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

OBPMARK (ON-BOARD PROCESSING BENCHMARKS) – OPEN SOURCE COMPUTATIONAL PERFORMANCE BENCHMARKS FOR SPACE APPLICATIONS

David Steenari[1], **Leonidas Kosmidis**[3,2], **Ivan Rodriguez-Ferrandez**[2,3], **Alvaro Jover-Alvarez**[2,3], **and Kyra Förster**[1]

[1]*European Space Agency, ESTEC, The Netherlands*
[2]*Universitat Politècnica de Catalunya (UPC), Barcelona, Spain*
[3]*Barcelona Supercomputing Center (BSC), Barcelona, Spain*

ABSTRACT

Computational performance benchmarking of on-board processing (OBP) applications has often been done in a case-to-case basis, taking into account only a small subset of devices and specific, often proprietary, applications, limiting domain coverage and reproducibility. While commercial benchmarks exist for embedded systems, they are usually limited to CPUs and are based on synthetic algorithms non-relevant for space. Consequently, they are not generally suitable for assessing highly parallel processors (GPUs, DSPs, etc.) and/or hardware implementations (i.e. ASICs and FPGAs) which are commonplace in space systems.

In the space domain, there are a number of OBP applications which reoccur over multiple missions. Such applications are often driving the overall computational requirements of the mission, e.g. in the case of image and radar processing, RF signal processing and compression. There are certain performance metrics in each case – such as the number of pixels processed per second – which are well-known and easily understandable by equipment designers and customers.

With the recent rise of machine learning applications in on-board space applications, tasks such as image classification and object detection using SVMs and CNNs are becoming commonly used. These new processing methods put additional requirements on on-board systems, and must be understood in terms of their specific performance parameters.

In this paper, OBPMark (On-Board Processing Benchmarks) is introduced. OBPMark defines a set of benchmarks covering the typical classes of applications commonly found on-board spacecraft. The benchmark suite is publicly available to enable easy comparison of different systems and to quickly down-select possible processing solutions for a mission. It is open source and includes multiple implementations, as well as easily extensible, which allows it to be ported and optimized to target platforms, including heterogeneous ones, for a fair comparison. Currently, implementations in standard C, OpenMP, OpenCL and CUDA are provided.

1. INTRODUCTION

In the last years the number of different device used for OBP is increasing, driven mainly by the the availability of COTS components implemented in deep sub-micron process nodes. Heterogeneous system-on-chip devices are now available that integrate several types of processors and accelerators, such as multi-core clusters, task-specific DSP processors, embedded GPUs, and dedicated accelerators for tasks such as FFTs, image/video compression, encryption and NN (neural network) inference processing. A survey of processor and FPGA devices that are being used or proposed for use in on-board processing is outlined in [1].

While FPGAs are still the most commonly used device type for on-board tasks that require a high performance, massively parallel devices are now also used in space, including many-core devices, stochastic arrays, GPUs (graphics processing units), VPUs (visual processing units). The introduction of massively parallel processing devices poses its own challenges for performance comparisons. Using traditional metrics such as MIPS or MFLOPS as a metric for the performance of all processing cores, only depicts the peak theoretical performance as if all core units were utilized at the same time, and does not take into account issues related to memory throughput and synchronization. Considering the architectural differences between such devices, it is becoming increasingly difficult to accurately assess their relative computational performance for representative OBP applications, when considering only traditional metrics of peak performance.

Extending the scope to compare processors to FPGAs, simple performance metrics are difficult to find. For instance, one could take the number of MACs or DSP blocks provided by an FPGA and multiply by the maximum clock frequency to get an estimate of the maximum theoretical performance, but again – such numbers would not be feasible to achieve for a real algorithm implementation. More realistic performance measurements should take into account accelerator soft-core IPs for specific processing tasks. As this requires implementing the specific algorithm as dedicated IP, comprehensive performance benchmarking can become quite time demanding.

Certain systems may be optimized for increased performance for highly parallel tasks (such as GPUs), while others offer higher performance for pipelined operations or bit-manipulation tasks (such as FPGAs). Finally, heterogeneous systems provide multiple means of implementing the same processing tasks, or even to divide certain pipelines of processing tasks over processor cores and accelerators with different architectures, depending on which is most efficient for the specific pipeline stage.

Processing performance benchmarks for space application are commonly made on a case-to-case basis, targeting actual software implementations to only on a few target devices based on schedule and effort limitations, due to the inherent complexity of implementing full applications on several devices with different architectures and software models. While these benchmarks target similar application cases, it is difficult to compare the results, due to the large amount of test parameters that may vary from case to case. Even in the case of standardized processing tasks, such as CCSDS image compression, published performance results of different software and hardware implementations are difficult to compare – as both compression settings and input data sizes may affect the performance results. As such, these compressor throughput results are usually accompanied with clarifications of the settings used.

In recent years, ESA has funded several activities for application benchmarking of different processing devices. The ESA-funded "HIPNOS" study aimed at benchmarking a VBN (visual-based navigation) algorithm on different processors and FPGAs. In the conclusions of the study, MPSoCs (multi-processor system-on-chips) were found to be the most favorable for the application targeted. In addition, SoC with processors and accelerated specifically for image processing were also identified as particularly energy-efficient for VBN tasks [2].The ESA-funded "HP4S" study aims to benchmark several multi- and manycore processors, such as the quadcore GR740 LEON4 processor. In the activity, software running OpenMP, to enable parallel execution over multiple cores, was used to measure and optimize the performance of the target applications [3]. The currently on-going ESA-funded "MLAB" activity is targeting the benchmarking of machine learning for several FPGAs and processors using space applications [4].

