

Increasing Web Site Performance: Advanced Infrastructure and Caching Concepts

By Marc Mulzer

A distributed Web site architecture and advanced caching techniques help companies deliver content to the edge of the Internet, which reduces download times for users and enhances their Web site experiences. This article describes the use of intelligent domain name system (DNS) routing and edge clusters to implement a globally distributed Web infrastructure. It also discusses various strategies for caching dynamic content.

In today's competitive e-business market, corporations that operate large Web sites face two major problems. First, e-businesses need ways to make site visits more enjoyable for users and to convince customers to return. In addition, they need to minimize the cost of setting up and maintaining such Web sites while ensuring scalability and fault tolerance. This article explores advanced Web site infrastructure and caching concepts that can help resolve these issues.

A distributed approach

In a traditional Web site implementation, a single data center houses servers that store all of the Web content and data. The Internet provides the infrastructure for any user to access the servers at this central location from any place worldwide. If a user is geographically close to the central content store, the user's request for information will take a short route and be fulfilled quickly, especially with a broadband or local network connection to the Internet. However, requests from distant locations or dial-up connections must take a longer route to connect to a server, and downloads take more time.

Local caching can decrease response time as well as the amount of work the Web servers must do. Nevertheless, a site visitor from

afar may still perceive a Web site to be slow because the requested information travels a long way before it arrives at the browser.

Enterprises are addressing this latency issue by moving away from a centralized approach toward a globally distributed one. In a distributed approach, regionally relevant content is delivered from a location close to the user as quickly as possible. The implementation of this approach requires intelligent domain name system (DNS) routing systems together with content replication and delivery mechanisms.

Intelligent DNS routing systems

To access a Web site, a user types the Web address into the browser. The browser then queries a DNS server to resolve the IP address for this Web site. The DNS server uses DNS databases to retrieve the IP address for the particular domain and sends it back to the browser. The browser then uses the IP address to contact the Web server to request content.

Normally, a DNS server matches one domain name with a single IP address. Using intelligent DNS routing, a DNS server has a pool of IP addresses available for a given domain name. The IP address that the server chooses depends on conditions set by the administrator, which are based on the following:

- ▶▶ User location
- ▶▶ Connection bandwidth such as dial-up, broadband, local area network (LAN), analog, or Integrated Services Digital Network (ISDN)
- ▶▶ Internet service provider such as AOL or UUNet

For example, consider a company, CO1234.COM, based in Atlanta, Georgia, which has two regional data centers: one in Atlanta and one in London, England. The administrator has assigned two IP addresses for the CO1234.COM Web site on an intelligent DNS server; one IP address corresponds to a Web server in the Atlanta data center and the other to an identically configured Web server in London.

The administrator has defined a rule on the DNS server to route visitors to the closest regional data center. When a client requests the IP address for CO1234.COM, the DNS server inspects the IP address of the client, determines its location, and redirects it to the appropriate regional data center: U.S.-based clients are redirected to Atlanta and European-based clients are redirected to London (see Figure 1).

If both data centers have identical content, European-based users are now much closer to the original content than if the company had only one central data center in Atlanta. Their download times are ideally much shorter. In addition, either data center can be used to maintain the availability of the company's Web site when one data center is unavailable because of technical failures or natural catastrophes.

Clustering on the edge

Companies can make specific content available at the outermost edge of the Internet—bringing content as close to the end user as possible—by using locally distributed DNS servers in addition to

globally distributed DNS servers. These local servers forward user requests from a regional data center to satellite server farms, or edge clusters (see Figure 2). Local routing decisions depend on specific regional factors, such as the user's Internet service provider or Internet connection type.

Dell offers intelligent DNS routing with the Dell® PowerApp.BIG-IP 220, a turnkey load-balancing appliance designed for high availability and efficient IP traffic management over local and global networks. The appliance can be deployed as a single unit or a redundant pair.

Advanced caching concepts

Although intelligent DNS routing is a first step in the right direction for distributed enterprise Web sites, it works well only in scenarios where the Web applications serve mostly static content. To generate content on the fly, Web applications rely on both specifically designed application servers and database servers. Replicating the entire application framework in several regional data centers simply to reduce load time is complicated, time consuming, and very cost intensive. Advanced caching strategies help eliminate the negative aspects of this scenario.

