A Parallel Synchronous Language for Computational Control Systems

Master 2 Thesis Proposal
PARKAS group, INRIA and DI, ENS Paris

March–August 2016

The internship will explore the design of a programming language suitable for the parallelization of hard real-time, safety-critical applications on multiprocessors. The application domain is very large and growing rapidly: starting with fly-by-wire controllers in the early 80s, it has grown to driving assistance and automated vehicles, health monitoring of complex machinery or industrial production lines, robotic equipment for medical systems, etc. Synchronous programming languages such as LUSTRE or SCADE are popular vehicles for the design and implementation of such reactive control applications. While increasingly computational components make their way into control systems, and despite the ubiquity of multicore processors, the state of the art in synchronous programming remains desperately rooted in a sequential world.

Synchronous languages enable the formal verification and the generation of efficient embedded code, *starting from a single source*. Esterel Technologies' SCADE Suite has become the dominant platform for the implementation of critical systems certified at the highest safety standards, such as SIL 4 in IEC 61508 (railways) or DAL A in DO-178B (avionics). This success has a lot to do to its expressive yet well-defined semantics, and to its traceable compilation in a certified development flow. Unfortunately, synchronous design methods and tools have not yet been successfully applied to multiprocessor platforms.

An existing compiler will be extended to exploit parallelism in the dataflow equations of the synchronous program, and to distribute the program over computation, communication and storage resources of an embedded multiprocessor.

One key difficulty has to do with the real-time requirements of safety-critical applications. Interferences on shared buses and caches of conventional multicores make it very hard to establish a practically useful worst-case execution time for a given task, or for a given reaction of a synchronous program. Leveraging previous studies of multi- and manycore processors, we will extend the synchronous language compiler to implement protocols

for parallel execution while preserving the spatial and temporal isolation of real-time components.

Prerequisites

- Solid background in the design and implementation of programming languages. (*required*)
- Solid programming experience in OCaml. (*preferred*)
- Prior experience with synchronous programming or shared-memory parallel programming. (*optional*)

Variations on the topic are possible. For example, while no prior experience with embedded multiprocessors is required, but the subject can be adapted to put more emphasis on the computer architecture. Among possible variations on this subject, one may also consider carrying this research into a mechanically proven synchronous language compiler developed with the Coq proof assistant.

Administrative information The internship will take place in the PARKAS team of INRIA and DI, ENS Paris:
`http://www.di.ens.fr/ParkasTeam.html`

Advisors. Albert Cohen and Marc Pouzet.

The intern will receive a stipend or salary according to her or his administrative status.

The internship is partially funded by the ECSEL *EMC*^{2} and ITEA *ASSUME* European projects.

PhD thesis The internship opens on a 3 years PhD thesis at ENS Paris. Frequent interactions with industry leaders such as Esterel Technologies and Kalray are expected, with concrete use cases and applications to ongoing research and development at SAFRAN and Airbus.