A Study on Automatic Parallelization with OpenMP using Large Language Model

Name: Soratouch Pornmaneerattanatri

Laboratory's name: Software and Design Laboratory

Supervisor's name: Hajimu Iida, Kazutoshi Fujikawa, Kohei Ichikawa, Keichi Takahashi, Yutaro Kashiwa

Abstract

To fully utilize multi-core processors, the development of parallel programs is needed. However, developing parallel programs is a demanding task. Automatic parallelization techniques have been studied to simplify this process by automatically transforming sequential code into parallel code. Most existing automatic parallelization tools employ static analysis, which can identify certain types of parallel structures but fail to detect all, leading to suboptimal performance gains.

In contrast, the recent emergence of the Large Language Models (LLMs) in the Natural Language Processing (NLP) field has led software engineering researchers to adopt them, as LLMs have demonstrated state-of-the-art performance in various tasks. Motivated by these advancements, this study proposes an automatic parallelization tool based on LLMs. To replicate the functionality of existing automatic parallelization tools, two models are developed. The first model classifies parallelizable for-loops, while the second model generates OpenMP directives. Datasets are gathered from two sources, Google BigQuery public datasets and GitHub public repositories, and pre-processed to improve the quality of the OpenMP source code and to facilitate downstream tasks by fine-tuning CodeT5/CodeT5+, one of the Code LLM models.

The proposed models are evaluated using the NAS Parallel Benchmarks. The classification model achieves an F1 score of 0.713. The generation model successfully parallelizes the for-loops in 73% of cases and achieves speedup in five out of the eight NAS Parallel Benchmarks.