Abstract

This paper discusses connectivity solutions for VSE mainframe environments. Connectivity issues must be addressed for VSE to be a viable operating system. There are two primary methods of communication between VSE and other platforms: terminal-based communication and program-to-program communication. Examples of terminal-based communication protocols include 3270, High-Level Language Applications Programming Interface (HLLAPI), and asynchronous communication (ASYNC). IBM's Advanced Program-to-Program Communication (APPC) provides VSE with peer-to-peer communication capabilities. The SAS® System supports all of these communication protocols and provides the VSE community with a variety of connectivity solutions. This paper focuses primarily on program-to-program communications, as the most advanced form of communication.

Introduction

Terminal-based communication is designed to display data on a terminal. Using this type of communication for connectivity is common in VSE environments, but it has its limitations. Drawbacks include enforcement of data into a screen image format, even though it is not intended for users to read from the screen; character set encoding must be facilitated; and transmission packet sizes are restricted to screen size, which is generally around 2K bytes. HLLAPI is a standard 3270 programming interface that uses the LU2 protocol of System Network Architecture (SNA) to communicate with a mainframe host. Emulator High-Level Language Applications Programming Interface (EHLLAPI) is identical to HLLAPI, but EHLLAPI is used in references to Extended Edition OS/2®. 3270 protocol refers to non-HLLAPI interfaces that also use SNA's LU2 protocol. Asynchronous communication uses phone lines or RS232 cabling as its medium for data transmission. Even though ASYNC is inexpensive, it is the least desirable form of terminal-based communication. It is relatively slow and requires
much more handshaking, error checking, and recovery on the part of the application software.

Program-to-Program communication is designed for the interaction of two programs or processes. In the mainframe environment, the two available program-to-program protocols are APPC and Transmission Control Protocol/Internet Protocol (TCP/IP). This type of communication method is more advanced in that it adheres to a client/server model, whereas terminal-based communication exhibits a master/slave relationship. For many years, SAS Institute has supported VSE connectivity with terminal-based communication solutions. For sites running VSE/ESA and VTAM V3.2 or later, the SAS System takes full advantage of APPC and provides more options for distributed processing. APPC in IBM's SNA is implemented by Logical Unit Type 6.2 (LU 6.2) and allows peer-to-peer communication between applications in an SNA network.

Table 1.1 illustrates the communication access method(s) available on each platform for connecting to a VSE server using the SAS System.

<table>
<thead>
<tr>
<th>Platform</th>
<th>VSE</th>
<th>MVS</th>
<th>CMS</th>
<th>OS/2</th>
<th>Windows 3.x</th>
<th>MS-DOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>APPC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>APPC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>APPC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>APPC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1.2 illustrates the communication access method(s) available to a VSE client for connecting to other platforms using the SAS System.

<table>
<thead>
<tr>
<th>Platform</th>
<th>VSE</th>
<th>MVS</th>
<th>CMS</th>
<th>OS/2</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>APPC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>APPC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>APPC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1.2
Available connections from VSE client

Advanced Program-to-Program Communication (APPC)

APPC is IBM's premier solution to peer-to-peer connectivity. It is known as Logical Unit Type 6.2 (LU 6.2), which evolved from its predecessors, LU 6.0 and 6.1. APPC enables high speed communications between programs on different computers by providing a common protocol in which these programs, if they adhere to the rules, can communicate or converse. APPC software is available for many IBM and non-IBM platforms, either as part of the operating system or as a separate software package. For the VSE/ESA operating system, it is a subset of the Virtual Telecommunications Access Method™ (VTAM), which is part of the operating system.

Transaction Programs and Conversations

The part of the application program that utilizes APPC communications (exchanges of data) is referred to as a transaction program or TP. The APPC access method component of the SAS System is a transaction program. For a transaction program to be able to request APPC services, a set of commands is needed. This set of commands is known as the protocol boundary. The protocol boundary isolates transaction programs from the low-level details of network operation, thus making an efficient, less burdensome application programming
environment. Figure 2.1 illustrates a logical representation of the transaction protocol boundary. Under the VSE operating system, the command set is executed in the form of VTAM macros (APPC-CMD), rather than verbs defined by LU 6.2.

The communication between two transaction programs is called a conversation. The initiator of the conversation is the client program, and the target of the conversation is the server program.

**Sessions**

Before a transaction program can start a conversation with a partner transaction program, a session must first be established. A session is a physical connection between two logical units. The two logical units in this case are the local and partner APPC LUs. There is a one-to-one relationship between a conversation and a session. In other words, a session can only support one conversation at a time. However, multiple sessions can simultaneously co-exist between two LUs (parallel sessions). Therefore, multiple conversations can simultaneously coexist between two transaction programs. Unlike a conversation, a session is long-lived. Once a session is no longer needed, it is placed into a session pool to be used later. By reusing sessions, the overhead of session start-up is avoided.

Before APPC sessions can be established, LU6.2 modes must be defined. A mode is defined to VTAM as an entry (MODEENT) in the mode table (MODETAB). This entry contains session parameters that determine how any two LUs using this mode will communicate.

**The Network**

As previously mentioned, APPC provides the necessary interface between transaction programs and the network. These network functions, utilized by APPC to communicate with partner applications on different computers, make up what is called Advanced Peer-to-Peer Networking™ (APPN). Discussion of APPN is included to complete the topic of APPC communications. Briefly, there are three types of APPN networking nodes: Low-Entry Networking Node (LEN), End Node (EN), and Network Node (NN). LEN nodes are the simplest node type, in that they are capable of peer-to-peer communication, but are unable to make direct use of APPN functions. NN nodes are the most intelligent node type. They know about other nodes in the network, and can make intelligent decisions in routing information through the network. VSE fits into APPN by using VTAM to implement networking functions. Host networks are called subareas. These existed prior to APPN. From the perspective of APPN, a subarea is viewed as a low-entry node.

