

Client/ Server and SAS® : End-user support in distributed environments

Gregory S. Nelson
University Computing and Networking
Services
The University of Georgia

Abstract

The concept of Client/Server computing has established some very concrete ideas of what we should be able to expect from our computing environments in the very near future. SAS Institute has been responsive in incorporating, at least conceptually, tools which allow us to use SAS in ways consistent with those notions. This paper will look at implementation and management strategies in consulting with clients moving towards a distributed computing environment.

Introduction

Organizations throughout the world are jumping on the client/server bandwagon. Education, government, health-care, banking, private as well as public entities alike are planning their next generations of computing infrastructure around the idea that information ought to be accessible - as Bill Gates said, "Information at your Fingertips." As new technologies and approaches evolve into what we see on our desktop, they bring with them an enormity of potential as well as complexity. The pendulum swings to and fro and we as visionaries are supposed to see where we've been and where we are to go all the while taking the organizations forward with technologies that we only have in hand and a snapshot of the pendulum as it is today.

® SAS, SAS/CONNECT, SAS/SHARE, and SAS/ACCESS are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.

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

As support personnel our tasks are clear: support our client base with applications and hardware. Operating systems that we have grown up with were easy to support. If a user had a problem, we could get the program, data files and if necessary, log onto their account and run the program. We knew the hardware, the architecture, the operating system and the software. However, in today's computing environments, our tasks are not as simple. As SAS consultants in a distributed computing environment we are supposed to be:

- * Computer consultants - knowledgeable in mainframe, workstation, and various desktop operating systems (OS),
- * Networking Specialists - linguistically capable with the various network operating systems (NOS) and interoperability issues,
- * Database Administrators - familiar with relational and non-relational database architectures as data repositories,
- * Systems designers - knowledgeable in designing cross-platform systems that will grow with the organization,
- * Application developers - adept at designing end-user systems and creating front-end systems,
- * SAS consultants - savvy in the use of SAS and the myriad of add-on products, including those specifically designed for distributed computing environments.

Clearly, in most organizations, these roles ought to be shared. If we've learned

anything as consultants, I think it is safe to say that our computing environments are not getting simpler, our data is not shrinking, our clients are not expecting less of us nor are they expecting less from their computers. So with all of this change, both in technology and responsibilities, how do we manage our sanity? Let's start by understanding what we're up against.

Definitions

Before describing the challenges of supporting end-users in distributed computing environments, it is important to understand what we mean. If we are to try to come up with some common definitions for terms which describe this phenomena of client/server computing, we probably could agree only on the fact that the industry as a whole doesn't agree on what these terms mean. It is not the purpose of this article to try to come up with definitions for these, but rather try to lay some framework for a shared understanding of these technologies and how SAS Software can meet these goals.

Cooperative Processing

A cooperative processing environment can be defined as one in which "portions of an application run on different computers, and data may be at various installations" (Atré, 1992). Cooperative processing implies that there is a distributed computing infrastructure that allows transparent access to resources. Cooperative processing can use a client/server model, peer-to-peer processing, or some other model. John Gantz (Gantz, 1991) describes at least one of the main features of cooperative processing as having a client/server or similar architecture to allow processing of a program across two or more processors.

Client/Server Computing

While cooperative computing deals with the physical computing environment, client/server computing makes the distinction between the client and server processes. A typical application has three basic components:

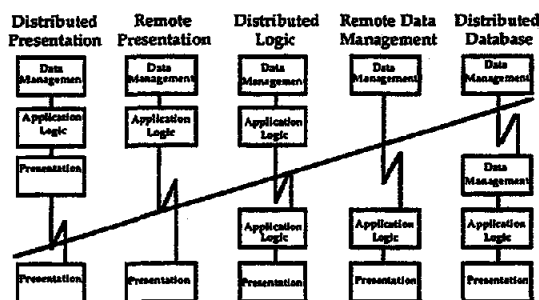
- * Presentation logic

- * Application logic
- * Data management logic

In the traditional model of mainframe computing, a user typically interacts with an application running on the mainframe by typing commands on a terminal (either a dumb terminal or an emulator running on a PC). The terminal does not have any processing ability, and all three of the components run on the mainframe. In a client/server model, these components would run on two or more systems. The Gartner Group describes five major models for distributing these functions in a client/server model, below I have described the three most common:

