CISTER
Research Center in
Real-Time & Embedded
Computing Systems

Technical Report

Online Admission of Parallel Real-Time Tasks

Cláudio Maia
Luís Nogueira
Luis Miguel Pinho

CISTER-TR-150701

Online Admission of Parallel Real-Time Tasks

Cláudio Maia, Luís Nogueira, Luis Miguel Pinho

CISTER Research Center

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail:

http://www.cister.isep.ipp.pt

Online Admission of Parallel Real-Time Tasks

Cláudio Maia, Luís Nogueira, Luis Miguel Pinho
CISTER-ISEP / INESC-TEC, Porto, Portugal
{crrm, lmn, lmp} @isep.ipp.pt

I. Context and Problem Description

Parallel real-time tasks can be assigned into a multiprocessor system in many different ways, with regards to the schedulability of the task system. In the best case a parallel task may be completely parallelized, i.e., when the platform has a number of processors no smaller than the degree of parallelism of the task and the interference from higher priority tasks does not compromise its parallel behaviour; in the worst-case (considering the parallelism) the task needs to be completely serialized. This latter scenario requires the adjustment of the timing properties of the parallel task so that the worst-case parallel behaviour is considered in the analysis in order to avoid missing any deadlines. Moreover, this scenario is very pessimistic with respect to the parallel structure of the task. Considering parallelism and real-time, it is important to achieve a good trade-off between timing guarantees and the parallel execution of tasks so that efficient execution of such tasks may be possible.

The schedulability of a task system composed of parallel tasks can be analysed by using different techniques (e.g., response-time analysis [1], decomposition) and considering different task models (fork/join, synchronous or directed acyclic graphs). Decomposition-based techniques ([2], [3]) decompose a parallel task into a set of sequential tasks (usually by assigning intermediate deadlines between sub-tasks in order to maintain the precedence constraints between them), so that existing sequential schedulability analysis techniques for multiprocessors can be applied. An important aspect of these techniques is that they require the task structure to be known offline in order to apply decomposition. Alternatively, non-decomposition techniques such as the one proposed in [4] do not require offline knowledge of the tasks' structure.

While the above techniques are suitable for offline schedulability analysis, one cannot simply apply them in dynamic systems (i.e., systems where tasks may arrive and leave the system at any time), due to their algorithmic complexity. In dynamic systems task acceptance is done according to an admission control test that takes as input the current state of the system and the new task, and provides as output a decision whether the system remains schedulable after accepting the task. As these decisions must be made online, the test must be as efficient as possible so that the test itself does not become an overhead.

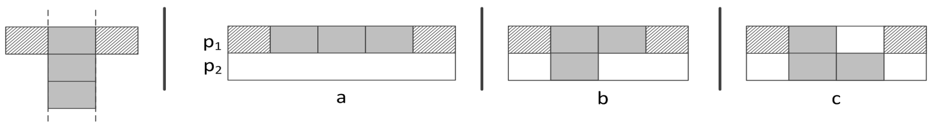

Fig. 1. Example of possible allocations of a fork/join parallel task with three segments, with one, three and one sub-tasks, respectively.

The analysis of such a system not only depends on the scheduling algorithm, but also on the properties of each parallel task. Let us consider a simple fork/join task composed of a parallel region with 3 sub-tasks and a platform of 2 identical cores as depicted in Figure 1 [1]. If a fully-partitioned scheduling algorithm is used, then the parallel task would have to be serialized, in order to be executed on a processor (see Figure 1, a); or if the task structure is known in advance, then it is possible to consider each sub-task individually, their release offsets and intermediate deadlines, and use a partitioned scheduling approach to assign each sub-task into a core (see Figure 1, b and c). Therefore, this will allow us to avoid migration overheads. If a global approach is used instead, then all the depicted allocations (a, b and c) may occur at runtime for different task instances, depending on the state of the system at a particular time instant.

The above example illustrates that different allocations are possible at runtime. Depending on the scheduling approach, the schedulability of the system might be affected according to several factors as, for instance, the priority of the parallel task under admission. Moreover, the example considers that the parallel task has a utilization not greater than one, otherwise it could not be serialized and assigned into a single processor (see Figure 1, a). If one considers the online admission of parallel tasks with a utilization greater than 1 (which is the most likely format for a parallel task), then one is faced with the forced parallelization of the task in order to obtain task schedulability.

Global approaches usually consider that a parallel task may have a utilization greater than one but not greater than the number of processors in the system, and these may not require the knowledge of the task structure in advance, at the cost of some pessimism in the analysis. On the other hand, if one considers partitioned approaches, then the task structure must be known in advance in order to decide where to allocate each sub-task, and therefore apply a schedulability test on each core. This leads to an online bin-packing problem, which is a NP-hard problem.

An admission control test that considers parallel tasks should be able to support all possible structures and timing properties of the tasks. Nevertheless, while it is possible to adapt existing techniques to consider parallel tasks with utilization no greater than 1, to the best of our knowledge there is **no admission control test that fits parallel real-time tasks with utilization greater than 1**.

[1]The figure does not depict any interference the task may suffer and the presented allocations are also valid if processor p_1 is switched by p_2 and vice-versa.

