
Level of qualifications required: PhD or equivalent
Fonction: Post-Doctoral Research Visit

About the research centre or Inria department

The Inria Sophia Antipolis - Méditerranée center counts 37 research teams and 9 support departments. The center's staff (about 600 people including 400 Inria employees) is composed of scientists of different nationalities (250 foreigners of 50 nationalities), engineers, technicians and administrators. 1/3 of the staff are civil servants, the others are contractual. The majority of the research teams at the center are located in Sophia Antipolis and Nice in the Alpes-Maritimes. Six teams are based in Montpellier and a team is hosted by the computer science department of the University of Bologna in Italy. The Center is a member of the University and Institution Community (ComUE) “Université Côte d’Azur (UCA)”.

Assignment

A basic step in machine learning is to compute parameter models using some stochastic gradient method in a distributed way. The dataset is split among different executors, each using its data subset to compute a noisy estimate of the gradient of the loss function. The gradient estimates are then averaged during a synchronization phase and this improved estimate is used to update the parameter models. The synchronization phase is time consuming also because of stragglers, so that asynchronous solutions have been proposed, where nodes may work on stale data and read-write race conditions may arise. Empirically, on single-node systems, these asynchronous algorithms have yielded order-of-magnitude improvements in performance (see e.g. [Feng12]). While a significant performance improvement can be obtained, convergence is not guaranteed, and a faster computational throughput does not necessarily guarantee a faster convergence [Kadav16]. In particular, faster nodes can bias the convergence. Large-scale data-parallel computation frameworks, like Spark, usually rely on the bulk synchronous parallel model with high synchronization overhead. [Gonzalez15] shows how asynchronous primitives can be introduced in Spark in order to implement stochastic gradient or alternating direction method of multipliers, but convergence is not guaranteed. [Kadav16] proposes another form of fine-grained synchronization, where each executor only depends on a few other executors, in the sense that requires only their state updates to be able to proceed in the computation. This partial dependency reduces communication overhead and may mitigate the effect of stragglers, but may also increase the number of iterations required for convergence.

Our goal is to propose and evaluate the performance of distributed asynchronous optimization algorithms for large-scale computation frameworks and in particular for Spark. There are a number of open research directions. First, as we mentioned, convergence to the optimum is not guaranteed for many of the proposed solutions. Second, performance models quantifying convergence speed improvement from asynchronous approaches are missing. Finally, when partial dependencies among executors can be introduced as in [Kadav16], it is not clear what is the optimal dependency graph. The candidate is invited to work on (a subset of) these research directions.

Main activities

Research activities

Skills

We are looking for two different possible profiles:

1) candidates with a strong background on optimization (and in particular distributed optimization) and probability,

or
2) candidates with a strong expertise on Spark, at the level to be able to modify Spark code.

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

**Remuneration**
Gross Salary: 2650 brutto per month