Saheem Granados
Richard Schultz

Encryption Facility R2 for z/OS Performance

Introduction

The Encryption Facility for z/OS® processes data at rest and is intended for encryption of media whose contents must be securely transported: physically moved such as being shipped in a truck, for example, or electronically sent over non-secure links.

The “security” of the movement here covers both network eavesdropping and unauthorized reading of physical media containing sensitive information. Consequences of such an unauthorized disclosure of information can be severe, as illustrated by many reported examples in which companies’ finances and image were affected after losing track of physical media with sensitive contents that are known to be easily readable when one has access to the media itself.

IBM® Encryption Facility for z/OS exploits the existing strengths of the mainframe and the IBM z/OS operating system. It is a host-based facility that leverages existing centralized key management in z/OS and the hardware encryption capabilities of IBM mainframes.

Encryption Facility can make use of ICSF to perform encryption and decryption and to manage cryptographic keys. To encrypt data files, Encryption Facility uses the following kinds of cryptographic keys:

- TDES triple-length keys
- 128–bit AES keys

Although the Encryption Facility for z/OS V1.1 implementation was based on a proprietary data format, the V1.2 release provides support for the OpenPGP Message Format standard as defined in RFC 2440. The OpenPGP standard was originally derived from PGP (Pretty Good Privacy). This support will meet the demand for standards-based solutions. For more about Encryption Facility for z/OS OpenPGP support, refer to *Encryption Facility for z/OS OpenPGP Support*, SG24-7434 (available soon).

This IBM Redpaper focuses on Encryption Facility for z/OS performance in the context of the two metrics: the use of the zAAP processor and the use of parallel processing. Statistics and

comparisons illustrate the impact of the different performance-enhancing options and demonstrate how the OpenPGP support on z/OS can provide full OpenPGP compliance while efficiently using system resources and completing tasks in a timely manner. Ultimately, the efficient use of system resources will result in an overall lower financial cost for performing essential data integrity services within the enterprise.

Attention: Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending on considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

This information is presented with general recommendations to help you better understand IBM products. In addition, note that:

- All tests were run on IBM zSeries® System z9™.
- All data was gathered using a 1.5 GB input file.
- All I/O is done to and from data sets on DASD.
- The encryption algorithm used was AES with 128-bit keys.
- Due to compression, the output file might not be the same size as the input file.
- The CPU utilization percentage given in many tables is the average utilization measured over all enabled processors of the same type. That is, an average might be given for all the zAAP processors and a different average might be given for all the general CPUs. As a result, when multiple processors are online, the calculated average might appear to be affected. However, the aggregate of CPU service unit consumption is roughly equivalent.
- The elapsed time is given in seconds.
- For statistics for zAAP processors, two internal throughput numbers are given: throughput for the general CPUs only and throughput for the general CPU and zAAP combined.
- When doing compression, the ZIP compression algorithm was used with level 1. The value of 1 represents best speed.

  Note: Tests have shown a compression level of 9 – best compression, results in a very significant performance degradation.
- Compression performance and compression percentages are greatly affected by the type of data being compressed.

Overview

Performance is critical to the deployment of any solution in a business's enterprise. The following two performance metrics are used to discuss the performance implications of deploying Encryption Facility V1.2's OpenPGP support:

- Execution time measured by a standard watch or clock and the amount of work accomplished within that time
- Cost measured by the CPU consumption on general-purpose processors

The OpenPGP support is a stand-alone Java™ application. Applications written in Java on z/OS require the IBM Java Runtime Environment for z/OS (JRE™). This environment consists of an address space where the JRE executes. The JRE interprets compiled Java byte code that serves as the OpenPGP support's program code. Given the additional layer of execution and Java's interpretive nature, the amount of CPU time consumed and the amount of execution time to complete a task are affected immediately. Moreover, the cryptographic nature of the Encryption Facility products requires the completion of computation-intensive operations that could further increase the amount of CPU time consumed and execution time. Finally, OpenPGP's main purpose is to provide data integrity services on data sets and UNIX® System Services (USS) files. This involves I/O-intensive operations that might result in inefficient utilization of the CPU. To address these issues, performance-enhancing options are available in three areas:

- IBM Java 5 SDK Runtime Environment for z/OS
- z/OS specialized hardware: zAAP and CPACF specialized processors
- OpenPGP support code

IBM Java SDK 5 Runtime Environment for z/OS

The IBM JRE for z/OS provides the just-in-time (JIT) compiler. The JIT compiler is responsible for optimizing Java byte code and compiling the interpreted code into persistent load modules. Compiling code into the persistent load modules alleviates the CPU service unit cost of reinterpreting code when it is executed again during the life of the application. Moreover, because it is aware of the processor type, the JIT compiler can use the latest hardware technology to optimize the compilation.

