Implementing mixed-critical applications on next generation multicore aerospace platforms

F. Federici, V. Muttillo, L. Pomante, G. Valente
Center of Excellence DEWS, University of L'Aquila
L'Aquila, Italy
{fabio.federici, vittoriano.muttillo, luigi.pomante, giacomo.valente}@univaq.it

D. Andreetti, D. Pascucci
Thales Alenia Space
Rome, Italy
{danilo.andreetti, dario.pascucci}@thalesaleniaspace.com

***Abstract* — In recent years, there has been a wide diffusion of multicore processing systems. The use of these architectures is of interest in many contexts, including the aerospace industry. As in a multiprocessor architecture multiple processing cores share various resources, their use for the implementation of modern Integrated Modular Avionics (IMA) systems is problematic. IMA systems are in fact mixed-criticality systems and thus, a strict spatial and temporal isolation in accessing resources is essential. This paper analyses the implementation of mixed-criticality applications on multiprocessor platforms, with specific reference to the aerospace domain requirements. Besides detailing application requirements, possible design solutions currently under examination are analyzed, with specific reference to the use of virtualization techniques and assuming ESA's Next Generation Microprocessor as the reference platform.**

Keywords—mixed-criticality; multicore; aerospace; satellites; hypervisors

I. INTRODUCTION

In recent years, multi-core and many-core computing architectures have been widely adopted. This trend has been motivated by the achievement of physical limitations in increasing the performance of single-core processing systems. From this point of view, multi-processor architectures offer many advantages, since they allow to obtain high performances with lower clock frequencies, reduced energy consumption and reduced overall dimensions. Multiprocessor systems are widely used, both in general-purpose computing systems and in the embedded domain. However, their use is still limited in contexts in which reliability and predictability are strict requirements, and certification is mandatory. The use of multicore processing platforms poses several problems with respect to system certification, both from the hardware and from the software perspective [1]. From this point of view, the avionics or aerospace domains are particularly interesting scenarios. The traditional approach for the implementation of avionics and aerospace systems was the so-called federated approach. In a federated architecture, a dedicated computing resource is allocated for each specific function to be implemented. In this way, determinism and non-interference between different functions can be insured. The increase of complexity of these systems, and the need to support an increasing number of functions, however, has shown the limits of this approach in terms of cost, power consumption and complexity of the system. The progressive improvement of the performance of computing architectures facilitated the transition to the so-called *Integrated Modular Avionics* (IMA) approach [2]. In an IMA system, different functions are implemented on the same computing platform. Since the various applications usually have a different level of criticality, an IMA system is inherently a mixed criticality system. In order to guarantee the execution determinism, it is necessary to ensure a rigorous spatial and temporal isolation among the various functions. For this reason, different software specifications supporting the development of IMA systems by means of partitioning have been proposed (e.g. the ARINC 653 standard). There are various established techniques for the practical implementation of partitioned systems (for example, the use of dedicated separation kernels [3]) and the problem of implementing IMA systems using single-processor architectures is quite consolidated at industrial level. The scenario is however more complex in the case of multi-processor systems. In fact, since in this type of architectures the various processing units share a large number of resources (for example memory or communication resources), it is difficult to ensure the spatial and temporal isolation as for single processor systems. In recent years, there have been various proposals for the implementation of mixed-criticality systems on multi-core platforms. For example, several separation kernel supporting symmetric multi-processing are currently available on the market. Other strategies include the use of hypervisor and virtualization [4] in order to manage the access to shared resources. Possible strategies at hardware level were also presented: for example, various deterministic architectures have been proposed in literature [5].

This work aims to analyze the possible approaches in implementing next generation satellite on-board systems using multi-core processing architectures. Possible multi-core architectures used in aerospace will be reviewed, with specific focus on the *Next-Generation Microprocessor* (NGMP) promoted by the *European Space Agency*. The implementation of mixed-criticality systems using multiprocessor architecture will be analyzed and a case study related to the implementation of a reference satellite architecture exploiting the NGMP platform and virtualization technologies will be presented.

II. NEXT-GENERATION MULTICORE PLATFORMS

In recent years, there have been various proposals for multiprocessor architectures for the aerospace domain. For

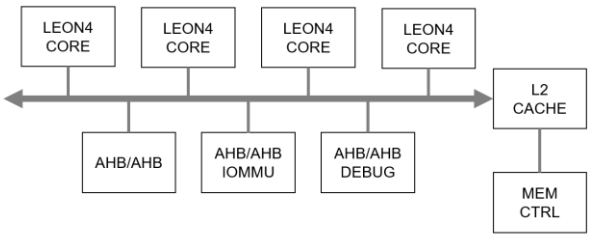

Figure 1 - Schematization of the NGMP architecture

example, *Gaisler* produced the GR712RC [6], a dual-core SPARC v8 processor based on LEON3FT, a fault tolerant, radiation hardened version of its LEON3 processor. *Space Micro* presented the Proton400k-L [7], a dual-core processor based on Freescale's Power architecture. There are also several proposals for many-core architecture designed for the space. For example, *Tilera* released in 2007 the TILE64 [8], a 64 core processor suitable for high-demanding processing applications.

