A Proposal of Automatic Parallelization using Transformers-based Large Language Models

Laboratory for Software Design & Analysis

established in 2005

Soratouch Pornmaneerattanatri[1], Keiichi Takahashi[2], Yutaro Kashiwa[1], Kohei Ichikawa[1], Hajimu Iida[1]
[1]Nara Institute of Science and Technology, [2]Tohoku University
pornmaneerattanatri.so.pn8@is.naist.jp

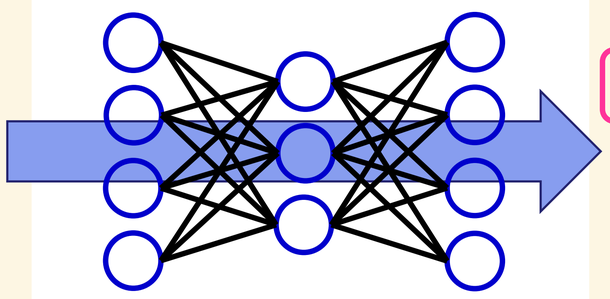
Introduction

In the past decade, CPU architecture transitioned from single-core architecture to multi-core architecture. Parallel programming is required to fully utilize multiple CPU cores. However, developing parallel software is error-prone and requires extensive knowledge of both hardware and software.

Automatic Parallelization using Generative Machine Learning

Input

```
for( int i = 0; i < N; ++i )
  for( int j = 0; j < N; ++j )
    for( int k = 0; k < N; ++k )
      C[i][j] += A[i][k] * B[k][j];
```



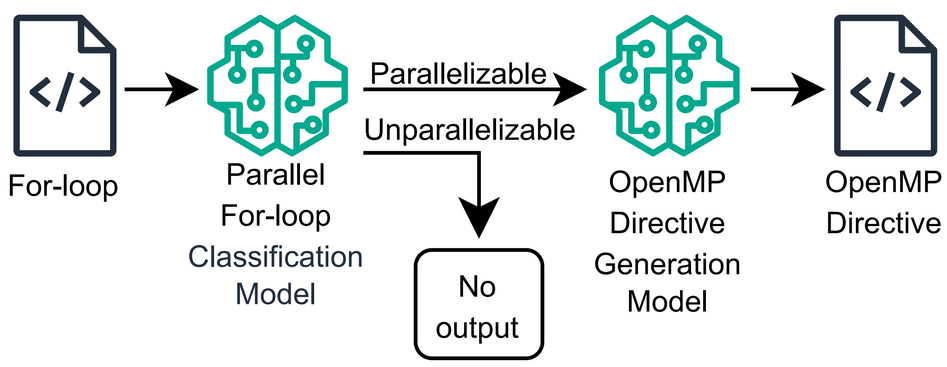
Output

```
#pragma omp parallel for private( i, j, k ) shared( A, B, C )
for( int i = 0; i < N; ++i )
  for( int j = 0; j < N; ++j )
    for( int k = 0; k < N; ++k )
      C[i][j] += A[i][k] * B[k][j];
```

