Accessing Applications from Multiple File Servers in a Computer Network

J.S.J. Daka and [1]J.S. Isaac
Department of Electrical Engineering, University of Botswana, Botswana
[1]CSIR-Mikomtek, South Africa

Abstract: This study proposes a technique of accessing applications from multiple file servers in a computer network that caters for a large user community. The technique is a step towards creating a computer network system that is simple to implement, scalable, easy to manage, transparent to the user and suitable for a university-type environment. The proposed technique has some features of a distributed system, but is not as complex. It is built around a very popular commercial network operating system called NetWare.

Key words: Application Access Problems, Computer Network, File Server, Distributed Systems, Virtual Machine, Database Server, LAN, Operating System, Transparency, Scalability, Security, Compatibility, Technique, Sub-network

INTRODUCTION

A computer network is an interconnected collection of autonomous computers. The network allows sharing of applications, data and devices. To circumvent hardware failures all files can be replicated on a few computers. This makes a computer network reliable. Users on the network can work together on projects though physically separated by distance. Computer hardware is becoming cheaper every year. An increase in the demand for computing resources can easily be met by adding more computers to the network. A computer network covering a small geographical area is known as a Local Area Network (LAN) e.g. connection of all computers on a university campus. LANs can be interconnected to form a Wide Area Network.

Current LANs are in two varieties: the workstation/server and the serverstation type. The workstation/server type classifies computers on a network in two mutually exclusive categories: servers and workstations. Workstations have very little interaction with each other and do not provide any service to other workstations. All workstations access the same set of servers e.g. Novell's NetWare networks. In the server type, every computer on the network can act as a server for the others. Accordingly, each computer has both server and workstation software. The serverstation type is commonly known as a peer-to-peer. In the serverstation type, management of user accounts and data becomes harder as the network grows in size. Eventually, some computers are designated as servers. At that stage, it will imitate the workstation/server type of LAN.

The main problem with current computer networks is that they are not transparent. The user logs into one computer explicitly and uses that computer only, until remote login to another computer. The user handles much of the network management. This is difficult for inexperienced network users. Making the network transparent removes this complication.

Distributed Systems: A distributed system is a special type of network whose software gives it a high degree of cohesiveness and transparency [1, 2]. The existence of multiple autonomous computers is transparent to the user. The operating system finds the best processor for a job and does all the management of file traffic to and from that processor. The user does not explicitly login to computers where the applications or files reside and does no network management.

A distributed system includes distributed processing; giving it a possibility of parallel processing. The operating system creates an impression of a single computing environment although there are multiple processors on the network. Therefore, the software for a distributed system is very complex. In making things easier for the user, the complexity has been shifted to the network management software.

A distributed system is more useful than a plain computer network because of the increased transparency and distributed processing. However, this same capability makes it very complicated and difficult to implement.

Virtual Machines: A degree of network transparency can be achieved by presenting the user with the illusion of being on a very powerful stand-alone computer that contains all applications, data files and peripherals. This powerful stand-alone computer does not exist; therefore, it can be termed a Virtual Machine (VM). This term classifies systems that lie between a plain computer network and a distributed system, with respect

to complexity. A VM does not have distributed processing capabilities. In a VM, an application could run on another computer but the load is not split between many processors. This difference distinguishes a VM from a distributed system. Software that presents a VM to the user can therefore be simpler than the software for a distributed system. VM software can be built around existing operating systems. Thus, a network system that is simple to implement can be devised. The choice of a good network operating system eases the design and increases the acceptance of a VM.

NetWare: Novell's NetWare versions 3.10 and 2.15 were selected as the base network operating systems for this research. These systems are based on the workstation/server type of relationship and provide a fairly transparent interface to users.

The workstation/server relationship of NetWare enhances security and provides an easy to maintain system. All files are assumed to reside on file servers. Compared to the total number of PCs on the network, the number of file servers is quite small. All installation of applications is done at file servers. Legitimate users of the network are defined and authenticated at file servers.

NetWare hides the existence of multiple sub-networks from the average user. Logging in to a server on another sub-network is no different from logging in to a local server. The file server's hard disk, remote printers and other devices appear as local devices on a PC. These features hide some of the complexities of a computer network. However, logging in to a file server is not transparent to the user.

