Maximizing Computer Resources with Cooperative Processing and Distributed Data Access

Jack Wallace
SAS Institute Inc., Cary, NC

ABSTRACT

Cooperative processing and distributed data access are emerging computer technologies of the 90s that have the potential to revolutionize the use of computer resources. Many systems' implementations of these technologies are also revolutionary, requiring complete application rewrites to use the new technology.

SAS Institute has chosen an implementation that allows users to practically apply these technologies to achieve business goals. This approach allows users to preserve their investment in existing applications (whether they are based on the SAS System or not) while integrating the applications with other existing or new applications across multiple computing environments. Applications written with the SAS System may be developed to allow execution of each discrete task in an environment that is most appropriate for that task. Together, these abilities allow users to make effective use of all the computing resources in their organization.

This paper is an overview of the SAS System's current implementation of cooperative processing and distributed data access. It presents a case study of a hypothetical company with disjointed business systems typical of the 80s which ran in two heterogeneous computing environments. The company has chosen to extend these business systems to the workstation environment and integrate them with the SAS System. The business systems and extensions discussed in this paper are also the basis of demonstrations of SAS/SHARE and SAS/CONNECT software in Formal Online Demonstrations at the conference.

Finally, the paper looks at the continuing evolution of these technologies planned in SAS/CONNECT, SAS/ACCESS, and SAS/SHARE software.

INTRODUCTION

Traditionally, we think of hardware computer resources when we speak of performance or maximizing usage. Hardware resources include CPUs, external storage, and communications networks. Usually, the primary CPU resource of concern is processor cycles or time, but often memory consumption is also critical. In communications networks we have to consider the available bandwidth of the communications media as well as the capacity of communications controllers, gateways, routers, and other network components. Access to specialized hardware resources, such as vector processors or plotters attached to particular processors, is also important to many businesses.

Software computer resources may be overlooked in analyses of resource usage and capacity planning. Systems software, business applications systems, and data are all software resources. Systems software includes operating systems and development tools. Business applications systems are written using development tools to run with particular operating systems on particular hardware. They implement the functions you depend on in your daily business activities and create and manage your business data. Your investment in systems software, business applications systems, and data may exceed your investment in hardware resources.

Cooperative processing is a general term for data processing that utilizes resources attached to more than one CPU to accomplish a task. Various models for cooperative processing are discussed in greater detail below. Distributed data access is one model of cooperative processing, describing the process of accessing data that are distributed across external storage attached to more than one CPU.

Maximizing computer resources with cooperative processing addresses both hardware and software resources utilized in processes executing on more than one CPU.

CONCEPTUAL MODELS OF COOPERATIVE PROCESSING

Most application programs include the following three functional elements or segments:

- Presentation / User Interface
- Application Logic / Analysis
- Data Access.

The models or types of cooperative processing differ according to the distribution of these functional elements of the application between the processors that are cooperating in the execution of the application. IBM describes the following three major models of cooperative processing in a Systems Application Architecture (SAA) environment.

- In distributed presentation, the presentation or user interface function executes on the front-end processor (local session), while the application logic and data access executes on the back-end processor (remote session).
- In distributed function, the application logic functions of the application are split between the front-end and back-end processors while presentation functions remain on the front-end and data access functions remain on the back-end.
- In distributed data access, presentation and application logic functions execute on the front-end while data access executes on the back-end.

The Gartner Group describes five models of cooperative...
processing depicted in the following diagram. They reserve the distributed presentation and distributed data access terms for applications that split presentation and data access among the processors. New terms of remote data access and remote presentation are substituted for the models in which the split occurs between data access and application logic and between application logic and presentation.

![Diagram of processing models]

MAXIMIZE — WHAT AND HOW?

Deciding what resources to maximize through these cooperative processing techniques is highly dependent on your current resources and their associated costs, your commitments for near term acquisitions, and your perceptions of the impact future technological developments will have on resources and costs. Some generalities about the impact of utilizing the various cooperative processing models on computer resources may be stated although exceptions do exist.

