NOTES

USENIX Board of Directors

Communicate directly with the USENIX Board of Directors by writing to board@usenix.org.

PRESIDENT
Amy Rich, *Redox*
arr@usenix.org

VICE PRESIDENT
Arvind Krishnamurthy, *University of Washington*
arvind@usenix.org

SECRETARY
Kurt Andersen, *LinkedIn*
kurta@usenix.org

TREASURER
Kurt Opsahl, *Electronic Frontier Foundation*
kurt@usenix.org

DIRECTORS
Cat Allman, *Google*
cat@usenix.org

William Enck, *North Carolina State University*
will@usenix.org

Laura Nolan, *Slack Technologies*
laura@usenix.org

Hakim Weatherspoon, *Cornell University*
hakim@usenix.org

EXECUTIVE DIRECTOR
Casey Henderson
casey@usenix.org

;login: Enters a New Phase of Its Evolution

Cat Allman, Rik Farrow, Casey Henderson
Arvind Krishnamurthy, and Laura Nolan

For over 20 years, *;login:* has been a print magazine with a digital version; in the two decades previous, it was USENIX's newsletter, *UNIX News*. Since its inception 45 years ago, it has served as a medium through which the USENIX community learns about useful tools, research, and events from one another. Beginning in 2021, *;login:* will no longer be the formally published print magazine as we've known it most recently, but rather reimagined as a digital publication with increased opportunities for interactivity among authors and readers.

Since USENIX became an open access publisher of papers in 2008, *;login:* has remained our only content behind a membership paywall. In keeping with our commitment to open access, all *;login:* content will be open to everyone when we make this change. However, only USENIX members at the sustainer level or higher, as well as student members, will have exclusive access to the interactivity options. Rik Farrow, the current editor of the magazine, will continue to provide leadership for the overall content offered in *;login:*, which will be released via our website on a regular basis throughout the year.

As we plan to launch this new format, we are forming an editorial committee of volunteers from throughout the USENIX community to curate content. This new model will increase opportunities for the community to contribute to *;login:* and engage with its content. In addition to written articles, we are open to other ideas of what you might want to experience. We welcome your comments and suggestions: login-comm@usenix.org.

Interview with Clem Cole

Rik Farrow

Clem Cole is an old school hacker and "Open Sourcerer" with more than 45 years of free and open source system development experience. Clem has held practically every position in the computer field from operator, programmer, and designer to VP of Engineering, CTO, and startup founder. He first encountered the early editions of UNIX in the 1970s while at Carnegie Mellon University, later doing his graduate work at the University of California, Berkeley. He has been designing and developing operating systems and technical computing systems ever since, currently leading an international team of engineers. He helped to write one of the original TCP/IP implementations in the late 1970s, and is known as one the authors of the precursor to IM, the UNIX talk program, as well as other more humorous and notorious hacks. He is honored to be a past President of the USENIX Association and the 2016 winner of the Linus Pauling Prize for Science. clem@ccc.com

I first met Clem Cole at a USENIX conference, probably in the 90s, but I had encountered him via a paper he helped with in 1985. Ted Kowalski had written `fsck`, bringing together ideas from three previously existing UNIX programs, `ncheck`, `icheck`, and `dcheck`, and experience using earlier IBM programs, Scavenger and Vulture [1], for recovering after disk crashes.

I had also heard that Clem had a long history with USENIX, and decided to interview him for this, the final print issue. I had learned by reading early issues of *UNIX Notes* and *;login:* that USENIX conferences were how UNIX users exchanged information in the early days, and it occurred to me that Clem was a participant I could ask about this.

Rik Farrow: When did you begin working with UNIX? I encountered UNIX in the early 80s, while working for companies in the Bay Area.

Clem Cole: By the 80s UNIX was even cooler than when I first encountered it. I started using UNIX with Fifth Edition version in 1976 while at Carnegie Mellon (CMU).

Truth is, my first experience with UNIX and C in particular, coming from the IBM 360 and DEC PDP-10, found me skeptical. But I had been schooled in the CMU gospel of using systems program languages (BLISS in this case) so I had already started to transition from 360 assembler.

What made UNIX/C really cool was that as much as I liked the stuff we had on the PDP-10s (like the XGP—the predecessor to the laser printer) BLISS on the PDP-11 required cross compiling. C was self-contained. The documentation for C was almost non-existent, with the exception of Dennis Ritchie's paper in V5 and V6 in the `c` directory in `/usr/doc`. The code from the compiler was not great compared to BLISS, but it was "good enough."

And we had our own PDP-11 in the CMU Electrical Engineering Digital Lab and we did not have to share it with many other folks.

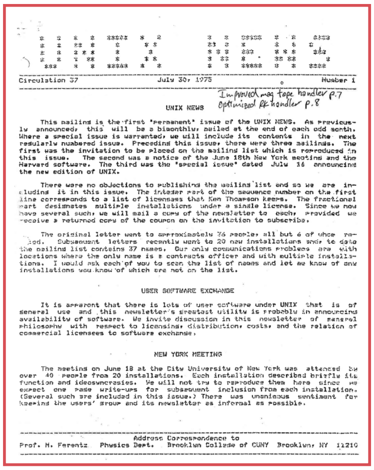
Ted had a xeroxographic copy of the Lions book [2] and I made my own copy. Then Ted came back with proofs for Kernighan and Ritchie [3] in a binder and I read those two documents that spring and things about UNIX started to click. Pretty soon I started to see that I could get most anything I had been able to do on the PDP-10s and the 360 on the PDP-11 and I only shared it with a few other people. That was way cool.

And then one day we had a disk crash on a machine in CMU's BioMed Department. I got a call from the guys that ran it, and they wanted to try to use the EE system to try to fix the disk. Ted and I both had used a disk reconstruction program on the IBM and Ted had been an IBM MTS hacker at the University of Michigan before he came to CMU. I remember spending a number of hours with `ncheck/icheck/dcheck` and grumbling to Ted as we were working with their disk.

