Performance Summary

- Using Blue Coat MACH5 technology together with SnapMirror enables organizations to complete backups within the assigned window and dramatically improve recovery time in a disaster
- Differential Mirror updates complete 80% faster – 1/5 the original time
- Full Mirror copies and restores are accelerated by over 40x
- Protocol optimization improves SnapMirror speed, byte caching and compression reduce bandwidth usage, and Bandwidth Management ensures sufficient resources to complete the backup on time

Test Scenario

SnapMirror testing was conducted using a WAN connection of 45Mbit/sec with 200ms round trip time and a dataset of 22GB. This represents typical network performance for geographically distributed datacenters. Tests included initial data replication between the two sites, incremental backup after additions and changes were made to the initial dataset, and full restoration from backup.

A "miss" is a test where none of the data has traversed the WAN through a ProxySG before.

A "hit" is s test where some (an incremental backup) or all (a full restoration) of the data has been seen by the ProxySG before.

ProxySG Appliances Optimize and Accelerate NetApp SnapMirror

Organizations collect vast amounts of information, and storing it all is only the first challenge in comprehensive data management. NetApp's SnapMirror application is a popular choice to automate the backup, duplication and recovery of data to ensure availability in the event of a disaster or user error. Quite often, that involves replicating large amounts of data across a wide area network.

WAN links, however, have higher latency and much less bandwidth than either a LAN or Fibre Channel network. Latency severely impacts the performance of protocols SnapMirror relies on for WAN communication, and the amount of data often dwarfs the bandwidth most organizations can afford. Even using SnapMirror's advanced differential backup techniques, the amount of data blocks touched on an array can still be significant, and much larger than can be moved across the WAN during the backup window. If differential backups take hours, and a mirror initialization takes days or longer, how can organizations effectively maintain backups? Or worse, how can they expect to restore them quickly in a real disaster?

Blue Coat Optimized SnapMirror

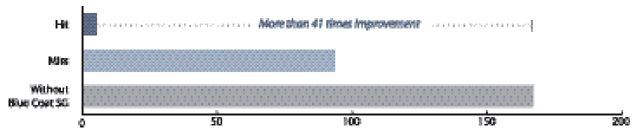
ProxySG's byte caching and compression, combined with TCP enhancements and bandwidth management, significantly improve SnapMirror data transfers. Mirror initialization triggers a massive amount of network traffic which is highly repetitive and very compressible. Differential Mirror updates also contain much repetition; although the data is new in each individual file block, the byte patterns between files are very redundant and respond well to compression and caching. Initialization, recovery, and update times are typically decreased by 90% with customer deployments.

The Blue Coat solution goes beyond just compression and protocol optimization, however, by providing bandwidth management to control access to network resources. Although not shown directly in this testing, ProxySG appliances can prioritize traffic based on user, application, time of day and content, ensuring that SnapMirror gets the bandwidth it needs both during the backup window, and in the event of an emergency restore.

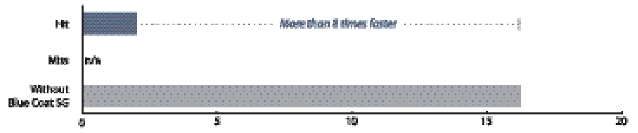
Measuring the Improvement

Tests conducted in labs and within customers environments show that ProxySG appliances with MACH5 significantly improve the performance of SnapMirror in real world scenarios. Using Blue Coat MACH5 optimization technology, the time needed to finish a Mirror backup operation was reduced by 58% on the first (cold) pass, and subsequent backups by almost 90%. Bandwidth used to complete the backup or restore was also reduced by 85%. Such a dramatic improvement makes it possible for our customers to complete their backups within the nightly window again. For a new mirror initialization, Blue Coat MACH5 reduced the overall time to clone by 41x; convenient for IT when creating mirrors, but that performance increase would be critical in the event of an actual restore due to data loss or disaster.

