Abstract ID 1010

DISTRIBUTION OF WORKLOAD IN IMA SYSTEMS BY SOLVING A MODIFIED MULTIPLE KNAPSACK PROBLEM

Vasily Balashov, Ekaterina Antipina

Lomonosov Moscow State University, Russian Federation
hbd@cs.msu.su, antipantipovych@gmail.com

Keywords: real-time systems,integrated modular avionics,workload distribution,branch and bound algorithm,scheduling

Abstract: Integrated Modular Avionics (IMA) systems are distributed real-time systems consisting of computing modules connected by a communications network. A typical IMA system utilizes several types of unified modules. Each module runs a set of periodic computational tasks on its central processor (CPU). The tasks are grouped into partitions according to their criticality and data coupling. Each partition is bound to a specific module. Data are transferred between partitions as messages. Frequency of message transfer equals to frequency of the message's sender task. Workload for an IMA system is the set of partitions to be executed on it. CPU load for a task is the multiple of its worst-case execution time (WCET) and frequency. Network load for a message is the multiple of its transfer time and frequency. CPU load for a partition is the sum of CPU loads for its tasks. As different modules can use different CPUs, a task generally has a specific WCET for every module, so CPU load by a partition depends on the module. If two partitions are bound to the same module, messages between them do not create network load. For each module, there is a specific upper limit on CPU load. Binding of some partitions can be restricted to a subset of modules, e.g. those with graphical co-processors. To ensure schedulability and scalability of an IMA system, the workload distribution problem is stated as follows: assign partitions to modules so that the total network load is minimized, and the constraints on CPU loads and partitions binding are met. In this paper, we represent this problem as a modified version of the Multiple Knapsack Problem (MKP). Modules correspond to knapsacks, partitions to items, CPU loads by partitions to item volumes, CPU load limits to knapsack capacities. Item's volume depends on the choice of knapsack. Difference from the original MKP is that profit is assigned to pairs of items (reflecting the network load by a pair of partitions); profit is considered zero if the two items are packed in different knapsacks. The goal is to maximize the total profit. We present a branch and bound algorithm for solving the modified MKP, including a new scheme for upper bounds calculation and heuristics for search space reduction. Algorithm scalability is analyzed on data from real IMA systems. The algorithm is implemented in a scheduling tool accepted for operation by one of the leading Russian aircraft design companies.