

“How to save \$30,000 in 4 Hours”

Migrating SAS[®] systems from the mainframe to the PC

Iain Robertson, Systems Seminar Consultants, Madison, WI

Abstract

Many companies are looking to use the power of PC's to off-load work from their mainframes. The SAS system is ideally suited to this due to the portability of its code and data and the client/server capabilities of SAS/Connect[®].

This paper will discuss how an MVS SAS batch system was converted to run on an OS/2[®] based LAN:

- Over 1.6 GB of SAS data and code were transferred to a server on the LAN.
- The SAS code was altered to run on the OS/2 platform.
- Several thousand pages of final reports were transferred back to the mainframe for printing.
- An MVS job was written that would automatically update the LAN SAS data sets from the DB2 source data.

The new suite runs overnight when the LAN and network are under-utilized. The longest running job takes 4 hours to complete when run on a single PC (486 16MB).

The resulting system saves over \$30,000 in mainframe CPU charges per year; the analysts are more productive and prefer working interactively with SAS under OS/2.

Background

Since 1987 the Wisconsin Dept. of Transportation, Traffic Accident Section (TAS) has operated a computerized Traffic Safety Record-keeping System (TSRS). TAS produces standard reports each quarter

for a variety of state agencies. Ad hoc and specialized reports are also produced for a diverse audience (state agencies, attorneys, Media and general public.)

The TSRS is maintained within an MVS DB2[®] database which is populated from accident reports filed by law enforcement officers and individuals. The current year's data is refreshed by a monthly extract to SAS from the DB2 tables. The data changes infrequently and is basically static after 6-9 months. The SAS data sets are then used for analysis and reporting. Four TAS analysts are employed to maintain the data and provide the analysis and reporting functions. All their work was performed on the mainframe in batch.

Since 1987, the number of reports requested and their associated costs had steadily increased. A more cost-effective method for processing the accident data was required.

In 1995 a project was initiated to assess the possibility of moving the analysis and reporting functions off the MVS mainframe - to save CPU costs and utilize the power of the OS/2 and the LAN.

Feasibility

A study was first performed to assess the technical feasibility of the project. From this, projected costs savings were calculated to decide whether to proceed with the migration.

The technical considerations stalled with the transfer of the data between the mainframe and the LAN. At that point in time data transfers were conducted using the EHLLAPI terminal access method which is slow, unreliable and difficult to automate. A

more robust method had to be found before the project could be seen to be feasible.

The solution was TCP/IP which had started to make its presence felt throughout DOT during 1994.

SAS/Connect transfers using the TCP access method were found to be 5-10 times faster, were reliable and, because TCP is a peer-to-peer protocol, transfers could be initiated from an OS/2 or MVS batch process! TCP/IP also had the benefit of enabling FTP to be used to transfer flat files rather than starting SAS and using the UP/DOWNLOAD procedures.

In June of 1995 the decision was made to implement TSRS on the TAS LAN.

Objectives

The first stage in the project was to clearly identify objectives that were manageable and measurable:

- Move data from MVS to LAN
- Move source code from MVS to LAN
- Create SAS production environment on the LAN
- Train analysts to use SAS interactively under OS/2
- Train analysts to identify the code that would need to be changed under OS/2
- Performance tune the LAN data
- Convert the quarterly reporting jobs to run under OS/2
- Implement an automated process to update the LAN data from the DB2 extract

Initial Data Load

The first two objectives were achieved easily using a SAS/Connect MVS batch

process to download the existing data and code during off-peak hours.

This was one of the benefits of using a peer-to-peer protocol like TCP/IP - transfers can be initiated from either MVS or OS/2 and can be automated to run unattended, overnight.

1.6GB of accident data (years 88-95) and 30 MB of source code were transferred over a two day period.

LAN Production Environment

The DOT had already implemented a standard SAS environment across their LANs. The SAS software simply had to be installed on the server and a common AUTOEXEC.SAS file customized to allocate the accident data libraries and source code directories. Each of the four analysts then had a consistent production environment to work from - which is what they were familiar with when working on the mainframe.

Training

Two or three informal training sessions lasting 2-3 hours were scheduled with the four analysts.

They were introduced to the interactive Display Manager of SAS under OS/2; on MVS they had always run SAS in batch and were not familiar with Display Manager on MVS.

They had no problem learning where their existing SAS programs would need to be altered to run on their PCs because 99% of their code ran unchanged.

The capabilities of SAS/Connect were also demonstrated. Analysts were shown how to transfer SAS data sets, SAS libraries and flat files. The concept of developing code locally on the PC and then uploading to the mainframe for execution was introduced.

No other training was required.

One of the most immediate benefits of the move to OS/2 was the increase in productivity of the four analysts. They were able to develop and run reports much faster than they had under MVS.

Performance Tuning

Discussions with the analysts and examination of their code showed that a large number of their reports were performed on data within a specific municipality or county in Wisconsin. It was decided to index the data sets on the municipality and county codes. The index data sets added a 6% storage overhead but never need to be re-created once the current year's data is complete and stable.

The existing code contained a large number of 'sub-setting IFs' on the index variables and was changed to use 'WHERE' processing. Indexing shortened runtimes by up to 60%.

The effect of compressing SAS data sets was also investigated and found to be beneficial. Reductions in the runtimes of the tests ranged from none to 30%. Storage reductions ranged from 10-30%. It was decided, therefore, to compress the larger data sets.

Quarterly Reports

The quarterly reports for Law Enforcement agencies accounted for approximately 40% of the CPU charges to TAS. These reports generate 7,000+ pages for 72 counties and 83 municipalities; they are run as two separate jobs known as the CNTYSET and CITYSET. The greater part of the project effort was to be directed at converting the CITYSET to run on the LAN and to then apply the same techniques to the other jobs run by TAS; specifically the CNTYSET and a semi-annual report of missing data.

