DCE/DFS on a Cluster of J90s

Alan K. Powers, Sterling Software Inc., Numerical Aerodynamic Simulation Facility, NASA Ames Research Center, M/S 258-6, Moffett Field, CA 94035-1000, USA

ABSTRACT: *The Numerical Aerodynamic Simulation (NAS) Facility at NASA Ames Research Center has installed four J90s to build a cluster environment. To build a J90 cluster, the filesystems must be in the same name space between the cluster members. NAS is beta testing Distributed Computing Environment and Distributed File System (DCE/DFS) version 1.1, which will provide a common name space within the cluster. This paper presents the trials and tribulations of using DCE/DFS in a cluster environment. Performance measurements were taken on the cluster and their analysis is discussed.*

J90 Mission

Recently, NAS purchased four CRAY J90s. The primary purpose of these machines was to be a testbed platform to help a selective set of CRAY C90 customers port their traditional vector applications to a parallel application using either a message passing interface (MPI) or high performance FORTRAN (HPF). The J90s were chosen because of their cost effective MFLOP rate and they have the same operating system and batch system as the C90 UNICOS and Portable Batch System (PBS). Customers could focus their time on porting their application instead of learning a new operating system and system tools.

J90 Configuration

All the J90s have 128 megawords (MW) of memory and three J90s have 4 CPUs, the other has 8 CPUs and is used as the front-end. The front-end has 72 gigabytes (GB) of SCSI disks (6SDD) and two IOS with 4 SCSI controllers, whereas the other J90s have 36 GB of SCSI disks and one IOS with 2 SCSI controllers. Each J90 has a FDDI and HIPPI interface; the front-end has an additional HIPPI interface.

The front-end is the only machine with all the compilers and is used for interactive work. The others are used as batch machines. The home filesystem (18 GB) is local to the front-end and is network file system version 3 (NFS v3) mounted to the other J90s. All NFS v3 traffic between the J90s is over HIPPI, configured with about 64 kilobytes (KB) maximum transfer unit (MTU) and about 256 KB transmission control protocol (TCP) to send and receive buffers. The front-end schedules all batch work over the J90s. Customers are strongly encouraged to use all CPUs on each J90 simultaneously.

Each J90 has a *big* filesystem (18 GB) configured with temporary directories for batch job sessions. Each of these filesystems are NFS v3 mounted to each other. These filesystems are created using four equal partitions from four different disks and are distributed evenly between two controllers. Using 1 megabyte (MB) blocks, the transfer rate to the local *big* filesystem is 16-18 megabyte per second (MB/s) and using NFS v3 to a remote *big* filesystem, the transfer rate is 1.5-3 MB/s.

There is a *cache* filesystem (6 GB) on each of the J90s with the transfer rate of 16-18 GB/s. This is used only for distributed file system (DFS) file caching.

DCE/DFS Software

The DCE/DFS was chosen to provide a common name space between the cluster of J90s due to its number of features (file caching, security, greater than 2 GB file size, large block network transfers, etc.) in comparison with NFS v2 features and the limited implementations of NFS v3.

There are four major parts to the DCE/DFS: security, Cell Directory Service (CDS), Distributed Time Service (DTS), and Distributed File System (DFS). Each part has clients and corresponding daemons. CRI's version also includes clients and daemons for kerberos V5 commands- telnet, klogin, krcp, and krsh. For sites within the United States some of these commands can send and receive encrypted data. CRI's version does not include the time (dtsd) server, the security (secd) server, and the cell directory server, If DFS is used, then the file location server (FLS) must depend on another platform for this service. Before the CRI's DCE/DFS software is configured, there must be DCE services already working on a platform in the network. The J90s used IBM workstations as the master and backup DCE services.

The time daemon must be configured on another workstation or the Network Time Protocol (NTP) must be used. Our site chose to used NTP to synchronize the time between all the hosts in the local network.

There are several new features in this release, but these features were not tested because of problems with the core components or a feature not needed by our site, like multilevel security.

DCE/DFS Learning Curve

Several years ago I worked with AFS and the Apollo Domain operating system, which Hewlett Packard purchased, and parts of this were used for DCE/DFS. Before starting this project about a year ago, I was aware of the different features of DCE/DFS, but did not have any experience using DCE/DFS. Working part-time on this project first on the C90s and now on the J90s, I considered myself to have an entry level knowledge of DCE/DFS.

There are very few books and training classes on the subject and vendors have very few experts for technical support. Before any site commits to DCE/DFS, several staff members should take two weeks of DCE/DFS training. Learning DCE/DFS is like learning another operating system. Most of the commands are non-UNIX like and have many *verbs* and *objects* (options). There is a lot of detective work to discern where the error messages are coming from, or understanding the vague or generic error messages, or determining why the DCE/DFS hangs when there are no error messages.

DCE/DFS Resources

The binary release of DCE/DFS consumes about 170 MB of disk space in the source filesystem and up to 200 MB once installed in the /opt filesystem. The DCE/DFS daemons' total main memory use is 3.5 to 4 MW. Even though the kernel memory buffer (*mbufs)* parameter has been increased to 10,000 (1.25 MW) at times the system still consumes all of the memory buffers. The J90 DCE/DFS enabled Unicos 9.0.2 kernel uses 4.6 MW of main memory.

DFS Configuration

The DFS cache sizes are 64 MB, 1000 MB, 2000 MB, and 4000 MB within a 6000 MB filesystem (16-18 MB/s) for the different J90s. Each cache file is 256 KB. The number of cache files depend directly on the size of the cache. The default number of files equal the cache size divided by 64 KB. Thus for 64 MB cache, it would create 1000 cache files. CRI has modified (for the better) how cache files are created by creating sub-directories and having a maximum of 256 cache files in each sub-directory. DFS was configured to use the HIPPI interfaces for the file transfers.