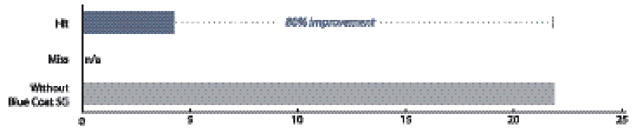
Full Mirror Initialization / Restore Time (in minutes)



Daily Differential Backup Transfer Time (in minutes)



Daily Differential Backup Bandwidth Cost (in gigabytes)



Bandwidth Management/Traffic Shaping

This technique assigns a priority to a particular type of application. This priority has an effect both on the order the traffic is sent in, and in the amount of guaranteed bandwidth the application is allocated, regardless of other traffic on the network. This technique ensures that the network is available for the highest priority traffic. Likewise, less important applications can be throttled back and assigned limited bandwidth to help ease network congestion.

Protocol Optimization

Protocol optimization takes protocols that are inefficient over the WAN (e.g., CIFS, MAPI, HTTP, TCP, HTTPS) and makes them more efficient – typically by converting a time-consuming serial communication process into a more efficient parallel process where many communication tasks are handled simultaneously. There are a variety of other optimization techniques, depending on the protocol (e.g., TCP session reuse). While protocol optimization does not reduce the amount of bandwidth an application consumes, it can greatly accelerate delivery of applications and reduce latency in the process.

Byte Caching

Byte caching is as it sounds – a low-level cache of small, sub-application-object bits of information. Byte caching observes repetitive patterns in application traffic, symbolizes those patterns with a token, and sends the token in lieu of the bulky traffic. These tokens are typically only a byte or two, but symbolize blocks of data as large as 64KB. Byte caching is typically not application-specific, and operates at a lower level, optimizing all TCP traffic.

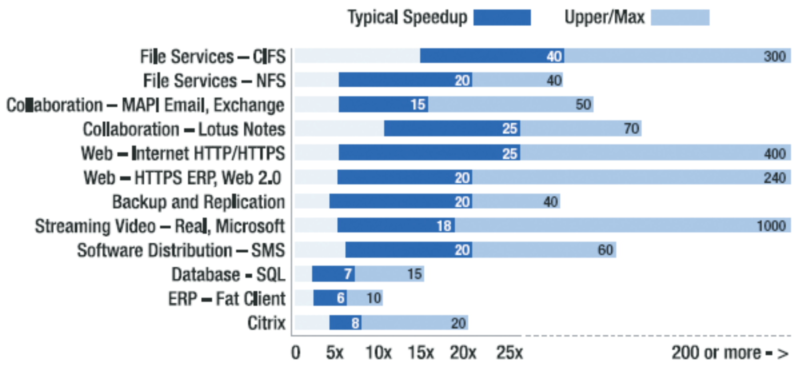
Object Caching

Object caching is very different than byte caching – it is protocol/application specific, and is an all-or-nothing affair. If the cache contains the object, the user is immediately served the object from a local store – virtually eliminating latency and WAN bandwidth consumption. If the cache does not contain the object (or contains an outdated version of the object), then for that particular transaction, a new object must be reloaded into cache and the performance gains are realized the next time the object is requested.

Compression

Compression technology uses a variety of common algorithms to remove extraneous/predictable information from the traffic before it is transmitted. The information is reconstituted at the destination based on the same algorithms. It is effective because performance gains are immediate – the first time something goes through is just as fast as the second. This technique, when applied with byte and object caching as discussed above, helps optimize bandwidth savings and performance.

With MACH5, all of these techniques work together to optimize application delivery to remote locations. For example, if the object cache contains an outdated copy of a document, the byte caching capability has patterns and tokens that require only the tokens, plus the changes to be sent. What little is sent is then compressed, and protocol optimized (reducing bandwidth consumed and latency/round trips). All of this is prioritized according the enterprise's preferences, using bandwidth management, such that the important applications get through first with the bandwidth they need.


Typical Speedup
Upper/Max
File Services – CIFS 40 300
File Services – NFS 20 40
Collaboration – MAPI Email, Exchange 15 50
Collaboration – Lotus Notes 25 70
Web – Internet HTTP/HTTPS 25 400
Web – HTTPS ERP, Web 2.0 20 240
Backup and Replication 20 40
Streaming Video – Real, Microsoft 18 1000
Software Distribution – SMS 20 60
Database - SQL 7 15
ERP – Fat Client 6 10
Citrix 8 20
0 5x 10x 15x 20x 25x
200 or more - >