The two main issues to be addressed for the conversion of CITYSET were:

- The original job failed completely when run under OS/2
- Printing the reports to the local LAN printer was impractical

Code conversion under OS/2

The original code from the MVS batch job was easily changed so that it would run (in theory) under OS/2. But, when it was run, the PC would hang after processing only one-tenth of the municipalities.

The code contained a loop to cycle through each of the 83 municipalities generating 18 reports for each.

It appeared that memory was not being de-allocated which, over the course of time, caused the OS/2 swap file to fill up. Eventually, the swap file consumed all available space on the hard drive and resulted in OS/2 halting.

A solution had to be found; otherwise there would be no point in proceeding with the project.

It was felt that alerting SAS Institute to the memory leaks would take too long to resolve and there was a feeling that the overall complexity of the job should be reduced - even under MVS the job had been split into 4 separate runs because its virtual storage limit had been exceeded.

The solution revolved around the need to free up the lost storage - this occurs when the SAS process ends and OS/2 'tidies' up after it. The plan was to start SAS for each municipality and just produce a single set of 18 reports and then close down SAS. This would be repeated for each of the 83 municipalities.

The analysts, obviously, were unwilling to sit and start SAS 83 times - besides, the entire process was to be run overnight which meant that it would have to be automated somehow.

The REXX programming language that is part of OS/2 was the ideal candidate to automate the initiation of the SAS process for each municipality.

The new design moved the loop that runs the set of reports for each municipality out of the SAS code and into a REXX program. The REXX program then started SAS 83 separate times for each municipality and passed in a municipality-code parameter to the SAS program.

After this change was made the job ran without hanging OS/2 and the swap file never grew beyond 30 MB.

Unfortunately, the job now had an estimated elapsed time of 26+ hours! (On average each set of reports took about 18 minutes to run.) Further analysis was required to see if the performance could be improved.

The original code was the concatenation of a variety of individual reports that had been amended over several years. Each report re-extracted the same data for every municipality; unnecessary variables from the source data sets were being kept; an entire report contained code that was no longer required and there was a lot of redundant code.

The CITYSET process was re-worked (most of the code was good and therefore a re-design was not needed.) The data for each municipality was extracted once, at the start of the job, and re-used by each report process; only the required variables were kept and redundant code was removed.

The resulting code was reduced in size by 50% and the runtime dropped from the estimated 26+ hours to 4 hours.

Printing

The resulting output from the CITYSET consisted of a separate PRINT file from each of the 83 invocations of SAS. It was impractical to expect the LAN printers to

handle this load and so another method had to be found to produce the printed reports.

A mainframe printer was the obvious choice but a method of transferring the reports to the host was required. The installation of TCP/IP for the SAS/Connect communications access method allowed SAS to use the File Transfer Protocol (FTP) to send the output files to the host. SAS/Connect could have been used but FTP is a more direct route when transferring text files.

A job was setup on the mainframe to connect to the PC that runs the CITYSET process and upload all 83 reports (3000+ pages) to an MVS partitioned data set. A second job was then run to send each report to the printer. The entire process takes about 30 minutes - 25+ minutes for the transfer and 1 minute for the print job.

The following issues have to be borne in mind when printing PC reports on the mainframe:

- The LINESIZE and PAGESIZE of the report should be set to those of the mainframe printer - carriage control information across platforms is different.
- The FORMCHAR SAS option should be set to characters that are printable on the mainframe.

LAN Update

The monthly extracts from the DB2 tables have to be applied to the LAN copies of the data sets. An MVS job was set up to use SAS/Connect to logon to a PC and transfer the data to the LAN. The ability to remotely submit SAS code across platforms allows the same MVS job to update the LAN data sets with the new data - obviating the need for the analysts to manually run an update process on the LAN.

This job is run monthly to refresh the last two month's data and is also run semi-annually to completely re-populate the LAN data sets.

Cost Savings

The following table summarizes the TAS costs on an annual basis. These costs do not include the DB2 extract which has not changed.

Suite	Old Cost	New cost
Law Enforcement	26,000	1,400
Missing Fields	2,100	10
Ad hoc	10,000	0
LAN Update	0	700
SAS License	0	2,000
Total	38,100	4,110

Conclusions

A year after the initial move from MVS to OS/2 the project has been judged a success. Costs incurred from mainframe usage have been dramatically reduced.

The analysts are much more productive working under OS/2 and would seldom consider developing code in batch again.

The standard reports have been converted to run efficiently under OS/2 and work is ongoing to move additional functionality from MVS to the OS/2 platform.

A solution to the volume printing requirements is in place and is reliable - this is a good example of when to use the power of the mainframe.

The initial thought that 99% of SAS code runs unchanged across platforms, whilst true in theory, is not always true in practice. It is important to know the capabilities of each platform that will be used and how best to utilize the resources of each.

One of the benefits of this project was the opportunity to review and performance tune the existing code. Many of the tuning efforts made under OS/2 could have been applied, just as readily, to the existing MVS code - with a corresponding cost saving.

Overall, it can be said that the SAS system is an excellent product when right-sizing legacy applications.

Acknowledgments

I would like to thank the following teams at the Wisconsin Department of Transportation for their input, help and support in making this project a success.

Program Services Unit
Development Facility Support Unit
Networking Services Unit

SAS and SAS/Connect are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. IBM, OS/2, DB2 are registered trademarks or trademarks of International Business Machines Corporation. ® indicates USA registration.

Iain Robertson
Systems Seminar Consultants
1220 Femrite Drive
Madison, WI 53716-3756

Voice: 608 222 7081
Fax: 608 222 7310
Email: train@sys-seminar.com