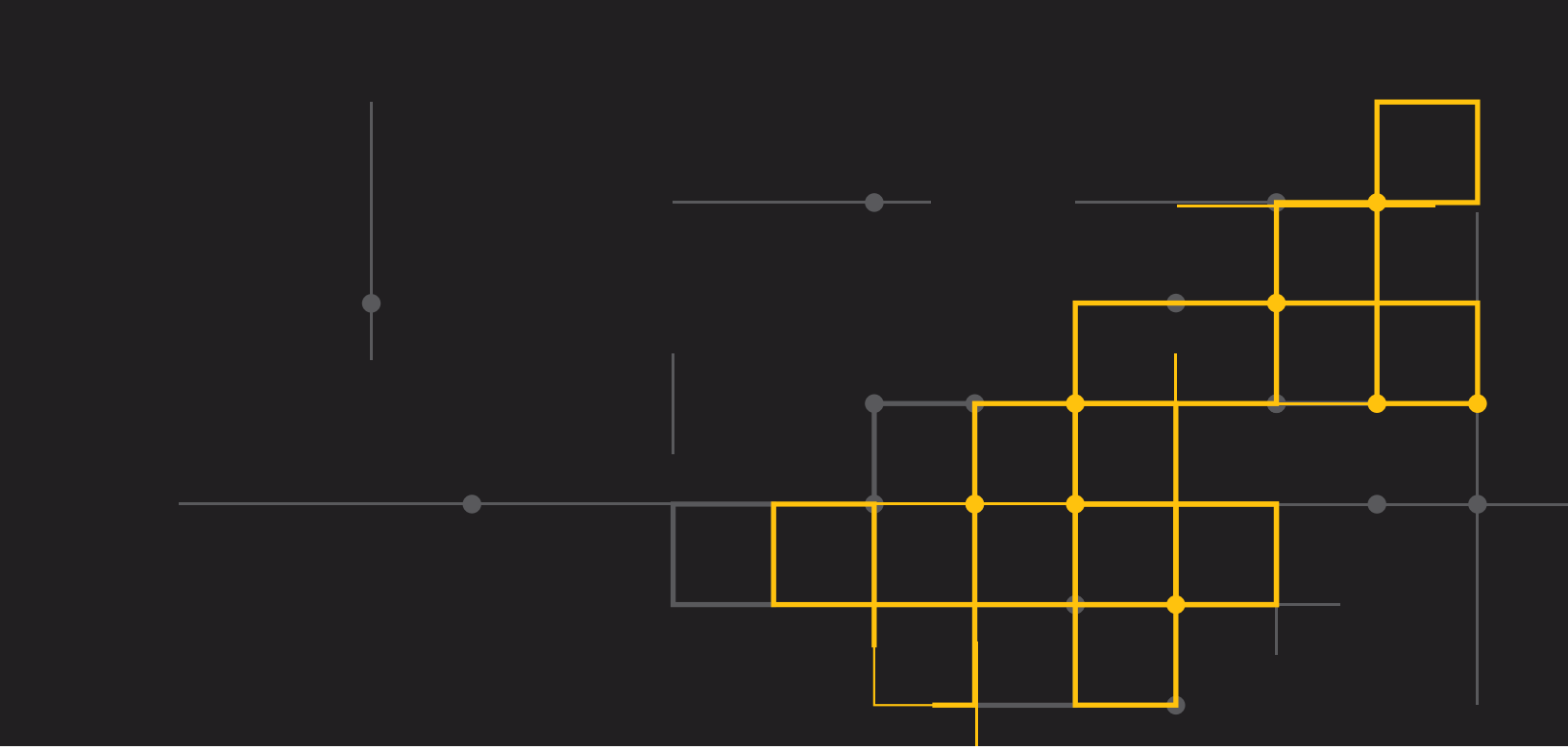

A Technical Review of Caching Technologies

Over the past 10 years, the use of applications to enable business processes has evolved drastically. What was once a nice-to-have is now a mainstream staple that exists at the core of business, education, and other operations across the globe. As application usage has increased, response time has become more and more important – and with good reason. Time wasted while waiting for data is money wasted and productivity lost. Factors affecting response time today include congested WAN pipes, inefficient/chatty protocols, and latency due to long distances between users and data. All these factors are exacerbated as applications and data resources are consolidated, centralized, or outsourced. In many cases, delays translate into frustrated users.