APPLICATION ACCESS PROBLEMS IN A MULTIPLE FILE SERVER NETWORK

With the availability of cheap and powerful PCs, it is now attractive to base a campus network primarily on these. Generally, PC sub-networks are based on a commercial PC network operating system such as Novell's NetWare. Usually, there is only one file server on this PC sub-network and users are provided access to a few applications on it. If the number of users, applications and file servers on the network increase, then users and network managers encounter problems relating to applications access. These problems partly arise because PC networks traditionally catered for the business sector. In the business sector it is not common to find a large user community, whose membership is constantly changing, nor is it common to find most users needing access to all file servers. Normally, a department in an organisation would have a few of their personnel defined on a file server containing a few relevant applications. Very few users would need access to more than one file server.

In a university environment where a network is based entirely on PCs, there would have to be many file servers on it due to a large number of users and applications. Most users would need access to all file servers to avoid unnecessary duplication of applications and data on the network. Problems in enabling users access information or applications from multiple file servers are encountered where users are created on file servers. A transparent network file system makes it easier to solve them. Accessing an application includes locating it, setting up the MS-DOS 'path' variable and other environmental variables and then starting it.

To access an application, the user must know on which file server it resides and log in to that server. Furthermore, the user must know how to access the application i.e. in which directory it is in and how to start it. If the location of an application is changed then the user will have the same problems all over again.

To illustrate the second problem, consider a 3000 user base with 8 file servers in a Novell network. Also, suppose that each user must be able to login to every file server and separate categories of users should have different access rights to applications and data. The file server administrators can employ one of two schemes to achieve these goals.

In the first scheme, the user population is divided into 8 groups. Each group is then created on one file server. Each user is given two usernames and associated passwords. The first username is unique to the user but the user can change the associated password. The second username is common to all users in the same category. The user cannot change the password for the second username. With the first username, the user can access applications the user is entitled to on the 'home server'. The user must also have access to personal data files. With the second user name and password, the user can access applications that reside on other file servers. All users within a particular category must know the password related to the username of that category. If the server administrator changes the password, this needs to be communicated to all the affected users. This scheme is therefore not secure.

In the second scheme, the 3000 users can be created on each of the 8 file servers. This is the standard method employed in most Novell networks. This scheme has better security than the former scheme. The problem is that the labour requirement in creating and maintaining 3000 users on each file server is large. In addition, users must remember the names of all file servers, their usernames and passwords for each file server. It is possible for a user to have different usernames on file servers due to typing mistakes when creating user accounts. This creates confusion for the user. The likely-hood of a user forgetting the password for one of the file servers is also very high. This increases work for the server administrators to set things right and to reset passwords.

