Systems Administration: The Mathematician's Perspective

Silvio Levy

Virtually all American mathematics departments have computer systems and cannot live without them. Running these systems is often a problem for lack of money, expertise, or vision. I argue here that we need to become better educated about computer administration, savvier in getting financial support for the task, and more competitive in acquiring and retaining good staff.

Workshops that focus on systems administration can be a powerful tool in achieving these goals. The Geometry Center has held the first such workshops for the mathematical community; I discuss their contents and some of their consequences.

Nowadays a large majority of mathematicians in the United States have access to computers at their institutions. But when the typical mathematician has occasion to think about the *process* by which computers and software are made to run, it is usually because something is not working. Arguably, this is as it should be: most people want to get on with their lives and to think as little as possible about tools. They do not often fuss over the innards of their car and telephone either.

Yet the analogy breaks down in this respect: cars and telephones have had a century and more to mature, and the pace of their evolution has been far more sedate. Today's computers and software are tremendously complicated creations, expected to perform an astounding variety of tasks at a rate utterly inconceivable a few decades ago. If I risk slipping into banality here, it is because this fact, easy to lose sight of, is crucial to an understanding of why computers will not, for a long time to come, be as "transparent" as some of our other technologies.

Computers demand attention, nurturing, and periodic renovation if they are to be as useful as they can be, especially because so much of their usefulness depends on their talking with one another quickly and reliably. In short, they demand competent administration. A lot of mathematics departments are understaffed in this respect; in many cases everything is in the hands of faculty members or graduate students.

The Need for Stable Systems Administration

There is evidence that leaving systems administration in the hands of faculty members or graduate students is a false savings and a misallocation of resources. It leads to frustration, waste of time, and fragmentation that outweigh the cost of adequate systems administration staff. My own experience and comments from many sources—both systems administrators and end-users at a variety of institutions—have persuaded me of this point.

At a professionally run site, the administrator performs many time-consuming and not always obvious tasks: keeping abreast of and evaluating new technology, writing purchasing specs and shopping around, advising faculty on the best use of grant money, installing new software and hardware, running the local network or interfacing with central campus networking, coming in on weekends to handle emergencies or catch up with work, helping users make their programs run. This is as true for a Macintosh or PC site as for a Unix network, despite claims that personal computers require little or no administration. (It is true, however, that a homogeneous site requires less maintenance; the drawback is that users have fewer choices.)

Silvio Levy is a writer, editor, translator, and programmer in the mathematical and computer sciences. He is the editor of the MSRI Book Series and of the research quarterly Experimental Mathematics. *His e-mail address is* levy@math.berkeley.edu.

Faculty are no cheaper than systems administrators and do a poorer job. It is unreasonable to expect a faculty member to devote a lot of time to build up and keep a good environment for the whole department, and with very few exceptions they do not do it. To the extent that they do, they are employing their time suboptimally, even from the point of view of the university, which logically should prefer them to do what they are best at and what they are paid for—research and teaching. Certainly, having a committee made up of computer-literate faculty to direct and oversee the computer administration process is a good thing: as end-users, faculty should retain control of the big picture. But beyond that, it is a waste of time.

Students have a high turnover rate. A computer environment administrated primarily by students will have no continuity over a period of several years. Moreover, students often lack a certain healthy conservatism: they are eager to experiment with new technologies, install new versions of software, and so on (which is good), but often stop supporting or delete too fast the old versions to which many users are wedded (which is bad). Because they pick up new things fast, they assume that all other users will too. Career systems administrators, whether or not they started as students, learn after a few years to be more tolerant; but if a site has new students in charge all the time, this never happens.

Neither faculty nor students are likely to maintain the level of commitment necessary to avoid annoying disruptions. During my Princeton years, there were two times when many computer users lost several days' worth of work: in each case there was a disk crash and it was then discovered that no backups had been made for a week, even though they were supposed to be done every day. On the first occasion the backups were the responsibility of three students; on the second, we had just hired an inexperienced systems administrator. Nightly backups are much more likely to be done reliably by someone whose performance evaluation depends on it than by students or faculty. The same goes for the many other boring, mechanical tasks that are necessary to maintain a computer system.