It turns that out Ted had started a new program but it was not complete. Now he had a mission. By the way, the original name of the `fsck` program used a different second letter.

The other thing I saw around then was a copy of some of the original issues of *UNIX News* that Columbia University was printing up. I don't remember who had them, but I think it was someone else with a connection to Harvard or maybe Columbia. I got on the mailing list somehow and started eating it up.

As an undergrad I could not travel, but when I first started to work for Tektronix in 1979, I went to my first USENIX conference (I want to say Toronto, but I could be wrong). An early winter one was Boulder where USENIX had rented a movie theater, the same theater that was featuring the new movie, *Black Hole*. What I do remember the most of that conference is that's where Tom Truscott regaled us on his homemade autodialer they built so they could run UUCP.



UNIX NEWS

Circulation 37 July 30, 1975 Number 1

Improved mag tape handler p.7
Optimized RK handler p.8

UNIX NEWS

This mailing is the first "permanent" issue of the UNIX NEWS. As previously announced, this will be a bimonthly, mailed at the end of each odd month. Where a special issue is warranted, we will include its contents in the next regularly numbered issue. Preceding this issue, there were three mailings. The first was the invitation to be placed on the mailing list which is reproduced in this issue. The second was a notice of the June 18th New York meeting and the Harvard software. The third was the "special issue" dated July 16 announcing the new edition of UNIX.

There were no objections to publishing the mailing list and so we are including it in this issue. The integer part of the sequence number on the first line corresponds to a list of licensees that Ken Thompson keeps. The fractional part designates multiple installations under a single license. Since we now have several such, we will mail a copy of the newsletter to each, provided we receive a returned copy of the coupon on the invitation to subscribe.

The original letter went to approximately 76 people, all but 6 of whom replied. Subsequent letters recently went to 20 new installations and, to date the mailing list contains 37 names. Our only communications problems are with locations where the only name is a contracts officer and with multiple installations. I would ask each of you to scan the list of names and let me know of any installations you know of which are not on the list.

USER SOFTWARE EXCHANGE

It is apparent that there is lots of user software under UNIX that is of general use and this newsletter's greatest utility is probably in announcing availability of software. We invite discussion in this newsletter of general philosophy with respect to licensing, distribution, costs, and the relation of commercial licensees to software exchange.

NEW YORK MEETING

The meeting on June 18 at the City University of New York was attended by over 40 people from 20 installations. Each installation described briefly its function and ideosyncrasies. We will not try to reproduce them here since we expect one page write-ups for subsequent inclusion from each installation. (Several such are included in this issue.) There was unanimous sentiment for keeping the users' group and its newsletter as informal as possible.

Address Correspondence to
Prof. M. Ferentz Physics Dept. Brooklyn College of CUNY Brooklyn, NY 11210

Vol. 1, No. 1 of *UNIX News*, July 30, 1975. The circulation was 37.

Originally, we came to those meetings representing our orgs—universities or commerical entities. You were supposed to have the signature page of your AT&T UNIX license to join. I don't remember when the first personal memberships were offered, but I was the seventh person to join USENIX.

So back to your question. In those days DEC did not support UNIX, so we had a "we all are in this together" attitude. Everything was "open source" because we all had licenses. I think the thing that is lost today is that it was the cost of the hardware that was the limit to being part of the "UNIX club," not the cost of the UNIX software sources.

RF: That's interesting, since AT&T raising the license fee for the source for System V Release 4 (SVR4) was the main reason for the UNIX wars [4] that began in the late 80s.

CC: Actually, that's not quite true. The UNIX wars had started long before then. The 1988 SVR4 release and the raising of the license redistribution fee in particular was the source of the "fair and stable license terms" of Open Software Foundation (OSF) verses UNIX International (UI). You have to understand that each time AT&T had released a commercial redistribution license (starting with V7) the fees had gone up. The vendors had been having a knock-down, drag-out war for 5–10 years by 1988. UI vs. OSF was just the final battle.

The problem for the vendors was they treated UNIX like they owned their OSs and made them private with lots of local hacks to create vendor lock-in for their customers. The UNIX wars were really based on who got to decide what the definition of UNIX was going to be.

AT&T thought they got to say it because they owned the intellectual property. But the Berkeley Software Distribution (BSD) version had the greatest mind share as it included TCP/IP support. DEC, Apollo, HP, Masscomp, Sun, IBM and others had their customers running some version of UNIX on their hardware. And independent software vendors were annoyed because life had not gotten better—hence the 1985 /usr/group UNIX Standard that would later beget the IEEE POSIX work.

RF: How large were those early USENIX conferences? Dozens of people, hundreds of people? I've heard that by the late 80s there could be thousands of people attending.

CC: When they were at Harvard, Columbia, etc.—that is, the time of *UNIX News*—a conference fit into a classroom. By the time of Boulder it was probably about 100–150—about half a movie theatre full.

By the late 80s (when the final phase of the UNIX wars started) the San Francisco conference and I think the 10th USENIX conference in Portland were over a thousand. I think the peak was probably two or three thousand.

After Portland (summer 1985), USENIX started to fork into smaller dedicated conferences targeting subtopics and trying to keep it to be about 150–200 per conference.

RF: So what was it like at Boulder? Most of us know what modern USENIX conferences are like, with most having a focus on paper delivery, and some on talks.

CC: Remember there were no papers or proceedings in those days. Just talks. No PowerPoint either. Just overhead slides. And people came with prepared talks and signed up to give them. Nothing was preplanned.

It was also the first time I met Bill Joy. He had already started to build a cult around him. He talked about UCB Pascal and was very interesting. I also met Dennis Ritchie and Steve Bourne for the first time and was awed at how down to earth they were. They asked me questions and wanted my opinion. That was so cool. It was really a collegial setting. We were all sharing our experiences.

