IBM Z Data Compression (zEDC) for z/OS and zEDC Express

Explosive amount of data

According to industry experts, the digital information that exists in the present time will grow from 3.2 zettabytes to 40 zettabytes by 2020[1]. Data is the lifeblood of every organization. It doesn't matter how large or how small an organization is today, data and the ability to access growing volumes of data is critical to running the business.

Compression offerings widely used today

Hardware compression of data on IBM Z® has been available for some time. It can help reduce storage costs by decreasing the size of data that needs to be stored and by improving I/O throughput. This compression is implemented directly on the Z processor chip with the CMPSC instruction, an IBM compression using a dictionary-based approach. The processor chip provides advantages over software compression by using one-fifth the number of processor cycles.

A compression gap still exists

Massive amounts of data are still not compressed often because the processor time to compress and/or decompress compared to the savings to store and the wall clock time to move the data.

The zEDC solution

IBM Z Data Compression (zEDC) offers a compression acceleration solution designed for high performance, low latency compression with few processor cycles and little additional overhead.

zEDC is especially optimized for larger data files. That means that zEDC is complementary to existing Z compression technology. Smaller records and files best suited for hardware chip compression will still use hardware compression. z/OS® will intelligently determine the larger files which are candidates and directs those to zEDC.

And zEDC is 'dictionary-less' and uses optimized algorithms to deliver high performance.

Industry standard

zEDC is compatible with industry standard, open zlib based compression – used today by Java and other applications.

z/OS V2 or later provides the zlib library, which can send compression and decompression requests to zEDC Express.

The z/OS-provided zlib library can be statically linked into IBM, ISV, or customer applications currently use zlib, enabling additional exploitation of compression through zEDC Express and expanding potential compression opportunities. When zlib uses zEDC, there can be up to 118X reduction in CPU and up to 24X throughput improvement.[1]

Excellent cross platform solution

zEDC is well suited for an environment where large files created and compressed using zlib on one system need to be transferred to another system for decompression and processing. Having zEDC on each server can reduce latency for large data files that use zlib compression.

Benefits from zEDC Compression

Compress SMF records

zEDC with z/OS SMF Logger alleviates SMF constraints across the entire life cycle of a record using compression technology while storing up to 4X less data in System Logger and reducing Logger CPU usage by up to 30%.[2]

BSAM/QSAM

Customers using large sequential BSAM/QSAM extended format, files can use zEDC to help reduce disk space up to 75% and improve effective bandwidth without significant CPU overhead.

DFSMSdss™/DFSMShsm™ for backup and restore

zEDC can deliver efficient compression when backing up / restoring data. The result is less physical data movement with less CPU overhead than with other software-based compression technologies.

ZSB03059-USEN-06

Transparent Java exploitation

A zEDC java.util.zip.Deflater in memory test improved elapsed time up to 55x and CPU time up to 240x when compared to zlib software compression.[3] And, using java.util.zip.GZIPOutputStream class, there was up to 90% reduction in CPU time using zEDC hardware versus zlib software.[3]

IBM Encryption Facility for z/OS

zEDC can provide IBM Encryption Facility users reductions of up to 60% in elapsed time and up to 70% in CPU time for environments where compression is already in use. It complements the software compression that exists today with IBM Encryption Facility OpenPGP support.

IBM zSecure™ V2.3

zSecure uses zEDC to reduce the output size of Access Monitor and UNLOAD files by up to 10X and CKFREEZE files reduced by up to 4X.

z/OS RMF™ Reporting

z/OS offers enhanced RMF reporting to report on vital statistics relevant for compression. The SMF 74 SubType 9 record has been updated with PCIe and zEDC statistics. RMF Monitor III can be used to post-process these records.

zEDC candidate analysis

The IBM z System Batch Network Analyzer (zBNA) is a no charge, as-is tool that analyzes batch windows using SMF records to help determine if you have files that are candidates for zEDC. It can also help estimate the number of hardware features you will need. It is available from: http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS5132?OpenDocument&ExpandSection=5%2C6#_Section5

zEDC Requirements

The minimum HW requirements are:

- IBM z14™ (z14), IBM z13s™ (z13s), IBM z13™ (z13), IBM zEnterprise® EC12, or zEnterprise BC12
- zEDC Express - recommend minimum of two adapters
- For best performance, all systems accessing the compressed data should have zEDC Express installed

The minimum software requirements are:

- z/OS V2.1 or later with zEDC for z/OS feature, and applicable PTFs

Summary

The z/OS V2 or later zEDC capability and the hardware feature zEDC Express offers a new compression acceleration option. zEDC is compatible with the current coprocessor compression – not a replacement for it. The use of zEDC allows the opportunity to store compressed data on Z more cost effectively and can enable you to more effectively use network bandwidth when sending compressed files.

1. Business2Community http://www.business2community.com/big-data/top-10-amazing-facts-know-big-data-01357650#dECHlI6zsZBDiyzi.99
2. These results are based on projections and measurements completed in a controlled environment. Results may vary by customer based on individual workload, configuration and software levels.
3. These results are based on projections and measurements completed in a controlled environment. Results may vary by customer based on individual workload, configuration and software levels.
4. Results are based on internal controlled measurements using java.util.zip.deflater on data already in memory. Results may vary based the application's use of java.util.zip classes and other work done by the application.

ZSB03059-USEN-06