. This trend will move quickly on the Hype Cycle in the next couple years, as the private cloud moves from a strategy to pilot and

production deployments. According to Gartner research, by 2015, the majority of private cloud computing services will evolve to leverage public cloud services in a hybrid model. ■

Mitesh Soni works as a Technical Lead. He loves to write on technical and social subjects and enjoys photography. His technical blog is at <http://clean-clouds.com>, and his photography blog is at <http://myvividvisions.com>.

|||||
Send comments or feedback via
[**http://www.linuxjournal.com/contact**](http://www.linuxjournal.com/contact)
or to [**ljeditor@linuxjournal.com**](mailto:ljeditor@linuxjournal.com).

If You Use Linux, You Should Be Reading **LINUX JOURNAL**™



Subscribe now for instant access!
For only \$29.50 per year—less than \$2.50 per issue—you'll have access to *Linux Journal* each month as a PDF, in ePub format, in Mobi format, on-line and through our Android and iOS apps. Wherever you go, *Linux Journal* goes with you.

SUBSCRIBE NOW AT:
[**WWW.LINUXJOURNAL.COM/SUBSCRIBE**](http://WWW.LINUXJOURNAL.COM/SUBSCRIBE)



DOC SEARLS

A Pain in the Person

At what point will we say “enough”?

To illustrate what a negative externality (<http://economics.fundamentalfinance.com/negative-externality.php>) is, it helps to get literal about it. So let's start with horse shit—specifically, the Great Horse Manure Crisis of 1894. That was the year a *Times of London* reporter guessed that the city's streets would be buried under nine feet of manure within 50 years. In the Pulitzer-winning *Gotham: A History of New York City to 1898* (Oxford: 1998, http://www.amazon.com/dp/0195116348/ref=rdr_ext_tmb), Edwin G. Burrows and Mike Wallace say horses deposited 2.5 million pounds of manure and 60,000 gallons of urine on 250 miles of city streets, every day. By 1900, possibly the peak year for horse-drawn transport, New York was served by 100,000 horses producing 1,200 metric tons of manure, daily. About half that much was collected daily and hauled,

by horse, to Barren Island (http://en.wikipedia.org/wiki/Barren_Island,_Brooklyn), off the coast of Brooklyn. The rest accumulated, bred flies, smelled to heaven and spread disease—all negative externalities. Add to those the costs of breeding, raising and feeding horses with hay grown on farms in nearby countryside and hauled—also by horse—to stables in cities.

But the positive internalities of the horse-drawn system outweighed the negative internal and external ones. Civilized city life required horse-drawn transport, so citizens put up with the bad stuff, as they always do. And, there were positive externalities as well. Cruelty to horses drove Henry Burgh to found the ASPCA in 1866 (<http://www.aspca.org/about-us/about-the-aspca/history-aspca>). Horse poop and landfill expanded Barren Island until it eventually became part of Brooklyn. In 1930, it was paved with

tarmac and runways to become New York's first major airport: Floyd Bennett Field (https://en.wikipedia.org/wiki/Floyd_Bennett_Field). Today, it's a park by the same name (<http://www.nyharborparks.org/visit/flbe.html>).

But, the most positive outcome of horse-drawn transport was that it legitimized gas-powered mechanical transportation: cars and trucks. These too would produce negative externalities along with many positive ones. But let's not go there, because we know them all well. Instead, let's look at negative externalities we put up with today in the digital realm, starting with advertising.

Traditional advertising—the kind that runs on TV, radio and in print—wastes no more time, space and electricity than it takes to generate it. Although it does waste a substantial sum of all three, those all have physical limitations. The same is not true of advertising on-line, where virtual space is virtually infinite, and pollution by wasted messages is entirely ignored by those who create it. All they care about are "exposures" and "click-throughs". If an ad doesn't get read, it doesn't matter to the producers, because the costs of the waste are mostly external: borne by others. Last December, Fred Wilson gave a

speech (<http://avc.com/2013/12/my-talk-at-leweb-yesterday>) in which he fingered "data leakage" and "pollution" as a Major Issue (starting around 23 minutes in). This could be a turning point (Fred's an influential guy). Or, it could just be sign of the times that we'll ignore for another few years or decades.