DFS Performance

Throughout the testing, each of the J90s had less than 50% CPU utilization. The previous *big* filesystems (16 to 18 MB/s) were used to export to the DFS global name space. The HIPPI interfaces could transfer data between 60 to 65 MB/s. The file sizes used for testing were 1, 10, 100, and 500 MB. Using ***ftp*** to transfer binary files between the two J90s' *big* filesystems, the transfer rate was 13 to 15 MB/s; using ***rcp,*** the transfer rate was 2 MB/s. When transferring data between an NFS v3 filesystem and the *big* filesystem, the transfer rate was 1.5 to 3 MB/s. The DFS cold cache transfer was 0.5 to 3 MB/s. DFS hot cache was 3 to 6 MB/s and between DFS filesystems the transfer rate was 0.5 to 1 MB/s.

Table 1. Transfer rates for different applications using the *big* filesystem.

	Transfer Rate (MB/s
Hippi (Bandwidth)	60-65
Local Filesystem	16-18
ftp	13-15
rcp	2
NFS v3	1.5-3
DFS (cold cache)	0.5-3
DFS(hot cache)	3-6
DFS-DFS	0.5-1.0

Overall, when transferring files using DFS, the increased CPU utilization on the systems was minimal. For small files DFS did very well, but for large files the file cache did not help increase the transfer rate. On the C90s there are over 1.4 million files and 90 % of the total number of files are less than one MB and consuming 2 % of the total data. Files greater than one MB are 10 % of the total number of files, but are 98 % of the total data. There is up to 120 GB a day of data transferring between the C90 and other systems. It seems the DFS file cache would have to be extremely large to be effective.

Outstanding Problems

- Most of the time it takes a reboot to restart DCE/DFS, when there are DCE/DFS problems.
- Kerberos ***telne***t does not support the *-k* (realm) option as stated in the man page and usage statement.
- Kerberos ***rcp*** fails when using the *-k* option (realm).
- There is a lack of documentation to properly configure kerberos version 5 to enable it to work with the DCE security server.
- Source is needed to debug problems.

CRI Advantages

- CRI's documentation for configuring DCE/DFS (***dce_config***) was informative and easy to follow.
- CRI developed a couple of scripts (***dfsmkfs, dfsrmfs***) to easily mount and unmount the local filesystems to DFS namespace.
- CRI is prompt with DCE/DFS bug fixes for the kernel.

Suggestions For CRI

- Create a site tunable parameter for file size to bypass DFS file cache.
- Develop a better DFS file caching algorithm. Use only one cache file for the large files (site tunable parameter).
- Develop documentation explaining how to tune DFS and determine the best file cache size to use.
- Use CRInform or Web pages to provide the latest documentation online. An early product release is bound to have problems and it is difficult to keep the hard copy documentation up to date.
- Create an evolving FAQ Web page to cover CRI implementation of DCE/DFS. Specialized knowledge can become common knowledge once documented well and made easily accessible.
- Develop a utility to test the major pieces of DCE/DFS and kerberos to identify problems. For example ***ping*** and ***traceroute*** are networking tools to debug some network problems. Instead of taking a long time to develop experts to know DCE/DFS, develop expert tools.
- Change error messages to be more meaningful.

Other Possible Solutions

New releases of NFS v3 are now available from several vendors. With NFS v3 and friends, there have been a lot of feature improvements to be able to compete with DCE/DFS without all the extra complexity or system and staff requirements.

Secure shell (www.cs.hut.fi/ssh) could functionally provide the similar security of the kerberos commands ***klogin, krsh, krcp,*** and ***telnet***. The secure shell was developed by Tatu Ylnen at Helsinki University of Technology, Finland and there is even a Unicos version for the C90.

Several vendors are developing new or enhancing their operating systems to provide a single system image. It seems these new operating systems will soon replace the functionality DCE/DFS offers.

Summary

The features of DCE/DFS are very desireable but the current implementation is strongly lacking. CRI is not the only vendor having problems providing a good DCE/DFS implementation. DCE/DFS consumes lots of resources, has poor error messages, and takes a long time to learn well.

DFS transfer rates compared to NFS v3 are adequate, but should be closer to the ftp transfer rate.

CRI has several opportunities to enhance DCE/DFS, by providing better online documentation, better error messages, and a test utility to help identify problems.

Acknowledgments

This work was performed by Sterling Software at the Numerical Aerodynamic Simulation Facility (Moffett Field, CA 94035-1000) under NASA Contract NAS2-13619.

The author can be reached at powers@nas.nasa.gov and online CUG papers can be accessed on the Web at www.nas.nasa.gov/~powers.

Additional Reading

[1] Cray DCE Client Services/ DCE/DFS Server Release Overview (RO-5225 1.1) www.cray.com
[2] Guide to OSF/1: A Technical Synopsis, O'Reilly & Associates, Inc., 1991, www.ora.com
[3] Open Software Foundation—*DCE Administration Guide—Core Components* R1.1, Prentice Hall PTR, ISBN 0-13-1858440-0, www.prenhall.com
[4] Open Software Foundation -- Introduction to OSF DCE, Prentice Hall ISBN 0-13-490624-1, www.prenhall.com
[5] Open Software Foundation—*DCE Application Development Guide*, Prentice Hall, ISBN 0-13-643826-1, www.prenhall.com
[6] Introductory DCE/DFS Papers , www.citi.umich.edu/u/cja/DCE/
[7] The NFS Distributed File Service, March 1995, www.sun.com/sunsoft/solaris/desktop/nfs.html
[8] WebNFS, The Filesystem for the World Wide Web, Brent Callaghan, May 3, 1996, www.sun.com/solaris/networking/webnfs/webnfs.html