MapReduce, Spark or TensorFlow. Our team has been active in research to provide distributed programming easily accessible in some restricted domains, for instance of the spectrum) consensus or data flow. Frameworks and languages are making Highly successful and explicative abstractions already exist, such as (at opposite ends general distributed programs.

We believe that the situation is ripe for a new, high-level approach. The proposed deterministic, and weak consistency.

Thus, currently, large numbers of non-expert programmers are required to navigate uncomfortable and weighty trade-offs in the presence of non-composable, non-deterministic, and weak consistency. Furthermore, applications have opposing requirements. On the one hand, correctness (controlling what the system does), requires events to happen in a reliable, deterministic way. On the other, application performance (including availability, responsiveness and throughput), requires concurrent, asynchronous execution. There is no single right solution to this trade-off; it depends on the application requirements, the expected environment and workload, the available resources, etc. A promising direction is a hybrid approach, where updates avoid coordination by default, but specific operations that are essential to application correctness are synchronised. Getting this right is difficult: current practice in building distributed systems rests on programmer expertise, i.e., trial and error, which is costly and dangerous.

Thus, currently, large numbers of non-expert programmers are required to navigate incompatible and weighty trade-offs in the presence of non-composable, non-deterministic, and weak consistency.

The post-doc aims to develop methods, tools and languages to aid the programmer of applications in a hybrid approach, where updates avoid coordination by default, but specific operations that are essential to application correctness are synchronised.

The research has both a fundamental and an applied aspect and aims for practical results. Candidates to this position should hold a PhD in Computer Science/Informatics or a related field. They should have an excellent academic record and experience in distributed systems, distributed databases, and/or verification tools. In addition to research experience, he or she should be a good developer and experimenter at large scale, and have teamwork and communication skills. Industrial experience and good knowledge of Erlang and/or node.js is a plus.

A resume or Curriculum Vitæ.
A list of courses and grades of the last two years of study (an informal transcript is OK).
A list of your top publications and/or open-source developments.
Names and contact details of three references (people who can recommend you), whom we will contact directly.

Sécurité défense:
Ce poste est susceptible d’être affecté dans une zone à régime restrictif (ZRR), telle que définie dans le décret n°2011-1425 relatif à la protection du potentiel scientifique et technique de la nation (PPST). L’autorisation d’accès à une zone est délivrée par le chef d’établissement, après avis ministériel favorable, tel que défini dans l’arrêté du 03 juillet 2012, relatif à la PPST. Un avis
both high availability and correctness, e.g., developing high-level distributed data
types (CRDTs), and efficient and correct protocols (NMSI, TCC).
Our AntidoteDB database supports concurrent data types, including simple ones such
as counters, flags, sets and maps, and more complex ones enforcing common
invariants, such as uniqueness, bounded counter, referential integrity, and access
control; multiple operations compose into atomic transactions. Our CISE tool verifies
that the execution model of an application is sufficiently synchronised to guarantee
the application’s safety invariants.

However, currently, the heavy lifting remains manual. We propose to develop
programming methods and tools that allow the developer to make use of the full
power of concurrency and distribution; this may include:

1. Shared and persistent data objects, with replication and versioning.
2. Asynchronous (concurrent) and synchronous (consensus-based) operation
   invocation, with transactional and causal consistency guarantees.
3. Publish-subscribe/data flow, with forward and backward paths, and dataflow
   combiners. Data flow carries any mixture of state, delta, or operation.
4. Metadata, such as timestamps, provenance, security labels, or accounting
   information.
5. Specifying preconditions, effects and invariants, in order to enable verification.
6. Elastic configuration of the number and placement of computation and data
   entities, transparently to the program text.

At the same time, our approach helps avoid many of the opportunities for error, by
focusing on the essential properties of application correctness. It is often the
invariants required over application data that dictates the protocol for accessing
the data; this is an intuition that programmers commonly apply. Hence, we aim to
apply leverage language and verification tools, to aid the programmer in choosing the
best consistency level and in synthesising a program that respects its specification.

Avantages sociaux
- Subsidised catering service
- Partially-reimbursed public transport

Rémunération
- Location: Sorbonne Université, 4 Place Jussieu - 75006 Paris
- Gross Salary per month: 2 653 €

Security and defense procedure:

This position is likely to be situated in a restricted area (ZRR), as defined in Decree
No. 2011-1425 relating to the protection of national scientific and technical potential
(PPST).

Authorisation to enter an area is granted by the director of the unit, following a
favourable Ministerial decision, as defined in the decree of 3 July 2012 relating to the
PPST. An unfavourable Ministerial decision in respect of a position situated in a ZRR
would result in the cancellation of the appointment.