LET MVS BE THE DATA SERVER

By Stephen Force

Introduction

Let's face it. MVS-based systems have an identity crisis. One faction believes they can do the job of data processing better on distributed platforms rather than on MVS. Users have their pet software products and natural preferences as to which hardware platform and operating system environment to use. And it seems that the bean counters and budget meisters concur, allowing this to happen.

If the data processing trend seems to be getting away from the MVS operating system environment, what is the value in keeping MVS?

Not a Dinosaur

The theory that the mainframe is a dinosaur and is becoming extinct simply does not reflect reality. MVS is still evolving in new and very exciting ways, implementing state-of-the-art connectivity features inherent in the base control program. VM also has maintained its steady evolutionary pace, offering a myriad of options simply not envisioned in the past. And what about DOS? This operating system was supposed to be laid to rest several years ago, but now we have VSE/ESA. UNIX has also been ported to the mainframe. Additionally, both IBM and Amdahl offer mainframe-based UNIX systems.

Applications still are being developed on MVS, but perhaps at a slower pace; consideration is being given to program portability and connectivity (more programs are being written in the C language with TCP/IP and/or APPC functions being used). Does this sound rather fanciful? Are application system developers really using these more today, or is this just more of the same old stale pundit and media hype?

For example, application programs developed and written for CICS/MVS [1] can be ported to:

- VSE/ESA CICS;
- VM CICS;
- OS/2 CICS;
- AS/400 CICS;
- AIX CICS; and
- CICS running on Hewlett-Packard UNIX systems.

MVS-based TCP/IP application program sockets written in C and using standard TCP/IP protocols can be ported to.[2]

- VM;
- OS/2;
- MS-DOS;
- AIX;
- UNIX; and
- Apple Macintosh.

CICS/MVS applications can now be written as TCP/IP sockets by using CICS-provided TCP/IP protocol commands. Using this powerful CICS feature, without going through many serious gyrations, an enterprise can now exchange data with any TCP/IP socket on the internet.[3]

Why Keep the Mainframe?

We could argue all day over the pros and cons of distributed processing vs. mainframe. Obviously, there is no one single solution to this complex, multi-faceted problem. The article, "The Mainframe's Role in Multiplatforming" by Anne Davies and Bob Reinke (***Technical Support***, December 1992) illustrates quite well some of these pros and cons, as well as some of the major roles of the mainframe in the future.

This article shall attempt to provide arguments towards retaining the mainframe in the enterprise, while using the local area network (LAN) and data repository servicing function on MVS as the example.[4]

Reasons for Using MVS as the LAN and Data Repository Server

Cost Consideration — Even though mainframe hardware is costly, much of this cost goes into the redundancy that makes it reliable. Competition from several vendors, a shrinking customer base, a large used equipment market and competition from other platforms have forced the price down.

MVS software is expensive; however, if the proper sized machine is used (right-sizing), and if only software products that are necessary are installed, then the cost of this software should be generally in line (price tiering).

Before stating that MVS DASD is expensive, the enterprise needs to make a complete cost estimate of all the disk drives on the LAN, then calculate percentage of usage (occupancy), availability, the cost of having several types of disk drives as spares (don't forget the controllers), and the time and expense needed to recover mission-critical data. Only then can a fair cost comparison be attempted.

Tape devices on MVS also cost money. An enterprise again needs to define what role tape plays in the organization. Tape streamers on LAN servers are much slower than the IBM 3480-style devices and they are not cheap. LAN tape streamers are used only a fraction of the time, where MVS-based tape drives can be always in use (if desired). Tape cassettes for these tape streamers are also very expensive and hold only a fraction of the data that the IBM 3480-style device cassettes can.

CD-ROM disks and drives are a fairly recent phenomena in enterprises. These devices are also expensive and CD-ROM devices also need to be constantly in use to be cost-effective.

Effective LAN data management requires people to coordinate, initiate and verify the data backup. If several departments have LANs, and each has its own LAN data manager, then the cost per byte must be high.

In a well-planned, well-run MVS complex, a small group of people (perhaps only one person) need to be involved with the data management. MVS backups, whether full or incremental, normally execute without problems, and are extremely fast and very reliable. After the data are backed up, the cartridges are either replaced into their assigned position (maybe by a robot), placed into a fireproof vault close by, or sent to a off-site storage vault. Where are the LAN data backups stored?

