Implementing Client/Server Applications in a Heterogeneous Platform Environment

John McIntyre, SAS Institute Inc.
Dan Squillace, SAS Institute Inc.

Abstract

While client/server computing suggests to the casual observer a configuration of a personal computer connected to a "server" machine, to the system implementer it can be much more. Few enterprises find themselves in the position of designing their hardware architectures from "scratch". Rather, they more often are faced with implementing a distributed environment across a landscape of varied hardware and communication platforms. Selecting the proper platform for the execution of the logical segments of an application means knowing about the relative strengths and weaknesses of those platforms. This paper will review the requirements of the logical segments of applications, and compare those to the strengths of a variety of client and host based systems.

Introduction

Questions about client/server technology dominate almost all conversations within the information processing world. It is true for application developers using SAS System tools, as well as for many other participants within enterprises who might not consider themselves part of the their information systems organization.

In this paper, we will take a look at three topics related to the implementation of client/server applications. The first topic is a framework for coordinating the efforts of all of the knowledge workers who have a stake in the successful implementation of distributed computing environments. The second topic will address issues related to popular platforms found in modern enterprises, and how they, too, can be coordinated to yield the best results for all knowledge workers. And, finally, we will review some new facilities in SAS software for building enterprise client/server applications.

Section 1: Client/server for Everyone

All knowledge workers in all enterprises are members of information systems. There are few useful distinctions to be made between "end-users", application developers and large system managers. All of these enterprise participants either collect, massage, deliver or consume information. It is only where in a continuum they participate. Some will participate at numerous stages of the process, others only at discrete steps. However, all workers in an enterprise who come in contact with information should be legitimately considered part of the information systems organization.

Does this mean that everyone in this organization will have common concerns? Certainly not. They will all be part of the fundamental goal of the enterprise, which is to provide the best product or service possible at the most efficient cost. However, depending on where these workers contribute along the information systems continuum, their most visible concerns will be different. In the case of client/server computing, the concerns of data center managers begin with the prospect of implementing distributed applications as a way of avoiding costly mainframe upgrades. End-users, on the other hand, see client/server computing as a way to gain access to corporate managed data. On the surface, these views do not seem to have much in common. However, upon closer examination, it can be shown that there is a common goal set that both "groups" share, and the resolution of the concerns of these "groups" can establish a new environment in which client/server application development can proceed cooperatively.

Client/server. View from the data center.

What does client/server computing have to offer data center managers? The most obvious attraction is the promise of relocating or "off-loading" expensive processing to more cost effective platforms, thus delaying or even avoiding costly upgrades to corporate mainframes. There is no magic to the cost formulas which show that mid-range and workstation computers are cheaper ways to purchase computing power. There is plenty of discussion underway about whether these cost formulas accurately reflect the actual cost of computing, which we are not going to discuss in greate detail here. Even with the sparse data available from enterprises who have actually implemented client/server applications, it seems clear that, when all costs are considered, there are no clear savings. Suffice it to say that, taken in a narrow range of comparing the costs of operating mainframe versus workstation systems and the computing power they offer, there is much to recommend newer technology purchases over upgrades to mainframes, or as a strategy for removing computing tasks from environments managed by charge-back accounting. Taken in a broader view of computing costs, enterprises should not, however, be expecting savings.

The issues of cost savings are only a part of the real challenge data center managers face in today's enterprise. A larger concern is the ability to keep up with the rising demand for information from business units within the enterprise. While some of these requests come in the form of the large scale, customized applications data center staffs have been historically responsible for, the bulk of them are more ad hoc and non-repeatable, customized reports. The biggest challenge facing data center managers in meeting this demand for information is that they are ill-equipped because of the nature of their data structures and their application development tools. Hierarchical data structures are not particularly well suited for the variety of data requests presented to data center managers. Particularly in an environment where the programmer may have to translate a

complicated business question into a sensible retrieval program. Application programmers have not historically been required to have a real knowledge of the business process they were supporting. This lack of knowledge makes the process of translating a business request into a retrieval from an operational data structure even harder. More time lost, and more chance of error.

Compounding this difficulty of data structures is the fact that third generation languages were never imagined to be tools capable of rapid application development, with easy flexibility for formatting output. These structures and tools are still useful for the traditional customized report application. But, the time of development for this kind of application is often unacceptable to today's business user.

