Integrating the SAS® System for Personal Computers into Your Enterprise

Gary Mehler, SAS Institute Inc., Cary, NC

ABSTRACT

The large number of ways in which the SAS® System supports client/server architectures provide many opportunities for Information Systems designers to access enterprise data. In addition, PC-based workstations, networks, and fileservers represent additional variables in the enterprise computing equation. With these opportunities come decisions that effect efficient resource utilization. This paper attempts to analyze various configurations and their impact on the hard resources that support them. Although your mileage will vary, the examples presented herein may help in the enterprise system design process.

INTRODUCTION

As workstations and personal computers have become increasingly popular, many organizations are taking advantage of desktop computer availability to extend the effective reach of their centralized systems. For many, this involves interfacing desktop PCs to the large mainframe system. Although the minimal implementation of this approach utilizes the PC as a terminal to the mainframe, many other types of interconnections are possible as well.

The SAS System, by virtue of its support of various types of computer hardware ranging from PC to mainframe, provides for many ways in which cooperative processing configurations can make optimal use of this wide range of hardware. Various products and features allow inter-platform communication. The following features and products form the basis for enterprise-wide computing solutions.

- Multi-vendor architecture: Version 6 of the SAS System is available for several types of computer hardware and operating systems, including: MVS, CMS, VSE, OpenVMS VAX, OpenVMS AXP, UNIX, Windows NT, OS/2®, and Windows 3.1. Their compatibility in terms of SAS programs, user interface, and data transportability make inter-system use of the SAS System on various platforms an attractive possibility.

- Data transfer services with SAS/CONNECT® software: Data can be transported across systems from within the SAS System using commands available in the SAS/CONNECT product. If data will be analyzed multiple times, a local copy can be obtained to minimize network use.

- Remote computing services with SAS/CONNECT software: SAS statements and programs can be remotely submitted to the SAS System running on another computer using SAS/CONNECT software. This allows compute or data-intensive processing to occur on the most capable computing resource. As remote processing progresses, program listing and graphical output are presented on the local system as if the program had executed locally.

- Concurrent data access with SAS/SHARE® software: Multiple users and processes can fully access the same data sets utilizing SAS/SHARE. In this way, large transaction-based systems can be developed.

- SAS/ACCESS® provides the ability to access other DBMS systems, including DB2, ORACLE®, SQL Server, and AS/400. In addition, use of SAS/ACCESS with SAS/CONNECT allows access to database systems on other computing environments.

- Remote library services with cross-architecture data access: Available with 6.09 SAS Systems (and 6.08 Maintenance systems), this feature allows SAS/CONNECT and SAS/SHARE applications to provide transparent data translation so that SAS data sets can be accessed from any SAS system. An example use of these services is that the SAS System for Windows can directly utilize SAS data stored by a mainframe SAS application. The local SAS System is able to access any remote data set as if it were on the local system.

- Open Database Connectivity access: Support for Microsoft Corporation's Open Database Connectivity (ODBC) standard allows SAS System access to various Windows-based applications from a wide range of vendors. For example, the PROC SQL Pass-Through facility allows queries and other requests to be sent to applications such as dBASE, Paradox, Access, and Excel for execution.

  In addition, an ODBC driver for the SAS System allows other applications to access SAS data utilizing this communication standard. This would allow, for example, an Excel spreadsheet to access data in a SAS data set.

UTILIZING SAS SYSTEM FEATURES

These various products and features allow for the design of systems that can operate in a true client/server mode. This mode means that various portions of the computing network can provide services for clients, which will typically be desktop computer systems running the SAS System.

Some options for client/server configurations include:

- A SAS System application for data entry or retrieval that allows PCs to access data stored on a central computer system. This model works well if relatively small amounts of data are accessed infrequently. Use of this model in an application that accesses a large portion of the data multiple times would generate a large amount of network traffic that would slow completion of the task. In addition, this network traffic would also slow the operation of other network users.

- A SAS System application for data analysis that downloads data from a central system for processing by the local SAS System. This model would be used in an application that repeatedly required access to an entire data set. It would be inefficient, however, if the actual use was on a small subset of the overall data transferred. The time required to transfer the entire dataset would be large relative to the amount of useful data really transferred.

Again, though, another aspect is the amount of network traffic generated by this large transfer. If most of the resource use was for unused data, useful bandwidth could have been taken from other users.

- A SAS System application that remotely executes programs on the central system to manipulate the centrally stored data. This model works well for applications in which the central computing resources are plentiful and desktop resources limited. The drawback to this approach is that central computing resources can become limited and all users might be more efficiently served if large computing requests could be handled remotely by sufficiently powerful desktop computers.

So, although a wide range of configurations are possible, some do not make efficient use of computing and network hardware in certain system designs. A large number of references have been added to the end of this paper and can be consulted for more detailed information on these models.

SAS SYSTEM OPERATION ON PC NETWORKS

The SAS System supports operation on a wide range of network media and types. See Appendix A for a listing of supported networks.

In addition to the data transfer and remote computing opportunities offered by the SAS System in general, PC network administrators have other options that can be exercised to make the most cost-effective use of desktop hardware. Basically, these involve accessing data stored in remote locations. Each of these options allow for multiple, concurrent, read-only SAS System sessions or a single read/write SAS System session to access centrally stored catalogs, data sets, or executable components of the SAS System. SAS/SHARE software on the remote system can provide access for concurrent read/write sessions if the remote system is running Windows NT or OS/2.

The following locations can be used when accessing data from a PC application:

- Local data. No network is required or utilized. This case is illustrated in Figure 1.