MVS systems programmers are very

aware of the trend toward staff reduction. We must do more with less of everything. With the current streamlining and consolidation of computer centers, there are simply fewer shops around. MVS is much more stable now than it was in the past, requiring less MVS internals problem diagnostics time and expertise. This means more talent is available to support enterprise-related activities, such as optimized disks and reliable, timely data backup algorithms.

Outsourcing: To Be or Not To Be

Outsourcing means two things: being outsourced or being the outsourcer. A enterprise needs to know about outsourcing to consider using it, effectively using it or to become a outsourcer.

All MVS computer centers can provide "outsourcing" (services) to its "customers" (other departments) if it knows how to do it, can sell itself effectively, controls costs and provides comparable value as to a outside outsourcing vendor.

MVS Offers Powerful Services

MVS offers state-of-the-art fiber optics connectivity (ESCON) and some of the fastest commercially available data access from disk (fast, reliable, large capacity, multipathed, cached disk drives and control units). For archived, yet quickly available data, CD-ROM in jukeboxes can be installed or tape cartridges can be automatically retrieved and mounted by installing tape silos (that use robot arms).

To get network access to these services, users can choose several vehicles:

- TCP/IP;
- APPC;
- APPN;
- SNA;
- JES Remote Job Entry (RJE) or Remote Job Processing (RJP);
- JES Network Job Entry (NJE);
- VM RSCS to JES NJE connections;
- etc.

Reliability, Availability and Security

MVS mainframe resources are reliable, and in case of DASD hardware failure, data can be recovered to a similar device reliably and quickly.

MVS operations are either staffed or have software monitoring systems to help ensure availability. Tape silos can be installed so computer centers can offer true non-stop computing environments without people.

For maximum availability, MVS computer centers can isolate the mainframe-based LAN servers from other MVS production systems by:

- running the MVS LAN server on a separate CPU;
- running the MVS LAN server in another logical partition (IBM PR/SM or Amdahl MDF); or
- running the MVS LAN server as a virtual machine on VM.

MVS-based security systems, when properly implemented, meet federal security requirements. MVS resources are (or should be) in secure areas, protected from unauthorized entry, fire, bombings or sabotage.

Automation

Several data management functions have automation as inherent functions. Other functions can be automated by thinking things through and planning well, and then accomplished by using one of the several possibilities available to automate MVS systems.

MVS as a LAN and Data Repository Server: Five Possibilities

There are several ways to get data both to and from MVS. This article addresses two of the most common, widely known and used data transport protocols today:

- File Transfer Protocol (FTP); and
- Network File Server (NFS).

Additionally, three other possibilities will be discussed:

- MVS can also be used as a Novell Netware server by implementing the LAN Resource Extension and Services/MVS (LANRES/MVS) product from IBM.
- Data Facility Distributed Storage Manager (DFDSM), a new offering from IBM, not yet generally available, allows MVS to handle backup, restore and repository activities for a LAN.
- HARBOR from New Era Systems Services (Calgary, Alberta, Canada).

If DFDSM performs as envisioned (reliable, easy to implement, administer and use) and if the price is perceived as fair, it could very well be a welcome addition to the IBM family of products.

File Transfer Protocol (FTP)

For a simple, unsophisticated and inexpensive LAN server, the TCP/IP file transfer protocol (FTP) might be used. Data can be stored on the mainframe (server) and a client can either obtain or return the desired data. A caveat: The FTP protocol does not provide any locking mechanism. This means there is no simple way (other than a user-written interface) to provide assurance that the data is not being used by two or more clients. Also, once the data is obtained from the server, there is no way to guarantee that the changed data is returned to the server.

FTP is quite simple to use. After connecting to the desired host (server), and providing the user and password (if necessary), one can control both the source and target directories (or in MVS, high-level qualifiers) using provided commands, then either GET or PUT the file.

Most FTP functions can be automated. MVS-based processes can initiate a FTP request for data transmission or retrieval. To do this, however, the target host must be ready to receive TCP/IP FTP commands. Or, the client can initiate the FTP requests at a desired time or event. All of the data on the workstation can be fairly easily transmitted to the server for subsequent backup to a mainframe tape or cartridge. This can reduce the need to have tape streamers on either the departmental LAN server or on the individual hosts themselves.

