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

SPACECLOUD CLOUD COMPUTING AND IN-ORBIT DEMONSTRATION

Oskar Flordal(1), Aris Synodinos(1), Mattias Herlitz(1),Henrik Magnusson(1),David Steenari(2), Kyra Förster(2), Michele Castorina(3) ,Dr. Tom George(4) ,Dr. Ian Troxel(5) ,Simon Reid(6) ,Dr. Chris Brunskill(6) ,Dr. Fredrik Bruhn(1,7)

(1)Unibap AB, Kungsangsgatan 12, S-753 22 Uppsala, Sweden, Email: firstname.lastname@unibap.com
(2)European Space Agency - ESTEC, Keplerlaan 1, 2201 AZ Noordwijk, Netherlands, Email: david.steenari@esa.int /kyra.foerster@esa.int
(3)European Space Agency - ESRIN, Via Galileo Galilei 1, 00044 Frescati RM, Italy, Email: michele.castorina@esa.int
(4)SaraniaSat Inc, 9235 Scotmont Drive, Tujunga, CA 91042, USA, Email: tom.george@saraniasat.com
(5) Troxel Aerospace Industries Inc, 2023 NE 55th Blvd., Gainesville, FL 32641, USA, ian@troxelaerospace.com
(6)D-Orbit UK Ltd, 316 Beulah Hill, London SE19 3HF, UK, Email: simon.reid@dorbit.space
(7)Mälardalen University, Högskoleplan 1, S-722 20 Västerås, Sweden, Email: fredrik.bruhn@mdh.se

ABSTRACT

Processing requirements are exponentially increasing to keep pace with the data volumes generated by increasingly “Big Data” sensors. These requirements are compounded when factoring in the data movements planned for future spacecraft constellation mesh networks, i.e. connected spacecraft infrastructures for on-orbit fleet management, autonomous sensor fusion, data storage, very low latency actionable information generation, and real-time communication.

Unibap AB and Troxel Aerospace Industries, Inc. have worked together to develop a heterogeneous radiation-tolerant onboard cloud computing hardware platform bringing terrestrial Internet-of-Things edge processing to space, e.g. Infrastructure as a Service (IaaS), Big Data analytics and Artificial Intelligence (AI). This platform is part of Unibap’s SpaceCloud ecosystem, which makes virtual servers and other resources dynamically available to customers. Leveraging its powerful heterogeneous hardware platform, the SpaceCloud framework has been developed with support by the European Space Agency (ESA) to enable rapid and flexible application (“App”) development using containerized and isolated virtualization either for execution locally or on networked spacecraft. SpaceCloud allows exchange of information that is transparent between local or networked nodes to facilitate cooperation using a distributed mesh network communication architecture.

The focus of SpaceCloud is to enable commercial software to be reused onboard spacecrafts to decrease the overall cost and development time to deploy new capabilities on compatible space assets. As an example, SaraniaSat and Unibap have worked with L3Harris Geospatial to enable the geospatial intelligence software suite ENVI®/IDL® on SpaceCloud. A very low-latency onboard SpaceCloud application for detecting aircraft using ENVI®/IDL® and machine learning within 100 sq. km World View 3 multispectral satellite imagery has been successfully developed and demonstrated.

The SpaceCloud framework executes on the iX5 and iX10 families of x86 radiation-tolerant computer solutions featuring AMD multi-core CPU, GPU, Microsemi FPGA, and Intel Movidius Myriad X VPU accelerator and local high-speed solid-state storage. Radiation testing in the US has shown very promising results on both 28 nm and 14 nm processor nodes with high tolerance for single event latch-up (SEL) and total ionizing dose (TID).

To improve radiation tolerance, the SpaceCloud framework performs real-time software monitoring through a SafetyChip feature working in tandem on the x86 software stack and an external real-time operating system with funding support from ESA. Radiation tolerance can be further increased by use of a single event upset (SEU) mitigating middleware that protects CPU and GPU processing.