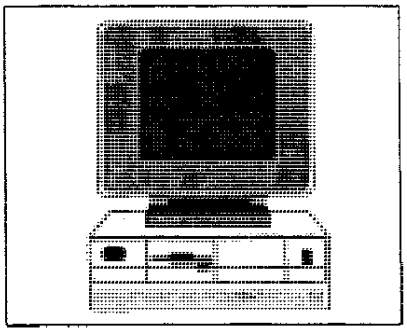

Figure 1. A stand-alone PC uses its own CPU and disk resources exclusively.

- Another desktop PC. Peer-to-peer networking packages, including Windows for Workgroups and Windows NT allow any PC to act as a data server to other PCs acting as clients of that PC. While an extremely cost-effective solution for data sharing, this approach does not handle a large number of client PCs, and even a small number of clients can significantly degrade the performance of the desktop PC acting as the data server. This case is illustrated in Figure 2. SAS/CONNECT software extends this model into the realm of remote access of CPU resources on OS/2 and Windows NT computers running the SAS System. This allows the SAS System on the local PC to execute code on the remote computer that is can run the SAS System under OS/2 or Windows NT.

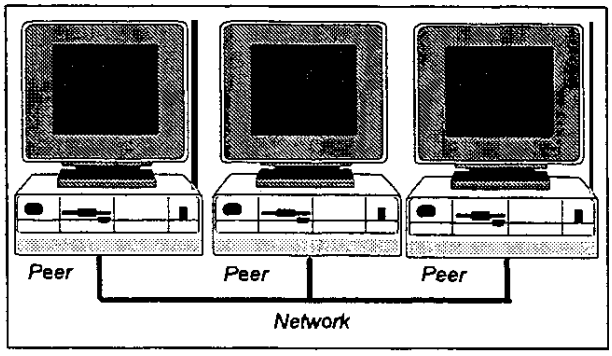

Figure 2. PCs on a peer-to-peer network (such as Windows for Workgroups, Windows NT or OS/2 LanServer) are able to share each other's disk and printer resources through the network.

- A PC-based fileserver. Since these fileservers are typically dedicated to the task of providing access to data, this approach can support a large number of client PCs. Among the most capable of these is Novell Netware, which allows a single server to handle up to 1000 simultaneous client connections. Figure 3 shows an example use of a PC-based fileserver servicing multiple clients.

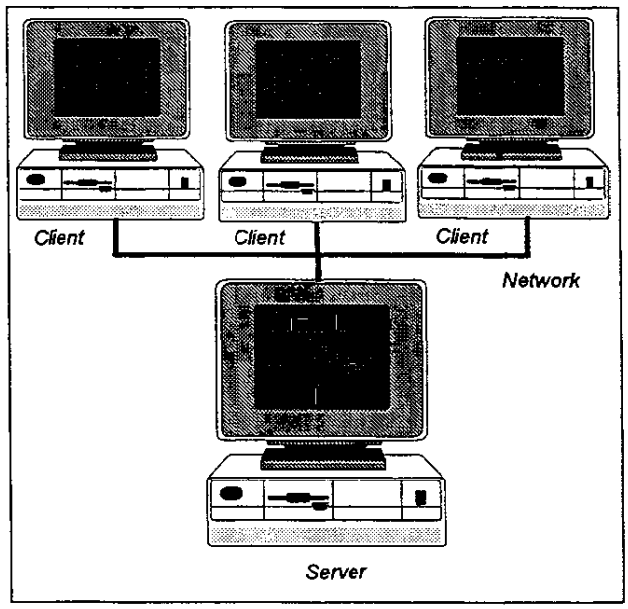

Figure 3. A Client/server topology. Client PCs make requests of the server computer's disk or printer resources. This is the model used for Novell fileserver systems.

- UNIX and other NFS-based systems. This approach allows the desktop PC to access files stored on the remote system as of they were stored on any other remote disk. Since the computers acting as servers in this case can be more powerful than PC-based systems, these can be attractive for some applications. This case is similar to that shown in Figure 3, except that the server machine in this case is a non-PC computer.

Some of these remote systems support maximum disk sizes and file lengths greater than DOS itself currently allows. Since DOS systems don't allow the use of hard drives larger than one gigabyte, that value also effectively limits maximum file length. If the operating system on the remote data server supports sizes larger than one gigabyte, that exposes the limit for operating systems that use 32-bit file offset values. DOS, OS/2, and Windows NT normally use this type of file access, so this becomes the current limit for PC-based file sizes. This value is two gigabytes, which becomes the maximum size of a data set size that can be used on these systems.

These large disk drives can be expensive to obtain if not using an advanced operating system that allows multiple, smaller disks to be "spanned" or "striped" into a larger unit. Novell Netware and Windows NT Advanced Server support this capability.

Once the type of remote computer being utilized has been determined, options concerning the actual distribution of data across local and remote source can be considered. Some of these configurations include:

- Running the SAS System from the local hard drive and accessing local data sets. This is the simplest model, and doesn't utilize the network. It allows for the best performance of the SAS System, but requires the most disk space resources on each desktop. This case was illustrated in Figure 1 earlier.

- Running the SAS System from the local hard drive and accessing data sets from a network data server. If the data are very large, this can be an attractive solution. This configuration was illustrated in Figures 2 and 3 previously, depending on the network software used. If concurrent read/write access is required, SAS/SHARE software must be used, and the remote system must be capable of running the SAS System. SAS/SHARE software is available for all computers supported by the SAS System except for Windows 3.1.

- Running the SAS System from a network data server and accessing data sets from the local hard drive. If quickest access to data is required, this may work. Since the SAS System is a large application, accessing it entirely from a fileserver can be time-consuming if SAS System sessions are frequently started. This would be the case for a script of SAS System batch programs. This was also illustrated in Figures 2 and 3 earlier.