The recent wider use of relational database management systems for the storage of data and SQL as a query and reporting tool has provided data center staff with some relief from these problems. However, while giving them a more flexible data structure and easier query language, it has also raised the final dilemma. To the extent that these relational systems are supporting the operational, transaction processing systems within an enterprise, data center staffs using SQL query tools will sooner or later discover that increasing the number of queries they can craft and deliver to the DBMS will eventually present a conflict of performance. In a world where OLTP transactions and end-user defined queries compete for resources, which one wins. For the data center manager this question has only one answer, the transactions win.

Let's leave this rather bleak outlook for a moment, and consider what client/server computing has to offer to users in the business units of an enterprise.

Client/server. View from the business unit.

What then does client/server have to offer to end users in business units? Primarily access to corporate data. Over the past several years, enormous amounts of computing capability have been deployed in end user departments, far from the watchful eye of data center managers. Over time this computing power has become more powerful, better connected and has engendered a generation of users comfortable with a variety of productivity tools, most notably spreadsheets, word processing, electronic mail and, to some extent, data base programming. This emerging capability within business units has empowered end users to consider new views to their processes, and to actually produce answers to new questions. The next logical step for these business units is to increase the amount and quality of data they have access to. And they know where that data is kept. Much of it can be gleaned from standard corporate reports. If you do not believe this, look around within your own organizations until you find someone who is taking hard copy output from a standard report, and having it entered into a spreadsheet on their local workstation. Electronic cut and paste of standard information is the best indicator of the value attached to these data, and the extent to which business units are will to go to have the ability to manipulate it themselves. However, significant barriers keep them from easy access to these data.

In the first place, the data the business user requires often does not exist in the form that they see it in the standard corporate report. Few corporate reports are simple listings from enterprise data tables. Significant amounts of work are necessary to translate operational data into the validated and summarized figures observed in standard corporate reports. This work includes, but is not limited to applying standard formatting conventions, editing values for consistency, merging with data in other files and summarizing based on basic business rules.

What are these "groups" good at?

Let's draw a model which represents the path and requisite expertise which must exist to empower a client/server implementation. For this paper, let's assume that the primary goal to provide the business unit with data.

Diagramatically then, we can characterize a flow which maximizes the expertise of all players in the enterprise information system, prepares data into a form to become the "raw material" of business end-users, and promotes the goals of the enterprise.

The beginning, then, is the source, or the corporate managed data (see Figure 1). Usually managed in a system tuned to support operational and transaction based applications, these data are at their greatest level of detail, and support very specific business functions.

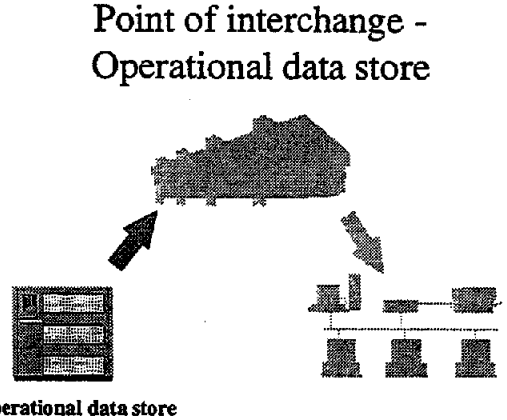


Figure 1

The first steps in this flow involve the transformation of the operational data from transactions to more meaningful units of work. At the most basic level, this could involve summing some number of individual entries to produce a customer

purchase or order. In almost all cases, this will involve adding formatting information to make the fields more understandable to reporting applications (work that may already be accomplished within application logic, but is not immediately available to other users).

As the action moves away from the OLTP applications, the work on the basic data include higher levels of summarization like time periods, lines of business and geographies (see Figure 2). In addition, basic data, now summarized, may be combined with data from other sources. This continuing action of massaging and adding value to "basic" data moves the data further from its "base", usually the hierarchical file system coupled with the transaction monitor program populating it.