1. Remote Presentation - The client handles the user interface (presentation logic), while the server handles the computing (application logic) as well as the keeper of the data (data management logic).
2. Remote Data Management - Presentation logic and application logic run on the client side and only the data is on the server system(s).
3. Distributed or Split Logic - The client handles the presentation logic, the server(s) handles the data management logic and the application logic is split between the client and one or more servers.

Styles of Client/Server Computing



Source: Gartner Group

Just to complicate things, in any of the above cases, it is possible that part or all of the data is local to the client system.

Downsizing and Rightsizing

According to Atre, (Atre, 1992) "downsizing is the process of migrating complete applications, or some functions of the applications, from centralized mainframes to networks or smaller decentralized systems. It's also implementing the latest technologies, resulting in reduced costs, improved user access, and greater flexibility." We have seen the pendulum swing from a mainframe dominated world to moving everything and everyone off the mainframe to local area networks with attached PCs. It appears that the pendulum is now swinging away from the all-or-nothing mentality (i.e., mainframes vs. personal computers) to the more realistic notion of "rightsizing".

According to Morse, (Morse, 1993) "rightsizing is the processing of choosing among available hardware, software, and connectivity options so that the actual needs of a given application are served correctly and the underlying business considerations are given their due." Put simply, rightsizing is using the right computer for the right job. A department may be outgrowing their small server and need to migrate the application to a larger environment. On the other hand, the price/performance ratio for PC and RISC based computers makes it easy to justify moving to smaller systems. Using a PC running OS/2 makes sense when you are trying to produce a graphic of your data, but makes little sense when you have to sort the three million records and produce a summary data set for the graphic. Rightsizing would suggest using your powerful mainframe for the sorting and summarizing and the OS/2 machine for producing, or at least displaying the graphic.

Distributed Computing Environment

The term distributed computing refers to the general idea that computers, often loosely coupled or clustered, act in cooperation as a system in carrying out the functions of an application designed to meet the needs of integrating an organization's information. To end-users, the term distributed computing

signifies a seamless information and computing environment that is accessed from a graphical front-end. The end-user of such as system, cares not where the application is running, where the data resides, nor the network topology that makes it all happen. To systems administrators, a distributed computing environment means changing technologies, increased complexity, and a lack of management tools. To application developers, distributed computing environments provides supporting services and a framework that offers reduced complexity, interoperability, portability, and provides them with the ability to reuse code. Organizations expect to obtain flexibility, scalability, real time access to information, and faster development and integration of applications.

In his book on distributed computing, Amjad Umar (Umar, 1993) states:

"A Distributed Computing System (DCS) is a collection of autonomous computers interconnected through a communication network to achieve business functions. Technically, the computers do not share main memory so that the information cannot be transferred through global variables. The information (knowledge) between the computers is exchanged only through messages over a network."

Distributed computing systems refer to the array of computers acting in conjunction with one another to achieve a computing solution. Distributed computing environments are to be thought of as much broader to include not only the hardware, but applications, software solutions and management tools. Taken as a whole, distributed computing systems are analogous to an operating system for distributed applications.

The Users

Thus far in our discussions of client/server computing, we have focused on the terminology used to describe the technologies and those used to describe the systems

supposed to make the life of the users more efficient. But who are our users? As SAS consultants, we work with a variety of users that I have thus far, lumped into the term end-users. In any SAS application, we find ourselves consulting the users at all levels: first-year graduate and undergraduates whose task is to use SAS for a home-work assignment or class project; the data entry clerk who uses SAS to enter customer orders; the systems programmer who finds it easier to generate DASD reports using SAS instead PL/I. In a distributed computing environment who are our users? Although we find a variety of users in any SAS community, I will generally distinguish between those who write the programs and those who use them.

