

A Quarter Century of LISA (with Apologies to Peter Salus)

SEAN KAMATH



Sean Kamath is a Production Engineer at Facebook. He has been a lurker at LISA conferences since 1992. kamath@moltingpenguin.com

In 1992, I had just hired another sysadmin named Dan. One day he asked me about going to this conference, which I had never heard of. I'd never been to any sort of conference before and had no idea what to expect. What we found, however, was amazing. We had found our people. It seemed like everyone there was just like us, all struggling with the same things, all looking for similar solutions. In retrospect, in such a young industry, this wasn't all that surprising. Perhaps the most surprising thing, however, was the vaunted "hallway track" and learning the backgrounds of everyone I met. I don't actually recall meeting a single CS major. Science was well represented (I'm a physics major, and there were plenty of math, biology, chemistry, and physics majors), but in a strange twist there were also plenty of music, art, theater, literature, and all sorts of other liberal arts majors.

This was my first LISA conference, LISA VI, held in Long Beach, CA. You won't find it on the USENIX website (you can find a reference to it if you try hard enough). I have attended every LISA conference since then, the last being the 31st LISA, held in San Francisco. While I've written before of my experience attending LISA ("Whither LISA," *login*: February 2010), what follows are my thoughts and observations about the conference over these 25 years rather than a history of the first 25 years. When I started in the system administration field, the sysadmin was a mysterious creature. Most companies didn't know they needed one. Most probably didn't need one. My first UNIX experience, when I went to college in 1984, was on the school's VAX running BSD4.1.

What a lot of people these days don't know is that system administration back then was very often done by one of two types of people: folks who wanted to be software developers but weren't quite there, and folks who just fell into it. People in the former camp usually only did the job for a year or two until they could get a job as a developer. The rest of us? We just did it because we could.

So this was the beginning of my experience at LISA, a conference full of people doing something they hadn't been formally trained to do but who did it because they loved it or were good at it (often both), where the goal of attending was to learn, experience, discover. Swag was nonexistent and unexpected. Vendors didn't exist. It was a conference set up and run by people attending. It was like a bunch of geeks just congregating in the same place and spending a week talking about everyone they were dealing with.

I hope I've set the stage appropriately as an attendee. However, I should take a moment to say that I know that organizing this seemingly spontaneous congregation of sysadmins actually took a lot of work, and that a lot of dedicated, smart, and talented people made it appear completely seamless. Further, I believe that over the years, as conference organizers have come and gone, it has become increasingly hard to do what they do. Having been on the papers committee, I know how incredibly hard it is to find quality papers, and I was not at all surprised when the refereed papers track went away, a victim of the internet, in my opinion.

As I look back on these last 25 years, I see a similarity to my college years. I've watched LISA start out as a freshman and develop into a senior. And, conveniently for me, I had two junior years, so that makes it a nice five years per...year. I'm reminded that no analogy is perfect.

Freshman

As I've explained, sysadmins during this time were mostly people who fell into the job. When I hired Dan (who had been at the company a long time in a different role), he admitted to me that he was unsure whether this was a direction he wanted to go in, uncertain whether system administration was something you could do for an extended period. In short, it wasn't a career path. And yet, at that first conference we attended, in one of the tutorials or talks (time clouds my memory), someone said, "It's OK to make system administration your career." Dan was so relieved to hear this. I was too young and naïve to realize it was a watershed moment until much later.

Sometimes it's hard to explain to people that this time was at the very beginning of the internet as we know it. Forget smartphones—this was before the graphical web browser (NCSA Mosaic was released in 1993). Discussions would often mention Gopher. What this meant in practical terms was that we all tended to be isolated, with no way to easily see new things being created, no easy way to discover new tools, no way to broadcast that you had written this cool thing that did something everyone needed done. It's not that there was no way (I'm looking at you, NetNews), but it took time.

In addition, so many building blocks of what we know and love today, while not exactly hot off the presses, were still pretty darn new. Everyone ran their own mail server, people were excited about using DNS—for large installations, DNS was almost a must—and might have used BOOTP (defined in an RFC from 1985), but it wouldn't be until 1997 that anyone was using DHCP. It could take years for enough people to realize their problems were similar to others', to craft some notion of how their problems might be solved, and to actually write and disseminate code.

This is what made LISA so amazing. We could all get together, those of us charged with the responsibility of running large installations, and accelerate the dissemination of these tools, practices, ideas, and experiences. One of the great things about LISA at this time was that the refereed papers were where people could publish their code. The loudest groans from the attendees would come when someone who'd written a paper on some great new tool, when asked about getting it, would reply, "We're still trying to see if we can release this."

It wasn't just the technical side where we came together and discussed things. Rob Kolstad's tutorial on the ethics of system

administration opened our eyes to issues never thought of. There were sessions on being a manager. On running a team. On the human side of dealing with so many users.

