DISTRIBUTED COMPUTING ENVIRONMENT

ABSTRACT

The high volume of networked computers, workstations, LANs has prompted users to move from a simple end user computing to a complex distributed computing environment. This transition is not just networking the computers, but also involves the issues of scalability, security etc. A Distributed Computing Environment herein referred to, as DCE is essentially an integration of all the services necessary to develop, support and manage a distributed computing environment. This term paper discusses the three important issues addressed by DCE in detail, Remote Procedure Calls [IRPC], Distributed File Systems [IDFS][OSF91] and Security [Isec][OSF92].

1 INTRODUCTION

The present day computing industry depends on the efficient usage of resources. So instead of duplicating the resources at every node of computing, a remote method of accessing the resources is more efficient and saves costs. This gave rise to the field of distributed computing, where not only physical resources, but also processing power was distributed.

Distributed computing was driven by the following factors.

a) Desire to share data and resources
b) Minimize duplication of functionality
c) Increase cost efficiency
d) Increase reliability and availability of resources.

The Open Source Foundation's DCE (OSF DCE) has become the De facto standard for DCE applications and has the backing of IBM, COMPAQ, HP and the likes [OSFInter92].

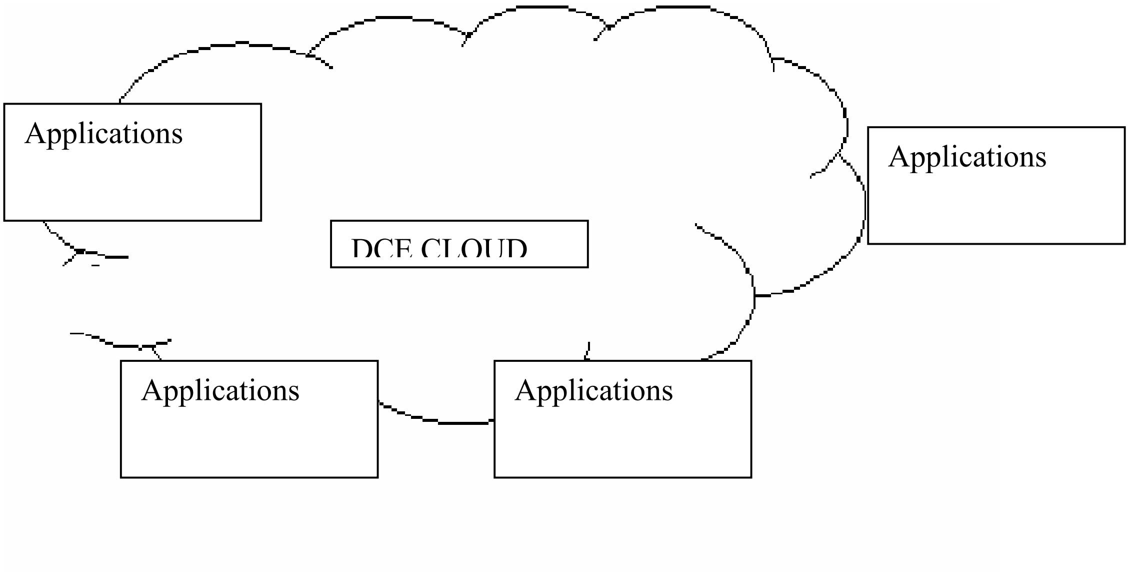
1.1 Why DCE?

When an organization migrates from networked computing to distributed computing a lot of factors are to be taken into consideration. For example replication of files gives rise to consistency problems, clock synchronization becomes important, and security is a bigger consideration.

A DCE addresses all these issues by providing an integrated set of cross platform, comprehensive services which aids in the development and application of distributed applications.

1.2 DCE Architecture

The following diagram gives a simple view of the DCE architecture.



The DCE cloud refers to the distributed computing environment tools that facilitate distributed computing.

The DCE Cloud Consists of the Following Components [IAIX]

a) Distributed File Service
b) Distributed Time Service
c) Security Service
d) Cell Directory Service
e) Threads Service

All these services are achieved by the use of Remote Procedure calls (RPC)

1.3 Properties of DCE

A DCE Provides a Global Computing Environment, which can interoperate with other services like DNS and X.500 [OSFInter91]. This sort of global interoperability provides the much-needed interface for Write Once Run Anywhere Applications. Also the suite of components is completely integrated and interoperable, which facilitates the networking of two systems for processing even though they have different hardware and software configurations.