The ESA-funded "GPU4S" study aims to evaluate low-powered GPUs for the use in space [5][6]. In the study, several embedded GPU SoCs have been evaluated and benchmarked. A survey of typical on-board processing tasks among multiple space domains was conducted. Each task was divided down into building blocks (such as FFT, convolution, etc.) and a selection of building blocks was done for implementation – aiming to cover as many of the identified applications as possible. In addition, an application with a complete processing pipeline, based on the on-board processing algorithms from the ESA Euclid mission, was implemented [7]. In the study, the lack of openly available benchmarks for space applications was identified. As such, it was decided to release the benchmark suite that was implemented targeting processing building blocks for embedded GPUs as the "GPU4S Bench" [8]. In addition, it was decided to continue the work on an openly available application-level benchmark in the later parts of the activity, in coordination with internal work carried out at ESA on the implementation of on-board processing applications based on on-going work of several ESA missions. The work resulted in the OBP-Mark (On-Board Processing Benchmarks) suite, which is presented in this paper.

2. COMPARISON TO OTHER BENCHMARKS

2.1. Comparison to Existing Benchmarks

Traditionally, in the field of space processors, metrics such as MIPS (millions of instructions per second) or MFLOPS (millions of floating point operations per second) have been used to determine the peak processing performance of processors. DMIPS (Dhrystone MIPS) – or DMIPS/MHz – is a commonly used used benchmark for processors, and is also referred to for space processors. It targets general CPU integer performance and is based on synthetic applications, covering topical computational loads. It is highly compact and portable, and has become a de-facto standard for benchmarking CPUs [9]. However, such processing loads may not be fully representative of typical satellite on-board high-performance processing tasks - and they serve only for the benchmarking of processors.

The EEMBC (Embedded Microprocessor Benchmark Consortium) has released several benchmarks for processors and embedded systems. The most widely used is the CoreMark, which is intended to replace the Dhrystone benchmark [10]. As Dhrystone, it is also based on synthetic applications, and addresses issues with Dhrystone such as complications to compiler optimizations and the fact that there is no standard way of reporting Dhrystone results. CoreMark has been used in the space domain, e.g. [11] and [12] provides CoreMark results of single- and multicore LEON processors. While such benchmarks are useful for classic CPU-type processing devices, it is not suited to compare CPUs (central processing units) to other types of processing devices, such as FPGAs, GPUs and ML accelerators.

Other EEMBC benchmarks, such as Multibench for multicore processor performance and FPMark for multi-threaded floating point performance are provided by EEMBC. Two benchmarks for heterogenous systems have also been released, ADASMark and MLMark. ADASMark targets advanced driver-assistance systems (ADAS) tasks for autonomous driving in SoCs (system-on-chips). It includes a image processing pipeline consisting of typical pre-processing (debayer, dewarp, color convert, etc.) as well as object detection (e.g. sobel edges) which are common for ADAS systems. The benchmark is released in OpenCL, targeting CPUs, GPUs and DSPs. While both the approach and image processing in general are close to processing done on-board spacecraft, it does not fully map to the specific image processing tasks for space applications. As such, it can be

useful for general image processing benchmarking, but it is not fully representative for space systems.

EEMBC's MLMark is a benchmark that targets machine learning (ML) tasks on the edge. [13] It includes implementations for TensorFlow/TensorFlow Lite and dedicated implementations for a number of devices, such as: Intel Myriad devices (OpenVINO); NVIDIA GPUs (TensorRT); ARM Cortex-A processors (ARM NN library); and ARM Mali GPUs (OpenCL); Google Edge TPU – which have all been proposed for the use in space. The ML workloads are based on de-facto standard deep neural network architectures: ResNet-50 v.10, MobileNet v1.0, and SSDMobileNet V1.0. Such model architectures are also expected to be used in space. However, the models have been trained on ILSVRC2012 (ImageNet Large Scale Visual Recognition Challenge 2012) and COCO2014 (Common Objects in Context), which are not related to space. In addition, they do not take into account the necessary pre-processing stages required to execute such applications using typical sensor sizes used in space (i.e. by downsampling or tiling).

MLPerf is a set of benchmarks aimed at the computational performance evaluation of machine learning (ML) tasks, initiated by a consortium of researchers. It includes includes both benchmarks for training and inference of ML models [14]. While both MLPerf and OBPMark include benchmarks for ML inference, such as image classification and object detection, the inference models used in MLPerf (as those used for MLMark) are not targeted at space applications. Instead MLPerf uses image data sets such as ImageNet and COCO, which both target general visual object recognition with large networks. The ML inference computational benchmarks of OBPMark are targeting specific applications related to space imaging and Earth Observation that are suited (both in terms of overall complexity and memory footprint) for execution on space hardware.