I think that's also where I met Bruce Borden for the first time. He had written the first parts of `mh` and he gave a talk about it. I wanted to try it immediately after I got back. As I said, Truscott talked about his fake autodialer and I remember being so impressed with it.

Boulder was sort of the start of Usenet. Truscott described the three schools connected using UUCP in North Carolina in his talk. Brian E Redman—the "ber" of "Honey-Dan-Ber" UUCP—said he would like to be able to call him from the systems in BTL in Whippany, NJ, named after the Marx brothers. Dennis Ritchie offered up the Bell Labs `research` system in Murray Hill.

At Tektronix, we were working with the University of California, Berkeley (UCB) on things like Spice, so I had talked my boss into letting me buy an autodialer to call `ucbvax`. So when I got back we joined up, and UCB was talking to `research`. Within a year Armando Stettner added `decvax` and `ihnp4` was added during the same time frame.

The fact is Bell Telephone Labs (BTL) had a large internal UUCP-based network before Usenet (which was smaller), but once `research` and `ihnp4` joined that changed the dynamic. Also, BTL was not on the ARPANET which a couple of the Usenet sites sort of were. Funny thing, forwarding from the ARPANET was discouraged originally. But at some point, the DARPA folks realized "Metcalf's Law" [5], that joining two networks make each a lot more valuable.

RF: Gaining membership in the early Internet/ARPANET was extremely weird from my perspective.

CC: When the ARPANet and the early Internet were set up, the US government paid for everything. To join, you had to be a DoD contractor and had to be sponsored. I read that in 1975 dollars, the cost per host was $250K a year and that also explains why some of the choices were made. A site on the network, like MIT and CMU, were not going to put their connection at risk.

Part of the problem was that many universities wanted to be part of the ARPANET but could not be as they were not doing ARPA work. Sometime early in the Reagan administration, the US government wanted to get out of funding the networking experiment they had started. Originally, CS-NET was set up by the NSF and contracted to BBN, so any research group could get a connection, but you had to pay for it and your leased telco connections. In fact, to help keep costs down, CS-NET set up a UUCP-like system, called "Phone-Net."

Within a a year or so, CS-NET was allowed to connect commercial sites too. That's how Masscomp got its connection to BBN and became part of the emerging Internet for $50K a year. Rick Adams had forked out UUNET and they too joined CS-NET and became the "offical" (recognized) UUCP/ARPANET bridge.

Finally, we started to get regional networks competing with CS-NET because the telco costs could be kept in check better and the era of the ISP appeared. The other change was that telco costs had to drop. That $50K/year Masscomp connection was a 56Kbit/second leased line from Westford, Massachusetts to Cambridge—a distance of about 30 miles. A T1 (1.44Mbit/sec) connection was closer to $50K/month.

So for a long time, the bulk of the email and netnews traffic was UUCP over dial-up.

The good news is that the telcos did start to figure out how to build cheaper digital circuits. And, once "cheap enough" connections arose, the need for UUCP and dial-up started to fade.

RF: I remember using `ihnp4` as a mail forwarder, but not many people today are going to recognize that hop. Was `ihnp4` in the Chicago area?

CC: Indian Hill New Products System 4, or `ihnp4` was in suburban Chicago. The three big national sites for Usenet were `decvax`, `ihnp4` and `ucbvax`. There was a study done by someone at BTL that concluded that for every call `ihnp4` underwrote, it generated between 10 and 20 downstream calls and that was good for AT&T so they continued to underwrite it. At its peak, `decvax` had a half to three-quarters of a million dollar phone bill. That was the trigger for USENIX to start to look for an alternative. Rick Adams, working for the USGS and running the site named `seismo`, proposed the

UUNET site, which USENIX helped fund [6]. UUNET become a commercial entity the following year.

RF: UUCP mail forwarding over IP is a big topic. I interviewed Mary Ann Horton [7], who had worked for BTL as well as doing a lot with Usenet. She explained things like the maps people were distributing at USENIX conferences and a separate tool to help people forward UUCP mail. Peter Salus' article [6] also explains her role in the maps project, and more details about the founding of UUNET.

CC: The printed maps were given away at conferences, but the tool was used to try to shorten paths for email and net news traffic. Remember, until IP where you have flat address space, UUCP was purely store and forward and at the complete message level. IP is store and forward at the packet level and the other difference is that the "store" time was in minutes to hours for UUCP, as opposed to microseconds for IP.

RF: What role did you experience USENIX meetings playing in getting an effective email network started?

CC: Well, it really was a confluence of time and events. Because of USENIX we were meeting. Most people could not be part of the ARPANET for reasons I've already covered. Because Version 7 UNIX included UUCP, everyone now had a way to send intersite email if you had at least a Version 7 UNIX box, a modem and a friend with an auto-dialer. Remember that self-dialing modems didn't exist yet and to dial out to another site required a DN11 and a Bell model 801 ACU—automatic calling unit—Truscott's trick not withstanding.

References

[1] Clem Cole's post about the origins of `fsck`: https://minnie.tuhs.org/pipermail/tuhs/2017-May/011467.html.

[2] J. Lions, "Commentary on UNIX 6th Edition, with Source Code": https://www.amazon.com/Lions-Commentary-Unix-John/dp/1573980137.

[3] B. Kernighan and D. Ritchie, "The C Programming Language": https://en.wikipedia.org/wiki/The_C_Programming_Language#History.

[4] "The UNIX Wars": https://en.wikipedia.org/wiki/Unix_wars.

[5] Metcalf's Law: https://en.wikipedia.org/wiki/Metcalfe%27s_law.

[6] P. Salus, "Distributing the News: UUCP to UUNET," *;login:*, vol. 40, no. 4 (August 2015): https://www.usenix.org/system/files/login/articles/login_aug15_09_salus.pdf.

