WHITE PAPER

APPEX NETWORKS

HyperCompression

Low-Latency Deep Compressiom

Data Compression and Byte Cache

Compression is one of the cornerstones of the WAN Optimization technology. It squeezes out the redundancies in the outgoing data before they hit the WAN, and restores the original data from the incoming compressed traffic and forwards them over to the LAN. As a result, the effective data throughput is elevated by a factor close to the data compression ratio, while the whole process is transparent to the end users in the LANs.

The term “compression” here is a general concept, which can be further categorized into Data Compression and Byte Cache:

- **Data Compression**: The compression algorithms are based on the general lossless data/file compressions. The compressed data are self-sufficient, which means that they can be decompressed and restore the original data without any other extra information.
- **Byte Cache**: The inbound and/or outbound data is cached in a network device. Before the new outgoing data are sent to the WAN, they are looked up in the cache. If a match is found, the entire matching data block can be replaced with a short token, given the peer also has a cache in sync with the local end. This requires the local and peer Byte Cache devices to have the same cache content. Otherwise the peer will not be able to map the tokens back to the original data.

Both these compression types require symmetric (bilateral) deployment. It is because that after the sender has compressed the flows, the receiver side must be able to identify and decompress the compressed data before forwarding them over to the LAN in order for the compressions to work transparently.

AppEx HyperCompression

AppEx HyperCompression supports both Data Compression and Byte Cache. On top of that, it is able to combine them together to achieve even higher compression ratio. AppEx HyperCompression distinguishes itself from other WAN Optimization compression solutions in its ability to automatically discover the peers and negotiate the compression methods, with all the discovery, negotiation and compression messages running inline the regular network flows that the end users have initiated. All the other WAN Optimization vendors that we know of implement their compressions on top of their own proprietary protocols. And most of them require manual configurations to pair up the devices deployed for the compression to work successfully. With AppEx HyperCompression’s auto discovery and inline messaging, it is virtually configuration-free and maintenance-free. At the same time, it is the least intrusive of all to the existing network infrastructure, offering the best firewall penetration. In the most majority of scenarios the existing firewall policies don’t have to be reconfigured for

AppEx HyperCompression, while the other WAN Optimization compression products almost certainly require.

As shown in the diagram below, AppEx HyperCompression comprises the following modules:

- **Auto-Discovery Auto-Negotiation (ADAN)**: injects and intercepts packets to and from the flows to perform Auto-Discovery and Auto-Negotiation. Once the peer is discovered as a compression device and the methods are negotiated, it sets up information in the flow so that the Flow Engine will divert it to the Buffer Manager.
- **Buffer Manager**: the major functionality is to normalize the buffer format between the Flow Engine, ZLIB and Byte Cache so that ZLIB and Byte Cache can be applied to a flow individually or concatenated. It also manages the synchronizations in feeding data to the Byte Cache module under the multi-engine configuration.
- **Compression**: consists of ZLIB and Byte Cache. Each one can be applied to a flow by itself, or combined together to achieve maximum compression ratio.

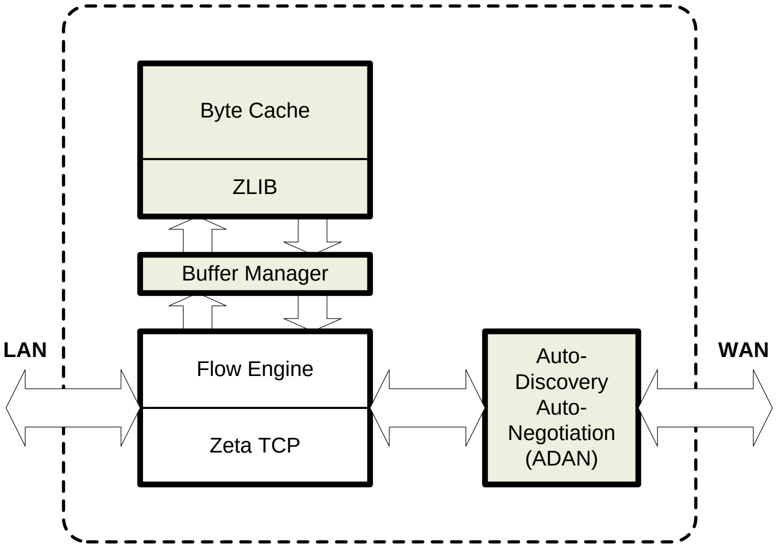

Figure 1: AppEx HyperCompression

The Buffer Manager and ADAN work directly with the Flow Engine so they are able to manipulate the flows. Therefore AppEx HyperCompression is able to inline all the messages directly into the flows initiated by the end users.

HyperCompression technology requires symmetric (also called bilateral) deployment, as illustrated in the following diagram.