The NAS Parallel Benchmarks (NPB) from NASA [15] are intended for the evaluation of the performance of highly parallel supercomputers. The benchmarks are taken from applications in the field of computational fluid dynamics. The application cases of NPB are not commonly used in on-board processing applications – which is the target of OBPMark. While both benchmarks share e.g. an implementation of FFT (Fast Fourier Transform), there is otherwise little other overlap between the benchmarks both in terms of target and choice of benchmarks.

2.2. Relation to ESA NGDSP Software Benchmarks

A set of benchmarks for DSPs on-board spacecraft were previously defined in 2008 in the "Next Generation Space Digital Signal Processor Software Benchmark" ("NGDSP benchmarks") document [16] which was mainly intended for the selection of DSP devices and architectures as part of the now cancelled NGDSP initiative. It also included aspects related to I/O data transfer rate and acquisition from mixed signal data converters. OBPMark succeeds and replaces the NGDSP benchmarks. Wherever possible, benchmark elements and reporting parameters have been reused from the NGDSP benchmark, to be able to retain results from previous benchmarks at best effort. OBPMark focuses only on the digital processing aspects. Note that with the introduction of of OBPMark, the use of the NGDSP Software Benchmarks is no longer recommended.

2.3. Relation to GPU4S Bench

The GPU4S Bench was developed for the purpose of evaluating highly parallel processors, such as GPUs for use in space applications. It includes implementations of multiple optimized kernels for the execution of GPUs and other processors, capable of executing parallel code using standard frameworks such as OpenCL and/or CUDA [8] as well as OpenMP [17]. In addition to simple algorithmic building blocks used in multiple space domains, GPU4S Bench includes also implementation of neural network layers and a CIFAR-10 inference chain.

The GPU4S Bench is closely related to OBPMark: some algorithmic building blocks are shared between the two benchmark suites. Specifically implementations of optimized kernels for e.g. FFT and FIR filtering from GPU4S are reused in OBPMark. Other benchmarks in OBPMark are mainly targeting multi-stage algorithms, which reuse the optimized parallel kernel implementations from the GPU4S Bench. GPU4S Bench and OBPMark are provided as complementary benchmarks and they are hosted together. They share the same benchmark structure and the same optional automation system for facilitating benchmarking. Due to their relation, when porting them to new architectures, it is recommended to start from GPU4S Bench blocks and reuse them in the complex OBPMark chains.

3. OBPMARK OVERVIEW

OBPMark (On-Board Processing Benchmarks) has been initiated by ESA together with BSC to define a set of benchmarks covering applications commonly found on-board spacecraft. Five categories of benchmarks are defined 1) Image Processing Pipelines; 2) Standard Compression Algorithms; 3) Standard Encryption Algorithms; 4) Processing Building Blocks; and 5) Machine Learning Inference. In each category, specific benchmarks are included, e.g. both image and radar image compression. The processing building blocks include e.g. FIR filters and FFT processing. In all the OBPMark consists of the following components:

1. Technical Note (TN) defining the benchmark algorithms and result reporting
2. Reference input and output data for verification
3. Reference implementations
4. Database of reported benchmark results

The TN contains the descriptions of the benchmark algorithms (or in the case of standard algorithms, references to the appropriate documents describing the algorithms) to a level that an implementer can opt to implement a specific benchmark without referring to any of the reference

implementations. The reference input data is provided to be used during both verification and performance benchmarking.

Several reference implementations are provided: a "golden model" in standard C, without any parallelization or optimizations; parallel implementations using standard parallelization software frameworks are provided to lower the porting effort of the benchmark suite to many multicore processors, GPUs, etc. OBPMark, as GPU4S Bench, features the same optional automation framework which facilitates the compilation, execution and result collection of its benchmarks. A database of reported performance test results per device and benchmark will be maintained as part of the OBPMark repository.

3.1. Selection of Benchmarks

Currently the processing requirements for on-board processing systems are driven by a number of key applications. These include image processing; multi- and hyperspectral image processing; SAR (synthetic aperture radar) processing; data compression and encryption; radio signal processing; etc. In the field of image processing, particular processing intensive tasks include: co-registration of successively acquired images; scrubbing of radiation effects in the detector; and image compression. Traditionally, the purpose of on-board image processing is to calibrate and correct the image as to efficiently remove non-homogeneity that can negatively affect the (lossless) compression performance. In the field of micro-satellites, further image processing and data reduction methods have been deemed necessary to meet data budget requirements. In recent times, there has also been an increasing interest for deep learning applications on-board, in particular for classification and segmentation of images.

Note that the OBPMark suite does *not* target the low-level aspects of performance benchmarking of processor cores, but focuses instead on the *system-level* and *application-level* performance of the target well-known (in the space community) processing applications.

3.2. Benchmark Objectives and Requirements

The objective of these benchmarks are:

- To provide a suite of *application level* benchmark for space on-board applications.
- To promote a standard set of benchmarks, as to enable a method of comparing end-user performance of different devices and systems – such as both RHBD and COTS processors, FPGAs and ASICs.
- To better understand limitations of different types of devices and systems.
- To quickly decide the division of tasks in hardware and software for implementations in heterogeneous systems.
- To allow ESA to quickly provide recommendations for processing systems in future missions, through identifying key parameters together with the project teams.
- Benchmark standard on-board processing functions, so that implementers will have a reference of the expected performance and even the possibility for reusing the invested work in real-world use cases.

