The Push to “Get Off of the Mainframe” and Move to Client/Server: Is the Software Industry Keeping Up With the Demand for Data Storage and Access?

Daryl R. Hoffman
Senior Applications Programmer/Analyst
The Pennsylvania State University

Introduction

In the computing industry today, the academic sector, as well as many of the largest corporations, are downsizing not only the staff, but also the expensive mainframe operating systems. The large IBM operating systems of CMS and MVS are slowly giving way to UNIX workstations and personal computers to run some of the programming jobs that had been reserved for mainframes or larger minicomputers in the past. So what is involved in moving an application or information system from the mainframe to a personal computer or workstation? Can the data still reside on the mainframe and be accessed from the remote system? Is client/server really the way of the future or is it already here? What about the security of the data and other issues dealing with data transfer and storage? What kind of network is necessary to run the new information system?

This paper will discuss some of the strategies used with an application that is currently under development for both the CMS and Windows 3.1 or Windows '95 platforms. It will also make an attempt to answer the question of what is necessary to move the data and applications from the mainframe to the PC. The presentation of the paper will demonstrate how with SAS' multi-vendor architecture (MVA), one can now easily port the code and the data to the PC. Also demonstrated will be how easy it is to move across platforms using SAS. The products used may involve SAS/CONNECT, SAS/SHARE, a Windows OS, and a network software product such as Novell.

The tools that SAS provides are more than enough to make the transition to client/server. Are you ready? Hold on as we take a voyage into the future, now, and make the move from mainframe to the PC or Unix workstation.

The Move to Distributed Processing

In the software industry, developers and programmers alike have been hearing about the word ‘downsizing’ from all sides. Businesses are attempting to cut costs by downsizing personnel. Universities and academic institutions are attempting to cut the costs as well by removing the need for large scale mainframe computers. The workstation market is beginning to attempt to fill the gap with faster processors, larger hard drives and other forms of data storage and access.

Distributed computing is allowing the end-user to have the workstation on their desktop and with the development of data warehouses and client/server technologies, the data is at their fingertips. The applications programmer is called on to provide a high end, easy to use, 'point and click' program to allow the executive, clerical or other user of the information system to easily retrieve the data or generate the report.

The Project

The Pennsylvania Commission on Sentencing has collected data on all of the sentences issued in the Commonwealth of Pennsylvania for a number of years. The Commission has decided to move from a DOS data entry program and utilize the power of SAS.

The initial development phase involved getting the application running on the mainframe under CMS, running VM/ESA 2.0. The application is being developed using SAS/AF and other Base SAS programs. Most of the code is being written with program entries and SCL because of the graphic problems that are involved when using SAS/AF Frame in the CMS environment. Once the code is developed and a working package is complete, some of the entries may be changed to Frame entries to attempt to move on to object oriented programming with SAS/AF and Frame.
First Steps Towards Distributed Applications

The SAS/AF application for the Commission on Sentencing is being created with the move to a workstation in the back of the developer's mind. Host system strengths and weaknesses are being built into the SAS/AF application as it is being developed on the CMS platform. Many of the drawbacks encountered on the CMS platform will also be fairly easy to overcome when the SAS/AF application is ported to the workstation.

Storing the SAS Data Sets

The data will currently reside on the mainframe. When the application is moved to the Windows platform, the decision will need to be made on whether to also move the data to the workstation or to use SAS/CONNECT and access and update the data using remote library services.

If the data is stored on the workstation, the necessary disk space will have to be purchased and the workstation will have to allow for the growth and addition of other data storage devices.

Data Warehouse

One of the main goals of the project being developed for the Commission is to create a data-driven application and to begin to create a data warehouse that will be used in the future to locate and create reports, develop new applications and provide the data for the validity checks on data entry. According to "Building Your Client/Server Solution with the SAS System" by SAS Institute, a data warehouse is necessary to provide executives with information retrieved by a desktop application or decision support system (DSS). The data needs to be stored in an easily accessible form and must also be in a location that can provide fast access to the data in the creation of ad hoc reports that may be required by the end-users.