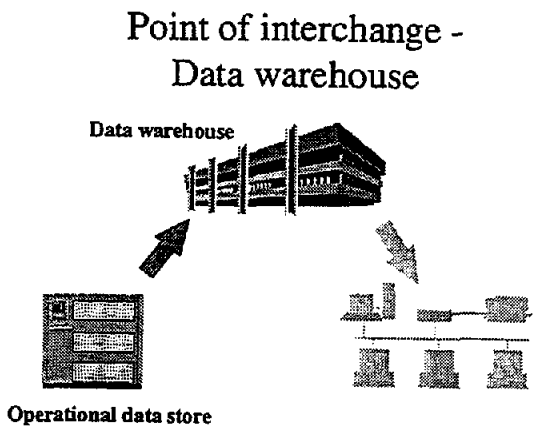


Figure 2

At it's interchange point, the data are arranged in relational tables and a central documentation is available for reference by all applications. The choice of a relational organization reflects the popularity of the systems available to manage data in tables, as well as a recognition of the need to maintain decision support data in a flexible arrangement. It is not clear today, nor will it likely ever become clear for long periods of time, just what demands for data today's enterprises will have. The best data center staffs can aim for is a well thought out model, which meets anticipated needs, but has flexibility for future, unanticipated needs.

Of course, what this describes is a data warehouse. The data being stored has been extracted from operational sources, has been formatted and normalized, has had a dictionary built to describe the elements and, most importantly, is made available as a read-only source. This makes it an excellent data source for decision support applications in a distributed environment. There are other papers which discuss in more detail the steps involved in constructing and maintaining data warehouses. It serves here to state that, without investing in the planning and implementation of a data warehouse strategy, client/server implementations of an enterprise wide, decision support nature, will be more difficult to achieve.

At the point of interchange is the data warehouse. This then becomes the starting point for the decision support and reporting application development (see Figure 3). It too proceeds in an iterative style, moving further from the source data, and closer to the specific presentation requirements of the final consumers.

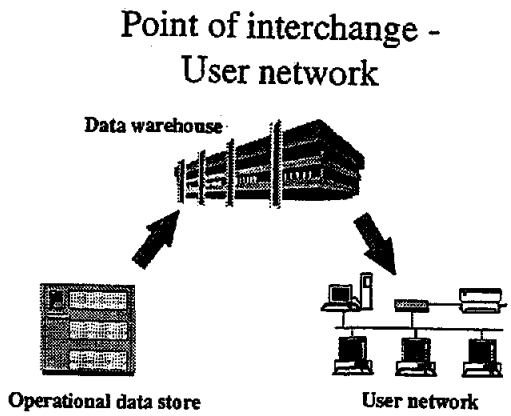


Figure 3

While this model has appeal for describing a flow of activity, it does not address the issue of infrastructure. Without the proper environments in place, and a dedication to resources for maintaining them, client/server implementations organized around this point of interchange will not succeed. In this case, the supporting environments include the availability of relational technology, for storage of the extracted data, network connections and expertise for the movement of data to server platforms and the connection of clients, and client based tool sets capable of working in a variety of platforms, with a wide variety of presenting problems.

In this model, the data center staff has a major interest in all of the infrastructure of the point of interchange , though they will not be responsible for all aspects of them. Data center staff should be responsible for the populating and maintenance of the data warehouse. In addition, they should share in the maintenance of the communication links. It will not be necessary for all communication to be managed centrally, because a great amount of expertise has grown up in business units around the operation of local area networks (LAN). However, the combination of this LAN knowledge with the historical expertise in wide area networking (WAN) can provide a robust environment for effective communication management. Finally, the end user tool set is more frequently the domain of the business unit. Much of the software brought in by these business units is not recognizable to data center staffs. This should not be a problem, however, because in this new organization, the choice of tool should be left to those doing the work. Data center staff have a role in reviewing such tool sets, where they can add value, but they will generally not be the consumers of the tools.

Section 2: Platform selection to support this point of interchange model

The discussion so far has been based on organization and data issues, and has not dealt with the hardware platforms in modern enterprises. The point of interchange model described above, suggests, without being explicit about it, a three level approach to hardware configuration. In general, this approach would have specific platforms dedicated to the tasks of the OLTP, traditional data gathering applications, one for the data warehouse and finally, one to be used by the business analysts and application designers. Theoretically, these could all be located on the same platform. However, based on the demands of the different applications of interest, some comment can be made on the use of the popular platforms for specific segments of the distributed business computing environment. First, lets take a look at some of the key characteristics of the three applications of interest: data warehouse applications, decision support applications and executive information systems or EIS's. Then, let's take a look at the major platforms available to modern enterprises, and examine their suitability for each of these segments.