Symantec

Executive Summary

To address the performance problems associated with data retrieval, optimizing delivery of application traffic over the WAN has emerged as the strategic solution of choice for enterprises worldwide. Optimizing the amount of data that traverses the WAN, the strategy at the core of most WAN optimization solutions, is an excellent solution – but it isn't the only one. One of the largest factors affecting data retrieval response time is the distance between the data and the user. Optimizing WAN traffic clearly provides an enhancement, but a solution that does not require data be retrieved across the WAN at all is clearly the most ideal. This can be achieved with various forms of caching. Because response time improvement is crucial, a comprehensive WAN optimization solution must include not only the most popular WAN optimization technologies such as protocol optimization, bandwidth management, and compression, but also two forms of caching – object caching and byte caching. While byte caching is more prevalent in WAN optimization solutions, object caching is a key component currently lacking in most solutions. Because caching minimizes the transmission of data over the WAN and allows data to be served immediately, this technology directly improves response time and bandwidth utilization.

The goal of this white paper is to discuss the role of caching, describe in detail how object caching and byte caching works, and show how these caching technologies work together to deliver true performance differentiation.

Caching – A Tried and True Concept

Caching, while new to some, is actually a technology that has been employed in various areas of the computer and networking industry for quite some time. Although there are many different ways of implementing caching, caching technology is a fairly simple concept. Caching is the storing of frequently-used data in an easily accessible location so that time and resources are saved because data does not have to be retrieved from the original source. Because time and resources are always a premium in the computer and networking industry, caches exist everywhere and are used in all high performance systems. In fact, the CPU of every networking device (router, switch, PC) takes advantage of caching to speed memory access. Certainly, several hundred million PCs utilizing cache technology in the processor illustrates the importance of caching, even at the hardware level. Another very common cache, found on almost all PCs, is the Web browser cache. Internet Explorer and Firefox, the most widely used browsers, for example, have web caches for storing requested objects so that the same objects do not need to be retrieved multiple times from the Web server. This process is known as object caching.

Different Forms of Caching

Regardless of the implementation, the goal of all caching is the same – to avoid reacquiring data that has already been retrieved so that response time is improved and bandwidth utilization minimized. The caching technologies detailed here are designed to achieve that goal by caching data locally and serving it immediately when a request for the data is received. With this feature, significant performance enhancements result as described below.

Object Caching

Object caching has been around for many years and has traditionally been used to accelerate access to HTTP content. In addition to object caching HTTP content, some vendors have extended their object caching support to include HTTPS content, streaming media objects, FTP, and CIFS files. Occasionally, object caching is referred to as "proxy caching" since it is implemented using a proxy for the given protocol (e.g. HTTP, HTTPS, FTP, CIFS, or RTSP/RTP).

How object caching works

The mechanism of object caching is fairly straightforward and requires no configuration. The client sends a request for an object (file, document, image etc.) to a server and the request is inspected by the proxy that is in between the client and origin server. Upon receiving the request, the proxy checks to see if it has a copy of the requested object in its cache. If so, the proxy responds by sending the cached object to the client. Otherwise it sends the request to the server. If the proxy requests the object from the origin server, that object data is cached by the proxy so that it can fulfill future requests for the same object without retrieving it again from the origin server. The result of serving a cached object is that there is zero server-side bandwidth usage for the content and a vast improvement in response time for the end user. While the benefit of caching is clear and the mechanism of the proxy simple, it does pose several challenges – most significantly, content freshness and storage.

