Client/Server Applications: Experience from the Management Information Systems Department at SAS Institute

Ria Poole
SAS Institute Inc., Cary, NC

ABSTRACT

In an effort to keep up with changing technology, applications developed by the Management Information Systems Department at SAS Institute have had to evolve rapidly in the past few years. Many applications began on mainframe computers and have successfully made the transition to the desktop. Others use the power of the mainframe to serve clients on other platforms. This paper will give an overview of the evolution of Client/Server applications developed for internal use at SAS Institute. The main focus will be on the Account Management System used by the sales organizations.

INTRODUCTION

The SAS System has many features to make the task of porting applications and using data from remote platforms relatively easy. From the standpoint of an applications developer, client/server design with SAS software is practically magic. There are a number of concerns, however, to keep in mind in order to make this transition as smooth and effective as possible. These can include, but are not limited to, size and speed of machines to be used as servers, network configuration and client connectivity, and cost. Most developers cannot control these aspects of their computing environment, and therefore they strive to create systems that best make use of whatever is available. At SAS Institute, members of the Management Information Systems (MIS) department have taken many existing internal applications and successfully moved to a client/server environment with different platforms working together.

History of the Account Management System (AMS)

Five years ago, the Software Sales and Marketing division took a look at their existing prospect management systems and decided to order a new design that better fit their changing styles of selling. They needed to have more real-time sharing of information across all levels of their organization. They also had the desire for more dynamic structures that would allow greater flexibility of text entry. It was also the dawn of Version 6 SAS System software, and they wished to take advantage of the new technology. It was at that time that the Account Management System, better known as AMS, was born.

AMS was designed on a VM mainframe as a multi-user Information System with SAS/FSF®, SAS/AF®, and SAS/SHARE® software. It had in excess of 100 users in five locations including: Cary, NC; Chicago, IL; Austin, TX; Irvine, CA; and Toronto, Canada. All sites were networked back to Cary and accessed the same SAS/SHARE server on VM. Along with the changing makeup of the user community for AMS, there have been many changes in the modes of access to the system. What started with VM native terminals and then added UNIX workstations, quickly has evolved to include personal computers with the aid of VM or UNIX terminal emulation. The next phase of AMS client/server design is referred to as "AMS-Lite", a version for the laptop which involves working with a downloaded subset of AMS data and functionality to take on the road. Finally, a direct interface to the UNIX data using a personal computer under Windows will be included in the multi-platform design.

Moving the AMS from VM to UNIX

Little re-coding was necessary to convert the AMS from a VM SAS/SHARE environment to that of one served from UNIX. A Korn shell executable script was created on UNIX that calls code, by means of an INITSTMT parameter, that is very similar to the AUTOEXEC code run on VM. The LIBNAME statements were changed to point to the new UNIX locations and associated servers. From the mainframe, a few additional lines of code are necessary in our environment to let the system know what protocol is being used and where the server is located.

Example 1 Sample Code for Cross-Domain Server

```plaintext
options commands=tcpi;
%let servhost=servhost.unx.sas.com;
libname xdat 'unix-path_name'
server=servhost.servnamei
```

A dedicated Hewlett-Packard 755 was set up to house the data and run four SAS/SHARE servers to help distribute the load. Because the AMS is the only system on this machine, the speed was a great improvement over the busy VM environment on which it had previously run. Even though many of the people were still accessing the data from VM, they did not suffer degradation in response time when compared to the VM native server.
Concerns about file-locking surfaced, but problems were prevented by having the data in NFS files owned and only editable by the server used. In this way, no user could access the data in write mode without the server.

The SAS/SHARE server is re-started each night and left running all day to serve the multiple users who connect and disconnect this each day, it makes it easier to have smaller logs to look from it as needed. Although it is not absolutely necessary to do this each day, it makes it easier to have smaller logs to look through should problems arise. We also run monitor programs that check every few minutes to see if the servers are still running and that restart them automatically if they are not. Messages are sent to the owner of the server when this happens so they can quickly determine if there is a problem.

Most of the application code remains exactly the same on both platforms which makes development and maintenance easier. However, in order to provide a more workstation-like appearance, icons and gray-scale were added to the UNIX version of AMS. Facilities such as printing have also been changed to match the new operating system.

AMS-Lite

As many offices open up around the country to better support the needs of our customers, connectivity becomes more of an issue to applications. Not all offices have the luxury of dedicated lines for speed of data transmission. The need for some stand-alone account management support has become increasingly important. Account managers also have time out of the office on planes and in hotels which could be productive time if they had their account data available.

