

RIK FARROW

musings



rik@usenix.org

IN MY LAST COLUMN, I ARGUED THAT we need to be willing to change. Change is inevitable. Just as the seasons change and the climate warms, our computing environment also changes, slowly enough that we sometimes miss noticing the changes until we run smack up against them.

Let's consider one change in the world of computing, and see how it relates to a topic that is only tangentially related—system administration.

One of the hot computing news stories eight years ago (1999) was the race to produce a desktop PC for less than \$1000. These computers came with 366MHz Celeron CPUs and no monitor [1]. Fast-forward less than ten years, and you can buy a system with a dual-core, 64 bit, 2-GHz CPU with a 19-inch LCD monitor and one hundred times the storage capacity for under \$700. I'd like to ignore discussions of Moore's Law and cut to the chase. Today's base system has capabilities generally undreamed of just ten years ago. Running multiple VMs on desktop systems is not only feasible but not uncommon.

Fertility

System administrators have more than high-performance desktops in their basket full of changes. I mentioned VMs, which seem to multiply like rabbits in the springtime. VMs encapsulate whole operating systems as well as at least one application, yet they still must be managed and configured, and the data—often just the data—must be properly backed up. So one system today may represent not just a faster system but many multiples of past systems.

What is true of desktops is true of servers as well. Servers have become just as fecund, if not more so, than desktops, expanding to fill ever-shrinking machine rooms, even as they release ever greater amounts of “waste” heat. It is not the energy by-product of servers that concerns me here, but their ever-increasing capacity for fruitful labor. Servers too become targets for VMs, so that every last erg of energy will be useful, and not simply add to the entropy of the universe and deplete our fast-declining store of fossil fuels for no good purpose.

Once upon a time, system administration meant editing configuration files, using your text editor of choice. Today, the hand-editing of files has gone

the way of the oxcart, with the rare exception of systems that are just too different to be managed en masse. As system administrators, we are faced with a choice: to become equally obsolete, or to change with the times and become meta-administrators instead.

Going Meta

One of the joys of being an editor is the ability to spout silly ideas, even attempt to coin words that may help us understand the world we live in. Meta-administration means system administration that has moved beyond the editing of configuration files. The knobs in those files remain there, just as adjustable as ever. What has to change is how we manage to tweak those knobs. It has always been difficult to know how to manage multiple versions of UNIX, where the configuration files may not be the same (think AIX or Mac OS X) and small changes may have large unintended consequences. Now we have multiplied the number of systems we must manage, even as the speed of those servers and desktops has grown exponentially. To expect that we can hand-edit configurations, or even launch a script that will zoom across the network making the same changes everywhere, is no longer plausible.

Meta-administration means that we will be providing guidelines for how the system of systems should behave. It does not mean that we will never again edit a configuration file. Hardly that. The deep knowledge will be more important than ever, because it will be skilled practitioners who will still be needed to solve the intractable problems that arise. But much of the day-to-day management of systems will be left to agents, guided through configuration management techniques that we are still evolving today.

The Lineup

This issue of *;*login**: has been dedicated to the inevitability of change in system administration. When I first approached the concept of system administration, I had little notion of what was involved. Happily, no one else did either at the time (1983), so I was not alone in my ignorance. I thought that all I needed to do to write a book to guide future system administrators was to go to UC Berkeley and interview real, live working sysadmins to understand what it was that they did. I did make that trip in 1985 to Evans Hall, but I discovered that the sysadmins working there had no idea what they were doing. I don't mean to say they were clueless, because any one of them could hand-edit configuration files with the best of us. But those UCB sysadmins had no knowledge of the bigger picture, that is, what it meant to be a system administrator. They merely carried out assigned tasks, such as adding user accounts, managing printer queues, and handling backups.

If you have managed to read this far, you have, hopefully, come to the same conclusion I have: those days are over. Although you may know a lot about what goes on "under the hood," the primary task of system administration has become meta-administration.

In that vein, we begin this issue with an article by Glenn Fink and Deb Frincke about autonomous systems. Glenn and Deb start off with the notion that we will be using autonomous agents to manage not just our systems but aspects of our networks as well, and they examine the consequences of utilizing agents. For example, what does it mean when an agent "decides" to purchase increased bandwidth from your ISP to handle an

expected increase in network traffic to maintain a service-level agreement? The agent made that buying decision, but you will be paying for it. Ultimately, who is responsible for the actions of autonomous agents?

