WebSphere Application Server

Concepts, Planning, and Installation for Edge Components

Version 6.1

GC31-6918-00

WebSphere Application Server

Concepts, Planning, and Installation for Edge Components

Version 6.1

GC31-6918-00

Note
Before using this information and the product it supports, be sure to read the general information under "Notices" on page 81.

First edition (May 2006)

This edition applies to:
WebSphere Application Server, Version 6.1

and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications through your IBM representative or through the IBM branch office serving your locality.

Contents

Figures

About this book

This book, *WebSphere® Application Server Concepts, Planning, and Installation for Edge Components*, serves as an introduction to the WebSphere Application Server Edge components. It provides high-level product overviews, detailed functionality discussions for key components, edge-of-the-network scenarios, installation and initial configuration information, and demonstration networks.

Who should read this book

WebSphere Application Server Concepts, Planning, and Installation for Edge Components is written for experienced network and system administrators who are familiar with their operating systems and with providing Internet services. Prior exposure to the WebSphere Application Server or to WebSphere Application Server Edge components is not required.

Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. These are the major accessibility features in WebSphere Application Server, Version 6.1:

- You can use screen-reader software and a digital speech synthesizer to hear what is displayed on the screen. You can also use voice recognition software, such as IBM® ViaVoice™, to enter data and to navigate the user interface.
- You can operate features by using the keyboard instead of the mouse.
- You can configure and administer Application Server features by using standard text editors or command-line interfaces instead of the graphical interfaces provided. For more information about the accessibility of particular features, refer to the documentation about those features.

Conventions and terminology used in this book

This documentation uses the following typographical and keying conventions.

Table 1. Conventions used in this book

Convention	Meaning
Bold	When referring to graphical user interfaces (GUIs), bold face indicates menus, menu items, labels, buttons, icons, and folders. It also can be used to emphasize command names that otherwise might be confused with the surrounding text.
`Monospace`	Indicates text you must enter at a command prompt. Monospace also indicates screen text, code examples, and file excerpts.
Italics	Indicates variable values that you must provide (for example, you supply the name of a file for *fileName*). Italics also indicates emphasis and the titles of books.
Ctrl-*x*	Where *x* is the name of a key, indicates a control-character sequence. For example, Ctrl-c means hold down the Ctrl key while you press the c key.
Return	Refers to the key labeled with the word Return, the word Enter, or the left arrow.
%	Represents the Linux and UNIX® command-shell prompt for a command that does not require root privileges.
#	Represents the Linux and UNIX command-shell prompt for a command that requires root privileges.

Table 1. Conventions used in this book (continued)

Convention	Meaning
`C:\`	Represents the Windows command prompt.
Entering commands	When instructed to "enter" or "issue" a command, type the command and then press Return. For example, the instruction "Enter the **ls** command" means type **ls** at a command prompt and then press Return.
[ ]	Enclose optional items in syntax descriptions.
{ }	Enclose lists from which you must choose an item in syntax descriptions.
|	Separates items in a list of choices enclosed in { }(braces) in syntax descriptions.
...	Ellipses in syntax descriptions indicate that you can repeat the preceding item one or more times. Ellipses in examples indicate that information was omitted from the example for the sake of brevity.

Part 1. Overview

This part introduces the WebSphere Application Server Edge components, Caching Proxy and Load Balancer, and discusses their integration with Application Server. It also defines the components of Caching Proxy and Load Balancer. In addition, this section introduces other related WebSphere family products.

This part contains the following chapters:

Chapter 1. Introducing WebSphere Application Server Edge components

WebSphere is Internet infrastructure software that enables companies to develop, deploy, and integrate next-generation e-business applications such as those for business-to-business e-commerce. WebSphere middleware supports business applications from simple Web publishing through enterprise-scale transaction processing.

As the foundation of the WebSphere platform, WebSphere Application Server offers a comprehensive set of middleware that enables users to design, implement, deploy, and manage business applications. These applications can range from a simple Web site storefront to a complete revision of an organization's computing infrastructure.

Processor-intensive features, such as personalization, offer a competitive advantage to every e-business. However, habitually relegating these features to central servers can prevent valuable functions from scaling to Internet proportions. Consequently, with the constant addition of new Web applications, a business's Internet infrastructure must grow in scope and impact. In addition, reliability and security are extraordinarily important to an e-business. Even a minimal service disruption can result in a loss of business.

Edge components (formerly Edge Server) are now a part of the WebSphere Application Server offering. Edge components can be used in conjunction with WebSphere Application Server to control client access to Web servers and to enable business enterprises to provide better service to users who access Web-based content over the Internet or a corporate intranet. Using Edge components can reduce Web server congestion, increase content availability, and improve Web server performance. As the name indicates, Edge components usually run on machines that are close (in a network configuration sense) to the boundary between an enterprise's intranet and the Internet.

The WebSphere Application Server includes the Caching Proxy and Load Balancer Edge components.

IMPORTANT: Caching Proxy is available on all Edge component installations, with the following exceptions:

- Caching Proxy is not available for Edge component installations that run on Itanium 2 or AMD Opteron 64-bit processors.
- Caching Proxy is not available for Edge component installations of Load Balancer for IPv4 and IPv6.

Caching Proxy

Caching Proxy reduces bandwidth use and improves a Web site's speed and reliability by providing a point-of-presence node for one or more back-end content servers. Caching Proxy can cache and serve static content and content dynamically generated by WebSphere Application Server.

Caching Proxy can be configured in the role of a reverse proxy server (default configuration) or a forward proxy server, providing either a point-of-presence for a

network or an internal network server tasked with improving request and response time. For more information on reverse and forward configurations, see “Basic Caching Proxy configurations” on page 13.

The proxy server intercepts data requests from a client, retrieves the requested information from content-hosting machines, and delivers that content back to the client. Most commonly, the requests are for documents stored on Web server machines (also called *origin servers* or *content hosts*) and delivered using the Hypertext Transfer Protocol (HTTP). However, you can configure the proxy server to handle other protocols, such as File Transfer Protocol (FTP) and Gopher.

The proxy server stores cacheable content in a local cache before delivering it to the requester. Examples of cacheable content include static Web pages and JavaServer Pages files that contain dynamically generated, but infrequently changing, information. Caching enables the proxy server to satisfy subsequent requests for the same content by delivering it directly from the local cache, which is much quicker than retrieving it again from the content host.

Plug-ins for Caching Proxy add functionality to the proxy server.

- The ICP plug-in enables the proxy server to query Internet Caching Protocol (ICP)-compliant caches in search of HTML pages and other cacheable resources.
- The Tivoli® Access Manager (formerly Policy Director) plug-in enables the proxy server to use Access Manager’s integrated authorization or authentication services.
- The PAC-LDAP Authentication Module enables the proxy server to access an LDAP server when performing authorization or authentication routines.
- The WebSphere Transcoding Publisher plug-in allows the proxy server to cache multiple transcoded versions of content for mobile devices when used in conjunction with WebSphere Transcoding Publisher.

You can further extend the functions of Caching Proxy by writing custom plug-in modules to an application programming interface (API). The API is flexible, easy to use, and platform independent. The proxy performs a sequence of steps for each client request it processes. A plug-in application modifies or replaces a step within the request-processing workflow, such as client authentication or request filtering. The powerful Transmogrify interface, for example, provides access to HTTP data and enables substitution or transformation of URLs and Web content. Plug-ins can modify or replace designated processing steps, and you can invoke more than one plug-in for a particular step.

Load Balancer

Load Balancer creates edge-of-network systems that direct network traffic flow, reducing congestion and balancing the load on various other services and systems. Load Balancer provides site selection, workload management, session affinity, and transparent failover.

Load Balancer is installed between the Internet and the enterprise’s back-end servers, which can be content hosts or Caching Proxy machines. Load Balancer acts as the enterprise’s single point-of-presence node on the Internet, even if the enterprise uses multiple back-end servers because of high demand or a large amount of content. You can also guarantee high availability by installing a backup Load Balancer to take over if the primary one fails temporarily.

Load Balancer intercepts data requests from clients and forwards each request to the server that is currently best able to fill the request. In other words, it balances the load of incoming requests among a defined set of machines that service the same type of requests. Load Balancer can distribute requests to many types of servers, including WebSphere Application Servers and Caching Proxy machines. Load balancing can be customized for a particular application or platform by using custom advisors. Special purpose advisors are available to obtain information for load balancing WebSphere Application Servers.

If the Content Based Routing component is installed together with the Caching Proxy, HTTP and HTTPS requests can even be distributed based on URLs or other administrator-determined characteristics, eliminating the need to store identical content on all back-end servers. The Dispatcher component can also provide the same function for HTTP requests.

Load balancing improves your Web site's availability and scalability by transparently clustering content servers, including HTTP servers, application servers, and proxy servers, which are surrogate content servers. Availability is achieved through parallelism, load balancing, and failover support. When a server is down, business is not interrupted. An infrastructure's scalability is greatly improved because back-end processing power can be added transparently.

Support for IPv6: Support for the extended IP addressing scheme of IPv6 is available with "Load Balancer for IPv4 and IPv6." Load Balancer for IPv4 and IPv6 is a separate install image consisting of *only* the Dispatcher component. This installation type provides load balancing for both IPv4 and IPv6 traffic to servers configured within your network using Dispatcher's MAC-based packet forwarding. It is important to note that any previous Load Balancer must be uninstalled, before installing Load Balancer for IPv4 and IPv6. Two Load Balancers cannot be installed on the same machine. (See "Dispatcher," for a brief overview of the Dispatcher component.)

Load Balancer includes the following components:

Dispatcher

For all Internet services, such as HTTP, FTP, HTTPS, and Telnet, the Dispatcher component performs load balancing for servers within a local area network (LAN) or wide area network (WAN). For HTTP services, Dispatcher can perform load balancing of servers based on the URL content of the client request.

The Dispatcher component enables stable, efficient management of a large, scalable network of servers. With Dispatcher, you can link many individual servers into what appears to be a single virtual server. Your site thus appears as a single IP address to the world.

If you are using a Load Balancer for IPv4 and IPv6 installation, see the chapter for deploying Dispatcher on Load Balancer for IPv4 and IPv6 in the *WebSphere Application Server Load Balancer Administration Guide*, which includes information on limitations and configuration differences.

Content Based Routing

For HTTP and HTTPS services, the Content Based Routing component performs load balancing for servers based on the content of the client request. The Content Based Routing component works in conjunction with the Application Server Caching Proxy component.

IMPORTANT: The Content Based Routing (CBR) component is available on all supported platforms, with the following exceptions:

- CBR is not available with Load Balancer installations on platforms running a 64-bit JVM.

  Alternatively, for this type of installation, you can use the cbr forwarding method of Load Balancer's Dispatcher component to provide content-based routing of HTTP and HTTPS requests without the use of Caching Proxy. See the *WebSphere Application Server Load Balancer Administration Guide* for more information.
- CBR is not available with Load Balancer for IPv4 and IPv6 installations.

  Load Balancer for IPv4 and IPv6 supports only the Dispatcher component's mac forwarding method. The nat and cbr forwarding methods are not supported.

Site Selector

The Site Selector component enhances a load-balancing system by allowing it to act as the point-of-presence node for a network and load balance incoming requests by mapping DNS names to IP addresses. In conjunction with Metric Server, Site Selector can monitor the level of activity on a server, detect when a server is the least heavily loaded, and detect a failed server.

This component is supported on all Edge component installations, with the following exception:

- This component is not available with Edge component installations of Load Balancer for IPv4 and IPv6.

Cisco CSS Controller

The Cisco CSS Controller component generates server-weighting metrics that are sent to a Cisco CSS switch for server selection, load optimization, and fault tolerance.

This component is supported on all Edge component installations, with the following exception:

- This component is not available with Edge component installations of Load Balancer for IPv4 and IPv6.

Nortel Alteon Controller

The Nortel Alteon Controller component generates server-weighting metrics that are sent to a Nortel Alteon switch for server selection, load optimization, and fault tolerance.

This component is supported on all Edge component installations, with the following exception:

- This component is not available with Edge component installations of Load Balancer for IPv4 and IPv6.

Metric Server

The Metric Server component runs as a daemon on a load-balanced server and provides information about system loads to Load Balancer components.

Chapter 2. Edge components and the WebSphere family

The IBM WebSphere family is designed to help users realize the promise of e-business. It is a set of software products that helps users develop and manage high-performance Web sites and integrate Web sites with new or existing non-Web business information systems.

The WebSphere family consists of WebSphere Application Server, including the Edge components, and other WebSphere family software that is tightly integrated with the WebSphere Application Server and enhances its performance. For an overview of WebSphere Application Server and its components, see Chapter 1, "Introducing WebSphere Application Server Edge components," on page 3.

Tivoli Access Manager

Tivoli Access Manager (formerly Tivoli Policy Director) is available separately. It provides access control and centralized security for existing Web applications and offers one-time authentication capability with access to multiple Web resources. A Caching Proxy plug-in exploits Access Manager's security framework, enabling the proxy server to use Access Manager's integrated authorization or authentication services.

WebSphere Portal Server

WebSphere Portal Server (available separately) offers a framework to meet the presentation, security, scalability, and availability issues associated with portals. Using Portal Server, companies can build their own custom portal Web site to serve the needs of employees, business partners, and customers. Users can sign on to the portal and receive personalized Web pages that provide access to the information, people, and applications they need. This personalized single point of access to all necessary resources reduces information overload, accelerates productivity, and increases Web site usage.

WebSphere Portal Server runs in a WebSphere Application Server cluster to achieve scalability and reliability. The Application Server Load Balancer component can also be used for additional load balancing and high availability.

WebSphere Site Analyzer

WebSphere Site Analyzer (available separately) helps enterprises to anticipate capacity and performance problems. With Site Analyzer, Caching Proxy and Load Balancer logs and other manageability aids can be used to anticipate the demand for additional resources by monitoring, analyzing, and reporting your Web site usage. In addition, Site Analyzer manageability components assist users who install and upgrade Edge components, manage and store configurations, operate Edge components remotely, and view and report events.

WebSphere Transcoding Publisher

WebSphere Transcoding Publisher (available separately) can convert a Web page for viewing on a mobile device, such as an Internet-capable phone, translate Web content to the user's preferred national language (by invoking WebSphere Translation Server), and convert markup languages. Transcoding Publisher enhances Caching Proxy's capabilities by allowing it to serve content for different devices and users. After accessing content from a Web server, Caching Proxy's Transmogrify interface can be configured to invoke Transcoding Publisher to transform the data and tag it for variant caching and possible reuse. At Caching Proxy's post-authentication interface, Transcoding Publisher then checks the proxy server for content matching the user and device requirements and, if a match is found, serves the content from the proxy server's cache.

Chapter 3. More information on Application Server and Edge components

The following documentation specific to the WebSphere Application Server Edge Components is available in the Edge Components Information Center.

- *Programming Guide for Edge Components* GC31-6919-00
- *Caching Proxy Administration Guide* GC31-6920-00
- *Load Balancer Administration Guide* GC31-6921-00

Other WebSphere Application Server documentation is available from the WebSphere Application Server library page.

Technote support information on Edge Components is available from the WebSphere Application Server support page.

The following is a list of Web sites to obtain information on Edge Components or related information:

- IBM Web site home *http://www.ibm.com/*
- IBM WebSphere Application Server *http://www.ibm.com/software/webservers/appserv/*
- IBM WebSphere Application Server library Web site *http://www.ibm.com/software/webservers/appserv/library.html*
- IBM WebSphere Application Server support Web site *http://www.ibm.com/software/webservers/appserv/support.html*
- IBM WebSphere Application Server Information Center *http://www.ibm.com/software/webservers/appserv/infocenter.html*
- IBM WebSphere Application Server Edge Components Information Center *http://www.ibm.com/software/webservers/appserv/ecinfocenter.html*

Part 2. Edge component concepts and discussions

This part includes detailed discussions that highlight some of the functionality available with Edge components. See Chapter 1, "Introducing WebSphere Application Server Edge components," on page 3 for an overview of the Application Server's Caching Proxy component.

This part contains the following chapters:

Chapter 4. Caching

Caching Proxy's caching functionality helps to minimize network bandwidth utilization and ensure that end users receive faster, more reliable service. This is accomplished because the caching performed by the proxy server offloads back-end servers and peering links. Caching Proxy can cache static content and content dynamically generated by WebSphere Application Server. To provide enhanced caching, Caching Proxy also functions in conjunction with the Application Server Load Balancer component. See Chapter 1, "Introducing WebSphere Application Server Edge components," on page 3 for an introduction to these systems.

IMPORTANT: Caching Proxy is available on all Edge component installations, with the following exceptions:

- Caching Proxy is not available for Edge component installations that run on Itanium 2 or AMD Opteron 64-bit processors.
- Caching Proxy is not available for Edge component installations of Load Balancer for IPv4 and IPv6.