This paper presents the SpaceCloud In-orbit Demonstration (IOD) compute architecture and framework configuration as implemented in D-Orbit’s Wild Ride ION SCV mission due for launch in Q2 2021.

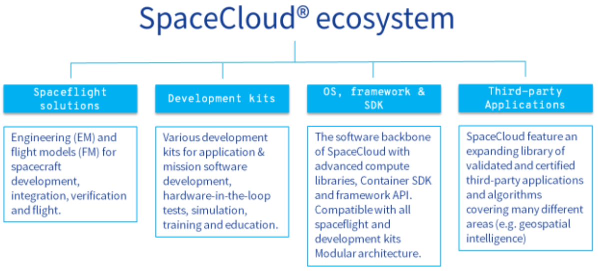
SpaceCloud® is a registered trademark by Unibap.

1. INTRODUCTION

Small satellites and especially CubeSats are increasingly used for Earth Observation (EO) and advanced scientific missions [1]. Over the past decade, there has been a solid drive to demonstrate the commercial value of remote sensing data with large-scale EO constellations being launched and operated by, for instance, Planet, Spire and Satellogic [2]. The number of launched tiny satellites is growing exponentially, with revisit periods frequently under a day and a wide range of sensor data options have been introduced (e.g. optical spectral bands, SAR, radio occultation, radio frequency (RF)). Small satellites with machine to machine and asset tracking capabilities have been taken operational (e.g. Software Defined Radio (SDR), ADS-B, AIS, Internet-of-Things (IoT), and Signal Intelligence (SIGINT)).

2. SPACECLOUD OVERVIEW

SpaceCloud is Unibap's solution to simplify high-performance on-board data processing in space. It consists of the iX5 and iX10 hardware, a standard Linux distribution with tailoring to simplify use of the HW resources and to bring robustness features as well as the SpaceCloud Framework.



Engineering (EM) and flight models (FM) for spacecraft development, integration, verification and flight.

Various development kits for application & mission software development, hardware-in-the-loop tests, simulation, training and education.

The software backbone of SpaceCloud with advanced compute libraries, Container SDK and framework API. Compatible with all spaceflight and development kits Modular architecture.

SpaceCloud feature an expanding library of validated and certified third-party applications and algorithms covering many different areas (e.g. geospatial intelligence)

Figure 1. *Overview of SpaceCloud components*

2.1. SpaceCloud Framework

SpaceCloud Framework (SCFW) have these primary design criteria:

- Simplify deployment and updates of applications for satellites in orbit using common developer tools found on ground.
- Allow sharing of a satellite by providing a multi-tenant framework where apps serving different users can be scheduled at different times based on events (positions, time, external requests).
- Add abstractions for sensor and communication interfaces that may otherwise need per satellite tailoring for developed apps.
- Allow monitoring of resources per application, applying thresholds to computational resources and scheduling the access to the hardware capabilities of the platform.
- Simple compression, logging and prioritization of communication.

SCFW implements this by:

- Providing process isolation and inter-process communication using docker and gRPC
- Provides an abstraction layer for common functionality on satellites such as sensors (including imaging sensors), OBC connectivity, location services and abstraction for the communication.
    - ○ Communication is done via synchronising directories of data generated by an application to ground.
    - ○ The imaging sensor abstraction can be packaged with pre-processing functionality for a specific sensor and provides a hyperspectral image to the application.

SCFW is built as an asynchronous framework where the framework will collect the data and queue it up for an application that can fetch e.g., an image and associated location and other sensor data from when the image was collected.

SCFW does extend to ground services but on this mission primarily the on-board services are tested by running a set of applications in rotation.

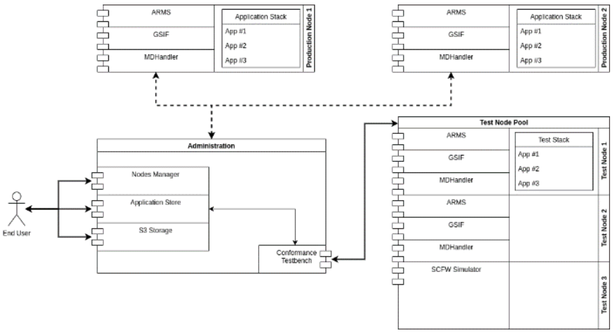


Figure 2. SpaceCloud framework overview

2.1.1 Efficient application uploads

To reduce the typical large bandwidth and storage requirements of docker containers the following approach is taken.