1.4 DCE Cells

A collection of machines, users and resources that are a part of a group and having their own directory service and security service can be called a DCE Cell. In an organization there may be a large number of cells, say one for each department.

The configuration of DCE cells and the setup of the different DCE components are outside the scope of this Term Paper. Sections 2, 3 and 4 discuss the three major components of DCE namely, the Remote Procedure calls, Distributed File Systems and Security. Section 5 discusses the advantages of DCE.

2 DCE REMOTE PROCEDURE CALLS

The Remote Procedure Call (RPC) in a DCE is the facility that lets users make remote procedure calls and connect to another system on the DCE. The application programmer is essentially hidden from the fact that it is a remote procedure call, by the components of RPC.

2.1 Components of RPC [IRPC]

The RPC components are

a) The Interface Definition Language and its Compiler:

The skeletons and stubs are created by the IDL and then compiled by the IDL compiler. The server stubs replace the remote part of the procedure call, and at the server the skeleton replaces the client.

b) Runtime RPC Library:

The RPC runtime library is actively involved in the sending and receiving of remote procedure calls and finding the necessary server services and communicating between the client and the server.

c) Secure RPC Components:

The Secure RPC components work along with the security APIs to provide authentication and authorization for the remote procedure calls.

d) Name Service Independent APIs:

The Name Service Independent (NSI) APIs help in locating the right server to process the request. It is integrated into the directory services Component to facilitate the Association and Binding of the Client to the Server.

e) UUID Facilities:

This UUID Stands for Universal Unique Identifiers. This is useful to generate UUIDs, to uniquely identify each server and client on the DCE.

2.2 Specifications Conformance

The DCE Architecture conforms to the Network Computing Architecture (NCA) [IRPC] specifications. Transport independence and hence the OS independence is achieved as the NCA supports both connection oriented as well as connection independent protocols.

2.3 Facilities Supplied by the RPC

The following are the facilities that are supplied by the DCE RPC which are shielded from the application programmer [IRPC]

a) Security services
b) Use of the Directory Service to Find the right server for remote calls
c) Managing the data formats which are different in each cell of the DCE.
d) Management of messages for example its fragmenting and reassembly.
e) The communication protocols used. RPC can communicate over TCP/IP and UDP.

2.4 Communication Methodology using RPC

The Following are the commonly used communication methodologies in RPC.

2.4.1 Creation of the IDL File:

The interface for RPC is defined in the IDL file and not the actual procedures. The IDL file advertises the input and output of the Services offered by the remote server. The IDL file is written based on the server's procedure and then compiled using the IDL compiler. Compilation of the IDL file produces client and server stubs.

2.4.2 Client's View of the RPC:

The client is then provided with the Stubs generated by the compilation of the IDL file and it is incorporated into its procedure calls. A simple procedure call is now converted to a complex RPC over the network.

2.4.3 Server's View of RPC:

The server side has the subroutine to perform the function as given in the IDL. The server receives the parameters passed through the IDL and performs the procedure execution and sends back the results as published in the IDL to the client.

2.4.4 Binding:

The client finds the appropriate server to send the remote call by looking up the server's services. This is called Binding .The server when it starts must advertise the services it provides by registering with the directory services. The client then accesses the directory service to find about the server, which offers its services and then addresses that server.

2.5 Advantages of Using DCE RPC [IRPC][FS92]

The following are the advantages that are obtained by using the DCE RPC

Operating System Independence:

The RPC calls do not depend on the underlying OS's network calls mechanism

Machine Independence:

Even if the machines connecting through RPC are different, RPC can be successfully used as it provides the instructions in native format for both the client and the server.

Language Independence:

Any modern programming language can access the stubs and the skeletons that are produced by the IDL compiler.

Protocol Independence:

The server when registering with the DCE directory service explicitly states the protocol that it uses. Hence the clients can use that protocol or access a different server. The connection oriented and connection free protocols can be interchangeably used. More can be learnt on the protocol independence in [LOSF90]

2.6 Security Inbuilt in RPC

The secure RPC is called the Authenticated RPC. There are various levels of authentication. [Isec]

a) None – No Authentication
b) Connection – Authentication through encryption occurs at the first connection or handshake
c) Call Authentication – The first data packet which is sent to the server is authenticated
d) Packet Authentication – Each packet of data sent through the RPC Interface is authenticated

In addition to these levels packet integrity and privacy can be protected by the use of Cryptographic Checksums, [Misc] which is discussed in detail in Section 4 on security.

