Oracle® Universal Content Management
Clusters Concept Guide
10*g* Release 3 (10.1.3.3.1)

May 2007

Clusters Concept Guide, 10*g* Release 3 (10.1.3.3.1)

Copyright © 2007, Oracle. All rights reserved.

Contributing Authors: Deanna Burke, Jean Wilson

Contributors: Alec Kloss, Sam White, Bob Wicklund

Table of Contents

C

Chapter

1

INTRODUCTION

OVERVIEW

This chapter covers the following topics:

PRODUCT OVERVIEW

A server cluster is a group of independent servers managed as a single system that can be used as a multipurpose platform for database management, file and intranet data sharing, and general business applications.

Content Server can be a stand-alone system, or it can be modified and integrated in unlimited configurations to alleviate problems of maintenance, load balancing, and scalability. In a cluster configuration where multiple servers share a Content Server instance, all of the servers also use a common content server file system, database, and search collection. A load balancer can be used to provide high availability and scalability for consumption.

Important: The clustering concepts and implementation information included in this document are intended only as a reference and not as complete examples of how you should establish your cluster configuration.

Important: Not all clustering configurations have been tested, and this guide does not specifically cover all possible clustering environments. We recommend that you test your clustering configuration in a development environment before deploying it to production.

About This Guide

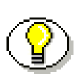
This Clustering Concepts Guide provides general overview, background and conceptual information about the clustered Content Server product. It discusses recommended hardware and software for various cluster server platforms where the machines are running the Windows 2000 Server, or the UNIX operating system (Sun Solaris, HP-UX, IBM AIX, or Linux Red Hat). The information contained in this document is subject to change as the product technology evolves and as hardware, operating systems, and third-party software are created and modified.

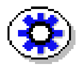
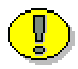
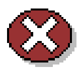
Intended Audience

This configuration guide is intended for individuals who are responsible for designing, implementing, and/or maintaining systems that combine the Content Server product with multiple servers in a clustered environment. In addition, this manual will be useful for database administrators who maintain the integrity of the document information stored in the Content Servers. This document assumes that you are familiar with Content Server products and the architecture of the Content Server.

Symbols and Conventions

The following symbols and conventions are used throughout this manual:

Symbols	Description
	This is a note. It is used to bring special attention to information.

Symbols	Description
	This is a tech tip. It is used to identify information that can be used to make your tasks easier.
	This is an important notice. It is used to identify a required step or required information.
	This is a caution. It is used to identify information that might cause loss of data or serious system problems.

Chapter Overviews

The following chapters are included in this guide:

- Chapter 2 (*Implementation Considerations)*: This chapter provides some background information about clusters in general and the standard clustered Content Server product in particular. It is intended to provide background information to help users assess their practical clustering technology needs and to evaluate various software and hardware combinations for cluster platforms.
- Chapter 3 (*Cluster Installations)*: This chapter provides sample cluster and installation architectures as they pertain to the clustered Content Server. These examples are intended to illustrate possible cluster configurations using the Content Server product. These are to be used as guideline and testing scenarios only and are not necessarily meant to be followed for customer implementations of clustered Content Servers.

Chapter

2

IMPLEMENTATION CONSIDERATIONS

OVERVIEW

There are several things to consider when planning a cluster implementation. This chapter covers the following topics which discuss those things which may affect your implementation planning:

ABOUT CLUSTERS

A Content Server cluster uses existing cluster infrastructure to provide enterprise-level software that seamlessly integrates into most commercially available cluster server platforms. A standard clustered Content Server is set up to simultaneously share file systems and databases. A shared database is one that is always highly available. A shared file system is one in which all nodes read and write to the same file system. All nodes also mount or map the shared file system identically.

A clustered Content Server consists of the installed Content Server product and multiple servers in a cluster environment. The clustered servers share a single content server dataset that includes directories for the database, search index, vault (native documents) and web layout (web-viewable versions). The vault and web layout directories are collectively known as the file system.

Conceptually, the definition of a clustered Content Server is quite different from most of the typical computer industry definitions. The standard definition of a server cluster is a parallel or distributed system that consists of a collection of independent servers or computer systems that work together and are managed as a single, unified computing resource.

Content Server is not a cluster-aware application. It depends on the established infrastructure to provide the cluster functionality. The clustered Content Server implementation emphasizes the software structure involved in cluster configurations. Contrastingly, most sources define server clusters in terms of the hardware that constitutes a physical cluster configuration.

Cluster systems are used for many reasons, but the greatest demand for cluster technology is to improve the availability and scalability of mission-critical data, applications and resources. Using clusters ensures that these key corporate components always remain available, even during maintenance upgrades, system failures, or backups.

Different studies confirm that the most common uses of clustering involve database management, file/intranet data sharing, messaging, and general business applications. Databases and electronic mail have become essential to daily operations and large corporations as well as small businesses are seeking higher availability for their mission-critical, server-based data and applications.

In general, the cluster is designed to prevent a single point of failure. Thus, server clusters preserve client access to data, applications and resources during both failures and planned outages. Server clusters make it possible to share a computing load over several systems without either the users or system administrators needing to know that more than one system is involved.

Content Server and File System Interaction

Content Server and its associated products are supported in conjunction with file systems that function reliably and operate according to the following common principles:

- *accurate time stamps*: the time stamps used by the file system are always up-to-date and all processes have an identical view of those time stamps. If one process makes a change to a file, its time stamp is updated immediately and all other processes making reference to that file see the updated time stamp.
- *file locking*: if one process locks a file, no other processes can perform operations which are a violation of that lock.
- *clean reads*: if a file has been updated by one process, all read actions by other processes will see and incorporate the data in the updated file. No ‘dirty reads’ of file caches are allowed.
- *transactional renames*: if one process renames a file, other processes use the new name immediately and no longer find the file at the previous location. This also implies that if two separate processes try to rename the same file, only one will succeed.

  The Content Server can handle a millisecond of latency. For example, Microsoft does not have transactional renames. There is a latency of a few milliseconds before some renames take effect and it is possible for two processes to rename the same file to different names. A test for the existence at both of the new locations will succeed for approximately a millisecond.
- *atomic file creation*: if two processes want to create the same file, only one file will succeed.