AMS-Lite is currently under development to satisfy this need. It will allow for the transfer of data from the UNIX server to a PC. The account managers can then dynamically add comments about a sales contact, request sales literature to be sent, or create a new lead record. These activities are dated, and only the observations added during the period that they were off-line will be appended to the live data on the server. This will eliminate any problems with having to lock records for an extended period or with timing of updates. They will do other sales tracking tasks while connected to the network.

The only portion of this system that will reside permanently on the PC will be the data porting facility. It consists of a single SAS/AF screen that prompts for whose data to download or for the date range off-line to upload. The date range for the upload can default to the time frame off-line. Any validation data sets and the SAS/AF catalog used by AMS-Lite are ported each time the facility is used to keep in sync with any system updates.

The simplest way to create a porting application such as this, is to use the Remote Connect button in SAS/ASSIST®. This invokes the SAS/CONNECT® product dialog. After using the menus to walk you through porting the data or catalog the first time, you can file the code generated by SAS/ASSIST and use it inside your own system. This is a great way to speed up the coding process. Example 2 shows a simplified piece of code.

Example 2 Sample Code for SAS/CONNECT session

libname inl 'pathname';
options comment='tcp';
SIGNON 'remote_script';
RSUBMIT;
   libname outl 'pathname';
   PROC UPLOAD TNLIB=inlib CTLIB=outl; run;
ENDRSUBMIT;
SIGNOFF 'remote_script';

AMS Future

Shortly, the AMS will also serve personal computers running Windows as clients from the UNIX data. These will be hooked into the network and have full use of all AMS facilities. At that point, we are hoping to discontinue the VM version. This will allow for the redesign of many portions to take advantage of the newer GUI features that are available to the desktop platforms. AMS-Lite will actually begin this redesign phase since it will be easier to manage changes on the scaled-down version.

Other Client/Server Systems at SAS Institute

Another style of client/server that has been used successfully at SAS institute is the linking of stand-alone systems on different platforms. Information Systems Inventory (ISINV) and Network Administration (NETMAIN) systems are linked via remote computing services. ISINV resides on MVS, while NETMAIN is on UNIX. Before cross-domain server technology, data was ported at night between the different operating systems. This meant that what was seen was at least a day old. Also the time required to transfer the data to the remote host took a substantial amount of time and required a great deal of space just to hold redundant data. The need for real-time information was solved with direct access to the data from the other system.

Even though ISINV is on a very powerful mainframe machine, it tends to slow down the processing seen from the UNIX side. This is mainly due to a greatly overburdened operating system, and is not a direct reflection of SAS/SHARE performance. There were also times in the early days of the new UNIX configuration that outages on the network caused the mainframe portion of the application to be unavailable. Not because that machine was not working, but because the data needed to remain synchronized. These issues demonstrate the need to keep the nature of each individual environment in mind when developing a client/server application.

Other items of concern when creating cross-platform servers involve things external to the application. And while an application's developer may not have direct control over these items, it is good to be aware of them. For example, just because you want a SAS/SHARE server to be of cross-domain type, does not mean it will "automagically" work that way. The systems people on both sides of the server have to make entries into files to enable that capability. Another thing to watch out for is the security issues on the platforms you are accessing. Our MVS platform uses RACF to check for valid users of that platform. Therefore, when we use a secure MVS server from UNIX, we must supply a userid and password that is approved by RACF.

Example 2 Sample Code for SAS/CONNECT session

libname inl 'pathname';
options comment='tcp';
SIGNON 'remote_script';
RSUBMIT;
   libname outl 'pathname';
   PROC UPLOAD TNLIB=inlib CTLIB=outl; run;
ENDRSUBMIT;
SIGNOFF 'remote_script';
This is accomplished by using the statement seen in Example 3 before setting the LIBNAME to the served data. It prompts the application's user for their userid and password on the remote platform.

Example 3 Sample Code for security check option

```plaintext
&let tcpsec_prompt;
```

CONCLUSION

Application developers in the Management Information Systems department at SAS Institute have successfully incorporated client/server strategies into their systems to better serve their user communities. SAS System Remote Library Services have made this endeavor possible and relatively simple to accomplish. As the need to share information between people in different organizations has grown, so have the capabilities of the systems designed to deliver this information. By taking advantage of the strengths of the computing environment, MIS is building a strong network of interconnected systems to help promote efficiency throughout the company.

SAS, SAS/AF, SAS/ASSIST, SAS/CONNECT, SAS/FSF, and SAS/SHARE are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

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