For more information on the TCP/IP and the FTP protocol, please see the November and December 1992 ***Technical Support*** issues on the two-part series "What is TCP/IP?" Publications referred to in these articles offer a wealth of information certainly not covered in these articles. For specific usage, please refer to the TCP/IP FTP documentation applicable to your systems and environment.

Network File Server (NFS)

The MVS implementation of Network File System (NFS) offers more possibilities to the MVS operating environment than FTP. NFS was designed to be a LAN server, and thus provides the power and flexibility needed to be a true enterprise data server.

The NFS protocol, developed by Sun Microsystems, Inc., allows computers in a network to access each other's file systems. Once accessed, the file systems appears to reside on the local host (client).

To use NFS, a TCP/IP network must be available. NFS access requires usage of TCP/IP protocols, such as eXternal Data Representation (XDR) and the Remote Procedure Call (RPC). XDR is used to provide differing systems data representation resolution and the RPC provides the ability for client programs to invoke a procedure on the server without having to write complicated interface programs or special sockets.

NFS can be easily compared to a LAN file server. When a PC connects to a LAN file server, it assigns a letter to each logical device. For example, device "C" on a LAN file server might be known to the PC (client) as logical device "V". NFS operates similarly. After successful server connection, the client treats the server device as its own, using normal commands.

NFS provides inherent data security by requiring all files that can be accessed be "Export" files. Only these files can be accessed. Export files can be defined by the server administrator as read-only or full access (read/write).

MVS NFS requires all data serviced be cataloged in Integrated Catalog Facility (ICF) catalogs.

Data Representation in MVS Files

Because NFS uses normal MVS functions and access methods, it must store the client data in MVS format. This does not mean, however, that the data is changed; rather, it is simply stored in a format acceptable to MVS. The data storage format is transparent to the client.

The client can either choose to control how NFS stores the data on MVS or can opt for the installation defaults.

Under DFSMS and DFHSM Control

Since all data are in classical MVS data set file format, they can be stored and maintained under Data Facility Storage Management Subsystem (DFSMS) and Data Facility Hierarchical Storage Manager (DFHSM) implementation and control.

Component of the MVS Data Facility Product (DFP)

MVS users who elect to use NFS need to have implemented a version of DFP with NFS included. DFP 3.2 or higher contain the NFS server and client software necessary to become functional.

Client NFS commands included with MVS NFS include support for:

- MS-DOS;
- OS/2; and
- AIX or UNIX.

These client commands can be retrieved from MVS by using FTP.[5]

APPC Support Not Currently Inherent in NFS

For those interested in using NFS, but who have only opted for APPC rather than TCP/IP communications, there is currently no direct connection available. Perhaps a APPC program that issues TCP/IP NFS RPC requests can be written to emulate direct connection. Readers who know of a solution are urged to either write a letter to the editor, or better yet, write an article on just how it can be done.

LANRES/MVS

LANRES/MVS Release 2.0 (IBM product 5695-123) establishes a server environment on MVS that allows Netware clients transparent access to mainframe resources such as DASD and printers. A key feature of LANRES/MVS is the transparent manner in which it provides functions to end users who are accustomed to receiving services from Netware LAN servers. End users on a variety of programmable workstation platforms interact with Netware in their usual manner and are virtually unaffected by the LANRES/MVS software running on the Netware server and MVS.

As with other Netware implementations (and also with NFS), there appears to be no difference between data stored on the client, server or MVS software supporting the Netware server on the LAN. Each Netware volume can be to a disk drive letter in the client.

Netware users can either route print requests to the MVS system from printing, or MVS-based applications can route their output to a LAN-based printer.

Because MVS-based mainframe systems offer such tremendous data storage and fast access to the data, LANRES/MVS allows large amounts of data to be processed efficiently and shared among large groups of end users, thus making data movement between data servers quite unnecessary. The product has been enhanced to allow multiple MVS-connected Netware servers access to the same data in read-only mode. This allows LAN software administrators to maintain a single copy of commonly used software, rather than several.

Fitting the DFSMS Philosophy

LANRES/MVS stores its data as VSAM linear data sets (LDS). Each LDS contains many client files and is treated by Netware as a separate disk. These VSAM LDS files can be under the control of DFSMS and DFHSM.

Connectivity

LANRES/MVS supports several connectivity protocols:

- Systems Network Architecture (SNA) [including APPC (LU6.2) support];
- TCP/IP; and
- VM PWSCS.

