Full stack optimization for streaming applications

Global Information

Domain: High Performance Computing
Laboratory-Institute: Inria Bordeaux Sud-Ouest, Talence
Duration: 6 months
Contact:

Diane Orhan / diane.orhan@inria.fr
Laércio Lima Pilla / laercio.pilla@inria.fr
Lana Scravaglieri / lana.scravaglieri@inria.fr
Mihail Popov / mihail.popov@inria.fr

Keywords

auto-tuning, machine learning, streaming applications, heterogeneous mapping

Background

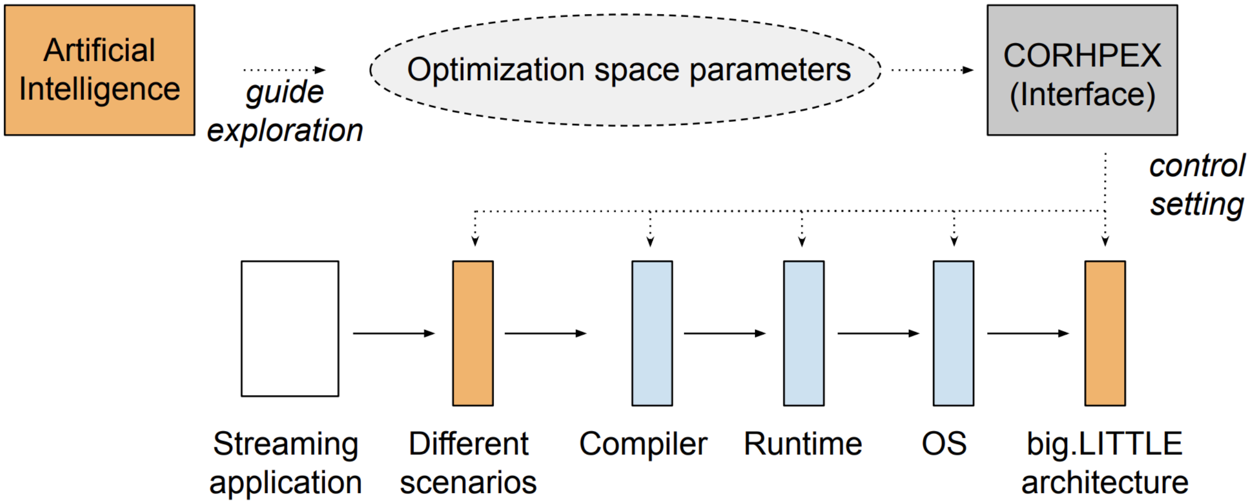
TADaaM is a project-team collaboration between Inria, the University of Bordeaux, and Bordeaux-INP, focusing on the characterization and optimization of scientific applications executed on High Performance Computing (HPC) parallel systems. HPC is crucial as it enables scientific discoveries as well as industrial and societal advancements. However, achieving the full performance potential of an application on a system is very challenging, as both applications and systems are extremely complex.

Our focus in this internship are streaming applications [1], with a specific emphasis in digital communications. Historically, most of these systems and protocols were implemented onto dedicated hardware for peak performance. However, new communication standards such as 5G are emerging with large specifications and numerous possible configurations. Software-Defined Radio (SDR) offers flexible and portable solutions that can address this ever-changing set of standards. However, as SDR operates over diverse hardware, it must adapt its behavior to each system to achieve low latencies/high throughput and energy efficiency. Such adaptation requires the exploration of multiple tunable variables.

Indeed, the hardware stack exhibits diverse parallelism, heterogeneity, and is subject to NUMA effects, employing out-of-order execution, intricate cache hierarchy, shared resources, and proactive data prefetching. The software stack presents diverse levers (tunable, such as parallelism, prefetch, cache, or mapping) to align applications (without code changes) with the unique characteristics of the hardware they run on. While tuning these parameters separately provides some gains, co-optimizing them together is key, as their behaviors can synergize [2].