Basic Caching Proxy configurations

Caching Proxy can be configured in the role of a reverse caching proxy server (default configuration) or a forward caching proxy server. When used by content hosts, the Caching Proxy is configured in the role of reverse caching proxy server, located between the Internet and the enterprises's content hosts. When used by Internet access providers, the Caching Proxy is configured in the role of a forward caching proxy server, located between a client and the Internet.

Reverse Caching Proxy (default configuration)

When using a reverse proxy configuration, Caching Proxy machines are located between the Internet and the enterprise's content hosts. Acting as a surrogate, the proxy server intercepts user requests arriving from the Internet, forwards them to the appropriate content host, caches the returned data, and delivers that data to the users across the Internet. Caching enables Caching Proxy to satisfy subsequent requests for the same content directly from the cache, which is much quicker than retrieving it again from the content host. Information can be cached depending on when it will expire, how large the cache should be and when the information should be updated. Faster download times for cache hits mean better quality of service for customers. Figure 1 on page 14 depicts this basic Caching Proxy functionality.

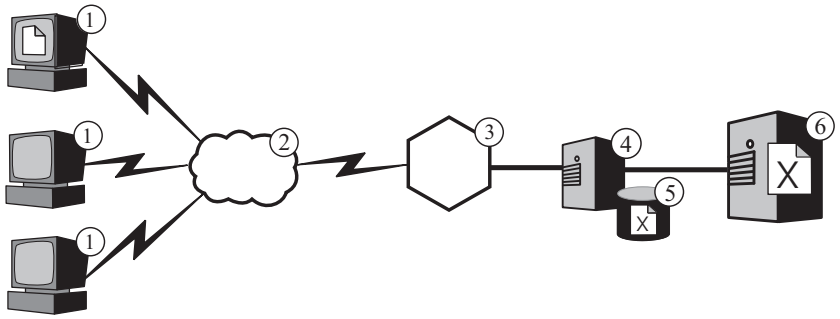

Figure 1. Caching Proxy acting as a reverse proxy. Legend:
1—Client 2—Internet 3—Router/Gateway 4—Caching Proxy 5—Cache 6—Content host

In this configuration, the proxy server (**4**) intercepts requests whose URLs include the content host's host name (**6**). When a client (**1**) requests file **X**, the request crosses the Internet (**2**) and enters the enterprise's internal network through its Internet gateway (**3**). The proxy server intercepts the request, generates a new request with its own IP address as the originating address, and sends the new request to the content host (**6**).

The content host returns file **X** to the proxy server rather than directly to the end user. If the file is cacheable, Caching Proxy stores a copy in its cache (**5**) before passing it to the end user. The most prominent example of cacheable content is static Web pages; however, Caching Proxy also provides the ability to cache and serve content dynamically generated by WebSphere Application Server.

Forward Caching Proxy

Providing direct Internet access to end users can be very inefficient. Every user who fetches a given file from a Web server generates the same amount of traffic in your network and through your Internet gateway as the first user who fetched the file, even if the file has not changed. The solution is to install a forward Caching Proxy near the gateway.

When using a forward proxy configuration, Caching Proxy machines are located between the client and the Internet. Caching Proxy forwards a client's request to content hosts located across the Internet, caches the retrieved data, and delivers the retrieved data to the client.

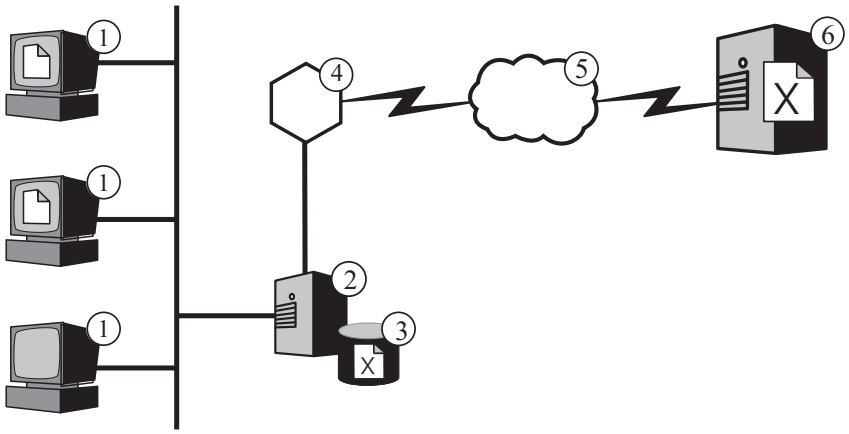

Figure 2. Caching Proxy acting as a forward proxy. Legend: 1—Client 2—Caching Proxy 3—Cache 4—Router/Gateway 5—Internet 6—Content host

Figure 2 depicts the forward Caching Proxy configuration. The clients' browser programs (on the machines marked 1) are configured to direct requests to the forward caching proxy (2), which is configured to intercept the requests. When an end user requests file X stored on the content host (6), the forward caching proxy intercepts the request, generates a new request with its own IP address as the originating address, and sends the new request out by means of the enterprise's router (4) across the Internet (5).

In this way the origin server returns file X to the forward caching proxy rather than directly to the end user. If the caching feature of the forward Caching Proxy is enabled, Caching Proxy determines whether file X is eligible for caching by checking settings in its return header, such as the expiration date and an indication whether the file was dynamically generated. If the file is cacheable, the Caching Proxy stores a copy in its cache (3) before passing it to the end user. By default, caching is enabled and the forward Caching Proxy uses a memory cache; however, you can configure other types of caching.

For the first request for file X, forward Caching Proxy does not improve the efficiency of access to the Internet very much. Indeed, the response time for the first user who accesses file X is probably slower than without the forward caching proxy, because it takes a bit more time for the forward Caching Proxy to process the original request packet and examine file X's header for cacheability information when it is received. Using the forward caching proxy yields benefits when other users subsequently request file X. The forward Caching Proxy checks that its cached copy of file X is still valid (has not expired), and if so it serves file X directly from the cache, without forwarding the request across the Internet to the content host.

Even when the forward Caching Proxy discovers that a requested file is expired, it does not necessarily have to refetch the file from the content host. Instead, it sends a special status checking message to the content host. If the content host indicates that the file has not changed, the forward caching proxy can still deliver the cached version to the requesting user.

Configuring the forward Caching Proxy in this way is termed forward proxy, because the Caching Proxy is acting on behalf of browsers, forwarding their requests to content hosts via the Internet. The benefits of forward proxy with caching are two-fold:

- If a file is cached, end users receive it much more quickly than when their requests must cross the Internet, because the forward caching proxy is on the local network. As more and more files are cached, the total response time that users experience for Internet requests continues to go down.
- There is no traffic generated outside the enterprise's local network. This effectively increases the capacity (available bandwidth) of the enterprise's gateway to the Internet by freeing it to handle requests for files that are not cached. It also reduces Internet access charges, which is especially important in environments where such charges are based on the number of packets.

Caching Proxy can proxy several network transfer protocols, including HTTP (Hypertext Transfer Protocol, FTP (File Transfer Protocol), and Gopher.

Transparent forward Caching Proxy (Linux systems only)

A variation of the forward Caching Proxy is a transparent Caching Proxy. In this role, Caching Proxy performs the same function as a basic forward Caching Proxy, but it does so without the client being aware of its presence. The transparent Caching Proxy configuration is supported on Linux systems only.

In the configuration described in "Forward Caching Proxy" on page 14, each client browser is separately configured to direct requests to a certain forward Caching Proxy. Maintaining such a configuration can become inconvenient, especially for large numbers of client machines. The Caching Proxy supports several alternatives that simplify administration. One possibility is to configure the Caching Proxy for transparent proxy as depicted in Figure 3. As with regular forward Caching Proxy, the transparent Caching Proxy is installed on a machine near the gateway, but client browser programs are not configured to direct requests to a forward Caching Proxy. Clients are not aware that a proxy exists in the configuration. Instead, a router is configured to intercept client requests and direct them to the transparent Caching Proxy. When a client working on one of the machines, marked 1, requests file X stored on a content host (6), the router (2) passes the request to the Caching Proxy. Caching Proxy generates a new request with its own IP address as the originating address and sends the new request out by means of the router (2) across the Internet (5). When file X arrives, the Caching Proxy caches the file if appropriate (subject to the conditions described in "Forward Caching Proxy" on page 14) and passes the file to the requesting client.

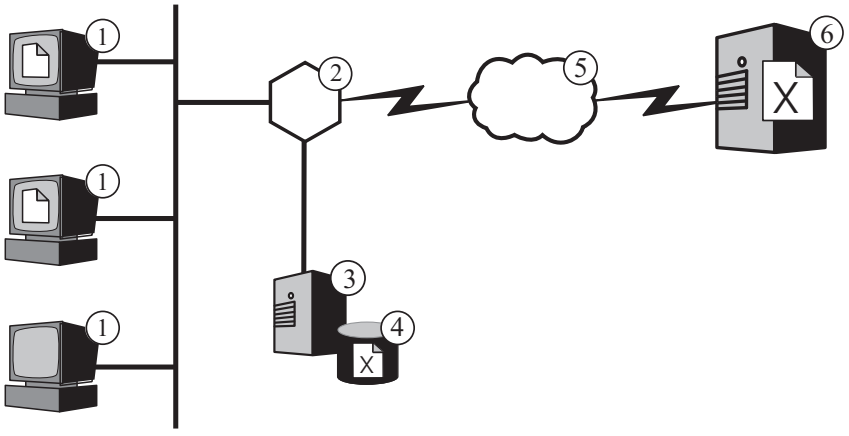

Figure 3. The Caching Proxy acting as a transparent forward proxy. Legend:
1—Client 2—Router/Gateway 3—Caching Proxy 4—Cache 5—Internet 6—Content host

For HTTP requests, another possible alternative to maintaining proxy configuration information on each browser is to use the automatic proxy configuration feature available in several browser programs, including Netscape Navigator version 2.0 and higher and Microsoft Internet Explorer version 4.0 and higher. In this case, you create one or more central proxy automatic configuration (PAC) files and configure browsers to refer to one of them rather than to local proxy configuration information. The browser automatically notices changes to the PAC and adjusts its proxy usage accordingly. This not only eliminates the need to maintain separate configuration information on each browser, but also makes it easy to reroute requests when a proxy server becomes unavailable.

A third alternative is to use the Web Proxy Auto Discovery (WPAD) mechanism available in some browser programs, such as Internet Explorer version 5.0 and higher. When you enable this feature on the browser, it automatically locates a WPAD-compliant proxy server in its network and directs its Web requests there. You do not need to maintain central proxy configuration files in this case. Caching Proxy is WPAD-compliant.

Advanced caching

Load-balanced Caching Proxy clusters

To provide more advanced caching functionality, use Caching Proxy as a reverse proxy in conjunction with the Load Balancer component. By integrating caching and load-balancing capabilities, you can create an efficient, highly manageable Web performance infrastructure.

Figure 4 depicts how you can combine Caching Proxy with Load Balancer to deliver Web content efficiently even in circumstances of high demand. In this configuration, the proxy server (**4**) is configured to intercept requests whose URLs include the host name for a cluster of content hosts (**7**) being load-balanced by Load Balancer (**6**).

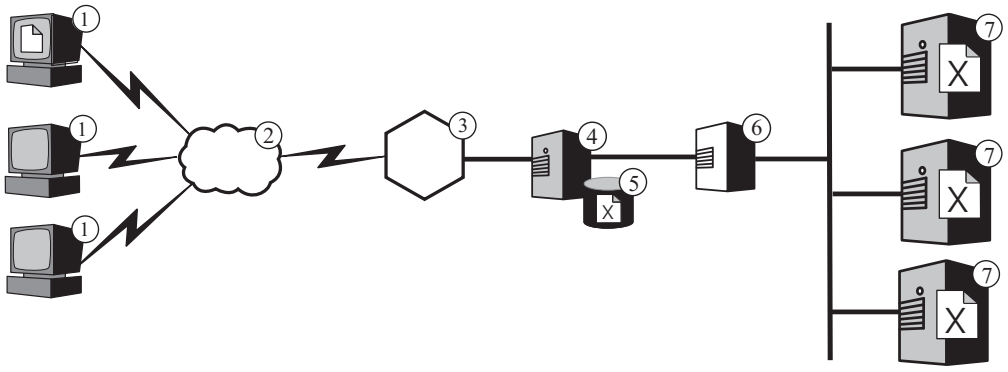

Figure 4. Caching Proxy acting as proxy server for a load-balanced cluster. Legend:
1—Client 2—Internet 3—Router/Gateway 4—Caching Proxy 5—Cache 6—Load Balancer 7—Content host

When a client (**1**) requests file **X**, the request crosses the Internet (**2**) and enters the enterprise's internal network through its Internet gateway (**3**). The proxy server intercepts the request, generates a new request with its own IP address as the originating address, and sends the new request to Load Balancer at the cluster address. Load Balancer uses its load-balancing algorithm to determine which content host is currently best able to satisfy the request for file **X**. That content host

returns file **X** to the proxy server rather than via Load Balancer. The proxy server determines whether to cache it and delivers it to the end user in the same way as described previously.

Caching dynamic content

Advanced caching functionality is also provided by Caching Proxy's Dynamic Caching plug-in. When used in conjunction with WebSphere Application Server, Caching Proxy has the ability to cache, serve, and invalidate dynamic content in the form of JavaServer Pages (JSP) and servlet responses generated by a WebSphere Application Server.

Generally, dynamic content with an indefinite expiration time must be marked "do not cache" because the standard time-based cache expiration logic does not ensure its timely removal. The Dynamic Caching plug-in's event-driven expiration logic enables content with an indefinite expiration time to be cached by the proxy server. Caching such content at the edge of the network relieves content hosts from repeatedly invoking an Application Server to satisfy requests from clients. This can offer the following benefits:

- Reduced workload on Web servers, WebSphere Application Servers, and back-end content hosts
- Faster response to users by eliminating network delays
- Reduced bandwidth usage due to fewer Internet traversals
- Better scalability of Web sites that serve dynamically generated content

Servlet response caching is ideal for dynamically produced Web pages that expire based on application logic or an event such as a message from a database. Although such a page's lifetime is finite, the time-to-live value cannot be set at the time of creation because the expiration trigger cannot be known in advance. When the time-to-live for such pages is set to zero, content hosts incur a high penalty when serving dynamic content.

The responsibility for synchronizing the dynamic cache of Caching Proxy and Application Server is shared by both systems. For example, a public Web page dynamically created by an application that gives the current weather forecast can be exported by Application Server and cached by Caching Proxy. Caching Proxy can then serve the application's execution results repeatedly to many different users until notified that the page is invalid. Content in Caching Proxy's servlet response cache is valid until the proxy server removes an entry because the cache is congested, the default timeout set by the ExternalCacheManager directive in Caching Proxy's configuration file expires, or Caching Proxy receives an Invalidate message directing it to purge the content from its cache. Invalidate messages originate at the WebSphere Application Server that owns the content and are propagated to each configured Caching Proxy.

Note: Dynamically generated private pages (such as a page showing the contents of a user's shopping cart) generally cannot and should not be cached by Caching Proxy. Caching Proxy can cache and serve private pages only when it is configured to perform authentication and authorization to ensure that the private pages are served only to their intended users.

Additional caching features

Caching Proxy offers other key advanced caching features:

- The ability to use very large caches

- An option to automatically refresh the cache with the most-frequently accessed pages
- The possibility to cache even those pages where the header information says to fetch them every time
- Configurable daily garbage collection to improve server performance and ensure cache maintenance
- Remote Cache Access (RCA), a function that allows multiple Caching Proxy machines to share the same cache, thereby reducing the redundancy of cached content
- The ICP plug-in, which enables Caching Proxy to query Internet Caching Protocol (ICP)-compliant caches in search of HTML pages and other cacheable resources

Chapter 5. Network performance

Network performance is affected by the introduction of Caching Proxy functionality. Use Caching Proxy alone or in conjunction with Load Balancer to improve the performance of your network. See Chapter 1, "Introducing WebSphere Application Server Edge components," on page 3 for an introduction to these systems.

Caching Proxy's performance within your enterprise is only as good as the hardware on which it runs and the overall architecture of the system into which it is introduced. To optimize network performance, model hardware and overall network architecture to the characteristics of proxy servers.

Basic configuration and administration of the Caching Proxy software and tuning at the operating system level also greatly contribute to Caching Proxy performance. Many software configuration changes can be made to yield increased performance; these include, but are not limited to, adjusting the logging directives, mapping rules, plug-ins, timeout values, cache configuration values, and active thread values. Details on configuring Caching Proxy software is presented in the *Caching Proxy Administration Guide*.

Many operating system configuration changes can be made to yield increased performance as well; these include, but are not limited to, tuning TCP and ARP, increasing the file descriptor limits, synchronizing system clocks, tuning Network Interface Cards (NICs), and following common good practices when performing system administration tasks.

IMPORTANT: Caching Proxy is available on all Edge component installations, with the following exceptions:

- Caching Proxy is not available for Edge component installations that run on Itanium 2 or AMD Opteron 64-bit processors.
- Caching Proxy is not available for Edge component installations of Load Balancer for IPv4 and IPv6.

