Karlsruhe Institute of Technology

Institute for Program Structures and Data Organization
Chair for Software Design and Quality
DFG Research Group ("Emmy Noether" Programme)

Software Design and Quality

Descartes Research Group
http://descartes.ipd.kit.edu

Descartes research

Datum: 2013-01-18

Diploma Thesis

Parallel Simulation of Queueing Petri Net Models

QPME

Motivation
QPME (Queueing Petri net Modeling Environment) offers a discrete-event simulator, called SimQPN, for the quantitative analysis of queueing petri nets (QPN). This simulator is currently implemented sequentially, i.e. it can only fully use single-core processors. As multi-core processors are now standard in most computers a parallel implementation provides the potential to speed up simulation runs significantly in many usage scenarios, e.g. when using QPME for online performance prediction.

Goals
In this thesis, you will implement and evaluate different techniques to parallelize the simulation of QPNs in order to optimally utilize modern multi-core processors. The following areas of the simulation promise a potential speed up through parallelization:

- If the method of independent replication is used for output data analysis, the replications are currently run sequentially. A major speed-up can be reached by executing the replications in parallel.
- Certain helper functions, such as the random number generation or the checks of the stopping criteria, could be moved from the main simulation loop into auxiliary threads. The potential overhead due to required synchronization protocols needs to be considered.
- In the case of independent replications parallel execution can be leveraged to determine the initial transient period of a simulation. For details see Bause and Eickhoff [1].
- Structural knowledge about QPNs might be leveraged to simulate certain parts of the model in parallel.

You will start by identifying alternative techniques for parallelizing the simulation in the areas listed above. This will be based on existing literature and an in-depth analysis of the SimQPN program. You will then implement these techniques in the current SimQPN simulator. This will include a refactoring of its current architecture in order to support the distribution of the simulation over multiple threads. Finally, you will evaluate the parallel version of SimQPN in order to assess the speed-up offered by the implemented parallelization techniques.

[1] Bause, F.; Eickhoff, M.: Truncation Point Estimation using Multiple Replications in Parallel 2003 Winter Simulation Conference December 7-10, New Orleans, Louisiana, USA, 2003.

Duration
6 Months

Contact
M.Sc. Simon Spinner, Dr.-Ing. Samuel Kounev
simon.spinner@kit.edu, kounev@kit.edu
http://descartes.ipd.kit.edu

BUILT ON eclipse