EUROPEAN WORKSHOP ON ON-BOARD DATA PROCESSING (OBDP2021), 14-17 JUNE 2021

LOW LATENCY ON-BOARD DATA HANDLING FOR EARTH OBSERVATION SATELLITES USING OFF-THE-SHELF COMPONENTS

Michele Caon, Paolo Motto Ros, Tiziano Bianchi, Maurizio Martina, and Enrico Magli

Department of Electronics and Telecommunications,
Politecnico di Torino, Turin, Italy

ABSTRACT

Satellite Earth Observation is nowadays receiving significant attention. In this regard, the latency of Earth Observation product provision to the ground segment is undoubtedly among the first key performance indicators for these systems. The European Union Horizon 2020 EO-ALERT project aims at overcoming the limitations of traditional Near Real-Time onboard data chain architectures by moving all the critical processing tasks on the flight segment and accelerating them using high-performance commercial off-the-shelf devices. The resulting architecture minimizes the amount of transmitted data and eliminates ground-based data processing from the Earth Observation data chain, hence achieving actual real-time product delivery in less than $5\,\mathrm{min}$ with optical and Synthetic Aperture Radar data. This paper presents the performance benefits of a mixed software-hardware design of the EO-ALERT CPU Scheduling, Compression, Encryption, and Data Handling Subsystem responsible for data compression and encryption as well as data routing and scheduling tasks. Compared to a software-only solution, the exploited High-Level Synthesis methodology enables 5 to 7-fold speed-up in onboard image compression and encryption tasks and 2 to 5-fold reduction in the contribution of the CPU Scheduling, Compression, Encryption, and Data Handling Subsystem to the overall onboard image data chain while contributing by less than $1\,\mathrm{s}$ to the delivery of the alerts to the end-user.

Key words: Onboard Compression and Encryption; Low Latency Data Handling; Earth Observation; High-Level Synthesis; CCSDS-123.

1. INTRODUCTION

The short-time availability of Earth Observation (EO) products has gained importance in the last decades due to their employment in environment monitoring and civilian security applications. This lead to the creation of comprehensive EO programmes, like Copernicus [1] in Europe, with missions aimed at delivering high-resolution image products to the Ground Station (GS) within a few hours from acquisition time as in [2] and [3]. The data chain structure of such EO satellites relies on the Flight Segment (FS) for raw data acquisition and compression, while the image processing tasks are performed at the GS. The latency of raw data transmission prevents such systems from achieving better than Near Real-Time (NRT) delivery of EO products, which are typically available to the end-user after $1\,\mathrm{h}$ to $3\,\mathrm{h}$ from acquisition time [4].

Nowadays, the call for actual real-time delivery of EO products to the end-user is gaining traction in many fields. The technological advances of commercial off-the-shelf (COTS) components and their employment onboard next to space-grade hardware has been proven to enable extremely low-latency delivery of EO products to the end-user. The EO-ALERT project promises next-generation satellites capable of delivering high-priority EO products to the end-user in less than $5\,\mathrm{min}$ in all foreseen application scenarios as discussed in [5]. The results obtained during the preliminary tests with the reference implementation of the EO-ALERT system have confirmed the feasibility of this goal.

When dealing with such strict latency requirements, optimizing onboard data handling, operation and transmission scheduling, and compression-encryption functions become a priority. For this reason, special effort was spent in developing a modular, configurable, and powerful CPU Scheduling, Compression, Encryption, and Data Handling (CS-CEDH) Subsystem. The CS-CEDH Subsystem, already described in [6] and [7], is the centrepiece of the EO-ALERT System and essentially fulfils two roles: 1. acquire and move images and products among the image processing and communications subsystems, therefore also coordinating their tasks; 2. compress and encrypt the input and output data with different settings depending on the mission requirements. From an optimal design and resource allocation perspective, these aspects are complementary. The former is software-focused, aiming at maximizing modularity, flexibility, and dynamic scalability, required by the inherent system-level real-time event-driven nature of the CPU Scheduling processes. The latter represents an intrinsically highly-specialized, computationally expensive data-processing function better suited for hardware implementation. In order to achieve the overall goal of minimizing the system-level latency, it is therefore mandatory to effectively co-design a mixed hardware/software solution, leveraging the performance of COTS Multi-Processor System-on-Chip (MPSoC) devices featuring a Processing System (PS) directly interfaced to a Programmable Logic (PL) unit. Such platforms enable, thanks to state-

