Letter of Interest

" Code parallelization for Computational Physics: Challenges, solutions, and future direction."

Authors:
Yacine Hakimi Laboratoire de Méthodes de Conception des Systèmes (LMCS) Ecole nationale Superieure d'Informatique (ESI) Algiers, Algeria y_hakimi@esi.dz

Riyad Baghdadi Computer Science & Artificial Intelligence Lab (CSAIL) Massachusetts Institute of Technology (MIT) Massachusetts, USA baghdadi@mit.edu

Yacine Challal Laboratoire de Méthodes de Conception des Systèmes (LMCS) Ecole nationale Superieure d'Informatique (ESI) Algiers, Algeria y_challal@esi.dz

Abstract :

Most of today's research in many fields, including physics, is based on simulating and analyzing experiments on computers to save time and money and accelerate research while saving time and money. These simulations, as well as the analysis of data and results, require significant computational power. Today's microprocessors are limited in terms of frequency. As a result, the only way to get more performance is to parallelize codes. Writing parallel codes or parallelizing legacy ones (porting legacy code to a parallel architecture) is a non-trivial task, given the heterogeneity of current architectures (We believe that this issue will be the trend in the future as well). Unfortunately, parallel code optimization in these architectures is error-prone and time-consuming task and needs experts that possess the required skills and back-end expertise. Moreover, a parallel code optimized for a given architecture may perform better or worse on a different architecture. We discuss in this document the challenges, current solutions, and the future direction in this research area.

Introduction:

The need for more computational power is a central aspect in many fields of science, such as mechanics, electromagnetism, and thermodynamics. It has been a hot topic in active research in physics. Modeling and simulation of complex physical phenomena are computationally challenging in many disciplines, and to accelerate research in these fields we need more powerful computational architectures.

Due to the physical limitations (power and memory wall), there is no faster CPU through a frequency increase. As a result, manufacturers have turned to heterogeneous parallel architectures with more cores and different accelerators for more performance. Exploiting the full potential of these architectures is crucial, given the tremendous computational needs of computational physics. The only way for computational physics to benefit from these new platforms is to write efficient parallel code optimized for a specific architecture. Due to the complex interaction between parallel code and parallel architectures, optimizing parallel code for heterogeneous architectures is a difficult task that poses numerous challenges.

Challenges:

Nowadays, we find ourselves facing very complex heterogeneous architectures, where writing parallel code has become more challenging due to the complex interaction between parallel code and these parallel architectures. Many parameters affect the parallel execution of the code, such as choosing the algorithm, the parallel programming model, determining the granularity of parallelism, scheduling, and mapping strategy. We have to select the optimal combination of these parameters to get the best performance. It is very challenging, not only for researchers in fields such as physics but even for experts in computer science.

Writing new parallel code is not the only challenge that we face. Millions of lines codes in several fields, including computational physics, were written for sequential architecture over many years. Rewriting and optimizing them for parallel architectures is an error-prone and time-consuming solution.

The number of parameters that must be tuned is large, and the possible value for each parameter increases as the number of processors and new accelerators increases in these heterogeneous architectures. Therefore, the code that was written and optimized for a specific architecture must be rewritten and reoptimized for each new architecture. And the hardware is replaced on a regular cycle to keep up with the latest developments.

Current solutions:

To hide the complexity from the developer, more advanced compilers are designed to generate optimal parallel code automatically. These compilers use accurate optimization heuristics algorithms to automatically determine the best combination of optimizations that improve performance for a given architecture. In the last decade, machine learning has been used to automatically build such heuristics for compilers. The current trend is to utilize Deep learning especially NLP (Natural Language Processing) and GNN (Graph Neural Networks) techniques, to automatically decide how to optimize code.

Limitations and open issues that require additional research:

These compilers are still very experimental, and their advantages in terms of performance for real problems still need to be proven. This research area needs more attention from researchers to develop more efficient machine learning techniques for compiler optimizations and more attention from developers from other fields such as physics to test these solutions in a realistic environment and prove their efficiency. We strongly believe that with the collaboration between all the involved communities (researchers in compilers, machine learning, and physics), we can overcome the limitations and uncertainties in this area, which will significantly help research in both fields.