Recently, the European Space Agency promoted the development of a next-generation processor for aerospace [6]. The *Next Generation Microprocessor* (NGMP) is a quad core processor with SPARC v8 architecture developed by Cobham Gaisler in collaboration with the European Space Agency. The microprocessor integrates the latest version of Gaisler LEON processor, the LEON4, a 32-bit processor with a 7-stage pipeline. Each of the LEON 4 core has a private 16 kB data and instruction cache and is connected (Fig. 1) to a shared 256 kB L2 cache through a 128-bit AHB bus that supports a round-robin arbitration policy. The L2 cache is connected to the memory controller via a single communication channel shared by all cores. The cores communicate with peripherals and other I/O devices by means of three bridges towards additional AMBA bus (a bridge towards the system debug bus, a bridge towards slave I/O bus, and a bridge towards master I/O bus, also supporting IOMMU functionality). The presence of the IOMMU is a substantial improvement introduced in the architecture of the NGMP over previous systems based on LEON processors. The LEON4 is able to ensure an improvement in performance compared to previous generations of LEON processors. The performances of the NGMP have been analyzed in several studies. Some of these studies [9] also addressed the problem of isolation, even if in relation to specific architectural features (e.g. analysis of cache performances and interference).

III. Implementing Mixed-Critical Applications On Multicore Platforms

As said, a mixed-criticality system requires a rigorous partitioning of the different functionalities. The design usually involves the following distinct phases: definition of application modules, definition of partitions, and definition of partitions' scheduling. The first step allows the identification of all the different modules to be allocated on the single processing platform. Each module will be characterized by its own level of criticality. The second step allows the definition of the different partitions. Each partition can be thought as an isolated container in which different modules can be executed. Usually a partition will include software modules sharing functional similarities and the same level of criticality. Finally, the third step allow the definition of a scheduling plan for the various partitions. In literature, only few contributions focuses on the problem of the definition of partitions. For example, Salazar et al. [10] described the partitions' definition problem as a resource allocation problem and proposed a method based on graph coloring for the automatic definition of the partitions. In the case of a multiprocessor platform, it will be also necessary to consider the allocation of the different partitions over the multiple computing resources. Several studies have considered the problem of the scheduling of mixed-criticality systems in the case of multi-processor platforms. One of the most diffused strategies is the so-called MC2 framework [11], a two level scheduling scheme able to exploit the multiple processing cores while satisfying also the requirements of isolation and determinism.

The implementation and scheduling of partitions requires a specific software support. Among possible software techniques, virtualization appears of specific interest for the implementation of aerospace applications. In a virtualized system, applications are executed on top of a hypervisor emulating the native hardware, abstracting in this way the real platform. Hypervisors usually support:

- virtualization of the platform, allowing the re-use of legacy code. In the case of so-called para-virtualized solutions the lowest-level software layers have to rely on hypervisor's services, thus requiring an adaptation;
- spatial and temporal isolation of virtual machines, which actually implements the isolated containers described above;
- different scheduling schemes and additional support services in operating the platform (e.g. health monitoring).

The use of virtualization tools appears as a feasible way to implement partitioned systems, efficiently managing the multi-processor scenario. The following section discusses the possible design of a satellite reference architecture, with explicit reference to the use of next-generation processor described in section II and the techniques for the management of mixed-criticality issues introduced above.

IV. A real world case-study

Usually, the software architecture of a satellite can be subdivided into:

- *Command and Data Handling*: the part of the software supervising and managing the data flows within the system;
- *Guidance, Navigation and Control*: the part of the software supervising the management of the navigation operations, such as the attitude maintenance, control of the trajectory, etc.;

- *Payload software*: the part of the software dedicated to the management of the instrumentation. Usually executed by a dedicated computer system, it manages the communication of data to the central processing unit.

Specifically, this case study aims to analyze the command and data handling sub-system. The specific functionalities that should be kept into account in the application design are the management of telecommands and telemetry data. In order to implement these functionalities, two different modules are needed:

- *I/O manager*: a module collecting I/O requests from the applications and performing the required transactions on communication links, satisfying specific real-time requirements;
- *Telecommand and Telemetry Manager (TCTM)*: a module receiving, dispatching, and processing commands, and managing global system telemetry.

