Education

Benefits of Hardware Data Compression in Storage Networks

Gerry Simmons, Comtech AHA
Tony Summers, Comtech AHA

SNIA Legal Notice

- The material contained in this tutorial is copyrighted by the SNIA.
- Member companies and individuals may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced without modification
  - The SNIA must be acknowledged as source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of the SNIA Education Committee.

Abstract

Benefits of Hardware Data Compression in Storage Networks

This tutorial explains the benefits and algorithmic details of lossless data compression in Storage Networks, and focuses especially on data de-duplication. The material presents a brief history and background of Data Compression - a primer on the different data compression algorithms in use today. This primer includes performance data on the specific compression algorithms, as well as performance on different data types. Participants will come away with a good understanding of where to place compression in the data path, and the benefits to be gained by such placement. The tutorial will discuss technological advances in compression and how they affect system level solutions.

Agenda

- Introduction and Background
- Lossless Compression Algorithms
- System Implementations
- Technology Advances and Compression Hardware
- Power Conservation and Efficiency
- Conclusion

Agenda

- Introduction and Background
- Lossless Compression Algorithms
- System Implementations
- Technology Advances and Compression Hardware
- Power Conservation and Efficiency
- Conclusion

Introduction, Why Compress?

- Decrease file size and storage requirement
  - (A compression ratio of 3:1 means the input file is three times the size of the compressed file)
- Decrease data size and transfer over the network faster
  - (A compression ratio of 3:1 means data transfers up to three times as quickly across the network)

How to Use the 2:1 Compression Benefit

- Potentially Expand Storage Capacity by 3x.
- Retrieve or Store Data in much less time (up to 66% less.)
- Reduce equipment and Power consumption (up to 66%.) (HVAC power consumption is typically equal to the Equipment Power loading)

Lossless Data Compression, Background

- Lossless versus lossy compression
  - Lossless compression means that no information is lost when a file is compressed and then uncompressed
  - Lossy compression usually results in better compression ratio, but some information (e.g. resolution) is lost
- There are many algorithms and data types. The best solution is to classify files and match the data type to the correct algorithm.

File Types and Lossless Algorithms

File Type	Algorithm
ASCII	LZ based
Grayscale Image	JPEG2000 Lossless
RGBColor	JPEG2000 Lossless
Audio	Real Player Lossless, Apple Lossless

Data that has been previously compressed will typically expand if an attempt is made to compress it again.

Lossless Data Compresssion, Background

- LZ1(LZ77), LZ2(LZ78), were invented by two Computer Scientists:
  - Abraham Lempel
  - Jacob Ziv
  - They published papers in 1977 and 1978 describing two similar compression algorithms.
- LZ1, is the basis for GZIP, PKZIP, WINZIP, ALDC, LZS and PNG among others.
- LZ2 is the basis for LZW and DCLZ. LZW was introduced in 1984 by Terry Welch who added refinements to LZ2 . It is used in TIFF files (LZW).

Lossless Data Compresssion, Background

- Late 1980s DCLZ (LZ2 based), hardware implementation developed by Hewlett Packard and used a 4K linked list Dictionary with SRAM and hashing.
- Early 1990s the first hardware implementation of an LZ compression algorithm using Content Addressable Memory (CAM), DCLZ.
- Late 1990s the Sliding Window based LZ1 devices were becoming popular in tape backup systems and communications applications.

Agenda

- Introduction and Background
- **Lossless Compression Algorithms**
- System Implementation
- Technology Advances and Compression Hardware
- Power Conservation and Efficiency
- Conclusion

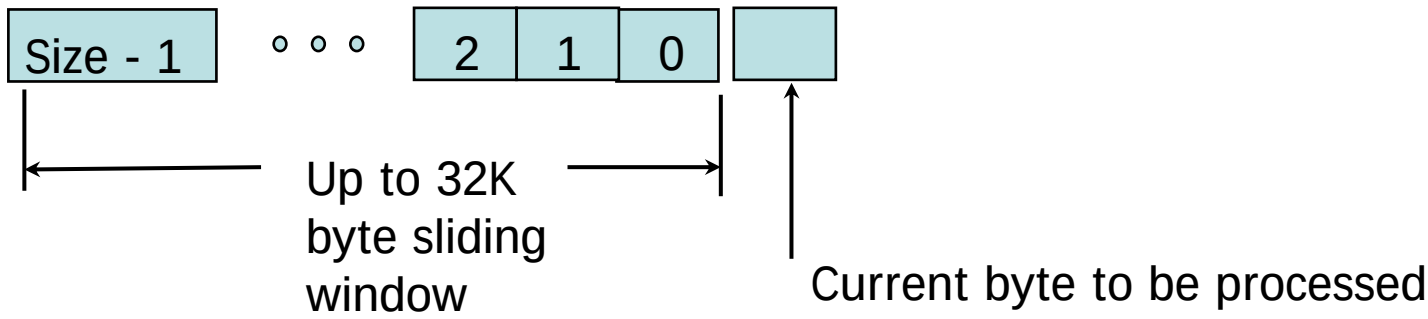
LZ1-Based Algorithms

- ALDC, LZS, and Deflate are LZ1 based algorithms
- Deflate is the algorithm in GZIP, PKZIP, WINZIP, and PNG
- ALDC, LZS, and Deflate Architecture consists of:
  - LZ1 function to identify matches in a sliding window history buffer
  - Post Coder to Huffman encode the matches (length and offset), and literals (uncompressed Bytes).
- ALDC, LZS, and Deflate differences:
  - Sliding window history buffer size
  - Static Huffman encoding
  - Deflate adds Dynamic Huffman and raw Byte encoding

LZ1 Architecture

- The String Matcher searches the history buffer to find repeating strings of Bytes
- The Sliding Window History Buffer adds one new Byte and drops off one Byte from the back end of the history buffer each time a Byte is input and processed
- The Post Coder is a prefix encoder. It can be Static Huffman or Dynamic Huffman. It uses statistics to encode the most common string matches with a smaller number of bits.