Four key requirements were considered for the definition of the benchmarks: application coverage, comparability, portability and openness. The benchmarks shall cover common OBP applications: image processing, compression, radar processing, encryption, signal processing and machine learning - and it shall be possible to add additional benchmarks in the future (through version updates). The benchmarks shall include metrics for comparing the results, including: overall performance, performance per power, and power dissipation of DUT - and dfine all necessary configuration parameters and test data.

For portability, the base version shall be provided in standard C, with additional ports using standard parallellization schemes (such as OpenMP, OpenCL and CUDA) and support the porting to FPGA implementations. For openness, the benchmark definitions and standard ports are provided as open-source on a public repository - to allow community response/feedback and contribution (e.g. additional ports). The benchmarks have been specified without a specific processing architecture or device in mind. However, the intention is to allow implementations in a multitude of device types, i.e. CPUs, DSPs, FPGAs, ASICs, GPUs, many-core devices, stochastic arrays, etc. – as well as heterogeneous systems. That is, systems that consists of more than one device type, such as combined CPU and FPGA systems, where a part of the processing is done in software in CPU and another part is done in logic in the FPGA.

3.3. Parallelization, Optimization and Porting

One important aspect of the applications is the parallelization scheme applied on the algorithm to allow efficient and full resource utilization of devices with parallel capabilities, such as multi- and many-core processors, as well as multiple DSP blocks in FPGAs. OBPMark does not define the parallelization schemes for the algorithms, as these will be significantly dependant on the type of device benchmarked. It is therefore up to the implementer to find the most suited parallelization scheme for the device they are targeting for a new port – and document the approach with the benchmark results. However, example parallelized implementations in common parallelization frameworks are provided as part of the code-base. These can be run as-is, or be optimized for specific targets. In a similar manner, in the case of multi- and many-core devices, the performance is often dependent of the use of intermediate software framework for task and data parallelization. Different parallelization frameworks for the same device may give significantly different results. It is therefore required to include information also any used software frameworks and libraries with the benchmark results. In the case of processors, there is usually significant gain in manual assembly optimization of the kernels part of the processing chains – especially for VLIW architectures and cores with vectorization modules. The type

level of software tools, i.e. use of assembly, compiler optimization flags, etc., should also be reported together with the benchmark results.

Overall, it is recommended to also document the implementation effort for the target device or devices when reporting benchmark results. It may be that a certain device gives better performance than another, but with the penalty of a larger effort in implementation. Such information is certainly very useful when considering the overall effort for the implementation of a processing algorithm in a certain system. In the case of FPGAs, it should also be considered that not all logic resources may be available in when the FPGA is integrated in a processing system, as dedicated data, control and memory interfaces will also be necessary to be included in the design. Hence, it is recommended to include with the benchmark results information regarding the overall FPGA system design and resources utilization (including any applied TMR strategy in the design).

One important aspect for space applications is the power consumption of a processing system. It may be that very good results can be achieved with a certain set of devices, but that additional effort is required to remove the heat dissipated to be able to operate the system. Conditionings regarding the temperature environment and power dissipation is therefor also considered important parameters. Another important aspect of systems on-board spacecraft is the radiation hardening and fault-tolerance. In the case the used device gives options of switching off fault-mitigation techniques this should be listed with the benchmark results and the power dissipation.

The implementation effort to cover all OBPMark benchmarks can be significant, particular in the case of FPGAs. However, it is not the intent that all benchmarks need to be implemented to report on the performance for a particular application case. In fact, some of the benchmarks have intentionally been split to allow reporting of the performance of a specific task – such as the standardized compression methods. In fact, it is not expected that all benchmarks are implemented in the targeting devices and systems, although we would welcome such implementation to make comparisons more compete.

4. OBPMARK BENCHMARKS

The OBPMark benchmarks are summarized in Table 1. In the sections below, an overview of each of the benchmarks are outlined and explained.

4.1. Benchmark #1.1: Image Calibration and Correction

Benchmark#1.1 is intended to represent typical on-board processing tasks necessary for imaging instruments in scientific remote sensing applications with panchromatic sensors, for instance for deep space telescopes where long exposure times are usually required. To overcome the limitations of the sensor, multiple frames are acquired from the front-end, which are then stacked/summed (also called ”temporal binning”) to form a final image. Prior to stacking several pre-processing stages are performed on the individual acquisition frames:

1. Image offset correction
2. Bad pixel correction
3. Radiation scrubbing
4. Gain correction
5. Spatial binning
6. Temporal binning

The computational performance of the benchmark includes the metrics: pixels/s, which is calculated as an average over the entire processing pipeline for a number of iterations, and pixel/s/W which is calculated by dividing the pixels/s by the average power consumption, as measured by sampling the power consumption during the time the processing is active.

4.2. Benchmark #1.2: Radar Processing

This benchmark is intended to cover the generation of images from raw radar data. It follows the range-Doppler algorithm. The following stages shall be performed for each frame in a series:

1. Range Compression
   (a) Range FFT
   (b) Range Matched Filter Multiply
   (c) Range Inverse FFT
2. Corner turn (matrix transpose)
3. Azimuth Compression
   (a) Azimuth FFT
   (b) Range Cell Migration Correction (RCMC)
   (c) Azimuth Matched Filter Multiply
   (d) Azimuth Inverse FFT
4. Multilook (spatial binning)

Just as for Benchmark #1.1, the performance is measured as function of the number of samples/s that can be processed, averaged over the entire processing pipeline, and samples/s/W for performance per unit power.