- On launch, SCFW is packaged with a superimage, a parent docker image with many commonly used packages for earth observation and other common tasks. Apps are primarily built on top of this image adding only the application-specific logic rather than a whole OS. Applications that have special needs can still replace components or even build on a separate docker base but without the bandwidth benefits.
- Only the necessary layers are uplinked to the craft, which typically only contains the application logic and specific libraries. Those layers can also be shared amongst different applications, reducing the uploaded size even further.
- Compression is applied to the uplinked layers bringing the size down to 10s of KBs for simple applications that lack binary blobs.

2.1.2 Scheduling

For the in-orbit demonstration, a simplified work scheduling is used where a fixed task schedule of 10 apps in ~20 different configurations are used. Each time the iX5 wakes up it runs as many tasks as possible from the list for 10 minutes and then carries on the work from where it stopped on the next invocation. The results are gathered in a report for analysis from ground. The applications work primarily from data already stored on the payload at launch which simplifies analyzing the execution for any possible anomalies.

2.1.3 Deployment

During the mission, new apps can be deployed by passing the necessary docker image layers together with a set of metadata describing the execution parameters such as command-line options, storage specifications, hardware resources access and what docker images to use [3]. During the IOD, this is uplinked as a special mission update packaged and communicated through the ION OBC. On future missions, this process can be replaced through an appropriate channel that talks directly to SCFW via for example TCP/IP. SCFW will store the new layers in cold storage to be rebuilt later and assemble the docker image on the system to be launched according to an updated schedule.

If there is a disk fail while in orbit all applications can be rebuilt from the cold storage on either of two redundant disks that contain the relevant applications.

2.1.4 Execution monitoring & Recovery

An application specifies the amount of compute resources such as RAM, number of cores, access to the GPU or the VPU as well as the maximum execution time. This is both to keep track of energy usage and when appropriate, concurrent applications. The SCFW ARMS component will keep track of execution and stop applications that does not perform within those limits. Bandwidth allocation is also monitored, so applications that generate more data than the specified 500KB maximum will be discarded in favor of system logs and other applications. This functionality is supported by linux kernel’s cgroups and namespaces, both for limiting access during execution and for monitoring overall usage of computational resources upon completion.

The SCFW monitoring system has multiple layers of recovery depending on the severity of the malfunction. Checksums are used to validate all applications during loading and execution, the system recovers from faulty applications that fail to execute or load automatically and skips that application to the next one. If disk corruption happens in the system configuration files, the system automatically rebuilds the configuration based on the already deployed applications in the system and the data in cold storage. If these recovery attempts continue to fail, and the SCFW goes into an unrecoverable state, then a full system clean is performed and the system tries to recover from scratch. This last recovery scenario is computationally intensive and is the last layer of defense for autonomously recovering the software. Manual intervention is still feasible even in that case, through the SCOS monitoring software.

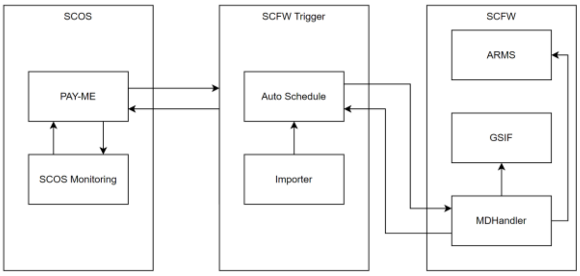


Figure 3. SCFW IOD information flow through system components

2.2. SpaceCloud compute and storage architecture

Bruhn et al have described the SpaceCloud heterogeneous compute and storage architecture of the iX5-100 [6]. Figure 3 shows the iX5-100 compute and storage architectural overview for the Wild Ride ION mission. A total of 256 GB solid-state storage (SSD) is available on two M.2 SATA slots where each SATA port is capable of 6 Gbps.