LZ1 Algorithm, Sliding Window



ALDC Huffman encodes the Length of String in Bytes. Deflate (GZIP) Huffman encodes Literals, String Matches, and Offset Pointers.

Example: LZ1 String Matching

Input String: ABCDABCFCDAB…..

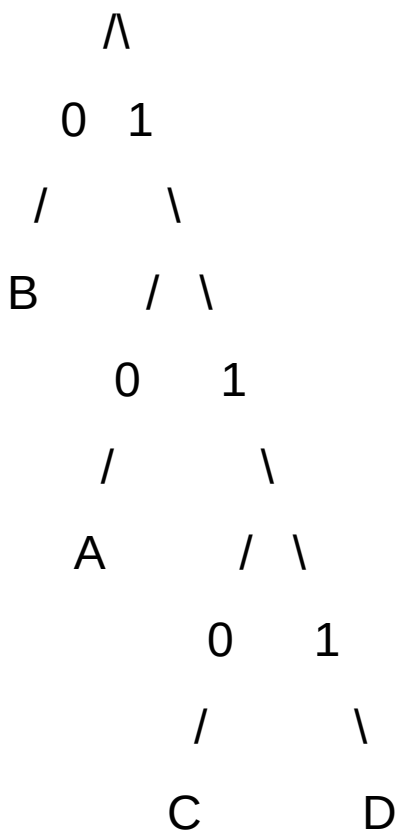
Input	Output
A	A
B	B
C	C
D	D
ABC	Distance=4, Length=3
F	F
CDAB	Distance=6, Length=4

Example 2: Huffman Encoder

Probability of Occurrence

Input character	Probability
A	0.25
B	0.5
C	0.125
D	0.125

Example 2: Huffman Encoder



Symbol	Code	Pr
A	10	0.25
B	0	0.5
C	110	0.125
D	111	0.125

- Reduction =

  ½[0.25(2) + 0.5(1) + 0.125(3) + .125(3)]
  = 0.875

- Reduction in data size due to Huffman encoding.

Compression Ratio Performance, LZ1 based

- Data dependent
  - Random data provides poor compression ratio performance
  - Data with repeating Byte strings, 2 Bytes or longer provides greater compression ratio performance
  - Compression ratios greater than 100:1 are possible
  - May expand if attempting to compress previously compressed data, but a system could detect this and send the original data

Compression Ratio Performance, LZ1 based

- Algorithm dependent
  - Size of sliding window
  - Static or dynamic Huffman encoding
  - Number of matches tracked
  - Length of matches the algorithm will search for