Description of the Task

This study proposes to explore and model the performance of streaming applications from StreamPU [3] across the different software/hardware optimizations. The intern will access a performance/energy tuning tool named CORHPEX to co-optimize NUMA effects, prefetchers, compilation parameters, processor frequency, and thread placement. The final goal is to optimize streaming applications on new Intel hybrid architectures with the machine learning strategies from CORHPEX. We consider the following steps:



1 Express SDR applications into CORHPEX

The intern can rely on the CORHPEX infrastructure to explore and apply the different optimizations. This first task consists of exposing the streaming and SDR applications from StreamPU to the CORHPEX infrastructure. This requires to provide the different application inputs, to express how to compile and run them, to collect the performance/energy measurements, and to describe the different optimization spaces that are relevant.

2 Interaction between optimizations and tasks

Once we have a set of measures of behaviors across the different configurations, we can study the interaction between the different optimizations for developers and designers using synthetic chains made with StreamPU. For example, data mapping strategies might be affected by worker placement, thus measuring how they interact enables us to co-optimize both of them for higher gains. The intern can leverage the CORHPEX infrastructure to guide (with Artificial Intelligence) and measure the interaction between the different parameters.

3 Deploy optimizations on real scenarios

For portability and flexibility, we aim to optimize an SDR scenario (DVB-S2 standard [4]) on a target system. The intern will implement, measure, and evaluate the portability of different optimizations (throughput, energy consumption, number of cores used) in our real world scenario. This is an opportunity to summarize extract insights for developers, designers, and practitioners. The intern can contribute to the scientific writing and dissemination of these results.

4 Migrate to heterogeneous systems (big.LITTLE)

The intern can start the measurements on a single system (i.e., Intel) and possibly extend them to a different architecture (i.e., E/P cores) to observe the architectural impact on the optimizations. This task requires the intern to update the configuration files to fit the custom architecture details and is key to achieve portable performance and energy efficiency. We have an ongoing collaboration with Uppsala University [5] to optimize such infrastructures: it is an opportunity for future collaboration.

Workflow and Goals

The intern will start by task 1. Then, they can perform tasks 2 and 3 in parallel. Task 4 will be done last if time permits. They are also strongly encouraged to attend group meetings and present their work progress: this is an opportunity to integrate the team TADaaM and the HPC ecosystem in Bordeaux. Depending on the research results, the intern can also participate in the writing process to publish a research article.

The internship goal for the intern is to develop their knowledge on runtime optimizations as well as their writing and presentation skills. The internship is also an opportunity to observe how academic research is conducted.

Requirements

Candidates are expected to be familiar with:

- Data analysis.
- C++ programming (to work with and understand StreamPU).
- Python programming (to work with and understand CORHPEX).
- Scripting languages (e.g., Python again or shell) for setting up experiments.
- Linux. The project requires updating OS environment variables to apply optimizations.

Candidates are expected to be motivated to work in a collaborative environment. Experience in machine learning methods and in NUMA systems is also a plus.

Reference

[1] Cassagne, A., Tajan, R., Aumage, O., Leroux, C., Barthou, D., & Jégo, C. (2023). A DSEL for high throughput and low latency software‐defined radio on multicore CPUs. Concurrency and Computation: Practice and Experience, 35(23), e7820.
[2] Scravaglieri, L., Popov, M., Pilla, L. L., Guermouche, A., Aumage, O., & Saillard, E. (2023). Optimizing performance and energy across problem sizes through a search space exploration and machine learning. Journal of Parallel and Distributed Computing, 180, 104720.
[3] StreamPU: https://github.com/aff3ct/streampu
[4] Cassagne, A., Léonardon, M., Tajan, R., Leroux, C., Jégo, C., Aumage, O., Barthou, D. (2021). A flexible and portable real-time DVB-S2 transceiver using multicore and SIMD CPUs. 11th International Symposium on Topics in Coding (ISTC), pp. 1–5
[5] https://www.diva-portal.org/smash/get/diva2:1888603/FULLTEXT01.pdf