In many ways, this was the Golden Age of LISA. It was achingly relevant. It was so necessary. It was everything—reckless, bawdy, outrageous, as well as thoughtful, helpful, and just plain wonderful.

Sophomore

As the profession of system administration grew and developed, things got more serious. No longer were the hotel room parties stuffed with people, with a bathtub full of beer and ice in the bathroom, the bed tipped up against the wall (my first BayLISA "hospitality suite"). Now they were stuffed with people, hosted by companies looking to hire. People, and companies, thought that much of what the sysadmin did was a "solved problem." If not solved, well, one or two software updates away from being solved. Just look at configuration management. (CFEngine vs. Bcfg2! And, yow, the Wikipedia page on CFEngine doesn't even mention Bcfg2.)

This was an age of vendors trying so hard to provide solutions for all the companies that were just starting to be aware that they needed people to run their computers, but couldn't find the people that had the experience to do the job well. No longer could you hire a wannabe software developer to run the computer. And when the sysadmin failed, real work was lost, either in lost time or lost data.

System administration was All Things Computer around this time, which meant trying to find ways of managing PCs as well as the central servers. The conference started to draw people from all sorts of shops, not just "Large Installations." The internet was starting to take off, especially for high-tech companies. This had two effects on the conference. First, it got the word out. People would put up URLs to get their code. Mailing lists cropped up, and communities formed around software. Second, and perhaps somewhat detrimental to the existing conference format, the ability to directly publish to the web meant that some folks started to release software, and then talk about it at LISA.

This period created two diametrically opposed problems for those running the conference. It saw the dot-com bubble, as well as the dot-com bust. And with that came the massive growth of the conference, followed by an implosion of attendees. Imagine one year with over 2000 attendees, and the next, fewer than 1000. I don't know the actual numbers, but I was there. One year you couldn't find a place to sit at the tutorial lunches, the next was a grim affair where we ate in silence with a lot of empty seats at all the tables. The number of companies that showed up

A Quarter Century of LISA

and plied their wares similarly imploded—not just the vendors, but the companies sending their employees. Where once multiple people would show up from major companies, there now was one person who got to go that year.

It was a grim time.

Junior

The conference had to find a way to stay relevant after the dot-com bust. Not because what it had to offer wasn't relevant anymore, far from it. All the companies out there still needed people to do their job. And now, more than ever, they needed to be as efficient as possible. Just like my (first) Junior year in college, it was time for the conference to buckle down and get to work.

The pace of innovation slowed only slightly around this time. People couldn't afford to attend the conference, but problems still needed to be solved. The dot-com bubble brought a whole new paradigm for how systems could be used and for the importance of the internet. With it came the challenges of running systems that people other than your users used.

Around this time, I think the conference reached the peak of trying to be all things to all people. Run PCs? Come to LISA. Run internet services? Come to LISA. Running corporate infrastructure? Networking? Storage? Suddenly, "All Things Computer" was simply too much. You could see it in the huge expanse of the refereed papers, invited talks, and BoFs.

The culmination of these two things (people couldn't afford to attend, and the conference trying to appeal to as broad an audience as possible to attract more attendees) resulted in both a wide variety of topics covered, as well as a similarly wide variety in quality and applicability. One thing to keep in mind about this time, the early aughts, is that even after the bubble burst there was an explosion of opportunity for companies with this internet thing, as well as for corporate systems (everyone had a computer at work, it seemed).

Additionally, while we had accepted methodologies for how to provide certain services, etc., the pace of solutions couldn't keep up with the new problems companies and universities found as they expanded. Issues of scalability, reliability, reachability, accountability, securability—well, you get the idea. These issues outstripped everyone's ability to find solutions.

The upshot of all this was that the conference tilted away from direct solutions and toward explorations of possible solutions, along with descriptions of problems encountered. The invited talks expanded from one track to two and became better attended as people looked for insights into problems they were running into. Workshops were added to the training, serving as a way to delve more deeply into problems that were moving from edge cases to commonly encountered.

Junior (Redux)

What happens when what you thought you were doing correctly turns out to need a course correction? It was in the middle of this period that I wrote "Whither LISA," my attempt to reflect on what LISA meant to me and to issue a call to action to come together and infuse new life into the conference. I felt that LISA had, to some extent, reached a crisis point. It could no longer be the conference that covered everything. There were other conferences competing for attendees, with a slightly different slant.

I think a lot of the genesis for this stems from the growth of the field. We went from system administrators to system engineers, then system/network/storage engineers, and then...site reliability engineers. Then along came DevOps.

When I think about the evolution of the role of "people who make machines work," I think of this: the very first system administrators were the people who wrote the operating systems for the simple reason that they built the OS and there was no one else to manage it. As with all things run by the people who built them, it was assumed that the person running the system knew everything the person who built it knew.