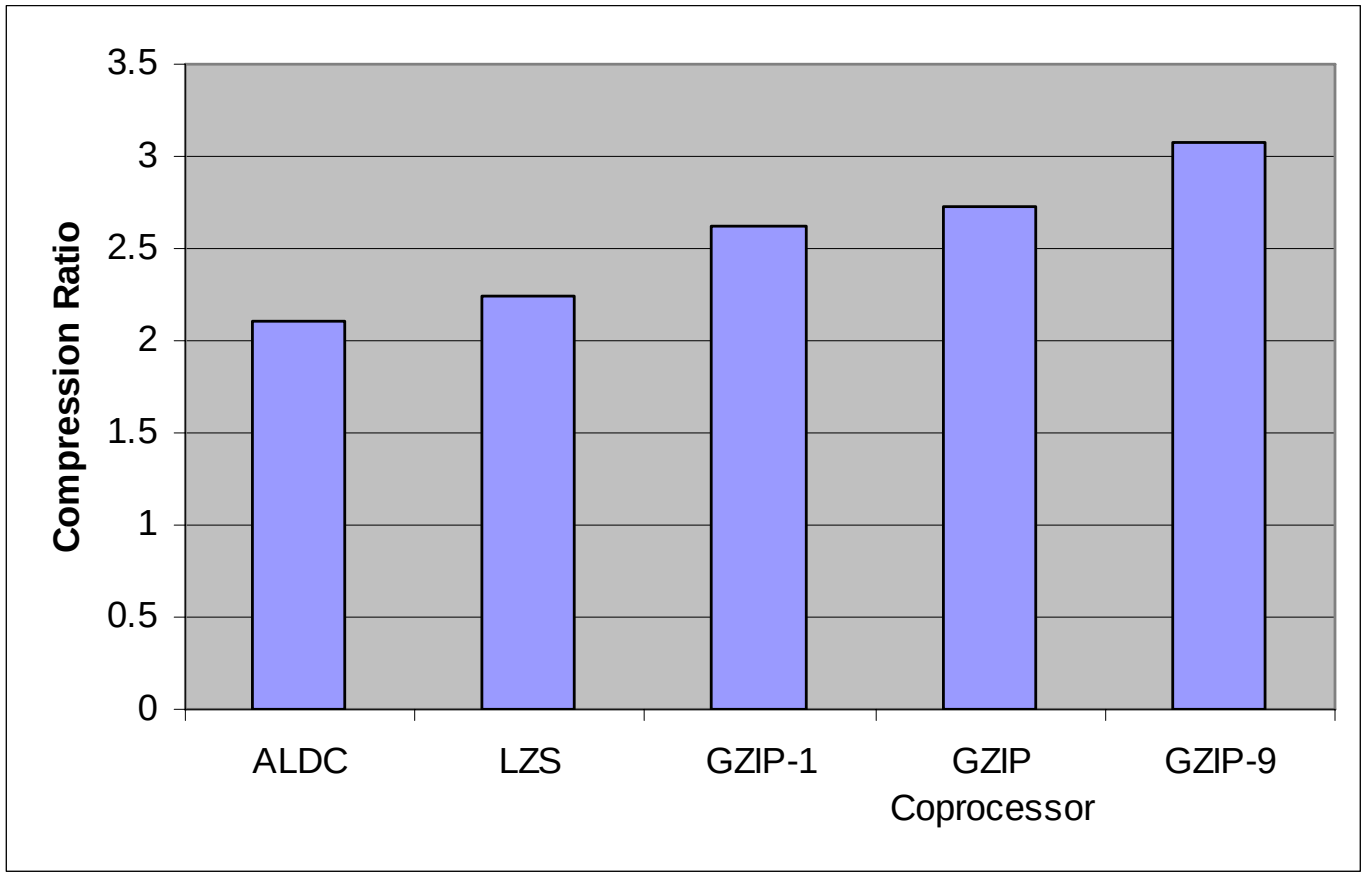
GZIP advantages

- Open standard algorithm – no software license required.
- Industry Standard
  - GZIP is embedded in most Internet Browsers
  - Commonly used in Un*x Operating Systems
- Software for compression or decompression is commonly available.
- Better compression ratio performance than other hardware implemented LZ based algorithms used today.

