

Moving from a Mainframe to a Server

Roger Konyndyk, Steelcase Inc.

Abstract

Because of the low cost and increasing power of workstations, many companies are turning to them for their computing needs instead of mainframes, where performance gains have been much slower. The decision whether to move from a mainframe to a server, however, must be based on much more than just the relative cost of computing on the two machines. Many time-consuming technical difficulties must be dealt with, people must learn to function in a new computer environment, and the new system must be able to share data with the mainframe. Much effort and expertise is required to set up a new system on a server. If the hurdles can be overcome, moving to server will provide users with much more control over their computing, faster response time, faster job turnaround, and less bureaucracy.

Background

The Steelcase Business Research department is responsible for the the company's sales reporting. With corporate downsizing, traditional data processing support for our department has gradually been withdrawn. To make up for the lack of programming support, our department has learned to use SAS®. Because of the many requests for data which our department receives, and because the primary file for our reporting has about 5.3 million records, our department was identified as the largest single consumer of mainframe resources, sometimes averaging 5 - 10% of our IBM 3090 during prime time. With expensive mainframe resources running out, and the price of workstations dropping as their power rose, moving our department to a separate processor looked more and more attractive to our Computer Center people. Comparing the cost of computing on the mainframe to a server running UNIX made the decision to move to a server look like a "no-brainer". In addition to cutting computing costs, we hoped to get better job turnaround and control over our computing environment.

Our Experience

In reality, the move was far more difficult than anyone in our department or in the Computer Center had expected. Numerous technical problems of all kinds suddenly appeared. We did not sign on to the server directly, but used networked PC's functioning as X-windows terminals. It would take a half-minute or more to

repaint the screen when switching between windows. The software we first used to access SAS on the server would often "hang" and force us to reboot. SAS bugs caused segmentation errors, which we had never seen on a mainframe. Our printer, an old Apple Laserwriter, would use only three quarters of the width of the paper when we tried to print 132 columns in landscape mode. A year later, we were still uploading reports to the mainframe in order to get print quality acceptable to an end user. We found out the hard way that UNIX is not exactly the ultimate in user-friendliness. It ranks well below native DOS in that regard. The commands are not obvious (`lpr` for "print", and `mv` for "rename", for example), responses are terse, and error messages are often non-existent. On the mainframe, we had a full-screen interface for basic tasks like printing and editing files; we know of no corresponding package for UNIX. Compared to the mainframe, UNIX seems quite unsophisticated. Data security is much more crude on our server, and we had some problems with data security. We had to reset defaults so that someone besides a file's "owner" could edit it. We also found that nothing prevented two people from editing the same SAS dataset simultaneously - without SAS/SHARE®. It seemed attractive, until it caused some major data losses. A call to the SAS Institute gave us the option setting to prevent that from happening again. People sometimes turned off their PC's without logging off SAS on the server, leaving their SAS session running there indefinitely. We set up a batch job to run every midnight to delete such things. Then we had problems with it deleting jobs that we wanted to run. Just figuring out which jobs are running on the system is a chore, and figuring out which SAS job is using which work library is no mean feat. On the mainframe, when a batch job runs out of disk space, it will abend. In UNIX, it will run indefinitely, hoping somehow to get more space. More than once we have had to kill "runaway jobs", and then delete their SAS work libraries as a separate step. Similarly, we have run out of space doing downloads, deleted the offending file, and still found no space freed up! We would still have to find the process I.D. of the downloading process on the server and kill that. When the download is initiated by a mainframe batch job, that is not obvious.

Switching to a new operating system meant a considerable amount of learning for everyone, and not everyone was enthusiastic about that. Especially because of the lack of a full-screen interface to UNIX, some people found little to like about UNIX, and continue to avoid it.

With no data on the server initially, we were not doing backups, either. Fortunately, we soon did nightly backups. We have found it distressingly easy to delete entire directories by accident; only the backups prevented disaster. We even had a few backups fail, for technical reasons not fully understood.

Our 5 million record sales history file is actually a summary file, condensed from a much larger (sequential!) file affectionately known as "Bigfile." At the time we began working on our move to a server, our department rarely used that file. However, with a more competitive environment, the company's need for information increased sharply, and requests for information requiring details found only in Bigfile skyrocketed. All of those requests are handled on the mainframe. Our department now must have expertise in both the mainframe and the server. We have even had to port some of our data developed on the server up to the mainframe to handle the Bigfile requests that could only be processed there. In the end, we have not been able to move to another platform as much as splitting our group into UNIX users and mainframe users.

Uploads and downloads became a major issue. At the outset, no one was aware that downloads consume many system resources. We found out by downloading our almost 2 gigabyte sales history file during prime time. Now downloads of that size are only done at night.

Our Computer Center is setting two directions for computing. One direction is distributed processing, which our department exemplifies. The other is centralized databases on a mainframe. This means that connectivity issues could become very interesting in the future. If the consolidation of data actually does take place, it remains to be seen how we do the processing on the server of large files residing on the mainframe.

