

Opportunist Co-Scheduling of Biological Applications

Research team name: GRAAL - CRI Grenoble - Rhône Alpes, Lyon Site

Intern tutors: Frédéric Desprez (desprez@ens-lyon.fr) and Frédéric Suter(fsuter@cc.in2p3.fr)

Intern level: Master's Thesis

Internship duration: 4 to 6 months

Possibility of a follow-up Ph.D: Yes

Internship description:

Over the last decade, computing grids have become a powerful instrument to solve many scientific problems from various fields such as biology or physics. These grids are large scale heterogeneous interconnections of commodity clusters. At the cluster level, computing resources, typically processors, are managed by batch schedulers. They are in charge of the distribution of jobs on the processors in a way that minimizes each job service time and maximizes the resource usage.

On such computing infrastructures two categories of jobs are executed: sequential and parallel, i.e., that use one or several processors. In this proposal we focus on a particular type of applications in each category. We consider parameter sweep applications for the sequential jobs and scientific workflows for the parallel jobs. A parameter sweep application consists in applying the same program with a large range of input parameters. this usually leads to a large set of totally independent jobs. On the other hand a scientific workflow is an application composed of several interdependent tasks. It is usually described by a graph expressing the precedence constraints and communications between computations.

When a batch scheduler deals with parallel jobs, it may create a schedule with idle times on some processors as shown on Figure 1 (a). Some recent work proposed an interesting solution to fill these gaps. When a slot cannot be used by a parallel job, some sequential jobs are launched (in grey in Figure 1 (b)). To prevent subsequent parallel jobs to be delayed by these sequential jobs, they are launched in a *best-effort* mode. This means that if a regular job requires a resource used by a best-effort job, the latter is killed.

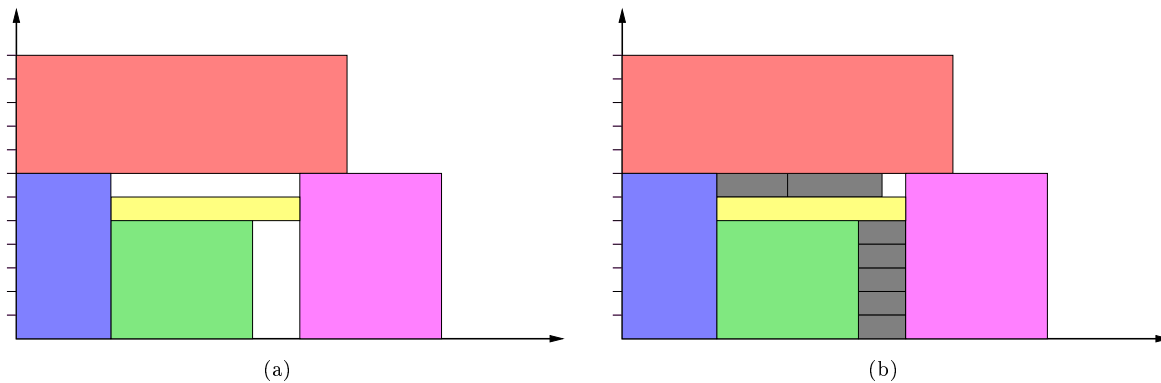


Figure 1: Example of a schedule of parallel jobs with idle times (a) and with idle times filled by sequential jobs (b).

When a scientific workflow with inter-task communications has to be executed on a cluster, it is easier to make a single resource reservation. It is common to use a scheduling algorithm to determine a mapping for each task in the workflow. This leads to a Gantt chart of a certain length (time) and width (number of required processors). These two dimensions give the parameters of the resource reservation.

This internship proposal comes from a simple observation. From the point of view of the batch scheduler, a scheduled workflow is a black box as in Figure 1 (a). But with a closer look, we can see on Figure 2 (a) that

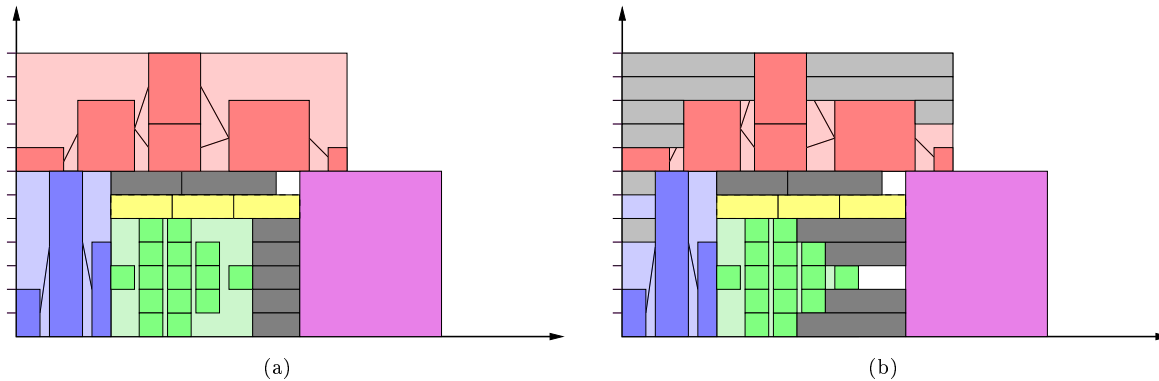


Figure 2: Same schedule view a clear view of the scientific workflows (a) and with idle times filled by sequential jobs (b).

some resources remains idle during the execution of the workflow. The main objective of this internship is to opportunely schedule more sequential tasks during these newly available idle slots as shown in Figure 2 (b).

The candidate will have to go through the following milestones during this internship:

- Study of some selected applications (workflow and parameter sweep) used in the Decryphon project¹ and previous scheduling studies of these applications.
- Define a mechanism allowing a workflow scheduling engine to publish the idle slots created by the produced schedule.
- Design a selection mechanism to select sequential tasks that can fill some idle slots.
- Handle uncertainty. Performance prediction is a complex problem. The estimations made by the scheduling algorithm to build the Gantt chart can be erroneous. The same kind of error may occur for the sequential jobs as well.
- Consider a time-sharing mode in which the opportunist sequential tasks are allowed to finish their execution even if they have to share resources with the tasks of the workflow. The objective is to prevent resource wasting due to the killing of best-effort tasks. This implies a study of the impact on the main application and thus to find a fair tradeoff.
- Develop a first implementation the proposed solution in a simulation context within the SimGrid² framework. Test and validate this solution.
- If successful, an actual implementation will have to be develop and tested on the Grid'5000³ experimental platform.

Prerequisites: Basic C programming skills and some knowledge about scheduling are mandatory.

¹<http://www.decrypthon.fr/>
²<http://simgrid.gforge.inria.fr>
³<http://www.grid5000.fr>