Eventually, the UNIX distributions landed at sites with people who didn't build them. And thus, people who didn't have that intimate knowledge of how things were constructed were responsible for keeping things working nonetheless. And, as often happens, those people, freed of the presumption of how things *should* work, could often make things run quite well, sometimes in unexpected ways.

As the problems faced by sites grew more diverse, complicated, and involved, the need for specialization grew. At first, the need was for base technologies (storage, networking), but eventually it grew even to encompass what the system was designed to solve. The skills and tools used to provide a reliable environment for doing software development were often dramatically different from, and yet in some ways the same as, providing a thousand office workers with functional, up-to-date PCs.

And then came the web. Suddenly we had a whole new denizen of the system administrator realm: software developers working on systems that were providing services that system administrators might support, but the support was for customers. Sometimes they even paid.

As for the companies that had *huge* environments, well, they experienced the most problems. They also had the resources (people, money, motivation) to solve the problems. And many of those companies were very involved in open source software and were more than willing to share with their communities.

The result was that there was a bit of a dichotomy at LISA. There were the people from the large companies who were facing problems that were unique to their industry but who were driving the

talks. There were also the folks from smaller shops who started to feel sidelined. I can't tell you the number of times that I heard someone talk about the lack of applicability of talks and papers to their job. When LISA first ramped up, people who were new, even those from very small shops, found a lot of useful information they could use, sometimes right away. Fifteen years later, what was someone running a small shop of 100 people and a few servers going to do with the information about running Hadoop clusters? What was I, working at a fairly large animation studio, going to do with that information? I tried to glean as much useful information from the various sources as possible. At least my environment wasn't that far off from the big guys. But I saw a lot of folks from small shops stop coming and heard a lot of people say this was their first, and likely last, LISA.

Senior

Around 20 years after I started attending LISA, after much internal upheaval and change over those intervening years, LISA started to morph. Not a lot, but just enough. Instead of everything being about concrete "solutions," instead of restatement of problems, instead of a little bit of everything, there started to be a convergence. Two things started to become apparent.

First, a thread that had been common since the beginning of LISA became mainstream: it's not about the current problems; it's about how we navigate around to find solutions. It's about the habits, behaviors, and techniques that people should develop to help them find their way in this job. Be it managing managers, budgets, time, or systems, useful information about strategies and examples of people facing and handling challenges became a fundamental part of the tutorials, invited talks, and, of course, the hallway track. As a smart friend of mine recently said: "Most of what we as sysadmins, of whatever flavor, do is to learn enough about how something works, then adapt it so that it works in our environment. We learn from examples but must be able to apply those examples." So true.

Second, everyone was an internet company now. While everyone still had the challenge of running internal systems, that turned out to be fairly static. But after 20 years of trying to figure out how to provision and configure internal systems (be they PCs or whatever), people and companies (and vendors!) knew how to do it. Even the explosive growth of mobile was addressed in short order by vendors and companies with BYOD policies and apps.

What made this go-round different from the last one was this: The Cloud. Yeah, a lot of us laughed at the term. I still do. But it turned out that companies were thrilled to offload their work onto other companies. We saw everything as a service, to the point that these days we have Services as a Service (turtles all the way down). Companies were more than happy to farm out their email to a few large vendors. Speaking as someone who still runs a small personal mail server, this freaks me out. We saw

companies migrate (sometimes with good reasons) to cloud-based document storage and editing.

But the killer app turned out to be cloud-based servers. No longer did companies shell out capital for hardware to run their services. Instead, they could just rent the machines and have all the benefits of a highly paid professional staff to run them. Software to run all these machines was created, and all you had to do was learn how to use it.

And so, in a bizarre twist, the cloud became ouroboros. The companies that were providing *aaS and the customers of those companies had a convergence of interest, one solved by attending LISA. Both groups of people used the same, or similar, tools, albeit for very different ends. Sure, Ansible/Salt vs. Chef/Puppet replaced CFEngine vs. Bcfg2, as even cloud-based services need configuration management of some sort. If you needed a huge Hadoop cluster, no problem; fire up a couple thousand machines for a week or two.

Graduate School?

So where does that leave us now? Well, recently USENIX announced a significant change to the LISA conference. Gone are the dedicated three days of training. Tutorials will live on in the mini-tutorial, but the need for half a day or a full day to fully grok some key concepts is gone. People need and want self-guided training, or access to knowledgeable people to ask questions of. The conference will become smaller and more focused, I think. I don't know if smaller in size, but definitely compressed in time.

Change is hard. And for an old-timer like me, it's really scary. I will be interested in, and trepidatious about, the changes that are in store and how they will play out. I don't think this is the last change we'll see and with good reason. If the conference hadn't changed from that first one in 1992, I would have stopped going by 1995. And I have a lot of trust in the fine folks at USENIX and the people from our own community who make up the steering committee of the conference. Because one thing has remained constant despite the changes and tumult over all the years: it's still a conference by and for the people who are putting it on.