Multiprocessor Scheduling Analysis: From UML MARTE to dynamic Priority Time Petri Nets Walid Karamti

Walid Karamti [1], Adel Mahfoudhi [1], Mohamed Abid [1]

[1] National Engineering School of Sfax
Road Sokkra, Km 3 Sfax,
Tunisia
walid.karamti@ceslab.org, Adel.mahfoudhi@fss.rnu.tn; Mohamed.abid@enis.rnu.tn

Abstract. Real-time systems (RTS) are now omnipresent in modern societies in several domains such as avionics, control of nuclear power station, multimedia communications, robotics, systems on chip, etc. Hence, RTS are characterized by complex applications that require powerful architectures to satisfy them. In practice, such architectures can be specified via a powerful single-processor or a multiprocessor architecture composed of a set of low processors. For the same power, a multiprocessor architecture is much cheaper, which represents an economical motivation for researchers to target such kind of architectures.

It is noteworthy to mention that two major families of multiprocessor scheduling can be distinguished. The first one is the global scheduling family in which each application task can migrate among the processor resources to be executed. However, the cost of migration is so important and without an optimal scheduling algorithm (1). As for the second one, it is the partitioned family, which is based on sharing tasks on different processors without migration. In this family, the optimality can be dedicated because the multiprocessor scheduling is reduced to a single-processor scheduling where optimal algorithms exist (2). This family includes two events, the first of which is assigning tasks to processors and the second is analyzing the scheduling of each partition (3). An important key challenge is to detect the errors of scheduling as early as possible in order to minimize the costs for its correction.

Unified Modeling Language (UML) profiles promote an adequate solution to represent different system views with their embedded and real time features. The recent Modeling and Analysis of Real-Time and Embedded systems (MARTE) profile (4) adopted by the Object Management Group (OMG), fosters the building of models that support the specification of scheduling analysis problem. Though it is a powerful and advanced standard for annotating models with the required information for performing scheduling analysis, it does not provide a technique for verifying models. The UML extension lacks a tool to check system properties and constraints. So, a need emerges for mapping used models to an external platform for scheduling test. Talking about scheduling analysis methods, we indicates that both of the specification and the analysis of the RTS with model checking has proved its ability to

allowing early and economical detection of errors at an early stage of the design process. This explains the growing popularity it enjoys in recent researches.

The Petri Nets (5) (PNs) are known as the adequate model checking formalism for scheduling analysis. Indeed, time, parallel processing and synchronization are primordial characteristics for scheduling analysis that PNs is able to specify.

The main objective of our works is to propose a new PNs formalism able to analyze the scheduling of a partitioned multiprocessor system. Hence, we aim to propose a mapping process from the standard MARTE (4) to the new PNs formalism in order to check the property of the system.

In previous works, we are interested to study the single-processor scheduling analysis trough a new Petri Nets extension: Priority Time Petri Nets (PTPN) (6), (7). The PTPN deal with fixed priorities. In fact, it can be used in reduced types of multiprocessor systems because most of those systems are characterized with dynamic priorities. We distinguish in literature an absence of PNs extensions supporting dynamic priority. Thus, we are proposed the first PNs able to support the dynamic Priority-driven scheduling, called dynamic Priority Time Petri Nets: dPTPN (8).

First, the dPTPN is introduced and its semantics is presented in (8) to deal with the Least Laxity First policy (LLF) and a set of independents tasks. Second, we have proved that our formalism is able to deal with even dynamic Priority-driven scheduling policy via its capacity to specify the Earliest Deadline First (EDF) in (9). Indeed, we have demonstrated the scheduling analysis via an experiment characterized with dependents tasks. Next, based on the object modeling, we have proposed a new modeling method to reduce the complexity of the dPTPN model for scheduling analysis (10). In fact, a new dPTPN object is defined (TaskC) and trough its instances the scheduling model is constructed.

Our next challenge is the checking of the model properties. In this stage, we just make sure that no stop-marking exists to deduce that the considered RTS is schedulable. But, it is important to verify properties such as vivacity, boundedness, safety, … before declaring the system as schedulable or non-schedulable.

References

1. On-line scheduling of real-time tasks. Leung, S.H et Kwang, J.R. 1988. IEEE Real-Time Systems Symposium. pp. 244–250.
2. Scheduling algorithms for multiprogramming in a hard-real-time environment. Layland, C.L et James, W. 1, New York, USA : J. ACM, 1973, Vol. 20. 10.1145/321738.321743.
3. Real time scheduling theory: A historical perspective. L.Sha, T. Abdelzaher, K.E. arz´ en, A. Cervin, T. Baker, A. Burns, G. Buttazzo, M. Caccamo, J. Lehoczky, and K.A. Mok. 2-3, Norwell, MA, USA : Real-Time Systems, 2004, Vol. 28. 10.1023/B:TIME.0000045315.61234.1e.
4. OMG, Object Management Group. A UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems. s.l. : Object Management Group, 2008.
5. Fundamentals of a theory of asynchronous information flow. Petri, C. A. 1962. IFIP Congress. pp. 386–390.
6. A Petri Net Extension for Schedulability Analysis of Real Time Embedded Systems. HadjKacem, Y, et al. Las Vegas, USA : s.n., 2010. PDPTA. pp. 304-314.
7. Compositional specification of real time embedded systems by priority time Petri Nets. Mahfoudhi, A, et al. 2012, The Journal of Supercomputing, pp. 1-26.
8. A Formal Method for Scheduling Analysis of a Partitioned Multiprocessor System: dynamic Priority Time Petri Nets. Karamti, W, et al. Rome, Italy : INSTIC, 2012. PECCS. pp. 317-326.

9. Using dynamic Priority Time Petri Nets for scheduling analysis via Earliest Deadline First policy. Karamti, W, Mahfoudhi, A et HadjKacem, Y. Madrid, Spain : IEEE, 2012. ISPA.

10. Hierarchical Modeling with dynamic Priority Time Petri Nets for Multiprocessor Scheduling Analysis. Karamti, W, Mahfoudhi, A et HadjKacem, Y. Las Vegas, USA : s.n., 2012, ESA