3 FILE SYSTEMS IN DCE

The OSF's Distributed File System herein referenced as DFS is based on the Andrew File system herein referenced as AFS [IDFS]. AFS was developed in the Carnegie Mellon University as part of the Andrew Workstation project and OSF DCE borrows a lot of ideas and features from AFS Version 4 to implement its DFS.

3.1 Stateful File Server

In DFS the file server keeps information about the clients and their cached copies of the files. Hence it is a stateful system. Stateless systems do not have this information about the clients. It has its own advantages and disadvantages. While the stateful systems are efficient in the reduction of the number of network messages it has a high overhead in tracking the files and logging them. The stateless file server is easy to manage as all the information is with the client but the need to register the changes at the server to propagate the information to all other clients' results in a large amount of Network Traffic.

3.2 Token Based System

The OSF's DFS uses a Token mechanism, where the clients who want to write to a file need a Write Token. When the server Hands the write token to the writer it revokes all the other read tokens that have been given to the other clients, which have cached the pages. Since the server is stateful it knows the clients who have cached the pages and notifies them that the copy in the cache is invalid. Then when the modifications are done the tokens are handed back to the clients.

3.3 Naming in DFS

The naming in DFS is uniform throughout the environment and it uses Global Name Spaces. This form of uniform file access is necessary to ensure the integrity of data. The DFS System adds a global filespace to a Namespace. That is the user can access his/her files on the DCE using the same name regardless of the physical presence of the user or file. This feature is made possible by mounting the user profile to his local filespace wherever the user logs into the DCE.

3.4 Components of a DFS [IDFS]

The DFS consists of the following components

a) Cache manager on DFS clients
b) File set Server and exporter that runs on the Serve
c) A File set location server that can run on any host.
d) Replication server to handle the replication.
e) Update server, which provides the distribution mechanisms of binary files to other DFS Servers.
f) Backup server

In addition to these some DFS Systems may provide gateways to access other Systems like the DFS-NFS Gateway. This gateway provides secure access to DFS Hosts from outside the DCE Cells.

3.5 Securities and Protection of Accesses

The security is managed by the DCE using Security protocols like Kerberos, which was developed at MIT. Security is a Two pronged problem Authentication and Authorization.

Authentication is the process of checking whether the user is really who he claims to be and the process of authorization is after authentication to check whether the user has permissions to access the resources. Authentication is handled by Kerberos and authorization is handled by the use of Access Control Lists.

Authentication is achieved by sending a secure Kerberos ticket to the Authentication Server. The server checks the secure ticket and send back another secure ticket to the client after authentication. Authorization is not inherently possible in UNIX based file systems and hence access control lists are used. A group level security is possible in UNIX based file systems but when individual user level security is to be implemented then ACLs provide that flexibility.

3.6 Replication

All of DCE DFS's Network Services are fully replicable. This replication is the key to prevent single node of failure. When one of the servers becomes unavailable then the DFS Clients automatically address the replicated Servers. This Replicated Servers are up to date with all the files as in the server so the clients can keep working on them until the Server is up again.

3.7 Availability

The availability of a file in DFS is a result of File Replication. A well designed DFS system is robust in its handling of file changes and backups. In normal Systems when the backups of the files are to be done by the administrator, the system has to be brought down. Then the backups are taken and the system is brought online.

In DFS the backups can be done from the replicated copies of the file without the need to bring the system down, thus increasing the availability of the files. Also since there is absolute Location Transparency the files can be moved around from disk to disk without the need to bring the system down. The filenames are maintained.

3.8 Recovery Mechanism

In normal UNIX based files systems when the server crashes the information is to be built from the ground up using logs of all the activity on the files. This is not a desired feature of a highly available file system as building it from the start using logs involves additional downtime.

In DFS high availability is guaranteed by the use of EPISODE ^{TM} file system. A log of all the disk operations is maintained and the recovery is done, not from the ground up but from the recent update and hence the downtime is considerably reduced and availability is increased.