The data warehouse for the PA Commission on Sentencing will be stored on a PC-based server running either Windows NT server some other server operating system that will allow users to share the data across platforms and also allow for concurrent access to the data for updates and retrieval.

Extended tables are used extensively to create selection lists that will allow the application to remain unchanged and only change as the data sets are changed and new observations are added or others are removed.

By creating the data warehouses for data access at a later time, the Commission on Sentencing will be able to provide easy access and continue to develop the current SAS/AF application and allow it to eventually evolve into a full DSS application requiring little or no end-user knowledge of the SAS system. As with most DSS systems this is a desired goal.

The concurrent access will be controlled by a SAS/AF application and SAS/SHARE. SAS/SHARE will take care of most of the database management, but the AF application will also have controls built in to try and prevent entire data sets from being locked, as well as only retrieving the data and variables necessary to create the report or to execute the application.

Workstations and Network Setup

The Pennsylvania Commission on Sentencing has chosen a server class machine to be used for the data storage. Once the data and application have been moved to the Windows platform. The chosen network software will currently be Novell. When the data is moved to the workstation, SAS/SHARE will be installed on the server to provide access for all of the clerical support, as well as the programmers and executives who will need to generate reports for others to use.

Experiences With SAS/SHARE and SAS/CONNECT

In the early days of SAS/SHARE, the Center for Academic Computing attempted to use SAS/SHARE to provide concurrent access to the SAS data sets for update and read-only retrieval. With little SAS experience the programming group found that SAS/SHARE seemed to create locks on the SAS data sets and to not allow for concurrent access. At the time, version 6 was still in beta and not release, so the group decided to move the data into a database system using SQL/DS a mainframe DBMS.
Hurdles Overcome with SAS/CONNECT with Version 6, Release 6.08

When attempting to set up a remote connection between the Windows platform and the Penn State mainframe running CMS, several problems were encountered and overcome. SAS/CONNECT uses telnet and WINSOCK to connect when the method of connection chosen is TCP/IP.

options COMAMID=TCP;

The problem that was discovered and may be unique to our site, was that SAS/CONNECT required a line-mode telnet port to be available on the mainframe to initiate the TCP socket open and the data transfers to take place. Our site has chosen to not allow any line-mode telnet sessions to take place. This became problem number one.

When SAS/CONNECT attempts to establish a session with one of the remote scripts provided with SAS software, it displays a port address, which has allowed SAS/CONNECT to be used at our site. One can simply capture the port number, make the change to the program and submit the remote SAS/CONNECT request from the local workstation. However, the goal is to make the connection between the workstation and the mainframe transparent to the user, so this created another small hurdle to overcome.

The circumvention for this problem was to attempt to capture the data displayed on the mainframe upon login in full-screen mode using a HLLAPI terminal emulation program. At Penn State one of the products we use that provided this was McGill University's TCP3270. The TCP3270 product allowed for HLLAPI screen captures using a Microsoft Visual Basic routine included with the license.

The Visual Basic program was developed to provide a way to initiate a logon to the mainframe, start SAS on the remote host (mainframe) and capture the port number given to SAS on the remote host. The port was then captured to a temporary file on the workstation and used to make a modification to the SAS remote connect program to be executed on the local workstation. The Visual Basic (VB) program was then finished and SAS was started on the local workstation with a sysin file used to initiate the connection.

Example of SAS Code for initiating SAS on the workstation

%let mynode=psuvm.psu.edu;
filename rlink 'd:\sas\stop.scr';
options set=tcpport xxxxx comamid=TCP remote=mynode;
signon;
libname mylib remote 'C' server=mynode;
proc fsedit
  data=mylib.shouser(where=(userid='DRH4'));
run;
signoff;

The stop.scr file only contains the following line:

   stop;

The host system will initiate SAS using the following SAS command:

SAS (DMR COMAMID=TCP)

The application developed using Microsoft's power-builder application language, Visual Basic, allowed our site to provide a way to use SAS/CONNECT without requiring our systems engineers to supply a line-mode telnet port. This solution is not the best and it would be much easier if SAS/CONNECT would allow the programmer to supply a port in real-time and allow the socket to be used by the remote and local hosts.