Note that the performance of the file system directly affects the performance of the Content Server. The Content Server does not perform well if the file system does not perform well.

Recommendations

Before implementing a clustered system, the following are strongly recommended:

- Advanced knowledge of and experience with Content Server.
- Installation of Content Server with a cluster environment in mind. More planning and effort is required to implement a cluster configuration with a Content Server system that is already deployed.

- Installation of a ‘mirrored’ environment in a development setting. As far as possible, this environment should mimic what will be used for the clustered environment (matching file system types, software, and infrastructure). While an exact duplicate of the production system isn’t necessary, the fundamental environment (for example, using SAN with Veritas) should be in place.
- Knowledge and evaluation of the current corporate infrastructure. Installing a clustered Content Server may require a re-evaluation of organization policies or procedures or a re-evaluation of current roles and duties of key personnel.

Best Practices

The following list describes some recommended best practices for your environment before setting up a cluster.

- Always use an alias for your servers, users, groups, and so on. This makes it easier to shift function from one machine to another in the event of a system outage.
- Establish a meaningful naming convention for machines and for users, accounts, and groups. For example, use EC for East Coast, or MSP for Minneapolis/St. Paul to indicate geographic distribution. This makes it easier to associate function with machines, groups, and regions. In addition, meaningful names will then appear in log files and reports, making it simpler to track problems and troubleshoot solutions.
- Make sure that instance names are unique. A good convention to follow is to set up a standard naming convention for your entire organization, as in this example:

  ```
  <environment>_<type>_<department>
  ```

  Using this convention, an instance named LA_PROD_RD indicates a production R&D instance in Los Angeles.
- Set up one Admin Server (as described in the Configuration Guide for your system) per physical cluster. Do not try to administer multiple physical clusters from one Admin Server. You can administer multiple nodes in one physical environment from one Admin Server.
- A Content Server cluster is different than other industry clusters. It is a single installation with configurations placed on multiple machines. It is not composed of multiple installations on multiple machines.
- Shared storage is necessary which is accessible in read and write mode by all systems in the cluster.
- A shared database schema needs to be in place. All systems in the cluster should have owner rights to the database.

- If load balancing is needed, a load balancing device should be in place. Content Server does not provide load balancing.
- It is advisable to use a specific node for archiving and to do all archiving on that node. Each node tracks the success and failure of archiving on that node. Archiving on an unspecified node is not supported and may produce unexpected results.

BENEFITS OF CLUSTER ENVIRONMENTS

Generally, only a small percentage of companies need clustering. It is used primarily for mission-critical database management, file/intranet data sharing, geographic load-balancing of data and systems, and general business applications.

The primary benefits of using server clustering include:

- **Availability**

  Clusters provide system robustness that reduces or eliminates planned and unplanned downtime and ensures high performance with backup and failover functionality. If one component (hardware or software) in the system fails, another component in the cluster replaces it with minimal or no noticeable difference to users. In fact, users may see degraded performance but will not lose access to the service. Also, the workload in a cluster is automatically rebalanced when a failed server comes back online.

- **Maintenance**

  Clustering simplifies administrative maintenance because multiple servers are managed as a single system and all of the servers share a single dataset that includes related files and shared database access. Not only does this minimize required maintenance efforts, it also ensures that servers can be added or removed from the cluster without requiring reconfiguration and downtime.

  Depending on the specific operating system or supplemental applications, clustering supports “rolling upgrades” that minimize disruptions in operations for planned downtime caused by maintenance or upgrades. During off-peak time, the workload can be shifted onto other servers within the cluster while the maintenance or upgrade is performed on the unloaded server.

- **Scalability**

  The size or configuration of the cluster can easily be expanded or upgraded to more appropriately accommodate changing conditions and requirements of the organization. New components can easily be added as system load increases. Also, clustered content server configurations support limited horizontal scaling (multiple machines sharing a single dataset) to optimize overall throughput.

- **Load Balancing**

  Consumption load can be distributed evenly among multiple instances of web servers within the cluster to ensure that no single device is overwhelmed. If one server starts to get overloaded, requests are forwarded to another server with available capacity. Load balancing schemes are especially important for cluster installations where it is difficult to predict the number of requests that will be issued to a server.

- **Distributed Environment**

  The content server can be configured to run in a decentralized, distributed environment as well as scale and search across multiple machines. The modular design allows organizations to split the main content server components across multiple servers to maximize system resources throughout the installation. For example, to relocate the content server processing, the underlying database and file system can reside on a server separate from other components.

ROBUSTNESS AND RECOVERY FEATURES

High availability is one of the primary concerns and expectations of businesses. If a server, resource, or application becomes unavailable due to failure or maintenance activities, the cluster must be able to ensure that mission-critical applications and data access remain available.

For this reason, the clustered Content Server can preserve maximum system availability through features that include shared-disk accessibility, active/active configuration, and failover and failback functionalities.

Shared File System Accessibility

A clustered Content Server uses shared file system technology. This means that multiple servers in the cluster share concurrent read/write access to a single file system and database.

To preserve system coherence and ensure that the contents and operations of the servers remain synchronized, a shared (or distributed) lock manager is used to mediate ownership of the files, govern disk writes, and keep each server's cache data current. Although the servers in the shared file system cluster rely on a common data transfer device, they do not require shared memory.

Because shared file system clusters allow each cluster server a view of a collectively available database, implementing applications on the shared file system cluster is similar to implementing applications on a single server.

Another advantage is that shared file system clusters support higher levels of system availability. If one server fails, the cluster continues to function because all the cluster nodes always have access to the shared database and, therefore, access to the failed server's data. This allows the remaining servers to continue handling their own workload plus the workload from the failed node.

Active/Active Configuration

Unlike many other clustered environments, a clustered Content Server can have every server in the cluster simultaneously active. This is called an active/active cluster configuration. This means that the shared cluster resources, applications and data are simultaneously available to all servers in the cluster. Consequently, active/active clusters support load balancing and high availability. In the event of a failure of any server, the shared resources on that node failover to one of the remaining node servers and users can continue to access these resources without interruption.