Application Characteristics

This review is not comprehensive, and is based on a few assumptions. The first assumption is that the goal of making the move to a distributed business computing environment is to maximize the use of personal workstations, which many business units are making available to their analysts and end-users. Having said that, it means that there will not be much said about on-line transaction processing applications. As has been mentioned before, OLTP applications are generally the primary data collector for all business operations. The performance of these applications is crucial to some industries, such as banking and travel reservation systems. The systems must be highly available, have efficient routines for adding and finding individual records, protect the integrity of the transaction data and support very fast response time. Applications written to execute with TP monitors have traditionally served very capably for these tasks. While there are some enterprises experimenting with distributed flavors of OLTP applications, that review could stand on its own as a topic. For the purposes of this paper, it is assumed that the OLTP transactions, which collect the data in the operational data store, will continue to execute in the current host system. Review of platforms, then, will concentrate on their suitably for hosting data warehouse, DSS and EIS style applications.

Data Warehouse Applications

Making data accessible to end-user applications and system developers, in a secure, manageable environment, without intruding on the performance of operational systems is the major function of data warehouse systems. Currently being reconsidered by many enterprises, data warehousing is not a new concept. Many SAS System users will be happy to learn that they have been building data warehouse structures, in SAS data file format, for years. As the concept has become more popular, the requirements (defined some time ago by Bill Inmon) have been receiving more attention. We will not do an exhaustive listing here, but some key elements include the ability to move extracted data to and store data in relational format, to provide a central data dictionary of the tables and fields, to be optimized for read-only query activity and to be available to a wide range of business applications and many end-users. Other features which are useful, but do not usually find themselves listed in the requirements of data warehouse software are the ability of the software to perform the extraction from operational systems and subsequent summarization's, and the portability of the data storage format.

GUI interfaces

The following types of applications, DSS and EIS, often make extensive use of graphical user interfaces. So, a word here about what GUI's are, and what characteristics they have. Graphical user interfaces (GUI) are the layers of software which act as the intermediary between the end-user and the application logic. They can have high network traffic demands because of the amount of graphical information they make use of in both receiving input from users, and displaying results of application logic. In order for this traffic to allow for reasonable response, GUI based systems are best used in environments with wide bandwidth. GUI's perform best when presentation management is performed on the desktop.

Decision Support Systems

Decision support systems provide generalized analytical capability through SQL or natural language query facilities, spreadsheet models, and possibly data visualization tools. These applications can be both I/O intensive, requiring reduction or scanning of very large volumes of data, and CPU intensive, requiring iterative calculations for complicated algorithms. These numerically intensive applications can be employed to produce forecasts over multiple data series, identification of select customer characteristics and the scheduling of large and multifaceted projects. Decision support applications often take advantage of GUI interfaces, which can increase the productivity of analysts.

Executive Information Systems

Executive Information systems draw together and integrate summary and key results information from critical corporate applications in an easily navigable form. The reports presented by the EIS can be prepared by other applications, and just displayed or subsetted by the EIS. The EIS can also produce its own reports from summary data prepared by other applications. EIS applications tend to rely heavily on graphical user interfaces and therefore have the attributes of GUIs noted above. They place a high degree of emphasis on presentation style and aesthetics, as well as content, and therefore produce long and complex graphics data streams. Although something of a generalization, EIS's are often implemented for decision makers in enterprises who are not tolerant of poor response time, so performance is a key issue.

Environmental Support

The following environmental support issues should be considered, as they cut across all of the applications of interest, and are often given insufficient consideration or weight in platform selection and cost.

Security

Legal requirements, as well as corporate confidentiality policies, dictate that access to most corporate data be controlled. The underlying operating system kernel must provide an adequate integrity and protection base for the security system software. Even
the best security software is rendered ineffective if it can be readily undermined by system "wizards" or knowledgeable users.

Encryption can be an important adjunct to access control, especially in networked environments where plain-text packets can be readily intercepted (ever wonder whether your bank card PIN code is really secure?). Encryption is also an effective way to let system managers have the data access authority they need to manage data backup, recovery, and migration, yet shield the contents from them.

Software Maintenance & Upgrades

This is an area that is easily overlooked in platform selection, especially by people who have a great deal of experience with centralized mainframe applications and are considering decentralized or distributed approaches. How do you distribute software upgrades to many decentralized or distributed nodes? How do you get rapid resolution to an application bug? In the distributed environment, how do you maintain synchronization between client and server software levels?