SAS and Client/Server Computing

In reviewing the components we have used to describe client/server computing, cooperative processing, and distributed computing environments, let's look at the tools that SAS has to develop applications for use in such an environment. Let's break the concepts that we discussed earlier into some major goals.

One of the first goals we found when discussing cooperative processing is that this implies that there is a distributed computing infrastructure that allows instant, transparent access to resources. End-users of a distributed application don't care whether the data is coming from a DB2 table from MVS or an Oracle database residing on a UNIX server. This allows those who develop the systems and maintain the data the flexibility of changing server locations with minimal interruption of service.

A second goal, described earlier as the ability to use one or more processors to handle the application, is a convenient one for both the users of the system as well as the developers. Since desktop platforms are very good at producing output, whether on-screen or in hardcopy, it makes sense to let people use the systems they are most comfortable with to handle the presentation logic. As someone interested in security, reliability and overall

robustness, I prefer to let the data management portion of the application be dealt with on a host-system such as MVS or a UNIX server. For the developers of such a system, it might be more productive to develop the prototype on a PC running Windows and then move the code to the host for production implementation.

As the application grows and/or changes, we ought to be able to move our applications to the platform where it makes the most amount of sense. As a third goal, scalability and portability are no minor feats, especially, if the application wasn't designed with a different architecture in mind. Furthermore, we ought to be able to divide the three components of client/server computing among the platforms and positioning in the network topology where it makes the most sense.

In the classic sense, my application should be able to access data regardless of location and size. I ought to be able to access the application from the client of my choice - from a PC AT to a Macintosh to a UNIX system running X-Windows. I shouldn't have to log into a computer just to access the billing data and log onto another for producing the monthly reports.

Various features of the SAS System have allowed us to achieve these goals:

Multi-vendor architecture:
Version 6 of the SAS System is available for different hardware platforms and operating systems. Ones that we license at the University of Georgia include: MVS, CMS, UNIX, DOS, Windows, and OS/2.

SAS/CONNECT software provides data transfer, remote computing, and remote library services that allow for distributed applications to be developed.

SAS/ACCESS provides the ability to access a variety of

proprietary data formats including: Oracle, Sybase, DB2, IMS, SQL Server and others.

SAS/SHARE allows concurrent access to data making the application more functional as a multi-user system.

ODBC allows SAS access to various Windows-based applications as a client as well as a server.

Other authors have discussed the use of SAS in developing and implementing distributed applications (e.g., Garner, 1993; Mehler, 1994). We know that the tools are there. We feel confident that SAS is committed to providing tools for developing applications in distributed environments. We are realizing the benefit of off-loading users from the mainframe to desktop machines and workstations. But how do we take our centralized support staff and support a distributed computing environment?

User Management Models

The traditional debate in MIS departments and computer centers was between centralized versus decentralized computing: whether computing should be managed by a single department or distributed among each department (King, 1983). With distributed systems it becomes more difficult for separate departments to be totally isolated. By virtue of the fact that it is distributed implies that there is a network infrastructure and a network connection that must be managed. While each department may maintain their own internal network, there still needs to be central services that provide inter-department connectivity and, possibly, even connectivity to the rest of the world. In addition, when one department or the central service puts up a service on the network that is useful to the other departments there is a great deal of pressure to make it possible to use that service.

Some form of central organization of computing services is crucial for any effective organization. However, the amount of control that the central services have is more a factor of the organization. This statement is justified by looking at the management models of computing through an examination of them and then proposing a model for distributed user management. There are a variety of models which describe the management of end-user computing. In fact a whole body of literature exists which explores the various types of users in organizations (e.g., Brancheau and Brown, 1993; Cotterman and Kumar, 1989).