Distributed presentation techniques are useful for preserving your current application investment but may not provide much relief for a constrained back-end processor. Simply replacing non-programmable terminals with intelligent workstations to offload the presentation processing is likely to increase your hardware costs without eliminating much workload on a mainframe.

Distributed data access may allow access to your application system databases on the back-end processor while allowing you to offload the application logic execution to the intelligent workstation. However, such techniques usually require you to rewrite applications for the workstation and may have a high communications cost if many data must be accessed across the network.

Well-designed distributed function applications may be written to minimize both data transfer across a network and workload on the back-end processor. However, there are few toolsets for creating these applications so most are written using third generation languages.

Many toolsets used to develop cooperative processing applications implement one particular model of cooperative processing. While some vendors have multiple toolsets to utilize the different models, you are usually required to rewrite your application from one toolset to another if your environment and needs change in the future.

To maximize your computer resource investment, you need a toolset or development environment that:

- provides access to your current business system data, allowing you to enhance and extend your existing application systems.
- allows you to use any combination of the cooperative processing models and switch amongst them with adjustments to application code rather than entire application rewrites.
- is portable to your current hardware environments and that you expect to be available on future hardware platforms as hardware technology evolves.
- provides state-of-the-art procedures and methods for data analysis and information delivery that allow you to quickly build new applications to meet your evolving business needs.

TECHNOLOGY FOR DATA ACCESS AND COOPERATIVE PROCESSING TODAY

The SAS System provides a toolset or development environment to help you maximize all your resources. Of particular interest are the technologies for data access and cooperative processing.

**Data Base Management Systems and 3GL Programs**

Most database management systems (DBMSs) are themselves a form of cooperative processing. They generally implement a server model in which an application program calls DBMS functions in its process, the DBMS functions communicate the request to the server process for execution, and the results are returned to the DBMS functions in the application process and on to the application.

![Diagram of DBMS application structure]

Historically, DBMS implementations required applications programs to run on the same processor as the server. Recently, DBMSs have been implemented for Local Area Networks (LANs) and UNIX networks, allowing servers and application programs to execute on separate networked processors. A few of these support or plan to support access between processors with different machine architectures and operating systems.

A true DBMS provides integrated data recovery facilities that guarantee that updated records will not be lost once a transaction
is completed and that no updates from a partial transaction will remain in the database. These facilities are necessary for applications containing critical data and/or no other paper or computer record of transactions that could be used to reapply lost transactions.

Providing recovery facilities can be an expensive use of hardware resources. Since many applications do not require these facilities, database systems that do not implement them have been created. Note the use of the term database system to refer to these systems that do not implement all functions of a DBMS.

Many of today's operational business systems rely on third generation language (3GL) programs accessing DBMSs. 3GL programs are frequently more efficient in processor cycles than corresponding 4GL programs because they are coded for a specific task and use compiled and linked code. Application programs that must support high transaction rates or volume are usually written in 3GLs for this increased efficiency.

Businesses that insist on using 3GL programs for all applications that access a DBMS generally experience large application backlogs, high program maintenance costs, and outdated business systems that have not been extended to meet new business objectives.

Accessing DBMS Data from SAS Applications
SAS/ACCESS software provides access to data in major database management systems from the SAS System. SAS procedures and DATA steps written to access SAS data files can directly access data in a DBMS also. A SAS/ACCESS engine receives the requests to access a SAS data set and translates the requests into the appropriate DBMS function calls to access the DBMS database.

Since SAS/ACCESS software uses the normal DBMS functions to access DBMS databases, SAS applications can share access to your data bases with your current business systems. All the security and integrity features of the DBMS are preserved. In addition, distributed data access extensions to the DBMS are either available automatically or can be supported in the next release of the SAS/ACCESS software.

SAS/ACCESS software allows you to use the power and flexibility of the analysis, presentation, and application development tools in the remainder of the SAS System as you

4GL solution for DBMS data. This combination allows you to reduce your DBMS application backlog, reduce your maintenance costs associated with the applications, and extend the functionality of existing DBMS business systems to meet new requirements.