The compute and data processing capability are distributed over the following elements.

Table 1. Summary of the iX5-100 compute and storage capabilities on the Wild Ride ION mission.

Compute device	**Performance**
Quad core, 64-bit x86 CPU (AMD Jaguar) with AVX vector instructions	40 GFLOPS
AMD Radeon GPU	87 GFLOPS
Intel Movidius Myriad X Vision Processing Unit	4 TOPS
Microsemi SmartFusion2 FPGA	72 DSP cores + fabric
Storage with redundant boot (through SafetyBoot mechanism)	2 x 128 GB M.2 SATA SSD (SLC Nand flash memory type)

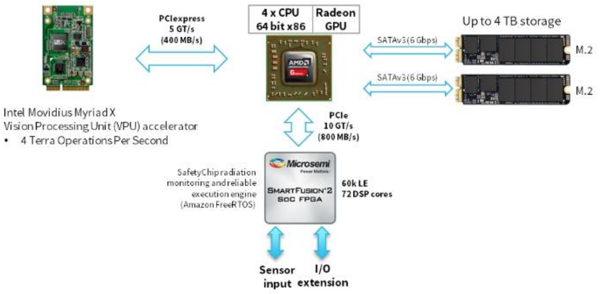


Figure 3. iX5-100 compute and storage architecture overview.

Figure 4 shows a photograph of the iX5-100 flight hardware stack mounted in the ION Wild Ride flight chassis. The SpaceCloud flight package measures approximately 10 x 10 x 7 cm3.

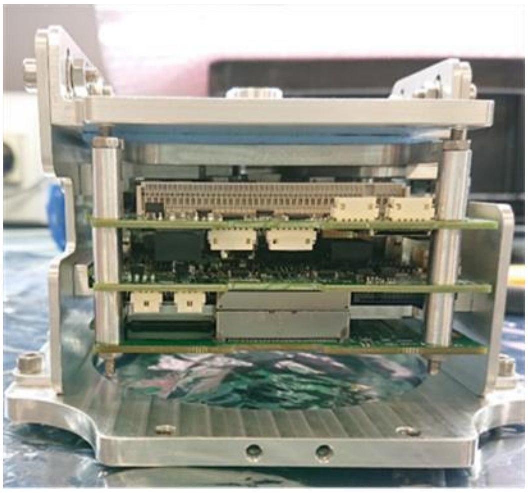


Figure 4. Unibap iX5-100 FM in flight chassis for the D-Orbit Wild Ride ION mission.

Unibap and Troxel Aerospace are pursuing the next generation compute and storage solutions for space. The SpaceCloud architecture is transitioning to 7 nm and 14 nm AMD system-on-chips from the Ryzen Embedded families.

The next-generation architecture incorporates nonvolatile memory express (NVMe) interfaces for higher data rate storage and compute storage solutions, i.e., storage devices with built-in neural network compute capability.

The companion FPGA is upgraded to Microchip/Microsemi PolarFire and up to 3 times more neural compute devices can be attached (e.g., Intel Movidius Myriad X/Keem Bay).

3. ION IN-ORBIT-DEMONSTRATION OVERVIEW

The ION Satellite Carrier Vessel (SCV) is a small satellite designed for space transportation services. Once on orbit, an SCV utilises comprehensive mission planning and propulsive manoeuvres to position itself in precise orbital configurations, from which it can deploy customer payloads directly into their desired orbit. Multiple orbits can be visited by an SCV in a single mission. Once the primary transportation mission is complete the SCV is available for IOD/IOV and other longer-term mission support services. As such, it provides an ideal platform for hosting the iX5 hardware in coordination with other on-board data processing and management services. Moreover, as the ION fleet is extended through subsequent transportation services, a robust in-orbit infrastructure is achieved, providing a comprehensive capability to support end-user needs.

