Laboratoire d'Informatique de Grenoble
UMR 5217 - CNRS, INPG, INRIA, UJF, UPMF

Téléphone : (+33) 4 76 61 20 64
Télécopie : (+33) 4 76 61 20 99
Adresse électronique : mailto:arnaud.legrand@imag.fr
: mailto:brice.videau@imag.fr
: mailto:frederic.desprez@inria.fr

Adresse : Inria / Équipe CORES
DRT / LETI / DACLE – Bâtiment 51C
Minatec Campus
17 rue des Martyrs
38054 Grenoble Cedex 9
France

Proposal for a M2R internship

Optimization of the Auto-Tuning of HPC Application Computing Kernels

Advisors: Brice Videau, Arnaud Legrand, and Frédéric Desprez
Required Skills:

- C programming, UNIX, shell, ssh, git
- Compiling for parallel architectures
- Basics of experiment analysis with R and of mathematical optimization is a plus
- Basic knowledge of ruby is a plus

1 Context

Scientific computing has a huge impact in our everyday lives. Most of our technology nowadays is designed with the help of modeling tools and simulation softwares. Weather forecasting, geological applications, bioinformatics or material science are among the most demanding applications in term of computing power. Very large High Performance Computing (HPC) infrastructures are required to obtain simulation models, more precise results or shorter time to solution. The supercomputers computing power has been tracked by the Top500 list (www.top500.org) for the past 20 years. It shows that supercomputer performance roughly doubled every 18 month during this period and nowadays the fastest supercomputer can manage 33 peta floating point operation per second (10^{15} flops). In the past three years. supercomputers computing performance only doubled and it is likely that the past evolution is not sustainable anymore.

One of the reason behind this stall is the energy wall. Supercomputers energy consumption has grown with each generation and is reaching the 20MW barrier. In order to keep improving the computing power, in this 20MW power budget, the energy efficiency of computing components has to drastically improve. To reach the exaflop (10^{18} flops) rate we need a $20\times$ improvement in power efficiency. In order to keep the energy consumption and the cost of such system lower, supercomputers have evolved from monolithic ad-hoc machines to interconnections of commodity systems. These systems have also become more parallel, with processors composed of several computing cores and accelerator composed of hundreds or thousands of cores. Using more computing units that are less powerful but more energy efficient allows to increase the global efficiency of the machine.

Nonetheless parallel architectures are much more complex to program than sequential ones. Recent supercomputers have become incredibly difficult to program efficiently and applications need to be tuned specifically in order to perform reasonably well. Moreover, several architectures compete in the HPC market and they don't share a common programming model. In this context the everyday work of HPC application developer has become much more oriented on the optimization aspect of software development rather than focusing on adding new functionalities to their application.

2 Environment

The CORSE and MESCAL INRIA teams share a common interest for HPC architectures and HPC applications. The research of MESCAL is focused on performance evaluation aspects of large scale distributed systems while the CORSE researchers focus on the interface between runtime systems and compilers. Both teams have a strong expertise regarding parallel applications and parallel programming and long-standing collaborations in national and international projects.

Both teams are involved in the Joint Laboratory for Extreme Scale Computing between University of Illinois at Urbana-Champaign INRIA, Argonne National Laboratory, Illinois' Center for Extreme-Scale Computation, the National Center for Supercomputing Applications, and the Barcelona Supercomputing Center.

Some of their members are also involved in the European Mont-Blanc Project (European scalable and power efficient HPC platform based on low-power embedded technology). Indeed, HPC systems developed from today's energy-efficient solutions used in embedded and mobile devices are an interesting alternative to accelerators such as GPUs or Intel Xeon Phi. As of today, the CPUs of these devices are mostly designed by ARM. However, ARM processors have not been designed for HPC, and ARM chips have never been used in HPC systems before, leading to a number of significant challenges. One envisioned possibility for designing such exascale platforms is the use of 100,000+ ARM processors and their GPU connected through hierarchical Ethernet networks.

3 Goal

One of the Mont-Blanc project goals is to develop tools and methodologies to help HPC application developers port and tune their code for current and future targets. BOAST is one of the approach proposed by the CORSE team and consists in a meta-programming and auto-tuning framework designed to write performance portable computing kernels. BOAST allows the description of a computing kernel and its possible optimization using and embedded domain specific language (EDSL). The kernel and a combination of optimization can then be generated in a target programming language of choice (FORTRAN, C, CUDA or OpenCL). BOAST can then benchmark (using a selected compiler and compiler options) and test the generated kernel for regressions.

Finding the optimal combination of optimization and compiler flags of a kernel for a given architecture+compiler is extremely time consuming. The goal of the internship is to study the different existing approaches to reduce the number of experiments required to find a satisfactory solution to such optimization problem. Several previous work exist (see bibliography) studied these problems but few, to our knowledge, have focused at the same time on optimizing the combination of meta-programming optimization and compiler flags optimization. The BOAST framework offers an opportunity to study this problem in an integrated environment.

Promising solutions will then be integrated to BOAST or new ones will be developed. This will allow BOAST to optimize more complex kernels as well as increase the quality of the optimization results.

References

[BWH11a] P. Balaprakash, S. M. Wild, and P. D. Hovland. Global and local search algorithms in empirical performance tuning. DOE CScADS Workshop on Libraries and Autotuning for Extreme-Scale Systems, 2011.

[BWH11b] Prasanna Balaprakash, Stefan M. Wild, and Paul D. Hovland. Can search algorithms save large-scale automatic performance tuning? In Proceedings of the International Conference on Computational Science, ICCS 2011, volume 4, pages 2136–2145, 2011.

[BWH12] P. Balaprakash, S. M. Wild, and P. D. Hovland. Efficient optimization algorithms for empirical performance tuning. SIAM Conference on Parallel Processing (SIAM PP 2012), 2012.

[CFH$^+$12] Yang Chen, Shuangde Fang, Yuanjie Huang, Lieven Eeckhout, Grigori Fursin, Olivier Temam, and Chengyong Wu. Deconstructing iterative optimization. TACO, 9(3):21, 2012.

[MF13] Abdul Wahid Memon and Grigori Fursin. Crowdtuning: systematizing auto-tuning using predictive modeling and crowdsourcing. In Proceedings of the International Conference on Parallel Computing (ParCo), pages 656–667, 2013.

[NZN$^+$11] B. Norris, Q. Zhu, T. Nelson, P. Balaprakash, and S. M. Wild. Comparison of search strategies in empirical performance tuning of linear algebra kernels. 2011 SIAM Conference on Computational Science and Engineering, 2011.

[RBHW15] A. Roy, P. Balaprakash, P. D. Hovland, and S. M. Wild. Exploiting performance portability in search algorithms for autotuning. Technical Report ANL/MCS-P5400-0915, Argonne National Laboratory, 2015.

[SYD08] K Seymour, H. You, and J. Dongarra. A comparison of search heuristics for empirical code optimization. In International Conference on Cluster Computing. IEEE, 2008.