Server clusters using active/active configuration ensures that every node in the cluster maintains its own workload. This helps achieve greater scalability in addition to maximizing total system performance. Thus, every server in the cluster is available to do real work, and every server in the cluster is also available to recover the resources, data and processing of any other node in the cluster. There is no need to have a wasted, idle server waiting for a failure. However, active/active cluster servers must be sized appropriately to handle the load of two servers (in the event of a failover).

Failover and Failback Functionalities

Clustered Content Servers support third-party applications that provide failover capabilities. Consequently, if a server or application fails, failover automatically occurs. Consequently, ownership of resources, such as disk drives and Internet Protocol (IP) addresses, is automatically transferred from the failed server to a surviving server within the cluster, greatly minimizing the downtime. The part of the workload that is capable of restarting on the surviving server is restarted and when the failed server comes back online, the cluster automatically rebalances the workload through a process called failback.

Failback is the capability to automatically rebalance the workload in a cluster when a failed server comes back online and, generally, requires no manual intervention. A failback usually occurs when the problem that caused the workload to be moved is resolved and the original server is back up and running. After the failed server reboots, another server in the cluster updates the original (failed) server with current cluster status information and the previously failed-over workload is moved back to the original server. The failover and failback features are important fault tolerance functions.

Failover and fallback are functions of the infrastructure and are not native to Content Server.

IDENTIFYING THE CLUSTER CUSTOMER

Obviously, not every customer needs to use a clustered Content Server for their business applications. Generally, cluster-enabled scalability will only be needed by extremely large enterprise applications.

For example, a cluster solution would be suitable for applications that are too large to run on a single high-end symmetric multiprocessing (SMP) server and cannot be partitioned to run on a distributed network. Conversely, marketing studies have shown that small businesses are increasingly interested in clusters for higher system availability because databases and electronic mail have become critical factors in their daily operations.

The following factors can influence the size/configuration of a cluster, as well as the initial decision to implement a cluster:

- The type or value of content that is contributed and consumed (for example, corporate procedure documents that might be infrequently referenced and viewed vs. a series of HTML forms that might require thousands of searches, updates and checkins daily)
- The amount of expected activity involving the number of page requests or searches against the system's content (including internal employees and external guest users and whether the usage involves read and/or write access)
- The potential number of cached and non-cached search requests submitted (depending on the number of documents being contributed and the likelihood of two users submitting identical search requests)
- The number of document checkouts, updates, and checkins per day
- The number and type of content items in the system and whether or not the documents are separated into different databases (if the total number of documents exceeds the database and/or search index maximum)
- The average search time and amount of system resources required to perform each search request (directly affected by the physical disk size of each document in the

database(s), the uniqueness of the words contained in each document, the number of metadata fields, and the uniqueness of the values in the metadata fields)

- The number of page requests per day (if the content server is responsible for dynamic page construction of a web site)
- The type of system platform used for the installation including the complete hardware equipment configuration and selected software options

COMPONENTS OF A STANDARD CLUSTERED CONTENT SERVER

The clustered Content Server components have the functionality to participate in a clustered environment. However, third-party software may be needed in some environments to enable the clustered Content Server components to function properly.

In general, creating a standard clustered Content Server involves the following minimum components/requirements:

- An installed Content Server product and one shared dataset (file system and database)
- Two or more servers connected by a network (public or private)
- A method for each server to access the other's common file system
- Load balancing to distribute load among multiple servers

For further details about the operating systems, web servers, and databases that are supported, see the *Content Server Cluster Configuration Guide for Microsoft Windows* or the *Content Server Cluster Configuration Guide for UNIX*.

Compatibility and Requirements

To ensure successful implementation of a standard Content Server cluster, the equipment and solutions at the customer site should meet established compatibility.

In a clustered Content Server configuration, the clock in each machine must be synchronized because several pieces of functionality require coordinating the time settings. For example, the work file queues are shared by all the nodes in the cluster. To ensure undisrupted performance, the clock synchronization must be configured to have a maximum disparity of five seconds.

The following table identifies current Universal Content Management product compatibilities with Content Server cluster configurations.

Product	**Version**	**Cluster capable**	**Proxied Support**
Content Server	8.0	X	X
Document Management Products			
Content Categorizer	7.5	X	X
Content Tracker	7.5	X	X
Dynamic Converter	7.2.1	X	X
Dynamic Watermark	7.0		X
Inbound Refinery	8.0	X	X
InterCAD	7.5	X	X
PDF Converter	7.5	X	X
PDF Merge	7.0		X
PDF Watermark	7.5	X	X
XML Converter	7.5	X	X
Report Parser	7.5		X
Web Based Training	7.5	X	
Digital Asset Management Products			
Image Manager	7.5	X	X
Video Manager	7.5	X	X
Compression	7.5	X	X
Web Content Management			
Content Publisher	7.2	X	X
Connection Server	6.5	X	X

Product	Version	Cluster capable	Proxied Support
Site Studio (Server)	7.5	X	X
Site Studio (Contributor)	7.5	X	X
Site Studio (Designer)	7.5	X	X
Publishing Utility	7.2.3	X	X
Records Management			
Records Manager	8.0	X	X
Collaboration Management			
Collaboration Manager	7.5	X	X
Sarbanes Oxley Solution			
Sarbanes Oxley	7.7		
Imaging			
Legacy Integration	7.0	X	X
Tiff Converter	7.5	X	X
Integrations			
Xmetal Integration	7.0	X	X
Desktop	7.5	X	X
Lotus Notes Integrator	7.5	X	X
Content Integration Suite	7.5	X	X
Content Portal Suite	7.5	X	X
OS Special Cases			
IBM z/Series		X	X
Solaris Intel		X	X

Important: Due to potentially high volumes of IP data exchange, it is critical that network access be at an exceptionally efficient level in a cluster environment. In particular, if you are clustering to a Network Attached Storage (NAS) system, an isolated one gigabit subnet (network bus) is recommend between servers in the cluster and the NAS.