of-the-art Electronic Design Automation (EDA) tools and methodologies, to obtain a Register-Transfer Level (RTL) model (to be deployed on the PL unit) directly from a high-level hardware-oriented software model through High-Level Synthesis (HLS), thereby effectively shifting from a software-only design to a more efficient and high-performance hybrid hardware/software one, without sacrificing run-time tuning of compression and encryption parameters and still meeting all the system-level requirements.

Section 2 describes how the functionalities of the CS-CEDH Subsystems are mapped on the selected target MPSoC; Section 3 motivates and presents the implementation of the CS-CEDH software and its hardware image compression accelerator. The timing performance of software and hardware compression and encryption functions are reported in Section 4, together with an overview of the overall contribution of the CS-CEDH Subsystem to the latency of the EO-ALERT image data chain. Finally, Section 5 summarizes the contributions of this work.

2. CS-CEDH TASKS MAPPING

To expand the description of the CS-CEDH role from Section 1, its tasks inside the EO-ALERT System can be summarized as follows: 1. Schedule all the operation involved throughout the entire scenario-specific acquisition sequences, from the acquisition of the sensor payload from a Sensor Board to the transmission of the generated data to the Communications Subsystem. 2. Exchange control information and data with the optical and Synthetic Aperture Radar (SAR) Processing Boards. 3. Implement compression and encryption functionalities for all the input and output data types. 4. Provide a suitable data transmission policy that aims at reducing the latency of alerts delivery as much as possible. 5. Access and manage the on-board storage device (SSD), deciding what data to keep. Compression and encryption data-driven functionalities imply the most computationally expensive operations. The other tasks can be classified as control functions and data transfers that represent sparse and short workloads (bursts). Therefore, the image compression and encryption routine for raw and generated image data was selected to be implemented on the PL of the CS-CEDH MPSoC device. This choice ensures the best compression and encryption performance for the largest data types (i.e. raw and generated images) in the EO-ALERT data chain and leaves most of the MPSoC PS processing power available for the other tasks, with significant advantages in terms of both throughput, system responsiveness, and ultimately latency. The compression and encryption of non-image data types such as alerts and geo-localization data (orders of magnitude smaller than image data) can still be performed in software with general-purpose methods without impacting on the overall system performance and responsiveness, provided that the multi-core architecture of the PS general-purpose microprocessor is exploited properly. The software also takes care of all those tasks related to data transfers, storage management and transmission or operation scheduling.

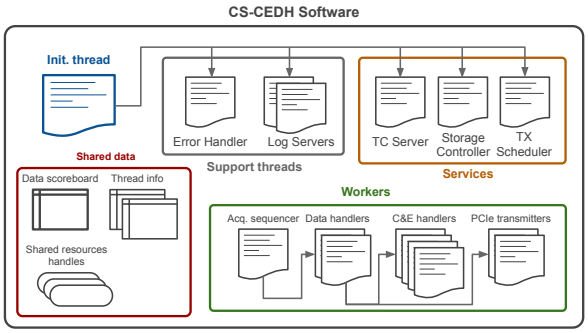

Figure 1: CS-CEDH SW architecture. Always-running services handle the synchronization with the surrounding subsystems and support features such as error handling and logs. On-demand worker threads implement data handling, compression and encryption tasks on the PS or PL of the CS-CEDH MPSoC.

3. SOFTWARE AND HARDWARE DESIGN AND IMPLEMENTATION

To fit all the functionalities listed in Section 2, a Xilinx® Zynq® Ultrascale+ ZU19EG MPSoC was selected as the target platform. This section briefly introduces the software architecture of the CS-CEDH Subsystem and the HLS design methodology exploited to implement the image hardware accelerator.

