Introducing Dynamic Adaptation in High Performance Real-Time Computing Platforms for Sensors

Baptiste Goupille-Lescar, Eric Lenormand, Christine Morin, Nikos Parlavantzas

Introducing Dynamic Adaptation in High Performance Real-Time Computing Platforms for Sensors

Baptiste Goupille-Lescar
Eric Lenormand
Thales Research & Technology
Palaiseau, France
firstname.lastname@thalesgroup.com

Christine Morin
Inria
IRISA, Rennes, France
christine.morin@inria.fr

Nikos Parlavantzas
INSA Rennes
IRISA, Rennes, France
nikos.parlavantzas@irisa.fr

ABSTRACT

In high-end, data-intensive embedded sensor applications (radar, optronics), the evolution of algorithms is limited by the computation platform capabilities. These platforms impose Size, Weight and Power (SWaP) restrictions on top of reliability, cost, security and (potentially hard) real-time constraints. Thus mostly static mapping methods are used, negating the system's adaptation capabilities. Through the study of several industrial use-cases, our work aims at mitigating the aforementioned limitations by introducing a low-latency dynamic resource management system derived from techniques used in large-scale systems such as cloud and grid environments. We expect that this approach will be able to guarantee non-functional properties of applications and provide Quality of Service (QoS) negotiation on heterogeneous platforms.

CCS CONCEPTS

• **Computer systems organization** → **Parallel architectures**; **Embedded systems**; *Real-time systems*; • **Computing methodologies** → **Modeling and simulation**; • **Applied computing** → *Military*;

KEYWORDS

Embedded systems, Heterogeneous Architecture, Real-time, Mapping, SWaP, Dynamic resource management, Context-awareness, Dynamic adaptation

1 INTRODUCTION

Currently, most embedded systems meet SWaP and cost constraints by the use of dedicated components and static mapping approaches based on worst-case scenarios. This, coupled with the emergence of workloads integrating both hard and soft real-time and best-effort applications and the increase of their variability, results in a massive over-provisioning and under-utilization of resources. Moreover, while this method allows the design of efficient and secure systems, it nearly nullifies their adaptation and evolution capabilities by preventing the deployment of highly variable or opportunistic applications for smart sensing.

To cope with these limitations, we propose an intelligent resource management system, able to fulfil low-latency run-time requests for application execution while providing non-functional guaranties (timing, heat dissipation, etc.). To achieve this, we seek to inspire ourselves from large-scale resource managers found in cloud or grid infrastructures and make the most of application profiling to reach both high-level predictability and low mapping latency.

Section 2 provides a quick summary of related research both in embedded and large-scale computing infrastructure systems communities. Section 3 describes our approach using a surface Active Electronically Scanned Array (AESA) radar use-case. Finally Section 4 draws some conclusions and perspectives.

2 RELATED WORK

Studies on current and emerging trends show that first, considering the nature of current high performance embedded applications, the addition of a supplementary applicative layer capable of handling dynamic resource management of applications is becoming an increasing necessity [8]. This is primarily due to the increase of workload variability and growing concerns about optimization of some non-functional properties of the systems such as power consumption, heat dissipation, reliability or QoS.

Then since more and more embedded applications need to offload tasks to a cloud environment to provide additional services to the user and to improve the device's battery life while conserving their responsiveness. The need for a certain level of real-time compliance is growing in certain cloud computing fields such as cloud gaming and IoT [3]. In the following sections, some relevant dynamic mapping solutions coming from both embedded and large scale systems studies are presented.

2.1 Embedded Systems

There exist numerous efforts on efficient application mapping onto Network-on-Chip (NoC) based architectures. However, they mostly focus on Design Space Exploration (DSE) methods to find optimal mappings at design-time and abstract the processing elements of the network. For example, in [7] the authors use a DSE evolution algorithm to find a set of Pareto-optimal multi-application mappings taking into account latency and power consumption.

Only few teams address run-time mapping approaches on MPSoCs and when doing this, they often neglect the network aspect of the problem, as in [6] where run-time adaptation is made by adjusting mappings found at design-time. Since real-time execution of applications is the major limiting factor in embedded systems, a number of studies aim at providing timing guarantees for either hard real-time applications or mixed-criticality systems. However, when trying to meet those guarantees, works often aim at aeronautic or automotive certification and thus discard dynamic methods as in [1] or focus on single-processor architectures [4].

2.2 Large-Scale Systems

On the other hand, in cloud, a vast majority of works focus on purely dynamic approaches with close to no prior knowledge of the applications and rely on cloud elasticity to compensate for QoS variation by allocating/deallocating and/or migrating virtual resources [9]. However, a fair number of studies try to incorporate predictability via the use of priorities or isolation mechanisms as in [2]. Unfortunately, due to the ever-changing nature of cloud environments, most "real-time" cloud resource managers react to deadline misses and don't prevent them. Thus, there exists no satisfying solution able to accurately provide real-time guarantees. On the other hand the high-performance applications commonly executed in grid environments are closer to ours than cloud applications but the time-scale at which the system evolves is completely different and the mapping methods used are often very costly and generate a high latency.

3 DYNAMIC RESOURCE MANAGEMENT

In this section, the main use-case we are studying, with its particular constraints and objectives is first introduced. Then our approach for the deployment of dynamic applications on these architectures is presented.