Hardware Comparisons

The needs for each customer's cluster implementation are unique and there are many different hardware components to combine and design configuration solutions. This section provides overview information about types of competing peripheral devices such as network storage devices and web server load distribution technologies.

Network Storage Technologies

Enterprise storage consolidation is increasingly important because it improves efficiency, decreases redundancy, and simplifies management.

The two most significant storage technologies currently used are Network Attached Storage (NAS) and Storage Area Network (SAN). Many of the previously existing differences between the two primary network storage products (traditional NAS and SAN storage solutions) are no longer relevant.

Instead, the main differentiation between NAS and SAN products is the choice of network protocol. Although both NAS appliances and SANs provide enterprise storage and offer storage consolidation, they also present different services, advantages, and limitations that need to be considered when designing the system architecture.

Summary of NAS / SAN Features / Capabilities

The following information summarizes the similarities and differences in NAS/SAN capabilities. See Comparison Summary: NAS/SAN (page 2-14) for a comparison table of features.

Network Attached Storage (NAS)

A Network Attached Storage (NAS) system is single-purpose, dedicated file sharing device that attaches to the LAN, just like any server or workstation. With a NAS device, storage is not an integral part of the server. In fact, a NAS device does not need to be located within the server but can exist anywhere in a LAN and can be made up of multiple networked NAS devices.

Rather than containing a complete operating system, it is specifically designed to process only file reads and writes. A NAS has a proprietary file system and can serve files across platforms because it can read all major file systems. Because NAS appliances have the ability to share a data instance on multiple application servers, they are excellent cross-platform, collaborative tools.

NAS devices do not provide typical server-centric activities such as e-mail, authentication or file management. Instead, in the storage-centric design using a NAS device, the server still handles all of the processing of data but a NAS device delivers the data to the user. Note that although the capacity of a large NAS appliance can be in the terabyte range, it is still subject to the variable behavior and overhead of a network that may contain thousands of users.

Adding or removing NAS devices is similar to adding or removing any node in a network and can be accomplished without shutting the devices down for maintenance and upgrades.

Using traditional LAN protocols such as Ethernet and TCP/IP, the NAS enables additional storage to be quickly added by plugging it into a network hub or switch.

Storage Area Network (SAN)

A Storage Area Network (SAN) is a high-speed, multi-storage network that connects multiple servers to a centralized pool of shared disk storage devices (machines that contain nothing but a disk or disks for storing data). A SAN's architecture works in a way that makes all storage devices available to all servers on a LAN or WAN. As more storage devices are added to a SAN, they too are accessible from any server in the larger network.

Each server on the network is directly connected to the SAN and treats its allocated storage space like a directly-connected disk. Because of the direct connections to every server, the SAN model acts as a secondary network to the LAN. As such, this secondary network relieves the main network of massive data transfer loads because backup traffic occurs between storage devices inside the SAN.

SANs allow sharing the storage infrastructure without implying data sharing. This allows higher utilization of storage devices and easier reconfiguration than is possible with direct attached storage. SANs provide storage allocation flexibility through repartitioning and management tools. When storage space must be reallocated from one server to another, the SAN can simply be repartitioned. And, because stored data does not reside directly on any of a network's servers, server power is utilized for business applications and network capacity is released to the end user.

Compared to managing hundreds of servers, each with their own disks, SANs improve system administration. By treating all the company's storage as a single resource, disk maintenance and routine backups are easier to schedule and control. In some SANs, the disks themselves can copy data to other disks for backup without any processing overhead at the host computers.

The SAN network allows data transfers between computers and disks at the same high peripheral channel speeds as when they are directly attached. SANs can be centralized or distributed. A centralized SAN connects multiple servers to a collection of disks, whereas a distributed SAN typically uses one or more switches to connect nodes within buildings or campuses. Note that the channel-attached SAN must be taken offline to reconfigure it.

Note: SAN environments can be used with Windows but restrictions on the use apply; contact Consulting Services before implementing a SAN in a Windows environment.

UNIX-based Content Server clusters only work in SAN environments using third-party cluster software to virtualize the volume.

Comparison Summary: NAS/SAN

The following table summarizes NAS/SAN systems:

NAS System	SAN System
Transfers only files, not data blocks.	Transfers data in disk blocks (fixed-sized file chunks).
Uses an IP protocol to serve files to clients.	Uses a SCSI protocol to serve data blocks to servers.
Resembles a network server from which clients can obtain files.	Appears as local disk for each server.
Is a dedicated storage server that is directly connected to the network.	Requires a direct connection to each network server.
Sends files on request.	Grants direct access to disks.
Clients request files directly and bypass the general-purpose servers on the network.	Clients send file request to the servers on the network because the storage sits behind each server.

NAS System	SAN System
Backup traffic is routed over the main network.	Relieves the LAN of backup traffic since storage resides on a secondary network.
Ethernet connects the NAS appliance to the entire network.	SCSI or fibre channel connects each server to a SAN.
Additional devices are easily added to the network.	Very scalable and flexible (for example, 10TB of disk space can be added and designated to specific servers).
Offers high performance and cross-platform support (the capability of serving files to heterogeneous servers).	Provides high-speed, direct data access and joins several storage devices into a storage pool with server-assigned storage partitions.

Web Availability Devices

Improving a web site's availability can be done any number of ways.

Some techniques involve replicating the web site across multiple servers and load balancing the traffic among them. What appears to the outside world (i.e., "the Internet") as a single web site, might actually be a number of systems all running the same software and serving the same files.

Other techniques involve using specific load balancing hardware or involve using a domain name system (DNS) server and routing to affect the load balancing.

One of the benefits of both load balancing and round robin DNS is the ability to take servers out of rotation for upgrades or repairs without shutting down the site. Sites using this technique commonly take half of a server group out of rotation, upgrade them, run tests then add them back into rotation. The second half of the group is then taken out, upgraded, tested, and returned to the rotation. This process allows the web site to continue running while servers are upgraded and tested prior to being placed back into service. In this situation, the site runs at half its normal capacity only during upgrades and testing.

