# A Decomposition Method With Minimal Communication Volume for Parallelization of Multi-dimensional FFTs

Truong Vinh Truong Duy[1,2] and Taisuke Ozaki[1]
[1]Research Center for Simulation Science, Japan Advanced Institute of Science and Technology (JAIST)
[2]Institute for Solid State Physics, The University of Tokyo
Email: duytvt@{jaist.ac.jp,issp.u-tokyo.ac.jp}, t-ozaki@jaist.ac.jp

## Motivation

**Fast Fourier Transform (FFT)**

An essential kernel in science and engineering.

**Parallelization of FFT**

- Existing domain decomposition methods pre-define the dimensions of decomposition and therefore are not adaptive.
- The order of data transpose may have an impact on the volume of communication.
- Little work has explored beyond 3-D FFTs, while 4-D and 5-D FFTs also have various applications.

## Purpose

**Purpose**

Develop a domain decomposition method with minimal volume of communication for the parallelization of multi-dimensional FFTs.

**Objectives**

- Minimal volume of communication.
- Applicable to 3-D, 4-D, 5-D FFTs, and beyond.
- Adaptively decompose in the lowest dimensions depending on the number of processes.
- Follow the most communication-efficient order.
- Able to work with an arbitrary number of processes.

## Method

### 1. Adaptive decomposition

- Translate the multi-dimensional data into one-dimensional data, and divide the resultant one-dimensional data equally to the processes using a block distribution.
- Treat the dimensions in a specific order: *abc*, *cba*, *bca*, etc.
- Decompose in the lowest possible dimensions depending on the number of processes.