Without a central vision, the system is at the mercy of short-term constraints. It is important that the system should survive without strain personnel changes and temporary budget cuts, that it should expand without disruption as the number of users grow, and that it should make it easy for users to gain a variety of competencies and enlarge their horizons. This is not easily achieved by a committee, and even dedicated and knowledgeable faculty members may not succeed in doing it unless one or two of them choose to devote most of their time to the task for a period of years. The result of failure is a rocky system and frequent frustrations and/or a pathetically low level of expectations. By contrast, a good, experienced systems administrator will set up a model that will serve users well long after his/her departure.

Why Workshops on Systems Administration

Most departments are more or less aware of the arguments just given; and if they still treat systems administration as a haphazard issue and leave it in the hands of faculty and students, it is sometimes not through their own choice, but because they cannot get the necessary resources to do otherwise. Good professionals are a scarce resource and command salaries that are high for academia, so not every department will be able to afford one. But more departments *would* be able to afford one if they pursued this as a clear and consistent goal.

Therefore the mathematical community as a whole needs to

become better informed about computer administration issues,

find ways to secure greater financial support for systems administration staff, and

whether or not professional staff can be hired, make it easier for systems administrators to grow in their jobs and keep abreast of current developments.

Creating opportunities for systems administrators (professional or casual) and interested faculty to come together can go a long way in helping address these problems. There is an annual conference (LISA) for UNIX systems administrators and similar conferences for PCs and Macs, not to mention trade shows. If systems administrators can go to these events, so much the better, but it is unlikely that the administrator at a department that has all three types of computers would want to attend three such conferences a year or that the department would pay the often steep registration fees. There are also subjects of specific interest to mathematicians that are addressed tangentially, if at all, at these conferences: symbolic manipulation systems, the use of computers in teaching, and so on. Chances that systems administrators will go to the specialized conferences on those topics are infinitesimal.

Given this background, a workshop focusing on the systems administration of mathematics sites, especially one with no registration fee, would seem like a winner. This has indeed proved to be so. Prompted by a suggestion made by Al Thaler of the NSF, the Geometry Center held in December 1994 the first such workshop, called SAMP, for "Systems Administration: The Mathematician's Perspective". SAMP brought together

about sixty academics and career systems administrators. In addition to general issues, such as user education, software updates, security, networking, and funding, there were talks and discussions on subjects of particular interest to mathematicians, including the role of computers in math instruction and communication, preparation of mathematical documents, computer algebra systems and other mathematical software, and MathSciNet.

SAMP was very well received (see a later section on evaluation), so the Center hosted SAMP2 in December 1995. Among the new topics discussed were technical advances such as better security, tools for electronic documents, and the explosive use of the Java programming language for interactive Web applications. I will return to them in the next section.

Not all lessons learned were technical. It was particularly useful for people from big and small universities to mix. Systems administrators at small universities are often the most resourceful and have a lot of good ideas to share. As an example, David Marshall, a graduate student at Humboldt State University in California, gave a talk on how he was able to procure funding to upgrade and maintain a lab intended for undergraduate mathematics instruction by allowing it to be used as a university-wide resource at times when it would be otherwise idle. Big and small institutions alike have much to learn in the matter of cooperation.

There was ample discussion of how math departments can obtain funds for staffing. If lobbying the university administration using the arguments enumerated in the preceding section is insufficient (and indeed no amount of rational argumentation will change the minds of some university officials), a department can use a variety of tactics such as levying a percentage of faculty grants; applying to outside sources for funding, especially for educational labs; and so on.

I regret to say, however, that no magic bullet was found that will work in every situation. Even the Geometry Center has been without an official systems administrator since the old one quit for another job, eight months ago as I am writing this; for various reasons the search for a replacement could not be initiated until recently. Fortunately, because of the Center's character, other staff members know the system very well and have not allowed it to deteriorate.

Why All the Excitement?

Here is a nontechnical overview of some of the technical topics that attracted most interest at SAMP2. I include Web addresses for readers who are interested in details; most of these addresses are collected as links at the Web page `http://www.geom.umn.edu/locate/workshop/SAMP2/`.