There are significant differences between load balancing and round robin DNS. Load balancing distributes connection loads across multiple servers, giving preference to those servers with the least amount of congestion. Round robin DNS provides congestion control by looping through the available servers. The primary difference between these two web-availability functionalities is that round robin DNS uses only a simple looping technique to perform load balancing while load balancing hardware usually sends traffic to servers based on the servers' current load (e.g., the server with the smallest number of active connections is sent the next connection).

Load Balancer

In clustering, load balancing involves evenly distributing the data, processing, and/or communications activity across available resources in a computer network so that no single device is overwhelmed. To accomplish this, the load balancer can evenly distribute the incoming transactions to all servers in the cluster or it might redirect them to the next available server. Load balancing is especially important for networks where it's difficult to predict the number of requests that will be issued to a server.

Round Robin Domain Name System (DNS)

A round robin domain name service (DNS) is a method of managing server congestion by distributing connection loads across multiple servers (with identical content). A single host name (e.g., http://www.AnyName.com) resolves to multiple addresses that are then used on a more or less equal basis. The round robin DNS process works on a rotating basis in that one server IP address is handed out and then moves to the back of the list. The next server IP address is handed out and then it moves to the end of the list, and so on, depending on the number of servers being used. This works in a looping fashion.

Avoid using pure round robin load balancing with your Content Server. A better solution is "sticky" round robin, where a particular client uses a particular node for a set period of time (for example, an hour). This type of round robin avoids caching issues that can occur with pure round robin load balancing. Using sticky round robin, the browser, web server filter, and Content Server all do some caching. When switching to a new node, the previous data found in the cache is recomputed until all the nodes have cached the appropriate data. This saves time as opposed to a pure round robin solution, where every request has to also be reauthenticated.

In order to implement sticky round robin load balancing, the load balancer has to track which client accessed which node and send requests from the same client to the same node.

Chapter

3

CLUSTER INSTALLATIONS

OVERVIEW

As described previously, every cluster installation depends heavily on the existing infrastructure and software at an organization. This chapter discusses some examples of cluster installations and what is needed for their successful use. It covers the following topics:

Important: Due to the complex interaction of cluster-related software and hardware components, you will need to retain Consulting Services to assist you in planning and implementing your clustered Content Server configuration.

Cluster Architecture

A standard clustered Content Server is defined as a high-performance shared-file system and database that consists of multiple servers sharing a Content Server instance. An optional load balancer or round robin domain name service (DNS) can be used to provide high availability for consumption.

The standard off-the-shelf architecture of the clustered Content Server includes the three main components of the Content Server:

- the content server binaries
- the Relational Database Management System
- the managed data (logical dataset) including the managed files and the index collection. The logical dataset consists of the file system, database and search index.

The basic clustered Content Server architecture is appropriate for medium to large enterprise customers or customers who have high up-time requirements for their Enterprise Content Management system. This architecture requires a fiber-attached shared disk to handle its infrastructure needs.

The basic cluster architecture allows for very high throughput with the ability to separate the Content Server components into multiple tiers (a distributed architecture). The Standard Content Server components (also collectively referred to as the dataset) include the:

- Relational Database Management System (RDBMS)
- File system (administration pages, binaries, executables, and the vault and web layout file repositories)

Important: If you are using third-party cluster software, you must use those agents to start and stop the content server.

Installation Architectures

The basic clustered Content Server can be implemented using the following different installation architectures that include:

- **Two-Tier Installations:** The most typical two-tier installation involves separating the Relational Database Management System from the server that holds the Content Server. This separation produces a gain of approximately 50 - 80% of the system resources for the Java core. (This installation architecture is used in approximately 95% of all Content ServerContent Server implementations.)

- **Web-Tier or Multi-Tiered Installations:** In the web-tier installation all of the following components are separated onto different servers:
  - the application tier, containing other components such as Publisher, Refinery, and so on
  - data tier (the RDBMS)
  - web tier, containing the web servers

  Each tier allows complete separation of all components that require resources.

The installation architecture of clustered Content Server configurations is n-tier, modular, and standards-based. This enables each installation to extend and scale to support changing enterprise requirements. This modularity allows clustered Content Server implementations to comprise a tiered cluster in a single, local setting or a geographically dispersed configuration with multiple server clusters in multiple locations.

See Sample Installation Configurations (page 3-6) for graphic illustrations of these different types of cluster configurations.

General N-Tier Characteristics

The following characteristics are typical of all architectures for clustered Content Server installations:

- The end-user client part consists simply of a Web browser.
- The server-side presentation logic consists of a standard Web server and dynamic delivery of content from the Content Server.
- The server-side business logic and services (processing and data manipulation) includes the Content Server, stub directories and other optional add-on modules for conversion and publishing.
- Optional front-end or back-end nodes incorporated into an installation architecture may be one or more of several types of components including firewalls, load balancers, web servers, gateways, or proxy servers.
- The resource (shared storage) architectural components consist of a standard relational database, a search index and managed data/managed files.

- ❖ Depending upon specific architectural requirements, the shared storage can be either a NAS or SAN and will hold the search index, common files, and the vault and web layout file repositories.
- ❖ Stub directories are identical on all servers within a clustered Content Server implementation. However, configuration entries in the directories are unique for each node.
- ❖ When all the processes associated with the file system are configured to run on separate machines, clustering the file system is relatively simple.
- ❖ Because the clustered Content Server is based on a shared file system, all of the servers within a cluster share the system resources that include the database and shared storage components.
- ❖ The serviceability of an installation architecture is directly correlated to the number of nodes in the cluster(s).

Two-Tier Installation Architecture

The two-tier installation architecture consists of load-balanced clustered servers and shared resources. One tier contains the front-end components along with the clustered servers that host the Content Server. A second tier contains the separated RDBMS. In this implementation, all the business logic and services, content server instances, corresponding stub directories, the presentation (web) logic, and the shared storage system components are hosted within the clustered Content Servers.