Caching of pre-rendered pages

Dynamic content eventually becomes static HTML once the application server has finished processing a request, such as a database query. An administrator can implement an automated process called *Web-site spidering* or *page pre-rendering* that requests every dynamic page from the application server prior to releasing the site (see Figure 3).

Instead of displaying HTML in a browser, a spider process saves the result for each request in a special location on the network. The process turns an entire dynamic site into static HTML pages, which

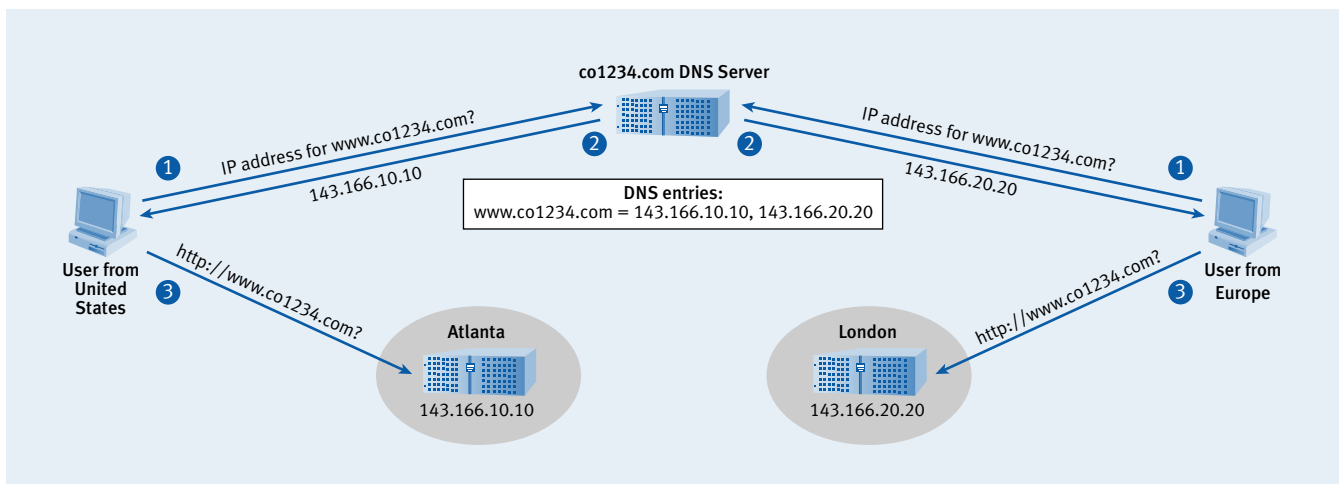


Figure 1. Intelligent DNS routing in a distributed architecture

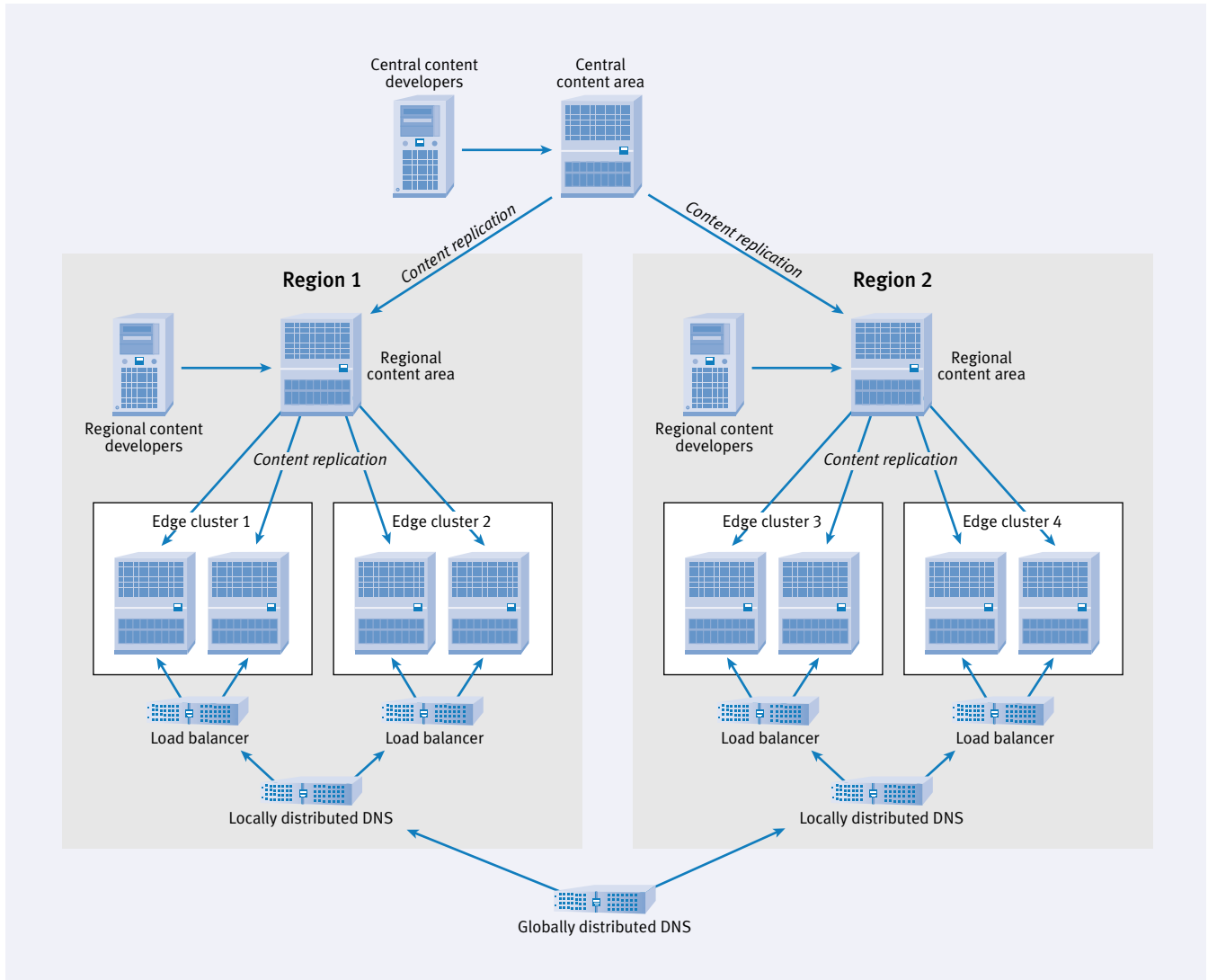


Figure 2. Regionally distributed content

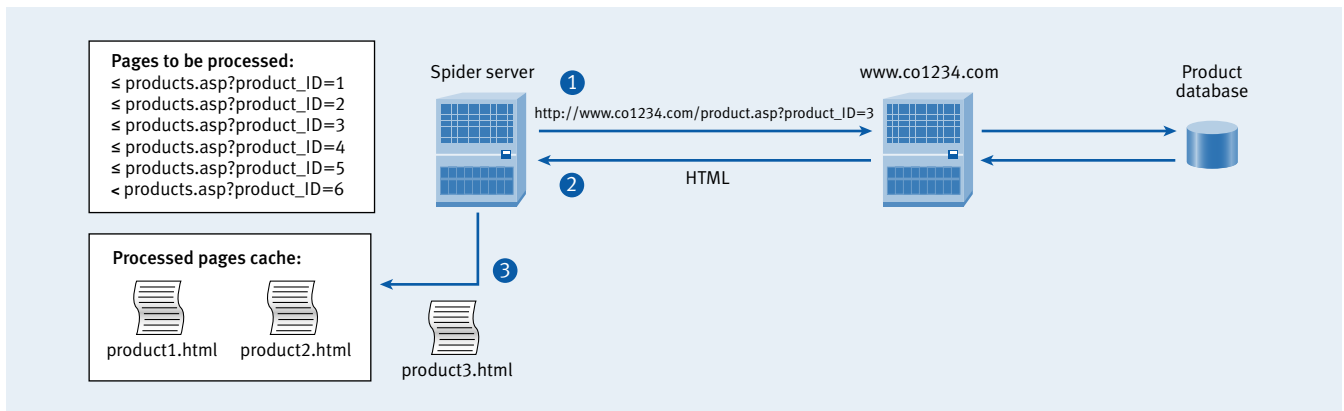


Figure 3. The spider process

can now be served by any inexpensive HTTP server instead of by a costly application server with a database back end. Page pre-rendering is especially effective for Web sites driven by read-only data, such as product catalogs or knowledge bases, because the data does not change frequently.

