Laboratoire d'Informatique de Grenoble
UMR 5217 - CNRS, INRIA, Univ. Grenoble Alpes

Guillaume Huard, UGA Associate Professor
Inria **POLARIS** project, LIG Laboratory
mailto:guillaume.huard@imag.fr

Internship proposal
Trace collection and analysis for ExaScale

Advisors : Guillaume Huard (UGA/LIG, Grenoble), Vincent Danjean (UGA/LIG, Grenoble), Arnaud Legrand (CNRS/LIG, Grenoble)

Required Skills :

— C programming, shell, ssh, git
— Basis of data analysis (with Python or R) is a plus
— Basis of statistics, experiment management, scientific computing is a plus

1 Context : The road to ExaScale

There is a continued need for higher compute performance : scientific grand challenges, engineering, geophysics, bioinformatics, etc. Such studies used to be carried out on large *ad hoc* supercomputers, which, for economical reasons, were replaced by commodity clusters, i.e., sets of off-the-shelf computers interconnected by fast switches. Indeed, the technological advances driven by the home PC market have contributed to achieving high performance in commodity components. For decades, computer performance had doubled every 18 months merely by increasing the clock frequency of the processors. This trend stopped almost two decades ago for reason of electricity consumption and heat. Indeed, the computational power of a computer increases nearly sub-linearly with clock frequency while the energy consumption increases more than quadratically.

As an answer to the power and heat challenges, processor constructors have increased the amount of computing units (or cores) per processor and proposed specialized accelerators (GPUs, TPUs, FPGAs, etc.). Modern High Performance Computing (HPC) systems comprise thousands of nodes, each of them holding several multi-core processors and several accelerators, interconnected by complex high-speed networks and memory hierarchies.

For example, the world fastest computers, the Frontier HPE Cray [1] at Oak Ridge National Laboratory, comprises 74 rack cabinets of 64 nodes each where each node consists of 1 AMD Epyc Trento CPU (with 64 cores) and 4 Radeon Instinct MI250X GPUs 9,472 (with 220 cores). This supercomputer is in production since 2023 and it is the first exascale computer but the race to exascale (i.e., capable of delivering more than 10^{18} Floating Point Operations per Seconds) was engaged more than a decade ago. Other exaflopic supercomputers will be built in 2024 in Germany (Jupiter) and in 2025 in France (Jules Vernes).

Recent evolutions amongst the world's fastest machines [2] confirm the trend of massive level of hardware parallelism and heterogeneity (e.g., Leonardo [3] comprises is made of 3,456 nodes with 1 Intel Xeon CPU with 32 cores and 4 NVidia Ampere GPUs and 1536 nodes with 2 Intel Sapphire Rapids CPUs with 56 cores, Summit [4] is made of 4,608 nodes consisting of 2 IBM POWER9 CPUs with 22 cores and 6 Nvidia Tesla GPUs, etc.) Unfortunately, such a level of complexity at such unprecedented scale raises many challenges from a software point of view : former programming/analysis/optimization/debugging/resource management paradigms need to be completely revisited.

1. https://www.top500.org/system/180047
2. http://www.top500.org
3. https://www.top500.org/system/180128/
4. https://www.top500.org/system/179397/

2 Environment : NumPex/ExaSoft and the POLARIS team

In France, this research effort is coordinated by the NumPEx PEPR [5], which gathers 80 R&D teams around a 41M€ budget, and whose research prototypes are expected to be put into production on the upcoming exaflopic supercomputers. In this project, the NumPEx Exa-SofT WP5 is devoted to the performance observation, analysis, and debugging of exascale scientific applications. It gathers teams from Bordeaux (STORM, TADAAM), Grenoble (POLARIS), and Orsay (Télécom Sud Paris - PDS) that have a long standing expertise in trace analysis.

The members of the POLARIS team focus their research on performance evaluation and optimization of large scale systems and parallel applications. They have a strong expertise regarding parallel applications and environment for parallel programming, performance evaluation of large scale distributed systems, trace collection/analysis/visualization.

Some of POLARIS members have been involved in the Joint Laboratory for Extreme-Scale Computing [6] between University of Illinois at Urbana-Champaign [7] Inria [8], Argonne National Laboratory [9], Illinois' Center for Extreme-Scale Computation [10], the National Center for Supercomputing Applications [11], the Barcelona Supercomputer Center [12], Julich and the Riken. Some of their members have also been involved in the European Mont-Blanc [13] (European scalable and power efficient HPC platform based on low-power embedded technology) that aimed at prototyping HPC supercomputers from from today's energy-efficient solutions used in embedded and mobile devices (e.g., ARM processors). Some of the POLARIS members are also in contact with research & development groups from Bull/ATOS that strongly influence the design and capacity planning of supercomputers.