Advantages:

- ❖ Separating the RDBMS from the clustered content servers improves overall system performance — the installation gains 50-80% of the system resources for the Java core.
- ❖ The load-balanced clustered Content Servers support demanding consumption and contribution of content and provide failover protection.
- ❖ Because the clustered Content Servers are load balanced, contribution and consumption servers can be segregated from each other through dedicated servers or all requests can be shared equally among the servers.
- ❖ This configuration is easily scalable given that additional servers can be incorporated with few modifications.

- The serviceability requirements of this system is less demanding than the three-tier or four-tier architectures since all of the application and web logic is consolidated within the cluster.

Three-Tier (Multi-Tier) Installation Architecture

The three-tier installation architecture separates the RDBMS and file system from the application logic that is, in turn, separated from the web servers or presentation layer.

Advantages:

- Separating the clustered web servers from the clustered application servers is particularly effective for installations that have high security demands for both consumption and contribution activities.
- The load-balanced clustered Content Servers support demanding consumption and contribution of content and provide failover protection.
- Because the clustered Content Servers are load balanced, contribution and consumption servers can be segregated from each other through dedicated servers or all requests can be shared equally among the servers.
- The application logic is hosted in a more modular fashion than in the two-tier installation architecture because the presentation and business logic resides in different tiers and clusters.
- The presentation and business logic tiers can be scaled separately — additional nodes can be added to the presentation cluster without adding additional nodes to the back-end cluster and vice versa.

  For example, if only half of the system requests require the use of a back-end cluster component, then the presentation tier might reach capacity before the back-end tier does. Additional servers can be added to the presentation cluster to offset the increased traffic.

  Conversely, if many of the actions on the back-end tier are batch-like (long-running and processor-intensive) processes, additional servers can be added to the back-end cluster even though the presentation cluster has not reached its maximum capacity.
- It is feasible that failover of requests can be transferred to the back-end tier. Failure of a server in the back-end cluster does not disrupt request processing since it is possible for a component in the presentation logic tier to transfer a request from the failed machine to an active machine.

The presentation tier can resist back-end server failures by enabling failover of requests. Additionally, if the entire back-end tier fails, the presentation tier can still be active and handle requests that do not require back-end access or processing. Unlike the two-tier architecture, a complete system failure does not prevent simple requests from being processed.

Disadvantages:

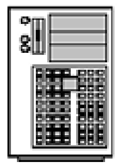
- Separating the clustered web servers from the clustered application servers actually causes the Content Server to perform less efficiently. Therefore, this is only done when security is of primary importance.
- In this multi-tiered installation architecture, the reliability of certain services is higher because inter-process communication is a requirement for front-end components to communicate with back-end components. The supplementary effort of additional inter-process communication increases the request response time.
- Implementation might be long and complex depending upon the number of required installations of content server instances.
- This installation architecture has more demanding serviceability requirements than the two-tier system. Two clusters rather than one will need to be managed along with the connections between components in each cluster.
- Load balancing both cluster tiers is needed. If the load for the presentation tier and the business logic tier varies, load-balancing the back-end requests provides processing distribution.

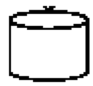
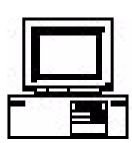
SAMPLE INSTALLATION CONFIGURATIONS

Note: As stated previously, these configurations are intended to be guidelines and testing scenarios, and are not meant to be strictly followed protocols on how to set up your cluster configuration.

The following figures illustrate some of the different configurations that are possible with Content Server in a clustered environment. Note that it is assumed that the web server is on the same machine as the Content Server if the web server is not shown in the graphic.

The following terms and symbols are used to illustrate these cluster configuration examples:

Symbols	Description
	**Content Server or Content Server Product:** This can include the core Content Server as well as products such as Refinery, Publisher, and so on.
	**Web Server:** A computer that delivers Web pages. Any computer can be a web server by installing server software and connecting the machine to the Internet. In some of the illustrations which follow, the web server and the Content Server can be combined in one machine. Many web server software applications can be used, such as public domain software from NCSA and Apache and commercial packages from Microsoft, Netscape, and others.
	**Database:** The Relational Database Management System used with the Content Server. This includes the indexing system.

Symbols	Description
	**Network Storage Technology:** A NAS device (network-attached storage) or a SAN (storage area network). A NAS device is a disk that is dedicated to file sharing. It does not provide server activities such as e-mail, user authentication, or file management. NAS allows more hard disk storage space to be added to a network that already utilizes servers without shutting them down for maintenance and upgrades. With a NAS device, storage is not an integral part of the server. The server still handles all of the processing of data but a NAS device delivers the data to the user. A NAS device does not need to be located within the server but can exist anywhere in a LAN and can be made up of multiple networked NAS devices. Storage Area Network (SAN) is a high-speed network of shared storage devices. A storage device is a machine that contains a disk or disks for storing data. A SAN's architecture works in a way that makes all storage devices available to all servers on a LAN or WAN. As more storage devices are added to a SAN, they too will be accessible from any server in the larger network.
	**Load Balancer:** A computer that evens out network bandwidth on all network interface cards, or balances the connection-oriented network traffic to be dynamically switched between network interface cards.
	**Switch:** A smart hub, usually required after most load balancers.
	**End User:** An individual who uses a computer, program, network, or related service for work or entertainment. Usually someone who is not a computer developer or administrator.

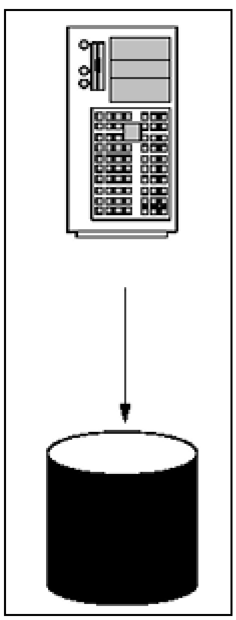
Other elements not depicted in these illustrations:

- Stub_dir: stub directories contain only a subset of the files of an entire program. The stub_dir in a clustered Content Server configuration includes the files located on the load balanced machines and other subdirectories.
- firewall: a firewall is a system designed to prevent unauthorized access to or from a private network. Firewalls can be implemented in both hardware and software, or a combination of both.
- switch: a device that routes traffic by filtering or forwarding data packets between LAN segments.

