

October 2000
Prepared by:
UNIX Software Division
Compaq Computer Corporation

Contents

**Advanced Server for UNIX
(ASU)3**
Cluster Objectives3
**Architecture and
Requirements4**
Implementation.....5
Summary6

Case study - Advanced Server for UNIX (ASU) - Becoming a Multi-instance Application in a TruCluster Server Environment

Abstract: Advanced Server for UNIX (ASU) is a multiprocess application that provides NT-style file, print, and security services. It serves a large number of networked Windows-based clients. This paper describes the advantages gained by becoming a multi-instance TruCluster Server application and the changes to ASU that were implemented to enable multi-instance operation. Further improvements are also noted which are needed to support 24x7 availability with no scheduled down-time.

Notice

Case Study - Advanced Server for UNIX (ASU) - Becoming a Multi-instance Application in a TruCluster Server Environment

White Paper prepared by UNIX Software Division

First Edition (October 2000)

©2000 Compaq Computer Corporation. Printed in the U.S.A.

Compaq and the Compaq logo are Registered in the U.S. Patent and Trademark Office.

AlphaServer and Tru64 are trademarks of Compaq Information Technologies Group, L.P.

Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation.

UNIX is a trademark of The Open Group.

Compaq Computer Corporation shall not be liable for technical or editorial errors or omissions contained herein. The information in this document is subject to change without notice.

Advanced Server for UNIX (ASU)

The ASU software is a Tru64 UNIX layered application that implements Windows NT Version 4.0 server services and functionality on a system running the Tru64 UNIX Version 4.0F and Tru64 UNIX V5 or higher or higher operating system software. The Tru64 UNIX system on which the ASU software is running appears as a Windows NT Version 4.0 server to other Windows systems and to users of Windows systems.

ASU is most commonly used to provide Windows users with scalable and high-performance storage as well as centralized printing services. The environments in which these users work is becoming increasingly more complex and, in many cases, includes support for mission-critical data. In addition, with the increasing number of applications designed for Internet access, ASU systems are being used for third-tier application storage where the ASU "clients" are really applications on other Windows-based servers which, in turn, provide data to web-based users.

These trends motivated the effort to make ASU highly available, more reliable, and scalable. At the same time, the effort to manage the product could not increase. Making use of the cluster capabilities of Tru64 UNIX was the best way to accomplish these goals.

Cluster Objectives

In order to provide a more scalable system, the first decision was to make ASU a multi-instance TruCluster application. Windows clients do not see any difference with this change - ASU still appears as one large, high-performance server. Being a multi-instance application provides several advantages:

Scalability - Increasing the number of nodes in a cluster provides capacity and availability beyond that possible in a single system. The processing power, the storage systems, and the network connections of the cluster can be utilized by ASU. This allows a larger number of users to connect, yet, with the TruCluster Server V5 and its cluster file system, all users make use of a single set of data. This avoids data duplication and update synchronization.

Availability - The services of ASU can be maintained even if nodes are stopped or if they fail. The connections on the remaining nodes remain active. Connections on the failing node are automatically reestablished on the remaining nodes with a minimum of disruption to the Windows clients.

In addition, converting to a multi-instance application allows other improvements to be made to ASU:

- Ease of management - Management commands can be made to apply cluster-wide.
- Ease of installation - Installation is required only once because changes to the cluster become configuration changes.
- Dynamic configuration - Nodes can be taken on- or off-line as needed to provide maintenance or reconfiguration. Nodes can be added or shut down without interrupting ASU operation.

- Load distribution - The TruCluster Server software automatically distributes the client load among the nodes of the cluster. This allows balancing the load among nodes on an equal or unequal basis.
- 24x7 service - By making additional changes to the software to allow node-by-node software upgrades (rolling upgrades), continuous nonstop service can be provided, even during software upgrades. (This feature is currently being considered).

Architecture and Requirements

An ASU server consists of several cooperating processes. The controller process starts and stops all other processes, accepts client connections, passes them to server processes, and facilitates communication among other processes. The server processes perform the real work of the product, handling client requests for file and printer services. In addition, several other specialized processes handle various functions such as NT account replication and browsing.

The server processes communicate their state and share file information using mechanisms which include messaging, shared memory, and common files. For example:

- Information about which clients are connected, and about files that are open is kept in shared memory.
- Pipes are used to communicate between the controller and each server process. Each process waits in a poll call until a message is received and then processes the message.
- Information about users, file access controls, logged events, and server parameters are stored in database files.