Centralized. Computing began in a centralized environment. The mainframes were controlled by the MIS department or computer centers and those who wanted computing services had to go through them. The computer center controlled the configuration and access to the computer. In many ways, one had no choice but to take whatever they provided. It was simply the case that there was no way individual departments could purchase a computer. Even if they could, they would have to spend large amounts of money maintaining the system, as well as spending more money hiring qualified people to run and program the machines.

Decentralized. The introduction of the minicomputer was the first step in significantly changing the management paradigm. The cost of ownership and maintenance had reached a level where individual departments that could take advantage of this fact and split away from the central facilities. This still required a sizable investment as staff was still necessary to maintain the computers and even provide programming support. However, the focus of the computing staff was now tuned to the departmental level and not to the organizational level so the users enjoyed a more suitable environment. In many ways, though, the environment is still a centralized environment, it is just that the users don't mind as much because the environment is probably less restrictive in some ways but just as restrictive in other ways they had not yet noticed. We enter the age of the

decentralization. There were still organizational bindings, but each computing facility was generally independent of all others.

Every Person for themselves. The workstation and the personal computer created another shift in management. Today, the cost of obtaining computers and software is at a level where most people can afford to purchase. Users no longer have to be at the whim of their department or any central computing facility. They can go out and purchase their own computers, and finally, do whatever they want - no more restrictions on CPU time, disk space, or user id naming - the machine is all theirs. Unfortunately, the cost of ownership has many subtle aspects. Ownership of a computer is similar to owning a house. They can no longer expect someone else to fix the problems that arise. It becomes the owner's responsibility for fixing the problem or paying for the repairs, and for the new owner, most of the problems that do arise are unfamiliar and require assistance. This situation describes a management environment best described by Peg Schaefer as the "Warring States of Greece" (Schaefer, 1992). This environment is characterized by lack of central support. Some central facilities may exist, but it is really up to each individual department or person to provide their own computing support. Cooperation may occur but competition is more common. This extreme state of decentralization generally results in a large degree of duplicated work. The quality of the computing environment for each department is almost solely dependent on the quality of their system manager.

The Compromise. An intermediate option, the laissez-faire model, exists. In the laissez-faire model, central management creates specific services in order to address the problems in each management category. It is up to each department to take and use these services and customize them, or to provide solutions on their own. The laissez-faire environment is a balancing act between flexibility and manageability.

In some organizations, only one of these models may exist or there may be a variety of

differently managed environments. The next section will describe the management structure of end-user computing at the University of Georgia and some things we've learned.

Distributed Management

Most organizations choose a decentralized model to support a distributed computing environment. In the terminal-to-mainframe computing model, there is no need to have local support, as all the complexity is on the mainframe. In the distributed computing model, supporting end users becomes complex; it is very hard for a central organization to support end users. When problems crop up, they could be with personal workstations, LANs, bridges, routers, servers, the software or hardware anywhere along the route. In many cases, a peer-support model emerges in which users who have become experts help their coworkers. Unfortunately, the local experts become burdened with the task of performing tasks that they may not be sufficiently prepared for, nor paid for, nor evaluated on when it comes time for performance appraisals.

The University of Georgia Computer Center has generally been a strong, central resource for our computing community. The computer center, known as University Computing and Networking Services (UCNS), is divided into three major components: Computing and Communications Facilities - responsible for the maintaining the physical nature of the campus computing infrastructure, the Administration and Planning, and Client Services. Client Services is further divided into: Applications and Consulting Support, Training and Information Services, and Networking and LAN Support. Between these groups, we find most of the support infrastructure for our campus community. The key players are the Help Desk, the Mainframe and Statistical Software Support Group, the Microcomputer Support Group, the Workstation Support Group and the Networking Support Group.