3.9 Performance

The performance is increased heavily by caching large amounts of data and status information on the client. This reduces the number of requests to the file server and also the network traffic.

3.10 Interoperability

The DFS-NFS gateway, which is available in some DCEs, allows the NFS clients' access to the DCE using the protocol gateways supplied. The NFS Clients can also be part of the global name space used by the DCE.

4.1 Need for Security:

As more and more computers are networked the need for security becomes higher as only trusted users must use the services available on a network. Eavesdropping of these services and intrusion is a great cost drain on the network services. Unauthorized access to data is a potential disaster for any service. Hence the security of the DCE is a very important consideration.

4.2 Security Components in DCE [Isec]

a) Authentication Service to establish whether the user is really whom he claims to be

b) Privilege Service to determine if the user has authorization to access the necessary services

c) Registry Service to handle the cell's Security Service.

d) ACL Service the Access Control List Service to determine if the user has privileges to access the service

e) Login service to enable user logging using a password

f) Password Strength Service to determine whether the password set for a user account conforms to the password strength requirements

g) Audit Service to track the services accessed by the user and build a report in case of security breach.

4.3 Steps to authenticate the user

When the user logs in using his Username and password they are sent using encrypted messages to the Authentication server. The Security protocol used is the Kerberos Protocol. The Authentication Server checks with the Registry database to check the user's credentials and provide a ticket based on the user privileges after checking with the privilege service. The Privilege server sends an EPAC (Extended Privilege Attribute Certificate) to the user. The EPAC contains the user's access privileges to various resources on the DCE.

4.4 Authentication by Kerberos

The Kerberos is the security protocol used in DCE to authenticate users. It uses Secret Key Encryption to authenticate the user. Any entity on the DCE that needs to authenticate is called a PRINCIPAL. Secret messages are exchanged between the principals to authenticate. Confidential data is sent through the network using encryption and Decryption is possible only by the use of a secret key. Corrupted data is checked easily by the DCE by the use of Cryptographic Checksum techniques.

Kerberos provides secret key to the principals to decode the secure encrypted message sent over the network. If the site or the user possesses the security key then it means that the principal has been authenticated. Tickets containing the name and location of the principal are the credentials necessary to authenticate.

4.5 Data Integrity and data Privacy

The data integrity is protected by the use of checksums. The checksum is recalculated at the other end to see if the data was altered. Cryptographic Checksums are used, to prevent the updating of the data and checksum by unauthorized users. The data privacy is protected by the DCE RPC calls. The RPC calls use the trusted third party secret key encryption mechanisms to protect the data. A service called the Key Distribution Service (KDS) is used to hand out secure keys to both the principals.

4.6 Transparency

The authentication and authorization services are transparent to the user. The user just needs to set the level of security and the RPC mechanism takes care of the rest hiding the details from the user.

4.7 Authorization using ACLs

The authorization is done by the Access Control Lists. The Access Control Lists contain all the resources in the DCE and the principals and the level of access they have to the resources. Group level as well as person-to-person security level is possible.

5 COMPETING TECHNOLOGIES

Though DCE has been accepted as the De Facto standard adopted by IBM, HP, Compaq and other major players in the distributed computing technology, it is worth to look at other competing technologies here.

5.1 Microsoft's DCOM

Windows NT 5.0 Microsoft's Business Operating System (Also called Windows 2000), ships with a technology called DCOM (Distributed Component Object Model). DCOM relies on the RPC protocols defined by the OSF, which is the backbone for DCE. Other features provided are X.500 style

Directory Services called Active Directory, Kerberos based security model, Microsoft Transaction Server and Microsoft Messaging Queue. Security services in Windows NT 4.0 were based on the Secure Socket layer Protocol (SSL). But with Windows 2000 MIT's Kerberos has been introduced. Since every Windows NT copy ships with DCOM, it is the most widely used RPC protocol used today.

Also Windows NT ships with industry standard Protocols to implement Distributed Systems, so it can provide interoperability. This is an example of Middleware integrated with the Operating system.

5.2 OMG's CORBA

Common Object Request Broker Architecture (CORBA developed by the Object management Group ships with a protocol to write and invoke remote methods called IIOP (Internet Inter-Orb Protocol). Inbuilt in IIOP is a redirection facility to indicate to the clients that the requested objects were moved in the network. CORBA is a ground up redesign of the Distributed computing principles [LINC]. The security features in CORBA depend on SSL and not Kerberos, which is perceived as a weakness.