Clients supported:

- DOS;
- Microsoft Windows;
- OS/2;
- Macintosh; and
- UNIX.

Requirements

LANRES/MVS requires MVS/ESA 3.1.3 or higher, a PC running at least MS-DOS 3.3 and with enough RAM and disk space available to run Netware V3.11. Also, this PC must have the communications hardware and software necessary to support the desired connectivity to both the LAN and MVS.

Data Facility Distributed Storage Manager (DFDSM)

For users who do not want to relinquish control of their data or simply desire to maintain it locally, there are software products that allow us to accomplish this and still use the mainframe as the data repository.

One product, called DFDSM [IBM product number 5648-020]), which runs on both MVS/ESA and VM, provides server software that allow:

- total or incremental data backup; and
- archiving data from the clients hard disk, freeing up valuable disk space but having the data instantly retrievable when necessary for both IBM and non-IBM LAN networking environments;

DFDSM functions are very similar to the DFHSM product. Users comfortable with the DFHSM functions and commands should find DFDSM quite familiar.

Extending the DFSMS Philosophy to the Distributed Environment

DFDSM extends the DFSMS philosophy to the distributed environment, allowing users to establish policies to manage data availability, data recoverability and storage resources on heterogeneous platforms or networks. Although DFDSM is a component of MVS and VM DFSMS, it is not necessary for DFSMS to be active since DFDSM runs in its own separate address space or virtual machine, and does not need DFSMS-specific functions.

Data can be stored or retrieved from several storage medium: DASD, tape, cartridge or CD-ROM.

DFDSM Connectivity Protocol and Supported Clients

Communications between client and server are supported in a several ways, including: direct connection, LAN-connection through gateways and/or adapters using TCP/IP, VM Personal Workstation Communication Services (PWSCS), APPC or 3270 communications.

Non-IBM clients supported include:

- SunOS Version 4.1.1 or later using TCP/IP communications;
- Apple Macintosh Operating System Version 6.0.7 or higher, MacTCP;
- Novell Netware Version 3.1.1 or later;
- Microsoft Windows Version 3.0 or higher; and
- TCP/IP running in a Microsoft Windows window.

IBM clients supported include:

- Several PC DOS-based software systems;
- OS/2 V1.3 or higher TCP/IP products; and
- AIX V3 for RS/6000.

The Server

The DFDSM server maintains all details of the data held in the server data base from which the owning (or autho-

rized) client can obtain all pertinent information about its data.

DFDSM retains the original file name, not converting it to another name. This is necessary for data originating from OS/2 High Performance File System (HPFS), UNIX-based systems and MS-DOS. The data are stored in their original format (ASCII remains as ASCII), not converted to EBCDIC. Since the data retains its own name and data format, it is not possible for normal MVS and VM utilities to act upon it, hence the reason for DFDSM to run autonomous from DFSMS and DFHSM.[6] Also, since the data are not in EBCDIC, they are not usable by the mainframe application without conversion. However, if the data are required by a mainframe application, they can be accessed using the normal DFDSM client/server protocol and then converted.[7]

DFDSM operates quite similarly to DFHSM, although due to these reasons, it runs completely autonomous. There is no interaction between DFHSM and DFDSM.[8]

The Client

A DFDSM client can connect to the DFDSM server and initiate various types of backup. These may be full or incremental backups and tree selection, complete disk or individual file backups. Also, a DFDSM client can, at any time, restore or retrieve any file or group of files already backed up or archived on the DFDSM server.

Clients can also make archive copies of files and delete the original from the workstation or the file server, if desired.

Supported operating systems environment DFDSM client software is shipped with the MVS and VM DFDSM product. The workstation or LAN user can obtain installation diskettes from the DFDSM storage administrator to install the server software.[9]

Cross-Client File Retrieval

DFDSM allows cross-client file retrieval. This powerful capability allows clients access to information produced on other types of operation system platforms. Examples include:

- Sun workstation to IBM AIX;
- IBM AIX to Sun;
- OS/2 HPFS to MS-DOS; and
- Microsoft Windows applications (using MS-DOS naming standards) to OS/2 HPFS.

A practical example might be a document produced by MS-DOS-based Word Perfect for Windows that can be accessed and modified by a UNIX-based Word Perfect client. This can help minimize document exchange problems faced in a multi-platform environment.