An administrator can schedule a nightly run of the spider process to produce all Web pages. After the process has completed, the administrator manually replicates the pre-rendered pages to the edge clusters where they are immediately available for users to download. The manual replication is one drawback of this method, especially for sites with a large amount of content and a large number of globally distributed servers (see section “Managing content distribution”).

Caching on the edge

Caching appliances and software can eliminate the need for a spider process and manual content replication. A caching appliance resides between the client and the Web servers. It populates its cache over time by intercepting client requests and retrieving pages on the client's behalf only once from a central Web server. The caching appliance serves subsequent requests for the same information

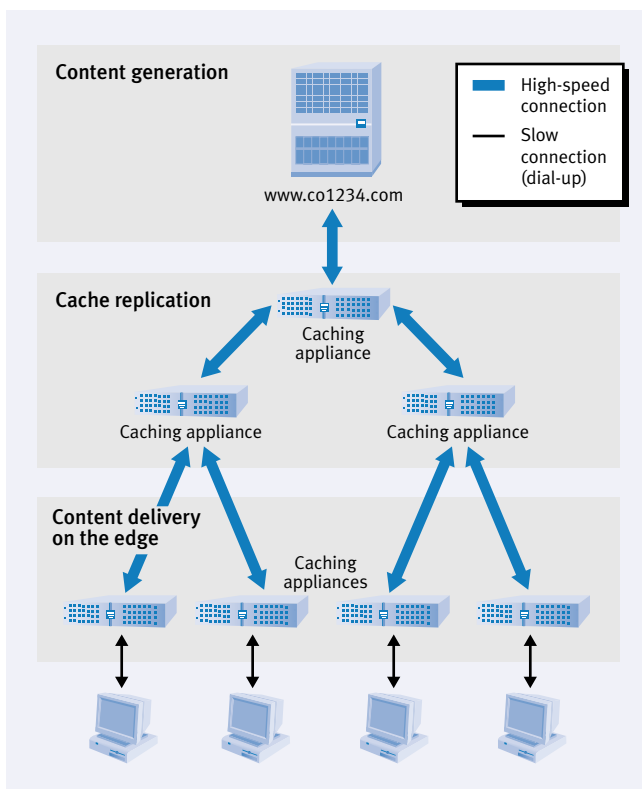


Figure 4. Distributed caching architecture

In a distributed approach, regionally relevant content is delivered from a location close to the user as quickly as possible.

directly out of its cache, eliminating the overhead of a round-trip request to the Web server. This process tremendously reduces the download time for frequently visited pages.

Advanced caching architectures allow system administrators to automatically distribute cache information to multiple destinations all the way to the edge of the network. These architectures implement several layers of caching appliances that are linked with high-speed network connections, as shown in Figure 4.

To simplify their Internet infrastructures, many companies choose to outsource caching to third-party, content-delivery service providers. These service providers offer globally distributed caching and content-routing frameworks, which comprise hundreds of servers located in popular local markets and interconnected with high-speed links. Instead of implementing its own caching hardware, a company simply routes its content to the edge of the Internet using a service provider's speed-enhanced network and caching infrastructure.

Caching on the edge of the network offers cost savings associated with the caching of dynamic content. Through a distributed caching infrastructure, even the most remote user can very quickly access content that was once dynamically created in a central location. Application and database servers have less work to do, which saves precious processing time as well as network bandwidth. Additionally, inexpensive caching appliances serve the content instead of the more expensive application server infrastructure. Finally, because caching appliances are less complicated than application servers, they require fewer staff hours for maintenance.

