



IC設計所

Institute of Integrated Circuit Design



SUSTAINABLE DEVELOPMENT GOALS

Data Analysis and Optimization of the GROMACS PME

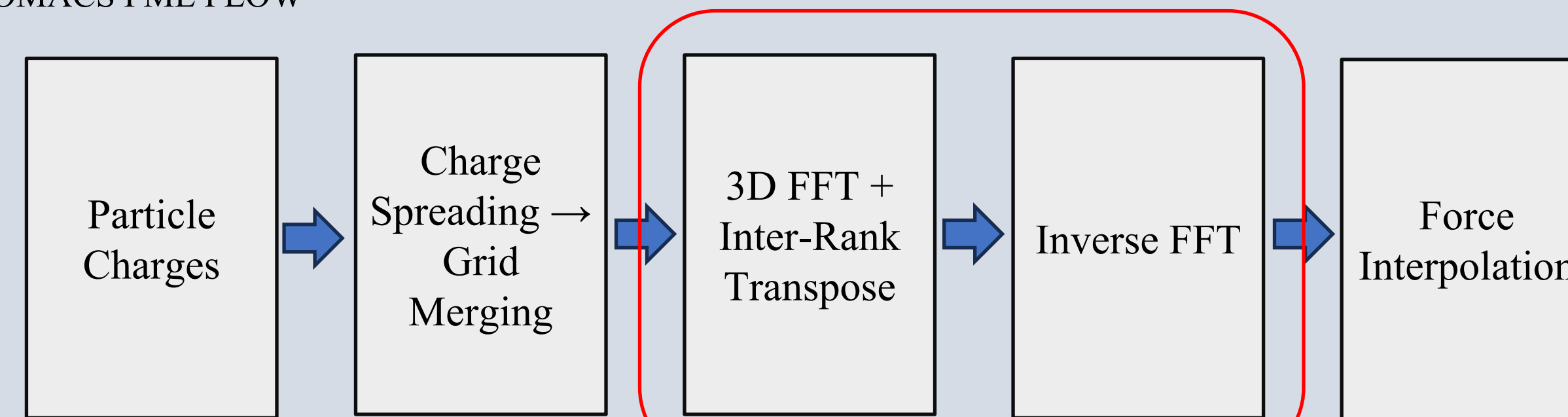
Student: 陳柏太

Adviser: Aman Sinha 辛艾曼

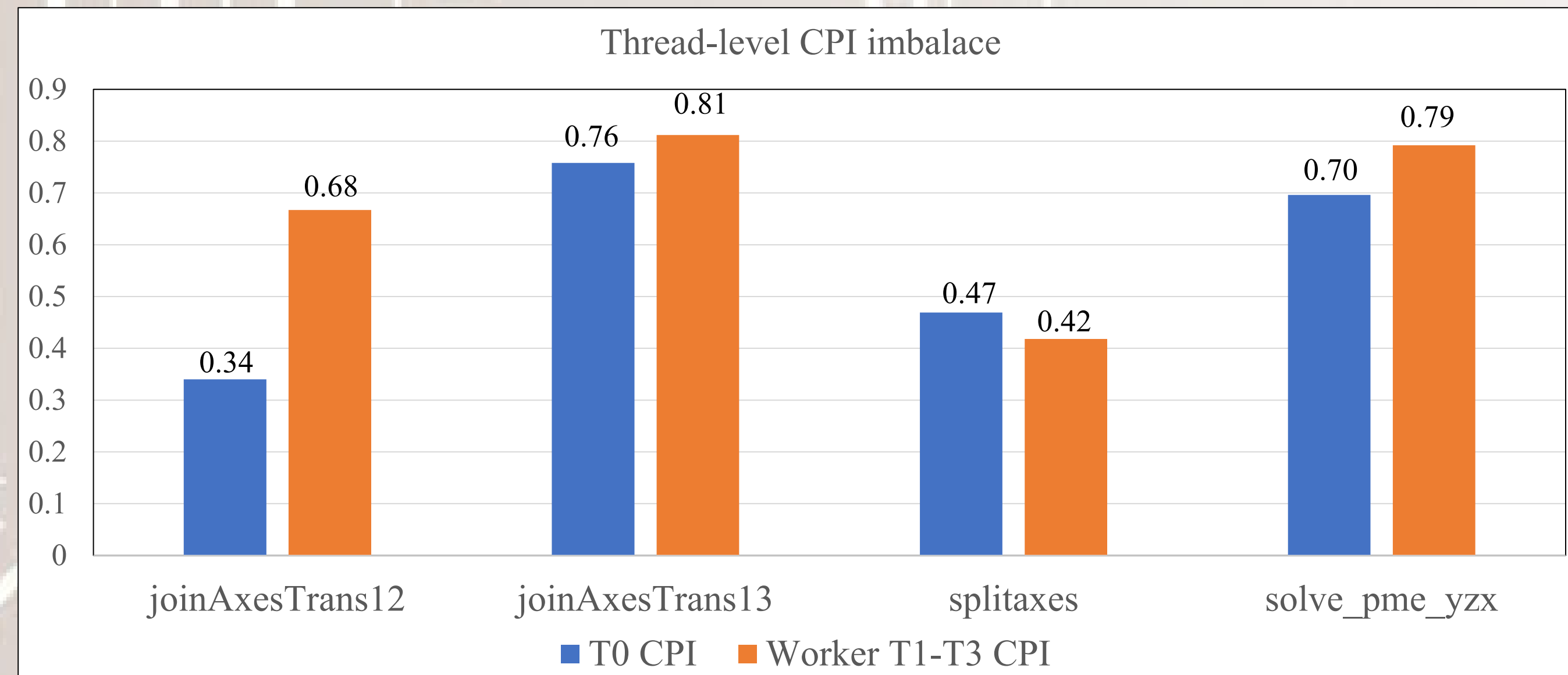
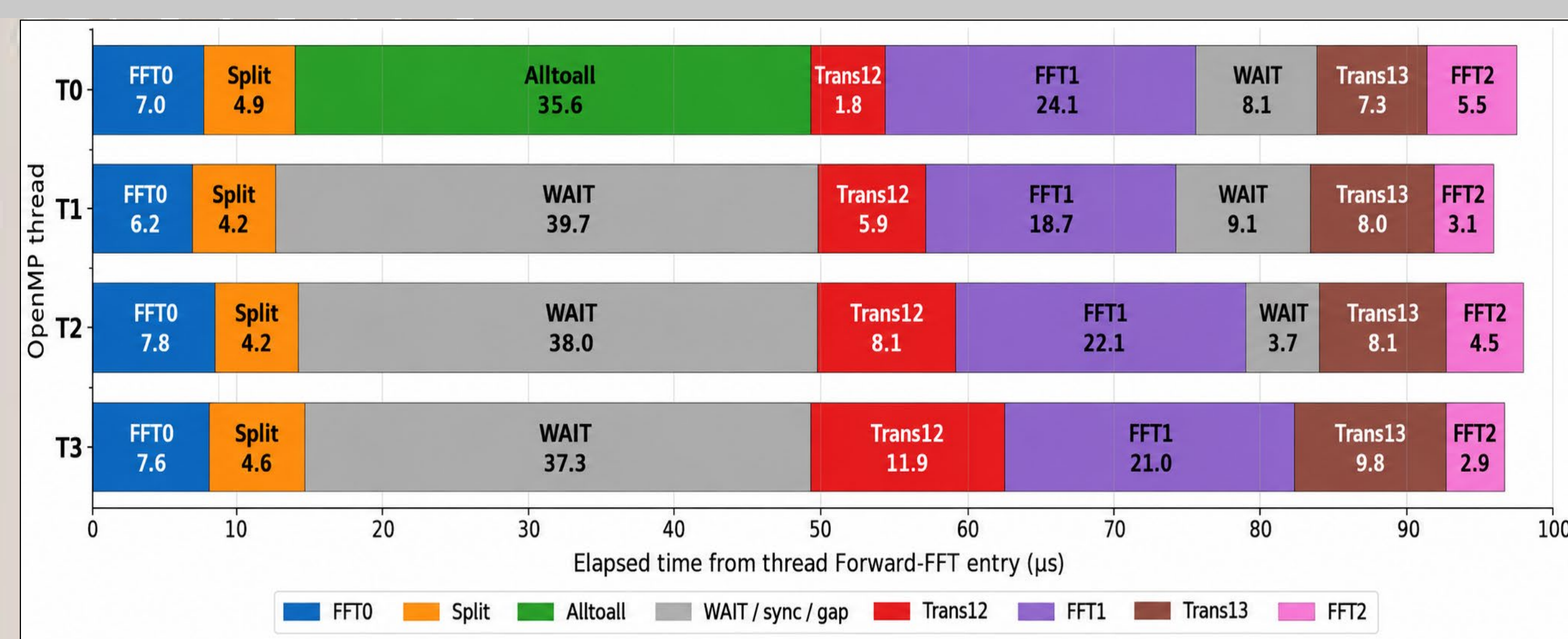
Introduction

Molecular Dynamics (MD) simulations are computationally intensive. In GROMACS, the Particle Mesh Ewald (PME) algorithm introduces **significant memory and communication overhead**. This project analyzes PME performance under **multi-rank and multi-thread execution**, identifies key bottlenecks through profiling, and explores optimizations to reduce data movement and execution time, with potential for future hardware acceleration.