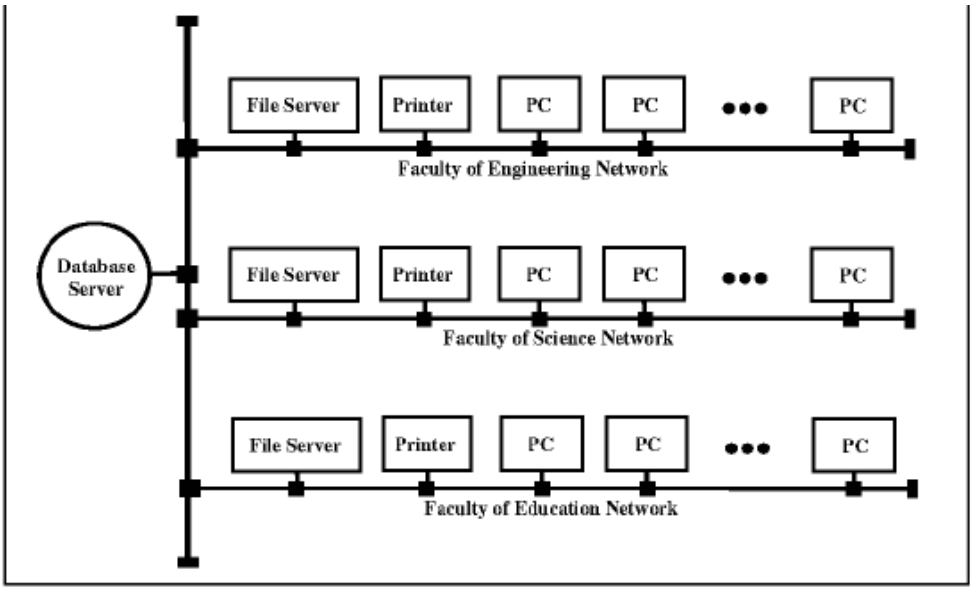

Fig. 1: Typical University Campus Sub-network Layout

SYSTEM COMPONENTS

In addition to normal hardware components found in a Novell network, the VM has a database server. A typical layout of a campus VM is shown in Fig. 1. The VM consists of a sub-network with at least one file server. The sub-networks should be connected to the backbone using bridges since most network traffic is expected to be 'local'. Bridges also localise problems to a sub-network.

Novell's NetWare consists of two main parts [2,3]. The first part is an operating system for a PC that makes it act as a file server of a network. The other part is the NetWare shell that runs on all the other PCs enabling them to communicate with each other and file servers. The shell also makes the file server's hard disk and other devices on the network appear as local devices.

Operation of the VM is based on a database server. The VM system requires that the system database be easy to maintain and manage. It is desirable to concentrate all data in a single database that can be used by all PCs. The database for the VM consists of the NetWare database on all file servers and the VM database on a PC VM database server. If the VM database server fails, the VM system is brought down. Resiliency to such failure is achieved by replicating the VM database on various servers on the network. The database contains the following information:

* User information on access rights, passwords, etc. This is a distributed database located in the bindery [4] of all file servers and is non-replicated.
* Software location on file servers and how it is started, etc. This is on the VM database server computer and is replicated on all database servers.
* Application files held on a distributed database located in the file server's directory structure and replicated on a few file servers.
* The menu contents, this is a distributed database and is replicated on all VM database servers.

The VM database server provides the information to requesting PCs. All requests for data from the PCs are served on a first-come-first-served basis. When updating the VM database the administrator runs a program to bring down all VM database servers in sequence, loads a new VM database and then restarts the servers.

To enable the operations of the VM, 'VM_MENU.EXE' is executed on a PC when the user types 'VM'. The program communicates with the VM database server to present menus to the user. It also checks the bindery of the user's 'home' file server to determine the user's access rights to applications and logs users to appropriate file servers with the desired applications. The VM programs that a user is aware of are: 'LG.EXE', 'VM' and 'LGO.EXE' to login, access the menu system and logout respectively.

From the user's viewpoint, the VM shown in Fig. 1 will appear as shown in Fig. 2. The existence of many sub-networks and file servers is hidden by a combination of NetWare and the VM software. The VM software interacts with the file-servers and the VM database server to create the illusion (Fig. 2). An inexperienced

computer user might not even be aware of the network or the file server.

THE APPLICATION ACCESS TECHNIQUE

The basic VM model involves a user logging into their 'home' file server from any PC on the network. The 'home' file server handles user authentication. Whenever the user wants an application, a menu program found on the search path of the 'home' file server is invoked.

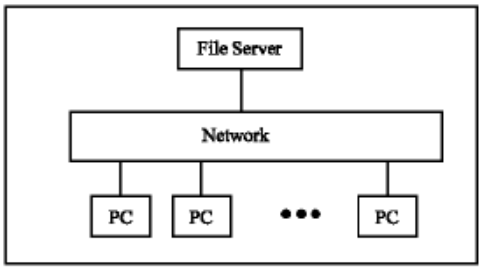

Fig. 2: Network Presentation to the User

This program contacts a VM database server from which menus are presented. The user selects the required application from the menus. The VM menu program contacts the relevant file server that has this application. Subsequently the VM menu program terminates and the application is started.

All communication between the VM menu program and the VM database server use NetWare Application Programming Interface calls [5] for MS-DOS. The applications are started using batch files that set all necessary environmental variables and search mappings. When the application terminates all environmental variables are reset and search mappings are deleted.

Users in a particular VM category are given access to all file servers using a common login name and password. The login name and password are obtained from the VM database server by the VM menu program and the actual login/logout to other file servers is performed on behalf of the user by the VM menu program.

