Interview with Rob Kolstad

RIK FARROW

After earning his PhD, Rob Kolstad spent eight years at successful supercomputer startup Convex Computers as the Manager of Operating Systems. During his tenure there, he was elected to serve the first of his three terms on the USENIX Board of Directors. He joined Prisma Technica, a Colorado Springs-based startup whose mission was to construct a high-speed gallium arsenide SPARC processor, which led to two years with Sun Microsystems, followed by seven at the first Internet server company, Berkeley Software Design (BSDi). He was also a program manager at SANS, and later Executive Director of SAGE for USENIX. Most recently he has been an intellectual property consultant, working on projects like analysis of voting machine code and patent infringement investigations (including disk drives, flash components, embedded software reverse engineering, and DVD recording formats). Rob also volunteered for 20 years as the head coach of the USA Computing Olympiad, training pre-college students to represent the US at annual international programming contests held around the world.
rob.kolstad@gmail.com

Rik Farrow is the editor of *;login:*.
rik@usenix.org

I believe I first met Rob Kolstad in the early '90s, during the time of the AT&T lawsuit against BSDi and the University of California. I was very interested in what was happening in that lawsuit, partially because I consulted for *UNIX World* magazine in those days, and largely because I believed in open source. Rob was the head of BSDi by then, and he arranged for me to receive versions of BSD as they were updated.

Rob also had history with USENIX by that time, having (among other activities) started LISA as a small workshop (see Peter Salus' article in this issue). Rob was the editor of *;login:* for many years too, as well as a popular speaker and conference organizer.

But there's surely more to his story, I thought, so I decided to see what I could elicit.

Rik: Tell us about your early days in computer science.

Rob: I was told "you must go to college" from the day Sputnik launched. My parents independently emphasized this starting from first grade when I mentioned standardized tests we had just taken. It was a given and as natural as being born or getting up in the morning. "You must go to college."

After spending half a dozen years as a very young programmer hanging out at the labs of the University of Oklahoma, I headed to SMU to work on one of the first computer science degrees. I followed some professors to Notre Dame for a master's degree in electrical engineering (all theory). From there it was six years at the University of Illinois, ending up with a thesis in the realm of distributed operating systems.

Convex Computer Corp. (then called Parsec Scientific Computers) was hiring OS folks to port UNIX to their new vector processor. As it turns out, I was one of the very few on the market at the time and thus joined them and learned how to play the startup game (this being the 1982 timeframe when supercomputers were the bubble instead of PCs or the Internet). Convex worked out well, going public and then increasing their share price.

I followed my boss to Colorado Springs to join Prisma Technica, who were going to manufacture a SPARC-compatible processor using gallium arsenide chips, which would have the then-blazing clock rate of 250 MHz. GaAs does have its challenges; the software group eventually ended up joining Sun Microsystems in order to ply their large system expertise for Sun's higher-end efforts.

After two years with Sun, I moved on to BSDi, whose goal was to commercialize Berkeley-style UNIX; it was my dream job. The lawsuit with AT&T over their misunderstanding of the provenance of BSDi's code cost over a million dollars and enabled the completely unencumbered Linux to gain its first foothold; the rest is history.

I then moved to work with conference-running organizations like USENIX and SANS before commencing independent consulting in the intellectual property world. Some of my projects there have included analyzing voting machine code, dissecting solid state disks in search of patent violations, and constructing the software to interpret and summarize high-speed bus traffic.

Rik: You started LISA in 1987 as the Large Installation System Administration conference. Can you provide some background about why you thought USENIX needed a conference about managing large numbers of UNIX systems? To be honest, I didn't even know there were large numbers of UNIX systems to manage in 1987.

Rob: I was privileged to attend USENIX conferences starting late in my graduate school career. It devolved in my circles that one's ticket to the conference was contingent upon publishing a paper there, and so that's what I did. I enjoyed the experience and worked on the things that seemed important: organizing late-night panels, helping with tasks, and generally slowly growing into a role of contributor (talks, tutorials, keynotes, chairing conferences, creating distribution tapes, etc.).

Early in my tenure at Convex Computer Corporation, USENIX founder Lou Katz suggested that I run for the USENIX Board of Directors. I wrote a platform statement that apparently resonated with the electorate and joined the Board.

I believe in "working boards" vs. those who use their contacts for fundraising (in the case of charities) or business promotion (in the case of corporations). I believe there was a general feeling that board members were in charge of presenting ideas for workshops, conference enhancements (including topics, tutorial ideas, and non-technical ideas), and generally creating the organizational plans which the main office then executed.

Of course, folks would often propose programmatic ideas in their own fields of interest, so I proposed a "Large Installation System Administrators' Workshop" (among others, including a supercomputer workshop that fared less successfully in the long-term). I was working as Manager of Operating Systems at Convex Computers; my job was to ensure that BSD UNIX ran seamlessly on the new Convex C1 affordable supercomputer. Even our organization of under 50 people was expanding into ever greater numbers of systems, and, of course, utilized large systems in order to ensure they worked when delivered to customers.