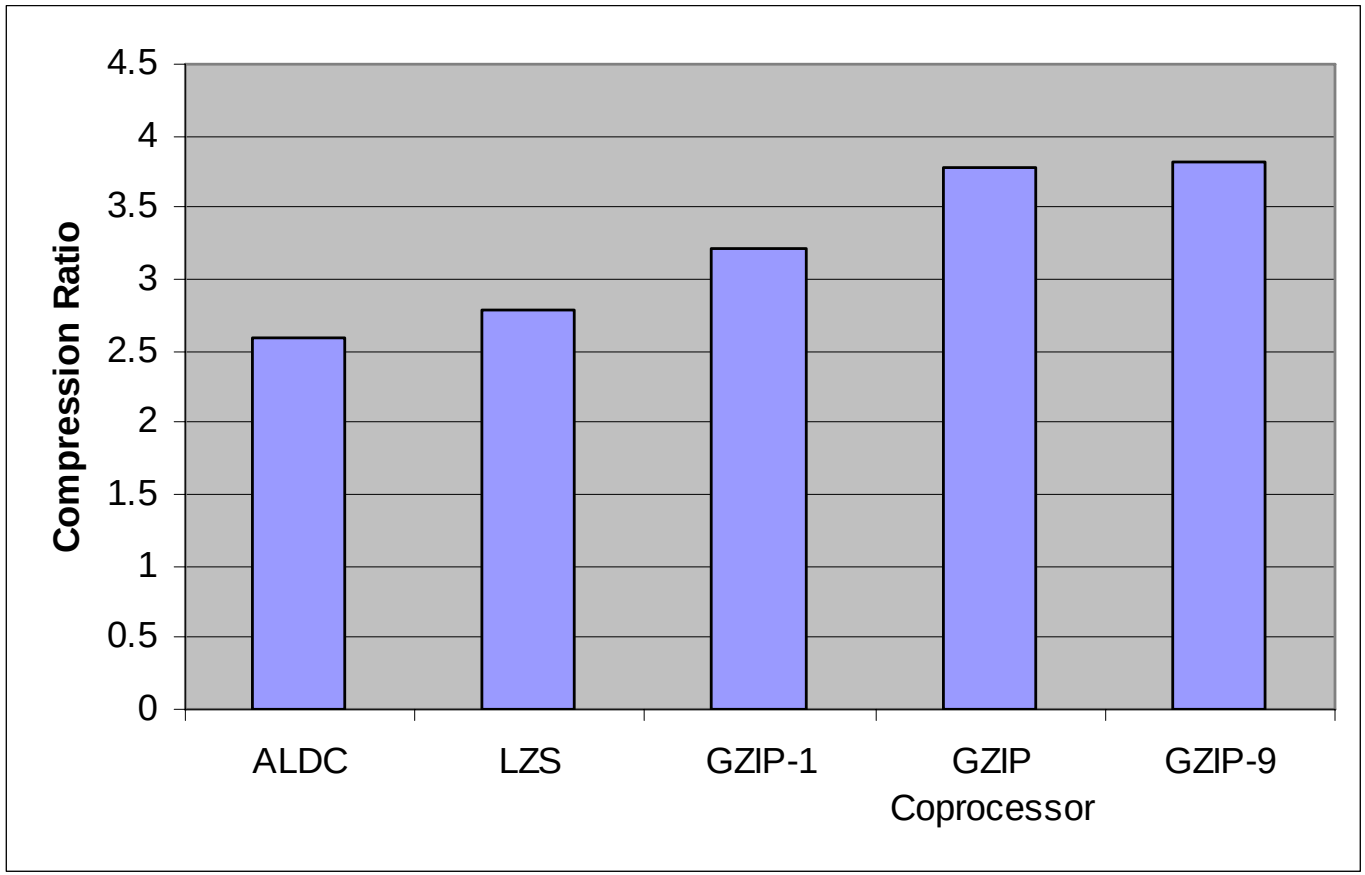
GZIP Software

- Compression levels
  - Level 1, 2 and 3 supports static Huffman
  - Level 4-9 supports dynamic Huffman
    - Each level has limits on:
      - Number of matches it will track
      - Length of matches it will search for
  - Lower levels better for higher throughput
  - Higher levels for better compression ratio performance