Java is a computer language specifically designed for networking whose introduction greatly increased the power of the World Wide Web. Ages ago (around 1992) the Web was an essentially static tool: you would see static text, static figures, and static links to other pages. Later came fill-up forms and other methods designed to allow greater interactivity; the Geometry Center, thanks largely to postdoc Paul Burchard (now at Utah), was a pioneer in interfaces to mathematical programs. Users could now send input to a remote program and obtain results over the Web. However, installing such interfaces was complicated and cumbersome; it was not something that a casual writer of Web documents would attempt. Moreover, a program running on the server can deal with only so many requests at a time. Java has changed the situation dramatically for the better by making it easy to write programs that are *downloaded on demand by the browser and run on the local machine*. The Web user can now solve a differential equation, manipulate a three-dimensional scene, run a spreadsheet, play a game, or get continuously updated data, provided she is running a browser (such as the current version of Netscape) that understands Java. A good Java introduction and tutorial can be found at `http://www.cs.ust.hk/~cheungkh/java/javatutorial.html`.

Electronic mathematical documents and WebEQ. WebEQ is a Java-based system that allows Web authors to include mathematics in their pages; it is in advanced stages of development at the Geometry Center by a team led by Robert Miner. If you have attempted to translate a TEX document into HTML, you know that doing any but the simplest formulas is essentially impossible in pure HTML. Automatic translation (using LATEX to HTML) is possible, but each formula is replaced by a bitmap, which looks different from the surrounding text, is completely unstructured, and takes much longer than text to download. In the 3.0 specification of the HTML language there is provision for mathematics and scientific notation, but it does not appear that browser manufacturers are in any rush to implement this new standard; obviously the market for sound and fancy backgrounds is much larger, so that is where all the commercial effort is being put. WebEQ is an attempt to implement the HTML 3.0 specification for mathematics notation and is an important step toward making the Web more valuable for scientific research. For details, see `http://www.geom.umn.edu/locate/WebEQ/`.

Network Security is a hot subject and was extensively debated at both workshops. The dis-

cussions and recommendations were much too technical to detail here, but interested readers may profit from looking at the URL `http://www.math.ubc.ca/~djun/samp2/security.html`, which contains the writeup of a talk and discussion led by Djun Kim (University of British Columbia).

ISDN. With a great increase in the number of people who have computers at home and expect to access their work accounts and the Web from there, the traditional modem-dialing setup has become inadequate. Paulo Ney de Souza of UC Berkeley gave a talk on ISDN (Integrated Services Digital Network), an international telephone standard now available in many areas. Because it uses digital rather than analog signals, ISDN can carry much more information (in the form of voice, data, or video) than the old system over the same existing copper wiring. ISDN is fast enough to allow the remote site to become a fully integrated part of the university network, mounting file systems, accessing software and other resources transparently, and so on. See `http://math.berkeley.edu/~desouza/isdn-crash.html`.

Why the Geometry Center?

The Geometry Center was uniquely placed to hold these workshops, being a leading institution in the use of computers in mathematics and in the communication of mathematics. The Center has an excellent computing and visualization lab with a variety of equipment, including SGI graphics workstations, other UNIX workstations, PCs, Macintoshes, and video production facilities. The Center's resources are used both locally and across the Internet for research, development, and education. The lab serves a diverse group of students, high school teachers, professors, and the Center's own staff; it is used regularly as the site for workshops with national and international participation. Center staff gave several talks at the workshops, sharing its experience in setting up and maintaining this environment.

There certainly exist other places where a similar mix of hardware and people are present. I believe that the reason a workshop like SAMP has not taken place elsewhere, before or since, is not that other sites have not understood the potential usefulness of such a meeting, but rather that their *mission* is different from the Geometry Center's. Cornell and Utah are examples of universities where mathematicians have a good computing environment, but their goal is research and education in math proper; I doubt that they will become the site of the next SAMP workshop. (Of course, I would love to be proved wrong.) By contrast, the mission of the Geometry Center is in part to promote computational tools for mathematicians and to facilitate communication among mathematicians and between mathematicians and the public at large.

Composition, Evaluation, and Consequences

The composition of participants at SAMP and SAMP2 is probably close to being a representative cross-section of bigger four-year colleges and universities in the U.S., with an admixture of other sites like the MSRI and the AMS. Among the participants of SAMP were two from Britain, two from Canada, and one from Israel. Women and minorities were particularly encouraged to apply; women were relatively well represented. About half the participants received total or partial support from the Geometry Center. Others had support from their home institution.

An evaluation sheet was passed out after each workshop. Over 95 percent of respondents chose a 5 or a 4 (out of 5) for an overall rating, and many praised particularly the helpfulness of the administrative and technical staff. The most common criticism of SAMP (apart from the cold, which was even worse the second year) was that there was too little time for informal interaction; this problem was addressed in SAMP2 by having fewer lectures each day and more time for discussion groups and informal conversation. Discussions were usually quite lively.