3.1. Data handling software

Because of the large number of functions that the CS-CEDH SW must implement, a single-threaded software architecture may introduce unwanted delays, especially while waiting for intensive I/O operations to complete or when executing intensive workloads like software compression or encryption routines, degrading the system responsiveness and increasing the overall latency. Besides, a purely sequential approach would increase the code complexity (with consequences on code maintenance and stability) and make error detection and handling much harder. A single-thread approach would also fail to exploit the full processing power of the fully-featured quad-core ARM® Cortex® A53 microprocessor available in the PS of the CS-CEDH MPSoC when performing computationally expensive tasks. So, a multi-thread software architecture was chosen instead. This approach also allows to handle the communication and data transfers towards the connected subsystems (e.g., communications and image processing) asynchronously, and therefore to handle data and requests as soon as possible. A simplified representation of the CS-CEDH thread architecture is reported in Fig. 1.

To better support such complex software architecture and provide convenient access to the communication and storage capabilities of the target device, a minimal Linux kernel was compiled and deployed on the PS Central Processing Unit (CPU) instead of relying on a bare-metal software implementation. The versatility of the Linux kernel also allows to easily tune the execution priority of each thread to optimize the latency of critical tasks, such as alert routing, encryption, and transmission.

General-purpose compression functionalities were implemented using the well-known *deflate* method from Zlib [8], while general-purpose encryption is achieved using a stream cypher based on an assembly optimized implementation of the Keccak sponge function [9].

3.2. Image compression-encryption hardware accelerator

On the EO-ALERT CS-CEDH Subsystem, image compression and encryption is achieved using an algorithm based on the Consultative Committee for Space Data Systems (CCSDS) 123.0-B-2 recommended standard explained in [10] and [11], and extended to embed encryption inside the compression throughput with negligible overhead on the compression rate as detailed in [12]. The availability of a reference implementation written in the common C language, already validated and largely characterised and tested, compared to the complexity of validating a brand new hardware implementation described from scratch at RTL level, strongly incentivized the decision to leverage state-of-the-art EDA tools to synthesize a hardware accelerator starting from a C model through HLS. HLS is well-known to have several benefits w.r.t. manual description using Hardware Description Languages (HDLs), all contributing to obtain high-quality implementations in a shorter time and with minimal effort. The validation and characterisation of the obtained Intellectual Property (IP) core can be easily performed by reusing the same test and validation suite defined for the reference software model, and most of the run-time reconfigurability of the original code can be maintained. The methodology followed to develop the hardware image compression accelerator can be summarized in these steps: 1. Develop a test suite aimed at continuously verify the behaviour and estimated run-time performance of the hardware-oriented C model with respect to the original one. 2. Establish a subset of the above test dataset to verify the correctness of the synthesised RTL model. 3. Rework the reference C code into a hardware-oriented model that implements the desired features and meets the system-level requirements. 4. Further modify this model to best exploit the hardware resources of target platform. 5. Evaluate the run-time hardware performance of the new model and iterate the previous step to further improve the throughput. 6. Wrap the hardware implementation into an IP core to be instantiated on the PL of the target platform. 7. Develop a software Application Programming Interface (API) to access the IP core from the software running on the PS of the target platform. 8. Validate the instantiated IP core and assess its performance. The hardware-oriented C model maintains most of the run-time configurability of the reference model. Among others, the hardware compressor supports arbitrary image dimensions up to 8-band, 8192 by 32768 images with arbitrary dynamic range up to 16 bits/pixel, configurable maximum absolute quantization error from 0 (lossless) to 32 (near-lossless), and the possibility to disable encryption.

The hardware compressor was synthesized from the hardware-oriented C model using the Xilinx® Vivado HLS 2018.3 tool and then deployed on the PL of the Zynq® MPSoC using Xilinx® Vivado. Several timing constraints were investigated by trying several clock frequencies from $100\,\mathrm{MHz}$ to $300\,\mathrm{MHz}$. In the final design, the clock frequency was set to $200\,\mathrm{MHz}$ since this solution achieves the best compression and encryption throughput without resulting in excessive resource utilization (for further details, see **(author?)** 6).