The ION SCV platform can carry and transport up to 64U of payloads and small satellites, including those deployed and/or hosted, up to a mass of 160 kg. The platform is equipped with a typical suite of on-board avionics, including eight attitude control sensor clusters in the form of “D-Sense” packages. For the WILD RIDE mission, a single iX5 on-board processor is integrated into the payload bay. Interfaces with the bus are managed through a dedicated payload management board, providing independent control of power state and data interfaces for transfer of payload data up to and down from the on-board processor payload.

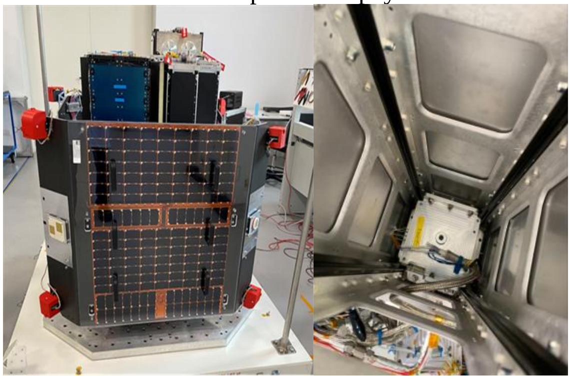


Figure 5. (l) ION SCV Dauntless David, fully integrated with customer payloads, including a UniBap iX5, ahead of launch; (r) Unibap iX5-100 FM mounted in the ION SCV payload bay

Integration, testing and validation of the iX5 payload onboard Dauntless David is now complete and the spacecraft is undergoing final preparations for launch in Q2 2021. After launch and following the completion of transportation of six customer payloads to distinct orbits and initial IOD activities, the on-board processor IOD phase will commence. This mission phase will operate the iX5 to execute SpaceCloud applications over an extended mission period as an initial proof of concept of an on-demand, on-orbit cloud computing and data storage service, integrated into ION platforms.

4. SPACECLOUD APPLICATIONS

One of the most important aspects of Cloud computing capabilities in space is enablement of software reusability by either porting existing applications and algorithms into the SpaceCloud infrastructure or to develop new applications that can be reused on many missions and be easily deployed using the orchestration layer of SpaceCloud framework.

Several applications have been studied with respect to the Wild Ride ION mission demonstration. A selection is presented here as examples.

4.1. Near-real time mid-air aircraft detection

SaraniaSat have developed an advanced SpaceCloud application for near-real time mid-air airplane detection, localization, and identification. The application uses onboard machine learning and leverages the ENVI®/IDL® software suite from L3Harris Geospatial included in the SpaceCloud framework accessible from an application container image for the ION mission.

The application has been certified on SpaceCloud hardware on ground and is pre-loaded into the Wild Ride ION mission for in-orbit demonstration using pre-loaded World View-3 multispectral imagery.

Using the heterogeneous compute architecture of SpaceCloud, the application can analyse 100 sq. km area patches in 43 seconds and produce geo-location information of any detected airplane together with a keyhole markup language zipped (kmz) file that is downlinked to earth. The geolocation information and kmz image data can be overlayed on Google Earth as shown in Figure 6. This approach allows downlinking 10s of KB of data to verify positions of detected planes rather than multi-GB tiles for detection on ground.

The measured consumption for this application on the iX5-100 flight hardware is approximately 18 W which corresponds to 774 Joule of energy for a 43 second run.

If satellite communication like OrbCOMM, Globalstar or Inmarsat is used the actionable information can be delivered to a user with 1 minute. This fits well within the definition of extremely low-latency data products from space.

Figure 6. SaraniaSat near real-time airborne airplane SpaceCloud detection application, An airplane is detected in reference World View-3 multispectral imagery.

4.2. Image and video compressions apps

Compression algorithms to reduce bandwidth of data that needs to be downlinked from the satellite. Together with deep learning and algorithm-based filtering it can reduce the data volumes significantly. On this IOD two types of compression are demonstrated.

The first application is the ENEA's implementation of CCSDS-123.0-B-2 [4] on multicore CPU, which was developed as part of an ESA Romanian Industrial Incentive Scheme activity. For this in-orbit demonstrator hyper-spectral and RGB data is compressed and can be compared with data compressed on the ground to investigate robustness during execution.