Content Freshness

To avoid data integrity issues, cached objects must be kept fresh. Because the content on servers changes, an object cache must keep its temporary store of content up to date. There are two approaches to maintaining content correctness – either some sort of periodic freshness check, or always verifying the content with the server before serving the cached object. Traditionally, for an HTTP proxy to deliver content to the end user with confidence that the data is fresh, it must send a "refresh check" to the origin server. However, to serve the content quickly, it must not wait until a user requests the content before it performs this refreshing activity. Due to the way that Web pages are constructed (many embedded objects often linked to many geographically disperse servers), if the refresh checks are performed only when the user requests the content, the user will endure the same round-trip delays that cause the data retrieval to be slow in the first place, as each Web object is validated – application response time will not be substantially improved. Intelligent object caching technology should be able to avoid stale data without negatively impacting the network with unnecessary refresh checks. One vendor's implementation of object caching uses an intelligent adaptive refresh algorithm to guarantee the freshness of the content without adversely affecting performance with refresh checks. For CIFS files services, the object caching approach is to always verify with the server before returning a cached CIFS object. This is necessary to validate that the file is unchanged and to check user permission. The overhead involved is negligible, however, compared to the overall inefficiency of the CIFS protocol. A single round-trip to verify freshness and permissions is considerably more efficient than transmitting entire files.

Storage

Another challenge often associated with object caching is storage, since storing millions of application objects on a general purpose file system may not be efficient and may result in added latency due to disk reads. The method of storing objects on disk, therefore, becomes critical for achieving both high performance and high scalability. It determines (1) how quickly a cached object can be accessed when a client requests it, (2) how rapidly new objects can be acquired and stored on disk, and (3) the rate at which client requests can be serviced per disk drive. Typical implementations of object caching use a file system, which can run poorly when full. A faster implementation of object caching uses an object storage system that is truly an object cache. Instead of using a directory or commonly used file system, object access done through a hashed table in RAM ensures that any object can be obtained in a single disk read. Unlike the file system, which has negative performance impact when full, an object cache achieves its highest performance when full. In an ideal object cache, old, seldom-used objects are continually removed to make room for new incoming objects. Disk layout and replacement algorithms should facilitate this process to optimize the speed of writing new objects to disk.

When is object caching most useful?

Object caching is a very beneficial technology for the following types of content:

- Content that does not change very often such as images, logos and some documents.
- Content that can be pre-populated on appliances before users try to access it (eLearning, multimedia applications).
- Files that do not change often and need to be accessed by multiple users.

Byte Caching

While object caching is the most effective caching method from a response time and bandwidth utilization perspective, it has three fairly significant limitations. The most obvious limitation is that it is limited to specific protocols, as described earlier in this document. Another is that even if one byte of an object changes, the entire object must be retrieved again as the object has changed. Object caching is also limited to files requested by the client, and is not used when a client is saving, or posting, a file to a server. To overcome these shortcomings, the natural evolution of caching has been to cache repetitive portions of an object, known as byte caching. Byte caching is a protocol, port, and IP address independent bidirectional caching technology that functions at the TCP layer by looking for common sequences of data. Byte caching is designed to work in a mesh, hierarchical, or any arbitrary network design and imposes no limitations on the design of corporate networks.

How byte caching works

Byte caching is a feature that requires a symmetric deployment between two endpoints. As such, byte caching is a common enhancement for WAN optimization networks, where one or more appliances are deployed at branch offices as well as at a core or datacenter. As appliances on both sides of the WAN communicate, they maintain a cache of all TCP traffic being sent and received over the WAN. Each time data needs to be sent, it is scanned for

duplicate segments in the cache. If any duplicates are found, the duplicate data is removed from the byte sequence, and tokens are inserted in its place. When the other appliance receives the data, the tokens are removed from the byte sequence and the original data is reinserted. Byte caching allows the amount of data traversing the WAN to be as small as possible with no loss of the original content.

Why is byte caching useful?

Byte caching is a very useful caching technology because up to 90% of WAN traffic is repetitive. The reason for this is that most enterprise traffic is composed of the following:

- Web application traffic – Users at the branch typically use the same or similar Web applications. Each interaction with these applications results in WAN traffic that is marginally different than the traffic for previous interactions resulting in the resending of the same bytes.
- File server traffic – File traffic makes several round-trips over the WAN while the user is working on a file. The typical office applications save copies of the file at small intervals resending only slightly modified versions of the same document over the WAN link. Because byte caching is bidirectional – files requested or saved take advantage of byte caching.
- E-mail traffic – Enterprise e-mails are frequently addressed to several people. For each recipient in the branch, a copy of the email travels over the WAN. In addition, replies to emails contain repetitive data resulting in further redundant traffic over the WAN.