The hardware accelerator is made available to the software running on the PS CPU as a user-space I/O device. The low-level interface with the PS is implemented as follows: a $32\,\mathrm{bit}$ Advanced eXtensible Interface (AXI) 4 Lite Slave to exchange configuration (e.g., pixel array address in memory, image dimensions, and compression level) and control (e.g., start, stop) signals; a $128\,\mathrm{bits}$ high-performance AXI4 interface to directly fetch and store data inside the PS main memory without requiring any intervention from the CPU; low-level signals (clock, reset, and interrupts). An overview of the interfaces between the PS and the PL inside the CS-CEDH MPSoC is reported in Fig. 2.

The resulting IP core uses 32915 ($50\,\%$) of the available Configurable Logic Blocks (CLBs), 196087 ($38\,\%$) of the CLB Look-Up Tables (LUTs), 149406 ($14\,\%$) of the CLB Flip-Flops (FFs)[1], 268 ($27\,\%$) of the block Random-Access Memory (RAM) tiles and 330 ($17\,\%$) of the Digital Signal Processors (DSPs). The higher usage of combinational components (CLB LUTs) reflects the computational complexity of the selected compression and encryption algorithm [6].

4. RESULTS

The CS-CEDH Subsystem described in Section 3 and deployed on the target platform was tested with real-world data and compared with a software-only version running entirely on the quad-core ARM® Cortex® A53 microprocessor available in the PS. In the latter version, the hardware image compression accelerator was replaced by the optimized C model from which it was synthesized. In the test setup, all the subsystems connected to the CS-CEDH board (see Fig. 2) were replaced by dedicated software emulators running on a common PC connected

[1] In the Xilinx® Ultrascale+ architecture, a CLB is composed of 8 LUTs and 16 FFs.

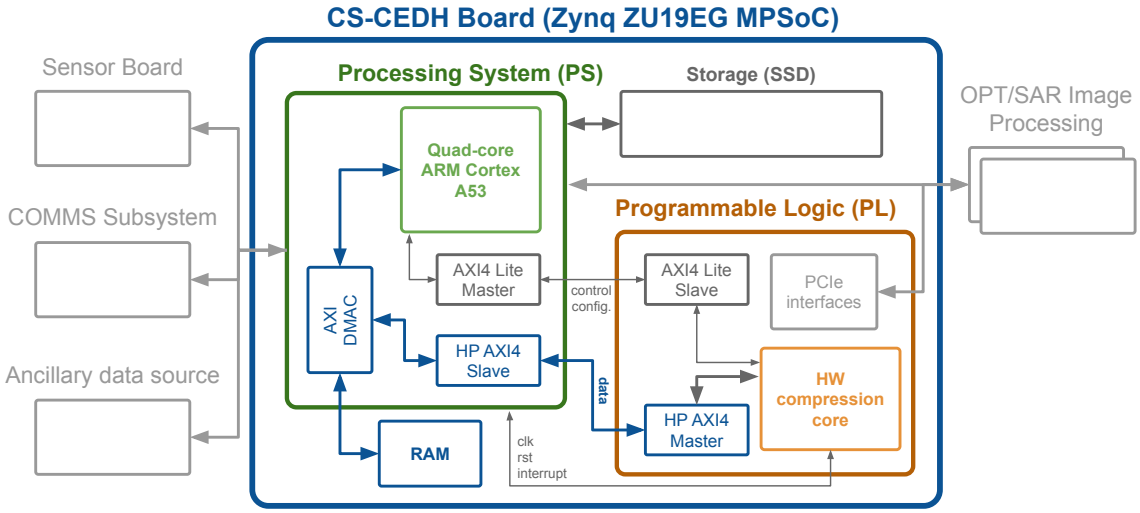

Figure 2: CS-CEDH physical architecture and interfaces.

to the MPSoC via Gigabit Ethernet interfaces. All the data transfers between the CS-CEDH board and the image processing emulators were skipped as their latency was measured to be negligible compared to the other tasks[2]. All the input files and the image processing products were therefore fetched from the board storage device. The complete test setup is represented in Fig. 3.

The dataset was defined to cover all the foreseen application scenarios and associated data types, as reported in the following paragraphs. All the image data files also contain generic data headers and geo-localization information that is compressed and encrypted in software (SW) using the general-purpose routines mentioned in Section 3.1.