Scalability

Before the age of heterogeneous platforms, scalability issues were covered by simply estimating the number of users needing access to the mainframe, and calculating how large a machine was necessary. The rather unitary dimension of scalability is of no real benefit to the system planner faced with the task of designing an environment where all users will be able to user graphical interfaces, efficient I/O engines and specialized numerically intensive platforms. And all of this at the optimized cost for hardware, software and maintenance. In today's multi-platform, each environment must be evaluated to quantify its potential contribution to the multi-faceted tasks required in a three level configuration.

Networking

In a distributed environment, the network is the defining infrastructure. No network, no distributed computing. Not all platforms, however, were designed with a network in mind. Those which are still do not provide a truly standard methodology for sharing information, and even less standardization in the management tasks of network establishment and maintenance. When integrating heterogeneous computing hardware, networks, operating systems, and applications, planning for fault diagnosis and remedies must not be overlooked as part of the integration process.

Performance Management

Performance management considerations include the ability to measure and manage the environment in terms of application service objectives, to anticipate potential performance bottlenecks which may degrade service, to rapidly identify and correct performance problems, and to schedule timely cost-effective configuration upgrades to meet workload growth.

It is well known that hardware capacities are rapidly increasing and that costs per unit of resource are rapidly declining. Vendors consistently display graphs showing the dramatic cost reduction per MIP, megabyte of memory, and gigabyte of disk storage (and of course their latest offering is setting a new low price point). A common perception from this is that resources for applications will be more abundant, that the overall cost of computing will drop, and that for these reasons performance management is not the issue it used to be.

What needs to be kept in perspective are the even more dramatic increases in resource requirements of modern operating system and applications development software. Object-oriented technologies and the new graphical interfaces have great advantages, but they also carry a stiff hardware resource price. The net result is that the cost-per-user or seat curve is a lot flatter than you would get looking at hardware price/performance alone.

Also, distributed platforms can at first glance appear very inexpensive on a cost-per-seat or node basis when compared to more traditional solutions, but expenses mount rapidly when the number of nodes, connectivity costs, and operational environment costs are figured in.

The bottom line is that acquisition, operating costs, and service levels are still important issues with sophisticated information systems, and that adequate performance management tools are important in managing these issues.

Comments on Several Architectures with Respect to Applications and Platforms

First, an interesting general observation from the Gartner Group:

"*Mainframe computers differ from workstations in one profound way: The ratio of data to MIPS for mainframes has stood between 3,000 and 5,000 Mbytes per MIP for 25 years; the ratio of data to MIPS on workstations has generally held between and 20 and 50 megabytes/MIP for the past 10 years.*"

This is consistent with the notion that mainframes may continue to have an important role to play as corporate data repositories,

and that increasingly resource-intensive applications and presentation logic might best be performed on the desktop.

IBM-(MVS, VM/CMS, VSE)

Mainframe systems are heavily optimized to support block I/O mode for data transfer, and they are very effective with most processing being done in outboard components. In contrast, they are extremely inefficient for applications requiring keystroke-by-keystroke
interaction, or high interactively with the presentation device, such as is the case with popular GUI environments.

In general, the mainframe operating systems have the robustness and operational environmental management tools needed to support corporate mission-critical applications.

MVS Advantages:

+ well developed infrastructure for performance analysis, capacity planning, and problem diagnosis and resolution.
+ powerful and robust environment of system services for networking (SNA/VTAM), transaction processing (CICS and IMS), and data base management (DB2 and IMS).
+ high performance I/O subsystem.
+ huge memories for data caching in I/O bound applications.
+ powerful utilities for data backup, archival, and retrieval.
+ access to specialized processors for vector processing, data compression, data encryption, and data base query.
+ good security.
+ The OS is highly reliable; recovery from errors is a tenet of the MVS architecture.

MVS Disadvantages:

- slow to support TCP/IP (current implementation has not been up to expected MVS standards in performance, quality, or robustness).
- antiquated file system architecture (limitations on filenames, separation of files from their catalog entries).
- high price/performance ratio for computing power.
- unsophisticated user interface (3270).