- Running the SAS System and accessing datasets from a network data server. This minimizes the desktop disk space requirement, but can place a high load on the PC network, especially for power users. This configuration was illustrated above in Figures 2 and 3. Once again, it bears note that concurrent read/write access requires the use of SAS/SHARE software on the computer acting as the data server.

- A hybrid approach. New features being developed in Release 6.10 of the SAS System allow monitoring of all SAS System components as well as user data sets and catalogs that are accessed. Using this information, the network administrator can design efficient SAS System installations that place the minimal working set of the SAS System on the local disk for optimal performance without extra disk space use. This configuration effectively looks like Figure 1 shown above, since the network isn't required to run this minimal set of operations.

EFFECTS OF CLIENT/SERVER AND FILESERVER CONFIGURATIONS

The network which interconnects desktop PCs to remote computing and data servers is a finite resource which works best when its overall utilization is low. Data are transferred across a network through encapsulations called "packets" which are typically sent between two computers. The packets include data and extra information to identify the sender and intended recipient of the data. The two computers are taking part in a conversation of requests and replies which allows files to be read and written along with other operations just as if those operations were to be performed on the local hard drive.

Many conversations can coexist on the same network, but large data transfers can result in a congested network which eventually slows down all of its users. Although the network was originally put into place to support necessary data transfer, unwise use of its bandwidth can create problems. Most commonly-used networks have a theoretical maximum throughput of ten to sixteen megabits per second, but their effective useful throughput is usually 70-85 percent of this, giving a lower bound of seven megabits per second. Translated into more useful megabyte per second terms, this is roughly one megabyte per second. So, the transfer of a 10 megabyte dataset should require no more than 10 seconds of the network's time. However, if other applications are using the network, this time increases, as do the relative time values for other network users.

In addition to the maximum throughput, the extra information required by the packet to specify various types of information adds to the total amount of data transferred. If the amount of data transferred in each packet is relatively small, say about 100 bytes, the typical packet overhead of 64 bytes becomes a significant factor. Another factor to remember is that the conversation model of requests and replies typically doubles the amount of packets from what is needed to simply transmit the data. Returning to the 100 byte per packet data example, we end up with 128 bytes of overhead, which would cause the network to transfer more than twice the amount of data, or perhaps 22 megabytes. This would take at least twice as long as we had estimated above.

Whether the data transferred across the network comprise a program or data set, time is required and the limited total bandwidth of the network is partially utilized. The key to the most effective use of the network resource for all users is to most efficiently utilize it for each application using the network.

MEASURABLE FACTORS

So that an analysis of various network, machine, and software configurations can be performed, some basic terms and measurements are defined. Many of these are standard measures, but some have been created especially for this purpose.

- Network medium - The transmission medium for information on the network. Contention schemes (like

ethernet), can communicate at any time, and tend to be inexpensive to implement. Token passing schemes (like token ring and FDDI) communicate through tokens that are passed around the network to pass information and control traffic problems. Ethernet networks usually run with a bandwidth of 10 megabits per second, while token ring networks run at either 4 megabits or 16 megabits per second. FDDI has a much higher bandwidth at 100 megabits per second.

- Network load - The amount of traffic being processed by the network. This can be a very important factor on ether net networks. Usually specified in terms of the percentage of maximum theoretical throughput, this measure will be considered in several ranges, including:
  - *Quiet* - little or no traffic present on the network. The network is being utilized at 0 to 10 percent of its maximum throughput.
  - *Active* - some useful work is being done by the network. It is being utilized at between 15 and 30 percent of its maximum.
  - *Busy* - the network is running at an optimal load, and is being utilized at 50 to 70 percent of its maximum throughput. It is in this range that an ethernet network can most efficiently service its users because maximal traffic is flowing with minimal delays.
  - *Saturated* - the network has too much traffic to effectively handle. At this point, ethernet network packets become likely to collide with each other, which requires retransmission by the sender. For ethernet networks, this occurs at utilization values above 75% of theoretical maximum and results in slowdowns. Token-based networks do not suffer this limitation, and are able to run at near their maximum bandwidth.
- Network topology - How the network is structured. A single network keeps all traffic on the same medium. More complicated layouts involve routers, which interconnect different networks. Traffic can flow across a router with minimal delay, although some network packages change their operating parameters when traffic crossing a router is detected.

  Ethernet networks can grow very large without effecting overall performance, but token ring networks get incrementally slower as more users are added since each of them partake in passing tokens around the network.
- Network traffic - The actual information that is transferred across the network. The information takes a conversational form of requests and replies, which together comprise the network traffic between two network nodes. Since the requests require extra data to transmit, this results in additional traffic in the conversation. Additionally, each packet has extra information added to it to specify its sender and destination which adds more traffic to the conversation. The amount of network traffic can be determined by adding up the sizes of each packet transferred, either by the sender or the receiver, across the network.
- Network efficiency - How efficiently data is transferred across the network. The extra data transferred due to request packets and packet identification information, while necessary to conduct a network conversation, are not specifically required by the application using the network, and can be considered overhead. Clearly, as the amount of overhead increases, the efficiency with which the network can transfer useful program data diminishes. A measure of network efficiency can be therefore defined as:

$$\textit{network efficiency} = \frac{\textit{amount of program data transferred}}{\textit{total amount of network traffic}}$$

  For example, if a program requires one megabyte of data to be read from a remote location and the resulting network traffic is two megabytes worth of network packets, the efficiency with which that data has been transferred across the network is 50%.

