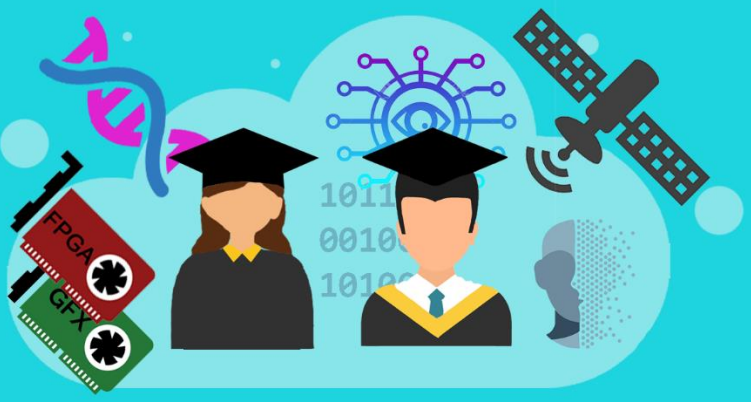
Diploma Thesis

Microproccessors and Digital Systems Laboratory



Hardware- and Code-driven compiler-level optimizations for modern Cloud workloads

Over the last years, the number and size of Cloud infrastructures have experienced a rapid increment. This expansion originates from the fact that more and more stakeholders from different and diverse domains, including but not limited to, healthcare, automotive and agriculture, are embracing Cloud computing as their de-facto model for application execution. Thus, cloud providers are called to handle and execute a diverse set of applications, such as data analytics, web streaming, web searching, scientific simulations and others, while also accounting for delivering performance- and cost-efficient solutions.

To increase resource efficiency, Cloud resource orchestrators perform several optimizations in multiple levels of the stack, i.e., application-level, cluster-level and system-level. From an application-level perspective, compiler optimizations form one approach for tuning application characteristics to enhance resource efficiency. For example, gcc/llvm offers more than 100 tunable optimization options , which can be altered to perform intermediate code optimizations, thus, leading to an optimized binary file. However, applying such optimizations is a non-trivial task due to the high-dimensionality of the configuration space and the possible inter-relationship between the different available configuration knobs. On top of that, determining an optimal set of compiler flags further depends both on i) the source code characteristics of the deployed application and ii) the specifications of the underlying hardware.

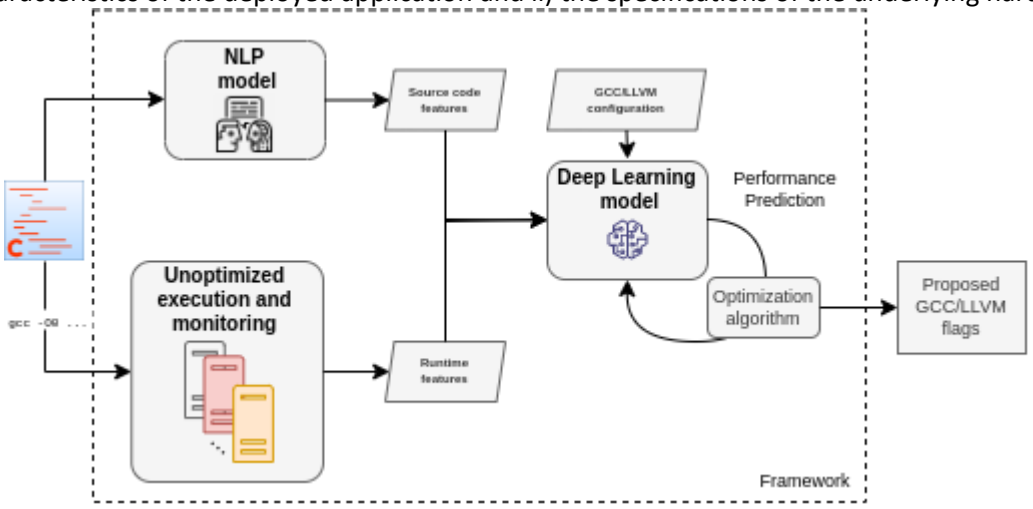


Figure (a)

In this diploma thesis, we will develop a framework that automatically determines optimal compiler configuration options based on source-code specific and hardware related characteristics, as shown in the Figure (a). The framework will leverage modern deep learning techniques (e.g., NLP, LSTMs, Transformers) to model the performance of applications given a certain compiler configuration vector. Moreover, given the high-dimensionality of the configuration space, we will also examine multi-

objective optimization algorithms (e.g., NSGA-II) to traverse efficiently through the entire configuration space.

USEFUL PREREQUISITES:

- Linux, Bash, C.
- Python, Machine Learning frameworks and libraries (e.g., sklearn, Pytorch)

RELATED MATERIAL

1. Cummins, Chris, et al. "Compilergym: Robust, performant compiler optimization environments for ai research." 2022 IEEE/ACM International Symposium on Code Generation and Optimization (CGO). IEEE, 2022.
2. Nikitopoulou, Dimitra, et al. "Performance analysis and auto-tuning for spark in-memory analytics." 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2021.
3. Xydis, Sotirios, Eleftherios Christoforidis, and Dimitrios Soudris. "DDOT: Data Driven Online Tuning for energy efficient acceleration." 2020 57th ACM/IEEE Design Automation Conference (DAC). IEEE, 2020.
4. Cummins, Chris, et al. "End-to-end deep learning of optimization heuristics." 2017 26th International Conference on Parallel Architectures and Compilation Techniques (PACT). IEEE, 2017.
5. Ansel, Jason, et al. "Opentuner: An extensible framework for program autotuning." Proceedings of the 23rd international conference on Parallel architectures and compilation. 2014.

CONTACT INFORMATION

- Dimosthenis Masouros, PhD candidate Microlab NTUA (dmasouros@microlab.ntua.gr)
- Dimitrios Soudris, Professor Microlab NTUA (dsoudris@microlab.ntua.gr)
- Sotirios Xydis, Assistant Professor Microlab NTUA (sxydis@microlab.ntua.gr)