Automated Operations

The DFDSM server can initiate and execute all backup and archive functions. To do so, the DFDSM client must be connected to the DFDSM server and the DFDSM client task must be prepared to perform the received commands.

Backup and archive requests can be initiated from either the DFDSM server or the client. Each client can automate its backup and/or archive activities by activating a "timer pop" at the user (client) specified time.

Availability and Specific Product Information

DFDSM should be available some time in early 1993. For more information, please obtain:

- The IBM Programming Announcement for DFDSM, dated May 19, 1992 (number 292-25);
- The IBM ITSC "Red" book *DFDSM Presentation Guide* (GG24-3913-00); and
- IBM manual *DFDSM General Information* (GH35-0114-00).

HARBOR Data Backup Product

HARBOR has been designed to back up MS-DOS, Windows, OS/2 and UNIX workstations networked to an MVS mainframe using the most common network protocols (3270, TCP/IP and APPC).

Features include:

- automatic daily backup
- one time storage of identical files (such as PC-based software products);
- self upgrade; and
- file distribution.

The DFSMS Philosophy

HARBOR runs on MVS as either a started task or batch job. It is a VTAM application. It stores client data as a VSAM LDS on the MVS host. Each workstation or server, depending on the options specified, will have one HARBOR "set" of host LDS files. A set consists of an image data set, one or more incremental data sets, and one or more archive data sets. HARBOR dynamically allocates these LDS files on behalf of the workstation user or servers. Support is provided for DFSMS storage, management and data classes. The product also allows clients to specify high-level qualifiers for the data sets that it allocates. This enables HARBOR to fit into any storage management philosophies in existence (DFSMS, DFHSM or Sterling Software's DMS.)

Data can be backed up either automatically (server-based automation) or initiated by the server. One nice feature is the file exclusion. Some files (e.g., the Windows swap file), simply do not need to be backed up, period. An option to bypass this is provided.

Incremental backups are consolidated into the VSAM LDS. This means that a client "volume" can be easily restored without issuing complicated commands.

The price seems right in line with the industry: tiered, based upon the number of MVS mainframes and clients.

Summary

Perhaps it is being somewhat of an alarmist to worry about the demise of MVS-based systems. MVS licenses are remaining about the same as the previous year; the application caseload is still increasing. In addition, more DASD is being ordered and installed, and still larger CPUs are being developed and shipped.

But it still does not hurt to examine trends, to anticipate directions and to be prepared to respond effectively when change is inevitable.

Footnotes

[1] *These CICS applications must not use operating specific functions, such as MVS Internal Reader (INTRDR) job submission, to be portable. Care must be taken in the application system design to take these constraints into consideration if true portability is to be achieved.*

[2] *Some rework of the programs might be necessary, since some operating system environments (MVS is a prime example) do not implement C in the "standard" way.*

[3] *However, native CICS converts lowercase data to uppercase. For most applications, this might not be a problem. But for others, such as case-sensitive UNIX commands or TCP/IP host tables created by the CICS application, this could be a serious limiting factor.*

[4] *VM, and of course, mainframe UNIX systems, could also be used to do some or all of these data functions. To illustrate all possibilities and permutations, one could write volumes. For brevity, and for illustration, not to mention treading in areas that the author has no real expertise, the example operating system shall be MVS.*

[5] *For more information on these client NFS commands, please refer to the IBM manuals applicable to your version and release level (for example,* MVS/DFP Version 3 Release 3 *would be* Customizing and Operating the Network File System Server *IBM Manual, SC26-4832).*

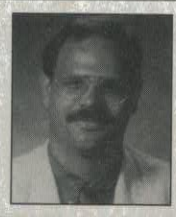
[6] *Data are stored as either a MVS LDS or as a VM physical minidisk.*

[7] *To do this, either a TCP/IP socket program or an APPC connection needs to be established between the mainframe application and DFDSM. This is the topic of another article.*

[8] *Indeed, DFDSM looks quite similar in implementation to DFSMS and DFHSM. Knowledge gained from implementation of one can quite possibly translate directly to implementation of the other, sharply reducing the learning curve and implementation operational problems sometimes inherent in other product usage.*

[9] *Prior to individual client activation, the installation must "register," set up, authorize, then provide each client site specific connection and customization information.*

/*



Stephen Force is an independent consultant based in Michigan.

Was this article of value to you? Please let us know by circling Reader Service No. 82.