so called “combinatorial explosion” problem that makes verification feasible only for
executions, even those that are unlikely to occur in practice, and thus at detecting so-
correctness. This component aims at exhaustively verifying the possible system
estimates for distributed programs running on various execution platforms. SimGrid
resembles as being much more complicated than sequential
programming, and the need to resort to complex scheduling or communications
architectures, accelerators such as GPUs or Xeon Phi, complex interconnection
networks etc.) have broken the traditional uniform parallel programming model. Hence, to exploit these new architectures, application developers have no other
choice but to make their applications very adaptive and to remove synchronizations as
much as possible. To cope with resource heterogeneity and complex topologies, they
also come up with over-sophisticated data management, scheduling, and load balancing strategies. This shift has two consequences. First, parallel programming has
always been recognized as being much more complicated than sequential
programming, and the need to resort to complex scheduling or communications
strategies and to loosen synchronizations makes it even more error-prone. The
resulting bugs occur almost never at small scale but quite frequently and in ways that
are extremely hard to reproduce at large scale, which makes them particularly difficult
to track down and remove. Second, the ever-growing complexity of software and
hardware makes the understanding of their performance almost impossible. There are
now dozens and dozens of layers with complex interactions that can be used to
optimize performance. Correctly configuring such systems has becomes a nightmare
even for experts.

SimGrid [1] is a software platform that provides models and APIs for simulating
distributed systems; it has been shown capable of computing precise performance
estimates for distributed programs running on various execution platforms. SimGrid
also contains a model checking component [2] for dynamically verifying system
correctness. This component aims at exhaustively verifying the possible system
executions, even those that are unlikely to occur in practice, and thus at detecting so-
called “Heisenbugs” that are extremely difficult to reproduce.

The main limitation of model checking as it is currently implemented in SimGrid is the
so called “combinatorial explosion” problem that makes verification feasible only for

**Contexte et atouts du poste**

**Environment of the position**

The post-doctoral researcher will be hosted by the VeriDis research group (http://veridis.loria.fr/) of Inria Nancy Grand Est and LORIA and will interact with the teams MEXiCo (https://www.inria.fr/en/teams/mexico) of Inria Saclay and LSV and MYRIADS (http://www.irisa.fr/myriads/) of Inria Rennes Bretagne Atlantique and IRISA. The objective of the VeriDis team is to contribute research on the modeling and verification of distributed algorithms and systems. VeriDis contributed to the existing model checker within SimGrid [2]. We intend to intensify our involvement in making formal verification techniques available to “real-world” programmers, and we believe that the combination of statistical model checking and simulation for performance evaluation will be fruitful. The statistical model checker COSMOS has been developed mainly by the team MEXiCo whose members are experts in probabilistic systems and their verification. The MYRIADS team is specialized in designing and implementing systems and environments for autonomous service and resource management. It hosts core SimGrid contributors.

**Mission confiée**

**Scientific context**

The recent evolutions of High Performance Computing platforms (multicore and NUMA architectures, accelerators such as GPUs or Xeon Phi, complex interconnection networks etc.) have broken the traditional uniform parallel programming model. Hence, to exploit these new architectures, application developers have no other
choice but to make their applications very adaptive and to remove synchronizations as
much as possible. To cope with resource heterogeneity and complex topologies, they
also come up with over-sophisticated data management, scheduling, and load balancing strategies. This shift has two consequences. First, parallel programming has
always been recognized as being much more complicated than sequential
programming, and the need to resort to complex scheduling or communications
strategies and to loosen synchronizations makes it even more error-prone. The
resulting bugs occur almost never at small scale but quite frequently and in ways that
are extremely hard to reproduce at large scale, which makes them particularly difficult
to track down and remove. Second, the ever-growing complexity of software and
hardware makes the understanding of their performance almost impossible. There are
now dozens and dozens of layers with complex interactions that can be used to
optimize performance. Correctly configuring such systems has becomes a nightmare
even for experts.

SimGrid [1] is a software platform that provides models and APIs for simulating
distributed systems; it has been shown capable of computing precise performance
estimates for distributed programs running on various execution platforms. SimGrid
also contains a model checking component [2] for dynamically verifying system
correctness. This component aims at exhaustively verifying the possible system
executions, even those that are unlikely to occur in practice, and thus at detecting so-
called “Heisenbugs” that are extremely difficult to reproduce.