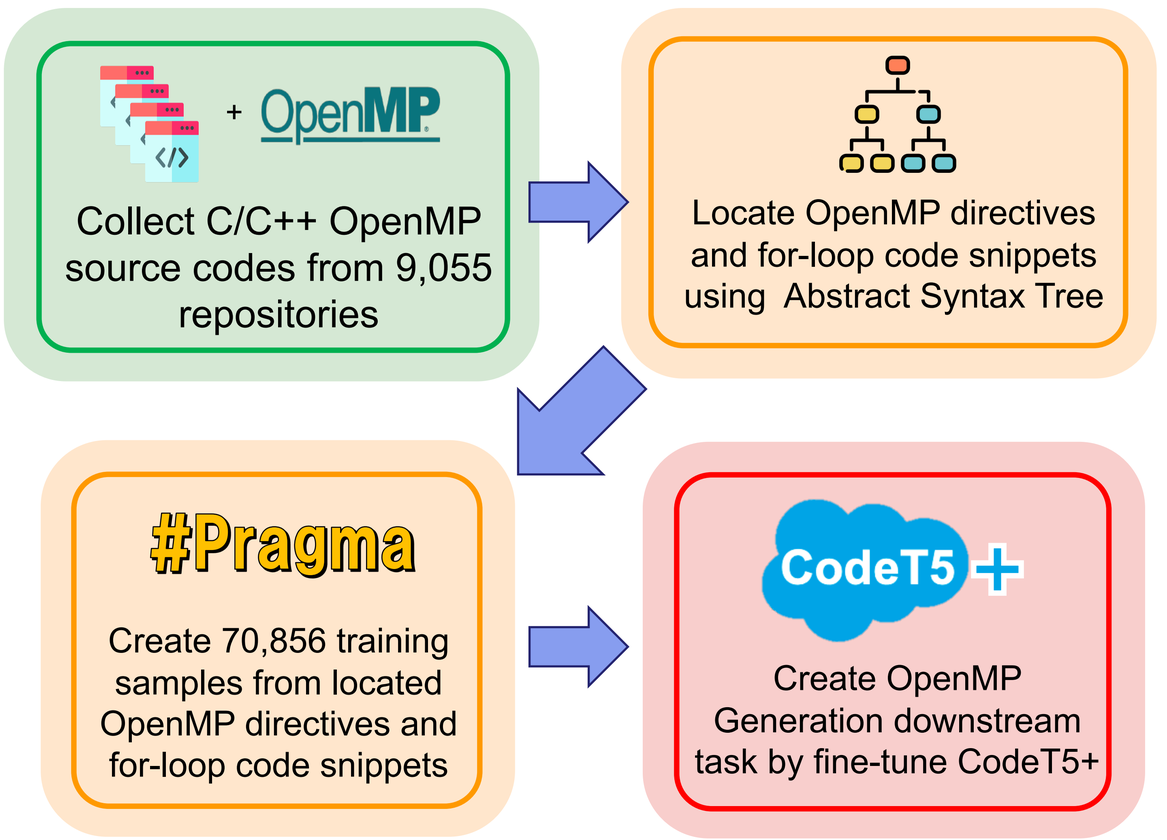
OpenMP Directive

Proposal

- Extract OpenMP directive from source code and fine-tune on a state-of-the-art Large Language Model (LLM)
- Develop two models:
  - For-loop Parallelizable Classification Model
  - OpenMP Directive Generation Model
- CodeT5+, state-of-the-art LLM, is code large language models based on transformers architecture that support code understanding and generation tasks



Development of OpenMP Directive Generation Model



Performance evaluation of Generation Model

- Model Performance Results

Exact Match	ROUGE-1	ROUGE-2	ROUGE-L
0.800	0.953	0.920	0.952

Exact match and ROUGE evaluation results of generation model fine-tune at epoch 52 using 4,324 code snippets split from the collected data

- Breakdown of Correctly Generated Directive Results

Generated OpenMP Directive	Percentage
`#pragma omp parallel for`	30.193%
`#pragma omp for`	10.170%
`#pragma omp parallel for num_threads(opt.num_threads)`	6.443%
Others	53.194%

Breakdown of 3,461 OpenMP directives from the exact match generated OpenMP directive in percentage

- An Example of Generated Result

Original directive

```
#pragma omp parallel for num_threads(USE_THREADS_NUM)
```

Generated result

```
#pragma omp parallel for num_threads(NTHREADS)
```

Our model successfully generated directives with 80% accuracy, and the rest of the misgenerated directives still have high similarity to the correct directives as shown by the ROUGE evaluation. However, the majority of correctly generated directives are simple OpenMP directives, such as, “`#pragma omp parallel for`” and “`#pragma omp for`”. We plan to further train and refine the model to effectively handle more complex OpenMP directives.