[7] R. Farrow, "Interview with Mary Ann Horton," *;login:*, vol. 45, no. 1 (Spring 2020): https://www.usenix.org/publications/login/mar20/horton

The USENIX meetings had been around for 8–10 years before Usenet comes into play. But it was already clear before what we now call the Internet replaced ARPANET, that people wanted/needed email—that Usenet was organically born.

Again, it was need and timing more than anything else that helped get UUNET started, plus the wild growth of the Internet we saw in the 90s.

RF: How did your involvement with USENIX change over the years?

CC: USENIX has a special place in my heart. Without a doubt it helped my career. When I first started coming I was in the audience soaking things in, then I transitioned to someone writing and presenting papers. Next I was asked to be on program committees and eventually chair a few conferences. I was nominated and elected to the Board and eventually became president. I still participate as I can and I would consider working on the Board again as well as other projects that folks consider.

Interview with Kirk McKusick

Rik Farrow

Dr. Marshall Kirk McKusick writes books and articles, teaches classes on UNIX- and BSD-related subjects, and provides expert-witness testimony on software patent, trade secret, and copyright issues particularly those related to operating systems and filesystems. He has been a developer and commiter to the FreeBSD Project since its founding in 1993. While at the University of California, Berkeley, he implemented the 4.2BSD fast filesystem and was the Research Computer Scientist at the Berkeley Computer Systems Research Group (CSRG) overseeing the development and release of 4.3BSD and 4.4BSD. He earned his undergraduate degree in electrical engineering from Cornell University and did his graduate work at the University of California, Berkeley, where he received master's degrees in computer science and business administration and a doctoral degree in computer science. He has twice been president of the board of the USENIX Association, is currently a board member and treasurer of the FreeBSD Foundation, a senior member of the IEEE, and a member of the USENIX Association, ACM, and AAAS.

In his spare time, he enjoys swimming, scuba diving, and wine collecting. The wine is stored in a specially constructed wine cellar (accessible from the Web at http://www.mckusick.com/~mckusick/) in the basement of the house that he shares with Eric Allman, his partner of 40-and-some-odd years and husband since 2013. mckusick@mckusick.com

I first met Kirk McKusick at a USENIX conference in the 1990s. By that point I was working with Dan Klein and others on the tutorial committee and listened to portions of all tutorials given during LISA conferences, so I might have met Kirk that way. Later, Kirk and I would sometimes meet during FAST workshops.

Once when we were sitting together during paper presentations, someone presented a method of speeding up `fsck` on Linux `ext` filesystems by caching the results of intermediate phases. Feeling a bit mischievious, I mentioned to Kirk that this sounded like an improvement that belonged in the Fast File System (FFS), something Kirk had written, taught, and still supported in BSD. Kirk replied that this should be easy, as policy and implementation were kept separate in BSD, unlike in Linux. By the next morning, he had created a new version of `fsck`.

Rik Farrow: When did you first encounter UNIX?

Kirk McKusick: I encountered UNIX for the first time while at the University of Delaware in 1976. Later that year, I was a graduate student at the University of California, Berkeley (UCB), and started the month after Ken Thompson ended his sabbatical, in August of 1976. Thompson had helped install UNIX Version 6 on a PDP 11 there, working with Chuck Haley and Bill Joy, two other UCB graduate students. They also worked on a version of Pascal Thompson had written, and demand for that lead to the first Berkeley Software Distribution in 1977 [1].

RF: When and why was the Computer Science Research Group (CSRG) started?

KM: Professor Bob Fabry, with help from Bill Joy, had been working on getting a research grant from DARPA and needed a project name, so he decided to call the project the Computer Systems Research Group (CSRG). That was in June of 1980.

RF: You are best known as the author of the Fast File System, today known as the UNIX File System [2]. How did that come about?

KM: The filesystem developed for the early UNIX versions had terrible performance, getting throughput of only 2% of the bandwidth of current disks. Doubling of block size, to 1024 bytes, managed to raise the throughput to 4%, so this area seemed like fertile ground for research. I was working for the university, but part time, as full time work would have required the university to provide benefits as well—still an issue today. My advisor's research grant had ended during the summer, and I asked Bill Joy, who I used to share an office with, if he could give me a project to work on.

Joy had started work on a filesystem prototype, but had only written the superblock and cylinder group structures so far. He handed the project off to me, and I finished the rest as a userspace file system.

Joy convinced me to drop the prototype into the kernel, and that took me months, as there are concurrency and race conditions as well as other things, like cache invalidation, to handle. Then Joy convinced me to store my own home directory on the new filesystem, to show that I believed in my work. I realized that there was no way to do backups, so I wrote `dump` to backup and `restore` to recover from backups.

I also got tired of running `icheck`, `dcheck`, and `ncheck`, the first three passes you'd see with `fsck`, so I got `fsck` running. All this was a sidetrack on my way to getting my PhD.

Later on, Joy funded my trip to a Boston USENIX conference with DARPA money. I created a couple of hand-written slides, and Joy took them to get typed up. There were about 1200 attendees when I went to speak, but the slides Joy had provided were nothing like the ones I had written. When I told people that, they laughed, and the presentation went well.

The FFS could get around 40% of the bandwidth of disks, 10x the performance of the older filesystem. I learned from this experience that you should pick problems where there is a lot of fertile ground.

References

[1] S. Pate, *UNIX Filesystems: Evolution, Design, and Implementation*, (Wiley, 2003): https://www.oreilly.com/library/view/unix-filesystems-evolution/9780471456759/chap01-sec008.html

[2] K. McKusick, "A Brief History of the BSD Fast File System," *;login:*, vol. 32, no. 3 (June 2007): https://www.usenix.org/system/files/login/articles/584-mckusick.pdf

;login: and Open Access

Laura Nolan

As a USENIX volunteer for the past several years, one of the things I value most about USENIX is the association's unequivocal support for open access—the distribution of research online free of cost or other access barriers. Paywalled content restricts the spread of ideas and knowledge, and it reduces the impact of the work that researchers and writers do.