DEC VAX and Alpha Systems

VMS Advantages:

+ VMS platforms (VAX and AXP) are completely and transparently compatible from the desktop system to the data center system. The exact same application, middleware enabling technologies, and fundamental OS features are identical across the line.
+ has a DOD C2 security rating for system and file security.
+ a highly network-capable system. Digital and VMS pioneered ideas like tightly and loosely coupled clustering systems, and today the idea of a computing cluster is measured against the VMSCluster benchmark technologies.
+ a good bridge of modern computing technologies (file systems, network kernels, etc.) with established and mature capabilities (system management, batch & print management, CPE, transaction engines).
+ AXP price/performance curve is excellent for VMS, even given the relative complexities (and therefore cost) of supporting a mature OS like VMS.
+ Vendor extensions are also supported on PC and UNIX platforms, making investments in some technologies not yet industry standard safer than on some platforms. Vendor (Digital) has a good track record at evolving its proprietary standards into industry standards. For example, XWindows and Motif grew from work sponsored by or developed by Digital.

VMS Disadvantages:

- systems typically are more compute capable than I/O capable; VMS systems do not have the sophisticated intelligent I/O functions of traditional IBM mainframes, for example.
- VAX price/performance curves are not as competitive, especially in the fact of UNIX-based systems that are "growing up" and encroaching on the data center.
- available on the desktop, but is still primarily viewed by the customer base as a department or data center system. This means that customers are slow to move to upgrading or replacing slower VAX desktop systems with the more nimble and capable AXP desktop systems. This means that the customers may not be able/willing to take advantage of the desktop->data center seamless integration.

UNIX

This platform is based on some variant of the UNIX operating system. Once found only in academic and scientific communities, UNIX has made significant inroads into commercial computing in recent years. Vendors of hardware platforms have licensed and extended variations of the UNIX kernel, bringing both enhancements, and proprietary feature sets to the platforms.

UNIX Advantages:

+ file system designed for distributed management and network wide access, capable of organizing large, distributed storage
+ networking capabilities built in as part of operating system, capable of enabling enterprise networks using TCP/IP
+ scaleable across many size platforms, up to very large processors
+ enabling symmetric multi-processing, especially useful for dividing up the work of large data base queries
+ excellent GUI technology
+ standards based implementation for security (Kerberos)
+ major database management systems vendors offer products
+ excellent for CPU intensive, numerically intensive calculations

UNIX Disadvantages:

- standards not consistent for distributed file systems or distributed computing capabilities (i.e. NFS vs AFS; OSF/DCE vs SUN ONC)
- system too complex to be managed by end-users
- not optimized for I/O intensive tasks
- GUI is resource intensive and can impact network performance
- common personal productivity tools not generally available (i.e. spreadsheets, word processors and E-mail)

32 bit PC based platforms (WINDOWS NT and OS/2)

These systems form a new class of platforms, taking advantage of popular GUI's like the Workplace Shell on OS/2 and the WINDOWS Program Manager, and running on the very popular INTEL based chip architecture. Both are 32 bit operating systems, not running over top of DOS, and provide the capability of taking advantage of the memory management available as a design feature of the 486 chip architecture.

32 Bit PC Platforms Advantages:

+ familiar GUI enables end-users to navigate among many software packages, including configuration management
+ systems can be configured and minimal system maintenance can be accomplished by sophisticated end-user personnel
+ good memory management for CPU intensive tasks
+ much software available, including database management systems
+ competitive pricing

32 Bit PC Platform Disadvantages:

- pursuing proprietary standards for object management and data base access
- file system limitations for large, distributed environments
- scalability limited, not currently available on other hardware architectures (though test versions of NT available on DEC Alpha platforms, and NT and OS/2 being modified to run on PowerPC architecture)
- networking not mature; included as an add-on to OS/2 and still under development for WINDOWS NT
- resource requirements too high for end-user client platform for WINDOWS NT
- system management tools still immature
- security not mature

PC Platforms (WINDOWS 3.1, OS/2 & Macintosh)

These platforms form the major "client" environments in today's client/server configurations. They run on the most affordable hardware platforms, are end-user friendly, and can be used as stand alone environments for end-user computing. The greatest growth area in software development today is providing personal productivity software to operate in these environments.