Page assembly on the edge

Modern sites implement highly dynamic and personalized pages, which are hard to cache. Although large portions of a page may be worth caching, specific portions are unique to each user and must be reprocessed every time the user accesses the page. A new technology called Edge Side Includes (ESI) accelerates dynamic Web-based applications by defining a simple markup language that describes cacheable and non-cacheable Web page components.

ESI allows developers to break a Web page into smaller fragments, such as data objects, streaming media files, stock quotes, and weather forecasts, which are aggregated, assembled, and delivered at the network edge. A developer creates an HTML template page by inserting ESI tags that point to cached HTML fragments. Each fragment, which contains HTML, ESI tags, text, and other objects, has its own access and cache

expiration settings. A caching appliance interprets the rules inside the ESI tags, fills the template with cached fragments, and delivers the assembled HTML page. ESI can insert cached content based on cookies, URL parameters, or user authentication.

Problems with distributed and cache-enabled Web sites

Administrators need to carefully consider the disadvantages of caching and distributed architectures. The speed and scalability of a Web site can compete with its supportability and manageability. For example, a centrally managed Web site is fairly easy to set up and maintain but scales at a higher cost. On the other hand, a cache-enabled, globally distributed design scales at a lower cost but is more complex to manage.

Although a distributed infrastructure is transparent to users and appears as a single server, an administrator still must maintain multiple machines that are potentially spread worldwide. To maintain these machines, an administrator must address several problem areas.

Managing content distribution

The content area of global Web sites is typically very large, often exceeding tens of thousands of documents in several languages. This large amount of content represents a problem in a distributed scenario because all files must be available in every regional data center or edge cluster.

Without caching appliances, an administrator must either manually duplicate all files or use configuration management tools to synchronize all dislocated servers. Sophisticated caching appliances can help automate the propagation of content from the source servers into the caches of the edge servers.

Harvesting log files and synchronizing time

Logging and reporting user access is no longer a trivial task in a globally distributed environment. Individual Web servers maintain their own local log files that contain client requests, system utilization data, and errors. To monitor a Web site's success and performance, an administrator must collect all these files at a central reporting facility, coordinate them based on their time stamps, and process them into a single file or reporting database. With potentially hundreds of servers located in multiple worldwide data centers, shipping and processing log files can be very complicated and time consuming. Problems, such as incorrect Web traffic reports, can occur if the internal clocks of the servers run out of sync.

A new technology called Edge Side Includes accelerates dynamic Web-based applications by defining a simple markup language that describes cacheable and non-cacheable Web page components.

Managing secure environments

Most enterprise Web sites have secure areas, which are fairly easy to configure in a single location using Secure Sockets Layer (SSL) encryption technology. Encryption key management can be tedious with distributed Web sites, especially those with a mix of servers and third-party appliances. Every time a key expires, an administrator must simultaneously update each server that uses the key.

Handling cache expiration

When incorrect content is rolled out, an administrator of a central Web site can usually roll back the changes and establish a healthy site in a matter of minutes. But in a distributed and cache-enabled environment, users may still see

the incorrect content after a rollback because the content is served from a cache. Mechanisms are available that allow an administrator to quickly flush all caches. This ability is particularly important in e-commerce sites, where outdated or false content may result in loss of revenue. However, cache expiration may not occur reliably in a global scenario, especially in remote locations with low-quality wide-area network (WAN) connections.

Separating content creation from content delivery

Caching technologies provide cost-effective ways to improve performance on a corporate Web site because they separate content creation from content delivery. This separation attempts to eliminate all disk I/O, network bandwidth, and processing cycles that the Web server spends to transfer content to the client. In an ideal situation, a Web application server uses its processing power to create content only, while more cost-effective systems, such as intelligent DNS routing and caching on the edge, deliver the actual information more quickly and from a location closer to the user. ☞

Marc Mulzer (marc_mulzer@dell.com) is a systems engineer in Server and Storage Systems Engineering, Web Technologies at Dell. Marc works with Web technologies to architect scalable, highly available, and cost-effective Web application solutions. Marc has a B.S. in Computer Science from the College of Advanced Vocational Studies in Mannheim, Germany, and is a Microsoft Certified Systems Engineer (MCSE).