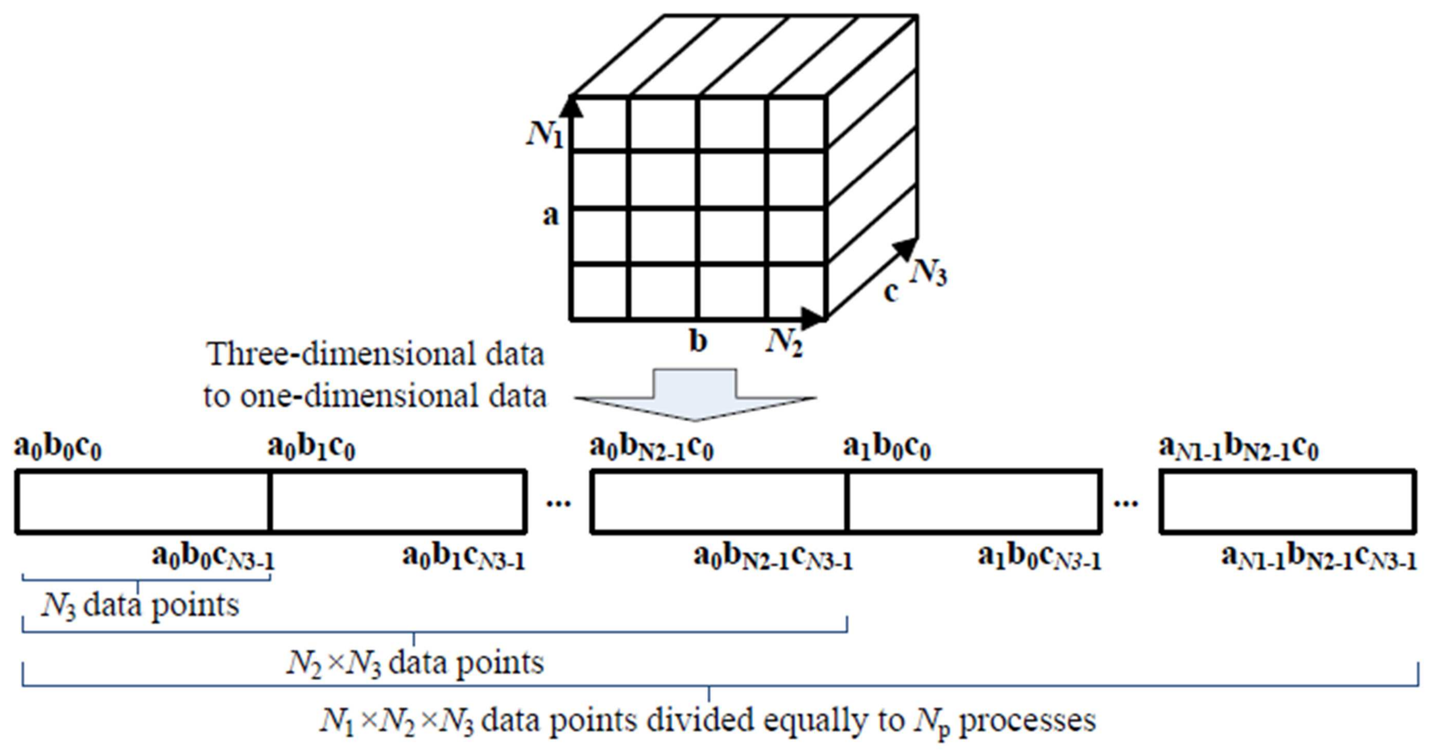
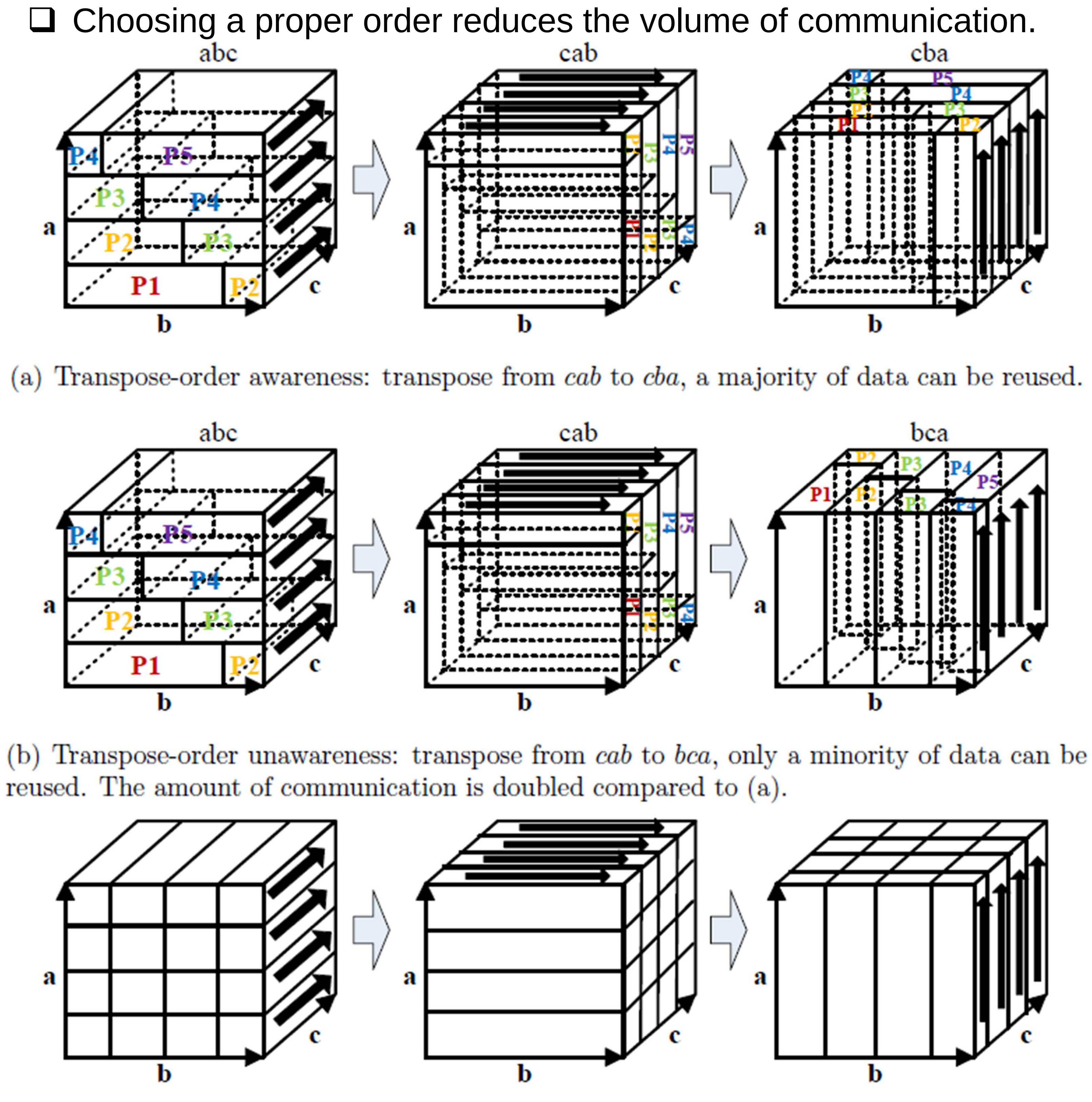