By eliminating redundant traffic, WAN Optimization solutions that implement byte caching can effectively increase WAN capacity up to 100x depending on the application. Byte caching works at the transport layer and does not depend on any knowledge of the application to cache the traffic. This allows byte caching to handle traffic for all applications. Since byte caching works at the transport layer, its deployment does not require any changes to the applications themselves or to application configuration. This satisfies the requirement that it should be transparent to applications and users. Truly flexible implementations of byte caching should provide the capability to associate the data with specific applications and users, providing control over what data gets cached and what gets blocked. This allows the administrator to create and set policy around their network usage and access.

While byte caching accelerates all TCP traffic, it further accelerates specific application protocols that make use of an object cache including:

- Web – HTTP, HTTPS (SSL)
- Streaming media – Progressive download
- Email – MAPI
- File services – CIFS

How Object Caching and Byte Caching Complement Each Other

Both object and byte caching technologies have their strong points and both are suitable in different situations; however, together they provide an even more powerful caching solution. Together, object caching and byte caching provide dramatic and unmatched acceleration of enterprise applications and reduction of WAN bandwidth, forming the backbone of a solution for a wide set of WAN challenges that face today's enterprises. In addition to providing coverage for a wide set of applications, synergy between the two caching approaches makes the overall system better than the sum of its parts. Object caching results in immediate retrieval of objects at LAN speed, with zero bandwidth utilization, while byte caching minimizes the amount of data transmitted across the WAN for data that must be retrieved. For example, consider a company logo used on all company documents. Byte caching would identify this common aspect of essentially different files and prevent that data from being transmitted over the WAN, even when users request a document that had never been seen before.

One commonly overlooked advantage of object caching is server offload. While a solution that implements only byte caching reduces WAN utilization, it generates tremendous and needless server overload because each request must still be retrieved from the server. The synergistic combination of object and byte caching is both WAN and server friendly. Further, the byte caching functionality allows the object caching subsystem to be aggressive with adaptive refresh algorithms that keep local content fresh, improving the response latency in the branch office. Similarly, the existence of fresh content in the object cache results in fewer TCP round-trips by the byte caching subsystem since the request can be served locally.

Solution Checklist

When choosing a caching solution, remember that there are a number of key features that make up an effective and comprehensive solution. A checklist of the key features discussed in this white paper is provided below for reference:

- The ability to perform object and byte caching
- Ability to cache HTTP, HTTPS, CIFS, FTP, and streaming objects
- Use of a true object cache and not a file system or directory structure
- Policy-based control over the caching functionality
- Ability to dynamically re fresh data
- Minimal to no configuration to perform caching

Conclusion

Caching is a widely-used, industry-recognized technology for improving user response time and minimizing bandwidth utilization. Today, this requirement is essential to keep up with the exponential increase in enterprise WAN traffic caused by server consolidation, centralization, and outsourcing. A WAN optimization solution is simply incomplete unless it includes both object caching and byte caching; these technologies are crucial components in achieving significant response time improvement, bandwidth savings, and, where needed, server offload. Symantec Systems recognizes the importance of caching and has been implementing object caching in its proxy appliances since it shipped its first product in 1998. As caching technology has advanced, Symantec has expanded its caching expertise to also include byte caching, resulting in a WAN optimization solution that provides all of the crucial components required for a total WAN Optimization solution. Because no application changes are required to benefit from caching, and because virtually no configuration is required, Symantec's caching technologies provide the components necessary to deliver a WAN optimization solution that can be easily deployed in a variety of corporate network architectures, providing significant enhancements to both end user response time and bandwidth utilization.

About Symantec

Symantec Corporation World Headquarters

350 Ellis Street Mountain View, CA 94043 USA | +1 (650) 527 8000 | 1 (800) 721 3934 | www.symantec.com