Network hardware

This section discusses network hardware issues to consider when introducing Caching Proxy functionality into your network.

Memory considerations

A large amount of memory must be dedicated to the proxy server. Caching Proxy can consume 2 GB of virtual address space when a large memory-only cache is configured. Memory is also needed for the kernel, shared libraries, and network buffers. Therefore, it is possible to have a proxy server that consumes 3 or 4 GB of physical memory. Note that a memory-only cache is significantly faster than a raw disk cache, and this configuration change alone can be considered a performance improvement.

Hard disk considerations

It is important to have a large amount of disk space on the machine on which Caching Proxy is installed. This is especially true when disk caches are used.

Reading and writing to a hard disk is an intensive process for a computer. Although Caching Proxy's I/O procedures are efficient, the mechanical limitations of hard drives can limit performance when the Caching Proxy is configured to use a disk cache. The disk I/O bottleneck can be alleviated with practices such as using multiple hard disks for raw cache devices and log files and by using disk drives with fast seek times, rotational speeds, and transfer rates.

Network considerations

Network requirements such as the speed, type, and number of NICs, and the speed of the network connection to the proxy server affect the performance of Caching Proxy. It is generally in the best interest of performance to use two NICs on a proxy server machine: one for incoming traffic and one for outgoing traffic. It is likely that a single NIC can reach its maximum limit by HTTP request and response traffic alone. Furthermore, NICs should be at least 100 MB, and they should always be configured for full-duplex operation. This is because automatic negotiation between routing and switching equipment can possibly cause errors and hinder throughput. Finally, the speed of the network connection is very important. For example, you cannot expect to service a high request load and achieve optimal throughput if the connection to the Caching Proxy machine is a saturated T1 carrier.

CPU considerations

The central processing unit (CPU) of a Caching Proxy machine can possibly become a limiting factor. CPU power affects the amount of time it takes to process requests and the number of CPUs in the network affects scalability. It is important to match the CPU requirements of the proxy server to the environment, especially to model the peak request load that the proxy server will service.

Network architecture

For overall performance, it is generally beneficial to scale the architecture and not just add individual pieces of hardware. No matter how much hardware you add to a single machine, that hardware still has a maximum level of performance.

The section discusses network architecture issues to take into consideration when introducing Caching Proxy functionality into your network.

Web site popularity and proxy server load considerations

If your enterprise's Web site is popular, there can be greater demand for its content than a single proxy server can satisfy effectively, resulting in slow response times. To optimize network performance, consider including clustered, load-balanced Caching Proxy machines or using a shared cache architecture with Remote Cache Access (RCA) in your overall network architecture.

- **Load-balanced clusters**

  One way to scale the architecture is to cluster proxy servers and use the Load Balancer component to balance the load among them. Clustering proxy servers is a beneficial design consideration not only for performance and scalability reasons but for redundancy and reliability reasons as well. A single proxy server represents a single point of failure; if it fails or becomes inaccessible because of a network failure, users cannot access your Web site.

- **Shared cache architecture**

  Also consider a shared cache architecture with RCA. A shared cache architecture spreads the total virtual cache among multiple Caching Proxy servers that

usually use an intercache protocol like the Internet Cache Protocol (ICP) or the Cache Array Routing Protocol (CARP). RCA is designed to maximize clustered cache hit ratios by providing a large virtual cache.

Performance benefits result from using an RCA array of proxy servers as opposed to a single stand-alone Caching Proxy or even a cluster of stand alone Caching Proxy machines. For the most part, the performance benefits are caused by the increase in the total virtual cache size, which maximizes the cache hit ratio and minimizes cache inconsistency and latency. With RCA, only one copy of a particular document resides in the cache. With a cluster of proxy servers, the total cache size is increased, but multiple proxy servers are likely to fetch and cache the same information. The total cache hit ratio is therefore not increased.

RCA is commonly used in large enterprise content-hosting scenarios. However, RCA's usefulness is not limited to extremely large enterprise deployments. Consider using RCA if your network's load requires a cluster of cache servers and if the majority of requests are cache hits. Depending on your network setup, RCA does not always improve enterprise performance due to an increase in the number of TCP connections that a client uses when RCA is configured. This is because an RCA member is not only responsible for servicing URLs for which it has the highest score but it must also forward requests to other members or clusters if it gets a request for a URL for which it does not have the highest score. This means that any given member of an RCA array might have more open TCP connections than it would if it operated as a stand-alone server.

Traffic type considerations

Major contributions to improved performance stem from Caching Proxy's caching capabilities. However, the cache of the proxy server can become a bottleneck if it is not properly configured. To determine the best cache configuration, a significant effort must be made to analyze traffic characteristics. The type, size, amount, and attributes of the content affect the performance of the proxy server in terms of the time it takes to retrieve documents from origin servers and the load on the server. When you understand the type of traffic that Caching Proxy is going to proxy or serve from its cache, then you can factor in those characteristics when configuring the proxy server. For example, knowing that 80% of the objects being cached are images (*.gif or *.jpg) and are approximately 200 KB in size can certainly help you tune caching parameters and determine the size of the cache. Additionally, understanding that most of the content is personalized dynamic pages that are not candidates for caching is also pertinent to tuning Caching Proxy.

Analyzing traffic characteristics enables you to determine whether using a memory or disk cache can optimize your cache's performance. Also, familiarity with your network's traffic characteristics enables you to determine whether improved performance can result from using the Caching Proxy's dynamic caching feature.

- **Memory versus disk cache**

  Disk caches are appropriate for sites with large amounts of information to be cached. For example, if the site content is large (greater than 5 GB) and there is an 80 to 90% cache hit rate, then a disk cache is recommended. However, it is known that using a memory (RAM) cache is faster, and there are many scenarios when using a memory-only cache is feasible for large sites. For example, if Caching Proxy's cache hit rate is not as important or if a shared cache configuration is being used, then a memory cache is practical.

- **Caching dynamically generated content**

  Caching Proxy can cache and invalidate dynamic content (JSP and servlet results) generated by the WebSphere Application Server dynamic cache,

providing a virtual extension of the Application Server cache into network-based caches. Enabling the caching of dynamically generated content is beneficial to network performance in an environment where there are many requests for dynamically produced public Web pages that expire based on application logic or an event such as a message from a database. The page's lifetime is finite, but an expiration trigger cannot be set in at the time of its creation; therefore, hosts without a dynamic caching and invalidation feature must designate such as page as having a time-to-live value of zero.

If such a dynamically generated page will be requested more than once during its lifetime by one or more users, then dynamic caching provides a valuable offload and reduces the workload on your network's content hosts. Using dynamic caching also improves network performance by providing faster response to users by eliminating network delays and reducing bandwidth usage due to fewer Internet traversals.

Chapter 6. Availability

Functioning in conjunction with content hosts, such as WebSphere Application Server, or with the Application Server Caching Proxy component, the Application Server Load Balancer component enables you to enhance your network's availability and scalability. (See Chapter 1, "Introducing WebSphere Application Server Edge components," on page 3 for an introduction to these Edge components.) Load Balancer is used by enterprise networks and is installed between the Internet and the enterprise's back-end servers. Load Balancer acts as the enterprise's single point-of-presence on the Internet, even if the enterprise uses multiple back-end servers because of high demand or a large amount of content.

Availability is achieved through load balancing and failover support.

IMPORTANT: Caching Proxy is available on all Edge component installations, with the following exceptions:

- Caching Proxy is not available for Edge component installations that run on Itanium 2 or AMD Opteron 64-bit processors.
- Caching Proxy is not available for Edge component installations of Load Balancer for IPv4 and IPv6.

Load balancing

Load balancing improves your Web site's availability and scalability by transparently clustering proxy servers and application servers. An IT infrastructure's scalability is greatly improved because back-end processing power can be added transparently.

Load balancing multiple content hosts

You can satisfy high demand by duplicating content on multiple hosts, but then you need a way to balance the load among them. Domain Name Service (DNS) can provide basic round-robin load balancing, but there are several situations in which it does not perform well.

A more sophisticated solution for load balancing multiple content hosts is to use Load Balancer's Dispatcher component as depicted in Figure 5 on page 26. In this configuration, all of the content hosts (the machines marked **5**) store the same content. They are defined to form a load-balanced *cluster*, and one of the network interfaces of the Load Balancer machine (**4**) is assigned a host name and IP address dedicated to the cluster. When an end user working on one of the machines marked **1** requests file **X**, the request crosses the Internet (**2**) and enters the enterprise's internal network through its Internet gateway (**3**). The Dispatcher intercepts the request because its URL is mapped to the Dispatcher's host name and IP address. The Dispatcher determines which of the content hosts in the cluster is currently best able to service the request, and forwards the request to that host, which, when the MAC forwarding method is configured, returns file **X** directly to the client (that is, file **X** does not pass through Load Balancer).

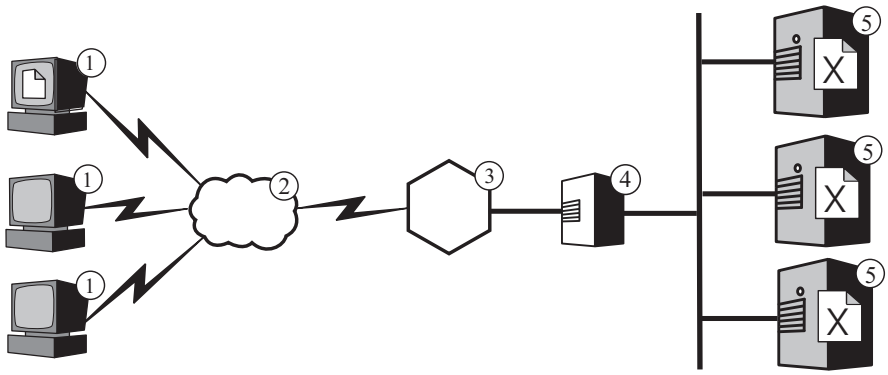

Figure 5. Load balancing multiple content hosts. Legend:
1—Client 2—Internet 3—Router/Gateway 4—Dispatcher 5—Content host

Note: The Dispatcher provides three forwarding methods:

- The MAC forwarding method is used to load balance incoming requests to the server. Responses are returned directly to the client.
- The NAT/NAPT forwarding method is used with servers located remotely. Incoming requests are load balanced by Dispatcher, which receives the responses from the server and returns them to the clients. (On Load Balancer for IPv4 and IPv6 installations, this forwarding method is not supported.)
- The content-based routing method (cbr) provides content-based routing of HTTP and HTTPS requests without the use of Caching Proxy. Content-based routing is performed for HTTP by using the ″content″ type rule and for HTTPS by using SSL session ID affinity. (On Load Balancer for IPv4 and IPv6 installations, this forwarding method is not supported.)

By default, the Dispatcher uses round-robin load balancing like DNS, but even so it addresses many of DNS's inadequacies. Unlike DNS, it tracks whether a content host is unavailable or inaccessible and does not continue to direct clients to an unavailable content host. Further, it considers the current load on the content hosts by tracking new, active, and finished connections. You can further optimize load balancing by activating Load Balancer's optional advisor and manager components, which track a content host's status even more accurately and incorporate the additional information into the load-balancing decision process. The manager enables you to assign different weights to the different factors used in the decision process, further customizing load balancing for your site.

Load balancing multiple reverse proxy servers

Load Balancer's Dispatcher can also perform load balancing for multiple Caching Proxy machines. If your enterprise's Web site is popular, there can be greater demand for its contents than a single proxy server can satisfy effectively, potentially degrading the proxy server's performance.

You can have multiple Caching Proxy systems performing proxy functions for a single content host (similar to the configuration depicted in Figure 1 on page 14), but if your site is popular enough to need multiple proxy servers, then you probably also need multiple contents hosts whose loads are balanced by Load Balancer. Figure 6 on page 27 depicts this configuration. The Dispatcher marked **4** load balances a cluster of two proxy servers (**5**), and the Dispatcher marked **7** load balances a cluster of three content hosts (**8**).

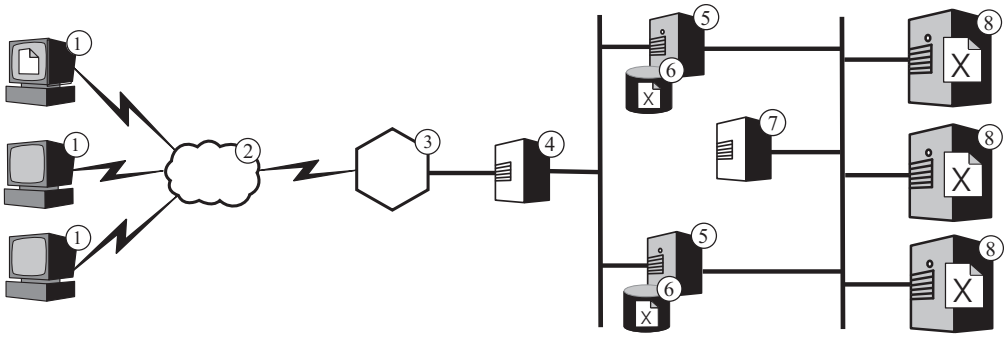

Figure 6. Load balancing multiple reverse proxy servers and content hosts.
1—Client 2—Internet 3—Router/Gateway 4—Dispatcher 5—Caching Proxy 6—Cache 7—Dispatcher 8—Content host

The cluster host name of the Dispatcher marked **4** is the host name that appears in URLs for the enterprise's Web content (that is, it is the name of the Web site as visible on the Internet). The cluster host name for the Dispatcher marked **7** is not visible on the Internet and so can be any value you wish. As an example, for the ABC Corporation an appropriate host name for the Dispatcher marked **4** is www.abc.com, whereas the Dispatcher marked **7** can be called something like http-balancer.abc.com .

Suppose that a browser on one of the client machines marked **1** needs to access file **X** stored on the content servers marked **8**. The HTTP request crosses the Internet (**2**) and enters the enterprise's internal network at the gateway (**3**). The router directs the request to the Dispatcher marked **4**, which passes it to the proxy server (**5**), which is currently best able to handle it according to the load-balancing algorithm. If the proxy server has file **X** in its cache (**6**), it returns it directly to the browser, bypassing the Dispatcher marked **4**.

If the proxy server does not have a copy of file **X** in its cache, it creates a new request that has its own host name in the header's origin field and sends that to the Dispatcher marked **7**. The Load Balancer determines which content host (**8**) is currently best able to satisfy the request, and directs the request there. The content host retrieves file **X** from storage and returns it directly to the proxy server, bypassing the Dispatcher marked **7**. The proxy server caches file **X** if appropriate, and forwards it to the browser, bypassing the Dispatcher marked **4**.

Load Balancer with multiple forward proxy servers

If you provide Internet access to a large number of clients, they can generate more demand for Internet access than a single proxy can provide efficiently. As the Caching Proxy becomes overloaded with requests, clients can possibly experience worse response times than with direct Internet access. And if the Caching Proxy fails or becomes inaccessible because of a network failure, Internet access becomes impossible. The solution is to install multiple Caching Proxy machines and use the Load Balancer's Dispatcher to balance the load between them.

Without the Dispatcher, you can provide true transparent proxy with multiple Caching Proxy machines only if your routers can route the same type of traffic to more than one Caching Proxy; not all routers support this. It is possible to provide regular forward proxy service on multiple Caching Proxy machines without the Dispatcher, but you must explicitly configure the client browsers to use one of the Caching Proxy machines as their primary proxy. If that Caching Proxy fails or

becomes overloaded or inaccessible, end users can become unable to access the Internet. To avoid that situation, you can create a proxy automatic configuration (PAC) file (as described in "Transparent forward Caching Proxy (Linux systems only)" on page 16) that directs the browser to fail over to one or more secondary caching proxies. A PAC file does not address the need to balance the load among the Caching Proxy machines; however, if one Caching Proxy receives many more requests than another, its performance is likely to degrade, subjecting its browser clients to slower response times. For all clients to experience similar performance, you must configure an approximately equal number of browsers to use each Caching Proxy, and track the distribution manually so that you can keep the load even as you add or remove browsers.

Figure 7 depicts a network configuration in which the Dispatcher load balances a cluster of Caching Proxy machines. One of the Dispatcher machine's network interfaces is configured to have the cluster's dedicated host name and IP address. Client browsers are configured to direct their Internet requests to the cluster host name. When, for example, a browser on one of the client machines marked 1 needs to access file X on a content host (7), it directs its request to the cluster host name or address, where the Dispatcher (2) intercepts it and directs it to the appropriate Caching Proxy (3). The Caching Proxy creates a new request, passes it through the enterprise's gateway (5) and across the Internet (6), and if appropriate stores the returned file in its cache (4) as described in more detail in "Forward Caching Proxy" on page 14

Note: The transparent proxy feature is available on Linux systems only.