Optical ship detection scenario 20 optical high-resolution 8400 by 6000 (50 Mpixel) raw data with 16 bit samples; 20 8 bit generated images with the same dimensions; 100 generated alerts (10 kB each).

Optical extreme weather detection scenario 10 combined 5-band 639 by 1119 (0.7 Mpixel) and single-band HRV 1917 by 3576 (6.9 Mpixel) raw data with 16 bit samples; 10 combined generated images with the same dimensions and embedded floating-point projections; 4 generated alerts (20 kB each).

SAR scenario 3 SAR single-band 8192 by 26000 raw data with 16 bit complex samples[3]; 3 16 bit generated images 3000 by 10000; 52 generated alerts (10 kB each).

Each of the above datasets was used twice with the minimum and maximum supported compression levels (maximum absolute quantization error set to 0 and 32, respectively). The compression and encryption throughput for each data type of the software-only and hardware-accelerated versions of the CS-CEDH is reported in Tables 1a and 1b, respectively. As shown, thanks to the hardware accelerator, the CS-CEDH can compress the raw data and the generated images 5 to 7 times faster than the reference software implementation. To better visualize the image compression and encryption speed-up enabled by the hardware accelerator, the compression and encryption throughput for the raw data, normalized with respect to the software image compression throughput, is reported in Fig. 4.

Table 1 also shows that software alert encryption, despite being scheduled with a higher priority than image compression and encryption, can still indirectly benefit from hardware-accelerated image compression by taking advantage of the additional CPU resources no longer employed for image compression. The alert encryption throughput is not only significantly higher in the hardware accelerated CS-CEDH system but also more consistent in time, as shown by the lower standard deviation. This is particularly true under heavy load condition such as in the optical ship detection scenario, where multiple images are compressed and encrypted at the same time with the consequence of overloading the CPU in the software-only version, therefore causing significantly lower alert encryption performance, with a throughput that was observed to be as low as 1 MiB/s in some cases. Therefore, the contribution of the CS-CEDH to the alert data chain is indirectly reduced up to an order of magnitude in the hardware-accelerated version. Similar effects were observed on other software-based data handling tasks, such as data forwarding to the communications subsystem. The combination of the indirect

[2]On the complete EO-ALERT system, data transfers between the CS-CEDH and the Image Processing Subsystem are performed using high-performance PCIe links with a throughput of about 2 Gbit/s.

[3]The real and imaginary pixel arrays are processed separately by the hardware compressor. Image pre-processing has negligible impact on the compression latency.

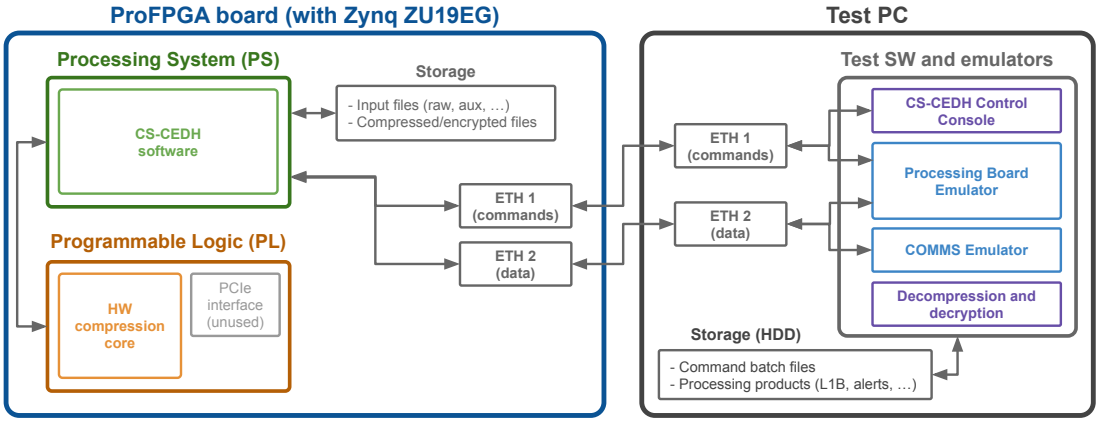

Figure 3: CS-CEDH test setup.