Beside the technical issues, organizational issues also made the move to a server more difficult. Like many companies, we have recently gone through reorganization and downsizing. That meant that our old reports were obsolete. Instead of porting existing SAS code and datasets down from the mainframe, our entire sales reporting system had to be redesigned and rewritten. New databases had to be created. Meanwhile the old system continued to run on the mainframe for a year, as a stopgap. Computer Center people were disappointed that they couldn't pull the plug on our mainframe jobs much earlier. Further, the ongoing tasks of the department had to be done, sometimes on the mainframe, sometimes on our server. The ongoing efforts meant time taken away from learning and using the new server, and great delays in rewriting old mainframe jobs on the server.

Overcoming the problems

One of the keys to our success was the ongoing availability of a dedicated, competent support person in our technical support group. Without such a person, our move to a server would have been a disaster.

The SAS macro language was a lifesaver for us. We wrote some macros that brought up windows for the user to fill in:

- to upload reports to the mainframe after checking that the report file exists on the server
- to submit batch SAS jobs at a time specified by the user, directing the log and reports wherever desired
- print report datasets at a non-local printer

In starting from scratch on our own machine, we had lots of decisions to make, such as organizing our data and programs. Over time, ~~programs tended to proliferate, and we were running many~~ programs out of our "hip pockets." To avoid being dependent on individuals knowing where the programs were, we had to get organized. Now each individual has a personal library for development work and one-time programs. We have separate libraries for production SAS programs, the command files which call those programs, and the logs and reports from those programs. We even mimicked our data processing people and developed naming standards. Thus we can tell which command file, program, job log, and report file go together. Every production report has a report I.D. number on it that ties it back to the program that produced it. Production jobs are all documented, too. The documentation is stored on-line in a common library where everyone in our department can get at it, using word processing software.

At our company, production reports produced on the mainframe are stored on-line. We needed to find a way to get our reports produced on the server uploaded to the mainframe, too. We now have a SAS macro that issues the upload command, and then calls another macro that writes mainframe Job Control Language (JCL) to a file on our server, then uploads that JCL file and submits it as a batch job. That batch job then takes the uploaded report and puts it into our on-line storage. It works very smoothly now, but no one had done it before in our company. Our data security people were not at all enthusiastic about the whole idea, especially since submitting the mainframe job required that a mainframe user I.D. and password be hardcoded on our server. Other people in the Computer Center were very eager to make this work, as they recognized that this was the only way to permanently shut down our production mainframe jobs.

Successes

In spite of all the difficulties, we have enjoyed successes, too. We spent some extra money to get faster disks, a move that has paid off handsomely, especially with our large files. We have solved most of the problems mentioned above, and response time and job turnaround is much better on our server than on our mainframe.

Our mainframe production reports ran for the last time in March, 1993. Moving to the server will save the company over a half million dollars per year in mainframe computing costs. In fact, we are now producing over 45 groups of sales reports on our server. A production mainframe job downloads our sales history file at night and starts the job on the server that formats it into a SAS file. Then the report jobs run. The last step of each production job on the server uploads the reports to the mainframe and initiates a production job to move them into our on-line storage.

We maintain many of our files on the server on-line; our Computer Center people would never allow such files to have production status.

Because we are functioning as a small group instead of as part of a large computer installation, we have been able to avoid a lot of bureaucracy. As a result, we have been able to get things done much faster than we could have on the mainframe.

Our mainframe is badly overloaded. Response time there is slow, and even small batch jobs will sometimes have a long wait before execution begins. On our server, response time is quick, and jobs run when we want them to.

Some lessons learned

- Most obvious: The decision to move to a server is hardly a "no-brainer" based solely on the cost of computation.
- Strong technical support will be needed for an extended period of time. A technically oriented person in the user area is needed as well. The increased (and ongoing) need for technical support should be calculated when looking at costs.
- The transition is a long and painful one, filled with unforeseeable technical problems.
- Expertise in both systems will be required of the user during the transition, and beyond. For example, we have production jobs on each system initiating production jobs on the other.
- Ongoing tasks and unforeseen projects will take precedence over the transition to the server, causing it to take longer than anyone expected.
- Downloading and uploading of data will become a major issue.
- Changes in a company's reporting needs mean that old systems cannot be ported down to the new server; new systems must be written to serve the new needs. This delays shutting down the old mainframe system as well as getting output from the new.
- Not everyone being asked to make the switch to a server will be happy about it. Learning to operate in a new environment is challenging, time-consuming, and painful, especially given the technical difficulties that may be encountered. For most

people, it's easier and more comfortable to do things the old way, unless you can show some dramatic improvements.

- In setting up jobs and files on a server, technical expertise is needed. We have, in effect, become our own data processing shop.
 - Jobs must be documented, and data must be backed up.
 - For all the difficulties, the benefits are real. We have shut down some very large mainframe jobs, moved processing to a cheaper platform, improved job turnaround, gotten faster response time, and gotten control over our computing.
-

SAS and SAS/SHARE are registered trademarks of SAS Institute Inc. in the USA and other countries.

® denotes USA registration.

Other brand and product names are registered trademarks of their respective companies.

Contact Information:

Roger Konyndyk

Steelcase Inc.

CD-5W-08

901 - 44th St. SE

Grand Rapids, MI 49508

Telephone: (616) 698-4518

FAX: (616) 698-4502