In-situ analysis with OpenMP task for leveraging unused core

Tatsuya Akazawa, Toshihiro Hanawa, and Yohei Miki, The University of Tokyo

Abstract

- We consider a framework “UTHelper” using unused cores.
- Parallelize the main computation and analysis using OpenMP task pragma.
- Compared with sequential processing, the computation time was reduced.

Introduction

- CPU in HPC clusters have cores that are not fully utilized.
- Allocate support processing to unused cores.

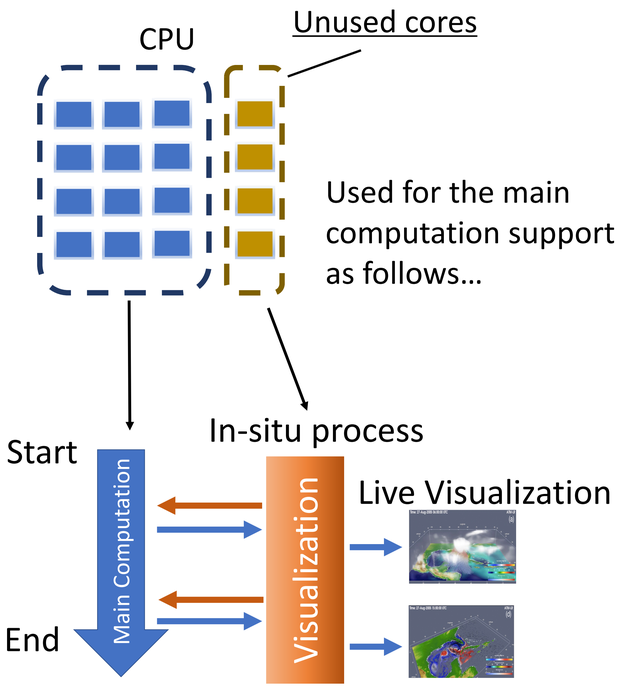
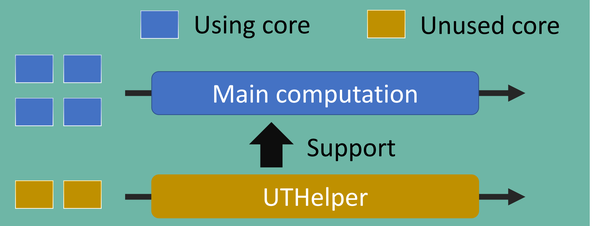


Figure 1: In-situ Visualization image.

User can
- Avoid I/O bottleneck
- Analyze all data
- Change perspective

Proposal

Our goal is to realize a user-friendly main computation support framework “UTHelper”.



- We selected the OpenMP task to parallelize the processing for the main computation support。

Functions of the UTHelper are… (example)
① Real-time performance profiling
② In-situ visualization
③ Automatic adjustment of the number of parallelism

Experiment

①Parallelization of main computation and analysis using omp task.
The code to which we have applied parallelization is Gravitational Oct-Tree code accelerated by HIerarchicaltime step Contolling code cite (GOTHIC)[1].
②Comparison of execution time between sequential processing and parallel processing.

Results

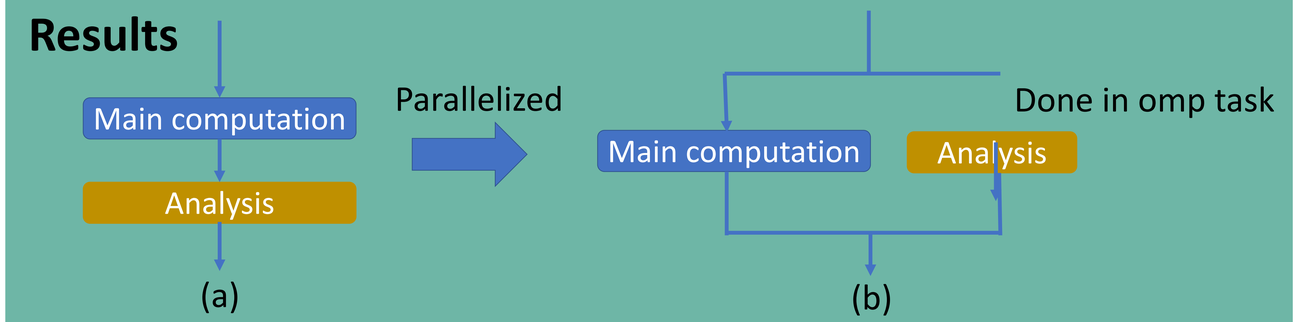


Figure 2: (a) shows the original sequential processing, (b) shows the parallelized image with omp task.

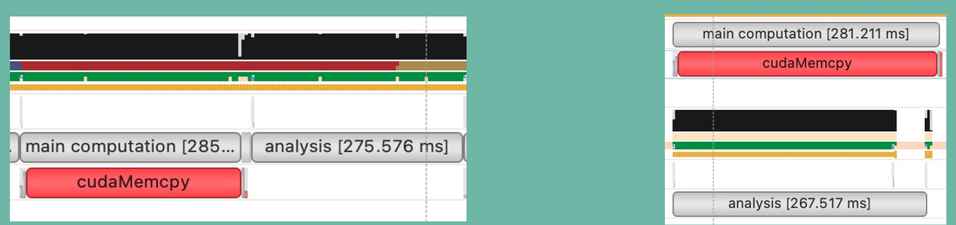


Figure 3: The execution of the program traced by NVIDIA‘s Nsight System. While (a) is the original and is a sequential process, parallel execution can be read from (b).

Table 1: Experiment Environment

System	Wisteria/BDEC-01
CPU	Intel Xeon Platinum 8360Y
GPU	NVIDIA A100
Compiler	gcc-8.3.1

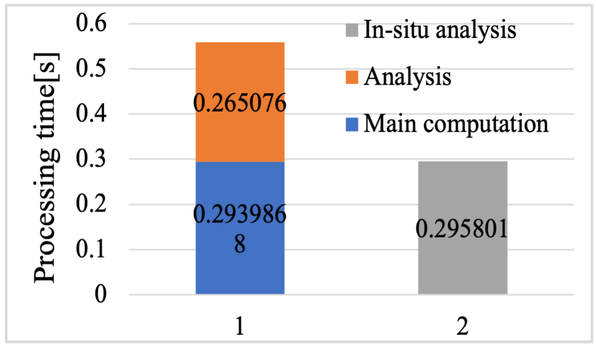


Figure 4: Execution Time Comparison between Sequential Processing and In-situ analysis.

- The execution time was reduced to about 52%.

Conclusion and Future Work

- We parallelized the main computation and analysis and evaluated the execution time.
- As the future work, we would like to study issues such as how to wait for threads that do not overload the CPU.

References

[1] Yohei Miki and Masayuki Uemura. 2017. GOTHIC: Gravitational oct-tree code accelerated by hierarchical time step controlling.
[2] Tianya Wu. Dynamic Performance Profiling for Leveraging “Unused Core”. Poster presented at: HPC Asia 2022.
[3] Ufuk Utku Turunco. Toward modular in situvisualization in Earth system models:the regional modelingsystem RegESM 1

This work was partially supported by **JSPS** KAKENHI Grant Number JP20H00580.