3 Goal : Scalable tracing and analysis

In the time where applications and supercomputers were relatively simple and regular (i.e., homogeneous clusters running MPI), the commonly used approach for analyzing an application consisted in tracing as much information as possible and then trying to display (e.g., through a Gantt-chart) the content of the trace to detect irregularities, load imbalance, synchronization issues, stragglers, etc.

Because of the sheer volume of data, tracing application in details quickly suffers from scalability issues. First, it is intrusive and can significantly change application performance and their behavior. Second, it requires a large storage space (gigabytes of data are easily produced by tracing a single function calls for a few seconds), which is impractical both from a tracing point of view. Third, it is also impractical from an analysis perspective since (1) just opening the files can become a nightmare and (2) the analysis often amounts to trying to find needle in a haystack.

Indeed, scientific applications are very regular, and although their behavior is stochastic (especially when they have to deal with complex heterogeneous architectures), there is a global structure that should be leveraged not only during the analysis, but also during trace collection. Unfortunately, generic tracing/analysis framework generally completely ignore (i.e., do not exploit) the application nor the supercomputer structure.

To goal of the NumPEx Exa-SofT WP5 is thus to exploit a priori knowledge (needs, structure, regularity) to (1) sensibly choose and highlight relevant information, and (2) smartly compress traces on the fly. In this context, the Bordeaux teams bring a strong expertise in HPC runtimes, the Telecom Sud Paris team brings a strong expertise in application tracing, and the Grenoble team brings a strong expertise in application analysis and visualization.

An ongoing work has started at Telecom Sud Paris Paris Saclay to propose a tracing framework (within EZtrace [14]) capable of detecting repetitive patterns on-the-fly [15] by exploiting the structure

5. https://anr.fr/fr/france-2030/programmes-et-equipements-prioritaires-de-recherche-pepr/numpex-numerique-hautes-performances-pour-lexascale/
6. https://jlesc.github.io/
7. http://illinois.edu/
8. http://www.inria.fr/en/
9. http://www.anl.gov/
10. http://iacat.illinois.edu/
11. http://www.ncsa.illinois.edu/
12. http://www.bsc.es/
13. http://www.montblanc-project.eu/
14. https://eztrace.gitlab.io/eztrace/
15. https://hal.science/hal-03750441/

of the control flow graph of the application. This enables to store particularly compact/hierarchical representation for regular applications (the HTF format).

In this context, several possible research topics are proposed in the POLARIS team, in strong collaborations with the Bordeaux and Telecom Sud Paris teams :

1. **Adapt existing techniques to this new scalable format** : Although many analysis/visualization tools have been developed for classical flat trace formats, none of them can handle the new HTF trace format. Providing tools to display and navigate the content of such traces while exploiting the regularity detected at runtime is paramount. As an example, one may propose showing structured overviews instead of displaying every information, detecting phases and displaying macro-structures, highlighting rare events, computing and displaying imbalance, etc. To this end, high-level languages and visualization frameworks such as R/ggplot/shiny or python/plotnine/Dash should reveal particularly adequate.
2. **Explore adaptive tracing schemes** : In the EZtrace frameworks, lots of tracers with different levels of details can be activated (application events, MPI/CUDA/OpenMP calls, network traffic, PAPI counters, energy probes, etc.). It is rare to require such level of details over the whole life cycle of the application and for all resources. Instead, the detailed tracing of some hardware/software components should be activated either on a regular basis for monitoring purposes or at very specific moments whenever particular situations occur (load imbalance, failure, network overload, etc.). The goal will be to implement some global monitoring in the tracer and allow users to specify when additional lower-level tracing should be activated.
3. **Guided lossy compression, trade-off information loss/efficiency** : Initial evaluation of lossy and non-lossy compression in HTF has already been done recently for regular MPI applications. Such evaluation remains to be done for task-based applications [16] both regular (e.g., dense linear algebra solvers) and irregular (e.g., sparse solvers). This will require to smartly combine the compressors based on the application structure and on the data nature and distribution. Ideally, compressors will be configured to measure the information loss on the fly to allow some tracer introspection and decide on the fly whether tracing should be adapted or not.

This internship will follow the reproducible research [17] good practices. The student will use notebooks (Jupyter [18] and/or Org mode [19]) and all the experiments will be done using Grid5000 [20] platform. It will be done with the support of the Bordeaux and Telecom Sud Paris teams. The bibliography of the related work will be provided to the intern.

This internship will allow you to sharpen your experimental and development skills both on a practical and theoretical side. We expect the intern to be curious, rigorous, and autonomous but advisors will happily provide all the help needed on the software, hardware, theory, experimental methodology.

Budget is secured within the ExaSoft WP5 project for several PhD thesis. This internship can thus lead to a PhD thesis on a similar or related topic, in collaboration with the same teams.

16. https://starpu.gitlabpages.inria.fr/
17. https://en.wikipedia.org/wiki/Reproducibility
18. http://jupyter.org/
19. https://orgmode.org/
20. https://www.grid5000.fr