Silver Peak | white paper

NOT YOUR FATHER'S CACHE

EXPLORING THE FUNDAMENTAL DIFFERENCES BETWEEN NETWORK MEMORY™ AND APPLICATION CACHING

white paper

NOT YOUR FATHER'S CACHE

EXPLORING THE FUNDAMENTAL DIFFERENCES BETWEEN NETWORK MEMORY™ AND APPLICATION CACHING

BACKGROUND

Enterprises have long been battling the inherent performance limitations of Wide Area Networks (WANs) when delivering applications to remote offices. Caching emerged in the 1990's as a potential way of addressing this problem, accelerating the performance of specific applications, such as web services, while reducing overall WAN traffic. While caching achieved reasonable success for a few short years, the market for these point products ultimately subsided as a result of several operational and functional limitations.

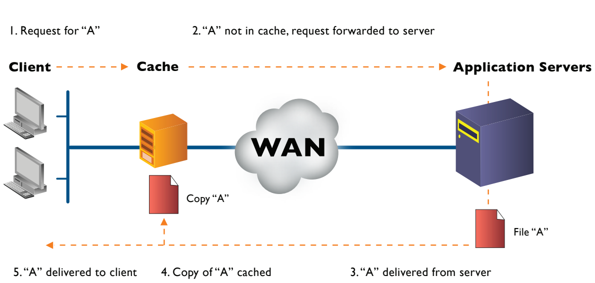
For one, enterprises require an application acceleration solution that spans all types of traffic; caches are application-specific, requiring separate devices for web, file, email, and other applications. In addition, enterprises require a non-intrusive solution that is easy to manage; caches require special configuration of clients so that they can act as a "proxy" device. Lastly, enterprises require 100% data coherency when supporting business-critical applications; caches store and deliver local "copies" of application objects, which can lead to the delivery of stale or inconsistent information.

Network Memory™ is a patented technology from Silver Peak that addresses these enterprise requirements. It is similar to caching in that it monitors traffic and stores information locally for future delivery. However, Network Memory is different from caching as it is a completely non-intrusive, application-independent solution that incorporates networking technology with innovations in data storage and data pattern matching. The result is a complete solution for improved application delivery across a distributed enterprise environment.

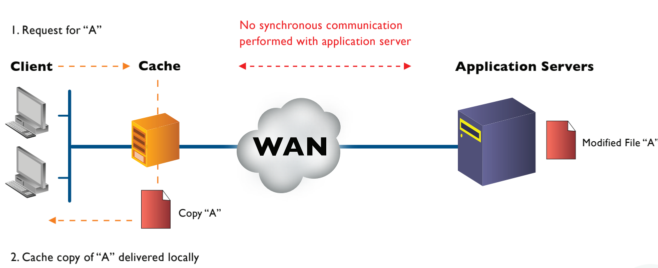
BEHIND THE SCENES: CACHING IN ACTION

A cache sits between clients and application servers (sometimes on both ends of the WAN link, but more often just at the client location). When a client requests an object, such as a file or .gif image, the cache intercepts the request. If the object has previously been stored in the cache, the cache locally returns the object directly, eliminating the backend request to the server and the transfer of data over the WAN. If the object is not already in the cache, the cache "proxies" a request for the object on behalf of the client. This is an important attribute of caching: the client is interfacing with a proxy, not directly with the server. As a result, caching changes the overall communications flow between client and server devices.

Scenario 1: ☐
If the cache has not seen the exact object previously, the request is forwarded to the application server. A new copy is stored locally in the cache for the future reference.



Scenario 2: ☐
If the cache can perform an exact match on a previously seen object, it will deliver a copy of the object locally. The request is never sent to the application server, which can lead to potential coherency issues.



Caching can provide a reasonable boost in performance. However, there are significant limitations when using a cache across a distributed enterprise, which include:

- **Caches are application-specific.** Caches only accelerate the performance of a specific application. For example, Wide Area File Services (WAFS) accelerate file transfer and file storage/access; web caches accelerate HTTP traffic. As most enterprises require increased performance across a broad range of applications, it can be costly to implement a separate caching solution for each application.