This does not mean that the SAS System and DBMS combination is the best solution for all your application and data needs:

- DBMSs frequently decrease efficiency because of the rollback and recovery facilities that they provide.
- DBMSs are not the most efficient data access and storage technologies for SAS System processing; frequently, data stored in a DBMS must go through one or more format conversions in the SAS/ACCESS engine.
- Many DBMSs cannot support all the SAS data set direct access capabilities so some limitations may be placed on the SAS programs used to access DBMS data that do not exist when accessing SAS data files.

For these reasons, SAS applications that do not have external requirements to store data in a DBMS should use SAS data files for data storage. SAS users who perform multiple analyses of data in a DBMS should consider copying the necessary subset of data to SAS data files unless current data are required for each analysis. An information database containing summary data of your operational business systems is best implemented in SAS databases if the majority of the data analysis will be using the SAS System.

Sharing Update Access to SAS Data by Multiple Users
SAS/SHARE software provides concurrent update access to SAS data libraries and SAS files by multiple users. SAS/SHARE software is in production under MVS and CMS systems. An experimental version is in limited availability under VMS systems. Similar limited availability experimental releases are planned for OS/2® and UNIX platforms.

SAS applications share update access to SAS files through a SAS/SHARE server that is similar to a DBMS server. The SAS procedures and DATA steps pass requests to access SAS data sets to the SAS/SHARE remote engine. The remote engine communicates the request to the SAS/SHARE server which dispatches a SAS task associated with the user’s task and passes the request to an engine to access the data. The engine must support locking between multiple SAS tasks to prevent multiple users from trying to update the same observation at the same time.
SAS/SHARE software supports some distributed data access in Release 6.07. In fact, as early as Release 5.18, SAS/SHARE software supported access to a MVS SAS/SHARE server from a user's session on another MVS machine if the machines were connected in a VTAM* network. Release 6.07 provides experimental support for access to a MVS server from a CMS user connected in a VTAM network. Release 6.07 provides data sets on MVS by CMS users and CMS SAS data sets by MVS users. In addition, users may access DBMSs on the other system, as depicted below. Such access is currently restricted to read-only access since most DBMSs think the SAS/SHARE server is a single user instead of multiple users' SAS tasks.

In Release 6.07, you should consider the SAS System for complete implementation of a broader range of concurrent update applications.

Utilizing All Computer Resources with SAS Software

While the DBMS and SAS/SHARE server models implement the distributed data access model of cooperative processing, the SAS System supports the distributed function and distributed presentation models as well. Since distributed data access models may put a high load on your communications network, you may want to execute your application logic and/or presentation functions on the machine where the data resides. This flexibility can choose where to execute your application logic and presentation functions, which is a key to optimizing your resources.

SAS/CONNECT software is a unique system allowing a single local SAS session to create and control additional remote SAS sessions on other systems. SAS applications on the local SAS session can then execute SAS programs on any of the sessions.

In the diagram above, a SAS Application executes on three processors. The primary local session is running on the workstation in the center of the diagram. In the local session, SAS/CONNECT SIGNON statements create a remote session on MVS and VMS systems. These remote sessions are now available for use by the application.

The application depicted is composed of four cooperative parts. It is using different cooperative processing models to distribute the total application function among the resources available. In the first, a SAS program of procedures and/or DATA steps are remote submitted to MVS to execute program logic where the data reside. In the second, UPLOAD or DOWNLOAD procedures are remote submitted to VMS to transfer data between the workstation and the VAX system. In the third part, SAS program logic executes on the local workstation session, possibly analyzing data transferred from the previous part. In the final part, graphics procedures are executed on the MVS session using distributed presentation to display their results on the workstation.

Users executing the SAS application are unlikely to realize they are utilizing resources on diverse systems through multiple SAS sessions. The SAS application programmer is aware of the multiple environments and can take advantage of the resources.
available to each environment. Of course, the portability of the SAS System and language allows the SAS programmer to work with one language on all the environments. The application programs and/or data may be moved easily from one system to another to optimize resource usage. Frequently, such changes can be made with modifications to one or two statements in the application system.

