

Towards Integration of COTS Systems for High Performance Computing Applications in High-Reliability Space Systems

A Distributed Redundancy Concept with Standardized Interface Module

Uma Parvathi Malliserry Manyan
Fraunhofer Ernst-Mach-Institut
Würzburg, Germany
uma.parvathi.malliserry.manyan@emi.fraunhofer.de

Stephan Busch
Fraunhofer Ernst-Mach-Institut
Würzburg, Germany
stephan.busch@emi.fraunhofer.de

Robin Franz
Airbus Defence and Space GmbH
Friedrichshafen, Germany
robin.franz@airbus.com

Abstract—This paper introduces a distributed computing architecture for integrating high-performance COTS components into high-reliability space systems. By combining a robust, space-qualified controller with dynamically managed COTS-based high-performance processing units and standardized interfaces – potentially aligned with the ADHA framework – the approach enables scalable performance, fault isolation, and simplified qualification. A functional prototype is presented, along with requirements analysis, system design, and verification results.

Keywords— *COTS, High Performance Computing Application, Redundancy Concept, Distributed System, FDIR*

I. INTRODUCTION

Modern space applications increasingly demand high-performance on-board data processing to support advanced functionalities such as autonomy, artificial intelligence, real-time fault detection and recovery (FDIR), and payload-in-the-loop processing. However, conventional space-grade components exhibit significantly lower processing capabilities compared to commercial off-the-shelf (COTS) systems. This performance gap is primarily due to technical limitations of space-qualified electronics and the fact that COTS systems benefit from rapid innovation cycles, economies of scale, and sustained investments by large commercial markets.

In the context of recent NewSpace applications, COTS technology has already been successfully employed in agile missions with lower reliability constraints, enabling more flexible, cost-effective, and iterative development approaches. In contrast, high-reliability space missions – such as those involving large satellite platforms, long-duration deployments, or critical infrastructure – continue to rely on thoroughly verified systems with guaranteed availability. The challenge lies in reconciling the flexibility and high performance of COTS-based solutions with the stringent reliability and verification requirements of traditional missions.

The proposed concept aims to enable the use of commercial high-performance components in high-reliability space systems by combining the strengths of both classical and NewSpace approaches, leveraging their respective advantages while compensating for their limitations. To facilitate the reliable integration of existing COTS technologies, a combination of strategies at the hardware, software, and test levels is necessary to support the seamless use of COTS

components within a distributed, high reliability computing architecture.

II. STATE OF THE ART

COTS components have found particularly widespread adoption in the context of small satellite missions, where limited budgets and rapid development cycles make them an attractive alternative to space-grade components. The introduction of the CubeSat standard in 1999 marked a turning point, dramatically accelerating the proliferation of small satellites and serving as a catalyst for experimenting with terrestrial technologies in space environments. Over the past two decades, extensive experience has been gathered from these missions, leading to a broad set of lessons learned. These insights have driven the development of practical methods to enhance the robustness and reliability of COTS-based systems, including fault-tolerant software architectures and dedicated standards and design guidelines addressing fault detection, isolation and recovery (FDIR). [1], [2], [3]

In particular, high-performance commercial hardware has been successfully implemented and operated in small satellite missions, demonstrating the feasibility of using advanced COTS components for demanding on-board tasks. One notable example is the Fraunhofer nanosatellite ERNST, which employs a high performance UltraScale+ MPSoC within its data processing unit to enable high-throughput, low-latency operations for autonomous payload handling and real-time decision-making in orbit [4], [5], [6].

Distributed architectures can enable redundancy, fault tolerance, and scalability, making them particularly suitable for integrating COTS processors into high-reliability space systems. There are few research projects that demonstrate the feasibility of distributed onboard computing, showing that a distributed architecture can be a viable solution to achieve both high computational capabilities and system reliability.

1. SCOSA: Scalable On-Board Computing for Space Avionics [7] – This project introduces concepts and architecture of a distributed, fault tolerant and reconfigurable onboard computing system. This is achieved through a hybrid solution which combines highly reliable space qualified hardware with high performing COTS modules, creating multiple nodes. These nodes are connected through a SpaceWire network.
2. CSP: A Multifaceted Hybrid Architecture for Space Computing [8] – The CHREC Space Processor (CSP)

project presents a hybrid computing system that integrates COTS processors, radiation hardened components and fault-tolerant computing techniques. This project led to the production of CHREC space processor version 1 (CSPv1). CSPv1 includes COTS components for data processing, while the radiation hardened devices monitor and manage their operations. Additionally, it incorporates fault tolerance in the form of hardware and software – both within and across COTS devices. The hardware is designed to be scalable and to fit in a 1U CubeSat. CSPv1 has been tested aboard the ISS (International Space Station) and was able to handle several radiation tests.