USENIX's support for open access has been a long-held stance. In our announcement of our move to open access in March 2008 we said that "we hope to set the standard for open access to information, an essential part of our mission." We do set that standard by making open access as straightforward as it can be: all USENIX authors and speakers retain copyright of their work, and all proceedings and recordings are freely published online—no money, or even a registration process, is needed to view any USENIX open access content.

USENIX became open access for all conference proceedings in 2008, followed by recordings of talks in 2010. Since 2010, *;login:* has been the sole exception to our open access philosophy, with access to each edition of the digital version of the magazine restricted to USENIX members for the first year after publication.

;login: is changing after this issue. From the start of 2021 it will no longer be a print magazine. But *;login:* will still be here, just in a different form—and this transition also brings the opportunity to resolve this final anomaly in USENIX's commitment to open access. All new *;login:* content will be available to all immediately after publication.

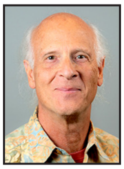
;login: is an old friend now and part of me is sad to see this chapter of its history end. However, as a regular contributor to *;login:*, I welcome the opportunity to share the content I create more widely. I believe that *;login:* is a very special part of USENIX—it spans the divide between research and industry, and it truly is a reflection of the best of all that we are. Making *;login:* freely accessible to all will make it even more meaningful.

Our Favorite *;login:* Articles, 2005–2019

Rik Farrow, Laura Nolan, and Arvind Krishnamurthy

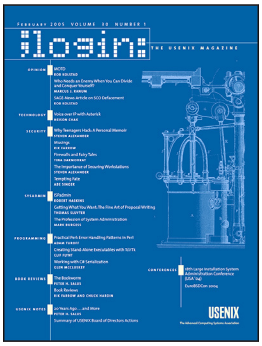
When we learned that print *;login:* was to end with this, the Winter 2020 issue, several of us decided that we would pick out some of our favorite articles published during the previous 15 years. To help divide the work of skimming through the 90 issues published during the period, we each focused on a four- or six-year set of issues.

Of course, you probably have your own favorites. One problem with printed magazines is that feedback is rare, other than the occasional email or comment made to the editor during a conference. Some of Rik's favorites are based on data collected from Web server log files. The new, digital format will provide more opportunities for engagement, with the ability to add comments to articles. We hope you enjoy this walk through *;login:*'s rich recent past and look forward to introducing you to new articles that continue this tradition in its new medium.



Rik Farrow: 2005–2010

Although I had the ability to use the log analyses I had collected during these years, I've included articles that were not just popular over time, but also some that I particularly liked. In a way, this was a difficult task for me, because most articles were published because I liked the topic and felt that the authors involved could write well. Also, at least one article in each issue was designated as open, meaning that those articles, typically the first article after "Musings," had an advantage over other articles that wouldn't be available for download by non-members for another year. For these reasons, I didn't adhere to the rankings found in the log analysis obsessively, and included other articles.



2005

I started working as editor this year, and I will confess that as I worked through this six-year period, the articles did get better over time. Editing *;login:* really was a learning experience for me.

I had been surprised at some of the things that appeared high in the log analysis when I first started. *;login:* included a column on C# by Glen McCluskey, and his most popular column was about serialization in C# [1]. Another column, this one by Adam Turoff, explained date and time formatting in Perl [2]. Ceph was introduced this year as well, with an article by the authors [3].

2006

Steve Johnson, past USENIX Board President, and author of yacc, lint, and the Portable C compiler while at Bell Labs, wrote one of my favorites, an article about how hardware affects performance. Steve demonstrated how the stride affects performance when processing large arrays of 64 bit values [4].

Rob Thomas and Jerry Martin of Team Cymru wrote about their very practical research into the underground economy. They joined dozens of underground sites, learning about just how much a stolen credit card or bank account is worth on the black market [5].

2007

Simson Garfinkel needed to analyze vast quantities of data collected from hard drives, and working with Amazon saved him thousands of dollars in hardware costs. Simson explained how he used S3 and EC2, sharing his experiences with what were new services at the time [6]. There were also two articles about Xen, the hypervisor technology used at Amazon, in this issue.

Sam Stover, Dave Dittrich, John Hernandez, and Sven Dietrich installed malware on Windows systems so they could analyze the Storm and Nugache trojans. What made these trojans different was that they used P2P communication instead of IRC for command and control [7].

2008

Edward Walker wondered if Amazon's cloud could take the place of large clusters for scientific computing, and discovered that there are definitely differences between a cluster you control and configure and the cloud [8]. There was also an article about Solaris virtualization options in this issue.

David N. Blank-Edelman wrote one of his most popular Perl columns, where he used the open source Timeline tool and some Perl modules to convert `crontab` files into Gantt charts [9].

2009

Alva L. Couch wrote "Is It Easy Being Green?", an article about the two different types of green, ecology and money [10].

As a nice example of the variety of work found in *;login:*, Rudi Van Drunen had a popular article about hardware. Rudi wrote about digital and analog signals, how they work and how analog gets converted into digital signals: [11].

2010

Konstantin Shvachko, one of the authors of HDFS, penned an article about the limitations of HDFS, due to its design: [12]. The same issue had another Ceph article, pointing out Ceph's scalability.

Andrew Tanenbaum, Raja Appuswamy, Herbert Bos, Lorenzo Cavallaro, Cristiano Giuffrida, Tomáš Hrubý, Jorrit Herder, Erik van der Kouwe, and David van Moolenbroek published an update on Minix3 [13]. There had been two other Minix3 articles published in *;login:* during the decade. This article focused on the ability to restart portions of the kernel, a topic of the first paper at OSDI '20.

