Emulation through Simulation with Simterpose

Executive summary: The goal of this project is to design an evaluation environment for distributed applications (HPC or P2P applications) where the real applications are executed unmodified within a virtual environment that is simulated by the SimGrid simulation framework.
Key skills required: system programming in C on Linux; deep understanding of OS principles
Research Unit: Inria Nancy – Grand Est, team AlGorille (leader: Jens Gustedt)
Advisors: Martin Quinson, Lucas Nussbaum (<first.last@loria.fr>)

Context

Distributed systems such as grids, clusters, peer-to-peer systems, high-performance supercomputers, cloud computing infrastructures or desktop computing environments, benefit of an ever increasing popularity nowadays. Distributed applications (such as decentralized data sharing solutions, games, high-traffic web applications or scientific computations) are executed routinely on these systems.

By nature, the resulting environments and applications are extremely complex and dynamic because they aggregate thousands of elements that are heterogeneous and shared among several users. This make these systems very challenging to study, test, and evaluate. Computer scientists traditionally study their systems *a priori* by reasoning theoretically on the constituents and their interactions. But the complexity of these systems makes this methodology near to impossible, explaining that most of the studies are done *a posteriori* through experiments.

Three main methodologies exist to experiment with computer systems: real-scale experiments using testbeds, simulation and emulation. *Real-scale* (or *in situ*) consists in executing the real application under study on an experimental platform like Grid'5000[1] or PlanetLab[2]. On the opposite, with *simulation*, both the application and the environment are replaced by models, and the interactions between both models are computed by a simulator. *Emulation* can be seen as an intermediate approach where the real application is executed within a synthetic environment. Typically, one will use a homogeneous cluster of machines as an execution environment, and use an emulation layer that degrades the physical performances in order to reproduce the complex conditions found on the real Internet. Distem is an example of such an emulator, developed in the Algorille team.

SimGrid is a toolkit (developed by the AlGorille team in collaboration) providing core functionalities for the simulation of distributed applications in heterogeneous distributed environments. This project aims at facilitating research in the area of distributed and parallel application scheduling on distributed computing platforms ranging from simple network of workstations to Computational Grids. It is however not possible to study real applications directly on SimGrid: users have to extract the logic of their applications and rewrite them using the specific interfaces of SimGrid.

The Simterpose project, that the proposed work aims at improving, tries to alleviate this by providing a way to use SimGrid as an emulator. This would allow real applications to be executed on virtual platforms emulated by SimGrid. Instead of degrading the performance of the physical platform as in classical emulators, Simterpose would intercept all computations and communications and delay them according to the computations of the simulator.

Description

A prototype of Simterpose was proposed during a master internship aiming at evaluating the feasability of the approach. It currently only allows the extraction of a trace of applications' actions that could theoretically be replayed in SimGrid. The simulator is not able to deal with these traces yet, neither for offline analysis once the complete trace has been captured, nor online directly to delay the application's actions according to simulation results provided by SimGrid.

There is thus a lot of work remaining on simterpose, that constitute an exciting field of investigation with many open lead for interesting research opportunities. Some of these leads are listed below.

[1] Grid'5000 experimental platform: `http://www.grid5000.fr`
[2] PlanetLab experimental platform: `www.planet-lab.org`

- It is first necesssary to continue the development of Simterpose to provide real-time online emulation on top of SimGrid. Several proof of concept implementations were developed in our team, and changing them into a working full prototype should be straightforward. Once done, it will be necessary to evaluate Simterpose by running real distributed applications. Two main categories of applications are currently tageted: P2P applications (such as domestic Bit Torrent or VoIP clients) and high-performance computing applications written using MPI (be they in C or Fortran). Other classes of applications may be considered, such as Business applications running on a Tomcat application server or HPC applications that are not written using MPI. A comparison to the existing emulators will certainly prove very interesting.
- To improve the scalability of the resulting tool, it may be necessary to distribute the execution of the user application on several cluster nodes (with a centralized SimGrid instance to provide the simulation). A similar effort is currently ongoing within the SimGrid project, and the student working on Simterpose may help this effort. The developed solution, if any, will be evaluated on Grid'5000. An interesting challenge would be to leverage the thousand of nodes available in Grid'5000 to either run billions of Bit Torrent clients (using the real implementations), or to run a Linpack test used to rank the HPC platforms[3] involving milions of nodes to predict the performance of future supercomputers.
- Another possible use of the Simterpose tool is to help understanding the semantics of collective operations in MPI implementations such as OpenMPI and MPICH2. The distributed algorithm used by these runtimes depend on parameters such as the amount of data to transfert and the amount of nodes participating to the communication. These parameters can be extracted by source-code extraction, but this bothersome task is to be done again for each version of the runtimes. It would be more interesting to use Simterpose to conduct blackbox analysis on the middleware, and check that the previously analysed semantics did not change with the new versions.
- Yet another possible lead would be to combine simterpose with the model-checker that is integrated within SimGrid. This task is certainly technically very challenging, but it would allow to conduct an exhaustive exploration of execution paths in a real application. Such tool could reveal very interesting for bug-finding purposeswhile contributing to the state of the art in the domain of dynamic verification of real applications. For example, the asynchronous group communication primitives recently introduce by the MPI-3.0 standard are losely standardized to give some optimization freedom to the implementations. This increases the possibility of semantic bugs resulting from the complex interactions between the applications, runtimes and platforms, and such a tool would be very welcomed to explore these issues.

As demonstrated by these few research leads given, the scope of possible researches on this is very wide. It is expected that the postdoctoral student will contribute his/her own ideas, and follow his/her own research agenda within the field provided by Simterpose.

Skills required

In addition to the skills that can reasonably be expected from post-doctoral students, the applicant should have a strong knowledge of system programming in C, and of modern Unix-based Operating Systems such as Linux.

Links

- SimGrid : `http://simgrid.gforge.inria.fr/`
- Algorille Research team: `http://www.loria.fr/equipes/algorille/`
- Tutors: `http://www.loria.fr/~quinson/` and `http://www.loria.fr/~lnussbau/`
- Publication from a previous master intern on Simterpose:
  M. Guthmuller, L. Nussbaum et M. Quinson. Émulation d'applications distribuées sur des plates-formes virtuelles simulées (`http://hal.archives-ouvertes.fr/inria-00565341/en/`)

[3] Top 500 SuperComputer Sites: `http://www.top500.org`