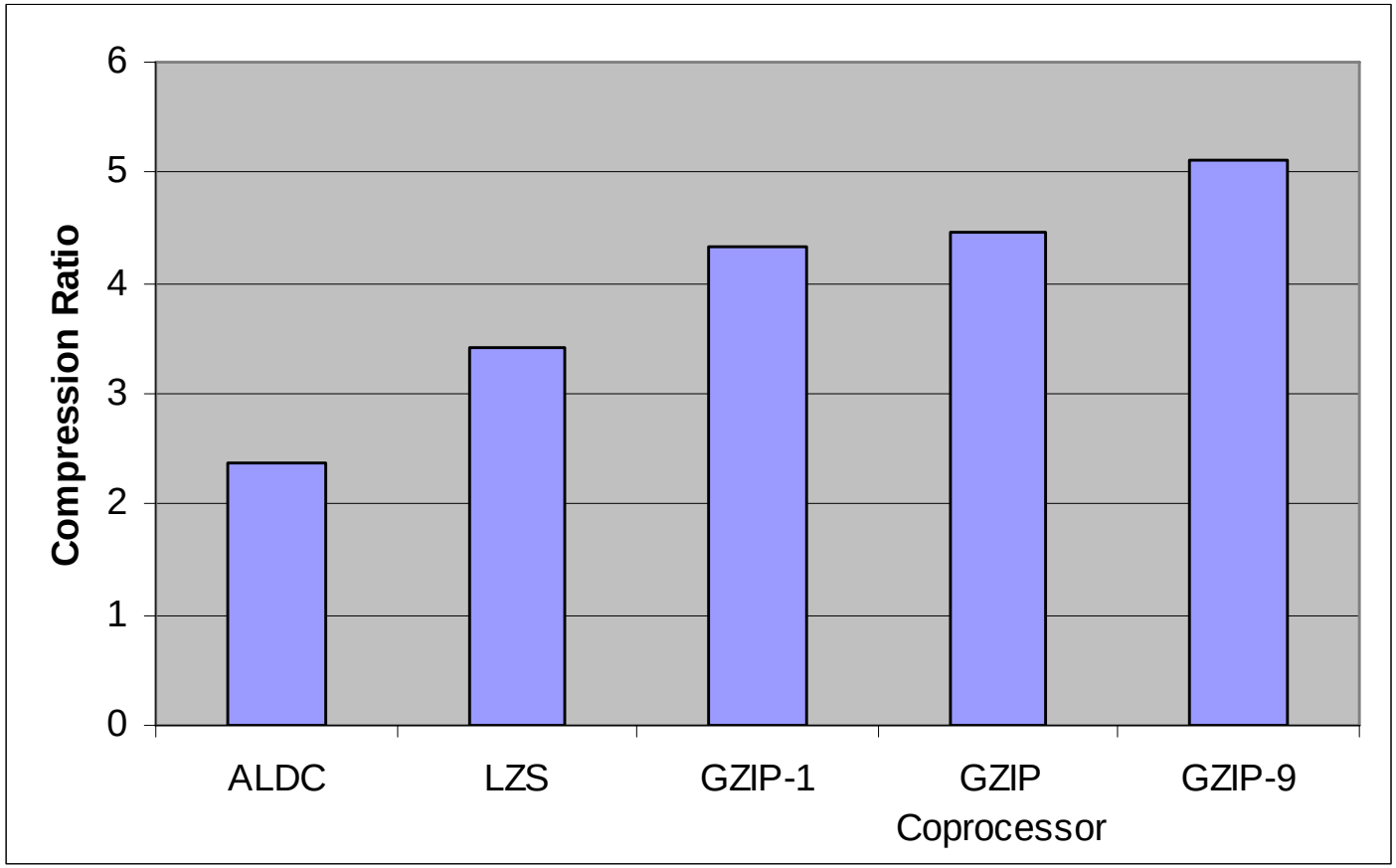
Compression Ratios, Calgary Corpus



Compression Ratios, Canterbury Corpus Corpus



Compression Ratios, HTML Data



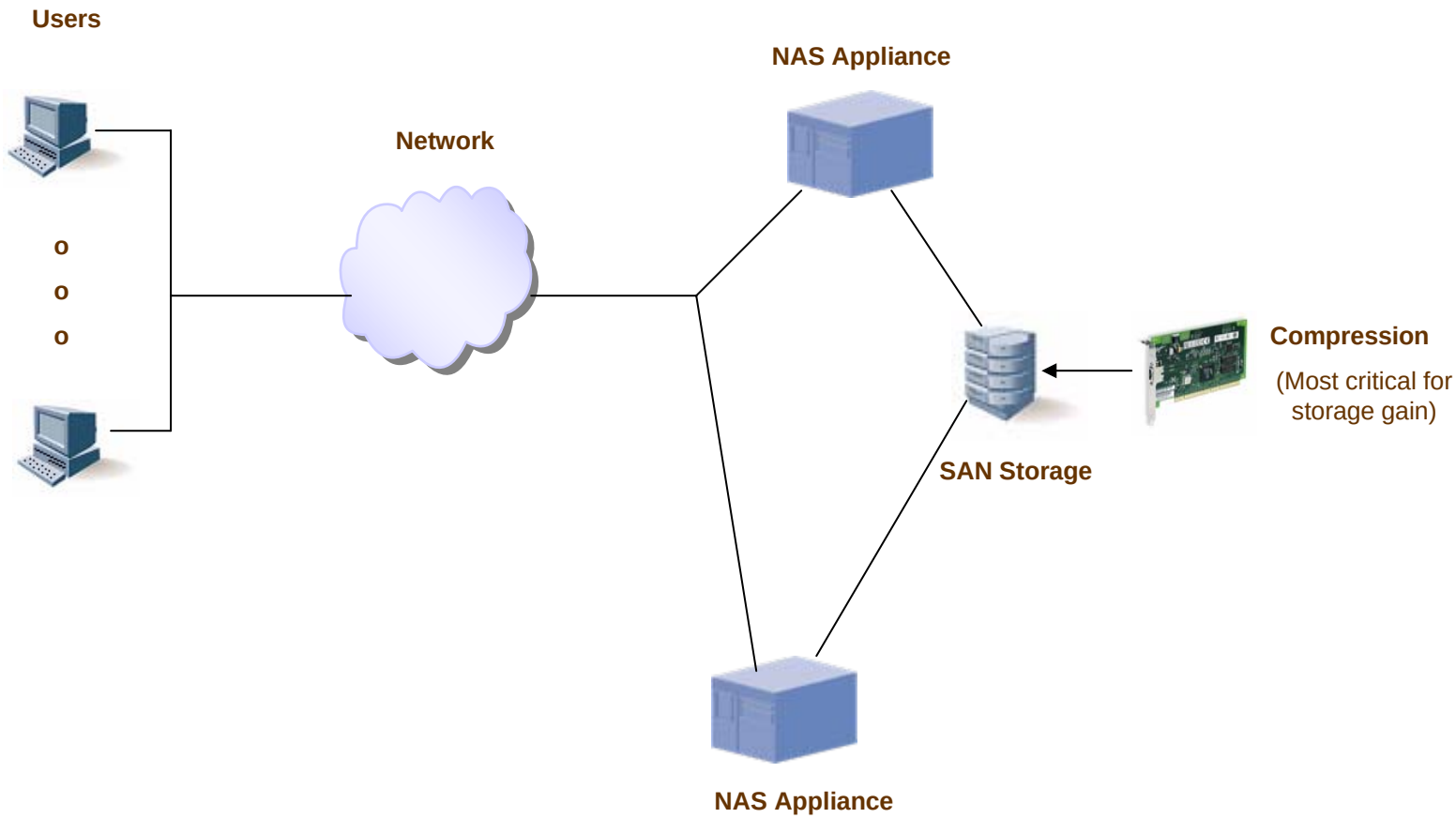
Hardware versus Software

- Higher data rate throughput (10x).
- CPU can offload the compression task, frees up valuable CPU bandwidth, and can reduce power consumption.
- Speed up a network link by sending shorter files
- If choosing GZIP, must evaluate the compressor configuration since there are many options that may or may not be supported.

