# Composability and Data Partitioning in High Performance Computing

Master Internship 2025

**Keywords:** High Performance Computing, Parallel languages, Data Partitioning, Composability.

**Advisors:** Christian Perez

**Laboratory:** LIP, ENS Lyon, UMR CNRS-INRIA 5668, 46 allée d'Italie, 69364 Lyon Cedex 07

**Email:** `christian.perez@inria.fr`

## Context

Numerical simulation is a key technology for many application domains. Thanks to the democratization of high performance computers (HPC), complex physics and more generally complex systems can now be simulated routinely. Numerical simulation is considered as the third pillar of sciences (with experiment and theory).

While the machines are very powerfull, their programming models face several challenges to easily and efficiently exploit the available computing resources. A particular challenge is to ease code composability to improve separation of concerns and code reuse. Code composability appears also more and more required to handle the complexity of hardware and software.

Though composable code is an old dream [4] that is very common in sequential and distributed computing [7], it is an open issue for parallel computing. Several previous works such as CCA [1], L2C [3], Comet [6] have shown that it is achievable on specific parallel patterns. In particular Comet transforms a data-partitionned data flow into a task-based code that is able to submit StarPU [2], OpenMP or MPC [5] tasks.

## Internship Objectives

This internship is part of the PEPR NumPEx Exa-soft project that in particular aims at improving parallel code composability.

The objective of this internship is to study how to improve partitionned data management to easily be able to add application specific efficient data partionning support. Currently, on one hand, advanced task runtime engines like StarPu need to understand the structure of partitionned data to efficiently move it from CPU to GPU/accelerator memory. On the other hand, higher level programming environnements like Comet [6] needs also to understand it to be able to generate adequate StarPU code.

After an analysis of the situation, the goal is to design an evaluate a proof of concept on a synthetic benchmarck based on Comet and StarPU.

The internship will require good parallel algorithmic skills (data partitionning) as well a taste for modeling (meta-models) and programming (Python/C++/C and task-based/StartPU programming).

### Comments

The internship will be located at the LIP, ENS Lyon. This internship could lead to a PhD.

## References

[1] B. A. Allan and al. A Component Architecture for High-Performance Scientific Computing. *International Journal of High Performance Computing Applications*, 2006.

[2] Cédric Augonnet, Samuel Thibault, Raymond Namyst, and Pierre-André Wacrenier. Starpu: A unified platform for task scheduling on heterogeneous multicore architectures. In Henk Sips, Dick Epema, and Hai-Xiang Lin, editors, *Euro-Par 2009 Parallel Processing*, pages 863–874, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[3] Julien Bigot, Zhengxiong Hou, Christian Pérez, and Vincent Pichon. A low level component model easing performance portability of HPC applications. *Computing*, 2013.

[4] M. D. McIlroy. Mass-produced Software Components. *Proc. NATO Conf. on Software Engineering, Garmisch, Germany*, 1968.

[5] Marc Pérache, Hervé Jourdren, and Raymond Namyst. Mpc: A unified parallel runtime for clusters of numa machines. In Emilio Luque, Tomàs Margalef, and Domingo Benítez, editors, *Euro-Par 2008 – Parallel Processing*, pages 78–88, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[6] Jérôme Richard. *Conception d'un modèle de composants logiciels avec ordonnancement de tâches pour les architectures parallèles multi-coeurs, application au code Gysela*. PhD Thesis, Université de Lyon, December 2017.

[7] Clemens Szyperski. *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley Longman Publishing Co., Inc., 2002.