DCE With HACMP/6000

AIX

By Jim Wade

This article describes how to increase the availability of DCE services by using the built-in replication feature and by running DCE in an HACMP/6000 environment. The article assumes the reader has experience with HACMP/6000 and DCE.[1]

The Distributed Computing Environment (DCE) is a set of services and tools that supports creating, using, and maintaining distributed applications in a heterogeneous computing environment. DCE provides the following services for distributed applications.

Threads: The DCE threads layer provides the basis for all DCE services. Threads allows an application to create, manage, and synchronize multiple concurrent tasks within a single process. With threads, an application server can respond to multiple clients simultaneously; and conversely, a client application can use multiple servers. DCE threads is based on the POSIX™ 1003.4 Pthread specification.

Remote Procedure Call (RPC): The DCE RPC facility provides tools and runtime services that extend the local function-call mechanism across a network. The RPC Interface Definition Language (IDL) compiler generates the stub code necessary for packaging the arguments and handling the calls to the server functions over the network.

Security Services: The DCE Security Services provides secure communication and controlled access to resources in a distributed system. The Security Service has a user registry, a login facility to initialize the user's environment, and Access Control List facilities to control access to resources.

Cell Directory Service (CDS): The DCE CDS provides the central repository for information about resources in a DCE cell. The CDS manages a database of information stored by DCE or RPC-based services.

Distributed Time Service (DTS): The DCE DTS servers provide synchronized time for the computers in a DCE cell. The DTS also has a set of library routines to convert and calculate times from several time formats including Coordinated Universal Time (UTC).

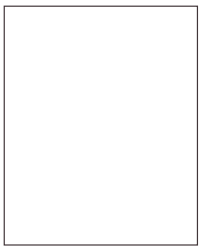
A DCE *cell* is an administrative grouping of machines in a network that share the same Security and CDS servers. The DCE cell relies on the CDS to register services and applications, and on the Security Service to authorize access to the services. These two servers are critical to the DCE cell because they must be up and running for the DCE cell to operate.

Replicated Services in DCE

The CDS and Security Services must be highly available since other services depend on them. CDS achieves this goal by replicating directories and providing ways to keep copies of data consistent. There are two types of replicas in the CDS namespace. The *Master Replica* is the writable replica in which any CDS updates are made, such as registering a server or creating a new entry. The *Read-Only Replica* is a copy that supports only lookup and read operations. All write, create, and update operations must be performed on the Master Replica.

The Security Service provides replicated services by creating slave (read-only) replicas of the Security server. All security updates are made to the master server database, and the master server propagates the changes to the slave servers listed in its replica list. Examples of updates include adding a new principal such as a new server or account for the user to the registry, or refreshing a server key. Examples of read-only access

[1] This article is based on working with DCE Version 1.2 and HACMP/6000 Version 1.2.



Jim Wade

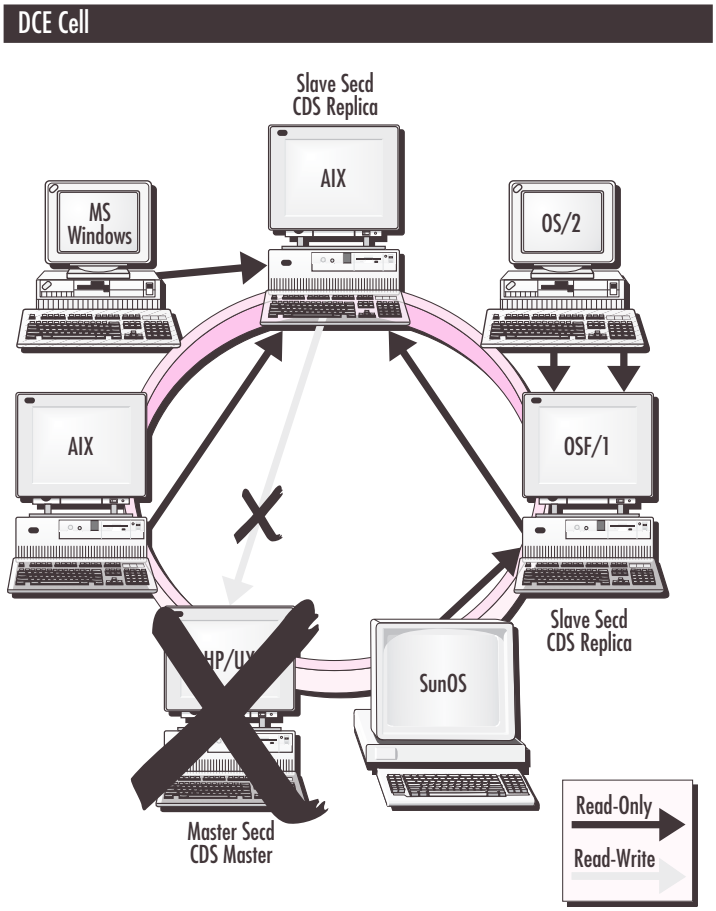


Figure 1. DCE cell with failed master server

include a user logging in (using `dce_login` or other utilities) to get the DCE network credentials.

In a normal running cell with replicated Security and CDS servers and several clients, read requests are directed to any servers on the network, as shown in Figure 1. If either of the master servers—CDS or Security—are down, operations such as logging in or locating a server on the network are not affected.

No additions or changes can be made to the Security User Registry database if the Security server is down. Also, DCE servers such as the CDS server will not be able to manage the server keys used to encrypt the tickets for secure communications.