Thr. [MiB/s]	Optical single-band	Optical multi-band	SAR
raw (MAE=0)	2.91	2.83	2.74
raw (MAE=32)	3.19	2.81	3.18
gen (MAE=0)	1.55	2.78	2.75
gen (MAE=32)	1.60	2.92	3.04
alerts (SW)	9.4 ± 1.9	3.91 ± 0.10	10.4 ± 1.0

(a) Software-only CS-CEDH.

Thr. [MiB/s]	Optical single-band	Optical multi-band	SAR
raw (MAE=0)	16.42	13.79	14.58
raw (MAE=32)	19.40	15.34	19.35
gen (MAE=0)	9.35	14.28	14.13
gen (MAE=32)	9.68	16.16	18.23
alerts (SW)	13.6 ± 0.7	3.86 ± 0.10	12.2 ± 0.3

(b) Hardware-accelerated CS-CEDH.

Table 1: CS-CEDH compression and encryption throughput. Relative standard deviation for image data types is always less than $0.5\,\%$ and therefore omitted.

speed-up of the accelerated image data chain allows the CS-CEDH contribution to the alert data chain to be lower than $1\,\mathrm{s}$ in all the foreseen scenarios (with up to 100 alerts per acquired image).

The overall contribution of the CS-CEDH Subsystem to the entire acquisition data chain is reported in Table 2 for each test scenario, together with the latency reduction achieved thanks to the hardware image compression accelerator. As shown, the optical ship detection and SAR scenarios greatly benefit from hardware-accelerated image compression, resulting in a four times lower latency than the software-only version. In the extreme weather detection scenario, the reduction in the image compression and encryption latency causes this task to no longer be in the critical path for the image data chain. The reason is that extreme weather image

	SW-only latency [s]	HW-accelerated latency [s]	latency reduction
optical ship	1328	316	4.2x
optical extreme	169	67	2.5x
SAR	959	225	4.3x

Table 2: Overall latency reduction.

data also contains a large set of floating-point projections that must be compressed and encrypted in software with general-purpose functions, whose latency is higher than hardware-accelerated image compression and encryption. Table 2 also shows that the contribution of the hardware-accelerated CS-CEDH Subsystem to the overall data chain latency is compatible with the target $5\,\mathrm{min}$ latency for product delivery. In the worst case (SAR scenario), the CS-CEDH takes up to $80\,\mathrm{s}$ to compress and encrypt all the acquired and generated data.

5. CONCLUSION

In this work, we presented the design and implementation of a combined software/hardware data handling, compression and encryption Subsystem for EO satellites. It was shown that by leveraging state-of-the-art EDA tools and HLS, it is possible to accelerate onboard data compression and encryption by 5 to 7 times without sacrificing run-time configurability and with significant benefits for the alert data chain. The proposed hardware compression accelerator achieves a throughput of $15\,\mathrm{MiB/s}$ to $20\,\mathrm{MiB/s}$ while offloading the most computationally expensive operations from the onboard CPU. Consequently, the proposed CS-CEDH Subsystem is able to meet the latency requirements for true real-time alert delivery to the end-user and very low-latency delivery of images to the GS, overcoming the limitations of traditional onboard data handling systems.