4.3. Benchmark #2: Standard Compression Algorithms

Benchmark #2 is included to give a guideline on parameters and data sets to use for measuring the computing performance of standard CCSDS compression algorithms:

- #2.1: CCSDS 121.0 Data Compression
- #2.2: CCSDS 122.0 Image Compression
- #2.3: CCSDS 123.0 Hyperspectral Image Compr.

The benchmarks are based on the following CCSDS standards: [18], [19] and [20]. The implementations and parallelization of the CCSDS 121.0, 122.0, and 123.0 algorithms are based on the work described in [21]. Instructions and guidelines for the implementation of the standard compression algorithms are outlined in their respective CCSDS Blue Books and Green Books. Parameters selection for the CCSDS 123.0 standard has been based on the recommendations outlined in [22].

ID	Benchmark Name	Sub ID	Sub-Benchmark Name
#1	Image Processing	#1.1	Image Calibration and Correction
		#1.2	Radar Image Processing
#2	Standard Compression	#2.1	CCSDS 121.0 Data Compression
		#2.2	CCSDS 122.0 Image Compression
		#2.3	CCSDS 123.0 Hyperspectral Image Compression
#3	Standard Encryption	#3.1	AES Encryption
#4	Processing Building Blocks	#4.1	FIR Filter
		#4.2	FFT Processing
		#4.3	Convolution
		#4.4	Matrix Multiplication
#5	Machine Learning Inference	#5.1	Object Detection
		#5.2	Cloud Screening

Table 1. OBPMark benchmarks overview

Reference implementations for the compression benchmarks are provided (both sequential and parallelized). However, existing implementations from 3rd parties can be benchmarked by following the OBPMark guidelines for parameter selection and input/output data.

4.4. Benchmark #3: Standard Encryption Algorithm

In the "Standard Encryption Algorithms" the throughput of the standard encryption algorithms shall be measured. The "AES Encryption" benchmark implements the Advanced Encryption Standard (AES) encryption algorithm as per [23] and the related CCSDS standard [24], with 128-, 196- and 256-bit key-lengths. The performance metrics are: encrypted samples/s and samples/s/W.

4.5. Benchmark #4: Processing Building Blocks

The "Processing Building Blocks" benchmarks are include processing functions that can be found in multiple on-board processing applications such as optical image processing, radar processing, SDR (Software Defined Radio) processing as well as AOCS processing. The following sub-benchmarks hae been defined:

- Benchmark #4.1: FIR Filters
- Benchmark #4.2: FFT Processing
- Benchmark #4.3: Convolution
- Benchmark #4.4: Matrix Multiplication

In the first benchmark, one dimensional real and complex data shall be filtered with the use of FIR (Finite Impulse Response) filters. The target applications include RF and other on-board signal processing of instrument time signals that require filtering. FFT (Fast Fourier Transform) is a computationally efficient algorithm for DFT (Discrete Fourier Transform), introduced by Cooley and Tukey in 1965. FFT processing is used in a multitude of on-board applications in optical and radar imaging systems, telecommunications and as well as other RF applications. The radix-2, decimation in time FFT variant of the algorithm shall be used for all benchmarks. All of the provided reference benchmark implementations in Benchmark #4 are based on the GPU4S Bench implementations [8].

4.6. Benchmark #5: Machine Learning Inference

The "Machine Learning Inference" benchmark includes processing tasks that have been identified for use of artificial intelligence (AI) and machine learning (ML) onboard spacecraft. Training of machine learning parameters (e.g. for neural networks) is not expected to be made on-board, and is hence not included in this benchmark set.

A survey of openly available annotated training data sets and available standard DNN (Deep Neural Networks) architectures have been carried out, to identify possible application benchmarks that can be made openly available. Two sub-benchmarks have been tentatively defined:

- Benchmark #5.1: Object Detection
- Benchmark #5.2: Cloud Screening

The object detection benchmark will tentatively be based on using EO (Earth Observation) imaging data for ship (or airplane) detection. This application was chosen due to the availability of training data, and the fact that it has already been used on in several other ESA activities targeting demonstration of ML techniques on-board.

Cloud screening is a common application for EO optical instrument. As mentioned, it is already implemented on Φsat-1 (using deep neural networks) and will be used on CHIME (using SVMs). In OBPMark, it will be implemented as a DNN segmentation task.

The selected approach includes the use of standard models as much as possible (eg. such as SSD MobileNetV2 for object detection), to ease the implementation effort and support many tools and devices out of the box.

Outside of the ML inference, the benchmarks will possibly also include specific image pre-processing that is required on-board to adapt typical sensor data (in the range of 1024x1024 or 2048x2048 pixels) to sizes that are appropriate for inference. This will be done through either downsampling (through binning) or ROI selection (through patching).

Pre-trained models will be provided, based on standard

formats such as TensorFlow and TensorFlow-Lite. Moreover, both non-quantized and quantized models will be provided. As a baseline at least FP32, FP16, INT16 and INT8 models will be provided. The training data will be based on openly available data, so that in-case retraining of the model of or a specfic framework or device is needed, it will be possible.