Transparency: Transparency is defined as the concealment of separation from the user and the application programmer, so that the system is perceived as a whole rather than a collection of independent components. Eight forms of transparency have been identified [6]. For the VM, these include the following:

Concurrency Transparency: This is provided by NetWare on concurrent use of shared data, for read-only. For read-write data, a non-transparent file locking mechanism that prevents interference is provided. NetWare also provides the Transaction Tracking System (TTS) [5, 7] that safeguards against inconsistent data during updates.

Replication Transparency: All VM database servers and replicated applications on file servers are transparent to the user except for data that is not part of the VM.

Failure Transparency: If a file server fails on the network, it affects all the users on that file server. However, if a VM database server fails, then users are transparently connected to another VM database server. The TTS provides a degree of failure transparency for files. Other forms of transparency in the VM are discussed below.

File Service: A major feature of the NetWare operating system is the transparent access to files offered to PCs. The transparency allows programs for a single computer environment to operate on a network. The file system on a file server is presented to a PC user as an MS-DOS file system, by mapping files on the file server to a local file system. The local file system is responsible for accessing local disk drives. If the data is on a file server, then the NetWare shell is used. The MS-DOS file system keeps track of which device driver handles which disk drive. Network requests are handled by 'NETx.COM' that decides where the data is located and makes an appropriate request to 'IPX.COM', which communicates with the relevant, file server.

File Server Naming: The file servers on the network are given unique two letter names, the shortest name length permitted by NetWare [8]. This is done to reduce the number of characters that a user types to login to a file server. In a single file server setting and the original NetWare environment, the user would type [9]: LOGIN USERNAME. In the VM, the user types: LG SN/USERNAME, where SN is the two-letter server name.

In the NetWare environment, servers are given names that users can easily remember. In the VM, this is not necessary since the user need not know the names of other file servers. A two-letter code to differentiate file servers is sufficient. The code allows 676 unique file server names, sufficient for all file servers on a typical network. Users are assigned login names of the form SN/USERNAME. In this way, the average user is not aware of the name of their 'home' file server.

Each user is created on only one file server. The user's login name should be unique on that file server. The login name assigned to a user is of the form <server name>/<user name>, where only the <user name> is required if the PC is attached to the 'home' server. Including the server name in the user's login name violates location transparency. If the user is moved to another file server, the login name also changes, but moving user accounts from one file server to another

rarely occur. Users can login to their file server and are authenticated from anywhere on the network. The <user name> portion of the login name does not have to be globally unique. Other existing NetWare applications such as email can be used without any modification because these will see a normal Novell network. As long as the user's account is not transferred to another file server, the user can always be accessed using a login name irrespective of location on the network.

Batch files that start applications should be given unique names. All the file servers that have a particular application can have a batch file with the same name to start it up. This ensures that a user does not inadvertently start up the wrong application, when attached to more than one server. Since the user starts up the application using a menu, the name of the batch file and its location is unimportant. Through a combination of the application database directory and the local PC menu, location transparency for applications is achieved.

Scalability: The computer network can easily be extended, as more PCs become available. However, this is limited by the networking technology, the configuration used [10] and NetWare operations. Software to present the VM environment on a PC is obtained from a file server on logging in. Therefore, updates to this software are made available to all PCs.

All application and data files are stored on file servers. The node address of the PC is unique. This feature reduces labour requirements in extending the network. Adding a new PC to the network is by merely connecting it and loading the MS-DOS system files and NetWare files.

Dividing the user population into groups according to access rights, i.e. the VM categories, helps create a scalable system. Users are created on only one file server, but have access to all file servers via the VM category.

Security, Compatibility, User and Station Mobility: NetWare does authentication of the user at login. The user is assigned a VM group with some access rights to applications at login. When the user selects an application from the VM menus, an attachment to the relevant server with the selected application is made. The VM group determines the name that the PC software uses to attach the user to the other server. The software also supplies a password with more than 125 characters. The password prevents unauthorised access to resources.

Servers on the network are physically secure. They run software that users cannot modify and do not run user programs. However, though PCs are not secure, malicious actions by individuals on one PC do not affect users on other PCs.

