Business Decisions in Selecting Platforms for Information Systems

Dan Squillace, SAS Institute Inc., Cary, NC
Gary Burchett, SAS Institute Inc., Cary, NC
Tom Cole, SAS Institute Inc., Cary, NC
Jim Weathers, SAS Institute Inc., Cary, NC

ABSTRACT

This paper is intended for anyone involved in the selection of computing platforms for new applications, or anyone considering moving existing applications to a new platform. Both the business and technical aspects of making such a decision are discussed here. This paper is a companion to the interactive panel discussion "Implementing Distributed Computing in a MultiPlatform Environment".

INTRODUCTION

Much of the debate over the selection of computing platforms is tinged with emotional overtones and even elements of religious fervor. The term "rightsizing" serves as a rallying cry for some, while raising the hackles of others. Blanket assertions such as, "the mainframe is dead," or, "Unix is a toy operating system," miss the mark and can hinder the search for optimal computing platform(s) for your information systems.

Motivations often cited for changing platforms include:

- The need to save money
  - lower hardware, software, and support costs
  - overall lower cost of computing.
- The ability to shorten the application development cycle, especially by taking advantage of object-oriented technologies for application development
- The ability to take advantage of newer technologies such as GUI (Graphical User Interfaces) and imaging. These technologies are not widely implemented for mainframe environments and, when they are, may have very limited functionality.
- The need to decentralize applications and give more control to the user organizations
- The need to improve application performance.

The search for an optimal platform or mix of platforms depends heavily on understanding your business requirements, the attributes of your data and applications, and their operational requirements. This paper discusses:

1. application characteristics which are important in platform selection
2. considerations relating to operational environment support
3. the relative strengths and weaknesses of several system architectures with respect to these application characteristics and operational requirements.

APPLICATION CHARACTERISTICS

It is difficult, if not impossible, to make a reasonable and cost-effective platform selection without first gaining some understanding of the characteristics of the application(s) you want to run on the new platform. Identify the categories of applications, and then consider the typical characteristics of those categories of applications. The categories discussed here include Online Transaction Processing (OLTP) systems, Decision Support systems (DSS), and Executive (or end-user) Information Systems (EIS).

OLTP Systems

Online Transaction Processing systems encompass applications such as banking services through automated teller machines, order processing/inventory management, customer service, and the interactive components of traditional accounting functions, such as accounts payable, accounts receivable, and general ledger.

Typically, there is relatively little computing work (CPU and I/O) per transaction in these types of systems. However, the volume of transactions can be very high, making the aggregrate computing load large. Generally though, these systems tend to be more I/O than CPU constrained. And the computing which is done tends to involve character and integer or decimal data types.

These systems can be implemented using centralized, decentralized, or distributed data models, depending on the type of business organization being supported. The model chosen influences platform choice decisions because each model can result in varying criteria for the systems' networking and computing components.

As an example, let's look at several scenarios for an order entry system and the characteristics of these scenarios.

1. Centralized inventory management, order entry in a number of locations.
   - The data are centralized.
   - The computing (order preparation) can be distributed or centralized.
   - Network load can be high in aggregate, though low per transaction.
   - Host/server utilization can be high.
   - Host availability requirement is high.
   - Network availability must be very high (no network, no orders).
   - Rapid response time is required (can't keep the customers waiting).
2. Decentralized inventory management and order processing.
   - Independent inventory is managed for each location.
   - Batch updates are made to corporate headquarters at night.
   - Network load is minimal.
   - Minimal network availability is required.
   - Host utilization is minimal.
   - Host availability requirement is minimal.
3. Distributed inventory management, decentralized order processing.
   - Variation of (2). Most orders satisfied out of local inventory, but other sites can be queried if stock is not available

locally.

- Network load is moderate: (traffic only for orders which can't be satisfied locally).
- Moderate network availability is required.
- Host utilization is moderate.
- Host availability requirement is moderate.

Decision Support Systems

Decsion support systems provide generalized analytical capability through SQL or natural language query facilities, spreadsheet models, and possibly data visualization tools.