Note: It is strongly recommended that the JIT compiler always remain enabled. Thus, we will not discuss the performance implications of disabling the JIT compiler. If disabled, the performance of any Java application may be degraded significantly.

z/OS specialized hardware: zAAP and CPACF

The CP Assist for Cryptographic Function (CPACF) is the z/OS feature that can be used to accelerate symmetric encryption and decryption and hash calculations. The CPACF performs cryptographic functions at an accelerated rate, helping to reduce the total elapsed time and CPU time.

The System z™ Application Assist Processor (zAAP) is specialized processing unit that only runs Java workloads. When workloads are dispatched on a zAAP processor, no general CPU service units are consumed. These features can ultimately result in a lower overall financial cost of performing data integrity services on data sets or files.

Encryption Facility OpenPGP support

The OpenPGP support provides configuration options that enable parallelized internal processing. This directly addresses the inefficient use of the CPU while I/O operations are being performed. These options pipeline the different tasks that are required to provide OpenPGP compliance; encryption, I/O, and compression (if enabled) may execute in parallel and on different CPUs.

General CPU Service Units reduction using z/OS specialized hardware

Two specialized processors may be leveraged to make a positive impact on performance: zAAP specialized processor and the CPACF embedded processor. Whereas zAAP usage is external to the OpenPGP configuration, the OpenPGP support allows two options for performing the cryptographic functions required by RFC 2440. The user may set the JCE_PROVIDER_LIST configuration option to `com.ibm.crypto.hdwrCCA.provider.IBMJCECCA` or specify the `–jce-providers` command line option with value `com.ibm.crypto.hdwrCCA.provider.IBMJCECCA`. This enables the hardware cryptographic JCE provider. This provider leverages ICSF and the CPACF to perform cryptographic functions. If these options are not specified and the security.provider.1 keyword in the $(java_home)/lib/security/java.security file has not been updated to list the com.ibm.crypto.hdwrCCA.provider.IBMJCECCA provider, Java code within the JCE component of the JRE will perform all the cryptographic function.

Hardware cryptographic acceleration

Table 1 lists the performance statistics measured *without* the hardware provider enabled, only *one* general CPU online, and compression level 1 enabled.

Table 1 Performance statistics without hardware provider and one CPU

Input data size (bytes)	Output data size (bytes)	Elapsed time (sec)	MB/sec	CPU utilization %	MB/CPU sec
1,506,431,796	346,727,868	172.86	8.31	74.38%	11.17

Table 2 lists the statistics measured with the hardware provider *enabled* and only *one* general CPU online.

Table 2 Performance statistics with hardware provider and one CPU

Input data size (bytes)	Output data size (bytes)	Elapsed time (sec)	MB/sec	CPU utilization %	MB/CPU sec
1,506,431,796	346,727,868	144.71	9.93	70.84%	14.01

The hardware provider reduced the elapsed time by 16% and improved the MB/CPU seconds by 25%.

zAAP usage

Table 3 lists the statistics for the same test but measured *without* the hardware provider being enabled and with *one* general CPU online and *one* zAAP process online.

Table 3 Performance statistics with specialized hardware

Elapsed time (sec)	MB/sec	zAAP utilization %	General CPU utilization %	MB/general CPU sec	MB/general + zAAP sec
167.32	8.59	69.36%	11.62%	73.89	10.60

Comparing the results in tables 1 and 3, general CPU utilization is reduced by more than 84%. Table 3 separates internal throughput (MB/CPU sec) into two categories: “MB/general CPU seconds” and “MB/zAAP+general CPU seconds.” This distinction is given to allow those who use throughput as an informal indicator of cost to quickly see the financial impact of a zAAP processor. Finally, the introduction of the zAAP processor showed a slight reduction in elapsed time of about 3%. Using a zAAP, the MB/zAAP+general CPU seconds was about 5% lower than the total MB/CPU seconds without a zAAP. Without the hardware provider, 86% of the total CPU activity is eligible to run on a zAAP.

These comparisons are also valid for the case when the specialized cryptographic processor is being leveraged.

Table 4 lists the statistics measured with the hardware provider being *enabled* and *one* general CPU online and *one* zAAP process online.

Table 4 Performance statistics with specialized hardware and hardware provider

Elapsed time (sec)	MB/sec	zAAP utilization %	General CPU utilization %	MB/general CPU sec	MB/general + zAAP sec
136.53	10.52	66.29%	12.83%	82.02	13.30

Comparing the results in table 2 and 4, general CPU utilization is reduced by more than 80% while the calculated general MB/CPU sec is increased by nearly 10%. Using a zAAP, the MB/zAAP+general CPU seconds was about 5% lower than the total MB/CPU seconds without a zAAP. Using the hardware provider, 84% of the total CPU activity is eligible to run on a zAAP.

Execution time reduction using parallel processing

The OpenPGP support’s processing characteristics allow for processing inefficiencies. Specifically, while the OpenPGP support code awaits the completion of an I/O operation the CPU is waiting. This holds true for both general-purpose CPUs and specialized zAAP processors. An idle CPU can be seen as lost opportunity to reduce the total execution time.