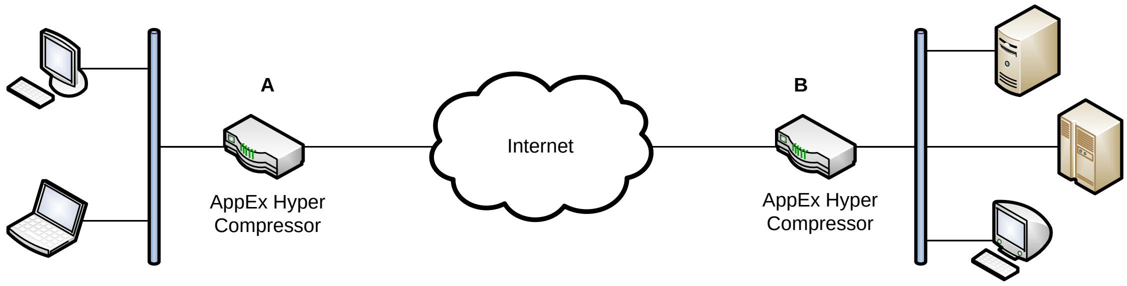

Figure 2: AppEx HyperCompression Requires Symmetric Deployment

Data Compression

AppEx HyperCompression currently uses ZLIB to perform data compression, which is a variation of LZ algorithm. ZLIB is very widely adopted, stable software which has decent performance for the data flow compression.
Thanks to the Auto-Negotiation, it will be easy to introduce new compression algorithms with backward compatibility.

Byte Cache

Byte Cache is an advanced compression technology which requires both parties of the communication peers to synchronize the information exchanged. We will use the deployment shown in Figure 2 above as an example. Assume A and B have synchronized history of the data exchanged between the two sites. Also assume an end user from side A is sending data to a server in side B. When A receives the outgoing data, it first performs a dictionary lookup to find out if the outbound data block can be located in its history. If a match is found, the entire data block will be replaced by a token of 2 to 7 bytes long. Then this token will be transmitted to B instead of the data block. Since B already has the same history data as A does, upon receiving the token, it will be able to map it back to the original data block and forward the data to the LAN side destined to the server.
While performing tokenization on A and restoration on B, they will also write the data block into their history cache respectively so that future data exchanges may benefit from the accumulated history data. The following diagram briefly demonstrates some details of the Byte Cache algorithms for both the compression and the decompression sides.

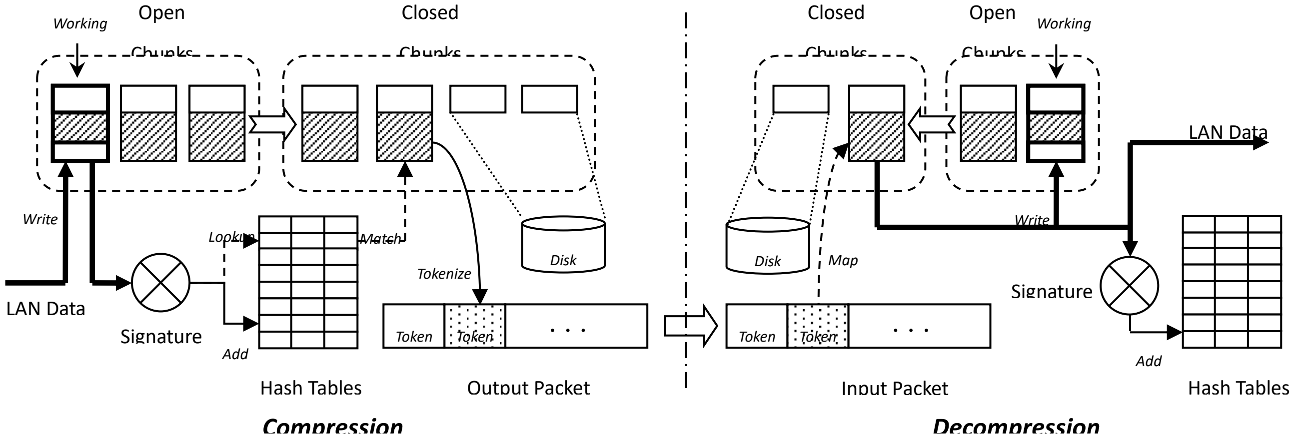

Figure 3: Byte Cache Operations

Byte Cache compression is not targeted to eliminate the redundant information entropy within the data blocks themselves. Rather it is meant to reduce the redundancy between data blocks over time from a higher level. Therefore it is capable of compress contents that are not compressible by the general data compression, such as compressed movies, videos, music and encrypted files, etc. This is a very attractive feature for content sharing in a network environment.

AppEx HyperCompression's Byte Cache has a number of advanced algorithms to achieve the best overall network performance.

