Master Thesis

“Language Independent Modelling of Parallelism”

Background

As multicore architectures are becoming prevalent in many of today's systems, less error-prone and more convenient ways of expressing parallelism are clearly gaining in importance. Throughout the history of software technologies a multitude of various techniques to model parallel execution have been developed. Varied as they are, most parallelisation mechanisms show some recognisable common features. While being able to make an intent to parallelise clear is crucial, the choice of a correct and unambiguous form of expression is of a great value as well. The analysis of various forms of expressing parallelism in contexts of varying degrees of abstraction and extraction of the most prominent commonalities would be most advantageous.

Tasks of the Thesis

The objective of the thesis is to analyse various forms of parallelisation found in modern software systems and formulate the commonly applicable generic principles of parallel execution based on the conducted analysis. Particularly significant are the areas of dependency on specific hardware, the expression of temporal constraints, and providing for sufficient predictability of the resulting behaviour.

The thesis is expected to meet at least the following requirements

- Analysis of currently available literature dealing with various parallelisation concepts aiming at finding and exploring techniques better suited for higher predictability and efficiency.
- Identification of the basic principles common to different parallelisation techniques.
- Identification of the most important differences along with the explanation as to the causes for their existence.
- Specification of the factors commonly responsible for incorrect behaviour (i.e. deadlocks, various kinds of race conditions) and the ways of their avoidance in the respective modelling technique.
- Appraisal of the interactions between software and hardware affecting the correctness, predictability and efficiency in different contexts including but not limited to real-time systems.
- Identification of the influences significantly affecting predictability of deadline violations.
- Assessment of the effects of using different programming paradigms on the resulting outcome based on a choice of at least two indicative combinations of a programming language and a parallelisation technique.
- Provision of multiple modelling and code examples serving as a proof-of-concept.
- An intermediate presentation is expected to be held 3.5 months after the starting date and the final presentation no later than 1.5 months after the submission of the written report.

In general, higher levels of abstraction should be sought and the aspects limiting the attainable degree of concept generality should be clearly denoted.

The following work items are optional tasks for this thesis:

- Evaluation of the software analysis principles pertaining to parallelisation as recognised by the Bauhaus software analysis suite.
- Quantitative analysis of beneficial effects provided by higher abstraction.

Supervising Professor: Prof. Dr. Erhard Plödereder
Consultant: Mikhail Prokharau
Starting Date: now