Reference implementations and optimization for additional frameworks and specific devices (such as OpenVino for Intel devices; TensorRT for NVIDIA; ROCm for AMD; or VitisAI for Xilinx) may also be considered for future versions. In addition, OBPMark will aim to harmonize its approach with other on-going ESA activities that are targeting machine learning benchmarking of specific systems.

5. BENCHMARKS IMPLEMENTATION STATUS

In the first phase of the activity, the following benchmarks have been implemented:

- #1.1 "Image Calibration and Corrections"
- #2.1 "CCSDS 121.0 Data Compression"
- #2.2 "CCSDS 122.0 Image Compression"
- #4.1 "FIR Filter"
- #4.2 "FFT Processing"
- #4.3 "Convolution"
- #4.4 "Matrix Multiplication"

Reference implementations for the following benchmarks will be implemented in the next phase of the activity:

- #1.2 "Radar Image Processing"
- #2.3 "CCSDS 123.0 Hyperspectral Image Compr."
- #3.1 "AES Encryption"
- #5.1 "Object Detection"
- #5.2 "Cloud Screening"

For Benchmark #1.2, a tentative definition of the benchmark has been specified. It could be foreseen that some adjustments to the benchmark may be done after the first implementations have been performed.

In regards to Benchmarks #2.3 and #3.1, specifications of these benchmarks are based on already CCSDS standardized algorithms and implementers can already use the specifications provided in the OBPMark Technical Note for algorithm and data parameters for testing.

For the ML-oriented benchmarks, #5.1 and #5.2, they are as of writing in their specification stage, more details are provided above.

6. INITIAL BENCHMARKING TEST RESULTS

Initial benchmark tests of a number of devices have been performed with the beta version of the existing benchmarks. Test setups for each device was developed, including dedicated power measurements, for several COTS GPU SoCs. The result of four GPUs are presented here: NVIDIA Xavier, NVIDIA TX2, AMD Embedded Ryzen V1000 (V1605B) and HiSilicon Kirin 970 Hikey970, featuring an ARM Mali-G72 GPU. For fair comparison, all devices were operated at equivalent 15W TDP modes. Additional tests at lower TDP modes will be published in a later paper. Dedicated power measurements were performed in some of the cases to get a better estimate of actual power consumed. The number of cores used in the OpenMP multicore benchmarks was limited to four (4).

Results from Benchmark #1.1 are presented in Table 2. The three standard image sizes (as specified in the OBPMark technical note) were used: 1024x1024, 2048x2048 and 4096x4096. Quasi-random data, generated with a fixed seed were used as input data. The input data may be replaced with representative space imaging data in the future. As can be seen in the table, the NVIDIA TX2 and Xavier were the highest performing in GPU performance, whereas the AMD V1605B had the highest CPU performance. Please note that due to an unofficial custom driver issue causing issues with OpenCL performance, the results presented for the V1605B GPU tests are significantly lower than the expected value. This issue is expected to be corrected, which will result in higher results. The issue is currently under investigation.

Device	Target Impl.	Image Size	Mpixels/s	Mpixels/s/W
TX2	CPU (OpenMP)	1024	4.99	1.61
TX2	CPU (OpenMP)	2048	5.41	1.75
TX2	CPU (OpenMP)	4096	5.42	1.69
TX2	GPU (CUDA)	1024	13.10	3.80
TX2	GPU (CUDA)	2048	20.76	5.97
TX2	GPU (CUDA)	4096	34.86	9.95
Xavier	CPU (OpenMP)	1024	6.93	1.05
Xavier	CPU (OpenMP)	2048	8.83	1.35
Xavier	CPU (OpenMP)	4096	13.94	1.98
Xavier	GPU (CUDA)	1024	42.85	6.29
Xavier	GPU (CUDA)	2048	57.59	8.52
Xavier	GPU (CUDA)	4096	55.41	8.28
V1605B	CPU (OpenMP)	1024	18.81	1.25
V1605B	CPU (OpenMP)	2048	17.35	1.16
V1605B	CPU (OpenMP)	4096	16.27	1.08
V1605B	GPU (OpenCL)	1024	(*)1.89	(*)0.13
V1605B	GPU (OpenCL)	2048	(*)2.06	(*)0.14
V1605B	GPU (OpenCL)	4096	(*)2.11	(*)0.14
Kirin 970	CPU (OpenMP)	1024	5.60	0.37
Kirin 970	CPU (OpenMP)	2048	11.23	0.75
Kirin 970	CPU (OpenMP)	4096	14.98	0.97
Kirin 970	GPU (OpenCL)	1024	5.60	1.00
Kirin 970	GPU (OpenCL)	2048	11.23	2.02
Kirin 970	GPU (OpenCL)	4096	14.58	2.61

Table 2. OBPMark Benchmark #1.1 "Image Calibration and Corrections" test results for COTS GPU devices

Results from two compression benchmarks are also presented, only results for the NVIDIA Xavier and the AMD Embedded Ryzen V1605B are presented here. Results of Benchmark #2.1, CCSDS 121.0 Data Compression, using three different block sizes (J): 16, 32 and 64 are presented in Table 3. No power (efficiency) results are presented for #2.1 here, details to be included in later publications. In the data compression benchmarks, the V1605B performed up to 2x higher than the Xavier in specific benchmark configurations when using the CPUs. When using the GPUs, the results were comparable between both devices.