Readers familiar with the origins of SAS/CONNECT software in the micro-to-host facility of Release 6.03 of the SAS System for PC-DOS may still think of it as a tool for connecting PCs to mainframes or minicomputers. In Release 6.07, SAS/CONNECT software adds support for the following advanced peer-to-peer communications protocols:

- APPC/LU6.2
- TCP/IP
- DECnet.

These communications protocols are generally more efficient for transferring binary data than are the terminal emulation-based protocols such as BHLLAPI, ASYNC, TELNET, and CTERM.

The local or controlling session of the application does not have to run on a PC or workstation with peer-to-peer communications protocols. Users with interactive access to SAS software on mainframes and minicomputers may choose to create a remote session on another mainframe, minicomputer, or PC to obtain access to the resources attached to those machines. More importantly, you may choose to centrally coordinate and automate distribution of data or reports to machines in your network from batch jobs running on mainframes or minicomputers.

 UTILIZING THE SAS SYSTEM IN A CAR RENTAL BUSINESS (1985-1992)

The hypothetical car rental business, Star Autos, is a franchised business. The headquarters for the franchisor, Star Autos, Inc., are in Chicago, IL. Star Autos, Inc. provides vehicle purchasing, marketing, and computer services to franchisees throughout the world.

Among the application systems provided by Star Autos, Inc. are a central reservation system, a contract entry system, and a vehicle records system. The reservation system manages customer reservations and inquiries about availability and pricing for rentals. The contract system manages the actual rental contract data when the car is picked up and returned to rental offices. Some information for the contract record is acquired from the reservation system if a reservation was made. The vehicle records system maintains information concerning purchase, resale, accident involvement, and maintenance for each vehicle.

The evolution of these systems from 1985 through 1992 shows how the SAS System has been utilized to enhance and expand existing applications and create new ones.

Computer Services in 1985

Star Autos had implemented both the reservation system and the contract systems as Information Management System (IMS) applications running on an IBM mainframe and accessed by 3270 terminals.

The Data Center and Franchised Office Hardware diagrams illustrate the setup of the systems:

- The IMS/VS Data Base and Data Communications System was chosen for the reservation system because:
  - It needs to process a high volume of reservation transactions.
  - Data recovery of reservations when system failures occur is considered critical for customer satisfaction.
  - The reservation system needs to be available 24 hours a day, 7 days a week.

Data recovery and system availability are not motivating factors for the contract system because a three part paper contract, created and signed during car pickup, can be used to complete the contract with the customer when the car is returned. The paper contract is a source for reentering data if the system failed and data are lost.

Still, the contract system was developed as an IMS application because:

- It had a medium high volume of contract transactions.
- It could obtain data from the reservation system.
- The company had invested in IMS and training of programmers in IMS.

The franchisee owns and maintains cars and local offices, purchasing the franchise name and services from Star Autos, Inc.
The vehicle records system did not exist in 1985. It was requested by the franchisees, but was part of a large backlog of application systems waiting for programming staff resources to implement.

**Expansion of SAS Software (1986-1989)**

The marketing department determined in 1986 that it could no longer depend on the corporate data center to meet its demands for market analysis and forecasting. The analysts were using the SAS System for data analysis and forecasting on the IBM mainframe. They had been restricted to batch job execution by the data center staff who feared that interactive use of the SAS System would interfere with the throughput of the IMS online transaction system. The data center staff members were using the SAS System for Computer Performance Evaluation (CPE) and Capacity Planning (CP) purposes. They assumed interactive use of SAS meant moving similar workloads from batch execution to online execution.

The market analysts wished to use the full-screen interactive environment in SAS software to build data-entry and analysis systems for customer satisfaction surveys and competitor data analysis systems. They justified purchasing a departmental VAX system to use with the interactive SAS System for marketing data-processing needs. New full-screen applications were written utilizing SAS/FSI® and SAS/AF® software on the VAX system.

Some marketing systems still needed weekly access to summary data from the IMS applications’ databases. Prior to purchase of the VAX system, weekly batch jobs using SAS/IMS-DL/® software would extract the summary data and put them in a SAS data set for the marketing systems’ use. These jobs were retained with an additional step to copy the SAS data sets to transport format on tape, which was transferred to the VAX system.