PC Platform Advantages:

+ familiar GUI enables end-users to navigate among many software packages, including configuration management
+ systems can be configured and minimal system maintenance by sophisticated end-user personnel
+ interprocess communication for sharing data among applications
+ much software available, including database systems
+ competitive pricing

PC Platform Disadvantages:

- networking capabilities are not integrated well into the operating systems
- limitations for large I/O intensive applications (WINDOWS 3.1 still a 16 bit operating system)
- WINDOWS 3.1 still not a true multi-tasking environment
- performance management tools immature
- limited scalability; not available on platforms outside of INTEL chip architecture (Mac available on Motorola architecture; Mac and OS/2 going to PowerPC, while WINDOWS can be run as emulation under some UNIX)
- security not mature

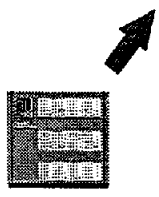
Which Platform for Which Job?

Having reviewed some of the issues for consideration about the strengths and weakness of some selected platforms, it is time to integrate this information into comments about the appropriateness of placing different segments of an enterprise's computing tasks on each of these platforms. Let's begin with the Operational Data Systems.

Platforms for Operational Data Systems

For the time being, the best choice for operational style systems (i.e. predominantly OLTP based, with massive storage of historical, transaction detail) continues to be the traditional host, mainframe environments (see Figure 4).

Operational Data Platforms



Operational data store

- Host/mainframe
- Usually not a choice to install, but rather to migrate from
- OLTP and historical storage already in place
- Possibly re-architect end-user interface of OLTP

Figure 4

This makes best use of OLTP systems which work, and the strength of the mainframe systems for managing large file systems and I/O intensive processing. For many enterprises, the "choice" of a platform for operational systems, is not really a choice of where to place these systems, but whether or not to move them at all. The examination of all operational applications is usually being done as a way of identifying systems which might be candidates for migration away from mainframe processing. Because DSS and EIS type applications are based on ad-hoc or summarization activities, and can involve use of GUI technology, they are better candidates for such migration than OLTP applications. To the extent that OLTP applications can be subdivided to place the user interaction layer in an environment that efficiently supports a GUI, then it may pay to re-architect them. However, in general, mainframe host systems continue to be the best location to continue to house OLTP and historical storage applications.

Data Warehouse Platforms

Of all of the segments, this is the one where the greatest variety is legitimately available to enterprises. Beginning with the traditional host (i.e. MVS and VMS) platforms, ranging down to the 32 Bit PC platforms, system implements can place parts, or all of data warehouses on these platforms. As noted above, each platform has strengths and weakness. However, for the enterprise which cannot migrate their systems to either UNIX or 32 PC platforms, their mainframe host platforms can still offer good solutions as a data warehouses (see Figure5). This is

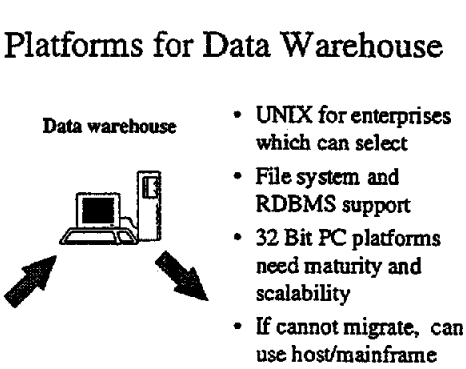


Figure 5

partly in recognition of the fact that many enterprises are already doing this, without necessarily acknowledging the strategy as defined.

Where enterprises can build their data warehouses, and can further choose between the UNIX platforms and the 32 Bit PC platforms, the edge today would go to the UNIX systems. This is based mostly on the maturity of these platforms versus the 32 Bit PC platforms, the superior file system management (talking into account the observations made earlier about the incompatibility of different proprietary file systems), the strong presence of the major RDBMS vendors offerings and the implementation of multi-processing. The 32 Bit PC platforms may become viable alternatives, however, much of their ability to deliver on the scale which might be required is still untested. In addition, the 32 Bit PC platforms will need to improve on their security offerings, possibly offering Kerberos capability and encryption.

Platforms for DSS and EIS Applications

While these applications have been considered separately, they are actually very similar in their demands on platform specific services. Both must be able to execute on affordable systems, have end-user friendly GUI's, have good communication to other platforms and integrate well with other popular productivity tools. DSS applications will need greater access to good CPU engines, for the more numerically intensive tasks, while EIS applications will have to have access to good GUI's and connections to other platforms for access to summarized information and electronic mail. The platforms of choice for these applications are the PC and UNIX platforms (see Figure 6).