The CDS server refreshes its key every two hours. This key is used for secure RPC communication between the CDS servers in the cell. Update propagation for replication may not be possible during this time.

The application server will fail to start if the CDS server that contains the master replica for a directory registers itself in CDS. If the complete namespace is not replicated to a clearinghouse in a secondary server, that information will not be available for clients. Clients would not be able to locate services that are already running until the CDS server containing the directory is brought back online.

Recovery Methods

Converting a secondary CDS server to a master CDS server requires the complete namespace to be replicated to the secondary server. The `cdscp` control program makes the conversion relatively simple. An administrator must change the secondary server to be the master and exclude the old master server from the replica list. After the previous master server comes back online, it can be added as a secondary server or converted back to be the master server.

Converting a secondary Security server to a master server is more complicated. The database files from the master Security server directories must be copied to the secondary server machine. The administrator can use the `sec_admin` command to convert the secondary Security server to become the new master server.

Both scenarios require a DCE cell administrator to convert a secondary server to the master server. This can be quite a problem in an operation that runs 7 days a week, 24 hours a day (typically referred to as 7x24). In this situation, a knowledgeable administrator must be on hand or be called in to get a master server available. By using HACMP/6000, you do not need an administrator on call.

High Availability Clustered Multiprocessing

HACMP/6000 is a clustered environment of loosely coupled RISC System/6000s running AIX. It supports high availability through shared resource access. HACMP/6000 controls access to the common resources and enables recovery after failures.

With careful planning, AIX DCE can be configured to run in the HACMP/6000 environment. This allows the DCE CDS or master Security server to recover without intervention from an administrator. HACMP/6000 will not ensure availability if the server process fails or is killed because HACMP/6000 is set up to recover only disks, machines, and network adapters.

Currently, AIX DCE supports HACMP/6000 recovery only in a rotating standby or hot-standby mode. That is because both IP address and hostname takeover are required for the DCE CDS and Security servers to be reliably restarted on a different machine from the one on which they were originally configured.

Implementing DCE with HACMP

HACMP/6000 should be configured before installing DCE. For the DCE servers to be recovered in the HACMP/6000 environment, the HACMP/6000 environment must take over several filesystems used by the DCE servers to store their on-disk databases and information. The filesystems used by DCE include `/krb5`, `/var/dce`, and `/etc/dce`.

Four files that are stored in the `/etc` filesystem must be copied to a filesystem that is shared between the HACMP/6000 machines. These files, `/etc/rc.dce`, `/etc/dce_cf.db`, `/etc/rc.dts`, and `/etc/mkdce.data`, are modified by the DCE configuration tools. They should be copied from the HACMP/6000 scripts into the `/etc/dce` filesystem while HACMP/6000 is stopped, and to the local filesystem before starting DCE.

```
Function start_servers()
.
.
.
#=================Fill In
Here==================
cp /etc/dce/dce_cf.db.save /etc/dce_cf.db
cp /etc/dce/rc.dce.save /etc/rc.dce
cp /etc/dce/rc.dts.save /etc/rc.dts
cp /etc/dce/mkdce.data.save
/etc/mkdce.data
sh /etc/rc.dce
#=======================================
========
.
.
.
Function stop_servers()
.
.
.
#=================Fill In
Here==================
sh /etc/dce.clean
cp /etc/rc.dce /etc/dce/rc.dce.save
cp /etc/dce_cf.db /etc/dce/dce_cf.db.save
cp /etc/rc.dts /etc/dce/rc.dts.save
cp /etc/mkdce.data
```

Figure 2. Code fragment from HACMP/6000 node.servers file

DCE should not be started when the system is rebooted since TCP/IP must be running before the DCE services can start (HACMP/6000 starts TCP/IP and then DCE). HACMP/6000 provides a set of configuration scripts for starting applications. Figure 2 shows two portions of the `node.servers` script that have been modified to save and restore the DCE configuration files, and to start and stop DCE.

Network interfaces should be considered when configuring DCE in an HACMP/6000 environment. For example, consider an HACMP/6000 configuration with two Token-Ring adapters (one active interface and one standby interface) and a serial connection for HACMP/6000 control. The DCE services use only the active adapter. System administrators must mask out the use of the second Token-Ring interface and the serial interface by using the DCE environment variable `RPC_UNSUPPORTED_NETIFS`[1]. This variable can be put into the `/etc/environment` file so that any DCE server programs will have these addresses masked out. The correct usage would be as follows:

```
RPC_UNSUPPORTED_NETIFS=sl0:tr1
```

With careful planning, AIX DCE can be configured to run in the HACMP/6000 environment.

Conclusion

Configuring the DCE CDS and master Security servers to use their built-in replication mechanisms results in continued operation, even if one of the servers goes down. Combining DCE replication services with HACMP/6000 provides the highest availability for DCE services without administrator intervention.

Jim Wade, *IBM Corporation, 11400 Burnet Road, Austin, TX 78758. Mr. Wade has been a member of the DCE development team since the Open Software Foundation® DCE integration project. He has ported the CDS, Time, and Security components of DCE to the AIX operating system. He has held several project lead positions for the AIX DCE licensed program products and worked as a consultant with customers who use DCE and HACMP/6000.*

[1] To enable this environmental variable, a PTF is needed. This PTF is implemented in AIX DCE 1.2 PTF 423300.