**Fig. 1: 3-D FFTs: 3-D to 1-D mapping in row-wise decomposition for the *abc* order.**

### 2. Transpose-order awareness

- The adaptive decomposition provides plenty of transpose orders.
- Different order results in different volume of communication.
- Choosing a proper order reduces the volume of communication.



(a) Transpose-order awareness: transpose from *cab* to *cba*, a majority of data can be reused.

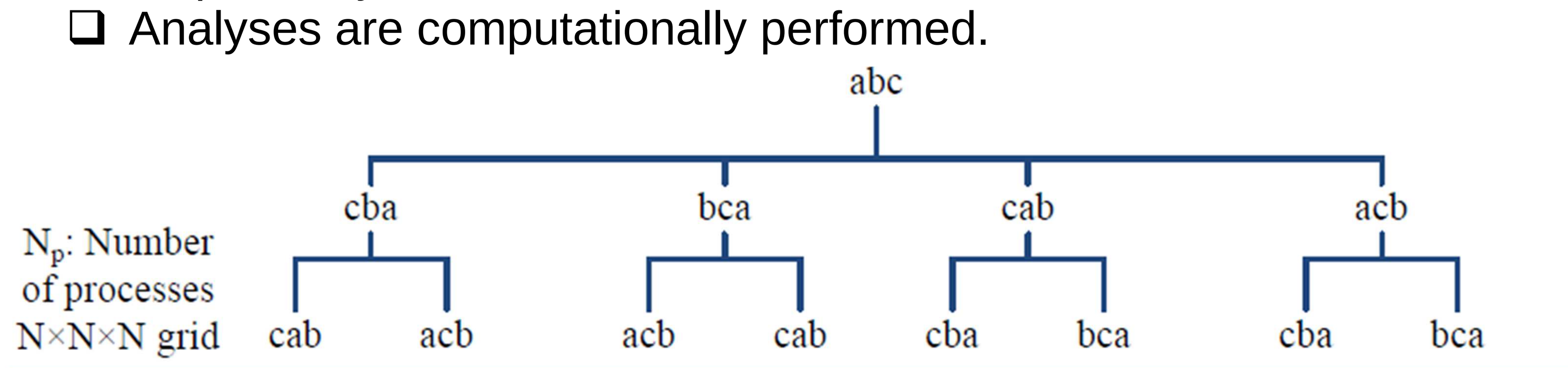
(b) Transpose-order unawareness: transpose from *cab* to *bca*, only a minority of data can be reused. The amount of communication is doubled compared to (a).

(c) Conventional 2-D decomposition: all dimensions are the same.