Results of Benchmark #2.2, CCSDS 122.0 Image Data Compression, using two images sizes: 2048x2048 and 4096x4096 are presented in Table 4. Again here it is

Device	Target Impl.	Msamples/s (J=16)	Msamples/s (J=32)	Msamples/s (J=64)
Xavier	CPU (Seq.)	12.10	6.13	3.18
Xavier	CPU (OpenMP)	22.00	10.31	6.52
Xavier	GPU (CUDA)	12.81	9.62	6.41
V1605B	CPU (Seq.)	18.59	9.67	4.49
V1605B	CPU (OpenMP)	41.00	22.45	11.50
V1605B	GPU (OpenCL)	16.91	10.28	5.50

Table 3. OBPMark Benchmark #2.1 "CCSDS 121.0 Data Compression" test results for COTS GPU devices

shown that the Xavier GPU performs the highest overall, both in absolute throughput and in energy efficiency. However, due to the issues with the OpenCL support for the V1605B described above – the V1605B GPU results shown here are not representative of the device capabilities. When measuring the CPU performance for the image data compression benchmark, the V1605B again outperforms the Xavier in throughput in all modes. However, in energy, efficiency, the results are roughly equivalent.

Device	Target Impl.	Image size	Mpixels/s	Mpixels/s/W
Xavier	CPU (Seq.)	2048x2048	3.84	0.48
Xavier	CPU (Seq.)	4096x4096	3.31	0.40
Xavier	CPU (OpenMP)	2048x2048	4.32	0.48
Xavier	CPU (OpenMP)	4096x4096	4.14	0.49
Xavier	GPU (CUDA)	2048x2048	17.07	1.87
Xavier	GPU (CUDA)	4096x4096	30.64	2.79
V1605B	CPU (Seq.)	2048x2048	5.48	0.37
V1605B	CPU (Seq.)	4096x4096	5.39	0.36
V1605B	CPU (OpenMP)	2048x2048	7.40	0.49
V1605B	CPU (OpenMP)	4096x4096	6.01	0.40
V1605B	GPU (OpenCL)	2048x2048	(*)0.88	(*)0.06
V1605B	GPU (OpenCL)	4096x4096	(*)0.88	(*)0.06

Table 4. OBPMark Benchmark #2.2 "CCSDS 122.0 Data Compression" test results for COTS GPU devices

These initial benchmark result shows that it is important to consider several applications, implementation approaches and parameters when making fair comparisons between different processing devices and systems.

7. SUMMARY AND CONCLUSIONS

Currently, there is a lack of openly available comparable processing benchmarks for space applications. In this paper OBPMark, a set of computational performance benchmarks for on-board processing applications has been presented, reusing and extending the implementations of GPU4S Bench open source benchmarking suite. It is proposed to use the benchmarks for general performance evaluation of on-board systems and devices.

The OBPMark and GPU4S Bench source code repositories are available at: ***OBPMark.org***. Researchers in the field of on-board data processing and vendors of processing devices, modules and systems who wish to work on performance benchmarks for space are encouraged to reach out and contact the authors through:
OBPMark@esa.int

8. FUTURE WORK

In the next phase of the activity, the remaining benchmarks (listed above) will be finalized and implemented. Additional ports of the benchmarks, targeting multicore RHBD (radiation hardended by design) and RT (radiation tolerant) processors are planned. In future versions of OBPMark dedicated optimizations for commonly used ISAs, such as SPARC, ARM and RISC-V may be considered.

OBPMark has been specified to allow the inclusion of additional benchmarks in the future. Possible expansions of the OBPMark benchmark suite include video encoding/-compression; additional compression standards; and/or additional machine learning benchmarks again related to representative space use. Future use of video encoders are currently being evaluated by ESA, including commercial standards such as H.264/H.265. Possible adaption of existing video standards for future use in space may be required. As commercial compression standards, such as JPEG2000, become more popular for space applications, guidelines for performance benchmarks may be also added. In addition, dedicated benchmarks for compression of SAR data based on upcoming standards may be added. As machine learning is a rapidly evolving field, there may be a need to complement the current plan with additional model architectures or new algorithms. Additional benchmarks covering AOCS/GNC applications and/or visual-based navigation may be added, as these are currently a topic of research and have been covered by several ESA activities. Application-level benchmarks for telecommunication applications, such as DVB-S2 transceivers, will also be considered. Inclusion of additional benchmarks will be subject of community feedback of early public released version of OBPMark.

ACKNOWLEDGEMENT

Part of the work for OBPMark has been performed through the ESA GSP (General Studies Programme) activity "Low Power GPU Solutions for High Performance On-Board Data Processing" (ITT AO/1-9010/17/NL/AF), GPU4S (GPU for Space). Moreover, some of the work was partially supported by the Spanish Ministry of Economy and Competitiveness under the grant FJCI-2017-34095 and by the European Commission's Horizon 2020 programme under the UP2DATE project (grant agreement 871465).

REFERENCES

[1] David Steenari, Kyra Förster, Derek O'Callaghan, Craig Hay, Mikulas Cebecauer, Murray Ireland, Sheila McBreen, Maris Tali, and Roberto Camarero. Survey of high-performance processors and FPGAs for on-board processing and machine learning applications. In *OBDP2021, 2nd European Workshop on On-Board Data Processing*. ESA/CNES/DLR, 2021.