These modules are implemented in two distinct partitions, one having exclusive access to I/O devices and acting as a server for the other partition (providing and dispatching telemetry data and commands). In order to evaluate proposed scheme, a reference demo platform has been implemented (Fig. 2). A Gaisler GR-CPCI-LEON4-N2X board, a functional prototype of the ESA Next Generation Microprocessor, represents the processing platform. Two additional boards simulate the sources of telemetry data. A test console simulates instead the command receiving interface.

In order to implement the system partitioning, a hypervisor is included. Two different possible technical solutions, both relevant for the aerospace domain, are currently under investigation:

- FentISS Xtratum, an open-source, bare metal hypervisor implementing the paravirtualization principle.
- SYSGO PikeOS, a platform providing RTOS, type I hypervisor and paravirtualization functionalities.

Both XtratuM and PikeOS support spatial and temporal isolation and allow the implementation of different scheduling schemes for described software modules. Preliminary work consisted in adding support for the selected board and required communication interfaces (serial port, Ethernet and SpaceWire router) for both the hypervisors. Proposed study will allow:

- to assess the NGMP in relation to the requirements of a real aerospace application (performance, isolation);
- to evaluate different multicore scheduling strategies, and the impact of virtualization on real time requirements;
- to compare the performances of different virtualization technologies.

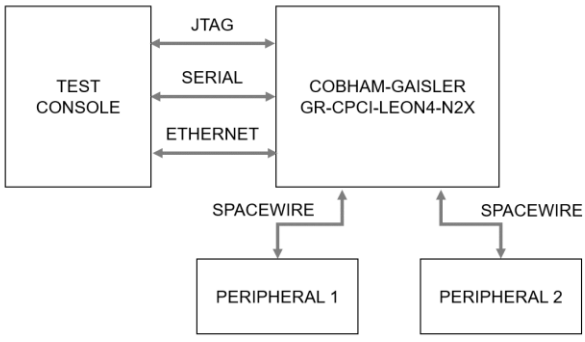

Figure 2 - Schematization of the proposed reference platform

V. CONCLUSIONS AND FUTURE DEVELOPMENTS

This work analyzed the issues related to the implementation of mixed-criticality systems over multi-processor computing architectures. Specifically, the design of a satellite reference platform has been considered as a valid reference application. A possible approach for the design of the software subsystem, with specific reference to a next generation multicore processing architecture has been detailed. In the next future, presented platform will be implemented and the proposed software architecture will be verified and benchmarked.

ACKNOWLEDGMENT

This work has been partially funded by the ARTEMIS-JU AIPP 2013 621429 EMC2 (*Embedded Multi-Core systems for Mixed Criticality applications in dynamic and changeable real-time environments*) European project.

REFERENCES

[1] L.M. Kinnan, "Use of multicore processors in avionics systems and its potential impact on implementation and certification", IEEE/AIAA 28th Digital Avionics Systems Conference, 23-29 Oct. 2009.

[2] C.B. Watkins, R. Walter, "Transitioning from federated avionics architectures to Integrated Modular Avionics", IEEE/AIAA 26th Digital Avionics Systems Conference, pp.2.A.1-1,2.A.1-10, 21-25 Oct. 2007.

[3] Y. Li, R. West, and E. Missimer. The quest-v separation kernel for mixed criticality systems. Proc. 1st WMC, RTSS, pages 31–36, 2013.

[4] A. Crespo, I. Ripoll, M. Masmano, "Partitioned Embedded Architecture Based on Hypervisor: The XtratuM Approach", European Dependable Computing Conference (EDCC), 28-30 April 2010.

[5] A. Wasicek, C. El-Salloum, H. Kopetz, "A System-on-a-Chip Platform for Mixed-Criticality Applications", 13th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC), 2010 , 5-6 May 2010.

[6] J. Andersson, D. Hellstrom, S. Habinc, R. Weigand, L. Fossati, "Current and Next Generation LEON System-On-Chip Architectures for Space", Workshops on Spacecraft Flight Software, 2012.

[7] Space Micro, Inc., "Proton400k-™ Single Board Computer", 2013.

[8] Tilera Corp., "Tile Processor User Architecture Manual", Rel. 2.4, 2011.

[9] M. Fernandez, R. Gioiosa, E. Quinones, L. Fossati, M. Zulianello, F. Cazorla, "Assessing the Suitability of the NGMP Multi-core Processor in the Space Domain", Proc. 10th ACM international conference on Embedded software - EMSOFT 2012, 2012.

[10] E. Salazar, A. Alonso, "An Automatic System Partitioning Algorithm for Mixed Criticality Systems", Actas de las XXXVI Jornadas de Automática, 2015.

[11] M. S. Mollison, J. P. Erickson, "Mixed-Criticality Real-Time Scheduling for Multicore Systems". Proc. 10th IEEE International Conference on Computer and Information Technology, 2010.