system administration research

Part 3: Challenges to System Administrators

by Mark Burgess
<*Mark.Burgess@iu.hioslo.no*>

Mark is an associate professor at Oslo College and is the author of cfengine and winner of the best paper award at LISA 1998.

The preceding two articles in this series tiptoed around the discipline of objective knowledge. How can system administration be kept within the boundaries of science and protected from the specter of personal opinion? The answer was straightforward: the scientific method, i.e., a mixture of inspired inquiry and exacting self-criticism. To round up the series, I would like to present a number of challenges to the system administration community, challenges that indicate a few of the important problem areas we have to examine in the coming years. This article will be of a more speculative nature than the previous pieces, with the aim of stimulating activity in the field; the list will not be complete, but you can fill in the blanks yourself. Speculation provides the creative impetus for a rational inquiry. System administration touches on a wide realm of topics; it is loaded with technical and social issues; it requires creativity and analysis; and we know very little about the empirical (verifiable, factual) basis for the subject. In short, it is a jungle waiting to be explored.

Theoretical

Perhaps the greatest challenge for system administration is the development of theoretical models. For some, "theory" is a dirty word, signifying the ivory towers of irrelevancy — charmless academics who have never gotten their hands dirty, preaching to the workers from their soapboxes. But this popular view is unfair. No field of study exists without a theoretical framework, a frame of reference — and you can be sure that no one gets into system administration without getting their hands a little dirty. The challenge of theory is to come up with ideas that can be tested by the kind of rational inquiry I have been discussing in this series. If one can build models, they can be tested by simulations or by experiment. Indeed, without a framework for inquiry, experiments are often meaningless.

As I explained in Part 2, there is no sense in collecting data and trying to guess what they might mean. You have to start with a question and then see what the data tell you about the question. Analyzing data blind is rather like Schiaparelli looking at Mars through a blurry telescope and seeing *canali*, which became even more blurrily interpreted as The Canals of Mars. It is easy to see things that are not there, if you don't have some theory to interpret what you see. I have repeatedly emphasized that theoretical work for system administration is about building models that link cause and effect. In some cases that is easy, in others it is complex. Computers are *dynamic systems*. One of the first tasks then is to develop models of competition for resources.[1]

Take, for example, the immunity model that I presented at LISA 1998.[2] This is based on the idea that a computer system has an ideal state. Recently I showed how such a state can be defined from a given system policy by mapping the average condition of a computer system onto a "state space," or lattice of possible states, where the origin was defined to be the ideal.[3] As faults build up in the system, the actual state gets farther from the ideal, moving through the lattice. This is now a precise model, not just words.

There is more to do. I did not show that the definition was the only possible definition. In order to detect anomalous or harmful behavior, we need to know how the system migrates through the space of states. Can certain regions of state space be classified into OK and dangerous? Is there a continuous blend, or are there rigid boundaries? The answers to these questions would be of enormous importance to anomaly detection, and thus also security. Is threshold behavior the answer, or something more subtle? Can immunological models be based on co-stimulation: a kind of lock-and-key signaling that switches on immune responses in the body only for a very specific signature (two or more complementary keys are required for confirmation)?

Empirical

Experiment and theory are symbiotes, feeding off each other in a constant dialog. Just looking for trends in data can lead to canali. To test theories one needs to collect data and see how well those data support or contradict the theories. The data feed back into models and refine them. An important issue here is scale — finding the right scale for the right problem. As in the examples from Part 2 in this series, the first step in analyzing a problem is to determine the time-scale at which it occurs. Problems therefore need to be classified by scale. There is no sense in measuring variations once a week if you are trying to see how users behave, since the time-scale relevant to users is minutes, not weeks.

It sometimes happens that problems arise interpreting data. If the measured data are troublesome, measure something different. For instance, many researchers have measured process lifetimes, or the lifetimes of network events. These fall into divergent distributions, since some processes or events hang or never die. This causes a problem. Solution? Try looking at a different variable, for example, the number of events, a quantity that cannot easily diverge.

Experimentation is already tied to many issues such as anomaly detection or time-series analysis. Use of neural networks is common, but neural networks are not well understood. Try to keep within the bounds of what you understand. Studies of reliability, effectiveness of certain policies, determination of what the relevant quantities are to measure in specific instances, dependency in causal webs: the possibilities are almost endless.

Psychological

The behavior of users is central to the behavior of computer systems. Users are as much a part of the system as are the programs they manipulate. Can user behavior be understood in any convenient way? There is almost certainly a relevant literature on this subject in connection with user-interface design and other issues. It needs to be traced and examined. It will not be completely relevant to understanding how users interact in network communities, but it will pave the way and inspire new work.

What shall we measure? Can user behavior be classified into groupings with particular characteristics that are useful when defining system policy? Psychological behavior can be useful in several circumstances, such as predicting user behavior in a crisis or deciding on adaptive security measures. Do users fit profiles? Is it possible, and, if so, useful to put users into categories? Should the categories be different for different system policy models? What are the ethical issues here? Can profiles be kept and stored in a form that is meaningful only to the computer system and which cannot be examined or used by humans for scurrilous purposes?