Standardized system architectures can further play a key role in enabling the flexible and reliable integration of COTS components into a distributed onboard computer system. This has been demonstrated within the CubeSat domain, for example through the UNISEC CubeSat Subsystem Interface Definition (CSID), which provides a common framework for interoperability and modularity in small satellite platforms. [9], [10]

The Advanced Data Handling Architecture (ADHA) is an effort initialized by the European Space Agency and conducted together with a consortium of all major European Large Space System Integrators to standardize a spacecraft data handling system. ADHA defines mechanical, thermal, power, and digital/discrete interface requirements on unit as well as on board level [11]. Although not originally designed to support high-performance computing, ADHA could be extended to enable the integration of redundant COTS-based modules in a distributed architecture as proposed in this work.

III. CONCEPT OVERVIEW

The *Intelligent Platform Study* conducted by the European Space Agency analyzed the requirements on modern spacecraft platforms based on different mission type scenarios. One primary need in most mission types is increased on-board processing performance compared to conventional spacecraft platforms. To support this the study's

baseline spacecraft architecture included a High-Performance Processing Module. In order to make use of the rapid development cycles in industry the study thus concluded that the high-performance processing module shall be centered around a failure isolated COTS module. The architecture presented in this paper was used as reference and the integration into ADHA was analyzed.

At the core of the concept, as depicted in Fig. 1, is a hybrid architecture that combines the strengths of both worlds: the robustness and proven reliability of space-qualified computers with lower performance, and the computational performance and flexibility of modern COTS processors. A central, highly reliable processing module orchestrates the execution of high-level application logic, while performance-intensive tasks are dynamically delegated to multiple high-performance COTS-based computing modules, depending on current workload and system status. Task scheduling, redundancy management, and system health monitoring are handled by a software-based orchestration layer running on the reliable unit.

Individual modules could be integrated into an ADHA rack to realize the architecture. The promising concept of ADHA to standardize environmental requirements as well as the verification baseline for ADHA modules will simplify the development and guarantee a wide acceptance and usability.

A key enabler of this architecture is the use of a standardized hardware interface (e.g. by adapting to the ADHA framework) which facilitate the seamless integration of diverse COTS modules. By isolating faults at the module level and enabling graceful degradation in the presence of redundant high-performance units, the architecture supports scalable and adaptive computing capabilities for space applications. Furthermore, it offers the potential to simplify the qualification process for COTS components through modular encapsulation, fault containment, and system-level verification.

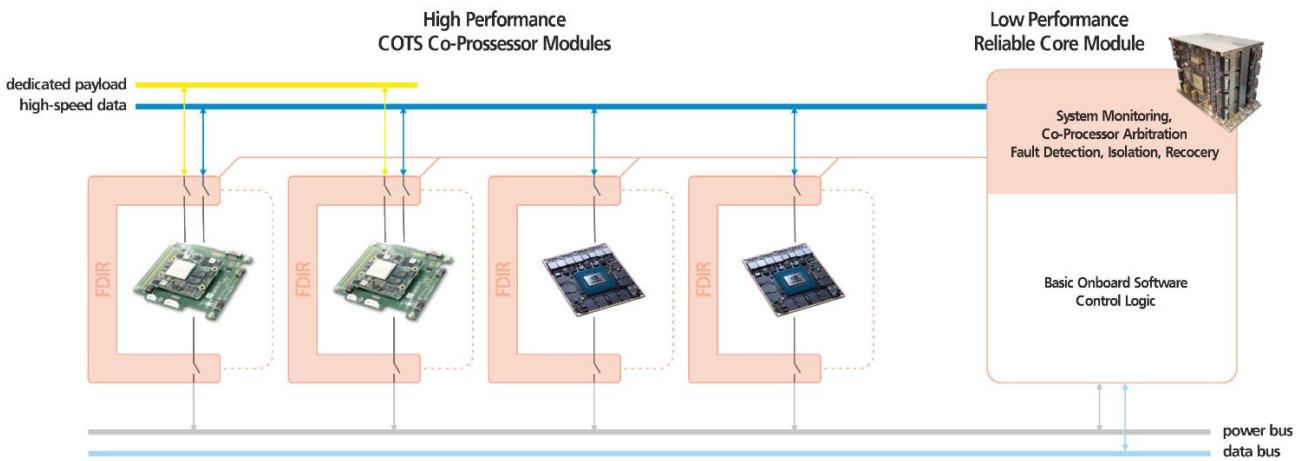


Fig. 1. Concept of the distributed architecture showing the standardized interface modules which allows using COTS components on a high-reliability spacecraft