Laura Nolan: 2011–2015

The time period that I reviewed for this piece is also when I first started to attend USENIX events and to read *;login:*. Rereading the editions from these years made me incredibly nostalgic (and not only for in-person conferences!). It was very difficult to choose only one article for each year, to the extent that I gave up and quite frankly, just cheated.

usenix ;login:
DECEMBER 2011 VOL. 36, NO. 6
The Danger of Unrandomized Code
EDWARD J. SCHWARTZ
Of Headless User Accounts and Restricted Shells
JAN SCHAUMANN
DevOps from a Sysadmin Perspective
PATRICK DEBOIS
Conference Reports from the 20th USENIX Security Symposium, CSET, HotSec, and more
usenix

2011

;login: has a very strong track record on security. The article I've chosen to represent 2011 is Sergey Bratus, Michael E. Locasto, Meredith L. Patterson, Len Sassaman, and Anna Shubina on "Exploit Programming: From Buffer Overflows to 'Weird Machines' and Theory of Computation" [14]. This article was dedicated to the memory of one of the authors, Len Sassaman, who had passed away earlier that year. It's a very thoughtful piece that characterises well-known security exploits (such as printf-family string format vulnerabilities) as a form of "weird instruction," and casts security as a problem of computability: what execution paths can our programs be trusted not to take, under any circumstances?

2012

My favourite article from 2012 is an example of cascading failure writ small: "Understanding TCP Incast and Its Implications for Big Data Workloads" by Yanpei Chan, Rean Griffith, David Zats, Anthony D. Joseph, and Randy H. Katz [15], which provides a systems model that explains pathological network throughput problems seen in early big-data systems. It's also a research and industry collaboration, which is apt for an association that spans industry and academia.

2013

;login: has had a variety of wonderful regular columnists, but none can top James Mickens for sheer entertainment value. His 2013 column "The Saddest Moment" [16] combines savage satire of papers and presentations about Byzantine fault tolerance with effortless education on the topic.

2014

2014 was a tough year to pick one favourite, because this year included both Brendan Gregg's debugging mystery "The Case of the Clumsy Kernel" [17], as well as "Analysis of HDFS under HBase: A Facebook Messages Case Study" by Tyler Harter, Dhruba Borthakur, Siying Dong, Amitanand Aiyer, Liyin Tang, Andrea C. Arpaci-Dusseau, and Remzi H. Arpaci-Dusseau [18], which demonstrates how "mechanical sympathy" between workloads and the infrastructure they run on is critical at scale, but can easily get lost with layered abstractions.

However, for sheer controversy, the article of the year has to be Todd Underwood's "The Death of System Administration" [19], based on his LISA keynote in 2013. Underwood proposes a future where operations engineers with software sensibilities (or vice-versa) working with better platforms will supersede manual systems administration work. We may not be sitting on the couch sipping bourbon and eating bon-bons quite yet, but I think Underwood is fundamentally correct about the direction we're traveling in.

2015

My 2015 pick (albeit with an off-by-one error) is a brace of articles about bugs. Ding Yuan, Yu Luo, Xin Zhuang, Guilherme Renna Rodrigues, Xu Zhao, Yongle Zhang, Pranay U. Jain, and Michael Stumm's article "Simple Testing Can Prevent Most Critical Failures: An Analysis of Production Failures in Distributed Data-Intensive Systems" [20] highlights how basic attention to error-handling code can increase the reliability of production systems.

"What Bugs Live in the Cloud?: A Study of Issues in Scalable Distributed Systems" by Haryadi S. Gunawi, Thanh Do, Agung Laksono, Mingzhe Hao, Tanakorn Leesatapornwongsa, Jeffrey F. Lukman, and Riza O. Suminto [21] analyses three types of troublesome bugs found in distributed systems such as Hadoop, HDFS, HBase, Cassandra, ZooKeeper, and Flume. The analysis of the varieties of "SPoF bugs" that can crash entire systems that are intended to be redundant should be required reading for all software engineers and SREs.

Arvind Krishnamurthy: 2016–2019

I focused on articles published over the last few years and what struck me was the rich diversity of the articles. *;login:* has routinely included articles from both academia and industry, often provided tutorials on recent developments in software engineering, and discussed emerging trends in the computing industry.

2016

My favorite article from 2016 is "Pivot Tracing: Dynamic Causal Monitoring for Distributed Systems" by Jonathan Mace, Ryan Roelke, and Rodrigo Fonseca [22]. Debugging distributed systems using logs is a difficult task, as what is recorded on logs is defined a priori and since it is hard to correlate log entries across a distributed system. Pivot tracing provides a novel approach that combines dynamic instrumentation with causal tracing and is thus suitable for production systems.

2017

My 2017 pick is an article describing an industry system that is in widespread use. Daniel Firestone describes a cloud-scale programmable virtual switch in "VFP: A Virtual Switch Platform for Host SDN in the Public Cloud" [23]. The article describes how Microsoft Azure enforces SDN policies across its large datacenters using the virtual switch. In addition to laying out the motivation for building the system, the article describes the design constraints that are unique to a public cloud.

;login:

SPRING 2016 VOL. 41, NO. 1

usenix THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

Filebench: A Flexible Framework for File System Benchmarking
Vasily Tarasov, Erez Zadok, and Spencer Shepler

Pivot Tracing: Dynamic Causal Monitoring for Distributed Systems
Jonathan Mace, Ryan Roelke, and Rodrigo Fonseca

Streaming Systems and Architectures: Kafka, Spark, Storm, and Flink
Jayant Shekhar and Amandeep Khurana

BeyondCorp: Design to Deployment at Google
Barclay Osborn, Justin McWilliams, Betsy Beyer, and Max Saltonstall

Columns

Red/Blue Functions: How Python 3.5's Async IO Creates a Division Among Function
David Beazley

Using RPCs in Go
Kelsey Hightower

Defining Interfaces with Swagger
David N. Blank-Edelman

Getting Beyond the Hero Sysadmin and Monitoring Silos
Dave Josephsen

Betting on Growth vs Magnitude
Dan Geer

Supporting RFCs and Pondering New Protocols
Robert G. Ferrell

2018

For 2018, I picked a practitioner's guide to working with XDP, a new programmable layer in the kernel network stack. In the article, "XDP-Programmable Data Path in the Linux Kernel" [24], Diptanu Gon Choudhury provides background information on Berkeley Packet Filter, a core kernel technology introduced almost two decades ago, and how it has been recently extended to provide a power programmable layer inside the kernel that is intended to close the performance gap with respect to kernel-bypass solutions.