  In addition, an effective network efficiency can be derived, which also takes into account the effective use of the data after transmission. Citing the previous example of a one megabyte data request, if the amount of utilized data from that request is smaller than one megabyte, the effective network efficiency value decreases accordingly.
- Machine efficiency - The time spent by a machine that is waiting for the network to provide it with the requested information is essentially wasted if the machine can do no additional processing while it is waiting. Since this is typically the case with a desktop PC and network delays can be considerable, this factor needs to be measured.

  It is assumed that a machine that is accessing strictly local data (from its own hard drive) is running as quickly as it is able, taking into account factors such as disk I/O performance and CPU speed. Therefore, the time required to accomplish a given task can be used as a baseline of comparison against other configurations which can add delays.

$$\textit{machine efficiency} = \frac{\textit{time required for local operation}}{\textit{time for remote operation}}$$

  An example is a case in which a task that could be completed in one minute using data stored on the local hard drive requires three minutes to complete when the data is accessed across a network. The resulting machine efficiency in this case is 33%, since the local machine is still doing the same amount of productive work that could be done in one minute with local data, while the extra two minutes are spent waiting for the network to deliver the data.

  If the local machine is capable of multitasking, this may not be bad as it sounds if the remainder of the computers resources can be used effectively.

  Another use of this measure is to compare the relative processing power of different computers. If a program can be run on a remote computer more quickly than if run locally, an effective machine efficiency value can be derived. While not strictly relating to the local machine, a task that runs more quickly remotely can be thought of as increasing the local machine efficiency to over 100%. For example, a task requiring ten minutes to process locally that can be completed in two minutes remotely effectively increases the amount of work that can be accomplished at the local machine to 500%.

SAS SYSTEM UTILIZATION OF NETWORK RESOURCES

Network analysis tools like Novell Inc.'s Lanalyzer for Netware allow network conversation monitoring and packet capture for detailed analysis. Several factors regarding the efficient use of the network by the SAS System have been analyzed and results presented below. It is important to note that the range between inefficient and efficient use of network resources is large. Although a more complete analysis will follow, some summarized findings are:

- Running the SAS System from a network fileserver image is efficient. Bringing up the SAS System in DMS mode to a prompt involves reading up to three megabytes of program material. The conversation between the PC and fileserver can be very efficient, with average packet sizes in the range of 1100 bytes.

- Accessing catalogs and data sets from a network fileserver can be very efficient. For both reading and writing, the SAS System utilizes large packets on the order of 1100 bytes.

- The operating system is able to recognize SAS System requests for large amounts of data and capitalize on them using Novell's packet burst technology. This optimization allows data request to range up to 8192 bytes, and individual response packets up to 1400 bytes. The key optimization is that the response packets don't wait for acknowledgement until the requested amount has been completed. These features can generally optimize data transfer rates and decrease total conversation size. Packet burst technology is included with Netware versions 3.12 and above.

- Older Netware requestors cut maximum packet size to 512 bytes when a router's presence has been detected. These small data sizes magnify the packet overhead ratio and require roughly twice as many request packets to be sent.

- NETBEUI protocol packets (used with LanServer and Microsoft Windows NT Advanced Server systems) allow efficient network utilization, with average packet sizes near 1300 bytes.

The effects of network load

Packet size analysis aside, the most important factor in determining actual network performance is based on time values. Hopefully, these numbers will be useful for SAS System application designers making implementation decisions about client/server and network resource organization. Several basic operations have been performed on the SAS System for various PC operating systems in multiple client/server and network utilization designs. Changes in elapsed time among the multiple designs indicate time gains that can be attributed to network use for data transfer.

Efficiency measures have been calculated that indicate the degree to which local resources are being fully utilized. Accessing local data is defined to be 100% efficient, since the local computer can do no better than this case. The increased time required to perform the same task when using network data can be attributed to delays on the network, which increase with network load.

For the network and processing data presented here, the test machines are:

Workstation: IBM PS/2 Model 90, 33MHz 80486, Windows 3.1
SAS/CONNECT remote computer: NCR 60MHz Pentium, Windows NT
Windows for Workgroups peer server. Dell 33mhz 80486.
Netware 3.12 Fileserver: Compaq 66MHz Pentium
Netware 3.11 Fileserver. Dell 66MHz 80486 DX2
IBM LanServer. IBM PS/2 Model 95, 66MHz 80486 DX2
Windows NT Advanced Server. NCR dual 60MHz Pentium.
Network type and adaptor:10Mb/sec ethernet, 16-bit network interface card
Network analyzer: Lanalyzer for Netware

Similar data can be expected for the SAS System under OS/2 and the SAS System under Window NT operating on the same workstation PC.

A series of tests were run to show how network response varies with the network load. Table 1 shows the results in which times from several runs of each time were averaged.

Table 1. Time required to start a full-screen SAS session with the SAS System for Windows, Release 6.08 from a Netware 3.12 fileserver. All of these cases use the network layout shown in Figure 3 earlier, except as noted.

Configuration	Elapsed time (seconds)	Machine Efficiency	Network Efficiency
Run from the local drive (Figure 1)	4.0	100%	—
Run from a quiet network (0-5% utilization)	10.0	40%	92%
Run from an active network (25-30% utilization)	11.7	34%	92%
Run from a busy network (65-75% utilization)	15.7	25%	92%
Run from a saturated network (80-85% utilization)	25.3	16%	92%

As is clear from the numbers in Table 1, running the SAS System from a network image does impose a performance impact. A highly saturated network can lead to very inefficient use of the local computing resource. All other tests discussed in this paper were performed on a "quiet" network.