IV. DESIGN OVERVIEW

The proposed standardized hardware interface is designed to integrate the high performing low reliable nodes (HPLR), with a low performing high reliable node (LPHR) in the distributed architecture. The goal is to develop a modular interface (MIF) to enable the seamless integration. The design follows a structured approach based on the following key system requirements:

SYS-MIF-01:

The interface module shall provide a modular electrical, thermal and mechanical interface for integration of the high performing COTS system into the onboard architecture.

SYS-MIF-02:

The interface module should be compatible with the electrical interfaces defined in table 1.

SYS-MIF-03:

The interface module shall provide a high speed data link between the HPLR and LPHR.

SYS-MIF-04:

The interface module shall provide electrical power supply for uninterrupted operation.

SYS-MIF-05:

The interface module shall monitor the HPLR and detect, isolate and respond to faults with predefined control procedures.

SYS-MIF-06:

The interface module shall prevent mechanical, thermal, electrical and functional failure propagation from the HPLR to the LPHR node.

SYS-MIS-07:

The interface module shall support the LPHR to monitor and control the interface module.

SYS-MIF-08:

The interface module shall support the LPHR to orchestrate the HPLR components and perform redundancy switching.

SYS-MIF-09:

The interface module shall support the LPHR to access the connected HPLRs for software updates and logging during development, ground operations, and in-orbit operations.

SYS-MIF-10:

The interface module shall provide interfaces for the user to access the HPLRs for debugging during both the development stage and ground operations.

To design the interface module, a market survey of various COTS processors was conducted. Table 1 lists processors evaluated based on power requirements, communication and debugging interfaces.

This analysis helped determine the necessary power regulatory circuits, data interfaces and fault management strategies required for integration. The list is a result of a trade-off between highest compatibility on the one hand and reduction of interfaces on the other hand. Fig. 2 depicts the initial block diagram of the interface design, highlighting its core features. The design includes a microcontroller unit (MCU) for control and monitoring, memory modules for data storage, power management circuits and communication interfaces. The design also includes interfaces for debugging and maintenance and incorporates mechanisms for fault detection, isolation and recovery.

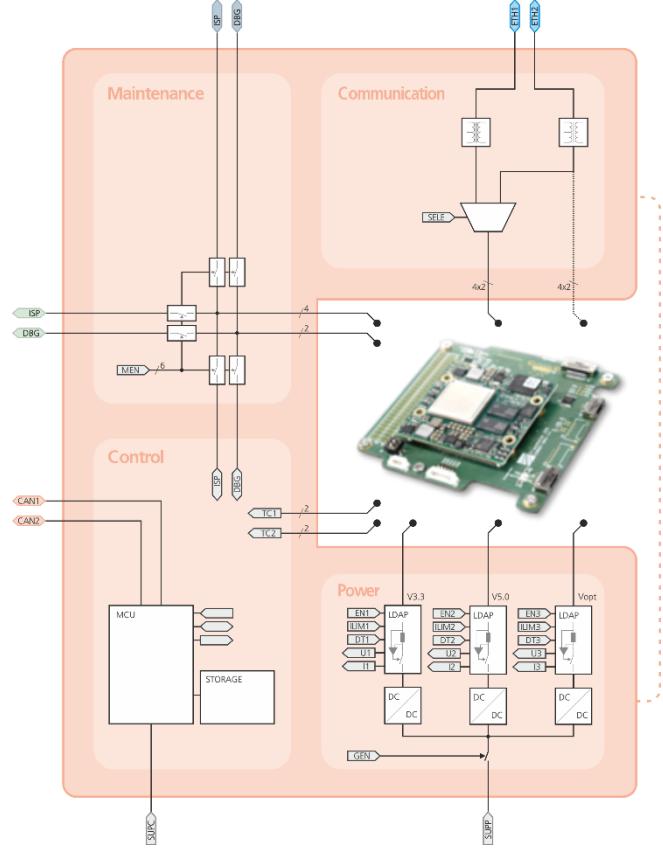


Fig. 2. Functional block diagram of the standard interface

This full paper will present the proposed interface design in detail, including the functional design, implementation, and verification. A comprehensive requirements analysis and architectural rationale will be provided, followed by an in-depth description of the system architecture, including both hardware and software aspects. The implementation and test results of a functional demonstrator will be discussed to evaluate performance, fault isolation, and scalability. Finally, the paper will outline how the proposed architecture could be aligned with the ADHA standard, highlighting the potential for future integration into standardized space system frameworks.

