

post-doc proposal:

Converting System-Level Checkpoints of HPC Applications for their Simulation and Verification

Advisor: Martin Quinson (IRISA, team Myriads). Martin.Quinson@ens-rennes.fr

Postdoc Duration: 18 months, starting as soon as possible.

Executive summary: This work aims at automating the conversion of checkpoints captured on real distributed applications (typically, HPC code using MPI) so that they can be loaded in the SimGrid simulator and model-checker. This requires (among others) to convert on the fly internal data structures of the MPI libraries and runtime. The motivation is to switch between methodologies (real execution, simulation, formal verification) to ease the debugging of these large, dynamic and complex applications.

Expected Skills: Very strong ability in C/Linux system programming.

Context

High performance computing (HPC) platforms have always been very challenging for the programmers in terms of size, heterogeneity and dynamicity. The design and evaluation of monolithic programs is currently becoming intractable, and the developers must rely on other approaches, more dynamic and adaptive, to deal with the HPC systems that will become available in the next decade.

This poses a major challenge to the applications' correction, as designing distributed applications is notoriously difficult when they don't have a rigid and simplistic communication pattern. Several techniques and tools have been developed Formal Methods, but these tools remain difficult to use in practice. The SimGrid project¹ [1] aims at filling this gap, making it possible to study both the performance and correction of unmodified MPI applications within the comfort of the simulator.

SimGrid 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 the system correction. 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.

This post-doctoral work will address one of the major limitation of SimGrid in this context: currently, the users must choose the used methodology (real execution, simulation, formal verification) beforehand. This makes for example difficult to track rare bugs: a single execution will probably not uncover them while a full-range exhaustive formal verification is too time-consuming in practice.

Research Questions

Our goal is to allow the hot switch between real execution (on top of a runtime such as MPICH3) simulation and verification, during the course of a given execution. The idea is to restrict the formal verification by starting directly from a checkpoint that led to a problem during the real execution. Once discovered through exhaustive testing, the problematic path will then be studied in all extends in the simulator, in a complete and reproducible manner.

The considered research questions are thus as follows: How to match the concepts and internal mechanisms of each considered methodological approach? How to convert dynamically the state of an application executed under a given methodology to make it usable with another methodology? To answer these questions, a first prototype of the proposed solution should be developed to evaluate in practice the relative advantages of the different possible approaches.

A promising approach is to extend a classical checkpointing solution such as DMTCP [3]. This tool proposes mechanisms to virtualize some parts of the applicative state to ease its restoration (this is for example used for the socket exact file descriptor or the child's pid after a fork). It seems interesting to use this to virtualize and rewrite on the fly the data structures used in MPI interface, that constitute the main differences when switching between methodologies.

¹ SimGrid: Versatile Simulation of Distributed Systems (<http://simgrid.gforge.inria.fr/>).

Work Plan

The concrete objectives of this 18 months postdoc 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 formal verification available in practice to users programmer of real HPC applications through the use of SimGrid.

- Identify the part of the applicative state that are specific to real executions (resp. formal verification) in MPI applications, and match the corresponding concepts.
- Implement a first prototype of the targeted framework using DMTCP or another similar approach, and prove its effectiveness with small-scale applications.
- Evaluate experimentally the proposed techniques on medium- or large-scale applications. Our goal would be to discover some previously unknown bugs in large-scale applications, such as Star-PU² [4].

The results should then be published to highly ranked scientific conferences and journals. Since this work is multi-disciplinary, we see many publication opportunities. The list of possible venues include IPDPS, SuperComputing, ICS High Performance, HPDC, EuroMPI, ACM TPDS (in HPC and Distributed Systems), TACAS or Runtime Verification (in Formal Methods), as well as EuroSys, USENIX or ACM TOCS (in Operating Systems).

Skill required

In addition to the skills that can reasonably be expected from a doctor in Computer Science, the applicant should have a **very strong** knowledge of system programming in C, and of Linux and other modern Unix-based Operating Systems. An experience in distributed systems and HPC programming will be appreciated. No experience in formal methods is expected from the applicants.

Environment

The post-doctoral researcher will be hosted by the Myriads research group (<http://team.inria.fr/myriads>) of Inria Rennes and IRISA, gathering some authors the SimGrid simulator and model-checker. S/he will interact closely with the team SUMO (<https://www.irisa.fr/sumo/>) of Inria Rennes and IRISA, specialist of formal testing methods. The team STORM (<https://team.inria.fr/storm/>) of Inria Bordeaux and LABRI, gathering the authors of StarPU, will also be associated to the current work, as part of the Hac Specis research program (<http://hacspecis.gforge.inria.fr/>). S/he will also benefit of the long-standing collaboration between the Myriads team and the research group of Gene Cooperman (Northeastern University), author of the DMTCP framework.

References

- [1] H. Casanova, A. Giersch, A. Legrand, M. Quinson, F. Suter. *Versatile, Scalable, and Accurate Simulation of Distributed Applications and Platforms*. Journal of Parallel and Distributed Computing (JPDC), 2014. <http://simgrid.gforge.inria.fr/>
- [2] S. Merz, M. Quinson, C. Rosa: *SimGrid MC: Verification Support for a Multi-API Simulation Platform*. Formal Techniques for Distributed Systems (FMOODS/FORTE 2011), LNCS 6722, pp. 274-288. Springer, 2011.
- [3] J. Cao, G. Kerr, K. Arya and G. Cooperman. *Transparent Checkpoint-Restart over InfiniBand*. ACM Symposium on High Performance Parallel and Distributed Computing (HPDC), 2014.
- [4] L. Staniscic, S. Thibault, A. Legrand, B. Videau, and J.-F. Méhaut. *Faithful Performance Prediction of a Dynamic Task-Based Runtime System for Heterogeneous Multi-Core Architectures*. Concurrency and Computation: Practice and Experience, 27:4075–4090, May 2015.

²StarPU: A Unified Runtime System for Heterogeneous Multicore Architectures (<http://starpu.gforge.inria.fr/>).