GROMACS PME FLOW



Analysis



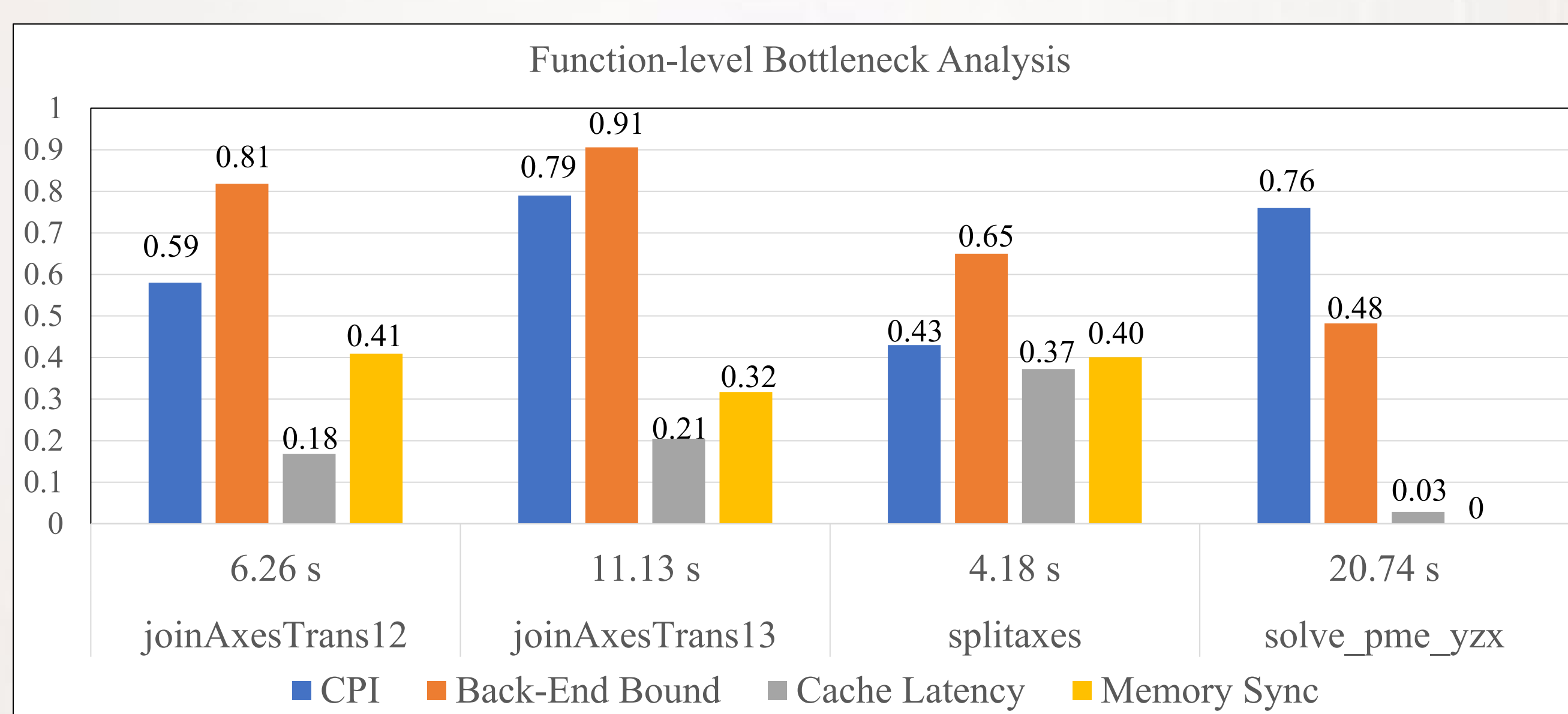
- Compare the performance of different functions in T0 and Worker.
- joinAxesTrans12 exhibits the strongest thread-level imbalance, with worker-thread CPI nearly **2× higher than T0**.
- In contrast, joinAxesTrans13, splitaxes, and solve_pme_yzx show much smaller differences between T0 and worker threads.