The actual conversational details of this test case can also be analyzed to discover the following facts: starting the SAS session required 2.6 megabytes of the SAS System image to be read across the network, in a conversation that included 2.8 megabytes of total traffic. This results in a network efficiency rating of 92%, since 92% of the total data transferred were actually used (as opposed to the overhead required to transfer the data on the network).

Further analysis shows that the average packet size value for data read from the network is 1275 bytes/packet. This is a very efficient mode of data transfer, considering that the theoretical maximum in this case is 1450 bytes/packet. Packet size efficiency is high, which helps to minimize additional network traffic problems.

One final piece of analysis is worth mentioning. While watching files being read from the network, it is easy to determine which files are required to begin a SAS System session. It is possible to use this information to determine a small subset of the SAS System which can be placed on the local hard drive to speed up session startup time. Although this feature of determining a SAS System image subset for selective installation is planned to be available in 6.10 Releases of the SAS System, this information is summarized in Table 2 for the simple case of starting a full-screen SAS session. By installing the following images on the local hard drive, network use can be eliminated during the process of start a SAS System session.

Table 2. SAS System images used to start a full-screen SAS session.

```
SAS.EXE
CONFIG.SAS
CORE/SASDLL/SABEB.DLL
CORE/SASDLL/SABXDM.DLL
CORE/SASDLL/SABXKRN.DLL
CORE/SASDLL/SABXSHL.DLL
CORE/SASDLL/SABYH.DLL
CORE/SASDLL/SASANSI.FON
CORE/SASDLL/SASHOST.DLL
CORE/SASDLL/SASVBMP.DLL
CORE/SASDLL/SASVICON.DLL
CORE/SASDLL/SASVRES.DLL
CORE/SASDLL/SASVWU.DLL
CORE/SASEXE/SABXINI.DLL
CORE/SASHELP/CORE.SC2
CORE/SASMSG/CORE.MSG
CORE/SASMSG/HOST.MSG
CORE/SASMSG/MSGDIR.MSG
```

Effects of fileserver configuration on performance

Network administrators have many parameters related to fileserver configuration that can be modified to enhance performance. Although these are complicated and difficult to test, benchmark tests similar to those performed in this paper can be used to gather information about the effects of these parameters. Some of the more important parameters to consider include:

- fileserver cache memory. The more memory a fileserver has available, the more likely that client requests can be serviced from this cache. Since the cache access happens at memory transfer speeds, this single factor can greatly effect system performance. This memory is, however, shared across all client requests, so a fileserver that services diverse requests may not be able to effectively cache for everyone's benefit.

- read-ahead operation. Some fileservers (such as Novell Netware 3.12 and above) support configurable read-ahead that allows the fileserver to populate its read cache memory if sequential file access is detected and the fileserver isn't doing anything more important. This increases the likelihood that the cache will be pre-loaded and effectively used at little cost to overall operation.

- write-behind operation. This allows the disk subsystem to delay disk writes to more efficiently write data to the disk drive. Also known as "lazy" writing, this can have a dramatic effect on systems where applications do a lot of writing to network drives. It works by organizing the order in which data are written to the disk to minimize time-consuming disk seeks. This is also a dangerous option, since power or hardware failures can cause data loss if this feature is used. It is generally used only on computers that have an uninterruptible power supply (UPS).

- dedicated fileserver operation. A computer that is acting as a fileserver in addition to other tasks has less power to offer to fileserver clients. If these other tasks are intense (such as backup operations, foreground tasks running data processing jobs, or central service tasks), performance can be severely effected.

  Some peer-type network packages (such as Windows for Workgroups) allow the relative priority of fileserver tasks to be set. Maximizing fileserver process priority is required to get maximal performance out of a fileserver.

- disk technology used in the fileserver. Newer disk drive technologies, such as fast-and-wide SCSI, allow faster data transfer rates when the server disk drives are used. Software approaches to parallelizing disk access (such as data striping and spanning) can also enhance disk system performance by decreasing effective disk wait periods.

  In addition, disk drive format types can effect performance. IBM's high performance file system (HPFS) and Microsoft's NT file system (NTFS) provide for quicker access than the standard DOS file access table (FAT) based format. An additional enhancement available on OS/2 LanServer 3.0 systems is 386 HPFS, which further optimizes disk access.

Effects of network topology on performance

The previous discussion of packets and maximum throughput assumes the most simple network model of a single network that handles all network traffic. Other configurations are used in organizational settings that involve multiple networks, and routers that send packets between them. The chief benefit of multiple networks is that the total effective bandwith increases with the number of networks. This occurs because the ten megabyte example data transfer discussed earlier can occur on a different network than other transfers, effectively multiplying total capacity.

Unfortunately, there can be drawbacks to the use of multiple networks. One such drawback is that delays can be incurred when crossing the router that interconnects the networks. Although these delays are very small, they can quickly add up to a significant performance problem when massive amounts of data are involved on a heavily loaded network. Another drawback is that some network interface software (usually called network requestor software) that allows the PC to communicate with the network actually sets an artificial limit to the maximum packet size if it detects that a router lies between itself and the destination.

To illustrate this effect, the SAS System was run from a variety of fileserver and network configurations. Table 3 shows the times required for starting a session of the SAS System under Windows on Novell Netware 3.11 and 3.12 fileservers, differences resulting from using Netware ODI requestors versions 3.26 and 3.32, and the effect of accessing a fileserver across a network router. Additionally, other types of server types are included for comparison.

Table 3. Effects of network topology on starting a SAS session. All of these comparisons assume a "quiet" network. Each of these tests use a network setup like those shown in Figure 3 earlier, except as noted.