3.1 Use-Case

The main use-case addressed by our study is a joint work with a Thales team whose objective is to develop next generation surface multi-function radar algorithms. The targeted heterogeneous platform is constrained by a limited amount of processing and communication resources. A mission management system is already implemented and able to generate, depending on the observed environment and external inputs (automated weaponry, operator), a list of aperiodic job requests to the antenna every few milliseconds. Each job has a variable processing time depending on the observed context. Thus, on top of being computationally intensive and having to deal with both soft and hard real-time requirements, the created workload is highly variable and unpredictable. This, in turn, imposes the design of largely oversized architectures to be able to absorb the computing load even under stress scenarios. However, in this case an intelligent dynamic resource management could both allow the execution of opportunistic applications under normal load and provide timing guarantees for critical applications under stress without having to over-dimension the platform.

3.2 Our Approach

In this study, we aim at providing timing and SWaP guarantees to applications executing on a dynamic heterogeneous platform. We seek to achieve this by the mean of an intelligent resource manager using mapping methods inspired from the ones present in cloud and grid environments. These methods are not usable in their current form, since we target much more constrained platforms and hard real-time applications. However, we have the advantage of having a certain prior knowledge of the applications, even if data-dependent applications tend to have a highly variable processing time. To evaluate and adapt different mapping strategies, a simulation framework (shown in Fig.1) is under development using an industrial high-precision AESA simulator, the Ptolemy II framework [5] and a platform simulator calibrated using actual devices. This allows us to develop mapping methods using the MAPE loop (Monitor, Analyse, Plan, Execute) and efficiently test them under real-life scenarios as well as to measure the impact of the computing platform architecture on application placement and to evaluate the possible addition of new radar applications.

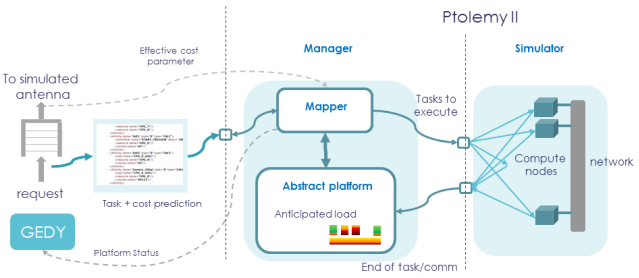

Figure 1: Simulation environment

4 CONCLUSION

In this paper we define and present a first analysis of the problem of dynamic mapping of real-time applications on a heavily constrained embedded heterogeneous architecture. We are focusing on making the most of design-time studies and on using cloud computing-inspired mapping methods to provide timing guarantees as well as flexibility to dynamic applications. This work is supported by the study of several industrial use-cases.

REFERENCES

[1] G. Giannopoulou, N. Stoimenov, P. Huang, and L. Thiele. 2013. Scheduling of mixed-criticality applications on resource-sharing multicore systems. In *2013 Proceedings of the International Conference on Embedded Software (EMSOFT)*. 1–15. https://doi.org/10.1109/EMSOFT.2013.6658595

[2] Bhavesh Khemka, Ryan Friese, Sudeep Pasricha, Anthony A. Maciejewski, Howard Jay Siegel, Gregory A. Koenig, Sarah Powers, Marcia Hilton, Rajendra Rambharos, and Steve Poole. 2015. Utility maximizing dynamic resource management in an oversubscribed energy-constrained heterogeneous computing system. *Sustainable Computing: Informatics and Systems* 5 (March 2015), 14–30. https://doi.org/10.1016/j.suscom.2014.08.001

[3] Dimosthenis Kyriazis, Andreas Menychtas, George Kousiouris, Michael Boniface, Tommaso Cucinotta, Karsten Oberle, Thomas Voith, Eduardo Oliveros, and SÃűren Berger. 2014. A real-time service oriented infrastructure. *Journal on Computing (JoC)* 1, 2 (2014). http://dl6.globalstf.org/index.php/joc/article/viewFile/907/970

[4] Haohan Li and S. Baruah. 2010. An Algorithm for Scheduling Certifiable Mixed-Criticality Sporadic Task Systems. In *Real-Time Systems Symposium (RTSS), 2010 IEEE 31st*. 183–192. https://doi.org/10.1109/RTSS.2010.18

[5] Claudius ptolemaeus. 2014. *System design, modeling, and simulation: using Ptolemy II*. Vol. 1. Ptolemy. org Berkeley.

[6] Wei Quan and A.D. Pimentel. 2013. A scenario-based run-time task mapping algorithm for MPSoCs. In *2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC)*. 1–6.

[7] J. Sepulveda, M. Strum, Wang Jiang Chau, and G. Gogniat. 2011. A multi-objective approach for multi-application NoC mapping. In *2011 IEEE Second Latin American Symposium on Circuits and Systems (LASCAS)*. 1–4. https://doi.org/10.1109/LASCAS.2011.5750275

[8] A.K. Singh, M. Shafique, A. Kumar, and J. Henkel. 2013. Mapping on multi/many-core systems: Survey of current and emerging trends. In *2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC)*. 1–10. https://doi.org/10.1145/2463209.2488734

[9] D. Warneke and O. Kao. 2011. Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud. *IEEE Transactions on Parallel and Distributed Systems* 22, 6 (June 2011), 985–997. https://doi.org/10.1109/TPDS.2011.65