The VM programs have been written for the INTEL 8086 microprocessor, the lowest in the series. Since higher processors are downward compatible, they can execute 8086-instruction code. Software that communicates with the VM database server only needs protocol compatibility.

User Mobility is a feature supported by NetWare, hence the VM. This is because the VM software builds on the capabilities offered by NetWare. Any user can obtain system services from any PC on the network and gain access to their private files.

System or network services are independent of the address of the station. The PC can be moved to any locality on the network without affecting access capabilities. When a PC is moved to another sub-network, its network address will change, but this is assigned at boot time.

Ease of Application Access: The VM PC menu program removes the difficulty of locating applications in a multiple file server network. When the user selects an application from the menu, a connection to a file server with that application is made and the application is run via a batch file. The batch file prepares the MS-DOS environment and creates search mappings that permit an application to locate the files required for its execution.

Information about applications is stored in the VM database. Thus any changes to location and type of applications need only be noted in the VM database. This then becomes available to all users through the PC software. The user need not know where the application is located or how to access it.

Functions of the VM Database and File Servers Administrators: The VM database administrator creates, manages the VM database and co-ordinates with the file server administrators to determine the VM groups and access rights for applications. The administrator sets the base passwords for various categories of users in the VM to login to other file servers. The administrator also selects the PCs on the network that act as VM database servers and invokes the server program. When the VM database is changed, the administrator remotely reloads a new copy of the database on the servers.

The file server administrator's function is similar to that in the single file server NetWare environment [4]. The administrator copies the necessary VM programs to the file server and creates special groups and users for VM access to that server by non-resident users. When an application is added or deleted from the file server, the VM database administrator must be informed to make the necessary changes to the VM database.

The VM database administrator should be contacted to see whether an application has already been installed in a file server elsewhere on the VM, before installing it. The VM database administrator and the file server administrators determine the application access rights assigned to the VM groups. This avoids users being assigned different access rights based on the file server. The file server administrator can assign additional rights to some users, by creating a group on the file server that contains these users.

Local Sub-network Control: A degree of autonomy at the local sub-network level is necessary on a university campus. However, each faculty should be able to decide how its resources are used. To allow for this, the local system manager sets the trustee directories and queues of the special users accordingly [4]. In this way, all local users can have greater access to applications and other resources than users resident on other file servers.

CONCLUSION

With recent trends of PCs becoming cheaper and more powerful, it is possible to have a network consisting entirely of PCs. The study has presented problems of accessing applications from multiple file servers in a Novell network. A technique using a VM system to ease application access has been proposed. The VM system can take advantage of the benefits of a PC network and NetWare. The value of the application access technique becomes apparent when there are many users and many file servers on a network.

ACKNOWLEDGEMENTS

The research reported in this study was done in the School of Engineering and funded by the University of Zambia.

REFERENCES

1. Coulouris, G.F. and J. Dollimore, 1989. Distributed Systems: Concepts and Design. Wokingham, England, Addison-Wesley.
2. Halsall, F., 1992. Data Communications, Computer Networks and Open Systems. 3rd Edn., Addison-Wesley, USA.
3. Sloman, M. and J. Kramer, 1987. Distributed Systems and Computer Networks. Englewood Cliffs, N.J. Prentice-Hal.
4. Novell, 1990. Incorporated, Supervisor Reference Set for SFT/Advanced NetWare. Provo, Utah, Novell, Inc.
5. Novell, 1990. Incorporated, NetWare C Interface-DOS. Vol. I and II, Austin, TX: Novell, Inc.
6. Walker, B.J. and G.J. Popek, 1989. A transparent environment. BYTE., 14: 225-233.
7. Novell, 1990. Incorporated, Maintenance/Upgrade Set for SFT/Advanced NetWare. Provo, Utah: Novell, Inc.
8. Novell, 1990. Incorporated, Installation Set for SFT/Advanced NetWare. Provo, Utah: Novell, Inc.
9. Novell, Incorporated, User's Manual for SFT/Advanced NetWare. Provo, Utah: Novell, Inc.
10. Novell, 1990. Incorporated, Supplements. Provo, Utah, Novell, Inc.