2019

My favorite article from 2019 is "Noria: A New Take on Fast Web Application Backends" by Jon Gjengset, Malte Schwarzkopf, Jonathan Behrens, Lara Timbó Araújo, Martin Ek, Eddie Kohler, M. Frans Kaashoek, and Robert Morris [25]. This article describes a system that addresses performance problems faced by many web application backends. It outlines a system design that doesn't neatly fit into traditional categories, such as databases or streaming engines, but rather creates a bridge across these technologies.

References

[1] "Working with C# Serialization," by Glen McCluskey, February, 2005: https://bit.ly/3e5IUfV.

[2] "Date and Time Formatting in Perl," by Adam Turoff, June, 2005: https://bit.ly/2TGKFqw.

[3] "Ceph as a Scalable Alternative to the Hadoop Distributed File System," by Carlos Maltzahn, Esteban Mol Ina-Estolano, Amandeep Khurana, Alex J. Nelson, Scot T A. Brandt, and Sage Weil, August, 2010: https://bit.ly/3jItHTg.

[4] "Algorithms for the 21st Century," by Stephen C. Johnson, October, 2006: https://bit.ly/3kMIvRS.

[5] "The Underground Economy: Priceless," by Rob Thomas and Jerry Martin, December, 2006: https://bit.ly/3jG0hF7.

[6] "Commodity Grid Computing with Amazon's S3 and EC2," by Simson L. Garfinkel, February, 2007: https://bit.ly/2HR2n87.

[7] "Analysis of the Storm and Nugache Trojans: P2P Is Here," by Sam Stover, Dave Dittrich, John Hernandez, and Sven Dietrich, December, 2007: https://bit.ly/35RrNuB.

[8] "Benchmarking Amazon Ec2 for High-Performance Scientific Computing," by Edward Walker, October, 2008: https://bit.ly/38hFTbh.

[9] "Practical Perl Tools: Back in Timeline," by David N. Blank-Edelman, April, 2008: https://bit.ly/3n1zjKd.

[10] "Is it Easy Being Green?" by Alva L. Crouch, June, 2009: https://bit.ly/32ltMGw.

[11] "Signals," by Rudi Van Drunen, June, 2009: https://bit.ly/36aMim1.

[12] "HDFS Scalability: The Limits to Growth," by Konstantin V. Shvachko, April, 2010: https://bit.ly/3exxeCE.

[13] "MINIX 3: Status Report and Current Research." by Andrew Tanenbaum, Raja Appuswamy, Herbert Bos, Lorenzo Cavallaro, Cristiano Giuffrida, Tomáš Hrubý, Jorrit Herder, Erik van der Kouwe, and David van Moolenbroek, June, 2010: https://bit.ly/2IvP2T4.

[14] "Exploit Programming: From Buffer Overflows to 'Weird Machines' and Theory of Computation," by Sergey Bratus, Michael E. Locasto, Meredith L. Patterson, Len Sassaman, and Anna Shubina, December, 2011: https://bit.ly/3kkSRr5

[15] "Understanding TCP Incast and Its Implications for Big Data Workloads," by Yanpei Chan, Rean Griffith, David Zats, Anthony D. Joseph, and Randy H. Katz, June, 2012: https://bit.ly/3pmmE6w.

[16] "The Saddest Moment ," by James Mickens, May, 2013: https://bit.ly/2Uft09H.

[17] "The Case of the Clumsy Kernel" by Brendan Gregg, April, 2014: https://bit.ly/3eQiGhD.

[18] "Analysis of HDFS under HBase: A Facebook Messages Case Study" by Tyler Harter, Dhruba Borthakur, Siying Dong, Amitanand Aiyer, Liyin Tang, Andrea C. Arpaci-Dusseau, and Remzi H. Arpaci-Dusseau, June 2014: https://bit.ly/32B9Csc.

[19]"The Death of System Administration, " by Todd Underwood, April, 2014: https://bit.ly/3loTilN.

[20] "Simple Testing Can Prevent Most Critical Failures: An Analysis of Production Failures in Distributed Data-Intensive Systems, by Ding Yuan, Yu Luo, Xin Zhuang, Guilherme Renna Rodrigues, Xu Zhao, Yongle Zhang, Pranay U. Jain, and Michael Stumm, February, 2015: https://bit.ly/3kpawy1.

[21] "What Bugs Live in the Cloud?: A Study of Issues in Scalable Distributed Systems" by Haryadi S. Gunawi, Thanh Do, Agung Laksono, Mingzhe Hao, Tanakorn Leesatapornwongsa, Jeffrey F. Lukman, and Riza O. Suminto, August, 2015: https://bit.ly/38vGjuV

[22] "Pivot Tracing: Dynamic Causal Monitoring for Distributed Systems" by Jonathan Mace, Ryan Roelke, and Rodrigo Fonsec, Spring, 2016: https://bit.ly/3eONpf0.

[23] "VFP: A Virtual Switch Platform for Host SDN in the Public Cloud," by Daniel Firestone, Fall, 2017: https://bit.ly/2JUdHkB.

[24] "XDP-Programmable Data Path in the Linux Kernel," by Diptanu Gon Choudhury, Spring, 2018: https://bit.ly/32DxrPY.