Generally, we take the approach that we will continue to manage those resources that can only be successfully managed centrally (e.g., the mainframe and backbone services) and by developing and providing tools to departments which help them effectively manage their own local resources. For example, we provide general services like offering central e-mail and internet services, access to application software on our mainframe resources. In addition we provide some specialized computing support in computational chemistry, physics, scientific visualization and statistical software support. On the other hand, our Networking and LAN Support Group has domain liasons in each department responsible for the local networking environments.

In addition, we have other types of facilities available for helping users. We take advantage of some most common forms include the following:

- * Help Desk - a central place where users can call or come in person for help. This serves as first-line help for users as well as a central point of contact to direct questions and find assistance.
- * Electronic mail addresses - although most questions are addressed to helpdesk@uga.cc.uga.edu, some specialized questions get routed to more specific e-mail addresses where turnaround is usually better and questions are usually asked in a clearer manner.
- * On-line help systems - Gopher has provided the best mechanism we've found thus far with gopher-space available for FAQ's as well as the ability to use "ask-blocks" for generating e-mail from end-users (not implemented at UGA, but an interesting thought)
- * Campus Wide Information Service (CWIS) - We currently have two systems

for document delivery. IDS was developed 10 years ago and was quite revolutionary at delivering plain text or script formatted documents from a central documentation repository

- * On-line and personal training, training through video-tapes - Currently, we do not use much on-line training with the exception of Seminar-on-a-Disk for selected micro software packages. We do have a very large training curriculum for each of our departments (e.g., mainframe, statistical software, Internet, PC's etc.)
- * Vendor help hot-lines - taking advantage of vendor supplied technical support is usually a last resort but often necessary
- * Bulletin boards, distribution lists and newsgroups - for discussion and announcement of official information we take full advantage of ListServ and Usenet News for both peer-support as well as announcements
- * Consultants on call - perhaps the best resource, consultants are always in high demand, overworked and underpaid
- * Expert systems and problem tracking/ change mangement systems - Perhaps the most ambitious of our goals for the next year, both areas will help us know more about our users and the kinds of problems/ questions they are asking and the concerns that they have
- * Surveys and Focus Groups - continuing to get-to-know our client base is an ever present task where we find that our perception is not our clients' reality.

Although not comprehensive, this probably describes one or more techniques employed by most sites to help their users. With user

management, a good rule of thumb is to help users help themselves. Various methods are useful in obtaining this goal including user-contributed areas, newsgroups, listserv's and FAQ's offered by more experienced users. Thinking about what levels of support you are giving a specific platform or software is probably an active process. How you plan to manage your users, is more often not something that most organizations address directly. However, methods of managing users becomes more critical when we distribute the computing environment.

Other tools and techniques that will help us along this path and are really inherent in the strategies above include helping the end-user help themselves. We also must help ourselves. Training in the use of: hardware, software, telecommunications, networking, database administration, systems design, SAS programming, and as consultants will always help us do our jobs better. Knowing what our jobs are and what others in our organizations can help us with might seem too trite to mention, but becomes imperative as our clients become users in distributed systems. A final suggestion might include managing the process we follow to help our clients. Design "process flow" charts to assist our help desks and first-line support personnel determine the best route a user should follow to get the best help.

There are numerous ways of implementing a user management system. Clearly, much of the implementation is dependent upon the current organizational structure. UCNS is in reality a broad mixture of management strategies defined above. Since we have a strong history of centralized support it was easy for us to define our support level for hardware and software for existing services. We continue to manage our mainframe based services running on our ES9000 and Cyber 960 as we did before - as a centralized system. Developing strategies that can be adapted and implementing systems that are scalable are perhaps the only way to manage these changing environments. One thing will always be constant - "things will change" and it is our response to change that will determine our success.