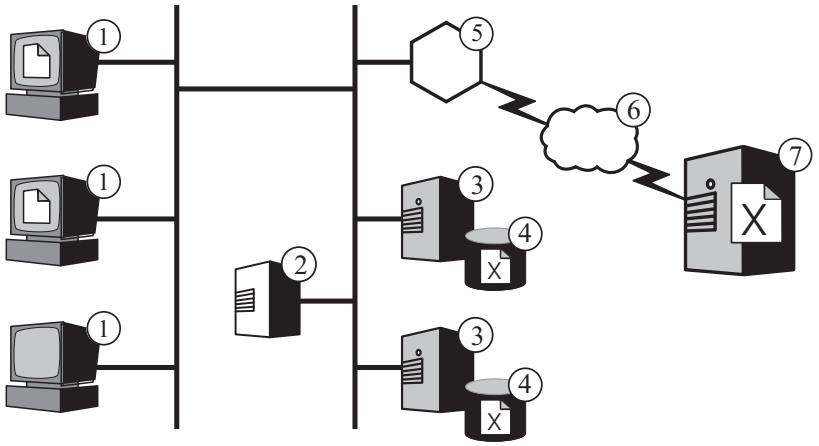

Figure 7. Using Dispatcher to load balance multiple caching proxies.. Legend: 1—Client 2—Dispatcher 3—Caching Proxy 4—Cache 5—Router/ Gateway 6—Internet 7—Content host

The Dispatcher detects when one of the Caching Proxy machines is unavailable and automatically routes requests to the other one. This allows you to shut down a Caching Proxy machine for maintenance without interrupting Internet access. The Dispatcher has numerous configuration options that you can use to control the factors it considers when making load-balancing decisions. You can also install auxiliary Dispatcher programs on the Caching Proxy machines to monitor their status and return the information to the Dispatcher. For details, see the *WebSphere Application Server Load Balancer Administration Guide*. Using multiple Caching Proxy's introduces a potential inefficiency, because more than one Caching Proxy can cache the same file if different clients request the file via different Caching Proxy machines. To eliminate the redundancy, you can configure remote cache access (RCA), which enables all of the proxies in a defined group to share the

contents of their caches with one another. The proxies in the RCA group all use the same algorithm to determine which Caching Proxy is responsible for a given URL. When a Caching Proxy intercepts a URL for which it is not responsible, it passes the request to the responsible Caching Proxy. The responsible Caching Proxy does the work necessary to satisfy the request, either retrieving it from its cache or forwarding the request to the relevant content host and caching the returned file if appropriate. The responsible Caching Proxy then passes the file to the original Caching Proxy, which delivers it to the requesting end user.

In the RCA group, if the Caching Proxy responsible for a given URL is failed, then the original Caching Proxy, which received the client request, will directly access the content host (or a backup Caching Proxy server, if it is defined). This implies that users can access files as long as at least one Caching Proxy in an RCA group is functioning correctly.

This configuration satisfies high demand for Internet access by using the Dispatcher to balance the load of requests across multiple Caching Proxy machines. One potential problem is that the Dispatcher is a single point of failure. If it fails or becomes inaccessible due to a network failure, browser clients cannot reach the Caching Proxy's or the Internet. The solution is to configure another Dispatcher to act as a backup for the primary Dispatcher, as depicted in Figure 8.

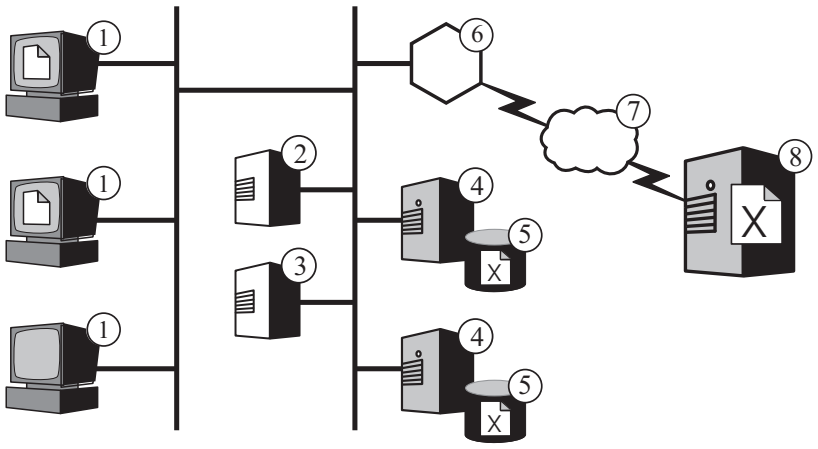

Figure 8. Using a primary and backup Dispatcher to provide highly available Internet access..
Legend: 1—Client 2—Primary Dispatcher 3—Backup Dispatcher 4—Caching Proxy 5—Cache 6—Router/Gateway 7—Internet 8—Content host

Here a browser running on one of the machines marked 1 normally directs its request for a file X to the primary Dispatcher (2), which routes the request to the Caching Proxy (4) selected on the basis of the Dispatcher's load-balancing criteria. The Caching Proxy creates a new request, routes it through the enterprise's gateway (6) across the Internet (7) to the content host (8), and stores the returned file X in its cache (5) if appropriate (for a more detailed description of this part of the process, see "Forward Caching Proxy" on page 14).

In this configuration, the backup Dispatcher (3) does not perform load balancing as long as the primary is operational. The primary and backup Dispatchers track each other's status by periodically exchanging messages called heartbeats. If the backup Dispatcher detects that the primary has failed, it automatically takes over the responsibility for load balancing by intercepting requests directed to the primary's host name and IP address. It is also possible to configure two Dispatchers for mutual high availability. In this case each actively performs load balancing for a

separate cluster of caching proxies, simultaneously acting as the backup for its colleague. For further discussion, see the *WebSphere Application Server Load Balancer Administration Guide*.

The Dispatcher does not generally consume many processing or memory resources, and other applications can run on the Dispatcher machine. If it is important to minimize equipment costs, it is even possible to run the backup Dispatcher on the same machine as Caching Proxy. Figure 9 depicts such a configuration, in which the backup Dispatcher runs on the same machine (3) as the Caching Proxy.

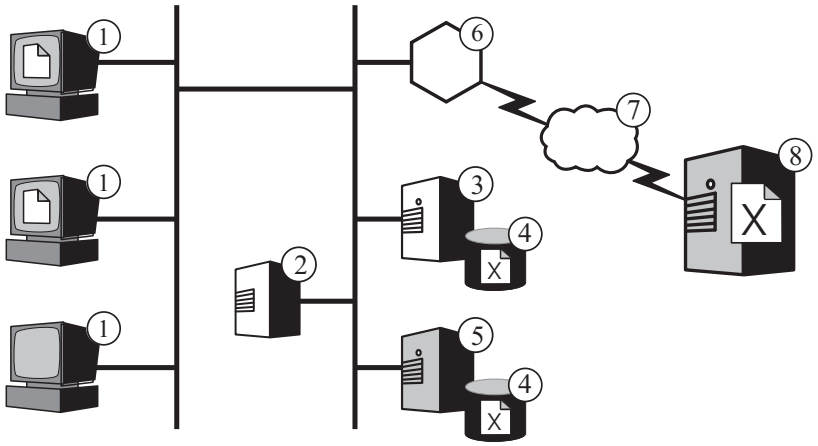

Figure 9. Locating the backup Dispatcher on a Caching Proxy machine.. Legend: 1—Client 2—Primary Dispatcher 3—Backup Dispatcher and Caching Proxy 4—Caching Proxy 5—Cache 6—Router/ Gateway 7—Internet 8—Content host

Failover support

Load Balancer acts as a single point-of-presence for your enterprise's content hosts. This is beneficial because you advertise the cluster host name and address in DNS, rather than the host name and address of each content host, which provides a level of protection against casual attacks and provides a unified feel for your enterprise's Web site. To further enhance Web site availability, configure another Load Balancer to act as a backup for the primary Load Balancer, as depicted in Figure 10 on page 31. If one Load Balancer fails or becomes inaccessible due to a network failure, end users can still reach the content hosts.

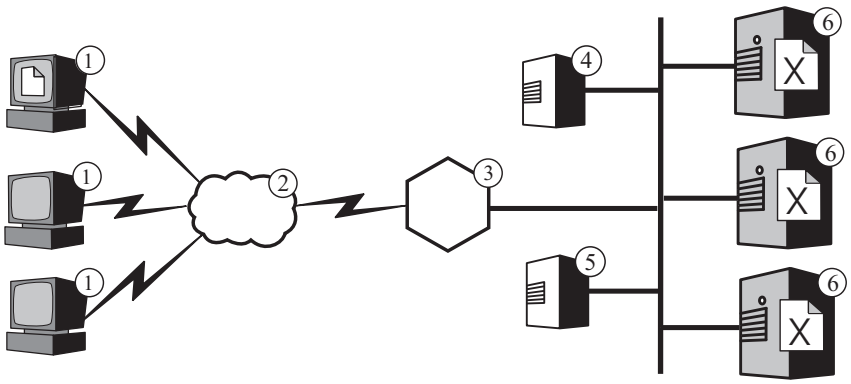

Figure 10. Using a primary and backup Load Balancer to make Web content highly available. Legend: 1—Client 2—Internet 3—Router/Gateway 4—Primary Dispatcher 5—Backup Dispatcher 6—Content host

In the normal case, a browser running on one of the machines marked **1** directs its request for a file **X** to the cluster host name that is mapped to the primary Load Balancer (**4**). The Dispatcher routes the request to the content host (**6**) selected on the basis of the Dispatcher's load-balancing criteria. The content host sends file **X** directly to the browser, routing it through the enterprise's gateway (**3**) across the Internet (**2**) but bypassing Load Balancer.

The backup Dispatcher (**5**) does not perform load balancing as long as the primary one is operational. The primary and backup Dispatchers track each other's status by periodically exchanging messages called *heartbeats*. If the backup Dispatcher detects that the primary has failed, it automatically takes over the responsibility for load balancing by intercepting requests directed to the primary's cluster host name and IP address.

It is also possible to configure two Dispatchers for *mutual high availability*. In this case, each actively performs load balancing for a separate cluster of content hosts, simultaneously acting as the backup for its colleague. (On Load Balancer for IPv4 and IPv6 installations simple high availability is supported, but mutual high availability is not.)

The Dispatcher does not generally consume many processing or memory resources, and other applications can run on the Load Balancer machine. If it is vital to minimize equipment costs, it is even possible to run the backup Dispatcher on one of the machines in the cluster it is load balancing. Figure 11 on page 32 depicts such a configuration, in which the backup Dispatcher runs on one of the content hosts (**5**) in the cluster.

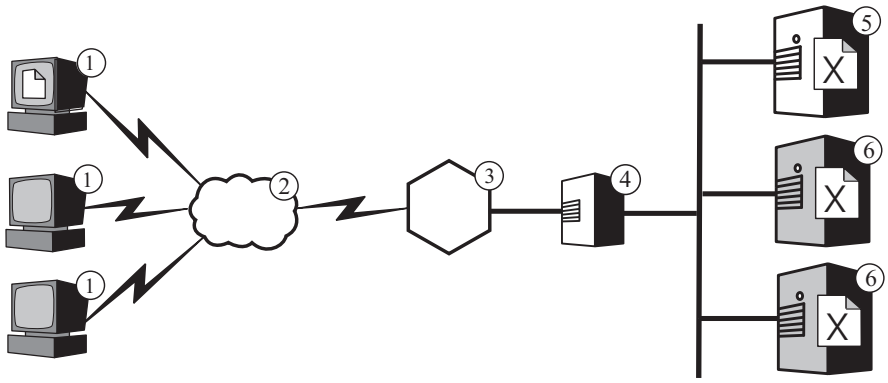

Figure 11. Locating the backup Load Balancer on a content host. Legend: 1—Client 2—Internet 3—Router/Gateway 4—Primary Dispatcher 5—Backup Dispatcher and content host 6—Content host

Chapter 7. Content Based Routing

IMPORTANT: The Content Based Routing (CBR) component is available on all supported platforms, with the following exceptions:

- CBR is not available with Load Balancer installations on platforms running a 64-bit JVM.

  Alternatively, for this type of installation, you can use the cbr forwarding method of Load Balancer's Dispatcher component to provide content-based routing of HTTP and HTTPS requests without the use of Caching Proxy. See the *WebSphere Application Server Load Balancer Administration Guide* for more information.
- CBR is not available with Load Balancer for IPv4 and IPv6 installations.

  Load Balancer for IPv4 and IPv6 supports only the Dispatcher component's mac forwarding method. The nat and cbr forwarding methods are not supported.

Functioning in conjunction with the Application Server Caching Proxy component, the Application Server Load Balancer component enables you to distribute requests to multiple back-end servers that host different content. (See Chapter 1, "Introducing WebSphere Application Server Edge components," on page 3 for an introduction to these Edge components.)

If Load Balancer's Content Based Routing (CBR) component is installed together with Caching Proxy, HTTP requests can be distributed based on URL or other administrator-determined characteristics, eliminating the need to store identical content on all back-end servers.

Using CBR is especially appropriate if your Web servers need to perform several different functions or offer several types of services. For example, an online retailer's Web site must both display its catalog, a large of portion of which is static, and accept orders, which means running an interactive application such as a Common Gateway Interface (CGI) script to accept item numbers and customer information. Often it is more efficient to have two different sets of machines perform the distinct functions, and to use CBR to route the different types of traffic to different machines. Similarly, an enterprise can use CBR to provide better service to paying customers than to casual visitors to its Web site, by routing the paid requests to more powerful Web servers.

CBR routes requests based on rules that you write. The most common type is the *content rule*, which directs requests based on the path name in the URL. For example, the ABC Corporation can write rules that direct requests for the URL http://www.abc.com/catalog_index.html to one cluster of servers and http://www.abc.com/orders.html to another cluster. There are also rules that route requests based on the IP address of the client who sent them or on other characteristics. For a discussion, see the *WebSphere Application Server Load Balancer Administration Guide* chapters about configuring CBR and about advanced Load Balancer and CBR functions. For syntax definitions for the rules, see the *WebSphere Application Server Load Balancer Administration Guide* appendix about CBR rule types.

Figure 12 on page 34 depicts a simple configuration in which Load Balancer's CBR component and Caching Proxy are installed together on the machine marked **4** and route requests to three content hosts (**6**, **7**, and **8**) that house different content.

When an end user working on one of the machines marked **1** requests file **X**, the request crosses the Internet (**2**) and enters the enterprise's internal network through its Internet gateway (**3**). The proxy server intercepts the request and passes it to the CBR component on the same machine, which parses the URL in the request and determines that content host **6** houses file **X**. The proxy server generates a new request for file **X**, and if its caching feature is enabled, determines whether the file is eligible for caching when host **6** returns it. If the file is cacheable, the proxy server stores a copy in its cache (**5**) before passing it to the end user. Routing for other files works in the same manner: requests for file **Y** go to content host **7**, and requests for file **Z** go to content host **8**.

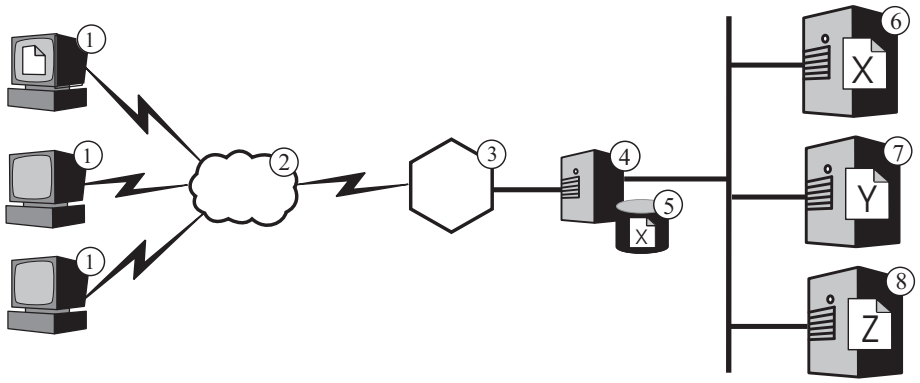

Figure 12. Routing HTTP requests with CBR. Legend: 1—Client 2—Internet 3—Router/ Gateway 4—Caching Proxy and Load Balancer's CBR component 5—Cache 6, 7, 8—Content host

Figure 13 on page 35 depicts a more complex configuration which is possibly suitable for an online retailer. Load Balancer's CBR component and the proxy server are installed together on the machine marked **4** and route requests to two Load Balancer machines. The Load Balancer machine marked **6** load balances a cluster of content hosts (**8**) that house the mostly static content of the retailer's catalog, whereas the Load Balancer marked **7** load balances a cluster of Web servers that handle orders (**9**).

When an end user working on one of the machines marked **1** accesses the URL for the retailer's catalog, the request crosses the Internet (**2**) and enters the enterprise's internal network through its Internet gateway (**3**). The proxy server intercepts the request and passes it to the CBR component on the same machine, which parses the URL and determines that the Load Balancer machine marked **6** handles that URL. The proxy server creates a new access request and sends it to the Load Balancer, which determines which of the content hosts marked **8** is currently best able to service the request (based on criteria that you define). That content host passes the catalog content directly to the proxy server, bypassing Load Balancer. As in the preceding example, the proxy server determines whether the content is cacheable and stores it in its cache (**5**) if appropriate.