Server Configuration	Elapsed Time (seconds)	Machine Efficiency	Network Efficiency
Run from a local drive (Figure 1)	4.0	100%	---
Run from a local Netware 3.12 server, 3.32 Requestors	10.0	40%	92%
Run from a remote Netware 3.12 server, 3.32 Requestors	10.0	40%	92%
Run from a remote Netware 3.11 server, 3.32 Requestors	13.0	31%	90%
Run from a remote Netware 3.12 server, 3.26 Requestors	13.0	31%	90%
Run from a remote Netware 3.11 server, 3.26 Requestors	13.3	30%	83%
Run from LanServer 3.0	12.8	31%	83%
Run from Windows NT Advanced Server (Figure 2)	15.0	27%	79%
Run from a Windows for Workgroups peer (Figure 2)	13.0	31%	81%

As is clear from the data in Table 3, time increases as one moves to a 3.11 fileserver (mainly as a result of maximum data transfer size decreasing to 1024 bytes). The worst case scenario involves old requestors that access a Netware 3.11 fileserver across a router. This causes the maximum data transfer size to drop down to 512 bytes. Since more packets are required for the overall conversation and each packet adds its own overhead to the conversation, overall network efficiency drops.

This decreased efficiency not only causes an increase in time required for the session user, but also generates more network traffic and would cause a greater slowdown on a busier ethernet network.

Running the SAS System on a LanServer, Windows for Workgroups peer, and Windows NT server performed well, but their lower network efficiency values could cause them to be more susceptible to slowdowns on a busy ethernet network.

Accessing SAS data sets on a network

It is also possible to access a data set from a network fileserver and use it instead of from the local hard drive. A benchmark test originally published by Maddox and Williams has been used to illustrate different ways in which remote data sets can be accessed. The SAS System program for this test is included in Appendix B. The data in Table 4 show the time differential between local and network stored data.

Table 4. SAS Dataset access for a 1000 by 100 item numeric dataset comparing the use of local data with fileserver data using a Netware 3.12 fileserver from a SAS System session under Windows. The column of local data corresponds to use as shown in Figure 1 above; data for Network Data are as shown in Figure 2.

PROC	Elapsed time (sec) Local Data	Elapsed time (sec) Network Data	Machine Efficiency	Network Traffic
MEANS	0.67	2.73	24.5%	0.91M
SORT	1.39	6.70	20.7%	1.87M
PLOT	0.91	5.23	17.4%	1.83M

These tests involve the reading of 3.3 megabytes of data from the network in addition to writing 847,000 bytes to the network (as the output of PROC SORT). While the overall conversation remains efficient (92% of all data transfer can be tracked to actual file I/O), machine efficiency (as shown in Table 4, and compared with running the same program against locally stored data) remains at about one-fifth of that using local data.

Accessing data through Remote Data Library Services

Utilizing Remote Data Library Services, the SAS System is able to provide transparent data access to data sets stored on other machines, even if they are of a different architecture. As an example, this would allow client PCs running Release 6.08 of the SAS System for Windows to access data stored on a server PC running Release 6.09 of the SAS System for Windows NT. This allows powerful server machines to share data to various types of less powerful client machines. Used in conjunction with SAS/SHARE software, this approach allows multi-user concurrent read/write operation on records of the data set being accessed. This is not possible with the earlier approaches tested, which only support multiple concurrent read operations to be performed on the data set.

Various types of computers can be used as SAS System Library Data Servers via the SAS/SHARE or SAS/CONNECT products, including PCs running OS/2 or Windows NT, as well as workstation, minicomputer, and mainframe systems that run the SAS System. This access is geared towards applications that access a single record at a time, such as would be used in data entry or record browsing. This feature is available by adding the SERVER parameter to a LIBNAME statement as follows:

```
OPTIONS COMAMID=TCP;
LIBNAME RMT SERVER=SHR1;
```

The computer acting as the data server should already have a remote connection (via SAS/CONNECT software) or have executed the following in a SAS System session:

```
OPTIONS COMAMID=TCP;
PROC SERVER ID=SHR1; RUN;
```

The types of operations performed by the standard benchmark program used in this paper are not suited to to use with Remote Data Library Services. The PROCs in this benchmark make complete passes over a data set, and PROCS SORT and PLOT scan the data multiple times. Although these services would allow the benchmark program to run, performance would suffer significantly if multiple passes over data were made in this manner.

This mode of operation is more suited to use with PROC FSBROWSE, which accesses records one at a time. Since this type of interactive operation is difficult to capture in benchmark timing results, they will not be presented here.

Accessing data with SAS/CONNECT Data Transfer Services

If batch mode processing is to be performed on a remote data set, or if data will be accessed multiple times, the data can be downloaded from the remote server to the client machine for local processing. These transfers are performed through PROC UPLOAD and PROC DOWNLOAD. This approach can be useful if the remote machine is overloaded relative to the local machine.

The standard benchmark program was executed again, utilizing downloaded and uploaded data. The results of this test are shown in Table 5.

The following statements were added to the standard test program in Appendix B for this test. TCPWNT.SCR is a standard communications script shipped with the SAS/CONNECT product.

```
OPTIONS COMAMID=TCP REMOTE=NTSRV;
SIGNON 'C:\SAS\TCPWNT.SCR';
PROC DOWNLOAD DATA=A OUT=LOCAL; RUN;

/* NORMAL BENCHMARK PROGRAM */

PROC UPLOAD DATA=LOCAL OUT=A; RUN;
```

The remote machine is assumed to already have run the following to start a SAS spawer that is able to receive SAS/CONNECT requests (The spawner program is also shipped with the SAS/CONNECT product for Windows NT and OS/2):

```
SPAWNER.EXE -C TCP -T SHR1 -U -P
```

Table 5. Transferring a dataset using SAS/CONNECT procedures and a Windows NT Advanced Server. Local SAS System sesssion running under Windows; TCP/IP communications in use. This table corresponds to running as shown in Figure 2 shown earlier for mainly remote disk resource access.

