Chapter 5

Analysis of Multiprocessor Systems

The challenge taken in this chapter is to analyse an application running on a multiprocessor system with acceptable accuracy, without the need to explicitly store and compute the memory consuming distributions of the residual execution times of each task in the states of the underlying stochastic process. Also, we would like to avoid the calculation of the computation-intensive convolutions.

We address this problem by using an approximation approach for the task execution time probability distribution functions. Approximating the generalised ETPDFs with weighted sums of convoluted exponential functions leads to approximating the underlying generalised semi-Markov process with a continuous time Markov chain. By doing so, we avoid both the computation of convolutions and the storage of the $z_i$ functions. However, as opposed to the method presented in the previous chapter, which produces exact values for the expected deadline miss ratios, the alternative approach generates approximations of the real ratios.

The approximation of the generalised task execution time probability distributions by weighted sums of convoluted exponential distributions leads to a large continuous time Markov chain. Such a Markov chain is much larger than the stochastic process underlying the system with the real, non-approximated execution times, but, as the state holding times probability
distributions are exponential, there is no need to explicitly store their distributions, leading to a much more efficient use of the analysis memory. Moreover, by construction, the Markov chain exhibits regularities in its structure. These regularities are exploited during the analysis such that the infinitesimal generator of the chain is constructed on-the-fly, saving additional amounts of memory. In addition, the solution of the continuous time Markov chain does not imply any computation of convolutions. As a result, multiprocessor applications of realistic size may be analysed with sufficient accuracy. Moreover, by controlling the precision of the approximation of the ETPDFs, the designer may trade analysis resources for accuracy.

5.1 Problem Formulation

The multiprocessor system analysis problem that we solve in this chapter is formulated as follows.

5.1.1 Input

The input of the problem consists of:

- The set of task graphs \( \Gamma \),
- The set of processors \( P \),
- The mapping \( \text{Map} \),
- The set of task periods \( \Pi_T \) and the set of task graph periods \( \Pi_\Gamma \),
- The set of task deadlines \( \Delta_T \) and the set of task graph deadlines \( \Delta_\Gamma \),
- The set of execution time probability density functions \( ET \),
- The late task policy is the discarding policy,
- The set \( \text{Bounds} = \{ b_i \in \mathbb{N}\setminus\{0\} : 1 \leq i \leq g \} \), where \( b_i \) is the maximum numbers of simultaneously active instantiations of task graph \( \Gamma_i \), and
- The scheduling policies on the processing elements and buses.