**Fig. 2: 3-D FFTs with 2-D decomposition: transpose-order awareness.**

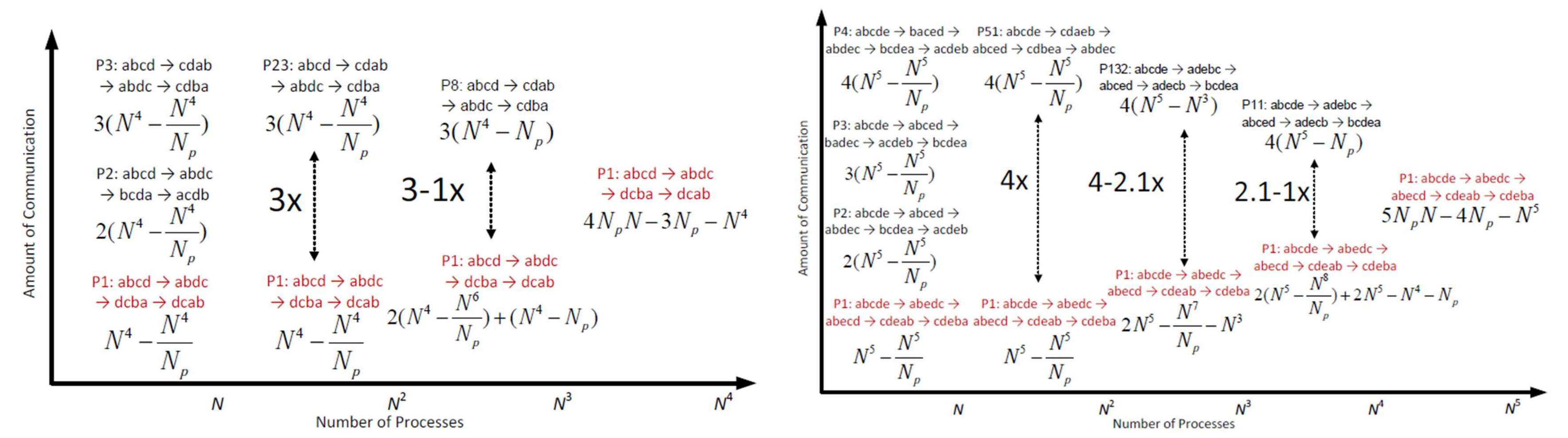
### 3. Transpose order and volume of communication

- $(M\text{-}1)!^M$ transpose orders for *M*-dimensional FFTs.
- 8, 1296, and 7962624 transpose orders for 3-D, 4-D, and 5-D FFTs, respectively.
- Analyses are computationally performed.




| N×N×N grid | cab | acb | acb | cab | cba | bca | cba | bca |
|---|---|---|---|---|---|---|---|---|
| $N_p \le N$ | $N^3-N^3/N_p$ | $2(N^3-N^3/N_p)$ | $2(N^3-N^3/N_p)$ | $2(N^3-N^3/N_p)$ | $N^3-N^3/N_p$ | $2(N^3-N^3/N_p)$ | $N^3-N^3/N_p$ | $N^3-N^3/N_p$ |
| $N < N_p < N^2$ | $(2-N/N_p)N^3-N^2$ | $2N^2(N-1)$ | $2N^2(N-1)$ | $2N^2(N-1)$ | $(2-N/N_p)N^3-N^2$ | $2N^2(N-1)$ | $(2-N/N_p)N^3-N^2$ | $(2-N/N_p)N^3-N^2$ |
| $N_p = N^2$ | $2N^2(N-1)$ | $2N^2(N-1)$ | $2N^2(N-1)$ | $2N^2(N-1)$ | $2N^2(N-1)$ | $2N^2(N-1)$ | $2N^2(N-1)$ | $2N^2(N-1)$ |
| $N^2 < N_p < N^3$ | $3N_pN-2N_p-N^3$ | $3N_pN-2N_p-N^3$ | $3N_pN-2N_p-N^3$ | $3N_pN-2N_p-N^3$ | $3N_pN-2N_p-N^3$ | $3N_pN-2N_p-N^3$ | $3N_pN-2N_p-N^3$ | $3N_pN-2N_p-N^3$ |
| $N_p = N^3$ | $3N^3(N-1)$ | $3N^3(N-1)$ | $3N^3(N-1)$ | $3N^3(N-1)$ | $3N^3(N-1)$ | $3N^3(N-1)$ | $3N^3(N-1)$ | $3N^3(N-1)$ |

**Fig. 3: Transpose order and volume of communication for 3-D FFTs.**



(a) 4-D FFTs: the patterns, the amount of communication corresponding to the number of processes, and the difference between their amount.