The end user places an order by accessing the retailer's ordering URL, presumably via a hyperlink in the catalog. The request travels the same path as the catalog access request, except that the CBR component on machine **4** routes it to the Load Balancer machine marked **7**. Load Balancer forwards it to the most suitable of the Web servers marked **9**, which replies directly to the proxy server. Because ordering information is generally dynamically generated, the proxy server probably does not cache it.

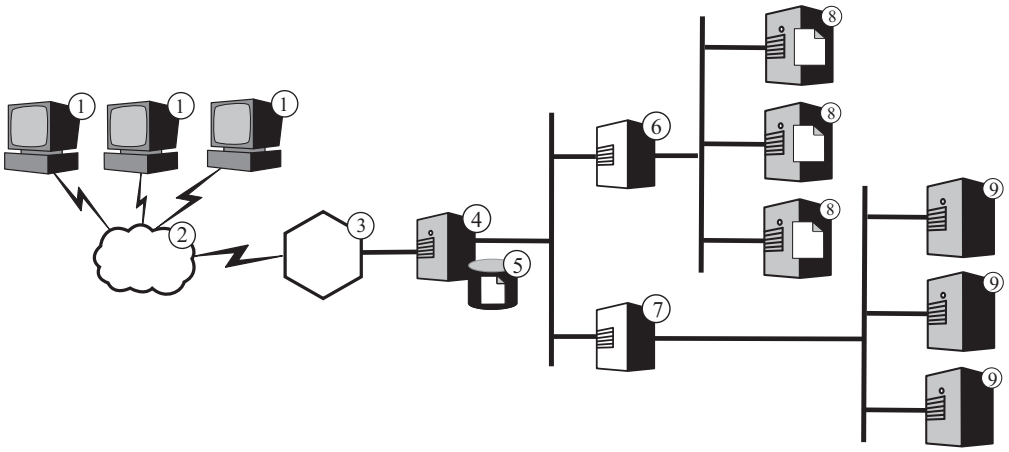

Figure 13. Load balancing HTTP requests routed with CBR. Legend:
1—Client 2—Internet 3—Router/Gateway 4—Caching Proxy and Load Balancer's CBR component 5—Cache 6, 7—Load Balancer 8—Content host 9—Web Server

Load Balancer's CBR function supports *cookie affinity*. This means that the identity of the server that serviced an end user's first request is recorded in a special packet of data (a *cookie*) included in the server's response. When the end user accesses the same URL again within a period of time that you define, and the request includes the cookie, CBR routes the request to the original server rather than reapplying its standard rules. This generally improves response time if the server has stored information about the end user that it does not have to obtain again (such as a credit card number).

Part 3. Scenarios

This part discusses business scenarios that use IBM WebSphere Application Server Edge components. These are architecturally sound and tested solutions that can provide excellent performance, availability, scalability, and reliability.

This part contains the following chapters:

Chapter 8. Business-to-consumer network

The basic electronic commerce Web site is a business-to-consumer network. In the first phase of Internet growth, businesses typically focus on simply creating a Web presence. Corporate information and product catalogs are converted to digital formats and made available on the Web site. Shopping can be available by providing e-mail addresses, telephone and fax numbers, and even automated forms. True online shopping, however, is not available. All transactions have an inherent latency because humans need to process the order.

In phase two, businesses eliminate this latency and streamline their sales operation by implementing secure shopping carts for direct online purchases. Synchronization with warehouse databases and integration with banking systems are crucial to completing these sales transactions. Product that is not available cannot be sold, and a customer's account cannot be charged for that item. Likewise, a product cannot be taken from inventory and shipped to a customer until a valid financial transaction occurs.

In the third phase, the corporate Web site evolves into a dynamic presentation site where the consumer begins to take on the aspects of a client and is provided with personalized content.

The following scenerio includes both Load Balancer and Caching Proxy.

IMPORTANT: Caching Proxy is available on all Edge component installations, with the following exceptions:

- Caching Proxy is not available for Edge component installations that run on Itanium 2 or AMD Opteron 64-bit processors.
- Caching Proxy is not available for Edge component installations of Load Balancer for IPv4 and IPv6.

Phase 1

Figure 14 on page 40 shows a small commercial Web site designed to provide efficient catalog browsing. All client requests pass through the firewall to a Dispatcher that routes the requests to a cluster of proxy servers with active caches that act as surrogate servers to the Web servers. Metric servers are colocated with the proxy servers to provide load-balancing data to the Dispatcher. This arrangement reduces the network load on the Web servers and isolates them from direct contact with the Internet.

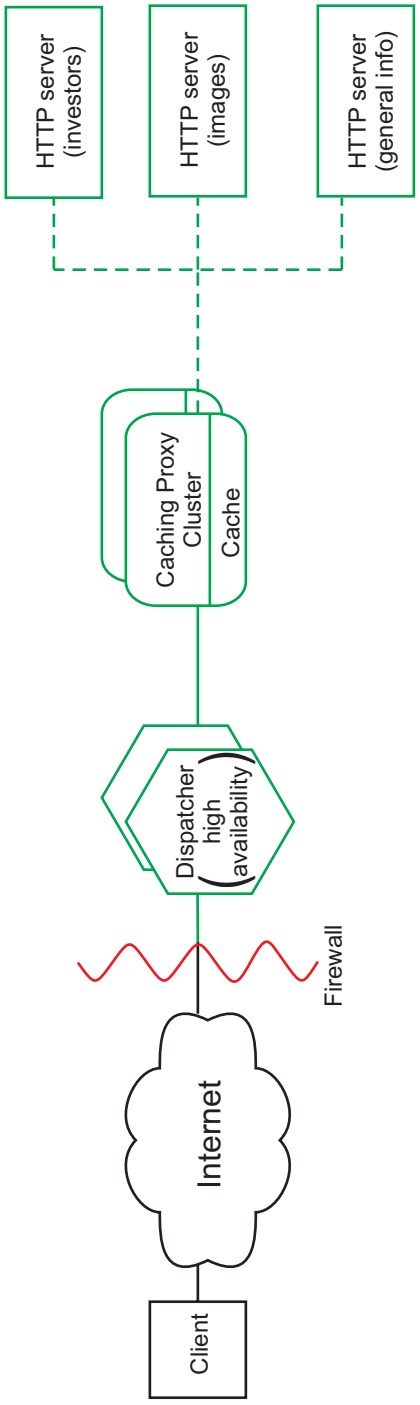

Figure 14. Business to consumer network (Phase 1)

Phase 2

Figure 15 on page 41 shows a the second phase of evolution for a commercial Web site designed to provide efficient catalog browsing and fast, secure shopping carts for potential customers. All customer requests are routed to the appropriate branch of the network by a Dispatcher that separates requests based on Internet protocol. HTTP requests go to the static Web site; HTTPS requests go to the shopping

network. The primary, static Web site is still served by a cluster of proxy servers with active caches that acts as a surrogate for the Web servers. This part of the network mirrors the network in the first phase.

The electronic commerce portion of the Web site is also served by a cluster of proxy servers. However, the Caching Proxy nodes are enhanced with several plug-in modules. The SSL handshaking is offloaded to a cryptographic hardware card, and authentication is performed through the Access Manager (formerly Policy Director) plug-in. A Dynamic Caching plug-in reduces the workload on the WebSphere Application Server by storing common data. A plug-in on the application server invalidates objects in the Dynacache when necessary.

All shopping cart applications are tied into the customer database that was used to authenticate the user. This prevents the user from having to enter personal information into the system twice, once for authentication and once for shopping.

This network divides traffic according to client usage, removing the processor-intensive SSL authentication and electronic commerce shopping carts from the primary Web site. This dual-track Web site allows the network administrator to tune the various servers to provide excellent performance based on the role of the server within the network.

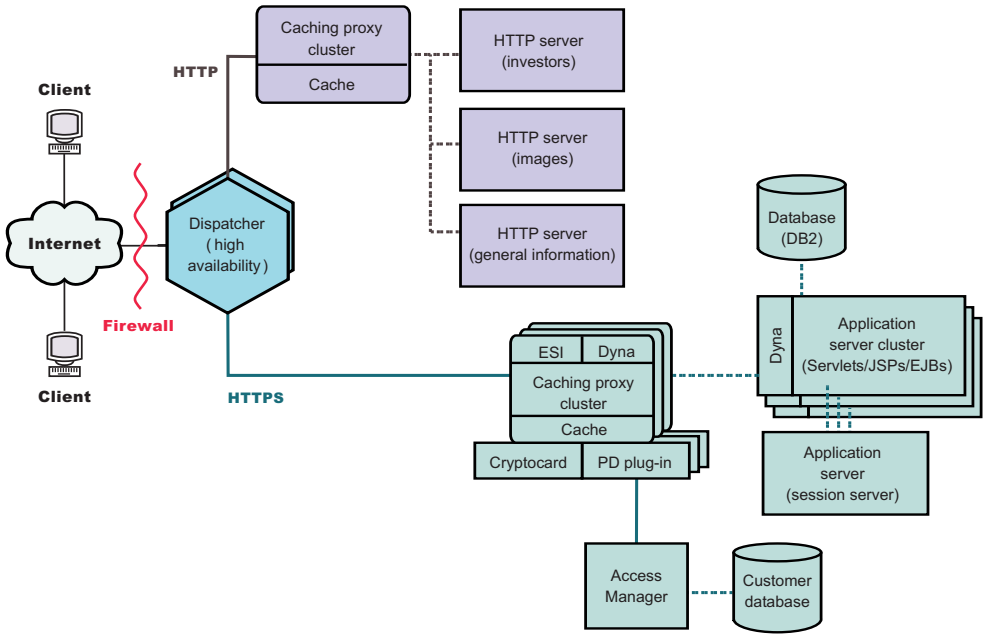

Figure 15. Business to consumer network (Phase 2)

Phase 3

Figure 16 on page 42 shows the third phase of the evolution of a business-to-consumer network, with the static Web adopting a dynamic presentation method. The proxy server cluster has been enhanced to support the caching of dynamic Web content and assembly of page fragments written to comply with the Edge Side Includes (ESI) protocol. Rather than using server-side include mechanisms to build Web pages on the content servers and then propagating these client-specific, noncacheable, pages through the entire network,

ESI mechanisms permit pages to be assembled from cached content at the edge of the network, thereby reducing bandwidth consumption and decreasing response time.

ESI mechanisms are crucial in this third-phase scenario, where each client receives a personalized home page from the Web site. The building blocks of these pages are retrieved from a series of WebSphere Application Servers. Application servers containing sensitive business logic and ties to secure databases are isolated behind a firewall.

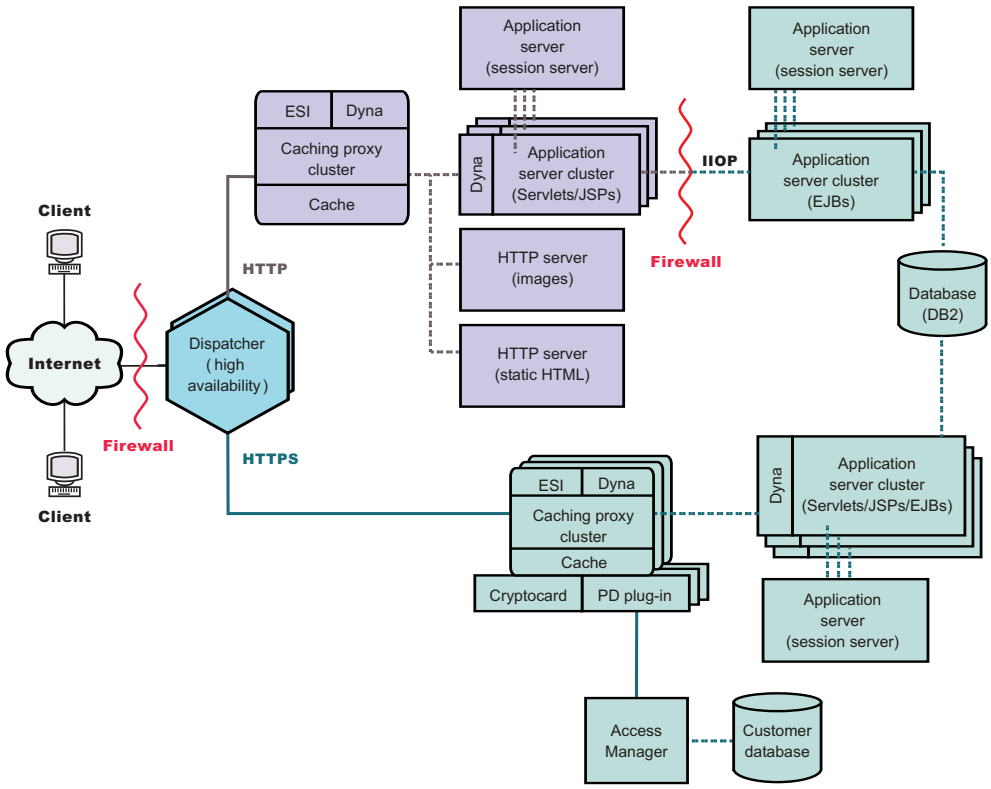

Figure 16. Business to consumer network (Phase 3)

Chapter 9. Business to client-banking-solution

Figure 17 on page 44 shows an efficient online-banking solution that is similar to the business-to-consumer network described in Chapter 8, "Business-to-consumer network," on page 39. All client requests pass through the firewall to a Dispatcher that separates traffic according to Internet protocol. HTTP requests pass to a cluster of proxy servers with active caches that act as surrogate servers for the Web servers. Metric servers are colocated with the proxy servers to provide load-balancing data to the Dispatcher. This arrangement reduces the network load on the Web servers and creates an additional buffer between them and the Internet.

HTTPS requests are passed to a secure network designed to provide clients with personal financial information and permit online-banking transactions. A cluster of enhanced proxy servers provides scalability to the site. These proxy servers support the caching of dynamic Web content and assembly of page fragments written to comply with the Edge Side Includes (ESI) protocol. A cryptographic hardware card manages SSL handshakes, which significantly reduces the processing required of the proxy server host, and an Access Manager (formerly Policy Director) directs client authentication.

A collection of application servers clusters distribute the processing of requests by separating the business logic, contained in EJB components, from these presentation layer, contained in servlets and JSP files. Each of these clusters is managed by a separate session server.

The following scenerio includes both Load Balancer and Caching Proxy.

IMPORTANT: Caching Proxy is available on all Edge component installations, with the following exceptions:

- Caching Proxy is not available for Edge component installations that run on Itanium 2 or AMD Opteron 64-bit processors.
- Caching Proxy is not available for Edge component installations of Load Balancer for IPv4 and IPv6.

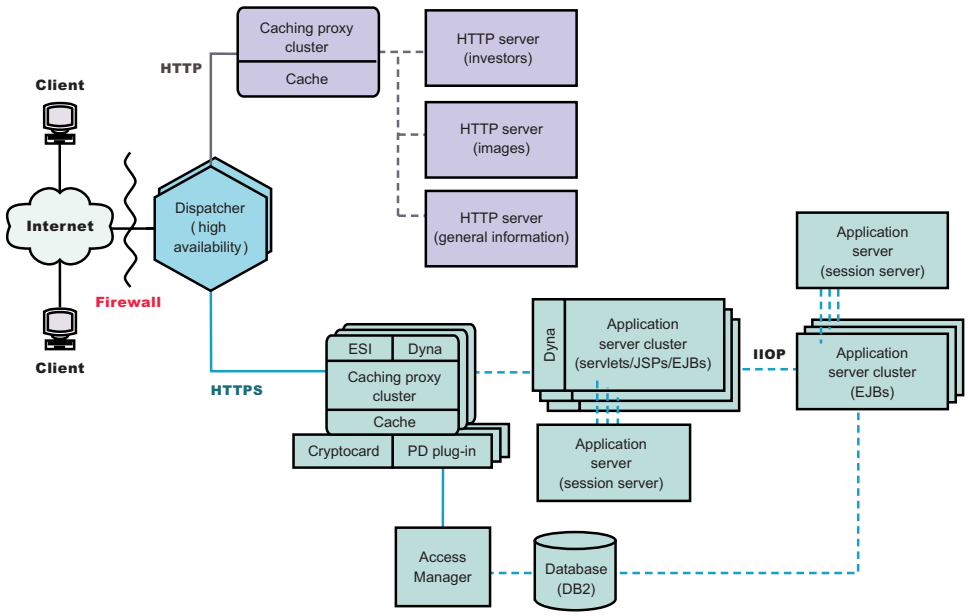

Figure 17. Business to consumer banking solution

Chapter 10. Web portal network

Figure 18 on page 46 shows a Web portal network designed to support a heavy volume of traffic while providing each client with personalized content. To minimize the processing load on the various servers, no part of the network carries SSL traffic. Because the portal does not deliver sensitive data, security is not an important issue. It is important for the databases containing client IDs, passwords, and settings to remain moderately secure and uncorrupted, but this requirement does not impair the performance of the rest of the Web site.

