Extending the system scenario based design: for inter-application resource negotiations, at runtime

Zubair Wadood Bhatti, Davy Preuveneers, Roel Wuyts, Yolande Berbers.
DistriNet, Katholieke Universiteit Leuven, Belgium.
{zubairwadood.bhatti, davy.preuveneers, roel.wuyts, yolande.berbers}@cs.kuleuven.be

System-scenario-based design [1] has become quite popular in the research community as an approach to mapping dynamic applications to MPSoC platforms. Although the approach is efficient for dealing with the dynamism exhibited by certain applications it is still not seen commercial solutions. We identified two hurdles in the way of commercialization of this approach, (1) the excessive time taken due to multi-objective optimizations at the design time and (2) the state space explosion that occurs while attempting design time exploration for all combinations of target applications sharing a platform such as in [2]. We propose an extension to the approach in order to deal with these two problems.

In the context of nomadic embedded systems energy is usually the only resource of importance to the end user, whereas other resources like memory, computation and on-chip communication resources are unimportant as long as the quality of service requirements for all the running applications are fulfilled. But we still optimize try to optimize all resources because other applications sharing the platform might suffer.

Typically in system scenario based design, a runtime situation is defined by the input data from the user or the execution environment. Runtime situations with similar resource requirements are grouped together into system scenarios. Scheduling and resource allocation decisions are made at design time for a scenario using worst case estimates for the scenario, extracted through static analysis. At runtime the scenario needs to be identified so that the scheduling and resource allocation decisions already made at design time can be used. In our extended approach, at design time, we annotate the scenarios with (1) different possible schedules that allow the quality of service requirements of the application to be met and (2) the energy cost of these schedules. Depending on the different scenarios of all the applications sharing the platform at runtime, suitable configurations are selected such that the total energy consumed by the system is minimized. For example if you have two active applications, the runtime environment will detect the scenarios in both the applications individually and then load the minimum cost schedules that are able to fit into the resource constraints of the platform. All of these schedules that are considered at runtime meet the deadlines imposed by Quality of service requirements and their costs are pre-calculated so that the runtime overhead stays small.

As we were confronted with multi-objective optimizations at design time, we delayed part of the optimization process to runtime when we have information about other applications sharing the platform and are now able to reduce it to a single-objective optimization. This is how apart from enabling the execution of multiple applications in parallel we can also reduce the time spent the offline analysis.

References:

1. S. V. Gheorghita et al. “System-Scenario-Based Design of Dynamic Embedded Systems”, ACM Transactions on Design Automation of Electronic Systems, 2009.
2. P. V. Stralen. “Scenario Based Design Space Exploration”, Master’s thesis, University of Amsterdam, 2009.