Python-based instrumentation was used to profile the execution time of PME FFT-to-complex FFT functions settings:

- 4-rank × 4-thread CPU
- Grid size: 80 x 96 x 112

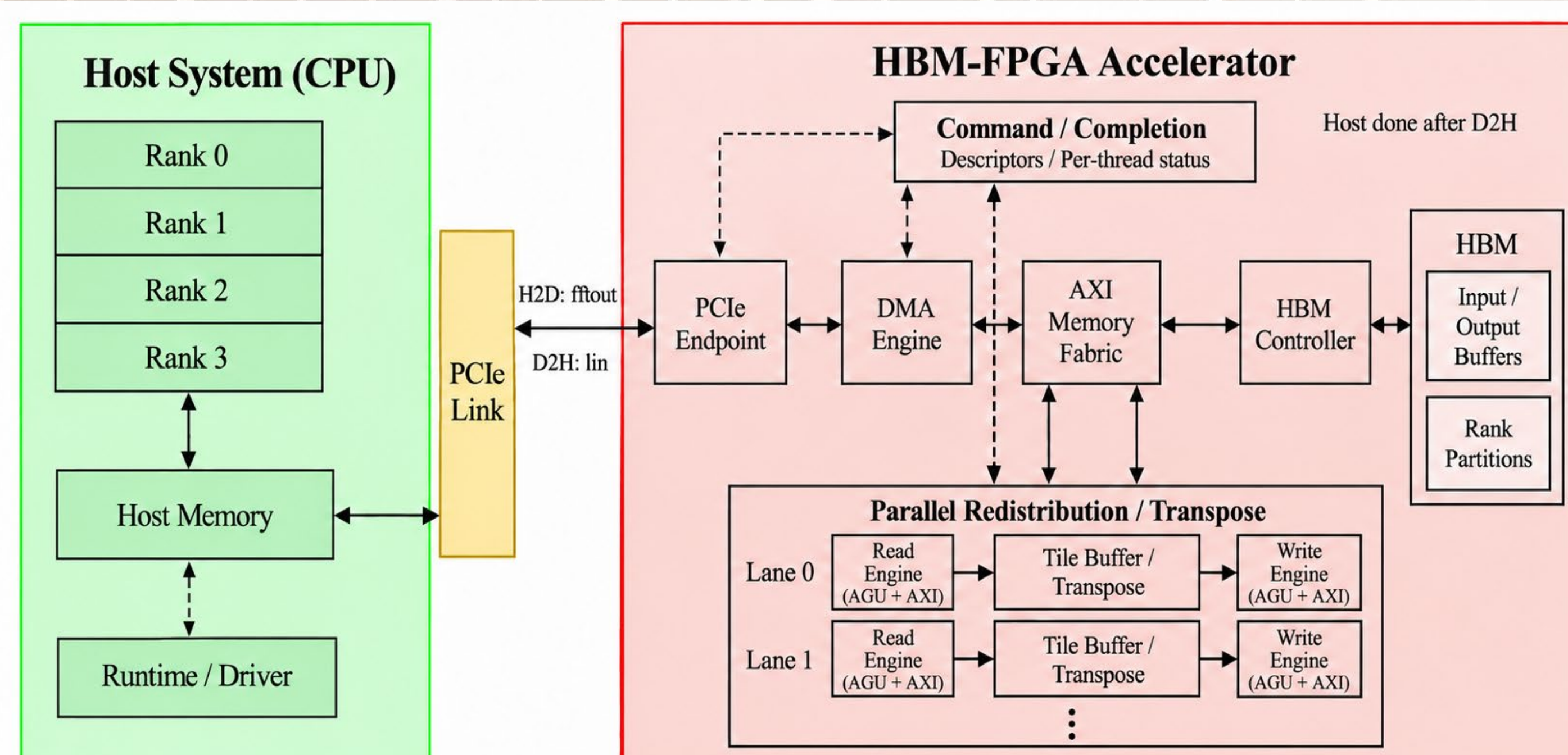
The serious issues:

- Post-communication imbalance** is observed in Trans12.
- AlltoAll approaches result in **waiting times and subsequent memory issues**.



- joinAxesTrans12 and joinAxesTrans13 are strongly back-end bound by **data movement and memory-related stalls**.
- splitaxes also shows that data rearrangement is affected by memory behavior.
- solve_pme_yzx has the largest CPU time, but it is unrelated to memory; we can spend more cost to optimize.

Proposal Optimization



Using the HBM-FPGA to improve the memory problem.

- FPGA**: Receive and buffer data in HBM, use parallel transfer and data mapping, and then **return reordered data** via PCIe DMA and report completion.
- CPU**: Submit **FFT task descriptors** and PCIe DMA requests; resume the next FFT stage when its input is ready and output is valid.

The objective is to **reduce end-to-end PME execution time and thread idle time** by fusing rank redistribution with local transposition. We also can try to move the computation workload to FPGA, so that can decrease data traffic and use the advantage of FPGA.