Client/Server in the Present

In the academic setting of Penn State University, the demand to move users off of the large mainframe began about five years ago. The University decided to first remove the batch operating system, MVS from the mainframe and the push is now on to move as many of the applications and software packages off of the mainframe and onto the workstations as possible.

In the past two years, the University has granted all students internet accounts that provide them with electronic mail (e-mail) and access to the Internet. The students who take programming courses are now being forced to use an IBM or Macintosh to do their course work and not the mainframe. It is the opinion of the author that the mainframe still has many valuable uses and does many tasks that can not currently be
accomplished in an efficient fashion on a workstation or PC.

Clients are being used to provide the access to electronic mail and the servers are being written for Unix platforms. The move is on to remove the need for the mainframe for academic computing at Penn State University.

Is Client/Server Really the Way to Go?

Many of the problems encountered when developing software to run on the mainframe may come back to haunt those who are pushing to move everything to a client/server system.

Some of the tasks that seemed easy to develop and code on the mainframe are now having to be rethought as the move to client/server and "get on the Net" continues to evolve.

Client/server has its problems in its early life. One of the main problems is how to get large amounts of data across a TCP or other connection at a fast and efficient rate. Of course, this means more data lines and more packets being sent over the network. If the network is not secured, packets containing sensitive and confidential data can be easily taken right off of the wire. This is only one of the potential problems that may be encountered when making the transition from the mainframe to a workstation or PC.

Using SAS to Develop a Client/Server Strategy

The tools that are necessary to develop a client/server application using SAS are SAS/CONNECT, SAS/SHARE, a working and reliable connection between the workstation, hosts and mainframe. Let us also understand that in many cases the mainframe running CMS or MVS is not being developed as the ideal server platform. Unix seems to be one of the chosen operating systems to easily develop a client/server system.

When using SAS to create a client/server system it should also be noted that SAS/CONNECT is required to be installed and licensed on both platforms and all workstations using SAS/CONNECT software.

One of the main parts that items that needs to be remembered when developing a client/server system and running across platforms, is to only transfer the data necessary to complete the transaction or generate the report. If the interactive AF or batch program is transferring large amounts of data across platforms it may be best to issue a remote submit and only bring the report or massaged data across the TCP link when the job is completed. To run a job remotely the programmer will need to create a program which issues an RSUBMIT command to the remote host and include the necessary Base SAS code to achieve the goal of creating the graph or report and then utilize some of the graphics features of the workstation to display the resulting data report or graph.

Example of using RSUBMIT to Remotely Process a Job

libname rock 'd:\sas\data';
rsSubmit;

proc download data=mylib.shoacct out=rock.account;
proc upload data=rock.solid out=rock.solid; run;
endsSubmit;
signoff;

Suggestions for Developing a Client

With the implementation of Remote Library Services (RLS) in version 6.08 TS 405, SAS provided an alternative for remote submission of jobs that simply require access to a data set that will only result in a small amount of data being transferred across a TCP/IP link or other similar connection supported by remote library services.

Remote library services is a SAS system tool that allows for the development of Intelligent client/server applications. Remote library services also allows the programmer to define a SAS library reference on a remote platform and provide the user with access to the SAS data sets transparently on the local host. To use remote library services, the application on the local workstation needs to supply a library reference and the information about the remote SAS data library. The "remote" engine is used to inform the SAS program to utilize the remote library services access engine.
Example of a remote library service reference:

```
signon;
  /* Initiate the necessary sign on to the
   remote host */
lbrd mylib remote "C" server=mynode;
  /* Specify which server to use for the remote
   library */
```

The above example sets up a library reference on the local PC, mylib, and specifies the path for the remote library on the remote host, "C".