PROC	Elapsed time (seconds)	Network traffic
DOWNLOAD	15.63	0.99MB
MEANS	0.67	0
SORT	1.39	0
PLOT	0.91	0
UPLOAD	22.87	1.01MB

Overall, this approach requires 41.47 seconds to complete, and generates 2.0 megabytes of network traffic. This is more efficient that the previous case (with Remote Data Libraries), which required 77.78 seconds to complete and generated 5.95 megabytes of traffic. This works out to a time efficiency increase of 47%, and requires 66% less network traffic than the previous example.

Of course, this test program utilizes and assumes that all data will be used. If only a single record were needed, this approach would be extremely inefficient. The previous method costs only a negligible amount of time to access a single record, while this approach adds about a 38 second overhead factor. In addition, 2 megabytes of network traffic for a single record access would result in an effective network efficiency rating of far less than one percent.

However, if this data transfer can take place during off-hours, the network traffic and time factor considerations could effectively disappear. SAS/CPE® software or a Lanalyzer can facilitate in the analysis of daily network loading cycles.

Using remote computing resources with SAS/CONNECT RSUBMIT

Another feature of SAS/CONNECT software allows a client machine to remotely execute SAS System commands on a remote computer using the RSUBMIT directive. This approach involves surrounding code to be remotely executed with RSUBMIT, ENDRSUBMIT pairs. The standard test program was again modified as shown below and results shown in Table 6.

The following statements were added to the standard test program in Appendix B for this test:

```
OPTIONS COMAMID=TCP REMOTE=NTSRV;
SIGNON 'C:\SAS\TCPWNT.SCR';
RSUBMIT;  /* Begin Remote Execution */

/* NORMAL BENCHMARK PROGRAM */

ENDRSUBMIT;  /* End Remote Execution */
```

Table 6. SAS Dataset access for a 1000 by 100 item numeric dataset using remote command submission to a Windows NT Advanced Server. This approach corresponds to Figure 3 shown earlier utilizing the CPU resources of the remote server computer.

PROC	Elapsed time (sec), local CPU	Elapsed time (sec), remote CPU	Machine Efficiency	Network traffic (bytes)
MEANS	0.67	0.10	670%	1024
SORT	1.39	1.06	131%	1024
PLOT	0.91	0.17	535%	1670

Clearly, these procedures benefit from the superior computing power of the remote computer. In this case, the remote computer CPU was far more powerful than that of the local PC, although PROC SORT did not enjoy the benefit fully (largely due to the dataset write I/O performed by this procedure). Since these tests ran faster remotely than possible locally, the effective machine efficiency values exceed 100%.

Furthermore, the amount of network traffic is negligible compared to earlier examples. However, if the remotely executed program generated a large volume of output, network traffic could become an issue.

The key factor in this type of operation is the relative speed factor between the remote system (under a normal computing load) and the local PC. In our example, the remote PC's unloaded dual Pentium configuration offered up to six times the throughput of the unloaded local PC. If five other user jobs were running concurrently, however, the benefit would likely be distributed to all users with a net effect of no speedup to any particular user. If larger loads can be present on the remote compute server, it may make more sense to process locally. Even if the local CPU is slower than that of the remote system, its 100% availability may make it a better performer than a heavily loaded remote system.

DISCUSSION

Various approaches to client/server computing and data access are possible and have been explored. Each can be useful in certain configurations and situations. The flexibility of the SAS System allows system designers to design a system that utilizes data and resources efficiently. Clearly, no single approach is best for all

configurations, and application needs also direct how a system is organized.

In a perfect world, each user's desktop machine would be as powerful as the Pentium server described above, and networks are fast end empty. More realistically, each resource needs to be used as effectively as possible.

Hopefully, the simple examples described and tested above will provide some basic information for system designers considering utilization of the SAS System in a client/server type environment. The basic factors to consider and weigh include:

- where the SAS System software resides. Network installations are possible, but can effect performance in some cases. Network effects of these installations have been shown. Minimal installations are possible that minimize local disk space requirements while keeping network delays small.

- the capability of the network to provide timely access to data. Variables include maximum throughput, typical load, data transfer method, and network topology. The examples in this paper illustrate the effects of some of these variables. Network traffic and efficiency values can be used to help limit bottlenecks.

- the processing power available on the desktop computer. As desktop PCs become more powerful, this becomes an increasingly important factor. Since the full power of the SAS System can be accessed even on desktop PCs, this can be of critical importance. As processors continue to improve, limitations in disk storage space become more critical. Offloading jobs from an overloaded central resource can get the job done quicker in some cases and help lower loads on central resources in general.

- the processing power available on the remote machine. While typically more powerful than the desktop computer, heavy user loads on this resource can diminish its usefulness. Relative computing factors can be computed as illustrated in an earlier example. Remote computing availability is an easy way to increase performance for power users.

- reliance on central services. Mainframe and other centralized computing resources typically have access to high quality system support and backup facilities that may be required in many cases. SAS/CONNECT Remote Library Services and SAS/SHARE software allow critical data to remain in the safest location.

- interoperability with other applications. SAS/ACCESS products, in addition to ODBC Pass-Through support add flexibility for system designers. This means that data can be access or processed in the most efficient manner.