We also suffer many negative externalities from the login/password convention, which is as stale today as horse-drawn carriages were in 1910. If you're Example.com, all you care about are the logins and passwords you require of your users—not the dozens or hundreds of logins and passwords the user has to remember, somehow. Right now, I'm in the early stages of changing many hundreds of logins and passwords on up to four different browsers, on several different computers, plus those on my phones and tablets. This is a huge project, slowed by a de-motivating sense of futility, plus a resentment about not coming up with something better. True, a variety of password managers are available to me, and I'm busy kicking their tires as well, but each of those brings its own set of vulnerabilities, chief among which is dependency itself: I become their vassal too.

I see the larger problem as

centralization: a box so huge that we can hardly think outside of it, much less develop solutions out there. When Target Stores got hacked, and more than 110 million credit cards needed to be replaced (http://www.mercurynews.com/business/ci_24889060/target-now-says-up-110-million-customers-victimized), almost nobody (far as I know) looked at the sum costs of the security breach to the individual credit-card holders, much less at the need to come up with alternative approaches that

would present bad guys with smaller surfaces to attack. Instead, all we got was hand-wringing and promises by feudal lords and their suppliers to build better castles, most of which consist of silo'd "loyalty" programs and other coercive systems for keeping their serfs—customers and users—trapped inside. Even every app, it seems, is a little castle of its own, and there are now more than a billion of those, both for iOS and Android.