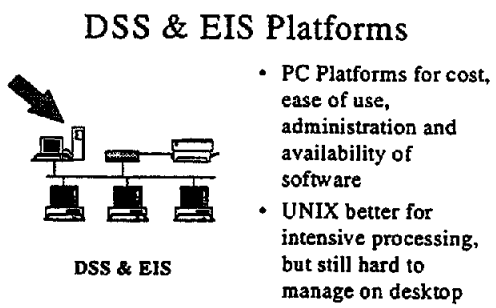


Figure 6

In this area, OS/2 can be included in the PC platform category because of its lower (than WINDOWS NT) system requirement. Between the two categories, the major differences are the communications systems, and system maintenance. Both systems have good GUI capability, can perform well for highly graphical intensive applications and, at the high end, can be price competitive. UNIX has the better communication system, with TCP/IP being built in. However, UNIX is still not manageable as a desk top environment for a typical end-user. Also, the PC platforms have much more personal productivity software available, and have moved further along in making it all work together. Taken together, these factors seem to point to the PC platforms as the current platform of choice for DSS and EIS platforms in a distributed environment. This does not mean that UNIX is not a viable alternative, only that there will be more sophisticated support required to allow end-users to avoid the complexities of the UNIX operating systems.

Section 3: SAS System Client/Server Enhancements

Where is SAS Institute heading in the enablement of its customers in a client/server world in the near future? The answer is, in a number of directions. As always, there are capabilities which you, our customers have identified as key to your continuing success implementing technology to meet your business needs. Specifically, SAS System tools will be enhanced in the areas of data transfer services and support of an increasing number of program to program communication protocols. Also, SAS Institute will be making available a way to allow non-SAS applications to access SAS managed data.

Data transfer services - Remote Library Services

With the production releases of SAS/SHARE and SAS/CONNECT on all platforms, SAS customers will be able to make local library references to data stored on remote platforms. This will enable applications to name data sets of interest via a LIBNAME statement in a local environment. When any files in the locally named library are called by a SAS program, data will be transferred from the remote platform to the locally executing requester program. The server platform will continue to hold locks on the data, as if the client program was executing in the server environment. This will be of particular interest to users who require the ability to edit data which is remotely stored, without having to actually have their program run in the remote environment. This remote library services capability will be available in both the SAS/SHARE and SAS/CONNECT products.

APPC support for WINDOWS and UNIX platforms

The SAS System will be expanding its support for the Advanced Program to Program Communication Protocol (APPC) by extending support to the WINDOWS platform as well as selected UNIX platforms. Up to now, the only platforms supporting APPC communications have been MVS, VM, VSE and OS/2. APPC support on WINDOWS and UNIX platforms will allow these platforms to have SAS sessions started in their behalf in APPC managed regions in MVS and VM environments. For the large enterprise which needs to continue to store large quantities of data in these centrally managed environments, this will enable remote SAS processing to go on, using interactive resources in mainframes, without the use of TSO resources. For the UNIX platforms, support for APPC communications means that mainframe data centers can choose an alternative to TCP/IP as a method for incorporating these workstations into their enterprise wide networks.

*Data access from non-SAS applications - SAS/SHARE*NET*

Beginning with releases this summer SAS Institute will be offering an access path to SAS System managed data to non-SAS applications. Using the Open Database Connection Services API from Microsoft Inc., non-SAS applications will be able to access SAS data in a local WINDOWS or remote server environment. SAS/SHARE*NET will enable non-SAS applications to connect to SAS/SHARE servers, offering them the opportunity to retrieve data from any source available to that server. Because the SAS System offers access to a wide variety of data, including SAS data files, these non-SAS applications will be able to use data from any of these available sources, through the SAS System.

Summary

Enablement of client/server applications is difficult. Many participants in the activity of an enterprise have a stake in the ability to access centrally managed data. While data center professionals look to reduce costs, client/server implementations can also offer them an opportunity to work cooperatively with end users in business units. The new point of interchange between the data center and the business units will be creation and maintenance of data warehouses, and the infrastructures necessary to support them. With the proper structures, end users in business units can be empowered to do what they know best, solve business problems, by making use of the vital resource of enterprise data..

For further information about SAS System tools for client/server computing contact:

John McIntyre
Enterprise Computing
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513

SAS, SAS/SHARE and SAS/CONNECT are registered trademarks of SAS Institute Inc., Cary, NC, USA.

Other products mentioned are trademarks or registered trademarks of their respective owners respective owners.