[2] George Lentaris, Konstantinos Maragos, Ioannis Stratakos, Lazaros Papadopoulos, Odysseas Papanikolaou, Dimitrios Soudris, Manolis Lourakis,

Xenophon Zabulis, David Gonzalez-Arjona, and Gianluca Furano. High-performance embedded computing in space: Evaluation of platforms for vision-based navigation. *Journal of Aerospace Information Systems*, 15(4):178–192, 2018.

[3] Franck Wartel and Antoine Certain. Hp4s: High performance parallel payload processing for space. In *OBDP2021, 2nd European Workshop on On-Board Data Processing*. ESA/CNES/DLR, 2021.

[4] Max Ghiglione, Vittorio Serra, Tim Helfers, Renato Costa Amorin, and Rodolfo Martins. Machine learning application benchmark for satellite on-board data processing. In *OBDP2021, 2nd European Workshop on On-Board Data Processing*. ESA/CNES/DLR, 2021.

[5] Leonidas Kosmidis, Jérôme Lachaize, Jaume Abella, Olivier Notebaert, Francisco J Cazorla, and David Steenari. GPU4S: Embedded GPUs in space. In *2019 22nd Euromicro Conference on Digital System Design (DSD)*, pages 399–405. IEEE, 2019.

[6] Leonidas Kosmidis, Iván Rodriguez, Álvaro Jover, Sergi Alcaide, Jérôme Lachaize, Olivier Notebaert, Antoine Certain, and David Steenari. GPU4S: Major Project Outcomes, Lessons Learnt and Way Forward. In *Proceedings of the 2021 Design, Automation and Test in Europe Conference and Exhibition, (DATE)*, 2021.

[7] Iván Rodriguez, Leonidas Kosmidis, Olivier Notebaert, Francisco J Cazorla, and David Steenari. An On-board Algorithm Implementation on an Embedded GPU: A Space Case Study. In *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1718–1719, 2020.

[8] Iván Rodriguez, Leonidas Kosmidis, Jérôme Lachaize, Olivier Notebaert, and David Steenari. GPU4S Bench: Design and Implementation of an Open GPU Benchmarking Suite for Space On-board Processing. Technical Report UPC-DAC-RR-CAP-2019-1, Universitat Politecnica de Catalunya. `https://www.ac.upc.edu/app/research-reports/public/html/research_center_index-CAP-2019,en.html`.

[9] Alan R Weiss. Dhrystone benchmark: History, analysis, scores and recommendations. 2002.

[10] Embedded Microprocessor Benchmark Consortium. CoreMark: An EEMBC Benchmark, 2018.

[11] Javier Jalle, Magnus Hjorth, Jan Andersson, Roland Weigand, and Luca Fossati. DSP benchmark results of the GR740 rad-hard quad-core leon4ft. 2016.

[12] Cobham Gaisler. GR740-VALT-0010, GR740 technical note on benchmarking and validation, 2019.

[13] Peter Torelli and Mohit Bangale.

[14] Peter Mattson, Vijay Janapa Reddi, Christine Cheng, Cody Coleman, Greg Diamos, David Kanter, Paulius Micikevicius, David Patterson, Guenther Schmuelling, Hanlin Tang, et al. Mlperf: An industry standard benchmark suite for machine learning performance. *IEEE Micro*, 40(2):8–16, 2020.

[15] Rob VanderWijngaart and Bryan A Biegel. Nas parallel benchmarks. 2.4. 2002.

[16] ESA. Next generation space digital signal processor software benchmark, issue 1.0, TEC-EDP/2008.18/RT, 2008.

[17] Alvaro Jover-Alvarez, Alejandro J. Calderon, Ivan Rodriguez, Kosmidis Leonidas, Kazi Asifuzzaman, Patrick Uven, Kim Grüttner, Tomaso Poggi, and Irune Agirre. The up2date baseline research platforms. In *Proceedings of the Design, Automation & Test in Europe (DATE)*, 02 2021.

[18] CCSDS 121.0-B-3, Lossless Data Compression, Blue Book, Issue 3, Recommended Standard, August 2020.

[19] CCSDS 122.0-B-2, Image Data Compression, Blue Book, Issue 2, Recommended Standard, September 2017.

[20] CCSDS 123.0-B-2, Low-Complexity and Near-Lossless Multispectral and Hyperspectral Image Compression, Blue Book, Issue 2, Recommended Standard, February 2019.

[21] Ivan Rodriguez, Alvaro Jover, Leonidas Kosmidis, and David Steenari. On the embedded GPU parallelisation of on-board CCSDS compressors: a benchmarking approach. In *OBPDC2020*. ESA, 2019.

[22] Ian Blanes, Aaron Kiely, Miguel Hernández-Cabronero, and Joan Serra-Sagristà. Performance impact of parameter tuning on the ccsds-123.0-b-2 low-complexity lossless and near-lossless multispectral and hyperspectral image compression standard. *Remote Sensing*, 11(11):1390, 2019.

[23] NIST. Advanced encryption standard (AES). federal information processing standards special publication 197, 2001.

[24] CCSDS 352.0-B-2, CCSDS Cryptographic Algortihms, Blue Book, Issue 2, Recommended Standard, August 2019.