Chad Verbowski follows up with an article about Flight Data Recorder (FDR), a new feature in Vista, which someday will be found in the next server version of Windows as well. Chad had delivered a paper about FDR at OSDI '06, but I wanted more details, and Chad has them here. Essentially, FDR provides the administrators of Windows systems with fine-grained details about significant changes to those systems, using a clever method for compressing the huge amount of log data before it ever leaves the Windows box, yet leaving that compressed data in a form that can be rapidly searched for nuggets of key information.

In the next article, Brendan Quinn shares his experiences in working with LDAP. LDAP means different things to different vendors, and Brendan provides useful hints on surviving conflicting expectations of those vendors and still having a useful directory service.

John Lloyd next attempts a difficult feat by explaining system engineering in the short space permitted (and goes a bit over the limits I generally enforce in terms of page count). I found John's information interesting and useful, both in the real terms of designing systems that will work the way you expect them to and in coming to true agreements about what management/powers-that-be expect those systems to do. If you have ever gotten into trouble when building systems for some specific task, you owe it to yourself to read this article.

Although Mark Burgess had completed his series about configuration management, I felt he had left some aspects uncovered. After a short pursuit, Mark agreed to be interviewed about his concept of "promises" and how they fit into the world of configuration management. Promises fit in very well with my own concept of meta-administration, and they had a strong hand in my creation of this column. Mark also relates Alva Couch's closures as well as Paul Anderson's aspects to promises, so you can at least begin to understand where some significant players in the configuration management community are heading.

Dan Appelman follows up with significant advice for any system administrator involved in the management of mail servers. Gee, did I manage to leave any sysadmin out there? Based on his tutorial at LISA, Dan describes some of the impact of U.S. law, as well as actual case histories on the uses and abuses of email. Dan warns us that anyone who sends out email may run afoul of anti-spam legislation and can suffer the legal and financial consequences of doing so.

Thomas Sluyter shares helpful tips for sysadmin consultants. Thomas explains how to produce regular reports as you carry out contracts, so that management is constantly aware of both your progress and any significant hurdles in the way of progress. Even if you are not contracting sysadmins, I highly recommend his advice to you. Keeping a log of the work you do, along with the tasks you have been assigned to do, will prove helpful when your next job review comes around.

David Blank-Edelman cheerfully describes climbing down trees in Perl. The trees are file systems, and the task is a common one for any system administrator. Robert Haskins writes about ways of providing VoIP at the provider level, and Heison Chak tells you how to reflash a cheap file server to turn it into an Asterisk server—PBX on the cheap. Robert Ferrell then provides advice on the care and feeding of sysadmins.

Elizabeth Zwicky follows up with her usual book reviews, that is, reviews that are honest and funny at the same time.

Nick Stoughton takes a stand against the abuse of standards and standards bodies. His target, OOXML, gets revealed not as a standard but as a statement of monopoly control with the sole purpose of providing a large vendor with a continued revenue stream. I personally would be more upset if abuses of power had not become so commonplace these days. But Nick makes a compelling case for the sometimes capricious nature of standards bodies.

The USENIX Notes section fills you in on the latest news about SAGE: The USENIX SIG for Sysadmins, and about the upcoming LISA '07 conference, and asks for your advice and assistance.

This issue concludes with summaries from LISA '06, including two workshop summaries (Configuration Management and Advanced Topics). If you attended a workshop and don't see your summary here, it's not because we aren't interested in reading it—it's just that no one provided us with one to include.

As the world turns, day becomes night, winter warms into spring, and fall cools into winter. The world of computing races forward as well, with ever-increasing complexity. Old-style system administration provides one with a certain comfort, as you can see the changes you make, tests their effects, and easily adjust your changes if necessary. In the new world, that level of comfort is fast disappearing, and the need for abstracting changes to support flexible control of more systems than ever has become a requirement. Don't get left behind.

REFERENCE

[1] List of desktop systems, capacities, and prices from fall 1999:
<http://www.cs.umd.edu/hcil/academics/courses/fall1999/cmssc838s/Apps/jintong/all.csv>.