(b) 5-D FFTs: the patterns, the amount of communication corresponding to the number of processes, and the difference between their amount.

**Fig. 4: Transpose order and volume of communication for 4-D FFTs (a) and 5-D FFTs (b).**

## Evaluation

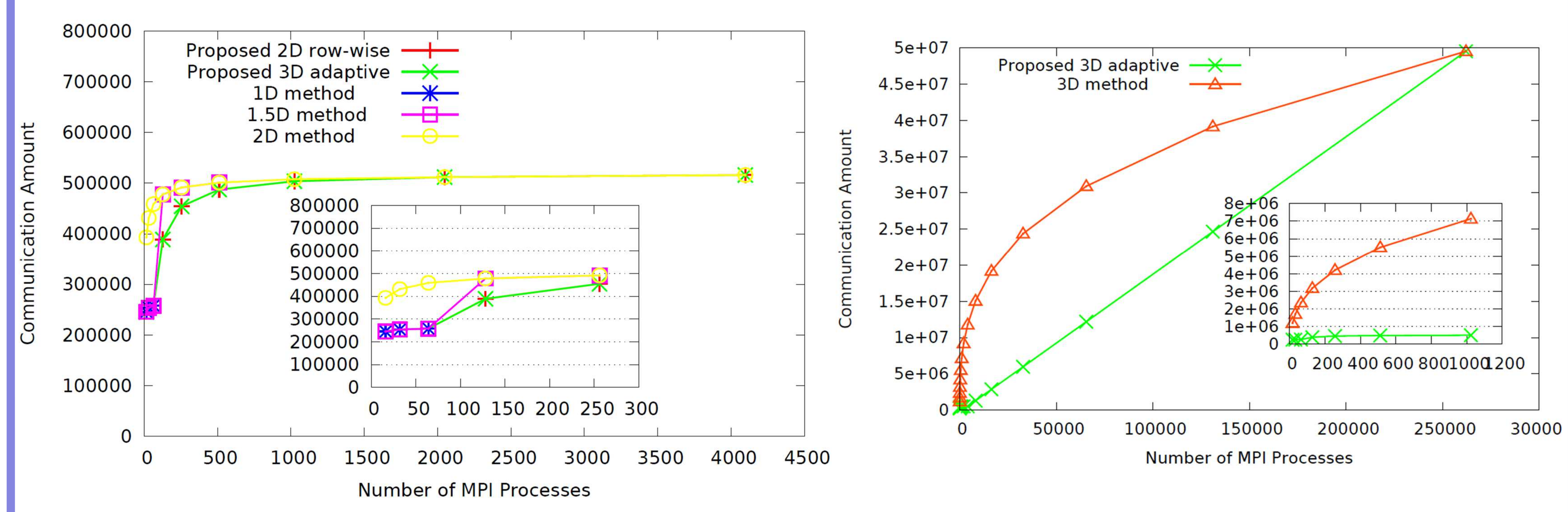
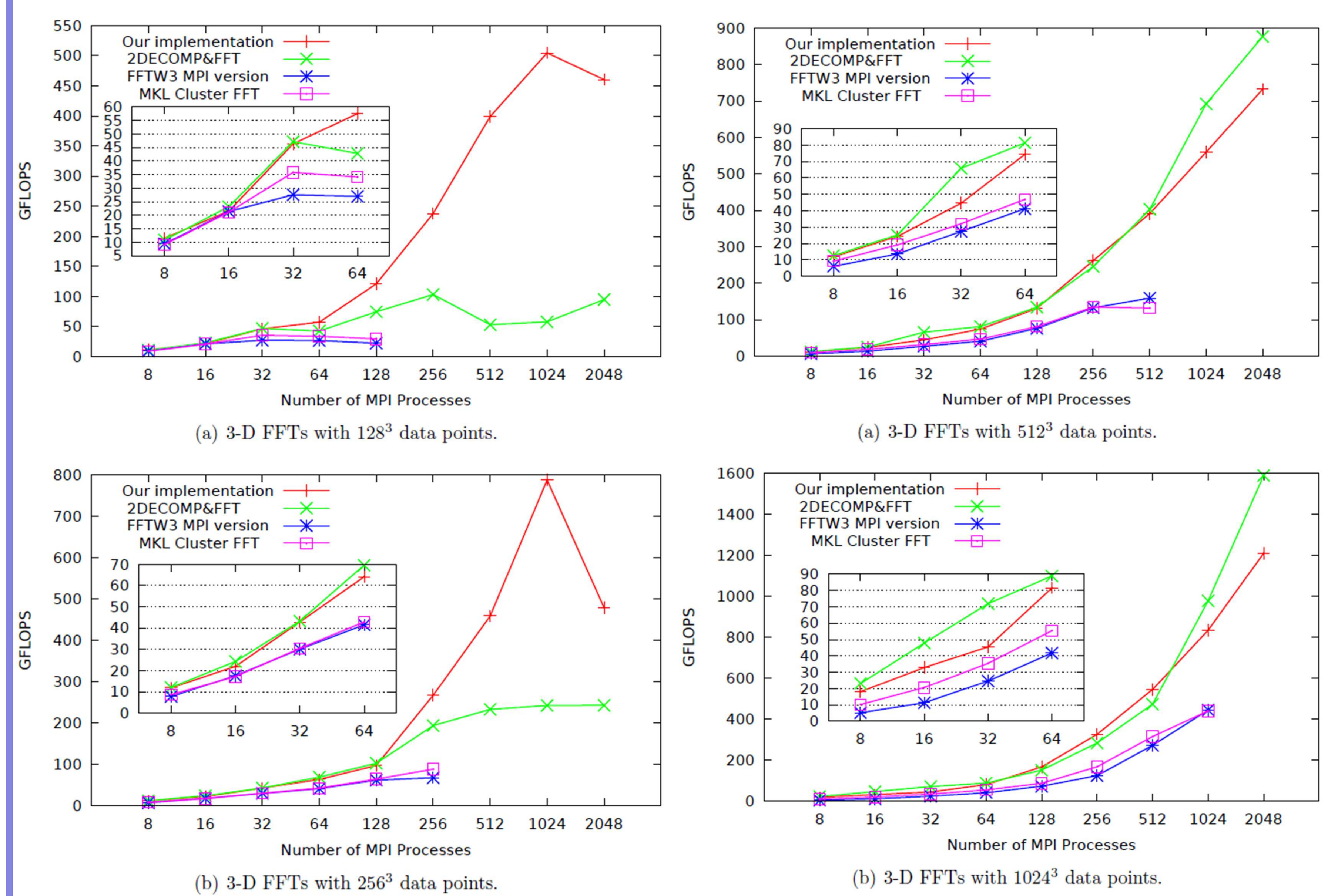



**Fig. 5: Theoretical comparison in terms of the volume of communication for 3-D FFTs.**




(a) 3-D FFTs with $128^3$ data points.

(a) 3-D FFTs with $512^3$ data points.

(b) 3-D FFTs with $256^3$ data points.

(b) 3-D FFTs with $1024^3$ data points.

**Fig. 6 : Numerical comparison in terms of GFLOPS for 3-D FFTs.**

## Summary

**Our method**

- Adaptive decomposition + Transpose-order awareness.
- Decompose in the lowest dimensions, and follow the most communication-efficient transpose orders.
- Numerical results show good performance and scaling property.

**Future work**

- Improve memory usage and communication implementation.
- Extend our implementation to *M*-D FFTs.

**Acknowledgements**

CMSI and Materials Design through Computics, MEXT.

**References**

[1] T.V.T. Duy and T. Ozaki, arXiv:1302.6189 (2013).
[2] T.V.T. Duy and T. Ozaki, arXiv:1209.4506 (2012).