The second application is running two of V-Novas implementations primarily targeted at RGB video streams. This is LCEVC [5] that acts as an enhancement layer on top of another compressed stream such as the ones generated by the iX5 video encode accelerator or a second software encoder. V-Nova has also provided an OpenCL accelerated copy of their VC6 codec which runs on the on-board GPU. VC6 can also do layered coding that support use cases like initially sending a low-resolution video of a scene which can be refined by downlinking further higher resolution layers.

4.3. Heterogenous execution car tracking app

To test all the execution units on the iX5 Unibap developed a simple car speed identification app. The application contains two steps.

- Running an object detection neural network (mobilenet2 ssd implemented in Tensorflow 2 [7]) that was trained on the COWC [8].
- A OpenCV [9] object tracker is run on the detected regions.

The application is meant to mirror a typical scenario of a deep learning application. Run a network and do some

post processing to get to actionable data, in this case the velocities of the cars in a certain region.
The application runs object detection network across the CPU and GPU using Tensorflow lite(TFLite) as well as the Movidius accelerator using OpenVINO. The tests are run on a set of preloaded overhead drone photos.

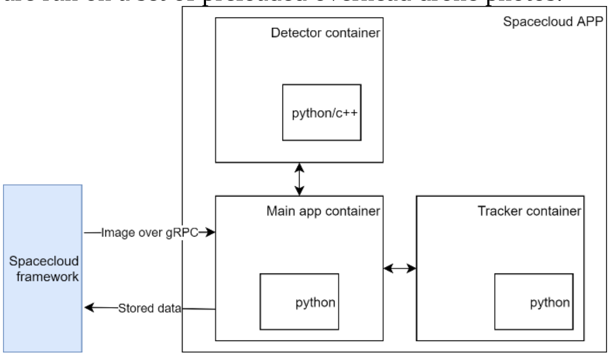


Figure 7. The SpaceCloud app consists of several separate containers demonstrating isolating reusable functionality.

The various functional blocks are split into separate containers to demo partitioning an application in several reusable containers. This way an inference server can be implemented which can rely on using e.g., the Movidius inference container or the TensorFlow lite inference container for platform redundancy or to adapt to the requirements of other workloads.

4.4. ESA On-Board Radiation Upsets in Machine Learning Inference Experiment

ESA has developed two deep learning applications to be executed on the IOD. The first application is a detection algorithm of Coronal Mass Ejections (CMEs). A Convolutional Neural Network (CNN) compares two coronagraph images and classifies if a CME has occurred. The application outputs the classification results and execution times of each processing stage. In addition, a comparison with the classification ground truth may show any errors that occurred in flight.

The initial model training and the CNN architecture was developed by the University of Politecnico di Torino [10]. Additional adaption and optimization for several FPGAs and embedded processors at ESA, including the Unibap iX5 computer.

The second application is an image segmentation task that detects radiation particle hits in scientific imaging data. Particle hits in the sensor can cause incorrect instrument data, which can lead to non-usable data -- particularly after image stacking. The algorithm uses pattern recognition to detect radiation particles in the image and generate a mask of upset particles. The generated masks are compared with the ground truth masks and potential errors are highlighted. The detected particles are then scrubbed by a subsequent algorithm.

The application was developed internally at ESA, targeting several FPGAs and embedded processors, using deep learning to solve an image segmentation problem lead to the generation of a radiation particle mask as network output. The training data was generated using an instrument simulator for optical instruments. Training was done in TensorFlow again with the subsequent conversion to TFLite.

For the IOD, the inference deployment for both applications were implemented on the AMD x86 CPU cores of the Unibap ix5. The models were trained in TensorFlow, then converted to TFLite for inference, using both the Python and C++ bindings. Additional inference implementations for the AMD GPU and Myriad X are on-going and are targeted to be deployed later in-flight. The implementation is configurable, and several parameters (such as e.g. number of runs, target files, size of input images) can be updated in flight.

