**Assignment**

This PhD will study the problem of scheduling scientific applications under memory constraints. The goal is to extend the results of the team in this area, which are for now limited to shared-memory platforms. The objective is to derive algorithmic strategies on a larger scale, and especially for distributed memory platforms.

**Main activities**

Scientific applications are usually modeled as Directed Acyclic task Graphs (DAGs of tasks) and they often involve data of large sizes [1]. However, the computing platforms on which these applications are to be processed have limited memory capabilities, which are distributed among the available computing nodes. Moving data among these nodes is costly, and reading/writing data from/to disks (usually of much less limited size) is even more time consuming.

There have been a number of studies focusing on data locality to optimize the mapping and scheduling of such applications. In particular, in the context of the past ANR project SOLHAR, the ROMA team has studied the problem of scheduling such task graphs with limited memory, and has proposed algorithms to bound the memory requirement of an execution [2,3,4]. However, most of these studies are limited to the case of a single memory, shared among the computing cores. The objective of this thesis is to extend these results to larger-scale platforms, composed of several nodes equipped with their own memory. There are several challenges that have to be taken into account when designing scheduling strategies for such platforms:

* One could think of the different distributed memories as chunks of a large, shared memory, as one node may access the memory of another node. However, this type of access is much slower than accessing a local memory bank, hence transfer times have to be taken into account to avoid moving time back and forth.

* When going large-scale, we cannot rely anymore on a centralised scheduler: most scheduling decisions have to be made locally and to avoid the frequent synchronisation of all nodes.

* Task graphs to be processed on such large-scale platforms are likely to include a large number of tasks. However, scheduling decisions have to be made in a very short time. Thus, scheduling
algorithms cannot rely on complex graph algorithms, as proposed in current solutions. Nevertheless, part of the scheduling process can be performed beforehand, by pre-computing important information on the graph, or as a background task, while other cores are busy computing tasks.

A promising concept to tackle all three challenges is the use of hierarchical scheduling: tasks of the graph may be clustered into coarser-grain groups of tasks. Such coarse-grain groups may be scheduled onto the distributed nodes, and the schedule may be locally refined when opening the coarse-grain groups to reveal the original tasks. The thesis will consist in exploring such a hierarchical approach as well as other possible approaches, such as adapting classical work-stealing scheduling algorithms for memory-bounded scenarios.


Skills
The candidate will be required to have a solid background in algorithms, be familiar with scheduling problems and a good level in programming with usual languages (C, C++, python).

Benefits package
- Subsidised catering service
- Partially-reimbursed public transport
- Social security
- Paid leave
- Flexible working hours
- Sports facilities

Remuneration

Monthly salary after taxes: around 1596,05€ for 1st and 2nd year. 1678,99€ for 3rd year. (medical insurance included).