A Survey of Multi-Processor Scheduling For Hard Real-Time Systems

Xin Lin[a], Xiaorong Zhu[a], Lijia Liu[a]

[a]*Department of Computer Science, The University of Texas at Austin*

Abstract

In class, both of scheduling algorithms [1] and priority inheritance protocols [2] in the context of a single processor were examined in details. Nevertheless, the emergence and popularity of distributed computing system gave rise to the need to solve multi-processor scheduling and priority inheritance problems. As the supplementary study, this paper surveys existing scheduling algorithms in the context of multiple processors. The very first section outlines the background of multi-processor scheduling problems, as well as system models, terminology, and the metrics of scheduling algorithms. After that, partitioned scheduling and global scheduling, as the primary objects of our research, will be fully explored. Moreover, we will also give brief sketch to the hybrid approaches of partitioned scheduling and global scheduling.

Keywords: System, Scheduling Algorithm, Task Management

1. Introduction

1.1. Problem Defintion

1.2. Preview Of Related works

1.3. Paper Organization

0. Background and introduction
1. System Models
2. Partitioned Scheduling
3. Global Scheduling
4. Hybrid Approach
5. Conclusion and Discussion

2. System Models

2.1.

3. Partitioned Scheduling

In this section, we will review some partitioned approaches to multiprocessor real-time scheduling.

3.1. Characteristic of Partitioned Scheduling

3.2. RMNF

3.3. RMFF

3.4. EDF-FF

3.5. EDF-BF

3.6. Comparision

4. Global Scheduling

In this section, we will review some global approaches to multiprocessor real-time scheduling.

4.1. Overview

The partitioned scheduling paradigm has several advantages over the global approach. First, once tasks are allocated to processors, the multiprocessor real-time scheduling problem becomes a set of single processor real-time scheduling problems, one for each processor, which has been well-studied and for which optimal algorithms exist. Second, not migrating tasks at run-time means reduced runtime overhead as opposed to migrating tasks that may suffer cache misses on the newly assigned processor. If the task set is fixed and known a-priori, the partitioned approach provides appropriate solutions

The global scheduling paradigm also has advantages, over the partitioned approach. First, if tasks can join and leave the system at run-time, then it may be necessary to reallocate tasks to processors in the partitioned approach [3]. Second, the partitioned approach cannot produce optimal real-time schedules one that meets all task deadlines when task utilization demand does not exceed the total processor capacity for periodic task sets [14], since the partitioning problem is analogous to the bin-packing problem which is known to be NP-hard in the strong sense. Third, in some embedded processor architectures with no cache and simpler structures, the overhead of migration has a lower impact on the performance [3]. Finally, global scheduling can theoretically contribute to an increased understanding of the properties and behaviors of real-time scheduling algorithms for multiprocessors.

Although there are various categories of global scheduling algorithms, the focus of this paper is on the Global Dynamic Priority Scheduling. In the following subsection, it will be chacterized in details.

4.2. Global Dynamic Priority Scheduling

In this subsection, we will present our in-depth exploration to the track of global dynamic priority scheduling algorithm. To the best of our knowledge, a number of global dynamic priority scheduling algorithms are optimal for periodic tasksets with explicit or implicit deadlines. For example, Proportionate Fairness algorithm and its variants including PD, PD^2, ERFair, BF, SA [3], and LLREF [4] as well, are all optimal for offline environment. Nevertheless, no algorithms until now are optimal to cope with online preemptive scheduling problem, where tasksets are sporadic and multi-processor environments are enforced. On the other hand, despite of its optimality and dominance in theory, the usage of global dynamic priority algorithms are limited in practice. This is because the existence of frequent preemption and migration between tasks gives rise to excessive overheads in potential.

The following part of this subsection will provide brief summary of three classic global dynamic priority scheduling algorithms. They are respectively Proportionate Fairness Algorithm (PFair), Largest Local Remaining Execution First (LLREF), and Earliest Deadline until Zero Laxity (EDZL).

4.2.1. PFair

Baruah et al [5] introduced Proportionate Fairness Algorithm. The Pfair class of algorithms that allow full migration and fully dynamic priorities have been shown to be theoretically optimal – i.e., they achieve a schedulable utilization bound (below which all tasks meet their deadlines) that equals the total capacity of all processors.

4.2.2. LLREF

LLREF was firstly introduced by Cho et al [4]. LLREF was designed based on a novel abstraction for reasoning over task execution behavior on multiprocessors, called Time and Local Execution Time Domain Plane (T-L Plane).

4.2.3. EDZL

Michele's work [**?**] gave a more comprehensive analysis about the schedulability of EDZL.

4.3. Summary

5. Hybrid Approaches

6. Conclusions

References

[1] C. L. Liu, J. W. Layland, Scheduling algorithms for multiprogramming in a hard-real-time environment, Journal of the ACM (JACM) 20 (1973) 46–61.

[2] L. Sha, R. Rajkumar, J. P. Lehoczky, Priority inheritance protocols: An approach to real-time synchronization, Computers, IEEE Transactions on 39 (1990) 1175–1185.

[3] A. Khemka, R. Shyamasundar, An optimal multiprocessor real-time scheduling algorithm, Journal of parallel and distributed computing 43 (1997) 37–45.

[4] H. Cho, B. Ravindran, E. D. Jensen, An optimal real-time scheduling algorithm for multiprocessors, in: Real-Time Systems Symposium, 2006. RTSS'06. 27th IEEE International, IEEE, pp. 101–110.

[5] S. K. Baruah, N. K. Cohen, C. G. Plaxton, D. A. Varvel, Proportionate progress: A notion of fairness in resource allocation, Algorithmica 15 (1996) 600–625.