

# Static Selective Instrumentation for Parallel Programs Verification

## Global informations

**Domain :** High Performance Computing

**Laboratory-Institute :** Inria Bordeaux Sud-Ouest, Talence

**Advisor :** Emmanuelle Saillard / emmanuelle.saillard@inria.fr

## Key words

HPC, Compilation, Static analysis, Verification, MPI, Automata

## Abstract

High Performance Computing (HPC) plays an important role in many fields like health, materials science, security or environment. The current supercomputer hardware trends lead to more complex HPC applications (heterogeneity in hardware and combinations of parallel programming models) that pose programmability challenges. As indicated by a recent US DOE report, progress to Exascale stresses the requirement for convenient and scalable debugging methods to help developers.

The PARAllel COntrol flow Anomaly CHecker[1,2,3] (PARCOACH) aims at helping developers in their debugging phase. It combines static and dynamic analyses to detect misuse of collectives (i.e., collective operations mismatch in MPI) in parallel applications. At compile-time, an interprocedural static analysis studies the control and data flow of a program to find potential deadlocks. During this step, a program is tagged as statically correct (no possible deadlock) or not verifiable (a deadlock may occur at runtime). For statically not verifiable programs, all collectives and exit statements are instrumented with Check Collective (CC) functions, starting from the first collectives that may deadlock in the program. The first "safe" collectives are not instrumented. At execution-time, the CC functions verify which collectives will be called at different steps of execution. In case of an actual deadlock situation, the program is stopped before the deadlock occurs.

The goal of this internship is to improve the selective instrumentation of PARCOACH. The student will work on developing an analysis that builds an automata of the program to compute the language of remaining collectives at different point of the program. The analysis will combine the static information in PARCOACH with the automata to reduce the number of checks at runtime.

The internship will first focus on MPI applications and will then be extended to other parallel programming models. The work will be implemented as an LLVM Pass [4] and fully integrated to the PARCOACH tool.

During the internship, the student should develop scientific skills on compilation techniques as well as scientific writing and presentation.

## Additional activities

- Attend group meetings and present progress of work
- Assist in writing research articles about the work

## References

- [1] Emmanuelle Saillard, Patrick Carribault, Denis Barthou. *PARCOACH : Combining static and dynamic validation of MPI collective communications*. International Journal of High Performance Computing Applications, SAGE Publications, 2014.

- [2] Emmanuelle Saillard, Patrick Carribault, Denis Barthou. *Static Validation of Barriers and Worksharing Constructs in OpenMP Applications*. IWOMP, Sep 2014, Salvador, Brazil. pp.73 - 86, 2014.
- [3] Pierre Huchant, Emmanuelle Saillard, Denis Barthou, Hugo Brunie and Patrick Carribault, *PARCOACH Extension for a Full-Interprocedural Collectives Verification* in Second International Workshop on Software Correctness for HPC Applications, Dallas, Texas, USA, November 12, 2018.
- [4] <http://llvm.org/docs/ProgrammersManual.html>