[25] "Noria: A New Take on Fast Web Application Backends," "Noria: A New Take on Fast Web Application Backends" by Jon Gjengset, Malte Schwarzkopf, Jonathan Behrens, Lara Timbó Araújo, Martin Ek, Eddie Kohler, M. Frans Kaashoek, and Robert Morris, Spring, 2019: https://bit.ly/38wzVna.

01.

“Amazing product, developed by some of the most seasoned pros in the industry.”

02.

“Great products that work, easy and quick to install and provide real value.”

03.

“We 🖤 our canaries.”

04.

“The concept and use of Canarytokens has made me very hesitant to use credentials gained during an engagement. If the aim is to reduce the time taken for attackers, Canarytokens work well.”

05.

“Their on-prem canary is one of the only things that caught me right away in post-exploitation without my knowing I was burned. Solid concept and product.”

06.

“Don’t think just get them.”

Attackers and Defenders Finally Agree

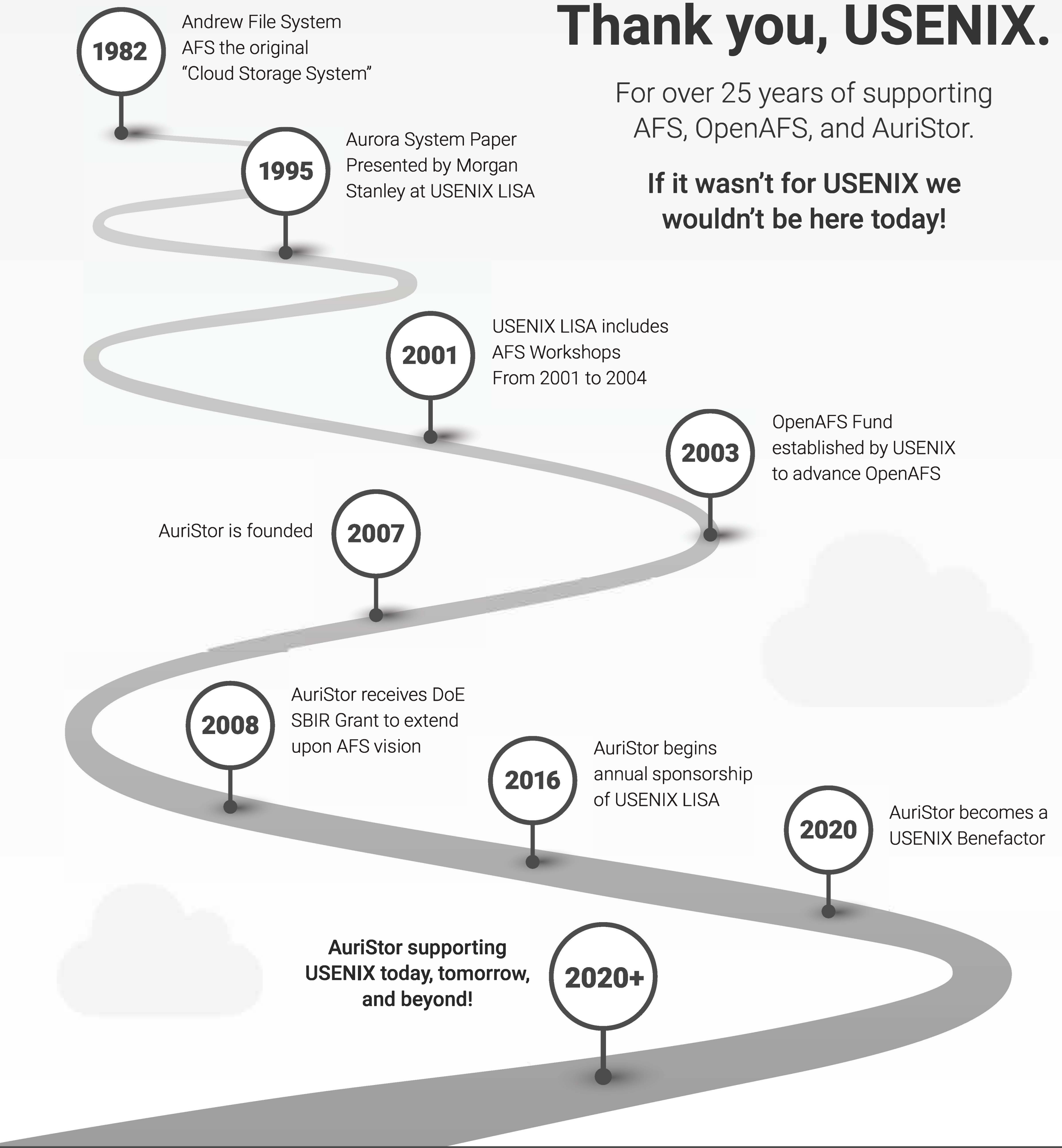
https://canary.tools/love

THINKST
CANARY

Thank you, USENIX.

For over 25 years of supporting AFS, OpenAFS, and AuriStor.

If it wasn't for USENIX we wouldn't be here today!



1982 Andrew File System AFS the original "Cloud Storage System"

1995 Aurora System Paper Presented by Morgan Stanley at USENIX LISA

2001 USENIX LISA includes AFS Workshops From 2001 to 2004

2003 OpenAFS Fund established by USENIX to advance OpenAFS

2007 AuriStor is founded

2008 AuriStor receives DoE SBIR Grant to extend upon AFS vision

2016 AuriStor begins annual sponsorship of USENIX LISA

2020 AuriStor becomes a USENIX Benefactor

2020+ **AuriStor supporting USENIX today, tomorrow, and beyond!**

AURISTOR®

THE GLOBAL NAMESPACE FILE SYSTEM

Learn more about AuriStorFS at **auristor.com/filesystem**