Conflicts of interest develop in any community where limited resources have to be shared by several players. How do such conflicts develop? What are ways of extinguishing them before they escalate? What is the effect of trust and privilege between users and the system? Do increased trust and privilege lead to fewer conflicts, or do they simply open the system to abuse? I do not know whether studies of this kind have been performed in other fields of research. Certainly the results have not been applied and made available in the field of system administration. Note how this final point is related to the issue of "security through obscurity" — if one attempts to conceal something, it is like stamping that item TOP SECRET in flashing neon lights.

Technological

Research into smart/safe algorithms that implement aspects of theoretical models will lead to an improvement in basic technology. For example, I have personally worked on the idea of "convergence to the ideal state," using cfengine as an embodiment of the immunity model. Load balancing or resource sharing can be implemented algorithmically. Network middleware already creates technological layers that help to standardize these issues for network services. Can similar middleware technologies (like cfengine) assist with the core problems of system administration?

Many problems in system administration would be solved more readily if there were standard log-file formats or mechanisms that allowed programs to share log data without having to reparse the log files. Today, programs that want to know about the log output of other programs (e.g., intrusion-detection programs) have to parse each other's log files. This is a waste of effort, made necessary by inadequate infrastructure. A signaling nexus for trusted processes to share their state information would assist in the task of anomaly detection and in dealing with security issues.

Changes in operating-system technology can have a huge effect on the ease with which system administration is conducted. Often system administration is a workaround for operating-system problems: fixing symptoms rather than cause. Operating-system technology at present pays little attention to the issues of system administration. If anything, UNIX-like operating systems are taking steps backwards in trying to emulate Windows (as if that were the standard for operating-system design), giving users privileges they should not have, in order to make things "easy." The careful analysis of operating-system technology in relation to system administration is long overdue.

How will encryption affect the issues of system administration? If system processes cannot detect threatening programs or data files because they are encrypted, they cannot protect against them. For instance, if system policy dictates that huge MP3 files should be deleted after 24 hours, but a user conceals these by encryption, the policy cannot be policed. This is reminiscent of the attitude of governments toward wiretapping. It is not as straightforward an ethical or technical issue as some would like it to be. Rather, it represents a conflict of interests.

Load Analysis

This is the area where the most work has already been done. Internet traffic, process lifetimes, capacity planning. High-performance and realtime computing issues for specialized problems. If you are going to analyze load, ask yourself why. What are you going to do with the data? What are you trying to show?

Databases

Bioinformatics is the new discipline in biology related to the creation and interpretation of databases of genetic and protein data. This a crucial sharing of important empirical data about the mechanisms of developmental biology, virus signatures, and disease markers. Creating and maintaining important public databases is equally important in the computing community (e.g., DNS, RBL, virus signatures). What new databases do we need? How will such databases be managed? What about trust and security issues relating to the database contents?

The very first database I would like to see created is a database of references to system-administration research. Without such a library of what has been done, we will be doomed to repeat earlier work. Such a database would be a valuable resource to the field. At Oslo, we have plans to set up such a database in BibTeX format.

The Shape of Windows to Come?

The day-to-day tasks of system administration change constantly, and we pay these changes little attention. However, improvements in technology always lead to changing work practices, as humans are replaced by machinery in those jobs that are menial and repetitive. The core principles of system administration will remain the same, but the job description of the system manager will be rather different. In many ways, the day-to-day business of system administration consists of just a few recipes that slowly evolve over time.

It is difficult to articulate just why the administration of computer communities is an exciting challenge, but if we are to succeed in pushing through programs of research that will bring about the level of automation we require, then it will be necessary to attract willing researchers. Fortunately, today a high proportion of system administrators have scientific backgrounds and the will and training to undertake such work. However, only the surface has been scratched. The tendency has been to produce tools rather than to investigate concepts. While the tools are necessary, they must not become an end in themselves. A clearer understanding of the problems we face, looking forward, will be achieved only with more analytical work.

In many ways system administration is like biology. Animals are machines, just billions of times more complex than our own creations, but the gap is closing and will continue to close as we enter into an era of quantum and biological computing techniques. The essence of experimental observation and of the complex phenomena and interrelationships between hosts is directly analogous to what one does in biology. We may have created computers, but that does not mean that we understand them implicitly. In our field, we are still watching the animals do their thing, trying to learn.

ACKNOWLEDGMENT

Parts of this article have been adapted from the author's new book, *Principles of Network and System Administration* (J. Wiley & Sons), by kind permission of the publishers.

REFERENCES

To keep this list short, I have given only a few pointers. For a full list of references, please look to the sources below. See especially<*http://www.iu.hioslo.no/SystemAdmin/scisa.html*>. At this location you will find many references and pointers. I would like to build a full list of key references to central issues. You can contribute to this database by sending references to work you feel to be important.

1. N. Glance, T. Hogg, and B.A. Huberman, "Computational ecosystems in a changing environment." *International Journal of Modern Physics*, C2:735, 1991. M. Burgess, "On the theory of system administration," submitted to*Journal of the ACM*.

2. M. Burgess, "Computer Immunology," LISA 1998.

3. M. Burgess, *Principles of Network and System Administration* (Chichester: J. Wiley & Sons, 2000).