Some folks think back and wonder about how "large" a shop could syadmins in our marketplace administer. Here's an excerpt of the original announcement:

> [The] Large Installation System Administrator's Workshop [will] be held in Philadelphia, PA, on April 9th and 10th, 1987. This workshop will bring together system administrators trying to conquer UNIX's historical bias towards smaller systems. It is believed these administrators battle many of the same problems repeatedly and can share their unique solutions to some problems in order to avoid duplication of effort as UNIX grows to run in ever larger installations.
>
> System managers of shops with over 100 users (on one or several processors) will find this workshop particularly valuable....
>
> Some topics to be considered include: large file systems (dumps, networked file systems), password file administration (including YP), large mail system administration, USENET/News/Notes administration, mixed vendor (and version) environments, load control and batch systems, handy new utilities, and large LANs.

I seem to remember that, early on, one of the qualifications for "large" was "administration of over a gigabyte of disk storage." Times have changed!

The lack of unified tool sets, the complexity of the tools that did exist, and the emergence of new challenges all made large installations an interesting problem. Recall that this is back in the day when a system crash (which was not an infrequent occurrence) required running the fsck program to repair the disks before the system could be fully booted.

Rik: Can you tell us what's involved in coaching USACO participants? Do you help choose participants, create test problems, manage logistics, or do other work?

Rob: As the computer industry matured over the years, so did the USACO coaching techniques. In the earliest years, the problem was simply finding students who had the skills to participate. Later, there were more than enough students, so a camp was organized to perform training via lectures and labs (along with both technical activities like program AI-players for games and non-technical activities like ultimate frisbee), which led to a pair of five-hour on-site contests to select the representatives for the international contest.

A number of things became clear:

- Manual task grading was onerous, time-consuming, and had a potential for errors.
- A host of students could benefit from earlier exposure to the task types.
- Ultimately, the Web could provide a fertile training environment.

Thus, the coaches developed a grading system to run a contestants' program using a number of different data sets for input and provide results. While this might sound easy today, it was revolutionary in its time (hearken back to the days when people were mystified by UNIX, processes, the Web, and networking in general). That system ultimately ran several of the international competitions.

The coaches created test data for these tasks, and that became a challenge since erroneous test data leads to competitor

complaints and a general tarnishing of reputation. Russ Cox (now Dr. Cox, at Google) led the way to "validators" whose job was to ensure that the test data conformed to the task description and was valid. Ultimately, the system could read the input descriptions and generate the validators automatically; this proved to be an extraordinary technique for ensuring that non-conforming data did not sneak into the grading process.

The automated grading system meant that USACO could hold competitions on the Internet, which led to an era of six annual contests (five shorter contests at three hours each; one five hours). These contests made it much easier to choose the best contestants to attend camp, since so many "performance data points" were collected. Open to the entire Internet community, it was not unusual for more than 1000 competitors from more than 65 countries to participate (several of whom were past their pre-college career but wanted the exposure and were ranked separately). A later upgrade enabled contests to be translated quickly, right before a contest began, thus expanding their scope.

Once that challenge was met, it was obvious that an online training system could move the students more quickly through the complexities of "algorithmic programming." This engendered "The USACO Training" pages at http://train.usaco.org. Open to the entire Internet community, the site enforces a path through a curriculum, with 100 sample tasks that amplify the concepts of its tutorial sections. Almost a quarter million folks from around the world are registered for the training pages, which have also been translated into half a dozen languages for international students.

Each contest featured three tasks in each of three divisions, nine tasks total. Six contests required 54 tasks, each with 10–20 test data sets, validation, several sample solutions in various programming languages, and an analysis to be shared after the contest. The USACO camp featured an additional 5–6 contests (only one or two divisions) of three tasks in addition to a game-AI-challenge or other challenge—a total of 18–36 more tasks and the special software to run the challenge. This totals up to at least 72 tasks per year, repeated with new tasks each year.

Keeping track of these tasks became a challenge, so I built the contest-task management system based on some heuristics developed by Greg Galperin, Hal Burch, and others at camp. It keeps track of task text, coaches' solutions, test data, validator, validation status, the analysis, the difficulty (and other) ratings, and other miscellaneous information. Ultimately, this system could create a contest with three divisions and ensure the contest started on time and finished on time—automation at its finest. Of course, being data-driven, one had to ensure that the data was present, a huge task recurring monthly through the school year.

In summary, yeah, it's a lot of work to run the season. A set of a dozen coaches online throughout the year complements perhaps eight coaches (usually the same folks) invited to keep the camp (now with more than 30 participants) running smoothly. Two coaches and four students participate in the IOI and, occasionally, other international competitions. It's great fun and a real inspiration for those students who are interested. See http://www.usaco.org for more information.