Semantic Annotations for Numerical Simulations

Project Context

Numerical simulations consist of the execution on a computer of a mathematical model that describes the behavior of real-world phenomena. They have applications in many areas such as life sciences, materials science, physics, climatology, as well as engineering, social sciences, economics.

Simulation software and computers evolve together. However, because computers and computer technology have a lifetime of a decade, simulation software has to be continuously adapted for their new execution platforms to guarantee correct results and achieve the best performance. This software maintenance, testing, and verification is an arduous task, in particular when numerical reproducibility cannot be guaranteed, for example, when porting a code to a new architecture (e.g., from CPU to GPU).

With this project, we would like to explore a new methodology, SimSem or Simulation Semantic, for the testing and verification of deterministic numerical simulations. The core idea is to annotate the simulation code to highlight the main characteristics of the simulation in terms of control flow and identify key variables and data items. Put together, these annotations will build a semantic representation of the simulation that is independent of the software implementation. The semantic insights given on the simulation data offer the possibility to compare independent executions, possibly from different implementations, while accounting for numerical variability inherent to parallel execution and heterogeneous architectures.

This approach can be applied to the testing, debugging, and verification of numerical simulation software in the long term. During this project, the SimSem methodology will be experimented with mini-apps and target use cases related to software engineering: result verification, regression testing and performance analysis.

Internship Proposal

Objectives

The project targets regression testing and high-level performance profiling, for the comparison of different implementations of the same problem.

It aims to

- prototype the SimSem instrumentation library that can be used to annotate numerical simulation codes and allow collecting runtime traces of simulation executions;
- and design tools to compare executions, results, and performances of simulations, independently of their algorithms or execution platforms.

Tasks

- Define an initial set of semantic annotations
- Prototype the SimSem instrumentation library based on LTTng
- Annotate a selected set of mini-apps, offering various implementations
- Apply trace analysis to regression testing and performance profiling

Duration

4 to 6 months

Required skills

- Experience with Linux, command line and control versioning systems (Git)
- Experience in using HPC platforms (e.g., Slurm)
- Knowledge in programming languages: C, C++ or Fortran
- Familiar with parallel programming models: MPI, OpenMPI, CUDA or similar
- Data processing: Python pandas/matplotlib or R tidyverse
- Fluent in English

Host Company

LuxProvide

https://luxprovide.lu/

Dept. of Supercomputing Application Services

Location

Bertrange, Luxembourg

Contact person

Xavier Besseron (xavier.besseron@lxp.lu)

Application Procedure

Send your application to Xavier Besseron (xavier.besseron@lxp.lu) including:

- An updated CV
- A short motivational letter (½ page)
- List of the courses followed during the Master studies

References

- T. L. Clune and R. B. Rood, “Software Testing and Verification in Climate Model Development,” IEEE Softw., vol. 28, no. 6, pp. 49–55, Nov. 2011, doi: 10.1109/MS.2011.117. Available online.
- A. H. Baker et al., “A new ensemble-based consistency test for the Community Earth System Model (pyCECT v1.0),” Geosci. Model Dev., vol. 8, no. 9, pp. 2829–2840, Sep. 2015, doi: 10.5194/gmd-8-2829-2015. Available online.
- T. C. Belding, “Numerical Replication of Computer Simulations: Some Pitfalls and How To Avoid Them,”, Jan. 2000, doi: 10.48550/arXiv.nlin/0001057. Available online.
- B. Geyer, T. Ludwig, and H. von Storch, “Limits of reproducibility and hydrodynamic noise in atmospheric regional modelling,” Commun. Earth Environ., vol. 2, Dec. 2021, doi: 10.1038/s43247-020-00085-4. Available online.
- H. Wan, K. Zhang, P. J. Rasch, B. Singh, X. Chen, and J. Edwards, “A new and inexpensive non-bit-for-bit solution reproducibility test based on time step convergence (TSC1.0),” Geosci. Model Dev., vol. 10, no. 2, pp. 537–552, Feb. 2017, doi: 10.5194/gmd-10-537-2017. Available online.
- [17] M. Desnoyers and M. R. Dagenais. “The LTTng Tracer: A Low Impact Performance and Behavior Monitor for GNU/Linux”. In: Ottawa Linux Symposium 2006. 2006. Available online.
- P. S. Crozier, H. K. Thornquist, R. W. Numrich, et al. Improving Performance via Mini-Applications. Technical Report SAND2009-5574. Sandia National Laboratories, 2009. doi: 10.2172/993908. Available online.
- I. Karlin, A. Bhatele, J. Keasler, et al. “Exploring Traditional and Emerging Parallel Programming Models Using a Proxy Application”. In: IEEE Int. Symp. on Parallel and Distributed Processing. 2013. doi: 10.1109/IPDPS.2013.115. Available online.