All client requests pass through the firewall to a Dispatcher that balances the network load across a cluster of proxy servers with active caches that act as surrogate servers for the Web servers. Metric servers are colocated with the proxy servers to provide load-balancing data to the Dispatcher.

The actual dynamic Web site is a cluster of application servers that generate ESI fragments that are passed to the proxy servers for assembly. Because of the reduced security concerns, each application server performs all necessary functions for constructing the Web site. All application servers are identical. If one application server goes out of service, the session server can route requests to the other servers, providing high availability for the entire site. This configuration also allows for rapid escalation of the Web site if excessive traffic occurs, for example, the hosting of a special event by the portal. Additional proxy servers and application servers can quickly be configured into the site.

All static content, such as image files and boilerplate text is stored on separate Web servers, allowing it to be updated as necessary without risking corruption to the more complex application servers.

The following scenerio includes both Load Balancer and Caching Proxy.

IMPORTANT: Caching Proxy is available on all Edge component installations, with the following exceptions:

- Caching Proxy is not available for Edge component installations that run on Itanium 2 or AMD Opteron 64-bit processors.
- Caching Proxy is not available for Edge component installations of Load Balancer for IPv4 and IPv6.

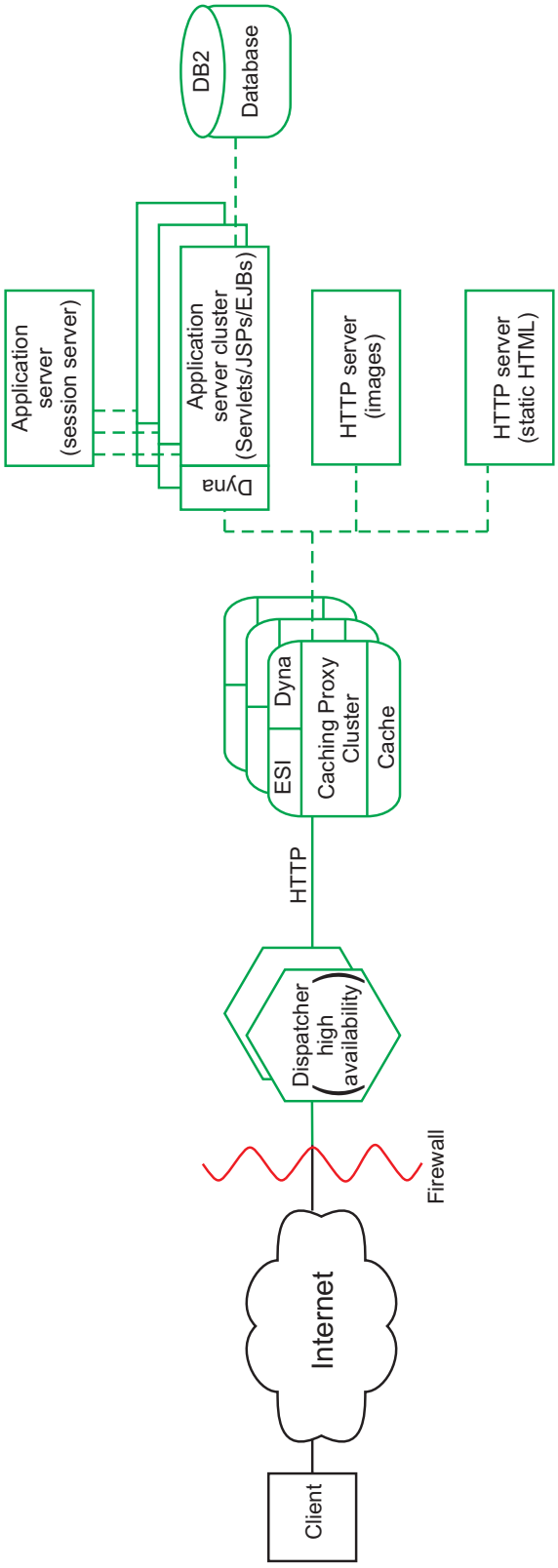

Figure 18. Web portal

Part 4. Installing Edge components

This part provides the procedures for installing Edge components.

This part contains the following chapters:

Chapter 11. Requirements for Edge components

This chapter provides a link to hardware and software requirements for Edge components and describes guidelines for using Web browsers with the Caching Proxy Configuration and Administration forms and with the Load Balancer online help.

Hardware and software prerequisites

For information on supported hardware and software requirements for WebSphere Application Server, Version 6.1 Edge components, link to the following Web page: *http://www.ibm.com/support/docview.wss?rs=180&uid=swg27006921*.

SDK installation: The Java 2 SDK automatically installs with Load Balancer on all platforms.

Using browsers with the Caching Proxy Configuration and Administration forms

Minimum browser requirements

To configure the Caching Proxy using the Configuration and Administration forms, your browser must do the following:

- Display frames.
- Have both JavaScript and Java enabled.
- Have color resolution set to at least 256 colors (operating system setting).
- Be set to cache documents and compare the cached document with the network document every time.

For Linux and UNIX systems: For the recommended versions of Mozilla and Firefox browsers, refer to the following Web site and follow the links to the supported software Web page: `http://www.ibm.com/support/docview.wss?rs=180&uid=swg27006921`.

For Windows systems: For the recommended versions of Internet Explorer, Mozilla, and Firefox browsers, refer to the following Web site and follow the links to the supported software Web page: http://www.ibm.com/support/docview.wss?rs=180&uid=swg27006921.

Note: On 64–bit PowerPC Linux systems, it will not be possible to access the Configuration and Administration forms with the Mozilla browser since there is no SDK available for this architecture. Alternatively, you can access the Configuration and Administration forms from a different machine with a supported web browser.

LIMITATION: The Administration forms left hand vertical scroll bar may not get displayed by the browser if the number of expanded elements is too large to be displayed in the window of the browser. This causes the expanded elements at the bottom of the list to be pushed out of the current viewing window of the browser and inaccessible. To solve this problem, limit the number of expanded elements in

the left hand menu . If the number of expanded elements is large, collapse some of the elements until the elements at the bottom of the list are displayed in the browser window.

In order to properly display forms, the operating system that is actually displaying the form (the one on which the browser resides) must contain the appropriate font sets for the language in which the form is written. The browser interface, however, does not necessarily need to be in the same language as the forms.

For example, a Chinese version of the proxy server is running on a Solaris 9 system. A Mozilla browser with an English-language interface is loaded onto the Solaris host. This browser can be used locally to edit the Configuration and Administration forms. (Forms are served to the browser in the character set used by the proxy server—in this example, Chinese; however, the forms might not be displayed correctly if the browser and its underlying operating system are not properly configured to display the character set sent by the proxy server.)

Alternatively, if a Windows workstation with Chinese language support is available to remotely connect to the proxy server, it is possible to load a Chinese version of a Netscape browser onto the Windows workstation and use this browser to enter values in the forms. This second solution has the advantage of maintaining a consistent language interface for the administrator.

The font sets specific to operating systems greatly affect the display of various languages, particularly of double-byte characters, within the browsers. For example, a particular Chinese font set on AIX does not look exactly the same as a Chinese font set on Windows platforms. This causes some irregularities in the appearance of HTML text and Java applets within the Configuration and Administration forms. For the best appearance, only browsers running on Windows operating systems are recommended.

Note about Mozilla 1.4 browser on S/390 and PowerPC

The Java plugin installed with Mozilla 1.4 must be updated to version 1.4.2 or higher in order for the Administration Forms to be displayed correctly. Use the following steps to update the plugin:

1. Link to *http://plugindoc.mozdev.org*
2. Select the platform from the ″Documentation″ section
3. Follow the instructions listed in the ″Java Runtime Environment″ section to update the plugin

Using browsers with the Load Balancer online help

To use the Load Balancer online help, your browser must support the following:

- HTML 4
- Cascading Style Sheets
- JavaScript technology
- Java applets

Using a browser that does not support these requirements can result in incorrectly formatted pages and functions that might not work correctly.

- **For Linux and UNIX systems:** The default browsers are Mozilla and Firefox. For the recommended versions of the browsers, refer to the following Web site and

follow the links to the supported software Web page: http://www.ibm.com/support/docview.wss?rs=180&uid=swg27006921.

- **For Windows systems:** The default browser is the system browser. For the recommended versions of Internet Explorer, Mozilla, and Firefox browsers, refer to the following Web site and follow the links to the supported software Web page: http://www.ibm.com/support/docview.wss?rs=180&uid=swg27006921.

Chapter 12. Installing Edge components using the setup program

This chapter provides instructions for installing Edge components using the setup program.

The Java 2 SDK automatically installs with Load Balancer on all platforms.

After installation, scripts within the Caching Proxy packaging attempt to start the proxy server using the default configuration. If port 80 is in use, such as by another Web server, the proxy server will fail to start.

IMPORTANT: Caching Proxy is available on all Edge component installations, with the following exceptions:

- Caching Proxy is not available for Edge component installations that run on Itanium 2 or AMD Opteron 64-bit processors.
- Caching Proxy is not available for Edge component installations of Load Balancer for IPv4 and IPv6.

Using the setup program for Windows

Use the setup program to install Edge Components onto your Windows® system as follows:

1. Ensure that the Windows system meets all of the hardware and software requirements (Chapter 11, "Requirements for Edge components," on page 49).
2. Log in as a user with administrator privileges.
3. Insert the Edge Components CD-ROM into the machine's CD-ROM drive. The LaunchPad starts automatically.
4. Click **Launch the installation wizard for WebSphere Application Server – Edge Components**. The setup program starts automatically. It prepares the InstallShield Wizard and opens the Welcome window.

   Note: If your machine does not support the Autoplay option, or if it is turned off, start the setup program manually by running the `setup.exe` program, located in the top-level directory of the CD-ROM.
5. Click **Next** to continue with installation. The Software License Agreement window opens.
6. Read the License Agreement and click **Yes** to accept all of its terms. The Component Selection window opens.

   If Edge components are already installed, the Maintenance Options window opens before the Component Selection window opens. Select the **Modify** radio button, then click **Next**. The Component Selection window opens.
7. Select the components to be installed.
8. To change the selection of subcomponents to be installed for a given component, click the name of the component to select it, then click **Change Subcomponents**. Another Component Selection window opens, showing the subcomponents of the active component. Use these same procedures to select the subcomponents to install, the language of the components, and the location where the components are to be installed.

9. Use the **Current Language** menus to select the language or languages in which you want to install Edge components. The available languages are listed in the menu on the left. The languages that are selected are listed in the menu on the right.
10. Use the Component Selection window to verify the installation location for the Edge components. You can either accept the default value or you can specify a new location by clicking **Change Folder**.

    Note: If you choose an installation location other than the default, make sure that there are no blank spaces in the path name, for example, avoid path names such as C:\My Files\edgeserver\.
11. Use the Component Selection window to verify that there is enough available space in the installation location that you selected. If there is not enough available space in the selected location, click **Change Folder** and specify a new installation location.
12. After you select your Edge components, installation location, and languages, click **Next**. Review the information in the Installation Confirmation window that opens. To change a choice or choices, click **Back** to return to the Component Selection window and then make your changes. After you verify your choices, click **Finish**.
13. The Edge components Product Setup Program begins installing the selected Edge components, and GSK if necessary, in the installation location that you specified.
14. The Setup Complete window opens. If you want to read the Edge components ReadMe file, ensure that the **Yes, I want to view the ReadMe file** check box is selected. The ReadMe file opens in your default browser.
15. Ensure that the **Yes, I want to restart my computer** check box is selected, and then click **Finish**. If you opted to view the ReadMe file, the machine is restarted when you close the browser window that displays the file. Otherwise, the Edge components Product Setup Program immediately closes and the machine restarts. Note that you must restart your machine before you can use the newly installed Edge components.

Limitation: Using the Tab key in the license agreement window toggles between the **I accept** and **I do not accept** options. However, you cannot use the Tab key to reach the **Back**, **Next**, or **Cancel** navigation options. As a workaround, use Shift+Tab to reach these navigation options. In addition, the Enter key works only on the navigation buttons, so you must use the spacebar to select **I accept** or **I do not accept** options.

Using the setup program for Linux and UNIX

If installing from the CD, you can use the setup program to install Edge components onto your Linux and UNIX systems as follows:

1. Ensure that the computer server meets all of the hardware and software requirements described in Chapter 11, “Requirements for Edge components,” on page 49.
   - Additionally for Linux systems: The compat-libstdc++-33 package, which contains the GCC 3.3 C++ compatibility libraries, is required to be installed.
2. Log in as the superuser, typically root.
3. Insert the Edge components CD-ROM into the machine’s CD-ROM drive. If necessary, mount the CD-ROM.
4. Change the working directory to the top-level directory of the CD-ROM.

5. Invoke the setup program by entering the following command:

   ```
   # ./install
   ```

   The Welcome window opens.
6. Click **Next** to continue with installation. The Software License Agreement window opens.
7. Read the License Agreement and click **Yes** to accept all of its terms. The Language Selection window opens.
8. Select the languages to be supported by this installation of Edge components. Click **Next**. The Component Selection window opens.
9. Select the components to be installed.
10. Click **Next**. The Installation Confirmation window opens.
11. Review the information in the Installation Confirmation window. If you want to change one or more of your choices, click **Back** to return to the Component Selection window and then make your changes. After you verify your choices, click **Proceed**.

    The setup program begins installing the selected Edge components and required packages.
12. The Installation Results Summary window opens. Review the results, then click **Finish**.

Limitation: Using the Tab key in the license agreement window toggles between the **I accept** and **I do not accept** options. However, you cannot use the Tab key to reach the **Back**, **Next**, or **Cancel** navigation options. As a workaround, use Shift+Tab to reach these navigation options. In addition, the Enter key works only on the navigation buttons, so you must use the spacebar to select **I accept** or **I do not accept** options.

On Red Hat Linux 3.0 Update 3: When running the installer program for Edge components , the buttons will not work if the GUI panel is maximized and then restored. To resolve this problem do the following:

1. Click the **X** button in the top right corner of the panel to close the installer program.
2. Reply **Yes** to the ″Do you want to exit?″ question.
3. Relaunch the installer program without maximizing and restoring the panel size.

On Linux and UNIX systems: If the setup program is used to install Edge components, then the GUI uninstaller can be used to uninstall Edge components. However, the Edge components GUI uninstaller cannot be used to uninstall a refresh pack that is installed using native commands. You must first uninstall the refresh pack using native commands (the operating system's commands) before you uninstall components using the GUI uninstaller.

For information on using native commands, see Chapter 13, "Installing Caching Proxy using system packaging tools," on page 57 and Chapter 14, "Installing Load Balancer using system packaging tools," on page 61.

Chapter 13. Installing Caching Proxy using system packaging tools

This chapter provides instructions for installing Caching Proxy using the system packaging tools.

After installation, scripts within the Caching Proxy packaging attempt to start the proxy server using the default configuration. If port 80 is in use, such as by another Web server, the proxy server will fail to start.

IMPORTANT: Caching Proxy is available on all Edge component installations, with the following exceptions:

- Caching Proxy is not available for Edge component installations that run on Itanium 2 or AMD Opteron 64-bit processors.
- Caching Proxy is not available for Edge component installations of Load Balancer for IPv4 and IPv6.

Using your operating system's package installation system, install the packages in the order listed in Table 2 on page 58. The following procedure details the typical steps necessary to complete this task.

1. Insert the Edge Components CD into your CD-ROM drive and mount the drive if necessary.
2. Become the local superuser root.

   ```
   su - root
   Password: password
   ```

3. Change to the appropriate directory on the CD.

   ```
   cd mount_point/package_directory/
   ```

4. Install the packages.

   On AIX®:

   ```
   installp -acXd ./packagename
   ```

   On HP-UX:

   ```
   swinstall -s source/ packagename
   ```

   On Linux:

   ```
   rpm -i ./packagename
   ```

   On Solaris:

   ```
   pkgadd -d ./packagename
   ```

Table 2. Caching Proxy components

Component	Packages installed (in recommended order)
Caching Proxy	1. gskit7 2. icu 3. admin 4. msg-cp-*lang* 5. cp
Edge component documentation	doc-en_US[1]

Notes:

1. Load Balancer documentation is supplied in two packages. The doc-en_US package includes all Edge components documentation, including the Load Balancer documents, and places them into the ../edge/doc/ directory. The documentation package associated with Load Balancer installations (Chapter 14, "Installing Load Balancer using system packaging tools," on page 61) installs only the Load Balancer documents and places them into a subdirectory within the ../edge/lb/ directory.

