Multiprocessor Real-Time Task Scheduling with Local Assignment Algorithm

Doan Duy (1620420)
Tanaka's Laboratory
Graduate School of Advanced Science and Technology,
Japan Advanced Institute of Science and Technology
[Information Science]

August 9, 2019

Abstract

With the bloom of smart devices and automatic systems, real-time embedded systems have been rapidly spreading into different aspects of daily life, science, and industry. Numerous real-time applications leads to introduction of efficient and powerful processing architectures such as multiprocessor platforms. The real-time embedded systems are nowadays becoming diverse and complicated in the sense that the system has to not only handle various types of tasks, but also manage multiple processing units. Real-time task scheduling in such systems is therefore challenging to researchers in spite of the fact that a number of scheduling algorithms have been introduced. One of the critical issues of multiprocessor real-time task scheduling is exploitation of the increased system capacity to improve system performance.

It is seen that the problem of real-time task scheduling in uniprocessor systems have been solved effectively with several optimal algorithms such as Earliest Deadline First (EDF) which achieves schedulability of 100% with low time complexity. The problem of multiprocessor real-time task scheduling is however faced with a trade off between the simplicity and optimality. Simple algorithms with low time complexity are introduced toward practical applications since they are commonly believed easier for developing, testing and implementing. Whereas, optimal algorithms are introduced with an attempt to utilize the entire capacity of the system. Unfortunately, simple scheduling approaches achieve low schedulability and optimal ones often cause high time complexity.

This dissertation is conducted in challenge of that scheduling trade-off. That is, scheduling varied workloads on multiprocessors at schedulability of up to 100% with a relatively low time complexity. To this end, algorithm, named *Local Assignment Algorithm (LAA)*, is introduced. In order to deal with static workloads such as periodic tasks, LAA exploits the notion of proportionate scheduling in the time-interval scheduling scheme. Calculations of LAA allow to guarantee full task assignments which reduce unnecessary idle times of processors on intervals for better utilization of the entire system capacity. In addition, LAA is associated with a selective method of arranging tasks to processors, which is effective to lessen task preemptions and migrations.

LAA is theoretically proved as an optimal scheduling algorithm with the schedulability of up to 100%. Software simulations are conducted to simulate the behaviors of LAA in comparison with several existing algorithms including Pfair, RUN, and semi-partition reservation. Assertive criteria consist of scheduler invocation, task migration, task preemption, and time complexity. Simulation results show that effectively LAA algorithm invokes approximately 50% of scheduler invocations fewer than the other candidates while still maintaining relatively lower time complexity. In addition, compared with the outstanding optimal algorithm RUN, LAA is comparable in terms of task migration and preemption.

Dynamic workloads such as aperiodic tasks cause difficulties to scheduling algorithms due to their unknown factors such as entering time, execution time. One of the difficulties is increase

of runtime overhead. In fact, it is possible for aperiodic tasks to be scheduled in background of periodic ones with low scheduling cost in time. Nevertheless, this approach makes the aperiodic responsiveness increase. In this dissertation, combination of LAA and concept of servers, called $LAA+$, is introduced to deal with the hybrid task sets of periodic and aperiodic tasks. The aperiodic tasks are therefore scheduled on-line together with periodic ones through service of servers. LAA+ is also extended with introduction of secondary scheduling events for better use of servers. The proposed scheduling approach then effectively improves the aperiodic responsiveness.

Software simulations are conducted to evaluate LAA+ in the context of mixture system of periodic and aperiodic tasks. The targeted criterion is switched to improving the aperiodic responsiveness while guaranteeing for periodic tasks to meet their deadlines. In evaluation, LAA+ continues maintaining the optimality of the original LAA. Simulation results show that LAA+ efficiently enhances the responsiveness of aperiodic tasks by approximately 30% compared with LAA and by about 10% compare with other scheduling candidates. Moreover, LAA+ still achieves lower runtime overheads and less number of scheduler invocations in comparison with the other candidates. Overall, achieving remarkably fewer scheduler invocations and relatively lower time complexity indicates that LAA and LAA+ have better exploitation of system capacity in a sense that the system time is preferred to spend for executing application tasks rather than the scheduling algorithm.

Finally, the Local Assignment Algorithm is implemented on a practical embedded system in order to confirm the applicability of the algorithm. The Zedboard FPGA Evaluation KIT is selected as the implementation environment. The system includes a built-in SoC Zynq7000 which supports a dual-core ARM Cortex-A9 processor. A multiprocessor real-time operating system (M-RTOS) in which LAA is utilized for scheduling application tasks is developed based on the ITRON RTOS kernel. It is found that the proposed algorithm is applicable to actual real-time embedded systems.

Key Words: **real-time task scheduling, multiprocessor, time complexity, aperiodic responsiveness, dynamic workload**