By 1987, the data center was approaching capacity workload on its IBM mainframe. The backlog of application development requests was growing to a critical stage. Based on the success of the marketing department’s systems, the corporate management decided to create a backlog reduction task force to investigate further use of SAS software for applications development.

The task force studied the vehicle records system and decided it was an appropriate candidate for a 1987 SAS System implementation because:

- A high volume of transactions per second was not anticipated.
- A paper trail of update transactions existed so that some transactions could be reentered in the rare case of system failure.
- Update transactions were simple in-place updates or additions of vehicle records.

The system did however require concurrent update access to the vehicle record database by the franchised offices. Since the franchisees were using 3270 terminals to access the IBM mainframe and the SAS/SHARE software for concurrent update access was not available for VAX systems, a decision was made to upgrade the IBM mainframe. The upgrade provided extra capacity for the existing online IMS system and for SAS System applications written to relieve the application backlog.

---

**The Integration of PCs in Franchised Offices (1990)**

In the late 1980s, franchisees began purchasing PCs instead of IBM 3270 terminals. They were purchasing DOS packaged systems for accounting and marketing and using 3270 emulation packages for accessing the franchised systems on the IBM mainframe.
By 1990, most franchisees owned one or more PCs. They wanted to expand their local systems to forecast and do market analysis on a local level and still utilize some of the corporate data available on the mainframe. They also complained that customer satisfaction surveys indicated long waits to pick up a car. When they investigated these complaints, they found that most were made during times when the mainframe contract system was unavailable.

The data center created another task force to study these issues and form a corporate strategy for integrating the franchisees' PCs with the corporate systems. The data center records showed that the IMS mainframe system was available 98 percent of the year. Further study results indicated that network outages were reducing the availability of the mainframe systems to individual offices to 93 percent of the year and that these outages often occurred during peak office hours. The data center was also worried about the variety of requests they were receiving for different PC-to-mainframe connection packages from the franchised offices.

The task force was chartered to:

- determine if customer service transactions could be distributed to the franchisees' PCs;
- recommend a standard solution for accessing corporate data from the franchisees' PCs.

The task force determined that the reservation system did not require distribution to the offices. The majority of reservation transactions were actually entered from the central reservations department in the corporate office, which handles the toll-free phone number for reservations.

The contract system study showed that:

- A contract record was generally updated only twice - created when the car was picked up by the customer, updated when the customer returned the car and deleted when moved to an archive file.
- Read access to the contract was normally limited to a few programs gathering summary statistics.
- 85 percent of the contracts involved pickup and return at the same office, 10 percent involved plans to drop off the vehicle at another office in the same city, 4.8 percent of the contracts included plans to drop off the vehicle in another city, and only 0.2 percent of the contracts involved plans to drop off an unplanned office.
- 70 percent of the rentals were reserved, and only 10 percent of the reservations were made the day of the rental.

The task force concluded that the contract system could be distributed to the franchisees' PCs by a system that copied reservation records to the appropriate office and copied the contract record to the expected drop-off office. Members recommended that records be copied in nonpeak hours.

Investigations of franchisee requirements for corporate data access indicated that most franchisees wanted daily access to the corporate IMS data systems and occasional on demand access to the marketing survey data on the VAX system. The task force hoped to find a solution to utilize the existing network connections to the IBM mainframe. Access to the marketing system data would be either to a copy updated daily on the IBM system or through dial-up access to the VAX machine.

Because many of the franchisees wanted forecasting and analysis capabilities equivalent to those of the marketing department on the VAX system, the task force asked SAS Institute to propose a solution for corporate data access. The data center was running Release 5.18 of the SAS System because of SAS/SHARE and SAS/IMS-DL/ applications. The marketing department had upgraded its SAS System for VAX to Release 6.06. Since the micro-to-host facility of the SAS System for PC DOS could provide the necessary connections to both SAS Systems, the recommendation was easy.