- Decision support applications
  - can be both I/O and CPU intensive
  - can be more numerically intensive than other commercial applications
  - can require reduction or scanning of very large volumes of data, a task at which the mainframe has traditionally excelled.
  - often take advantage of GUI interfaces that are more efficiently and readily implemented on PC or workstation platforms.
- GUI interfaces
  - can have high network traffic demands
  - need wide bandwidth for reasonable response
  - perform best when presentation management is performed on the desktop.
- Keystroke or mouse-position sensitive interfaces
  - require rapid interaction between the keyboard or mouse and the computing platform to have satisfactory usablility
  - require rapid response for small packet traffic if done through a network.
  - perform best when application logic, as well as presentation management, are performed on the desktop.

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
  - place a high degree of emphasis on presentation style and aesthetics, as well as content, and therefore produce long and complex graphics data streams.

OPERATIONAL ENVIRONMENT SUPPORT

The following operational environment support issues 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 underminded 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 and Upgrade Distribution

  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?
- Data Integrity

  Transaction checkpointing, rollback, journalling, and forward recovery are very important to maintaining data integrity in OLTP systems and ensuring that completed transactions are not lost.
- Data Backup/Recovery

  Even if you protect from hardware failure by choosing the most sophisticated RAID level 5 storage systems available, application software or human error can still lay waste to your data. You need both the ability to back up data in a timely manner, and to readily retrieve it when needed.
- Availability

  There are several strategies which can be used in combination or traded off to improve availability.
  - Replicate system components, paths, etc. so that an alternate is available in hot standby when needed.
  - Install systems components which are inherently more reliable because of sophisticated self diagnostics, error checking, and internal redundancy.
  - When integrating heterogeneous computing hardware, networks, operating systems, and applications, include planning for fault diagnosis and remedy 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. Your favorite vendors are incessantly plying you with 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 you need to keep 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. So, 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 envionment 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 PLATFORM SELECTION

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-Compatible Mainframe

General Comments:

One objective of using distributed client/server technology, with the mainframe acting as a server, is to free up relatively expensive mainframe cpu resources. The savings may not be there to the degree you would expect. The applications and presentation logic moved outboard to the remote node is replaced with communications and program-to-program protocol overhead.

Client/server technology is not new with workstations and PCs, although use of the term "client/server" is generally associated with these platforms. Both client and server processes can reside on the same platform and have done so for years on the mainframe in many application, data base, and transaction processing implementations.

ASCII/EBCDIC text translation and numeric data transformation issues can be a headache when integrating mainframes with PCs or workstations. Related considerations include sort sequence differences, and management of non-numeric data fields which should not be translated.

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 interactivity with the presentation device, such as is the case with popular GUI environments.

In general, 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.
- well understood centralized computing environment.
- 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.
- Posix support forthcoming shortly.
- The OS is highly reliable; recovery from errors is a tenet of the MVS architecture.
- superior control of applications to match corporate priorities.
- excellent manager of system resources.

MVS Disadvantages:

- slow to support TCP/IP (current implementation has not been up to 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).
- Numerous complex file systems and access methods makes application migration difficult.
- Interprocess communication is complicated and generally requires special authorization to establish.

VM/CMS Comments:

- Many VM services follow the client/server model, and have for years.
- Setting up a server is a well understood process.
- access to VM, MVS, and VSE heritage data.
- SQL/DS (soon to be DB/2 ??), VSAM.

VSE Comments:

- traditional data repository.
- Enterprise data is already here.
- SQL/DS (soon to be DB/2 ??), VSAM.

DEC VAX and Alpha Systems

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.

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 datacenter.
- has many legacy features and existing standards that make it slow to evolve new features and technology than some environments, such as PC/Windows
- available on the desktop, but is still primarily viewed by the customer base as a department or datacenter 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.

SUMMARY

If there is one single important point this paper has been beating around, it is this: Understand your business and applications requirements first and then find the proper technologies and platforms to meet those requirements.

As a final note, if one of your objectives is to create an open environment which gives you more freedom in choice of hardware vendors then consider one or both of the following.

- Develop your applications to prevailing open standards such as Posix. Avoid making vendor-specific extensions a critical component of your applications.
- Acquire your strategic software from vendors such as SAS Institute who are committed to supporting all the leading open and proprietary systems platforms.