Latency, energy, and schedulability of real-time embedded systems

Liu, D.; Liu D.

Chapter 7

Summary and Future Work

> Research is to see what everybody else has seen, and to think what nobody else has thought.
>
> Albert Szent-Gyorgyi

7.1 Summary and Conclusions

As we quoted in the epigraph of Chapter 1, "Almost all computer systems of the future will utilize real-time scientific principles and technology.", real-time systems and the systems that desire to apply real-time discipline are becoming ubiquitous with the advent of Internet of Things (IoTs) and Cyber-Physical Systems (CPS). The increasing complexity of real-time software and the emerging new hardware inspire us to revisit the “old-wise” in the embedded system community and the real-time community and to propose novel solutions dealing with the drastic changes in real-time systems.

The *HRT* scheduling framework proposed in [BS13] establishes a bridge between the data-flow models and the real-time theories, enabling us to directly apply real-time theories on the well-known data-flow models, e.g., SDF and CSDF. The *HRT* scheduling framework effectively converts actors in a CSDF graph into a periodic task set with implicit deadline and thus the majority of the theories developed in the real-time community can be applied to provide fast admission control and real-time guarantee for applications. However, the conversion done by the *HRT* scheduling framework comes at the cost of hurting the application latency which is one of the primary performance metrics for streaming applications. The proposers of *HRT* scheduling framework are aware of this issue [BS12] and suggest to select a smaller relative deadline for each

task to reduce the latency. In the real-time theories, such scaling down of deadlines of tasks negatively affects the schedulability of a multiprocessor real-time systems, and thus has to use more processors to compensate this negative effect, i.e., a larger number of processors are required to schedule the task set. The authors in [BS12] propose a simple way to uniformly select deadlines for all tasks but this approach is really ineffective in terms of resource minimization while meeting latency requirements. Therefore, to deal with this problem, in Chapter 3, we have proposed a new method to optimize the resource usage in the context of the *HRT* scheduling of CSDFs, where we formalize the resource minimization problem into an integer convex programming problem. By means of a off-the-shelf convex programming solver, we can obtain an optimal deadline selection for each task while minimizing the resource requirement and meanwhile ensuring the latency guarantee for CSDF-modeled streaming applications. The experimental results demonstrate the effectiveness of our approach over the existing approach in [BS12].

Due to the growing power consumption of increasingly complex applications, energy/power consumption is deemed as one of the major concerns when designing embedded systems. The single-ISA heterogeneous multicore systems are proposed to alleviate this energy pain. Nowadays such systems are prevalent in commercial electronic devices, such as mobile phones, TV boxs, etc. These systems provide designers a new opportunity to achieve energy efficiency and high performance and on the other hand they also require to find a new methodology to efficiently and effectively utilize the underlying hardware in an energy-efficient manner. Therefore, in Chapter 4, we have proposed a polynomial time algorithm to energy-efficiently map real-time streaming applications with latency and throughput constraints to cluster single-ISA heterogeneous multicore systems. Compared with existing approaches, the experimental results show that our proposed algorithm outperforms the existing approaches by finding a more energy efficient mapping. Our algorithm can save up to 34% energy on the cluster heterogeneous single-ISA multicore systems.

In Chapter 5, we have continued to study the problem of energy-efficient mapping on heterogeneous multicore systems. In this work, we have investigated the application of the C=D task-splitting [BDWZ12] on heterogeneous systems. We have analyzed and extended the C=D task-splitting for heterogeneous multicore systems. With our analysis and extension, we have proposed the ASHM algorithm to allocate and split real-time tasks on a heterogeneous multicore system. In contrast to fully partitioned allocation approaches, our proposed ASHM algorithm can effectively utilize energy-efficient cores to achieve more energy saving. The experimental results show the effectiveness of our proposed ASHM in terms of energy saving, where the maximum energy saving by ASHM compared to related approaches is up to 60%.

The trend towards integrating applications with different criticality levels on a

single HW/SW platform is emerging in safety-critical real-time systems. In order to satisfy the rigorous requirements of certification authorities and at the same time to better utilize the underlying HW/SW platform, a classical Mixed-Criticality (MC) model is proposed in [Ves07]. Although this classical MC model is able to capture the core features of MC systems, it receives criticism from system designers due to its pessimistic behavior of completely discarding all low critical application tasks when any high critical application task overruns. Imprecise MC (IMC) model is proposed in [BB13] to resolve the criticism, but its schedulability analysis under EDF-VD still was not studied. Therefore, in Chapter 6, we have studied the schedulability of the IMC model under EDF-VD and proposed a sufficient test. Based on the proposed sufficient test, we have derived a speedup factor function with respect to the utilization variation ratio α of all *high-criticality* tasks and the utilization variation ratio λ of all low-criticality tasks. This speedup factor function provides a good insight to observe the impact of α and λ on the speedup factor and enables us to quantify the optimality of EDF-VD for the IMC model in terms of speedup factor. Our experimental results show that our proposed sufficient test outperforms the existing AMC approach in terms of acceptance ratio. Moreover, the extensive experiments also confirm the observations we obtained for the speedup factor.

7.2 Future work

Although this dissertation has made several contributions to the real-time embedded system field, there remains interesting topics which can be researched based on our contributions. This section discusses some issues or challenges which deserve further investigation in the future.

7.2.1 The real convergence of data-flow models and real-time theories

An increasingly hot topic in the real-time community is the scheduling problem of parallel real-time directed acyclic graphs (DAG). We can clearly see the conceptual similarity between the DAG model used in the real-time community and the data-flow models used in the embedded system community. Considering the analogy of these models, it is worth to investigate how the DAG theories can be directly applied to data-flow models. Such directed application might be able to provide a better performance and real-time analysis framework and the conversion overhead occurred in the *HRT* scheduling framework, e.g., the increased latency (see in the problem studied in Chapter 3), might also be eliminated.

7.2.2 The multi-objective mapping of heterogeneous multicore systems

Our algorithms presented in Chapter 4 and 5 demonstrate the effectiveness in terms of energy efficiency. However, there are more objectives worth to be investigated in the complex HW/SW heterogeneous multicore, i.e., the thermal objective, the reliability objective and the security objective. These objectives interact with each other, thereby leaving us a large design space to exploit and requiring us to find a good trade-off between multiple objectives. Therefore, it is a very interesting and challenging problem to design an efficient and effective algorithm to map real-time applications onto a heterogeneous system with multiple objectives considered.

7.2.3 Practical and flexible MC model

Even though the IMC model has dealt successfully with some of the criticism from system designers, one assumption in the IMC model is still somehow pessimistic and in some cases impractical, i.e., if any *high*-criticality task overruns, all the other *high*-criticality tasks are assumed to overrun their smaller WCETs and thus be scheduled with their large WCETs (pessimistic ones). This assumption makes the system overreact to the overrun of a single *high*-criticality task and consequently it leads to an unnecessary degradation (i.e., reduced execution time) of *low*-criticality tasks. Therefore, further research can be conducted in the context of defining an MC model which can effectively reduce the pessimism of the current model and provide a more flexible execution semantics. To meet this goal, some existing theories, like control theories, might help.