The task force decided to complete a trial evaluation of Release 6.04 of the SAS System for PCs. The primary purpose of the evaluation period was to determine if this release of SAS software provided the forecasting, analysis, and data access capabilities needed by the franchise offices. The trial period proved to the task force that the SAS System for PCs provided the functionality required. Moreover, they found that:

- The micro-to-host link would allow franchised offices to utilize the compute power of the corporate machines for large analyses as well as selectively transfer data between the local and corporate offices.
- Version 6 of the SAS System provided substantial application development enhancements through Screen Control Language.

The task force decided to recommend the SAS System for PC application development and data center connectivity. It also decided to explore further use of the SAS System for applications development in the data center.


Although SAS/SHARE software was not production in Release 6.06, the data center did serve as a beta test site. Employees explored the advantages of using the Release 6.06 enhancements in a shared environment. They wanted to use the SAS System for applications in their backlog that had not been appropriate for implementations in Version 5 of the SAS System. Indexing and
WHERE clause support allowed them to consider much larger data applications since searches did not require sequential processing of all records. Applications requiring update transactions to reference and update multiple records in multiple SAS data sets were now feasible through the SCL enhancements.

They also used the new features to improve the vehicle records system that was implemented with SAS/SHARE software in Version 5. The database in Version 5 was a large single-observation record with a limit to the number of maintenance activities it could carry in its history. The new features allowed the database to be redesigned according to relational theory into third normal form in three SAS data sets. (This redesign is depicted with separate disks for the vehicle records database in the preceding figure.) Indexes on the vehicle numbers and franchisees were useful in speeding the search for records in the data-entry applications.

When Release 6.07 included SAS/SHARE software in production status, the data center had already converted its Version 5 applications to Version 6 applications and written new ones. In addition to the head start obtained by participating in the beta test program, the data center was able to help the Institute staff find and debug a problem encountered by one of its applications, ensuring the applications would work with Release 6.07.

**Distributing the Contracts System to Franchisees' PCs (1992)**

The PC integration task force followed up its recommendation to purchase Release 6.04 of the SAS System for PC's in 1990 with an evaluation of that system as a basis for distributing the contracts database to the franchisees. They built a prototype system that:

1. distributed the contracts database to the PCs of the franchisee where the car was picked up.
2. updated a local copy of upcoming reservations for the office by:
   - signing on to the data center system,
   - extracting from the IMS database reservation records that had been updated since the last extract for the office,
   - downloading the records to the PC system where they update the local reservation file copy.
3. provided a contract transaction in SCL for customer pickup that:
   - initializes a contract record from a reservation record if it exists either in the local reservation file or from the data center IMS database (if the reservation has not been copied to the local system),
   - sends an electronic copy of the contract to the system at the expected drop-off office if it is different from the rental office.
4. provided a car return transaction to complete the contract record by:
   - sending a summary record to the data center,
   - sending an electronic copy back to the originating office location if it is different from the return office.

The system worked but was dependent on the PC user to update the reservation copy and obtain contract records from other offices each night. The data center was not comfortable with the lack of central control over these processes. If a PC failed to connect during the night, no one would know until the office opened the next morning.

The data center also believed that PC DOS was not an operating system robust enough to rely on for such critical systems. Consultations with SAS Institute staff determined that the OS/2 system was a much better environment than PC DOS for integrating and automating applications. The peer-to-peer communications support in Release 6.07 includes APPC/LU6.2 communications between the PC and the mainframe and allows a mainframe SAS System to initiate a remote SAS session on the OS/2 PC. With this feature it is possible to automate the transfer of reservation updates and contract records between offices from a batch job executing at the data center. The job log indicates any communication failures so the data center can address the problems before the franchised office opened in the morning.

The data center did very little work to modify the prototype application to run the extracts as a batch job. The basic SAS code was ported to the mainframe. Minor adjustments were made because the data were now extracted by the local session and uploaded to the remote session. The code was enhanced to work for a list of franchised offices at one time. (The data center plans to put this in production as soon as Release 6.07 is available on the OS/2 system.)

