Using Code-LLMs to enhance performance models for GPU kernels

Olivier Aumage, Benjamin Negrevergne

July 25, 2025

1 Context & purpose of the internship

This internship is a collaboration between the STORM Team at Inria Bordeaux France (High performance computing) and the Miles Teams (Machine Learning), PSL University, Paris, France. The internship will take place at Inria Bordeaux.

Team STORM develops methodologies and tools to statically and dynamically optimize computations on High Performance Computing (HPC) architectures, ranging from task-based parallel runtime systems to vector processing techniques, from performance-oriented scheduling to energy consumption reduction. In particular, STORM's StarPU runtime system brings task parallelism programming to heterogeneous, accelerated computing platforms.

To properly schedule different tasks on the different computing units available, StarPU relies on simple performance models in order to predict the execution time of a task. Although simple, the performance of these models is inherently limited by the fact that they do not "look" into the code of the task to be executed. As a consequence, they are unable to generalize across distinct but similar tasks, or across various runs of the same task with similar inputs, yielding uninformed prediction for most executions.

Recently Code-LLMs (LLMs that can generate code) have demonstrated impressive code generation and code understanding capabilities. Enhanced with such abilities, we hypothesize that performance models could significantly improve their quality and predict a variety of metrics useful for efficient scheduling. Such metrics include execution time but also memory footprint, bandwidth usage, or cache affinities between tasks. Ultimately, such knowledge could be used to better schedule tasks on complex heterogeneous platforms and dramatically improve the performance of the computations.

2 Expected results

In a first phase of the internship, the student will design, train and evaluate a neural-network based regression models capable of predicting execution times (or other relevant metrics) for various GPU kernels.

The models will be inspired from modern open source Code LLM such as Code LLama, and will make predictions based on the source code of the kernel, the input data and the hardware specification of the targeted computing unit. The focus will be on linear algebra kernels such as the ones available in the Pastix library. This first part will require the student to develop a strong understanding of the performance and limitations of existing machine learning models when it comes to understanding the execution of programs.

In a second phase, the student will investigate how these models can be used to improve the scheduling and execution of GPU kernels on complex high performance heterogeneous execution platforms.

This phase will involve experiments to determine which metrics can be accurately predicted and which ones are most valuable for effective task scheduling. Given that LLM based models are also GPU intensive, it will also require evaluating the cost vs. benefit of different models of various sizes and performance.

3 Required skills

- Fluency in C language programming in Unix environment;
- Expertise in using modern development tools such as Git version control, GitLab and GitHub environments, continuous integration frameworks;
- Experience with machine learning and common deep learning libraries such as PyTorch.

4 The list of macro-activities that will be carried out during the internship:

- Understand the current models and their limitation;
- Conduct a detailed bibliographical study of code-LLM and their usage for performance analysis/prediction;
- Suggest one or more code-dependent performance model architecture based on code LLMs, and train it using historical data;
- Conduct thorough experimentation to understand which metric most benefit from this approach;
- Incorporate the new models into StarPU and evaluate the cost vs. benefit ratio.