Table 3. AIX, HP-UX, and Solaris package file names

Generic package name	AIX fileset	HP-UX fileset	Solaris file name
admin	wses_admin.rte	WSES-ADMIN	WSESadmin
cp	wses_cp.base	WSES-CP	WSEScp
doc	wses_doc.en_US	WSES-DOC-en_US	WSESdocen
gskit7	gskkm.rte	gsk7bas	gsk7bas
icu	wses_icu.rte	WSES-ICU	WSESicu
msg-cp-*lang*	wses_cp.msg.*lang*[1] .base	WSES-cpm*lang*[2]	WSEScpm*lang*[3]

Notes:

1. On AIX, the variable *lang* refers to the substitution of one of the following language specific codes: en_US, de_CH, de_DE, es_ES, fr_CA, fr_CH, fr_FR, it_CH, it_IT, ja_JP, Ja_JP, ko_KR, pt_BR, zh_CN, ZH_CN, zh_TW, Zh_TW.
2. On HP-UX, the variable *lang* refers to the substitution of one of the following language specific codes: de_DE, en_US, es_ES, fr_FR, it_IT, ja_JP, ko_KR, zh_CN, zh_TW. (HP-UX does not support Portuguese Brazilian (pt_BR).)
3. On Solaris, the variable *lang* refers to the substitution of one of the following language specific codes: br, cn, cw, de, en, es, fr, it, ja, kr.

Table 4. Linux package file names

Generic package name	Linux file name
admin	WSES_Admin_Runtime-*release-version*[1].*hardw*[2].rpm
cp	WSES_CachingProxy-*release-version*[1].*hardw*[2].rpm
doc	WSES_Doc_en_US-*release-version*[1].*hardw*[2].rpm
gskit7	gsk7bas.rpm
icu	WSES_ICU_Runtime-*release-version*[1].*hardw*[2].rpm

Table 4. Linux package file names (continued)

Generic package name	Linux file name
msg-cp-*lang*	WSES_CachingProxy_msg_*lang*[3]-*release-version*[1].*hardw*[2].rpm

Notes:

1. *release-version* is the current release, for example: 6.1.0-0
2. The variable *hardw* refers to the substitution of one of the following: i686, s390, ppc64.
3. The variable *lang* refers to the substitution of one of the following language specific codes: de_DE, en_US, es_ES, fr_FR, it_IT, ja_JP, ko_KR, pt_BR, zh_CN, zh_TW.

The documentation package contains English only. Translations of the Edge Component documentation set are at the following Web site: `www.ibm.com/software/webservers/appserv/ecinfocenter.html`.

Uninstall Caching Proxy using system tools

To uninstall the packages:

On AIX:

```
installp -u packagename
```

To uninstall all of the Caching Proxy packages, use the command:

```
installp -u wses
```

On HP-UX:

```
swremove packagename
```

To query the installed Caching Proxy packages, use the command:

```
swlist | grep WSES
```

Packages should be removed in the reverse order they were installed.

On Linux:

```
rpm -e packagename
```

To query the installed Caching Proxy packages, use the command:

```
rpm -qa |grep -i  wses
```

Packages should be removed in the reverse order they were installed.

On Solaris:

```
pkgrm packagename
```

To query the installed Caching Proxy packages, use the command:

```
pkginfo | grep WSES
```

Packages should be removed in the reverse order they were installed.

Chapter 14. Installing Load Balancer using system packaging tools

This chapter documents the installation of Load Balancer on AIX, HP-UX, Linux, and Solaris systems:

- "Installing for AIX"
- "Installing for HP-UX" on page 65
- "Installing for Linux" on page 66
- "Installing for Solaris" on page 68

Depending on the type of installation, not all the Load Balancer component packages that are listed in this section are provided.

- For Edge Component installations that can provide both Load Balancer and Caching Proxy, all the Load Balancer install component packages are available.
- For Edge Component installations that can provide Load Balancer but not Caching Proxy, the CBR component package is not included with Load Balancer.
- For Edge Component for IPv6 installations (Load Balancer for IPv4 and IPv6), the Dispatcher component package is included with Load Balancer. The CBR, Site Selector, and the Controller component packages are not included.

  The recommended order for installing the packages is slightly different for Load Balancer for IPv4 and IPv6 installations. It is important to note that the administration component package must be installed after the dispatcher component package. The recommended order for installing packages for Load Balancer for IPv4 and IPv6 using the system's tools is: base, license, dispatcher component, administration, documentation, Metric Server

If you are migrating from a previous version of Load Balancer, or re-installing an operating system, prior to installation, you can save any of your previous configuration files or script files for Load Balancer.

- After installation, place your configuration files in the `...ibm/edge/lb/servers/configurations/`***`component`*** directory (where ***component*** is either dispatcher, cbr, ss, cco, or nal).
- After installation, place your script files (such as goIdle and goStandby) in the `.../ibm/edge/lb/servers/bin` directory in order to run them.

If you log off a machine after Load Balancer has been installed, you must restart all Load Balancer services when you log back on.

Installing for AIX

Table 5 lists the AIX filesets for Load Balancer and the recommended order of installation using the system's package installation tool.

Table 5. AIX filesets

Load Balancer components	AIX filesets
Base	ibmlb.base.rte
Administration (with messages)	• ibmlb.admin.rte • ibmlb.msg.*lang*.admin

Table 5. AIX filesets (continued)

Load Balancer components	AIX filesets
Device Driver	ibmlb.lb.driver
License	ibmlb.lb.license
Load Balancer components (with messages)	• ibmlb.*component*.rte • ibmlb.msg.*lang*.lb
Documentation (with messages)	• ibmlb.doc.rte • ibmlb.msg.en_US.doc
Metric Server	ibmlb.ms.rte

Notes:

1. The following can be substituted for the variable *component*: disp (dispatcher), cbr (CBR), ss (Site Selector), cco (Cisco CSS Controller), or nal (Nortel Alteon Controller).
2. The following can be substituted for the variable *lang*: en_US, de_CH, de_DE, es_ES, fr_CA, fr_CH, fr_FR, it_CH, it_IT, ja_JP, Ja_JP, ko_KR, pt_BR, zh_CN, ZH_CN, zh_TW, Zh_TW

The documentation package contains English only. Translations of the Edge Component documentation set are at the following Web site: `www.ibm.com/software/webservers/appserv/ecinfocenter.html`.

Before you install

Before you install Load Balancer for AIX, ensure the following:

- You are logged in as root.
- The Edge components media is inserted, or if you are installing from the Web, the installation images are copied to a directory. Any earlier version of the product is uninstalled. To uninstall, ensure that all the executors and all the servers are stopped. Then, to uninstall the entire product, enter the following command:

  ```
  installp -u ibmlb
  ```

  or, for previous versions, enter the following command:

  ```
  installp -u ibmnd
  ```

  To uninstall specific filesets, list them specifically instead of specifying the package name `ibmlb`.

When you install the product, you are given the option of installing any or all of the following:

- Base Administration
- Administration (with messages)
- Device Driver (required)
- License (required)
- Dispatcher component (with messages)
- CBR component (with messages)
- Site Selector component (with messages)
- Cisco CSS Controller component (with messages)
- Nortel Alteon Controller component (with messages)

- Documentation (with messages)
- Metric Server

Installation procedure

It is recommended that you use SMIT to install Load Balancer for AIX because SMIT ensures that all messages are installed automatically.

Using SMIT to install Load Balancer for AIX

1. Select **Software Installation and Maintenance**.
2. Select **Install and Update Software**.
3. Select **Install and update from latest Available Software**.
4. Enter the device or directory containing the filesets.
5. In the ***SOFTWARE to Install** field, enter the appropriate information to specify options (or select List).
6. Press **OK**.
7. When the command completes, press **Done**.
8. Close SMIT by selecting **Exit Smit** from the **Exit** menu or pressing **F12**. If you are using SMITTY, press **F10** to close the program.

Installing Load Balancer from the command line

1. If installing from a CD, enter the following commands to mount the CD:

   ```
   mkdir /cdrom
   mount -v cdrfs -p -r /dev/cd0 /cdrom
   ```

2. Refer to the following table to determine which command or commands to enter to install the desired Load Balancer packages for AIX:

   Table 6. AIX installation commands

Packages	Commands
Base	installp -acXgd *device* ibmlb.base.rte
Administration (with messages)	installp -acXgd *device* ibmlb.admin.rte ibmlb.msg.*language*.admin
Device Driver	installp -acXgd *device* ibmlb.lb.driver
License	installp -acXgd *device* ibmlb.lb.license
Load Balancer components (with msgs). Includes: Dispatcher, CBR, Site Selector, Cisco CSS Controller, and Nortel Alteon Controller	installp -acXgd *device* ibmlb.*component*.rte ibmlb.msg.*language*.lb
Documents (with messages)	installp -acXgd *device* ibmlb.doc.rte ibmlb.msg.en_US.lb
Metric Server	installp -acXgd *device* ibmlb.ms.rte

   where *device* is:

   - /cdrom if you are installing from a CD.
   - */dir* (the directory containing the filesets) if you are installing from a file system.

3. Ensure that the result column in the summary contains SUCCESS for each part of Load Balancer that you are installing (APPLYing). Do not continue until all of the parts that you wish to install are successfully applied.

   Note: To generate a list of filesets on a specified device, including all available message catalogs, enter

   ```
   installp -ld device
   ```

If installing from a CD, to unmount the CD, enter the following command:

```
unmount /cdrom
```

Verify that the product is intalled by entering the following command

```
lslpp -h | grep ibmlb
```

If you installed the full product, this command returns the following:

```
ibmlb.base.rte
ibmlb.admin.rte
ibmlb.lb.driver
ibmlb.lb.license
ibmlb.component.rte
ibmlb.doc.rte
ibmlb.ms.rte
ibmlb.msg.language.admin
ibmlb.msg.en_US.doc
ibmlb.msg.language.lb
```

Load Balancer installation paths include the following:

- Administration — /opt/ibm/edge/lb/admin

- Load Balancer components — /opt/ibm/edge/lb/servers
- Metric Server — /opt/ibm/edge/lb/ms
- Documentation (*Administration Guide*) — /opt/ibm/edge/lb/documentation

Installing for HP-UX

This section explains how to install Load Balancer on HP-UX using the product CD.

Before you install

Before beginning the installation procedure, ensure that you have root authority to install the software.

If you have an earlier version installed, you should uninstall that copy before installing the current version. First, ensure that you have stopped both the executor and the server. Then, to uninstall Load Balancer see "Instructions for uninstalling the packages" on page 66.

Installation procedure

Table 7 lists the names of the installation packages for Load Balancer and the recommended order to install the packages using the system's package installation tool.

Table 7. HP-UX package installation details for Load Balancer

Package description	HP-UX package name
Base	ibmlb.base
Administration and messages	ibmlb.admin ibmlb.nlv-*lang*
Load Balancer License	ibmlb.lic
Load Balancer components	ibmlb.*component*
Documentation	ibmlb.doc
Metric Server	ibmlb.ms

Notes:

1. The variable *lang* refers to the substitution of one of the following language specific codes: de_DE, es_ES, fr_FR, it_IT, ja_JP, ko_KR, zh_CN, zh_TW.
2. The variable *component* refers to the substitution of one of the following: disp (dispatcher), cbr (CBR), ss (Site Selector), cco (Cisco CSS Controller), or nal (Nortel Alteon Controller).
3. The documentation package (ibmlb.doc) contains English only. Translations of the Edge Component documentation set are at the following Web site: `www.ibm.com/software/webservers/appserv/ecinfocenter.html`.

HP-UX does not support the Portuguese Brazilian (pt_BR) locale. The supported locales on HP-UX are:

- de_DE.iso88591
- en_US.iso88591
- es_ES.iso88591
- fr_FR.iso88591
- de_DE.iso88591
- it_IT.iso88591

- ja_JP.SJIS
- ko_KR.eucKR
- zh_CN.hp15CN
- zh_TW.big5

Instructions for installing the packages

The following procedure details the steps necessary to complete this task.

1. Become the local superuser root.

   ```
   su - root
   Password: password
   ```

2. Issue the install command to install the packages

   Issue the install command

   ```
   swinstall -s /source package_name
   ```

   where *source* is the absolute directory path for the location of the package, and *package_name* is the name of the package.

   For example, the following installs the base package for Load Balancer (ibmlb.base), if you are installing from the root of the CD

   ```
   swinstall -s /source ibmlb.base
   ```

   To install all the packages for Load Balancer issue the following command, if you are installing from the root of the CD:

   ```
   swinstall -s /source ibmlb
   ```

3. Verify the installation of the Load Balancer packages

   Issue **swlist** command to list all the packages that you have installed. For example,

   ```
   swlist -l fileset ibmlb
   ```

Instructions for uninstalling the packages

Use the **swremove** command to uninstall the packages. The packages should be removed in the reverse order they were installed. For example, issue the following:

- To uninstall all the Load Balancer packages

  ```
  swremove ibmlb
  ```

  To uninstall an individual package (for example the Cisco CSS Controller)

  ```
  swremove ibmlb.cco
  ```

Load Balancer installation paths include the following:

- Administration — /opt/ibm/edge/lb/admin
- Load Balancer components — /opt/ibm/edge/lb/servers
- Metric Server — /opt/ibm/edge/lb/ms
- Documentation (*Administration Guide*) — /opt/ibm/edge/lb/documentation

Installing for Linux

This section explains how to install Load Balancer on Linux using the Edge components CD.

Before you install

Before installing Load Balancer, ensure the following:

- You are logged in as root.

- Any earlier version of the product is uninstalled. To uninstall, ensure that all the executors and all the servers are stopped. Then to uninstall the entire product, enter the following command:

  ```
  rpm -e pkgname
  ```

  When uninstalling, reverse the order used for package installation, ensuring that the administration packages are uninstalled last.

Installation steps

1. Insert the Edge Components media or download the product from the Web site and install the installation image using RPM (Red Hat Packaging Manager).

   The installation image is a file in the format lblinux-*version*.tar.

2. Untar the tar file in a temporary directory by entering the following command:

   ```
   tar -xf lblinux-version.tar
   ```

   The result is the following set of files with the .rpm extension:

   - ibmlb-base-*release-version.hardw*.rpm (Base)
   - ibmlb-admin-*release-version.hardw*.rpm (Administration)
   - ibmlb-lic-*release-version.hardw*.rpm (License)
   - ibmlb-*component-release-version.hardw*.rpm (LB component)
   - ibmlb-doc-*release-version.hardw*.rpm (Documentation)
   - ibmlb-ms-*release-version.hardw*.rpm (Metric Server)

   Where —

   - *release-version* is the current release, for example: 6.1.0-0
   - *hardw* is one of the following values: i386, ppc64, ppc, s390, s390x, x86_64
   - *component* is one of the following values: disp (Dispatcher component), cbr (CBR component), ss (Site Selector component), cco (Cisco CSS Controller), nal (Nortel Alteon Controller)

   The documentation package contains English only. Translations of the Edge Component documentation set are at the following Web site: `www.ibm.com/software/webservers/appserv/ecinfocenter.html`.

3. From the directory where the RPM files are located, issue the command to install each package. For example:

   ```
   rpm -i package.rpm
   ```

   Red Hat Linux systems: Due to a known Red Hat Linux problem, you will also need to delete the _db* RPM files, or an error will occur.

   It is important to install the packages in the order shown in the following list of packages needed for each component.

   - Base (base)
   - Administration (admin)
   - License (lic)
   - Load Balancer components (ds, cbr, ss, cco, nal)
   - Metric Server (ms)
   - Documentation (doc)

   Note: At least one of the RPM files requires that Java™ is installed and registered in the RPM database. If Java is installed but not registered in the RPM database, use the installation command with a no dependencies option as follows:

```
rpm -i --nodeps package.rpm
```

4. Verify that the product is installed. Enter the following command:

   ```
   rpm -qa | grep ibmlb
   ```

   Installing the full product produces the following output:

   - ibmlb-base-*release-version*
   - ibmlb-admin-*release-version*
   - ibmlb-lic-*release-version*
   - ibmlb-dsp-*release-version*
   - ibmlb-cbr-*release-version*
   - ibmlb-ss-*release-version*
   - ibmlb-cco-*release-version*
   - ibmlb-nal-*release-version*
   - ibmlb-doc-*release-version*
   - ibmlb-ms-*release-version*

Load Balancer installation paths include the following:

- Administration — /opt/ibm/edge/lb/admin
- Load Balancer components — /opt/ibm/edge/lb/servers
- Metric Server — /opt/ibm/edge/lb/ms
- Documentation — /opt/ibm/edge/lb/documentation

If you need to uninstall the packages, reverse the order used for package installation, ensuring that the administration packages are uninstalled last.

Installing for Solaris

This section explains how to install Load Balancer on Solaris using the Edge components CD.

Before you install

Before beginning the installation procedure, ensure that you are logged in as root and that any previous version of the product is uninstalled.