- **Coherency issues.** When clients are retrieving information stored in a local cache, it is easy for this information to get out of synch with information stored in the original application server. Or, when multiple caches are used to deliver the "same" objects, these devices can end up delivering different information. This is because a cache retrieves and stores a copy of original server data, and delivers this copy to clients upon request. While the cache is responsible for making requests to the server over time to verify that the stored copy is in synch with the "golden" copy that is on the actual origin server, this process does not happen in real-time — that is, when the client is actually requesting data.

  Larger enterprises (with many caches), are especially susceptible to these types of coherency issues. While Web caches attempt to address the issue by assigning an expiry time to particular types of data, this is a rough way of ensuring that data remains accurate and up to date.

- **Requires exact matches.** A cache only makes a match when the exact same object it is storing is requested. For example, web caches deliver content locally only when a requested URL matches a stored URL; file caches deliver content locally when a requested file name matches a stored file name. In both examples, the cache has no ability to detect "similar" information — that is, the same content under a different file name or at a different URL. This information must be re-requested from the server, which limits the overall effectiveness of caching.

- **Management complexity.** Because a cache interfaces directly to clients, it is responsible for handling authentication and authorization, read/write, file locking, and other privileges. As a result, caching introduces a set of intelligent devices into a distributed network that requires management and proper security precautions, which increases IT support requirements.

- **Intrusive configuration.** Many cache environments require modifications to clients so that they point to a cache proxy server instead of to the original application server. This can be difficult to deploy and manage, particularly in large environments.

- **Designed primarily for static content.** Some solutions, such as web caches, are dependent on having cache tags set properly on web objects. Unfortunately, most web application developers do not take the time to set these tags. As a result, web caches treat most objects as being non-cacheable, reducing cache efficiency.

NETWORK MEMORY: SIMILAR, YET SO DIFFERENT

Network memory is a key component of Silver Peak's Virtual Acceleration Open Architecture (VXOA). As such, all Silver Peak appliances support Network Memory.

Silver Peak devices (physical or virtual appliances) are inserted in each enterprise location and use Network Memory to inspect all incoming and outgoing WAN traffic. Each appliance stores a single local "instance" of information in its application-independent data store. The local instance at each location is transparently populated based on day-to-day usage with the subset of the enterprises working data set most relevant to that location.

Each Silver Peak appliance examines all outbound WAN traffic, analyzing actual traffic patterns (as compared to application-level objects). Each appliance searches its local instance to determine whether the real-time traffic stream matches traffic patterns previously stored using Network Memory. If a match exists, a short reference pointer is sent to the remote appliance, instructing it to deliver the traffic pattern from its local instance. Repetitive data is never sent across the WAN, saving bandwidth and enabling LAN–like application performance.

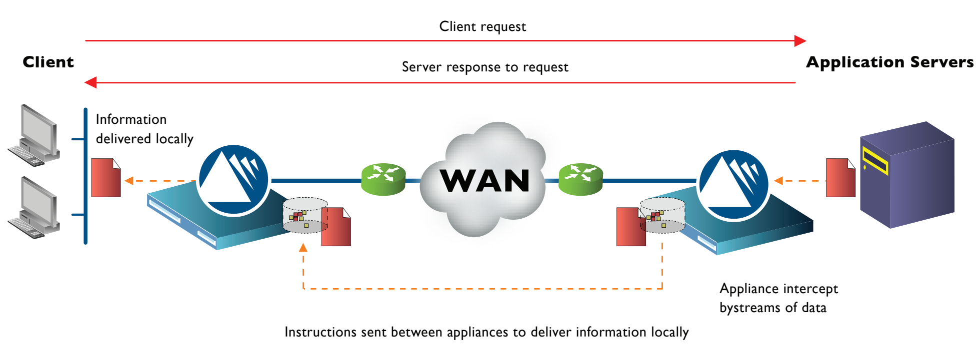
On the surface, Network Memory resembles traditional caching. However, there are fundamental differences between the two, which include:

- **Application Transparency.** As the name suggests, Silver Peak "Network Memory" works at the network-layer of the OSI stack. In contrast, caches operate at the application level, intercepting client requests and server responses and storing application-specific objects. As a result, only Silver Peak Network Memory is able to provide performance improvements across all enterprise applications, regardless of the transport mechanism (TCP, UDP, etc.). For example, in addition to file, email, and web services, which are traditional applications serviced by caching solutions, Network Memory improves performance for applications such as data replication, VM mobility, CRM, ERP, backup, video, voice, real-time data,and a plethora of other enterprise applications.

- **Matching traffic patterns vs. application-level objects.** Network Memory recognizes traffic byte streams, observing pattern matches as opposed to object references. In this way, Network Memory can detect when the same information is sent using different applications. For example, if a PowerPoint file is emailed to a remote user and then downloaded from the Intranet by another user in the same office, the second user's file download will benefit from the information that was stored in the Silver Peak appliance during the first user's email transfer.

  Network Memory detects when modifications are made to existing data and transmits only the delta across the WAN. For example, if one slide in the above Powerpoint example is changed, only the information in that slide is transferred across the WAN, where it is combined with the original data at the far end prior to being delivered to the client or server. Caching, on the other hand, would require the entire modified file to be resent in this scenario. Since Network Memory detects common byte patterns, it can detect "similar" information in addition to identical information. In this respect, Silver Peak better utilizes WAN bandwidth and increases application performance.

Network Memory reduces traffic over the WAN and provides LAN-like performance, without altering client/server communications in any way.



- **Seamless Integration.** A major difference between Network Memory and caching is that Network Memory does not alter the communication mechanism between clients and servers — it simply improves the method by which these devices deliver data. All requests for information from a server are delivered to the application server itself. Furthermore, the server's exact response, not an old cached version, is delivered to the user. This means that Silver Peak appliances can be seamlessly inserted between existing clients and servers with no special application-layer configuration. In contrast, caches act as an intermediate proxy between clients and servers, requiring intrusive reconfiguration and disrupting end-to-end communications.

- **100% coherency.** By preserving communication mechanisms between clients and servers, Network Memory ensures 100% coherency across an enterprise. Instead of delivering copies of information, Silver Peak appliances deliver the actual information as it is sent between a client and server. By examining all traffic in real-time, Silver Peak appliances are always dealing with up-to-date information, eliminating the possibility of stale content delivery. All application locking semantics and file/record locking capabilities are still performed by the native server, enabling the Silver Peak solution to ensure 100% coherency.

- **Better security, compliance, and management.** The Silver Peak solution does not replicate the access control policy mechanisms already present in existing servers. These can be centrally maintained within the servers themselves, eliminating potential security risks and avoiding unnecessary management headaches that come with replicating and maintaining access privileges across multiple devices. Hardware-based AES encryption ensures that any information stored on Silver Peak appliances (or passed between them via IPSec) is completely secure from unauthorized access.

CONCLUSION: CASHING IN WITH NETWORK MEMORY

Network Memory provides improved performance over caching, without the inherent management and operational limitations. By delivering a network-layer solution that leverages advanced byte-stream fingerprinting technology and high capacity local data stores, Network Memory provides order-of-magnitude performance gains across almost any enterprise application. In addition, by preserving existing communication mechanisms between client and server, the Silver Peak solution is easy to install and operate, and it is completely non-intrusive to ongoing operations.

Silver Peak localizes information, while centralizing the control of applications servers and storage. In doing so, Network Memory is an indispensable tool for enterprises looking to cash in on the cost and management savings associated with branch office infrastructure centralization.

888.598.7325 | www.silver-peak.com