When Should the Application Not Use Remote Library Services

The application or batch application should not use remote library services if the end-user is going to lead a large number of observations. Remote library services should also be avoided if the network which is allowing for the data transfer creates large delays in response time.

The applications programmer should always keep in mind that in the current world a few seconds can seem like a very long wait for the end-user.

Looking to the Future with SAS and Client/Server Development

As the workstation platforms become faster and the storage devices for local workstations also increase their access speeds, SAS will need to continue to provide start up times that decrease.

One of the biggest problems when using SAS applications or running SAS interactively is the slow start up times. This is especially noticeable on the mainframe platform where users are used to quick start up times and do not want to have to wait five or ten seconds to have the application start up.

SAS Institute also needs to continue to develop the TCP/IP support. The need to allow SAS to open sockets and ‘talk’ to a remote host are going to become more important as the move towards Unix and PC platforms continues to increase.

Currently SAS only allows the remote host to start SAS with a line-mode telnet port available. If SAS wants to compete for the client/server world and allow for the use of WINSOCK or other TCP/IP methods of connection, then SAS needs to allow for selection of sockets without the need to connect in a mode that all platforms do not support.

SAS as a Power-Builder

SAS Institute should also position itself to allow for the development of applications that create Visual Basic controls (VBXes). If SAS were to be incorporated into a Visual Basic application and not require that the SAS system be ‘up and running’, this would allow the developer to create applications that can utilize the strength of all of the different software packages on the market today.

With the release of SAS 6.11, the developer may be able to accomplish this. SAS Institute is continuing to move to a stronger power builder and object oriented programming language. In the near future this will allow SAS to compete with some of the other power builder software packages like Microsoft’s Visual Basic.

Conclusions

As the Internet continues to grow and the demand for more distributed applications grows, SAS will play a key role in their own destiny. Someday in the near future one may be able to connect to a Web site on the Internet and submit a SAS job which will create an extract of data stored at a remote site. The user could then take remote library services to a new level by accessing data across the Internet.

Client/server is here to stay, but it also has a long way to go before it takes over many of the large corporations that have millions of observations to access and generate reports.

The Pennsylvania Commission on Sentencing is moving toward distributed processing and away from the mainframe and DOS programming with the SAS system and other small software packages. The move to client/server will allow the Commission on Sentencing to be on the leading edge and provide the data and reports to the judicial system in the Commonwealth of Pennsylvania.

SAS is playing a key role in the development of the distributed application. Without the use of the SAS system, the application may have had to take a step into the future and be written in a lower level programming language and be run on the mainframe with the hope that the mainframe at Penn State University will be around forever. With the current growth of the University and the
demand for more state of the art networks and operating systems. CMS at Penn State will not be around in the next five to ten years. Without the current move to the workstation, the Commission on Sentencing would have to rewrite the application once the mainframe was deemed obsolete and removed from service.

By using the SAS system, the application and data can be easily moved to any other platform that SAS supports, and with SAS now running on virtually all platforms and SAS institutes continued support for multi-vendor architecture, the application can grow and move as the operating systems continue to be developed now and in the future.

References

*SUGI 20 Paper: Distributed Processing: Using Remote Library Services to Bring Storage to the Workstation, Daryl R. Hoffman, 1995*

*SAS/CONNECT Software: Usage and Reference, Version 6, Second Edition*

*SAS Companion for the Microsoft Windows Environment, Version 6, First Edition*

*SAS Companion for the CMS Environment, Version 6, First Edition*

*Building Your Client/Server Solution with the SAS System*

*Client/Server Computing with the SAS System, Tips and Techniques*

*Essential Client/Server Survival Guide, SAS Institute*

Acknowledgments

*SAS, SAS/CONNECT, SAS/SHARE and SAS/AF are all registered trademarks of SAS Institute, Cary, NC*

*Visual Basic, DOS and Windows are all trademarks of Microsoft Corporation*