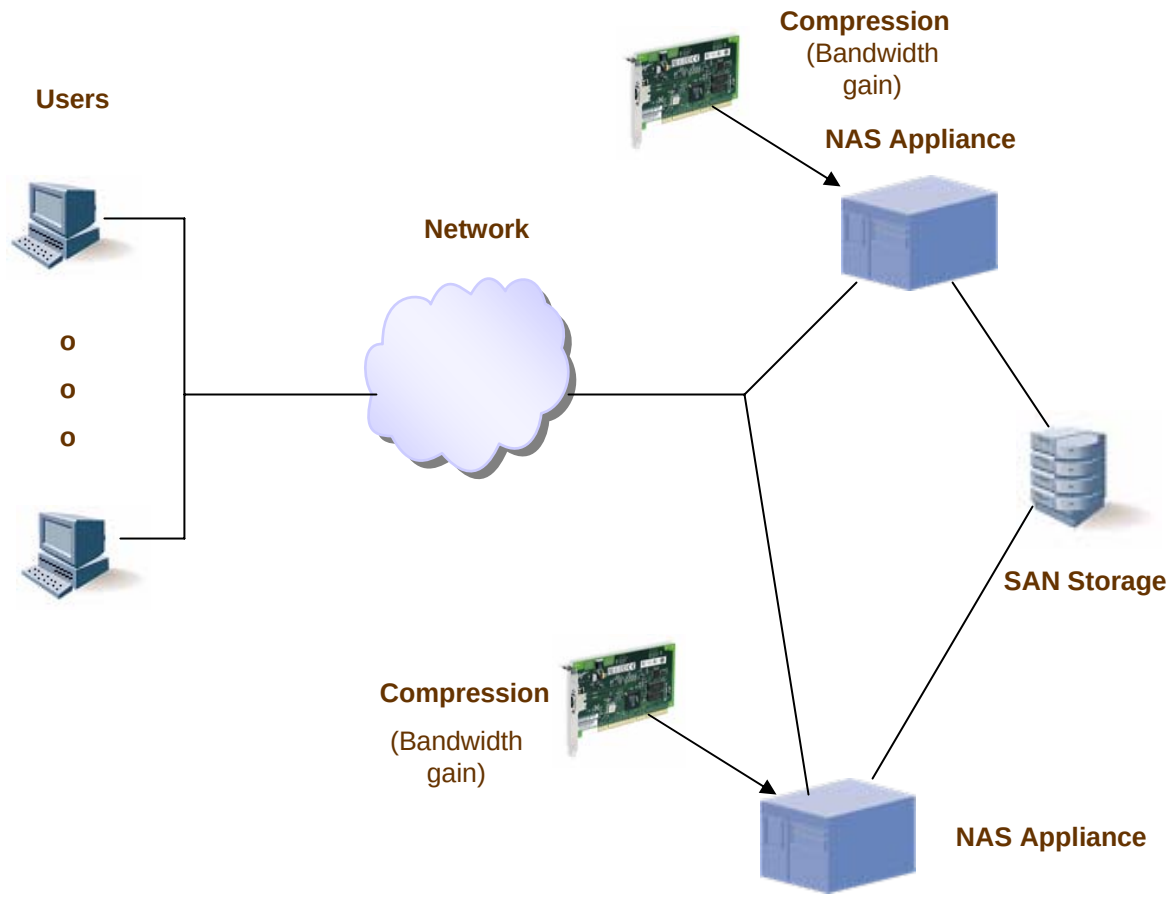
Agenda

- Introduction and Background
- Lossless Compression Algorithms
- **System Implementations**
- Technology Advances and Compression Hardware
- Power Conservation and Efficiency
- Conclusion