302
Client/Server Capabilities Provided by the SAS System under VSE using SAS/CONNECT® Software

Connectivity between different computing environments is very important today, but it must be flexible enough to allow users to develop intelligent client/server applications. The SAS System provides compute services and remote data services to enable the development of distributed applications. Compute services enable access to remote computing resources, while remote data services enable access to remote data, regardless of where those data are stored. With compute services and remote data services, the SAS System provides the flexibility to design and tailor a distributed application to meet the specific needs of the user.

Compute Services

Compute services are used to maximize computing resources within your organization. These enable a local application to gain access to remote computing resources, including hardware, software, and data. This remote access allows a client running on a local machine to take advantage of remote hardware resources, such as a CPU or plotter; use software that is not locally available; and execute against remote copies of data. Deciding where to execute (local or remote) is now under the user's control. The ability to decide where to execute incorporates application segmentation which enables the user to choose which part(s) of an application will execute remotely and which part(s) will execute locally, thereby, maximizing resources for individual applications. If remote computing is chosen, the results of the remote processing are conveniently returned to the local session with only minimal network traffic. Benefits of compute services include maximizing performance by choosing the machine where an application will execute; gaining access to software not available on the local machine; gaining access to remote devices, such as plotters and printers; and maintaining only one copy of the data.

Sample SAS Program Illustrating Compute Services

All of the processing is executed on the remote platform with the data also residing on the remote machine. By specifying the graphics device driver to be GRLINK (goptions device=grlink), the remote platform will transfer the resulting graphical chart to the local machine.
Remote data services enable access to data stored in a remote environment. These services can be further divided into data transfer services and remote library services.

Data Transfer Services

Data transfer services provide a method for moving a physical copy of data from one machine to another. This technology plays a useful role in certain situations by allowing work to be off-loaded from a remote system. Once data have been transferred, the local system can use these data without being impacted by network failure. Data transfer services provide no advantage when significant network traffic occurs during the transfer of large volumes of data; when security prevents having multiple copies of the same data; or when the remote data are updated too frequently.

Sample SAS Program Illustrating Data Transfer Services

The data that are needed for the graphical computation are downloaded to the local machine. Once the local copy of the data has been created, all other processing is accomplished locally.

Remote Data Services
Remote Library Services

Remote library services have recently been added to the SAS System to provide transparent data access to remotely stored data. These services are only available with peer-to-peer protocol access methods, such as APPC. Remote library services allow a local user to execute in his own environment, while transparently accessing data that are stored on a remote machine. This functionality provides the ability for a locally executing application to read and update data stored in a remote library data base. Remote library services are useful when the quantity of remote data needed by the local application is small; when the remote data are frequently updated making data transfers less desirable; or when maintaining multiple copies of data are not desirable. Another benefit of remote library resources is that it permits the local graphical user interface (GUI) to display the remote data. Remote library services should not be used when the volume of data are large, or when an application must make multiple passes over the same data. Instead of making multiple passes over the data, it may be more efficient to transfer a copy of the data to the local machine so that data transfers occur only once.

Sample SAS Program Illustrating Remote Library Services

All of the processing takes place on the local machine; however, the data remain on the remote platform. The SERVER option on the LIBNAME statement specifies where the remote data lives.
Demonstrate REMOTE LIBRARY SERVICES:

All of the graphics computations will take place locally, but the data still resides on the remote VSE machine.

Get a handle to remote VSE data

libname vsedata 'data1.saslib'
   server=appcvse;

SAS/SHARE software is made up of two procedures and an engine:

- **SERVER** procedure controls and executes input and output requests to SAS files on behalf of other SAS sessions (users and administrators).

- **OPERATE** procedure reports on and controls all servers’ resources, such as server libraries and server users, and terminates the server’s execution.

- **REMOTE** engine enables a user’s SAS session to access shared data by communicating with a SAS server.

With a peer-to-peer access method such as APPC, SAS/SHARE users and administrators can communicate with a SAS/SHARE server that resides on the same or different VSE machine. In addition, users and administrators can transparently access servers that reside on entirely different, heterogeneous systems such as OS/2 users accessing a VSE server or vice versa.

The following pictorial illustrates SAS data library access provided by a SAS/SHARE server in a homogeneous VSE environment.
Conclusion

Connectivity between mainframes and other platforms, especially workstations, is in high demand today. Peer-to-peer protocols such as APPC pave the way for developing distributed applications in a SNA network. The SAS System provides an adaptable and easy-to-use interface to APPC that allows VSE users to develop intelligent client/server applications with systems such as MVS, CMS, and OS/2. For the future, SAS Institute has targeted the AIX®, UNIX, and Windows 32-bit (Win32s, Windows NT™) operating systems to provide a SAS software interface to APPC. At this time, the Institute is also investigating the ability to interface a TCP/IP protocol access method to the SAS System on VSE, which will allow VSE users to develop peer-to-peer distributed applications in a UNIX network. OpenConnect Systems provides the TCP/IP solution for the VSE operating system, therefore the SAS System must interface directly to their TCP/IP socket access method. SAS software provides a variety of connectivity solutions to the VSE community and eliminates barriers for VSE users to the outside world.

Acknowledgements

SAS is a registered trademark of SAS Institute Inc., Cary, NC USA. SAS/CONNECT and SAS/SHARE are also registered trademarks of SAS Institute Inc. OS/2 and AIX are the registered trademarks and VTAM and APPN are the trademarks of IBM Corp. Windows NT is a trademark of Microsoft Corporation. OpenConnect Systems is a registered trademark of OpenConnect Systems Inc. All other tradenames referenced are the registered trademarks or trademarks of their respective companies.

References