In both cases the applications are executed on-board using reference data and the results are compared to ground through data, to detect if there have been any radiation-induced upsets in the used processor during execution. In particular, the second application deployed has the potential to analyse any errors caused by radiation effects in the target COTS processor in orbit due to the high number of computation and segmented execution.

A more detailed description of the algorithms and the results of the on-board inference for both applications will be published in a subsequent technical paper.

4.5. PUSopen

Standardization of space communication plays an essential role in the booming ecosystem of COTS space systems from different vendors. With easy deployment and efficiency of communication in mind, two well-established space communications standards ECSS PUS and CCSDS, joined in the PUSopen software communication stack. The PUSopen solution is especially suitable for space platforms with no native support of CCSDS or PUS communication.
On this mission, dockerized test cases are executed to verify the functionality:

- Test of generation and acceptance of various telemetry data as well as telecommands with responses.
- Test of generating PUS 8 request and triggering PUS 8 custom function with the various number of bytes of custom data.
- Test of downloading large data sets via PUS 13.

- In addition, a new CCSDS 124.0 compression scheme is tested on the produced housekeeping PUS telemetry.

Depending on topology, PUSopen can be deployed either on the onboard MCU of the iX5 or directly on the main CPU.

5. EXECUTION HARDENING

Unibap have teamed up with Troxel Aerospace Inc and the European Space Agency to increase the reliability of execution in the SpaceCloud heterogeneous architecture.

Execution reliability is improved using three main methods,

a) Component selection and robust electronics design according to ECSS and IPC standards, including support to detect critical FPGA errors on the clocks and IO banks in addition to single event upsets and single event latch-ups.
b) Implementation of SafetyBoot, a combined hardware and software reliability feature to provide a redundant boot mechanism for the x86 AMD system on chip (SOC).
c) Implementation of SafetyChip, a combined hardware and software reliability feature to monitor the complex x86 execution and overall system fault detection isolation and recovery (FDIR) (e.g., Linux kernel status, kernel drivers, critical user-space applications, monitoring of middleware)

5.1. SafetyBoot

For this IOD SafetyBoot selects between two mirrored SSDs. This is done by only powering one of the disks. A failed boot, i.e., by the boot sequence not flagging to SafetyChip firmware that boot was successful will cause SafetyChip to swap boot device. After boot, the second drive can be powered on if needed but is kept off by default to avoid introducing new issues.

To make sure the content of the second disk is valid periodic checks and optional updates such as mirroring newly updated applications is necessary.

6. NASA SCIENCE MISSION SPACECLOUD EXAMPLE

In 2017 NASA awarded Hawaii Spaceflight Laboratory (HFSL) the Hyperspectral Thermal Imager (HyTI) science satellite mission under the NASA's Earth Science Technology Office InVEST (In-Space Validation of Earth Science Technologies) program.

The HyTI mission will demonstrate how high spectral and spatial long-wave infrared image data can be acquired from a 6U CubeSat platform.

The mission will use a spatially modulated interferometric imaging technique to produce spectro-radiometrically calibrated image cubes, with 25 channels between 8-10.7 µm.

The value of HyTI to scientists and industry will be demonstrated via on-board processing of the raw instrument data to generate L1 and L2 products, with a focus on rapid delivery of data [11].

HSFL will demonstrate several internally developed software applications on the HYTI mission. These focuses on the following primary mission objectives. The application will run on the SpaceCloud hardware and leverage the SpaceCloud OS capabilities.

- Volcanic activity monitoring
- Precision agriculture monitoring
- Forest fires detection

Figure 8 shows the SpaceCloud connectivity on the HYTI mission. The payload data processing is separate from the onboard avionics computer (OBC), or command & data handling (C&DH). Hence, the SpaceCloud solution can operate independently when allowed by the OBC and mission operations.

The spectral sensor is read out using a Camera Link interface and the data is processed in real-time in the iX5-100 as it comes off the sensor and stored on the internal solid-state drive storage (SSD). The iX5-100 solution is mapped through direct memory access (DMA) to S-band and X-band radio transmitters. This allows the software developers to develop software quickly and use simple Linux devices mapped by the SpaceCloud kernel driver.