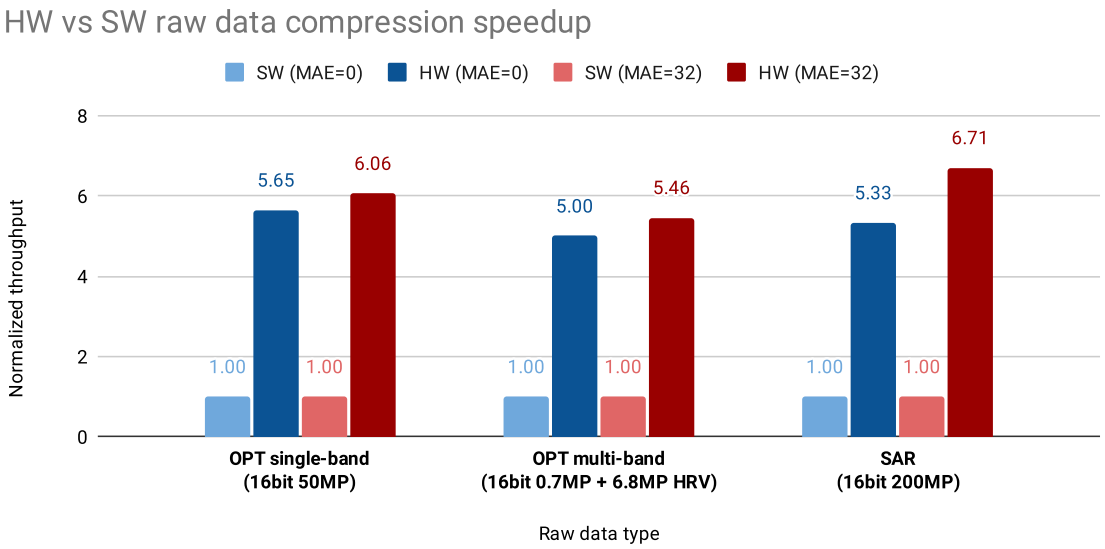

Figure 4: Image compression and encryption speedup.

ACKNOWLEDGMENTS

This work and the EO-ALERT project have been funded and supported by the European Union Horizon 2020 research and innovation programme under grant agreement No. 776311.

REFERENCES

[1] Josef Aschbacher. Esa's earth observation strategy and copernicus. In *Satellite earth observations and their impact on society and policy*, pages 81–86. Springer, Singapore, 2017.

[2] Ramon Torres, Paul Snoeij, Dirk Geudtner, David Bibby, Malcolm Davidson, et al. Gmes sentinel-1 mission. *Remote Sensing of Environment*, 120:9–24, 2012.

[3] Matthias Drusch, Umberto Del Bello, Sébastien Carlier, Olivier Colin, Veronica Fernandez, et al. Sentinel-2: Esa's optical high-resolution mission for gmes operational services. *Remote sensing of Environment*, 120:25–36, 2012.

[4] D. K. Davies, M. E. Brown, K. J. Murphy, et al. Workshop on using NASA data for time-sensitive applications [space agencies]. *IEEE Geoscience and Remote Sensing Magazine*, 5(3):52–58, 2017.

[5] M. Kerr, S. Tonetti, S. Cornara, et al. EO-ALERT: A novel architecture for the next generation of earth observation satellites supporting rapid civil alerts. In *71st International Astronautical Congress (IAC)*, 2020.

[6] Paolo Motto Ros, Michele Caon, Tiziano Bianchi, Maurizio Martina, and Enrico Magli. High-level synthesis of a single/multi-band optical and sar image compression and encryption hardware accelerator. In *2021 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2021. accepted.

[7] Michele Caon, Paolo Motto Ros, Maurizio Martina, Tiziano Bianchi, Enrico Magli, et al. Very low latency architecture for earth observation satellite onboard image data handling, compression, and encryption. In *2021 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2021. accepted.

[8] P. Deutsch. RFC1951: Deflate compressed data format specification version 1.3, 1996.

[9] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Keccak sponge function family main document. *Submission to NIST (Round 2)*, 3(30):320–337, 2009.

[10] A. Kiely, M. Klimesh, I. Blanes, et al. The new CCSDS standard for low-complexity lossless and near-lossless multispectral and hyperspectral image compression. In *ESA On-Board Payload Data Compression Workshop, OBPDC*, September 2018.

[11] D. Valsesia and E. Magli. High-throughput onboard hyperspectral image compression with ground-based CNN reconstruction. *IEEE transactions on geoscience and remote sensing*, 57(12):9544–9553, 2019.

[12] A. Migliorati, T. Bianchi, and E. Magli. Selective encryption in the CCSDS standard for lossless and near-lossless multispectral and hyperspectral image compression. In Lorenzo Bruzzone, Francesca Bovolo, and Emanuele Santi, editors, *Image and Signal Processing for Remote Sensing XXVI*, volume 11533, pages 221 – 228. International Society for Optics and Photonics, SPIE, 2020.