Some comments and criticisms from the SAMP2 evaluation forms:

> Because of the huge increase in use of the Web, this year the conference focused much more on cross platform issues. This made the conference this year much more valuable to me. Almost all of the talks this year contained aspects of interest to me. Last year as a Mac user in a largely Unix-oriented group, I had little to relate to in perhaps 30–40% of the talks. It's amazing how much the Web has changed things!

> I now see Java as more important [than before], and plan to update Web pages with info learned here.

> There were a lot of talks which were interesting but many didn't relate to my field. It was very slanted towards the educational arena, so getting more vendors would be a good idea.

> Get a few people from industry to give an alternate perspective to the academic environment.

> Great conference! I hope to be back next year!

What lasting benefits did the workshops bring? First of all, participants found them useful as opportunities to get acquainted with many topics that they did not know or knew only slightly. Almost everyone had heard about Java, for example, but few mathematicians and mathsite administrators had done anything with it, so the tutorials and talks on the language were very well attended. In the months after the workshop some of these participants started using Java to provide services at their sites.

Another example: Someone mentioned in passing during SAMP2 an indexing and query program called Glimpse, through which users can search through a large collection of files very quickly; it allows approximate matching of misspelled words, Boolean queries, and even some regular expressions. I find it invaluable. In spite of being free and requiring little setup time (though it does require disk space for the index files), Glimpse was familiar to less than a third of participants. After the workshop I set up a Web page of pointers to interesting software, including Glimpse; I've had reports of administrators picking it up and recommending it enthusiastically to their users.

I will mention two more ways in which SAMP and SAMP2 left their mark. Late in 1994 the mathsys mailing list had been established at the MSRI, with the purpose of networking together the same people for whom SAMP was designed. As a consequence of SAMP, its membership increased fivefold. It continues to increase, and currently stands at about 120. (To subscribe, send e-mail to `majordomo@msri.org` with the subject "subscribe mathsys".) The mailing list is used for consultations, job postings, and so on. It was extremely useful on the occasion of a major security break not long after the first workshop: the more knowledgeable subscribers were able to explain to the others what action to take, direct them to sites where they could obtain patches for faulty programs, etc.

The Survey of Math Computing Environments

The second initiative originated with Joe Stone, then of the School of Mathematics, University of Minnesota. It is a survey of the computer environment—demands and resources—of participating institutions. It has now been extended from the original audience of SAMP attendees to other members of the mathsys mailing list. The results are available on the Web (see last paragraph). Here I mention only two questions.

> "How would you classify the hardware resources at your site?" Choices "somewhat insufficient" and "adequate" got almost 50% each; the other two choices, "grossly insufficient" and "more than adequate", got a sprinkling. So the situation regarding hardware seems to be fair to middling.

By contrast, the question

> "How would you classify the human resources, as far as systems administrations is concerned?" got 50% "somewhat insufficient", another 30% "grossly insufficient", 20% "adequate", and 0% "more than adequate". A related observation is that about half of the respondents report a greater de facto workload than expected from their job descriptions.

Conclusion

Computers are here to stay, and running them will remain a complicated affair for the foreseeable future. The mathematical community can only profit by learning more about the issues involved and by supporting its systems administrators as much as possible.

Workshops such as SAMP can be a powerful agent in helping the mathematical community make informed decisions, in spreading familiarity with new computer tools, and in helping systems administrators do their job better. We must bring together, on a regular basis, the people involved with the administration of our systems—be they career systems administrators or academics—to teach each other more about a subject that is essentially unbounded. The ultimate goal, stressed by the title of the Geometry Center workshops (and of this article) is to help fulfill the needs of the *mathematicians* who *use* the computers.

More information about SAMP and SAMP2 is available on the Web at `http://www.geom.umn.edu/locate/workshop/SAMP2/`. It includes schedules of talks and discussions, many of the talks presented, comments from the evaluation sheets, pointers to interesting software and Web addresses, and more results from the surveys. If you are interested in attending a SAMP workshop in the future or think your systems administrator may be interested, send e-mail to `orgsamp@geom.umn.edu`.

Acknowledgments

I would like to thank the editors, Hugo Rossi and Steve Krantz, whose suggestions greatly improved this article, and Geometry Center director Dick McGehee, who provided the initial impetus for the writing of it.