When considering how to operate in a cluster, the first decision was to make ASU a multi-instance application. This approach provided the flexibility needed to operate in configurations from single-node systems to large, multi-node clusters. Part of this decision included the ability to run the complete product on each node. This ensured maximum flexibility and reliability in failure scenarios and required few changes to the original product.

The next level of decisions involved determining how to coordinate the cluster nodes and the processes in each node. It is critical that each node be aware of shared data and be coordinated with the other nodes.

When converting ASU to operate in a cluster environment, communication mechanisms needed to be changed to operate in a cluster. The TruCluster Server File System supports common files (including memory-mapped files). However, messaging between cluster members had to be provided in the application. Cluster members had to communicate in order to prevent access conflicts, to keep common databases of users and user information, and to allow cluster-wide management.

A final goal was to maintain the management and administration of ASU as a single entity, even with multiple simultaneous instances of the software. In fact, since most ASU management is done using a PC running the Windows NT User Manager, single entity management is mandated since the User Manager has no concept of a clustered server. On the other hand, where the administration is performed from the UNIX platform, there are circumstances where per-node management is desirable, for example, to show network counters.

Implementation

Several fundamental design changes were necessary to allow simultaneous multinode operation:

- Named-pipes had to be made unique. This was done by appending the node name to the pipe name.
- File locks, which are used to determine system states, were changed to use the TruCluster Distributed Lock Manager (DLM). They also used the node name as part of the lock name to make them unique.
- Some databases were locked against multiple access by using a shared memory semaphore. Since semaphores do not operate among nodes, file locks were used.
- Reporting structures and log files were changed to add the node name for identification.
- While the majority of parameters are applied cluster-wide, there are certain configuration parameters which vary on a per-node basis. Initialization files and processes were changed to allow node-specific parameters.
- Management mechanisms were modified to apply to a cluster environment.

An additional process was added to coordinate communication among cluster nodes. This "cluster process" centralizes the internodal communication and coordinates the exchange of status data among the nodes. Two types of mechanisms are used for coordination and communication among the nodes:

- DLM locks and lock value blocks are used to provide synchronization and limited data.
- Messaging based upon TCP/IP is used to exchange more extensive file-related data.

Certain functions in ASU provide network interoperability. In order to appear as a single large server, these functions cannot be duplicated on multiple nodes. The cluster processes on the nodes elect a "master" node by queuing for an exclusive DLM lock. The node receiving the lock is the master node and serves as the coordinator of all internodal communication, keeps a centralized database of cluster state information, and performs the operations that can be performed by only one node. Copies of each node's data are stored locally in case the master node stops. If that happens, another node receives the lock, becoming the new master. Then the surviving nodes populate a new database. This design, rather than utilizing a polling method, allows individual nodes to access local data for quick response-time, and limits off-node data access to the master node, limiting latency. Because the database contains users' file-locking information, good response time is very important.

Messages flowing among the nodes were designed to maintain the same format as those used within the nodes. This minimizes the effort and time spent reformatting data. This is especially useful for administrative messages which are transferred from the requesting node to a responding node.

Another major consideration in a multi-instance design is how to control load balancing. The TruCluster Server architecture provides a "cluster alias" which is a common address that clients can use to connect to the cluster as a whole. Connections received over the alias are routed to different nodes based upon the configuration of each node. Of course, if a node is stopped, new connections are routed only to the active nodes. Because ASU receives its client connections by listening on a well-known port, adding the ability to listen for that port on the cluster alias utilizes the connection load balancing inherent in the TruCluster architecture.

Implementing single-entity application management was done on a case-by-case basis as determined by the intent of the management function. Some administrative commands require that data from all the nodes be accumulated (for example, to report all connections), while other commands need to be forwarded to a specific node (for example, to shut down a specific node).

Supporting rolling upgrading (upgrading software and databases dynamically while still providing service) is a particular challenge. In addition to providing its own rolling upgrade, a product should continue to operate while the operating system itself performs such an upgrade. Currently ASU will operate throughout a rolling upgrade of the operating system, but does not provide a rolling upgrade itself.

Summary

Making a complex product like ASU operate as a multi-instance cluster application provides many benefits: increased performance, reliability, scalability, and fail-safe operation. The degree of effort required to make a transition from a single-instance to a multi-instance application depends upon the architecture and requirements of the particular application. However, TruCluster Server software already provides many mechanisms which assist this process. The examples in this paper show that some changes are necessary to correct internal designs which did not consider a multinode architecture, while other changes are required to allow simultaneous multi-instance operation and to take advantage of the increased functionality provided by TruCluster Server technology.