The main limitation of model checking as it is currently implemented in SimGrid is the
so called “combinatorial explosion” problem that makes verification feasible only for

**Informations générales**

- **Thème/Domaine** : Preuves et vérification
- **Ville** : Villers-lès-Nancy
- **Centre Inria** : CRI Nancy - Grand Est
- **Date de prise de fonction souhaitée** : 01-11-2018
- **Durée de contrat** : 1 an, 4 mois
- **Date limite pour postuler** : 06-06-2018

**Contacts**

- **Equipe Inria** : VERIDIS
- **Recruteur** : Merz Stephan / stephan.merz@loria.fr

**L'essentiel pour réussir**

**Skills and profile**

Required qualification: Candidates must have obtained a PhD in computer science on September 1, 2016 or later, and prior to the date of the start of the post-doctoral research. Candidates who are still preparing their PhD at the date of the application are expected to indicate the date of the PhD defense and the composition of the PhD committee.

**Additional information**

The duration of the post-doctoral position is 16 months. The starting date should be between Nov. 1st, 2018, and Jan. 1st, 2019. The salary is 2,653 euros gross per month (about 2,130 euros net), health insurance included. Inria can help with finding accommodation and with obtaining a residence permit.

**A propos d'Inria**

Inria, institut de recherche dédié au numérique, promeut « l’excellence scientifique au service du transfert technologique et de la société ». Inria emploie 2700 collaborateurs issus des meilleures universités mondiales, qui relèvent les défis des sciences informatiques et mathématiques. Son modèle ouvert et agile lui permet d’explorer des voies originales avec ses partenaires industriels et académiques. Inria répond ainsi efficacement aux enjeux pluridisciplinaires et applicatifs de la transition numérique. Inria est à l’origine de nombreuses innovations créatrices de valeur et d’emplois.

**Conditions pour**
We expect the candidate to be familiar with the following subjects:

**Principales activités**

**Project description**

The objective of this project is to adapt and implement the technique of statistical model checking [4] within SimGrid. In this approach, instead of attempting to exhaustively compute all possible system behaviors, executions are sampled according to the probability distributions associated with the target execution platform in order to estimate the probability that a property is satisfied in the system. Beyond the analysis of Boolean properties, the same technique can provide estimates for other relevant measures such as the average number of messages in a communication buffer or the expected time for achieving a task. Such a capability would be a useful complement to the current possibilities of deterministic performance evaluation through simulation, by allowing for a range of behaviors instead of a precisely determined one. Because statistical model checking is based on sampling individual executions, there is no need to compute and store the entire state space of the system, and much more complex systems can be analyzed than what is possible using traditional (probabilistic) model checking techniques (see e.g. [5]).

The concrete objectives of this post-doctoral research proposal are listed below. The range of subjects that will actually be covered will be determined taking into account the interests of the candidate and will be adapted according to the progress of the work. The subject combines conceptual research and implementation tasks to make statistical model checking available to users of the SimGrid platform, for applications to real-life programs and platforms.

- Identify relevant probabilistic parameters and properties of interest supported by SimGrid.
- Implement a generic interface for performing statistical verification on top of SimGrid. A first experiment has been carried out within a master's thesis and enables some simple analyses, using Simgrid as a black box. An interesting extension would be to follow the work done in COSMOS [5] to obtain and exploit information generated during the execution.
- Evaluate the scalability of the technique and the expressiveness of the extensions mentioned above by applying the approach to relevant examples that have already been implemented within SimGrid.
- Study a possible coupling with performance evaluation as traditionally performed within SimGrid.

**Bibliography**


**Compétences**

We expect the candidate to be familiar with the following subjects:

- Machine Learning
- Distributed systems
- Probabilistic model checking
- Statistical model checking
- Performance evaluation
- Verification of systems
• Solid knowledge on formal verification techniques, in particular model checking and/or run-time verification.
• Solid programming experience, in particular for system programming in C.
• Basic knowledge of probability theory and statistics.

We appreciate the willingness of the candidate to get involved in both conceptual research and in actual implementation of the model checking techniques within SimGrid. He or she should also be willing to visit the partner sites in Rennes and Paris.

Avantages sociaux
• Subsidised catering service
• Partially-reimbursed public transport
• French courses

Rémunération
Salary: 2653€ gross/month