- fileserver configuration parameters. Any configuration changes to central resources can lead to gains for all of its users. These factors control how efficiently the fileserver can service clients.

CONCLUSION

Efficient network and processing resource utilization is a complicated problem. Hopefully, the efficiency measures and relative time/network utilization statistics discussed in this paper will assist designers in effectively integrating the SAS System into enterprise computing solutions. As the numerous processing and data access configurations have shown, the SAS System provides for a wide range of system design options that should be able to meet just about any system requirement.

Because of the wide range of variables in areas such as: application design, network utilization, fileserver configuration, network topology, network type, remote data access, remote computing access, and desktop PC speed and disk size, your mileage can and *will* vary. As a result, this paper cannot specify the single, best configuration for all needs. Instead, it is only able to cover the wide range of options available to users of the SAS System on PCs. Many of these options effect performance, while others expand the range of tasks that can be performed. Simple measures, like machine efficiency, indicate whether these options are successfully integrated into a completed system.

The desktop PC can play in important role in your enterprise-wide computing system and increase productivity if effectively utilized. Utilization factors and measures discussed in this paper can enable knowledgeable network administrators and system designers as well as enlightened power users to determine optimal uses of computer and network hardware. These can lead to more optimal use ot that most vital resource, time.

REFERENCES

Barnhart, Andy (1994), "Sharing Enterprise Data with ODBC," *Proceedings of the Nineteenth Annual SAS Users Group International Conference*, 19.

Currid, Cheryl and Company (1993), *Novell's Guide to NetWare 3.12 Networks*, San Jose, CA; Novell Press.

Currid, Cheryl and Company (1993), *Novell's Guide to NetWare 4.01 Networks*, San Jose, CA; Novell Press.

Clifford, William D. (1992), "Database Features Exend the Scope of SAS/SHARE Software," *Proceedings of the Seventeenth Annual SAS Users Group International Conference*, 17, 335-340.

Creech, Stephanie and Van Wyk, Jana, (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.

Garner, Cheryl (1994), "The SAS System: A Complete Client/Server Solution," *Proceedings of the Nineteenth Annual SAS Users Group International Conference*, 19.

International Business Machines Corporation (1991), *Operating System/2 Local Area Network Server - Network Administrator Reference Volume 2: Performance Tuning*, Austin TX; IBM Corp.

"Introducing Netware 3.12," (1993), *Inside NetWare*, December 1993, 1-4.

Microsoft Corporation (1993), *Microsoft Windows NT Advanced Server System Guide*, Seattle WA; Microsoft Corp.

Microsoft Corporation (1993), *Microsoft Workgroup Add-on for Windows User's Guide*, Seattle WA; Microsoft Corp.

"Netware 4.0: Peformance Tuning and Optimization," (1993), *NetWare Application Notes*, May 1993, 1-13.

"New Distributed Computing Capability in SAS/CONNECT and SAS/SHARE Software Enables Transparent Access to Remote Data," (1993), *SAS Communications*, 19, 8-10.

Pallay, Karyn (1993), "Rightsizing: Time Waits For No Technology," *NetWare Connection*, May 1993, 10-16.

SAS Institute Inc. (1993), *SAS/ACCESS Interface to ODBC: SQL Procedure Pass-Through Facility*, Cary NC; SAS Institute Inc.

SAS Institute Inc. (1993), *SAS/CONNECT Software: Remote Library Services*, Cary NC; SAS Institute Inc.

SAS Institute Inc. (1993), *SAS/SHARE Software: Cross Architecture Access*, Cary NC; SAS Institute Inc.

SAS Institute Inc. (1992), *SAS Technical Report P-224 - SAS/CONNECT Software: Changes and Enhancements, Release 6.08*, Cary NC; SAS Institute Inc.

"SAS Institute Offers Interface to ODBC," (1993), *SAS Communications*, 19, 13.

"The Most Complete Solution: SAS System Provides Support for Multiple Client/Server Models," (1993), *SAS Communications*, 19, 13-17.

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.

Williams, C., and Maddox, J. (1991), "Optimizing the Performance of the SAS System under OS/2," *Proceedings of the Sixteenth Annual SAS Users Group International Conference*, 16, 476-480.

APPENDIX A - PC NETWORKS SUPPORTED BY THE SAS SYSTEM

The SAS System does not utilize any network-specific features other than the basic support required by the operating system. Therefore, if a network software package is compliant with DOS and Windows, it is "supported" by the SAS System under Windows. This does not imply a recommendation for any of the networks listed below, that this a complete listing, or that all networks are equally easy to install or maintain.

The PC network software packages supported by the SAS System include:

- Novell Netware (versions 2.2, 3.10, 3.11, 3.12, 4.x)
- Lantastic
- Banyan VINES
- Netware Lite
- IBM OS/2 LanServer (1.1+)
- IBM OS/2 LanServer Advanced (3.x)
- Microsoft Lan Manager
- Microsoft Windows NT Advanced Server
- Microsoft Windows for Workgroups
- DEC PathWorks
- PC-NFS

APPENDIX B - PROGRAM USED TO TEST SYSTEM AND NETWORK PERFORMANCE

The basic benchmark program is as follows:

```
/* Simple data step, creates file with */
/* 100 numeric variables, 1000 obs */
data;
array num[100];
do i=1 to 1000;
  do j= 1 to 100;
    num[j]=round(ranuni(j),.001);
  end;
  output;
end;
run;

/* Proc means on 2 variables */
proc means;
var num40 num88;
run;

/* Proc Sort by 2 variables */
proc sort;
by num33 descending num55;
run;

proc plot;
plot num72*num72;
run;
quit;
```