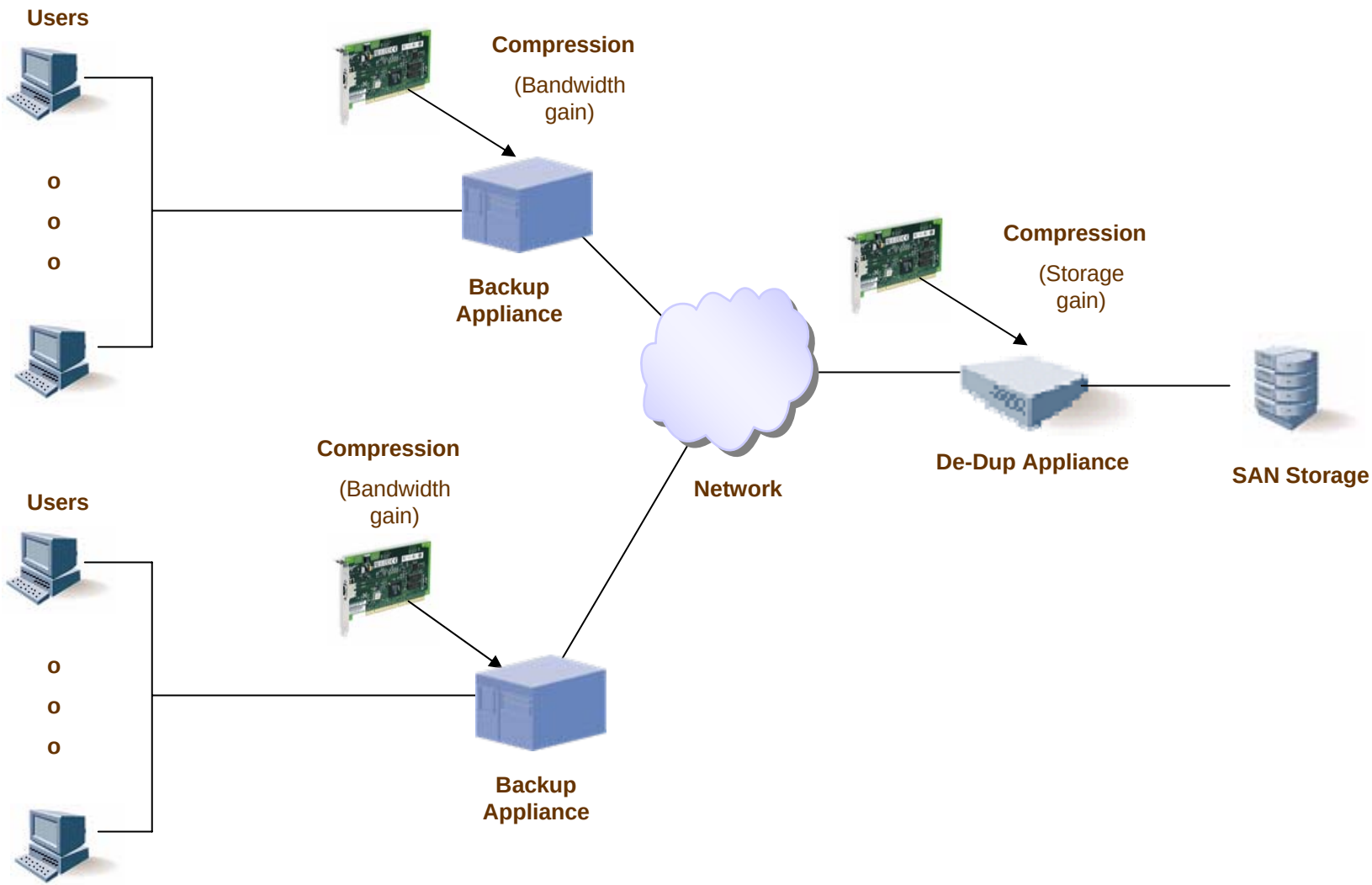
Implementing Compression in the SAN



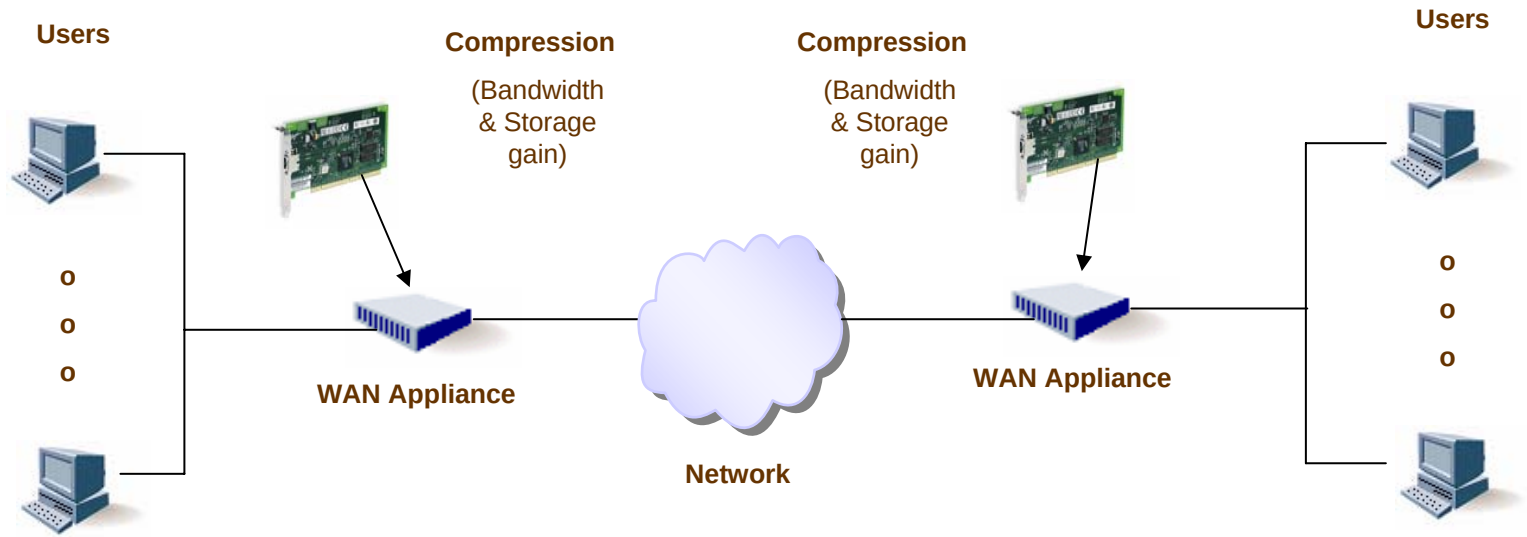
Implementing Compression in the NAS



Implementing Compression in Back-up Servers (with Data De-Duplication)



Data De-Duplication in WAN Optimization



Data De-duplication and Data Compression

- Reduces “macro” redundancies (e.g. attached files, images, signatures, etc.)
- Can get to very high Compression Ratios (40:1)
- VERY Large Data Dictionary (TeraBytes!)
- Uses Lossless Data Compression Algorithms for
  - storage of duplicate data (Data Dictionary) and
  - remaining “meta” data

Implementing Compression Hardware

- Install Compression board
- Install Device Driver and system libraries
  - May be more involved depending on where the compression function resides (i.e. Custom Driver for Appliance specific OS.)
- System Issues
  - Varying Compressed File Sizes
  - Varying Latency
  - Multiple Compression Processors

Agenda

- Introduction and Background
- Lossless Compression Algorithms
- System Implementations
- Technology Advances and Compression Hardware
- Power Conservation and Efficiency
- Conclusion

Technology Advances and Compression Hardware

- 10G Ethernet
- Fiber Transceivers at 10Gbps
- PCI express, 4-lane, 8-lane, 16-lane
- Scatter/Gather DMA
- Up to 8 Gigabit/s LZ1 compression boards

Other Algorithms

- If the data is an image type with multiple bits per pixel
  - JPEG2000 in Lossless mode
    - Uses 5/3 Wavelet Transform
  - PNG uses Deflate with preprocessing

JPEG2000 Comparison

- Original Photograph: 69 MegaBytes
- TIFF LZW: 38 MegaBytes
- JPEG 2000 Lossless: 11.8 MegaBytes