To uninstall, ensure that all the executors and the servers are stopped. Then, enter the following command:

```
pkgrm pkgname
```

Installation steps

1. Insert the CD-ROM that contains the Load Balancer software into the appropriate drive.
2. At the command prompt, enter the following command:

   ```
   pkgadd -d pathname
   ```

   where `-d pathname` is the device name of the CD-ROM drive or the directory on the hard drive where the package is located; for example: `-d /cdrom/cdrom0/`.

   The following is a list of packages displayed and the recommended order in which they should be installed.

   - ibmlbbase (Base)

- ibmlbadm (Administration)
- ibmlblic (License)
- ibmlbdisp (Dispatcher component)
- ibmlbcbr (CBR component)
- ibmlbss (Site Selector component)
- ibmlbcco (Cisco CSS Controller component)
- ibmlbnal (Nortel Alteon Controller component)
- ibmlbdoc (Documentation)
- ibmlbms (Metric Server)

The documentation package (ibmlbdoc) contains English only. Translations of the Edge Component documentation set are at the following Web site: `www.ibm.com/software/webservers/appserv/ecinfocenter.html`.

If you want to install all of the packages, simply type `all` and press Return. If you want to install some of the components, enter the name or names corresponding to the packages to be installed, separated by a space or comma, and press Return. You might be prompted to change permissions on existing directories or files. Simply press Return or respond `yes`. You need to install the prerequisite packages (because the installation follows alphabetical, not prerequisite order). If you type `all`, then respond `yes` to all prompting, and the installation completes successfully.

If you want to install just the Dispatcher component with the documentation and Metric Server, you must install the following packages: ibmlbbase, ibmlbadm, ibmlblic, ibmdisp, ibmlbdoc, and ibmlbms.

3. Verify that the product is installed. Issue the following command:

```
pkginfo | grep ibm
```

The Load Balancer installation paths include the following:

- Administration — /opt/ibm/edge/lb/admin
- Load Balancer components — /opt/ibm/edge/lb/servers
- Metric Server — /opt/ibm/edge/lb/ms
- Documentation — /opt/ibm/edge/lb/documentation

Part 5. Building networks with Edge components

This part provides procedures for building basic demonstration networks using Edge components. These networks are not intended to be used in production environments. The process of initially configuring a network can clarify many edge-of-network concepts for administrators who are new to the product. For complete coverage of all component features and for in-depth configuration information, refer to the *Caching Proxy Administration Guide* and the *Load Balancer Administration Guide*.

The procedures permit any computer system supported by the component to be used at any node.

This part contains the following chapters:

Chapter 15. Build a Caching Proxy network

Figure 19 shows a basic proxy server network using three computer systems located at three network nodes. This network binds the proxy server to a dedicated content host (IBM HTTP Server), which is located on Server 2, and the proxy server serves the host. This is visually represented by the Internet being located between the workstation and Server 1.

IMPORTANT: Caching Proxy is available on all Edge component installations, with the following exceptions:

- Caching Proxy is not available for Edge component installations that run on Itanium 2 or AMD Opteron 64-bit processors.
- Caching Proxy is not available for Edge component installations of Load Balancer for IPv4 and IPv6.

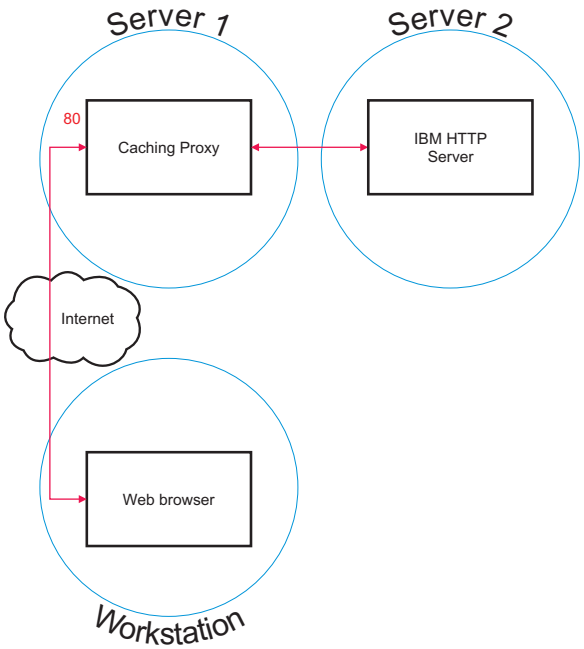

Figure 19. Caching Proxy demonstration network

Workflow

To build a Caching Proxy network, peform these procedures in the following order:

1. Review required computer systems and software.
2. Build Server 1 (Linux and UNIX systems) or Build Server 1 (Windows system).
3. Configure Server 1.
4. Test the Caching Proxy network.

Review required computer systems and software

The following computer systems and software components are needed:

- Computer system to act as Server 1. This system must have access to the Internet.
- Computer system to act as Server 2. An HTTP server must be installed on the content host.
- Computer system to act as the workstation. A Web browser must be installed.

Build Server 1 (Linux and UNIX systems)

Install and configure the Caching Proxy as follows:

1. Ensure that the computer server meets all hardware and software requirements.
2. Log in as the superuser, typically root.
3. Install the Caching Proxy component.
4. Create an administrator identification and password for accessing the Configuration and Administration forms by entering the following command:

   ```
   # htadm -adduser /opt/ibm/edge/cp/server_root/protect/webadmin.passwd
   ```

   When prompted, provide the **htadm** program with a user name, password, and real name for the administrator.
5. Continue with "Configure Server 1."

Build Server 1 (Windows system)

Install and configure the Caching Proxy as follows:

1. Ensure that the Windows 2000 and Windows 2003 operating systems meet all hardware and software requirements.
2. Log in as a user with administrator privileges.
3. Install the Caching Proxy component.
4. Create an administrator identification and password for accessing the Configuration and Administration forms by entering the following command:

   ```
   cd "Program Files\IBM\edge\cp\server_root\protect"
   htadm -adduser webadmin.passwd"
   ```

   When prompted, provide the **htadm** program with a user name, password, and real name for the administrator.
5. Continue with "Configure Server 1."

Configure Server 1

From the workstation, do the following:

1. Start a Web browser.
2. In the **Address** field of your browser, enter `http://`*server_1*, where *server_1* refers to the actual host name or IP address of the machine designated as Server 1.
3. Click **Configuration and Administration Forms**.
4. Enter your administrator name and password. The Configuration and Administration forms open in your browser.
5. Click **Server Configuration—>Request Processing—>Request Routing**.
6. Insert a new wildcard-mapping rule prior to the existing one by selecting the **Insert Before** radio button and the index value of the existing wildcard-mapping rule.
7. Select **Proxy** from the **Action** drop-down box.
8. Type /* in the **URL request template field**.

9. Type the host name for the site to which to redirect HTTP requests in the **Server IP Address or host name** field. Precede this value with `http://`.
10. Click **Submit**.
11. Create a mapping rule that allows access to the Configuration and Administration forms by selecting the **Insert Before** radio button and the index value of the mapping rule created in step 6.
12. Select **Pass** from the **Action** drop-down box.
13. Type `/pub/*` in the **URL request template field**.
14. Enter the location of the Configuration and Administration forms:
    - If Caching Proxy resides on a Linux or UNIX machine, type `/opt/ibm/edge/cp/server_root/pub/en_US/*` in the **Server IP Address or host name** field.
    - If Caching Proxy resides on a Windows machine, type `"C:\Program Files\IBM\edge\cp\server_root\pub\en_US\*"` in the **Server IP Address or host name** field.
15. Click **Submit**.
16. Click the **Restart Server** icon at the top of the configuration form.
17. Continue with "Test the Caching Proxy network."

Test the Caching Proxy network

From the workstation, do the following:

1. Start a Web browser.
2. Enter `http://`*`server_1`* in the **Address** field of your browser. HTML pages from Server 2 will be proxied through Server 1 and delivered to the Web browser.
3. To access the Configuration and Administration forms, enter `http://`*`server_1`*`/pub/` into the **Address** field of your browser. The home page of the Configuration and Administration forms is displayed.

Chapter 16. Build a Load Balancer network

Figure 20 shows a basic Load Balancer network with three locally attached workstations using the Dispatcher component's MAC forwarding method to load balance Web traffic between two Web servers. The configuration is similar when load balancing any other TCP or stateless UDP application traffic.

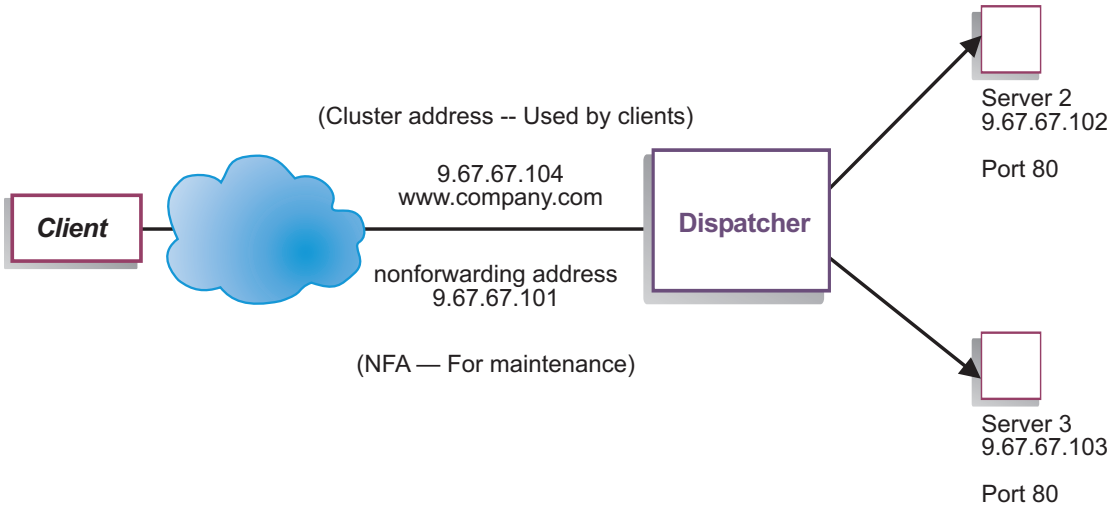

Figure 20. Load Balancer demonstration network

Note: This configuration can be completed using only two workstations with Dispatcher located on one of the Web server workstations. This represents a collocated configuration.

Workflow

To build a Load Balancer network, perform these procedures in the following order:

1. Review required computer systems and software.
2. Configure the network.
3. Configure the Dispatcher.
4. Test the Load Balancer network.

Review required computer systems and software

The following computer systems and software components are needed:

- Computer system to act as the Dispatcher. This system requires one actual IP address and one address to be load balanced.
- Two computer systems to act as Web servers. Each Web server requires one IP address.

Configure the network

1. Set up your workstations so that they are on the same LAN segment. Ensure that network traffic among the three machines does not have to pass through any routers or bridges.
2. Configure the network adapters of the three workstations. For this example, assume you have the following network configuration:

Workstation	Name	IP Address
1	server1.company.com	9.67.67.101
2	server2.company.com	9.67.67.102
3	server3.company.com	9.67.67.103
Netmask = 255.255.255.0		

   Each of the workstations contains only one standard Ethernet network interface card.
3. Ensure that server1.company.com can ping both server2.company.com and server3.company.com.
4. Ensure that server2.company.com and server3.company.com can ping server1.company.com.
5. Ensure that content is identical on the two Web servers (Server 2 and Server 3). This can be done by replicating data on both workstations, by using a shared file system such as NFS, AFS®, or DFS™, or by any other means appropriate for your site.
6. Ensure that Web servers on server2.company.com and server3.company.com are operational. Use a Web browser to request pages directly from http://server2.company.com and http://server3.company.com.
7. Obtain another valid IP address for this LAN segment. This is the address you provide to clients who wish to access your site. For this example, the information is as follows:

   ```
   Name= www.company.com
   IP=9.67.67.104
   ```
8. Configure the two Web server workstations to accept traffic for www.company.com.

   Add an alias for www.company.com to the **loopback** interface on server2.company.com and server3.company.com.

   - For AIX:

     ifconfig lo0 alias www.company.com netmask 255.255.255.0
   - For Solaris 7:

     ifconfig lo0:1 www.company.com 127.0.0.1 up
9. Delete any extra route that might have been created as a result of aliasing the loopback interface.

   You have now completed all configuration steps that are required on the two Web server workstations.

Configure the Dispatcher

With Dispatcher, you can create a configuration by using the command line, the configuration wizard, or the graphical user interface (GUI).

Note: The parameter values must be typed in English characters. The only exceptions are parameter values for host names and file names.

Configuring using the command line

If you are using the command line, follow these steps:

1. Start the dsserver on Dispatcher:
   - For AIX, HP-UX, Linux, or Solaris, run the following command as root user:
     `dsserver`
   - For Windows platforms, dsserver runs as a service that starts automatically.
2. Start the executor function of Dispatcher:
   ```
   dscontrol executor start
   ```
3. Add the cluster address to the Dispatcher configuration:
   ```
   dscontrol cluster add www.company.com
   ```
4. Add the http protocol port to the Dispatcher configuration:
   ```
   dscontrol port add www.company.com:80
   ```
5. Add each of the Web servers to the Dispatcher configuration:
   ```
   dscontrol server add www.company.com:80:server2.company.com
   dscontrol server add www.company.com:80:server3.company.com
   ```
6. Configure the workstation to accept traffic for the cluster address:
   ```
   dscontrol executor configure www.company.com
   ```
7. Start the manager function of Dispatcher:
   ```
   dscontrol manager start
   ```
   Dispatcher now does load balancing based on server performance.
8. Start the advisor function of Dispatcher:
   ```
   dscontrol advisor start http 80
   ```
   Dispatcher now ensures that client requests are not sent to a failed Web server.

Your basic configuration with locally attached servers is now complete.

IMPORTANT: With the **Load Balancer for IPv4 and IPv6** installation, the syntax for the Dispatcher command (dscontrol) is identical with one important exception. The delimiter for dscontrol commands is an at (@) symbol, instead of a colon (:). (It was necessary to define a delimiter other than a colon because the IPv6 format makes use of a colon within its addressing scheme.)

For instance (from the previous Dispatcher configuration example)

- On a Load Balancer for IPv4 and IPv6 installation, to add the http protocol port to the Dispatcher configuration:
  ```
  dscontrol port add www.company.com@80
  ```
- On a Load Balancer for IPv4 and IPv6 installation, to add each of the Web servers to the Dispatcher configuration:
  ```
  dscontrol server add www.company.com@80@server2.company.com
  dscontrol server add www.company.com@80@server3.company.com
  ```

For more information, if you are using a Load Balancer for IPv4 and IPv6 installation, see the chapter for deploying Dispatcher on Load Balancer for IPv4 and IPv6, which includes information on limitations and configuration differences, in the *WebSphere Application Server Load Balancer Administration Guide*.

Configuring using the configuration wizard

If you are using the configuration wizard, follow these steps:

1. Start the dsserver on Dispatcher:
   - For AIX, HP-UX, Linux, or Solaris, run the following command as root user:

`dsserver`

- For Windows systems, dsserver runs as a service that starts automatically.

2. Start the wizard function of Dispatcher, `dswizard`.

The wizard guides you step-by-step through the process of creating a basic configuration for the Dispatcher component. It asks questions about your network and guides you through the setup of a cluster for Dispatcher to load balance the traffic for a group of servers.

The configuration wizard contains the following panels:

- Introduction to the wizard
- What is going to happen
- Preparing for the setup
- Choosing a host to configure (if necessary)
- Defining a cluster
- Adding a port
- Adding a server
- Starting an advisor
- Server machine setup

Configuring using the graphical user interface (GUI)

To start the GUI, follow these steps:

1. Ensure that the dsserver process is running:
   - For AIX, HP-UX, Linux, or Solaris, run the following command as root:
     `dsserver`
   - For Windows systems, dsserver runs as a service that starts automatically.
2. Next, do one of the following:
   - For AIX, HP-UX, Linux, or Solaris, type `lbadmin`.
   - For Windows systems, click **Start** > **Programs** > **IBM WebSphere** > **Edge Components** > **IBM Load Balancer** > **Load Balancer**.

Test the Load Balancer network

1. From a Web browser, go to location http://www.company.com to verify that a page appears.
2. Reload the page in the Web browser.
3. Issue the following command: `dscontrol server report www.company.com:80:`. Verify that the total connections column of the two servers adds up to **2**.

Notices

First edition (May 2006)

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Corporation
Attn.: G71A/503
P.O. box 12195
3039 Cornwallis Rd.
Research Triangle Park, N.C. 27709-2195
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the document. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
ATTN: Software Licensing
11 Stanwix Street
Pittsburgh, PA 15222-9183
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples may include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of IBM Corporation in the United States, other countries, or both:

- AFS
- AIX
- DFS
- IBM

- iSeries™
- RS/6000®
- SecureWay®
- Tivoli
- ViaVoice
- WebSphere

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel Inside (logos), MMX and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Printed in USA

GC31-6918-00

Spine information:

WebSphere Application Server

Concepts, Planning, and Installation for Edge Components

Version 6.1

GC31-6918-00