II. POSSIBLE SOLUTIONS

Possible solutions for the problem of online admission control of parallel tasks may include variations of existing techniques for partitioned approaches such as the one proposed in [2]. Specifically, this approach decomposes a parallel task into a master thread which is assigned to one core in such a way that it fills the capacity of the core (100%). Then, the remaining parallel threads of the task are treated as constrained-deadline sub-tasks which are analysed by taking into account a release offset and intermediate deadlines. These constrained-deadline sub-tasks are assigned to processors by using the heuristic proposed in [5] (entitled FBB-FFD) which can be applied to task sets composed of sequential sporadic tasks if one selects partitioned deadline-monotonic scheduling. This approach has two limitations: (1) it requires an empty processor where the master thread of each task can be allocated; (2) it needs to be generalized to support different parallel task structures (originally it only supports fork/join tasks). From a conceptual standpoint, it seems to be a promising starting point with polynomial-time complexity. Following the same principle, another possible solution is the combination of the decomposition of parallel tasks with the constant-time admission control-test proposed by [6] and the first-fit decreasing (FFD) heuristic, which can be used for testing the schedulability of tasks scheduled by using a partitioned-EDF scheduler.

The previous approaches show that a possible solution to the online admission control for parallel real-time tasks using partitioned scheduling is one that during runtime decomposes a parallel task into its sub-tasks, and then each of these sub-tasks is subject to admission control during runtime. Then, one is faced with an online bin-packing problem that can be solved by combining an allocation heuristic such as FFD and an efficient schedulability test to evaluate the schedulability of the sub-tasks.

Global scheduling solutions for admission control of parallel tasks are more challenging to develop due to the need of knowing the global state of the system when new a task arrives. This may be affected by the migration of tasks that occurs among the cores. Another important aspect that should be taken into account is the interference that a task may suffer from the contention occurring in the hardware resources, which is difficult to determine.

III. DISCUSSION

This paper describes the problem of online admission control of parallel tasks with a utilization greater than one in a multiprocessor system. Considering these tasks, not only their allocation is important, but also the schedulability test that is used when performing the admission control of the tasks. As presented in the previous section, it seems that a natural candidate solution for the online admission of parallel tasks is the decomposition of tasks into sub-tasks and then the resulting sub-tasks are submitted to admission control before assigning them into the processors. However, this approach is not the perfect solution as it presents problems concerning the need for synchronization and the computation of release offsets which may suffer some release jitter due to the preemption of a sub-task that precedes some parallel region. Besides the presented solutions, no other solution is envisioned yet and to the best of our knowledge this is still a limitation in the existing literature.

Having a solution for this problem is important for the community, as it allows one to employ parallel real-time tasks in more dynamic scenarios where applications (e.g., multimedia applications) still need to provide timing guarantees. At the same time, with a solution for admission control, applications are able to take advantage of the multiprocessor platform in order to improve their efficiency as well as the utilization of the platform.

Other general questions may arise in the study of this problem, such as if it is possible to perform online admission of parallel tasks without knowing their internal structure. Nevertheless, such questions may have an answer after finding a good initial solution for this problem.

ACKNOWLEDGEMENTS

This work was partially supported by National Funds through FCT/MEC (Portuguese Foundation for Science and Technology) and when applicable, co-financed by ERDF (European Regional Development Fund) under the PT2020 Partnership, within project UID/CEC/04234/2013 (CISTER Research Centre); also by FCT/MEC and ERDF through COMPETE (Operational Programme 'Thematic Factors of Competitiveness'), within project(s) FCOMP-01-0124-FEDER-020447 (REGAIN); by the European Union, under the Seventh Framework Programme (FP7/2007-2013), grant agreement nr. 611016 (P-SOCRATES); also by FCT/MEC and the ESF (European Social Fund) through POPH (Portuguese Human Potential Operational Program), under PhD grant SFRH / BD / 88834 / 2012.

REFERENCES

[1] C. Maia, M. Bertogna, L. Nogueira, and L. M. Pinho, "Response-time analysis of synchronous parallel tasks in multiprocessor systems," in *Proceedings of the 22nd International Conference on Real-Time Networks and Systems*, ser. RTNS '14. ACM, 2014, pp. 3–12.

[2] K. Lakshmanan, S. Kato, and R. R. Rajkumar, "Scheduling parallel real-time tasks on multi-core processors," in *Proceedings of the 2010 31st IEEE Real-Time Systems Symposium*, ser. RTSS '10. IEEE Computer Society, 2010, pp. 259–268.

[3] A. Saifullah, K. Agrawal, C. Lu, and C. Gill, "Multi-core real-time scheduling for generalized parallel task models," in *Real-Time Systems Symposium (RTSS), 2011 IEEE 32nd*, Nov 2011, pp. 217–226.

[4] J. Li, K. Agrawal, C. Lu, and C. Gill, "Analysis of global edf for parallel tasks," in *Real-Time Systems (ECRTS), 2013 25th Euromicro Conference on*, July 2013, pp. 3–13.

[5] N. Fisher, S. Baruah, and T. Baker, "The partitioned scheduling of sporadic tasks according to static-priorities," in *Real-Time Systems, 2006. 18th Euromicro Conference on*, 2006, pp. 10 pp.–127.

[6] A. Masrur, S. Chakraborty, and G. Farber, "Constant-time admission control for partitioned edf," in *Real-Time Systems (ECRTS), 2010 22nd Euromicro Conference on*, July 2010, pp. 34–43.