The Internet was designed to solve this problem, starting in

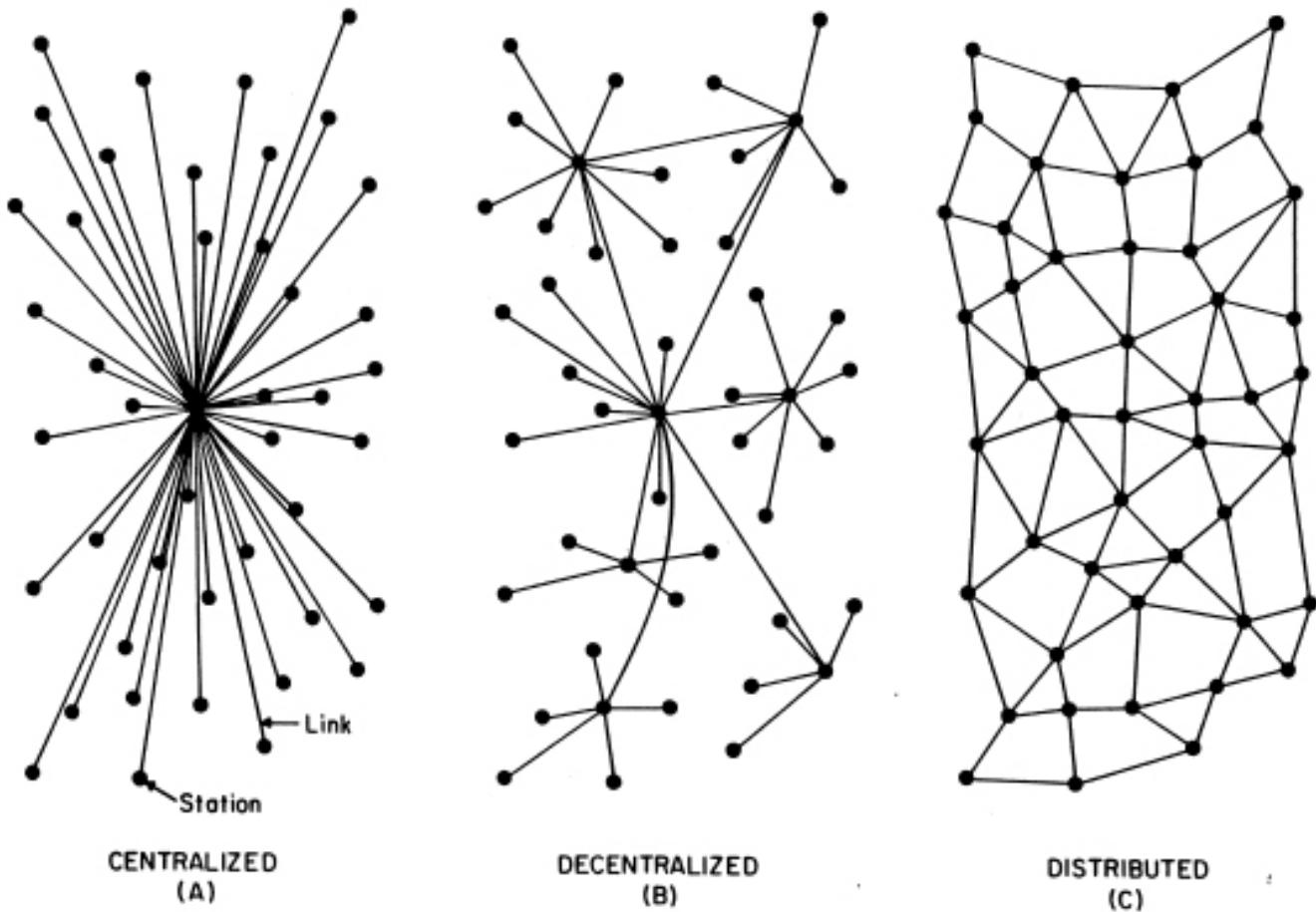


Figure 1. Centralized, Decentralized and Distributed Networks

1962. That was when Paul Baran (http://en.wikipedia.org/wiki/Paul_Baran) came up with a network model designed to avoid the vulnerabilities inherent in the only kind of networks anybody knew at the time (<http://www.rand.org/about/history/baran.html>). Those were centralized ones, such as we got from phone companies and TV networks. Baran's new model was what he called *distributed* (<http://www.rand.org/about/history/baran-list.html>), and he illustrated it with the graphic shown in Figure 1 (<http://www.cybertelecom.org/notes/baran.htm>).

Every node on a distributed network would be independent. And, although a node or a link might be vulnerable, it would not bring down the whole network if it failed. Such a network would be a *heterarchical*, a virtue I unpacked here in April 2014 (<http://www.linuxjournal.com/content/opening-minds-spheres-among-us>). The Internet we have today is actually both decentralized and distributed, but at least it gives us a platform for creating distributed solutions to the problems of centralization. Linus and thousands of collaborators have used that platform to create and continuously improve Linux for 24 years, all operating independently. Yet,

Advertiser Index

Thank you as always for supporting our advertisers by buying their products!

ADVERTISER	URL	PAGE #
Big Data TechCon	http://www.BigDataTechCon.com	73
Drupalize.me	http://www.drupalize.me	81
EmperorLinux	http://www.emperorlinux.com	25
Fossetcon 2014	http://fossetcon.org	2
Linux Journal DevOps	http://devops.linuxjournal.com	7
SharePoint TechCon	http://www.sptechcon.com/	23
Silicon Mechanics	http://www.siliconmechanics.com	3

ATTENTION ADVERTISERS

The *Linux Journal* brand's following has grown to a monthly readership nearly one million strong. Encompassing the magazine, Web site, newsletters and much more, *Linux Journal* offers the ideal content environment to help you reach your marketing objectives. For more information, please visit <http://www.linuxjournal.com/advertising>.

we also find Linux inside nearly every big centralized system on the Net—Twitter, for example (<http://toolbar.netcraft.com/netblock?q=twitter-network,199.16.156.0,199.16.159.255>). Because Twitter is centralized, it's easy for a government to shut it down. That's what happened in Turkey (<http://www.reuters.com/article/2014/03/21/us-turkey-twitter-idUSBREA2K0CB20140321>), back in March. But, that's beside the main point I want to make here, which is that distributed networks are composed of the same individuals that centralized systems burden with negative externalities, which those systems either rationalize or ignore.

In May 2014, the top European court ruled (<http://www.theguardian.com/technology/2014/may/13/right-to-be-forgotten-eu-court-google-search-results>) in favor of an individual's "right to be forgotten", and against Google, which produces an infinitude of search results, including many that some people would rather have the world forget. While the implications of the ruling were hotly debated, lost in the midst was the fact that Google's manners toward individuals often have been terrible. One example is StreetView (https://en.wikipedia.org/wiki/Google_Street_View), which

provides the world with pictures of everybody's homes and businesses. This freaked out a lot of people, and in some cases, freaked out whole countries. Another is Google Glass (https://en.wikipedia.org/wiki/Google_Glass#Criticism_and_privacy_concerns), which has many fine uses, but also suggests that the wearer is recording others without permission. Google's lame manners in those cases are made possible by centralized systems that ignore or rationalize negative consequences to individuals.

At some point, outsourcing negative externalities to individuals is going to become a burden too high for those individuals to bear, and a tipping point will be reached. When that happens, we'll start to see some forward motion toward creating the distributed individual-first solutions that Paul Baran first drew for us more than a half century ago. ■

Doc Searls is Senior Editor of *Linux Journal*. He is also a fellow with the Berkman Center for Internet and Society at Harvard University and the Center for Information Technology and Society at UC Santa Barbara.

||||||||||||||||||||||
Send comments or feedback via
<http://www.linuxjournal.com/contact>
or to ljeditor@linuxjournal.com.