The OpenPGP support provides three configuration file options that enable a multi-threaded approach to processing, as described in Table 5.

Table 5 Configuration file options

Configuration option	Processing task
USE_ASYNC_IO	File or Data Set Input/Output
USE_ASYNC_CIPHER	Encryption/Decryption
USE_ASYNC_COMPRESS	Compression/Decompression

When any or all of these configuration options are enabled, the associated processing tasks are performed in a separate thread of execution. In effect this relieves the CPU from waiting on the task to complete before continuing main line processing. Further, multiple CPUs can be used to handle processing concurrently. These benefits make way for a reduction in execution time.

Table 6 (shown here for simplicity) lists some performance statistics of encrypting and compressing *without* any of the options enabled and only *one* general CPU online.

Table 6 Performance statistics without any options enabled

Input data size (bytes)	Output data size (bytes)	Elapsed time (sec)	MB/sec	CPU utilization %	MB/CPU sec
1,506,431,796	346,727,868	144.71	9.93	70.84%	14.01

In contrast, Table 7 lists the statistics when USE_ASYNC_IO, USE_ASYNC_COMPRESS, and USE_ASYNC_CIPHER are *enabled* and only *one* general CPU is online.

Table 7 Performance statistics with options enabled

Input data size (bytes)	Output data size (bytes)	Elapsed time (sec)	MB/sec	CPU utilization %	MB/CPU sec
1,506,431,796	346,727,868	178.90	8.03	74.22	10.82

With only one general CPU available, full parallel processing actually shows an increase of elapsed time of about 24% and a 23% drop in MB/CPU seconds.

Table 8 shown next, lists the statistics when USE_ASYNC_IO, USE_ASYNC_COMPRESS, and USE_ASYNC_CIPHER are enabled and four general CPUs are online.

Table 8 Performance statistics with options enabled

Input data size (bytes)	Output data size (bytes)	Elapsed time (sec)	MB/sec	CPU utilization %	MB/CPU sec
1,506,431,796	346,727,868	94.02	15.28	33.91%	11.27

A 47% improvement is seen when comparing the elapsed times of *enabling all* of the parallel processing features and with *four* general CPUs online, to *no parallel processing* and only *one* general CPU online. External throughput (MB/sec) is significantly improved. Internal throughput (MB/CPU sec) shows a slight improvement.

Comparing to Table 6, on a system with four general CPUs the ASYNC_ options provide a 35% reduction in elapsed time. However, they increase the CPU cost by 24%.

Putting it all together

The previous sections have separately shown the effectiveness of introducing zAAP and hardware cryptography. Also, when multiple CPUs were available, the parallel processing options showed improvements in the elapsed time needed to complete encryption with compression. This section demonstrates the overall impacts of combining parallel processing with multiple CPUs and specialized hardware.

Table 9 lists the statistics for the same tests when USE_ASYNC_IO, USE_ASYNC_COMPRESS, and USE_ASYNC_CIPHER are enabled and two general CPUs are online and two zAAP CPUs are online.

Table 9 Performance statistics with specialized hardware and hardware provider

Elapsed Time (sec)	MB/sec	zAAP utilization %	General CPU utilization %	MB/general CPU sec	MB/general zAAP sec
128.11	11.21	56.10%	3.59%	156.19	9.39

Table 10 (show here again for simplicity) lists the statistics when one general CPU is online, no parallel processing options are enabled, and the hardware provider is not specified.

Table 10 Performance statistics without hardware provider

Input data size (bytes)	Output data size (bytes)	Elapsed time (sec)	MB/sec	CPU utilization %	MB/CPU sec
1,506,431,796	346,727,868	172.86	8.31	74.38%	11.17

Table 9 shows significant improvement over Table 10:

- Elapsed time was reduced by more than 25%.
- General CPU utilization was reduced by more than 95%.
- MB/CPU seconds increased by approximately 35%.
- While the total CPU cost (general processor and zAAP) went up by 19%, a full 94% of the CPU activity was able to run on a zAAP. For installation with multiple CPUs and zAAPs configured, this probably is the lowest cost option.

Conclusion

Performance characteristics can differentiate a product's usefulness to an enterprise. Two metrics that are key to the analysis of performance are elapsed clock/watch time needed to complete a task and the CPU time used to complete said task. We demonstrated some of the techniques that can used to achieve enhanced performance when using OpenPGP services. Clearly, all installations with CPACF should exploit this technology. Installations with multiple general CPUs and zAAP processors online can benefit from enabling of all of the parallel processing options for the OpenPGP support.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

This document REDP-4334-00 was created or updated on July 5, 2007.

Send us your comments in one of the following ways:
- Use the online **Contact us** review Redbooks form found at:
  ibm.com/redbooks
- Send your comments in an e-mail to:
  redbooks@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400 U.S.A.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

Redbooks (logo) ®	z9™	System z9™
z/OS®	IBM®	
zSeries®	System z™	

The following terms are trademarks of other companies:

Java, JRE, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.