Simple Install

This illustration shows the most basic installation of a Content Server. This type of installation is used by the majority of Content Server customers.

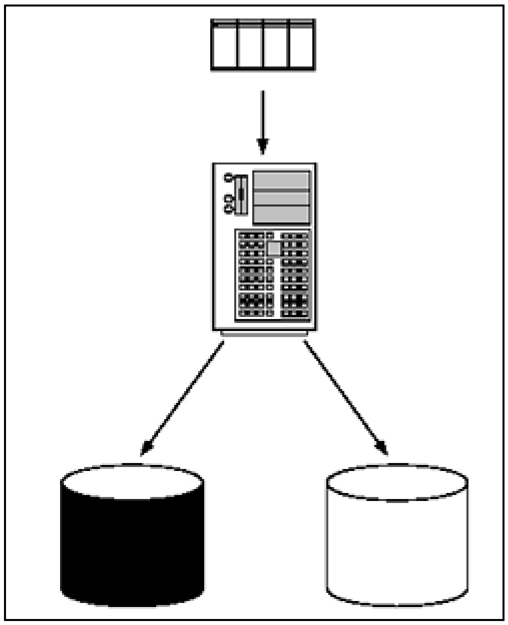
Figure 3-1 Basic installation



Basic N’Tier Install

The following illustration shows a simple n’tier installation, with the content server, the web server, and RDBMS with SAN/NAS split into separate tiers.

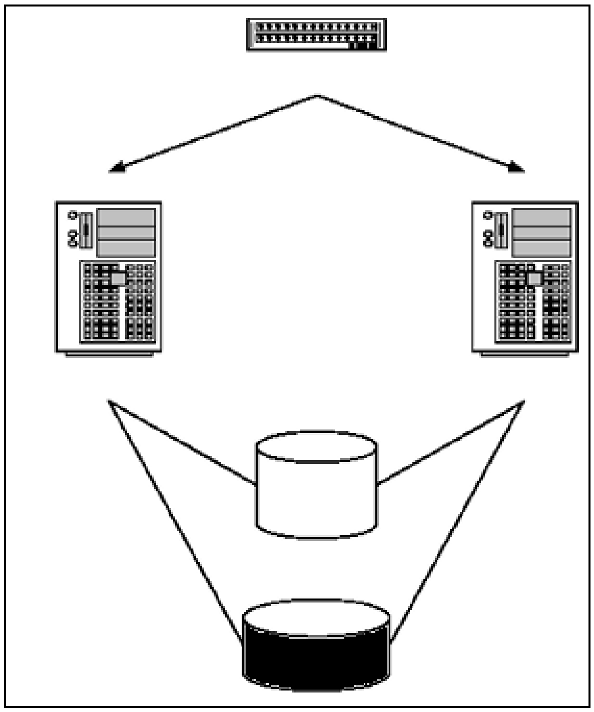
Figure 3-2 Basic n-tier installation



Basic Cluster Install

The following illustration shows a simple cluster installation with a load balancer and two Content Servers, a SAN/NAS appliance, and a RDBMS. The load balancer evens out network bandwidth or balances the connection-oriented network traffic.