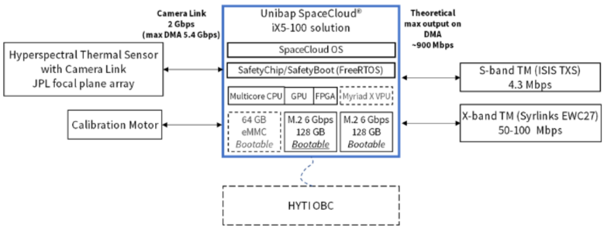


Figure 8. Unibap SpaceCloud iX5-100 connectivity on NASA/Hawaii Spaceflight Laboratory HYTI satellite.

The HYTI satellite is planned to be launched at the end of 2021.

7. FUTURE WORK

Unibap and Amazon Web Services (AWS) have announced that AWS cloud, storage, and edge services will be integrated into SpaceCloud and initial testing and validation will be done in 2021 [12].

8. CONCLUSIONS

This paper has given an overview of the in-orbit demonstration of SpaceCloud on the D-Orbit SCV.

Using a SpaceCloud like system is shown capable of handling a diverse set of applications developed by several different entities. This paradigm can simplify and extend the possibilities around on-board computing.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the support of the European Space Agency InCubed program office for project “SpaceCloud Framework”, ongoing support from the European Space Agency Business Space program under the ARTES framework and the ESA GSTP (General Support Technology Programme).
L3Harris Geospatial is acknowledged for supporting with ENVI®/IDL® licenses and engineering support.
US Space Force is acknowledged for contributing World View-3 imagery.

REFERENCESS

[1] Allied Analytics, LLC, Report “CubeSat Market by Size, Application, End User and Subsystem: Global Opportunity Analysis and Industry Forecast, 2020-2027", January 2021.
[2] Curzi, G.; Modenini, D.; Tortora, P. Large Constellations of Small Satellites: A Survey of Near Future Challenges and Missions. Aerospace 2020, 7, 133. https://doi.org/10.3390/aerospace7090133
[3] SpaceCloud framework manual, accessed 2021-05-31 https://unibap.com/en/support/spacecloud-framework-manual/#application-testing
[4] Consultative Committee for Space Data Systems. Low-Complexity Lossless and Near-lossless Multispectral and Hyperspectral Image Compression-CCSDS 123.0-B-2. In Blue Book; CCSDS Secretariat: Washington, DC, USA, 2019
[5] Maurer, Florian, et al. Overview of MPEG-5 Part 2–Low Complexity Enhancement Video Coding (LCEVC). ITU Journal: ICT Discoveries 3.1 2020
[6] Bruhn, F.C., 2020, Enabling Radiation Tolerant Heterogeneous GPU-based Onboard Data Processing in Space, CEAS Space Journal, DOI 10.1007/s12567-020-00321-9.
[7] SSD MobileNet v2 320x320, accessed 2021-06-11 http://download.tensorflow.org/models/object_detection/tf2/20200711/ssd_mobilenet_v2_320x320_coco17_tpu-8.tar.gz
[8] Mundhenk, T. Nathan, et al. A large contextual dataset for classification, detection and counting of cars with deep learning. European Conference on Computer Vision. Springer, Cham, 2016.
[9] Bradski, G., 2000, The OpenCV Library. Dr. Dobb’s Journal of Software Tools.
[10] Valsesia, Diego, et al. "Detection of Solar Coronal Mass Ejections from Raw Images with Deep Convolutional Neural Networks." IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium. IEEE, 2020.
[11] Wright, R., 2020, HYTI: High Spectral and Spatial Resolution Thermal Imaging from a 6U CubeSat, AIAA/USU SmallSat Conference 2020
[12] Official press announcement of Unibap and AWS SpaceCloud collaboration, accessed 2021-05-26, https://investor.unibap.com/pressmeddelanden/unibap-becomes-a-member-of-amazon-web-services-aws-partner-n-81530