In addition to these technologies, Sun Microsystems's Java provides RPC services through a technique called RMI (Remote Method Invocation). Recently, CORBA has been integrated with Sun's Java.

6 REVIEW OF DCE

A comparison of the competing technologies reveals that DCE holds the best tools available to facilitate distributed computing [LINAS]. The integration of these services play a key part in DCE's dominance in the distributed computing environment over Microsoft's DCOM and OMG's CORBA. Security, which is a major consideration for distributed applications, is tightly integrated into DCE, by the use of Kerberos protocol.

The performance of DCE is another important metric, which makes it superior to CORBA. This may be an avenue where Microsoft's DCOM, may have an advantage over DCE and CORBA, because of the seamless interaction between the middleware and operating system [RECO98]. Performance leads to an even higher amount of imbalance when the middleware is integrated with the kernel.

DCE by means of file replication greatly increases scalability and availability. The performance of the DCE depends entirely on the performance of the RPC calls. A lot of study has gone into analyzing the performance of RPC calls. A research study by IBM summarizes the performance in the following points [Misc]

a) RPC time increases linearly with the size of the data.

b) The "packet integrity" security level slows the RPC, nearly doubling the total time for calls.

c) The "packet privacy" level incurs a several-fold increase in time.

d) RPC is slower than simply making socket calls directly.

e) The time spent on RPC overhead was a small fraction of the total processing time in a realistic business scenario.

g) RPC is pretty fast on Intel processors (running OS/2).

6.1 Disadvantages

With all these advantages, DCE has now faded from the market as CORBA and DCOM have become dominant in the Distributed Computing Environment [RECO98]. This is due to the fact that DCE doesn't support Object oriented languages. Developers using C++ or Java must spend more time to achieve interoperability. The security system of Kerberos cannot scale to very high user limits.

7 CONCLUSION

Though DCE was the de facto standard for middleware applications, its lack of support for object-oriented languages has been the greatest cause for DCE to lose market share. Hence the once promising technology is now used only for Legacy system support. Many new Distributed-computing ventures are started using CORBA or DCOM. Hence we can safely say DCE was a very good start to Distributed Computing.

REFERENCES

[OSF 92] An OSF White paper: Security in a Distributed Computing Environment, 1992

[FS92] Dietmar fauth and Shikong Shue; Remote Procedure call: Technology, Standardization and OSF's Distributed Computing Environment. 1991-1992

[OSF 91] An OSF White Paper: File Systems in a Distributed Computing Environment, July 1991

[OSFInter92] An OSF White Paper: The OSF Distributed Computing Environment: Building on International Standards.

[BC91] Brad Curtis Johnson: A Distributed Computing Environment Framework, An OSF Perspective, and June 1991.

[OSFInter91] An OSF white paper: Interoperability, a key Criterion for open Systems, June 1991

[LOSF90] Norbert Leser: Towards a Worldwide Distributed File System, September 1990

[Kong90] Michael M Kong: DCE An environment for Secure Client Server Computing

[IAIX] IBM AIX Website

http://www-3.ibm.com/software/network/dce/library/publications/dceaix.html

[IDFS] IBM white paper on File Systems in DFS

http://www.transarc.ibm.com/Library/whitepapers/OSF_Whitepapers/dfs.html

[Idirect] IBM white paper on Directory Services

http://www.transarc.ibm.com/Library/whitepapers/OSF_Whitepapers/directory.html

[IRPC] IBM White Paper on Remote Procedure calls

http://www.transarc.ibm.com/Library/whitepapers/OSF_Whitepapers/rpc.html

[Isec] IBM White paper on Security

http://www.transarc.ibm.com/Library/whitepapers/OSF_Whitepapers/security.html

[Misc] Technical papers from OSF Website

http://www.opengroup.org/dce/

[Reco98] David Diskin and Sherrie Chubin MITRE Corporation; Recommendations for using DCE, DCOM and CORBA middleware

http://diicoe.disa.mil/coe/atd/ddc_readme.htm

[LINC] Linas Website for CORBA on Linux

http://linas.org/linux/corba.html