References

- Application Reference Designs for Distributed Systems. IBM. 1993.
- Atre, S. (1992) Distributed databases, Cooperative Processing, and Networking. McGraw-Hill.
- Barnhart, A. (1994) Sharing Enterprise Data with ODBC. Proceedings of the Nineteenth Annual SAS Users Group International Conference, 19, 284-288.
- Berson, A. (1992) Client Server Architecture. McGraw-Hill.
- Brancheau, J.C. and Brown, C.V. (1993, December) The Management of End-User Computing: Status and Directions. ACM Computing Surveys, 25(4).
- Can SAS Serve the Enterprise. (May 1, 1992). Datamation.
- Clifford, W. D (1992) Database Features Extend the Scope of SAS/SHARE Software. Proceedings of the Seventeenth Annual SAS Users Group International Conference, 17, 335-340.
- Cotterman, W.W. and Kumar, K. (1989, November) User Cube: A Taxonomy of End Users. Communications of the ACM, 32(11).
- Creech, S. and Van Wyk, J. (1991) Using SAS/CONNECT and SAS/ACCESS Software to Access Data in a Distributed Environment. Proceedings of the Sixteenth Annual SAS Users Group International Conference, 16, 572-579.
- Gantz, J. (1991, January) Cooperative Processing and the Enterprise Network. Networking Management, 25-40.

- Garner, C. (1993) Techniques for Distributed Application Development. Unpublished Manuscript.
- Garner, C. (1994) The SAS System: A Complete Client/Server Solution. Proceedings of the Nineteenth Annual SAS Users Group International Conference, 19, 247-257
- IBM Open Client/Server Computing. IBM, 1993.
- King, J.L. (1983) Centralized versus Decentralized Computing: Organizational Considerations and Management Options. ACM Computing Surveys, 15(4), 319-349.
- Mehler, G. (1994) Integrating the SAS System for Personal Computers into your Enterprise. Proceedings of the Nineteenth Annual SAS Users Group International Conference, 19, 428-437.
- The Most Complete Solution: SAS System Provides Support for Multiple Client/Server Models. (1993). SAS Communications, 19, 13-17.
- Morse, S. (1993, February) Rightsizing: Tailoring the Applications and the Platforms. Network Computing, 63-77.
- Nelson, G.S. (1993) "Rightsizing" with the SAS System for Information Delivery. Presented at the Annual Convention of the Regents of the University System of Georgia, Rock Eagle.
- SAS/CONNECT Software: Usage and Reference, Version 6, First Edition, Cary, NC. SAS Institute Inc.
- SAS/SHARE Software: Usage and Reference, Version 6, First Edition, Cary, NC; SAS Institute Inc.
- SAS and the Software Revolution. (May 10, 1993). Information Week.
- SAS: More Things to More Users. (February 10, 1992). Information Week.
- SAS Rolls Out Series of New Tools, Upgrades. (February 10, 1992). ComputerWorld.
- Schaefer, P. (1992, October) Is Centralized System Administration the Answer? LISA VI Conference Proceedings, Usenix Associations.
- Unlocking the Potential: The SAS System and IBM Information Warehouse Framework. SAS Institute White Paper.
- Umar, A. (1993) Distributed Computing: A practical Synthesis. Prentice-Hall.
- Wallace, J. (1992) Maximizing Computer Resources with Cooperative Processing and Distributed Data Access. Proceedings of the Seventeenth Annual SAS Users Group International Conference, 17, 641-650.
- Wallace, J. (1994) New Features for Remote and Open Access to Enterprise Data: Enhanced Distributed Data Access Through Multiple Engine Architecture. Proceedings of the Nineteenth Annual SAS Users Group International Conference, 19, 258-267.

Address

Gregory S. Nelson
179 Computer Services Annex
The University of Georgia
Athens, GA 30602-1911
gnelson@uga.cc.uga.edu

Acknowledgements

I would like to acknowledge the assistance of Glenn Leavell, Rex Walker and Joe Kelley for their helping in reviewing this manuscript.