Maximize History Data Utilization

Better history data utilization means better compression ratio. When looking up history data, AppEx HyperCompression applies a sliding-window algorithm that calculates on every offset of the data block to be compressed. This algorithm allows data blocks that differ only by an offset will be matched and compressed.

AppEx HyperCompression writes both the incoming and outgoing data blocks into the history. This allows the data histories on both directions to be used to tokenize future data, increasing the probabilities of finding a match.

AppEx HyperCompression also allows the history data to be used as soon as possible. As long as the newly added data block is confirmed by the peer, or it can be inferred from the other signals that the peer must have received the block, then it will be immediately made available for lookup as history. This enables the scenario that if two flows are transmitting similar data almost simultaneously, then they may be benefit from each other's transmissions.

Latency Control

The trade-off of compression ratio is the latency. In Byte Cache we wish to achieve as high compression ratio as possible. However under quite some circumstances that also introduces extra delays. Extra delays have an adverse impact on the effective

data rate. At the same time they introduce bigger jitters and bursts, hence more prone to packet losses. For chatty protocols such as Microsoft CIFS, delays may dramatically degrade the end-to-end throughput, deteriorating the performance significantly.
Byte Cache generally tries to pack as many tokens as possible into a single packet to minimize the overhead of the packet headers, etc., to improve the compression ratio. This is also shown in the diagram in Figure 3. But more token means more waiting time for new packet to come from the LAN without sending out a byte. The delays introduced by accumulating data on the LAN side are usually quite small and are also easy to mitigate. HyperCompression uses a time out as well as the PSH flag of the TCP flows as a trigger to flush out the packet to WAN.
Worse than that is the latency introduced by Hard Drive access. To store as many history data as possible for better average compression ratio, we use Hard Drives to save all the cached histories because the memory is a very limited resource. While the memory access is at the nanosecond level, a SATA Hard Drive access, however, ranges from several milliseconds to sub-100 milliseconds, which is considered big even for the remote network access. To solve this, AppEx HyperCompression intelligently calculates which data blocks are most likely to be used next and keep them in the memory. If they are not in the memory, prefetch them if possible. The disk IO is performed through Asynchronous IO for best performance. Hence there is an algorithm to check if the AIO for a memory chunk is already underway and if it is worth waiting for. With these measures AppEx HyperCompression is able to minimize the delays introduced by the Hard Drive accesses. To enforce a maximum latency limit, there is still a timer set for each of such IO wait. When the timeout fires, the data will be transmitted without looking up the missing history to ensure the overall latency is bounded.

Scalability and Flexibility

As with all the other AppEx technologies, HyperCompression is designed for scalability and flexibility. It can be built to support TBs of disk arrays, while also can be configured to work in a minimal diskless platform, utilizing only the memory to cache data. It can be easily ported to multiple platforms. At the time of writing, AppEx HyperCompression has both Linux and Windows releases.

Use Scenarios

AppEx HyperCompression can be enabled in all the bilateral deployments to significantly boost the effective bandwidth. The flexible design allows it to be applied to a variety of environments under different hardware capacities or budget limitations. The ease of use and low maintenance makes it an ideal solution for SMB with limited IT resources.

A small business customer had a 2Mbps leased line between its headquarter and the branch office. Before deploying AppEx HyperCompression products it had been suffering from unbearable slow file transmissions between the offices. Some business applications were not able to function properly due to the low network throughput. After the HyperCompression products were installed in both sites, the effective bandwidth was elevated by an order of magnitude and networking experience was significantly improved.

In the deployment tests performed by the customer, a large PPT file full of pictures was used to test the effectiveness of the Byte Cache. The first transmission of the file took 13 minutes to complete, occupying the full 2Mbps bandwidth. Then they made a few small modifications to the PPT files and sent it once again. This time it took only 23 seconds for the PPT file which had roughly the same size after the modification to be sent over. The effective bandwidth was boosted to 63Mbps with caching.

Conclusions

AppEx HyperCompression is a flexible and scalable WAN compression solution that offers the following benefits:

- Auto-Discovery / Auto-Negotiation and inline messaging brings about ease of use and low maintenance;
- Combination of Byte Cache and Data Compression offers deep compression with very high compression ratio.
- Optimized history data utilization maximizes the average Byte Cache compression ratio.
- Latency control helps the protocols running at their full capacities over the WAN even under the heavy compression and caching process.

The overall results are a dramatically elevated effective bandwidth over the WAN and a significantly improved network user experience.

AppEx Networks Corporation
4010 Moorpark Ave., suite 212
San Jose, CA 95117

P: +1 408-409-6966
contact@appexnetworks.com
More information can be found at:

www.appexnetworks.com

For a free trial of HyperCompression:

download.appexnetworks.com