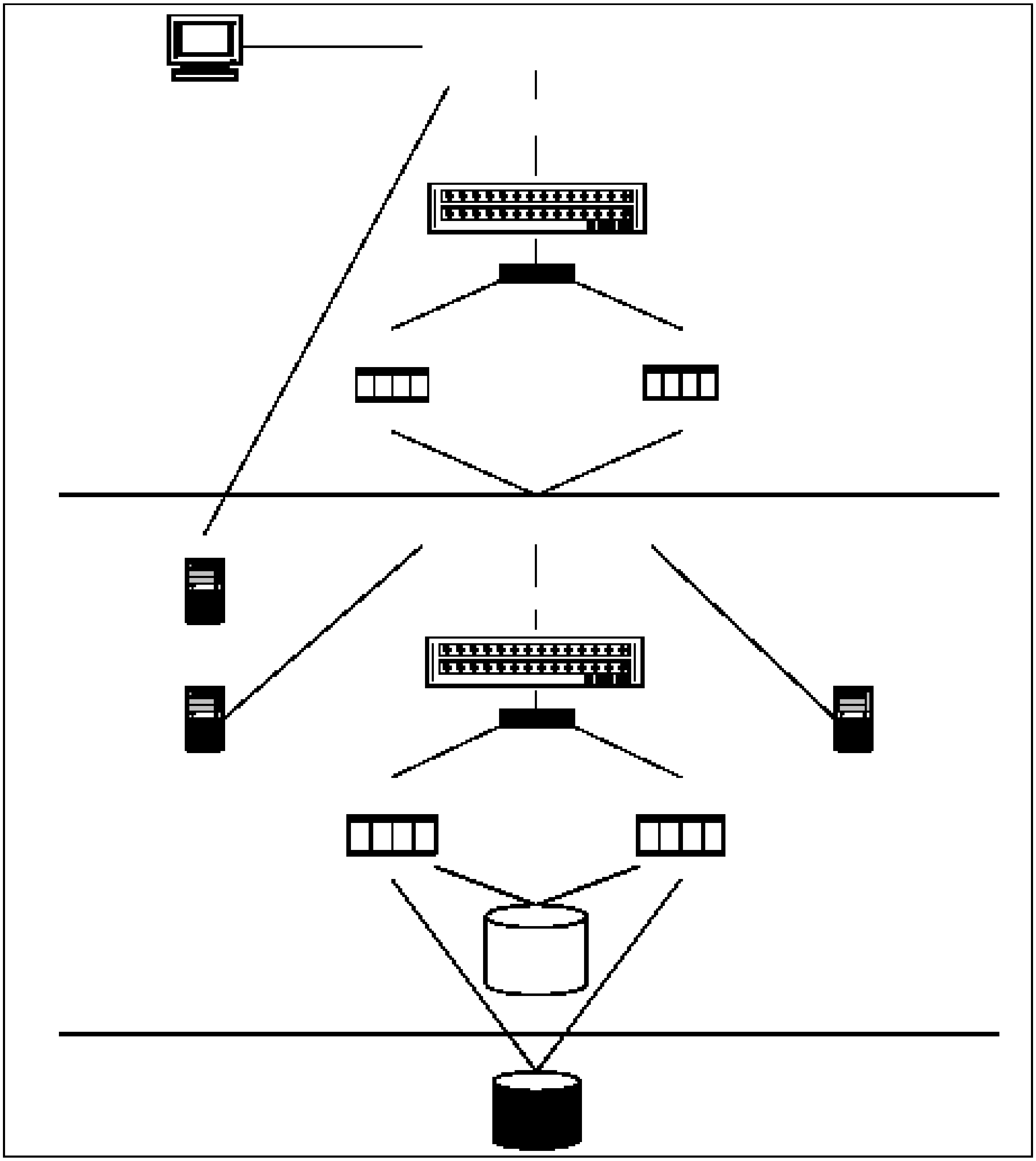
Figure 3-3 Basic cluster install



Multi-Tier Cluster Install

This tiered example illustrates another possible cluster configuration. Note the three 'tiers': the web, application, and data layers. The use of multi-tiered installation architectures combined with clustered Content Servers supports flexible implementation options.

Figure 3-4 Multi-tier cluster install



GEOGRAPHIC CLUSTER DISTRIBUTIONS

Geographic clusters can be used by large sites to assist in fail-over and in geographic load balancing.

In a geographic cluster (sometimes called a consumption/contribution cluster), an organization will set up clusters in different physical locations. Site A may be set up as a a contribution cluster/site, using a Content Server with other Content Server products (Refinery, Publisher, and so on). This would be designated as ‘contribution’ because all contribution from the organization passes through that site. This is considered the ‘master’ or ‘secure’ site.

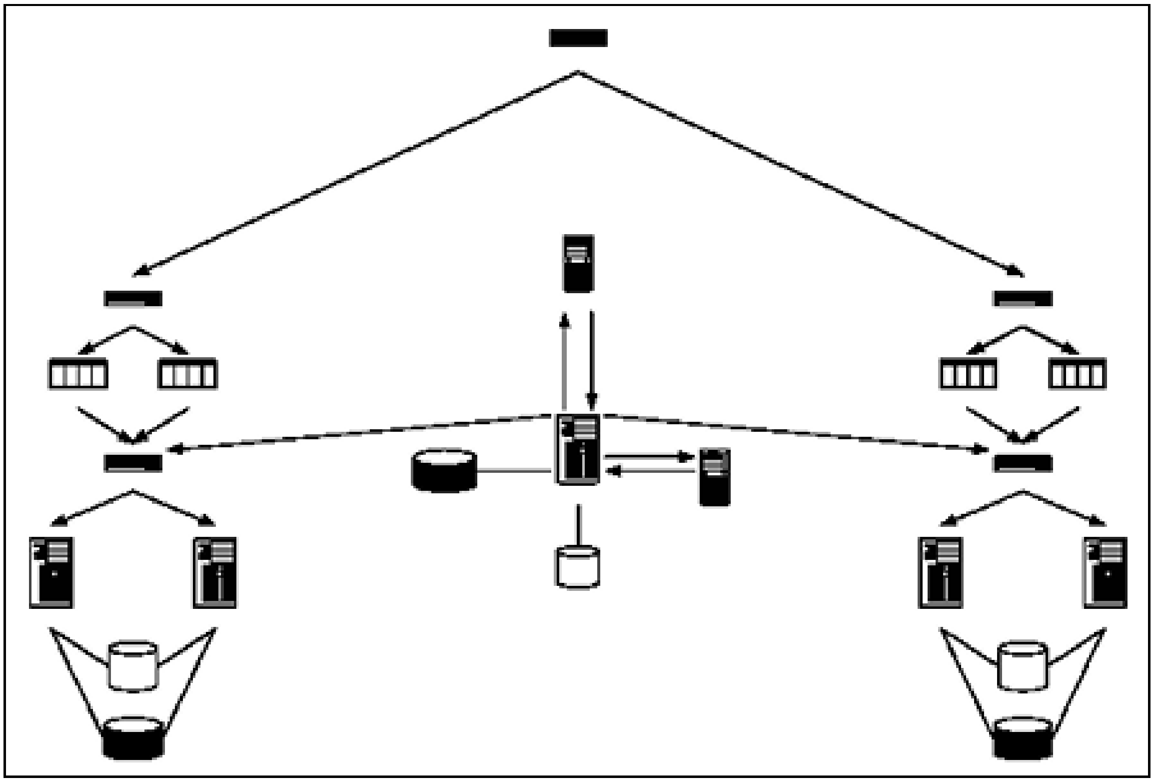
Sites B and C, located in other cities or countries, are designated as consumption sites. These are multi-tiered installed cluster sites with their own load balancers and web servers. They are tuned to operate at a high efficiency and are not considered as secure as Site A, the contribution site. These sites operate independently of the contribution site except that data from the contribution site is shared to the consumption sites.

The contribution sites can be kept synchronized by using replication and migration. In the event the contribution site experiences failure, one of those sites can be used as the primary contribution site. This can pose problems, however and those problems should be addressed before a disaster recovery plan is needed:

- If Site B, for example, has to become the main contribution site, it will soon be unsynchronized with the other sites. How long is this acceptable? Is there a plan in place to re-synchronize all sites if Site B returns to being a consumption site?
- If synchronization is a problem, will you move Site B to become the new contribution site and later re-task Site A to be a consumption site? What will be involved in that transition and how will it be handled?
- What is the acceptable length of time that a contribution site can be down?
- Site B and C, the consumption sites, were configured as consumption sites and not as contribution sites. Is the proper equipment and personnel in place if one of those sites has to shift over to a contribution site?

As these questions indicate, a disaster-recovery plan should be part of your cluster planning so that the proper equipment and software are in place. It is not enough to simply establish distributed clusters in the event of a failure of one cluster. You must know how to implement a backup cluster in case it is needed.

Figure 3-5 Geographic distribution



Index

I

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

S