Computer-science Post-doc position at LIP, Lyon, France

# Scheduling, parallelism and Polyhedral Model for sparse matrix operations

**Supervisors:** Christophe Alias and Matthieu Moy

**Laboratory:** Laboratoire de l'Informatique du Parallélisme (LIP)
École Normale Supérieure de Lyon

## Context

The polyhedral model [1] allows representing a set of operations on arrays together with their dependencies. This representation can be used to schedule operations (for example, change the execution order of a set of nested `for` loops), or to extract parallelism from a sequential program. It has successfully be applied to many matrix-based compute-kernels for aggressive compiler optimization and synthesis of efficient hardware circuits.

A strong limitation, however, is that the model is limited to *regular* applications: it supports `for` loops and array accesses, but loop bounds and array indices must be expressible as affine functions of variables and parameters of the compute kernel.

The newly created CASH team works on novel methods to analyze a program and targets in particular hardware generation. One of the research axis is to extend the polyhedral model to irregular applications, e.g. `while` loops and data-structures other than arrays.

## Objectives of the post-doc

The goal of the post-doc is to study the applicability of polyhedral methods to irregular applications. As a starting point, we will focus on sparse matrices (large matrices containing mostly 0 values, using data-structures that represent only non-zero values).

How does the polyhedral model extend to such applications? In particular how do the notions of iteration domain and affine schedule translate? Which static/dynamic analysis are required? How do the dynamic analysis translate to source code generation?

This work will be the first building block towards the compilation of hardware for FPGA accelerators. Even though hardware generation itself is out of the scope of the post-doc, the source code generated is expected to fit the input of a C-to-FPGA compiler and to be tested on a FPGA.

## Indicative outline

Here is a more precise list of tasks that could be carried out.

- If needed: study of existing polyhedral techniques for automatic parallelization [1] and related work on parallelization techniques for sparse matrices (for example [2]).

- Study a very simple algorithm on sparse matrices, e.g. sum of values of one matrix, sum of two matrices. In these examples, the dependencies between computations are very simple hence scheduling and tiling can be done with simple techniques even with sparse matrices.
- Study more complex algorithms like matrix multiplication.

## Expected skills

The candidate should have good background in compilation (knowledge of polyhedral methods would obviously be appreciated), and should be familiar with parallel algorithms.

## Applications

Please send your applications by email to both Christophe.Alias@ens-lyon.fr and Matthieu.Moy@univ-lyon1.fr.

Include a CV, a motivation letter, and any document that can support your application.

## Supervisors

This post-doc will be supervised by Christophe Alias (Inria Researcher, ENS-Lyon) and Matthieu Moy (Assistant professor, HDR, UCBL).

Christophe Alias (`http://perso.ens-lyon.fr/christophe.alias/`) has been working on high-level circuit synthesis (HLS) using polyhedral methods for more than 8 years. He co-supervised two Ph.D. He wrote a process-network compiler that he transfered to the Xtremelogic startup.

Matthieu Moy (`https://matthieu-moy.fr`)'s main research area is hardware simulation (using SystemC), on which he has been working for more than 10 years in partnership with STMicroelectronics. More recently, he started working on worst-case execution time for software and worst-case traversal time for networks-on-chip, and compilation for critical systems. He joined the LIP laboratory in 2017 and started working on HLS and polyhedral methods.

## References


[1] Paul Feautrier and Christian Lengauer. Polyhedron model. In *Encyclopedia of Parallel Computing*, pages 1581–1592. 2011.

[2] Anand Venkat, Manu Shantharam, Mary Hall, and Michelle Mills Strout. Non-affine extensions to polyhedral code generation. In *Proceedings of Annual IEEE/ACM International Symposium on Code Generation and Optimization*, page 185. ACM, 2014.