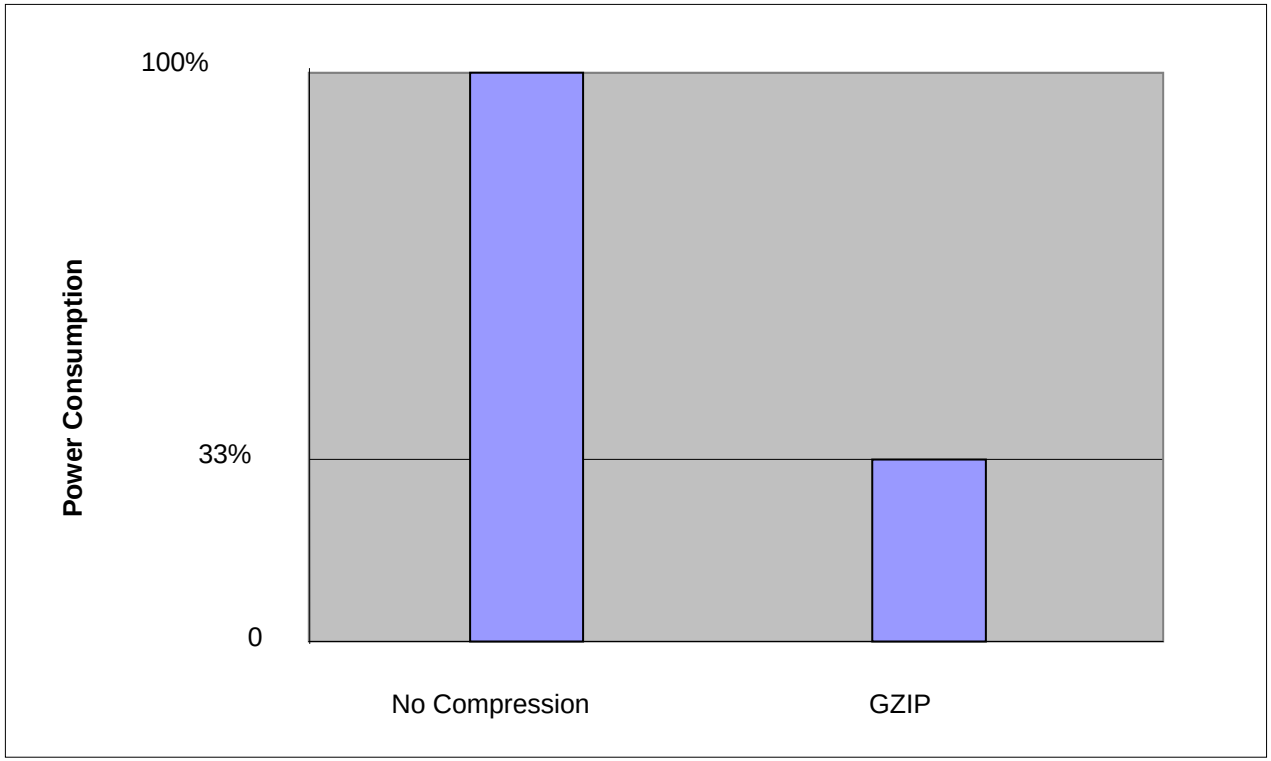
Agenda

- Introduction and Background
- Lossless Compression Algorithms
- System Implementations
- Technology Advances and Compression Hardware
- Power Conservation and Efficiency
- Conclusion

Going Green with GZIP Data Compression

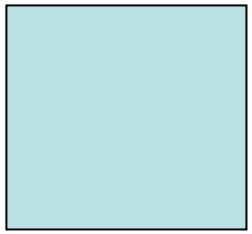
- Upgrading a data center from no compression to GZIP can result in up to 66% power reduction since you now need only 1/3 the equipment.
- Additional Notes:
  - This assumes you want to apply all the gain to power savings
  - HVAC power savings is typically as great as the power loading from the equipment removed.
  - Most Enterprise class systems cannot tolerate the speed of GZIP running in software.

Going Green

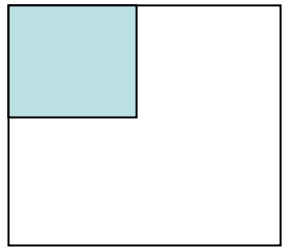


- Implement GZIP Compression

GZIP COPROCESSOR TO CPU COMPARISON



GZIP COPROCESSOR



¼ OF A QUAD-CORE CPU
RUNNING GZIP
SOFTWARE

POWER COSUMPTION,
MAX DATA RATE OF ONE GIGABIT/SEC

GZIP COPROCESSOR	= 1 WATT
¼ OF A QUAD-CORE CPU	= 25 WATTS

Agenda

- Introduction and Background
- Lossless Compression Algorithms
- System Implementations
- Technology Advances and Compression Hardware
- Power Conservation and Efficiency
- Conclusion

Conclusion

- GZIP is the best performing LZ based hardware compression solution available.
- JPEG2000 Lossless is the best performing multi-bit image compression algorithm.
- Offloading Compression to a Coprocessor frees up valuable CPU bandwidth and saves power.
- Benefits of Compression:
  - Pack 2 to 3 times more data onto mass storage media
  - Speed up a communications link by 2x or 3x.
  - Can reduce power consumption by as much as 66%.

References

- Welch, Terry A (1984)., A Technique For Performance Data Compression, IEEE Computer, vol. 17 no. 6 (June 1984)
- Network Working Group, RFC 1951. DEFLATE Compressed Data Format Specification, May 1996
- Keary, Major (1994). Data Compression [Electronic Version] Retrieved August 8, 2006 http://www.melbpc.org.au/pcupdate/9407/9407article.htm
- Milburn, Ken (2003).JPEG2000: The Killer Image File Format for Lossless Storage [Electronic Version] Retrieved August 18, 2006 http://www.oreillynet.com/pub/a/javascript/2003/11/14/digphoto_ckbk.html

Q&A / Feedback

- Please send any questions or comments on this presentation to SNIA: trackstorage@snia.org

Many thanks to the following individuals for their contributions to this tutorial.

SNIA Education Committee

Sean Gettmann
Nancy Clay
Brandy Barton
Rob Peglar
Lynne VanArdsdale