	Communication Interfaces		Power [mA]		Maintenance					
	<i>GB Ethernet</i>	<i>CAN</i>	<i>I2C</i>	<i>UART</i>	<i>GPIO</i>	<i>custom</i>	<i>JTAG</i>	<i>SWD</i>	<i>SBW</i>	<i>SPI</i>
High Performance COTS Systems										
Native GPU										
NVIDIA Jetson nano	●		●	●	●		●		●	
NVIDIA Hopper	●	●	●		●			●	●	
NVIDIA Jetson AGX Xavier	●	●	●		●			●	●	
LattePanda3 Delta	●	●	●		●			●		●
SBCProFIVE NUCR (AMDR 1000)										
Application processors										
OMAP-L138 C6000	●	●					●			●
CogniSAT-XE1	●	●	●	●	●		●	●	●	●
i.MX93	●	●	●	●	●		●	●	●	●
i.MX 95 SMARC SOM	●	●	●	●	●		●	●	●	●
TQMa8MPxL	●	●	●	●	●		●	●	●	●
Ibeos' EDGE Processor	●	●	●	●	●			●	●	
AI accelerator										
PC Engines APU.6B4 Board	●				●			●		
Jetson AGX Orin	●	●	●	●	●		●	●		●
FPGA based										
AMD Kintex UltraScale FPGA KCU105	●				●			●	●	●
Xiling Zynq-7000	●	●	●	●	●		●	●		●
Mercury XU5	●						●		●	

Table 1. Overview of potential COTS based high performance processors

REFERENCES

- [1] R. Di Roberto, E. Brandolini, G. Sparvieri, and F. Graziani, “Best practices on adopting open-source and commercial low-cost devices in small satellites missions,” *Acta Astronautica*, vol. 211, pp. 37–48, Oct. 2023, doi: 10.1016/j.actaastro.2023.06.001.
- [2] S. Busch and K. Schilling, “Robust and efficient OBDH core module for the flexible picosatellite bus UWE-3,” *IFAC Proceedings Volumes*, vol. 46, no. 19, pp. 218–223, 2013.
- [3] S. Busch, P. Bangert, S. Dombrovski, and K. Schilling, “UWE-3, in-orbit performance and lessons learned of a modular and flexible satellite bus for future pico-satellite formations,” *Acta Astronautica*, vol. 117, pp. 73–89, 2015, doi: <https://doi.org/10.1016/j.actaastro.2015.08.002>.
- [4] M. Schimmerohn *et al.*, “ERNST: Demonstrating advanced infrared detection from a 12U CubeSat,” in *SmallSat*, Logan UT, 2022, pp. SSC22-WKVIII-03.
- [5] K. Schäfer, C. Horch, S. Busch, and F. Schäfer, “A Heterogenous, reliable onboard processing system for small satellites,” in *2021 IEEE International Symposium on Systems Engineering (ISSE)*, IEEE, 2021.
- [6] M. Mejia, K. Schaefer, C. Horch, S. Busch, and F. Schaefer, “On-board image processing with FPGA acceleration using deep neural network inference,” presented at the 73rd International Astronautical Congress, Paris, France, 2022, p. IAC-22,B4,6A,x69376.
- [7] C. J. Treudler *et al.*, “ScOSA - Scalable On-Board Computing for Space Avionics,” presented at the 69th International Astronautical Congress, Bremen, 2018.
- [8] D. Rudolph *et al.*, “CSP: A multifaceted hybrid architecture for space computing,” 2014.
- [9] “CubeSat Subsystem Interface Definition (Version 2.0).” UNISEC Europe. [Online]. Available: <http://unisec-europe.eu/wordpress/wp-content/uploads/CubeSat-Subsystem-Interface-Standard-V2.0.pdf>
- [10] Oliver Ruf, “UNISEC Europe CSID – An Advanced Efficient Electrical Interface Standard for CubeSats,” presented at the 4th IAA Conference on University Satellite Missions and CubeSat Workshop, 2017.
- [11] K. Marinis *et al.*, “Advanced Data Handling Architecture (ADHA): On-board Computer (OBC) Module,” in *2023 European Data Handling & Data Processing Conference (EDHPC)*, Juan Les Pins, France: IEEE, Oct. 2023, pp. 1–5. doi: 10.23919/EDHPC59100.2023.10395962.