In the meantime, Release 6.07 under MVS and VMS operating systems support peer-to-peer communication between these environments with TCP/IP communications. The data center has implemented this connection with SAS/CONNECT software to replace the tape data transfer they were still using to move data between the marketing department systems and the IMS reservation system. This communication also allows access to the VMS system from the franchised offices through the network connections to the IBM mainframe. Dial-up access to the VAX system has been removed.
FUTURE DIRECTIONS FOR SAS COOPERATIVE PROCESSING PRODUCTS

The hypothetical business study above shows how the SAS System has evolved in the areas of data access and cooperative processing over the past few years. The Institute is actively working to enhance products in these areas. Unless explicitly stated, the features described below are being developed in the Research and Development tracks and will not be available until a future major release of the SAS System.

SAS/ACCESS software directions

SAS/ACCESS software will continue to be enhanced by adding support for additional DBMSs according to customer demand. Support for some additional DBMSs is may be available prior to the next major release. Contact your SAS marketing representative for current plans.

The SAS data model is being enhanced to support longer character variable names and SAS data set names. The new support will supply a syntax allowing use of national characters and blanks in quoted names.

Supporting longer data set names and variable names in the SAS language allows you to use the actual DBMS variable names instead of mapping them to eight character names in SAS/ACCESS views.

SAS/SHARE software directions

Plans include making SAS/SHARE software available on all true multitasking environments. The limited experimental release under the VMS system in Release 6.07 and those planned under the OS/2 and UNIX systems are SAS Institute's first releases under environments other than IBM mainframes. The limited experimental releases have restricted availability to a small number of customers willing to work with technical support to help evaluate and isolate problems in the new environments. If you are interested in participating in the experimental programs, contact your SAS marketing representative.

The Institute is also working to add cross-machine architecture support to the SAS data set access routines in the remote engine and server. This feature will allow users on one machine and operating system to access SAS data sets through a server on a machine with a different data architecture. For example, an OS/2 user might access a SAS data set through a SAS/SHARE server under MVS.

SAS/SHARE software will be further enhanced by adding future database features to the SAS System in this product. Under development at this time are integrated data integrity constraints and an audit trail that can be optionally used for roll-forward recovery of audited SAS data sets. For more information on these features, refer to the paper "Database Features Extend the Scope of SAS/SHARE Software" by William Clifford.

SAS/CONNECT software directions

The work to support cross-machine access to data through the remote engine and a server is also planned for inclusion in SAS/CONNECT software. The remote engine supplied with SAS/CONNECT software will be able to access a SAS data set either through a multiuser SAS/SHARE server or a limited single-user server running in a SAS/CONNECT remote session.

The Institute is also developing support for asynchronous (background processing) capabilities in the remote submission function of SAS/CONNECT software. The asynchronous support allows SAS applications to overlap program processing on multiple remote sessions with continued local processing.

NOTE: Many users have found the capability of running multiple SAS sessions in an OS/2 environment useful for overlapping remote data transfer operations with continued execution of other SAS sessions. The asynchronous feature of the R SUBMIT command will allow you to do that from one local SAS session.

CONCLUSION

The SAS System provides a rich toolset or environment for developing cooperative processing applications that allows you to maximize use of your computer resources today. The evolution of SAS software and the directions for further enhancements are indicative of the Institute's commitment to provide state-of-the-art technologies to its customers in a timely fashion. The power and portability of the SAS System guarantees that your investment in SAS software applications will be preserved as new hardware and software technologies evolve in the future.

REFERENCES


ACKNOWLEDGEMENTS

I would like to thank Sally Langthorn, John McIntyre, Cheryl Garner, and Bill Brideson for helping hypothesize the car rental business described in this paper and implementing pieces of the systems for use in the formal demonstration sessions.

SAS, SAS/ACCESS, SAS/AF, SAS/FSP, SAS/SHARE, SAS/IMS-DL/I, and SAS/CONNECT are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. * indicates USA registration.

IBM, OS/2, SAA, Systems Application Architecture, and VTAM are registered trademarks or trademarks of International Business Machines Corporation.

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

650