

# CoToCoA: A Communication-Efficient Framework for Coupling Programs

Takeshi Nanri  
nanri.takeshi.995@m.kyushu-u.ac.jp  
Kyushu University  
Fukuoka, Japan

Yuto Katoh  
yuto.katoh@tohoku.ac.jp  
Tohoku University  
Sendai, Japan

Keiichiro Fukazawa  
fukazawa@media.kyoto-u.ac.jp  
Kyoto University  
Kyoto, Japan

Yohei Miyake  
y-miyake@eagle.kobe-u.ac.jp  
Kobe University  
Kobe, Japan

Kazuya Nakazawa  
nakazawa.kazuya@stu.kobe-u.ac.jp  
Kobe University  
Kobe, Japan

## 1 INTRODUCTION

To construct complex software systems by coupling existing programs, there have been developed several tools. For example, preCICE [2] provides interfaces for partitioned multi-physics simulations, while VisIT [1] enables connection of its viewer program to the existing simulations to enable in-situ visualization. As more efficient framework, this poster introduces CoToCoA (Code-To-Code Adapter) [3] which leverages high-performance interconnects by utilizing MPI as the basement layer, supports asynchronous data transfer among programs and enables coupling with minimum modification to existing programs.

## 2 OVERVIEW

Figure 1 shows the control flows in CoToCoA. The control flows from a program, called the requester, to programs, called the workers. Each flow delivers a request which is forwarded by another program, called the coupler. The worker to handle a request is chosen by the coupler according to the data from the requester.

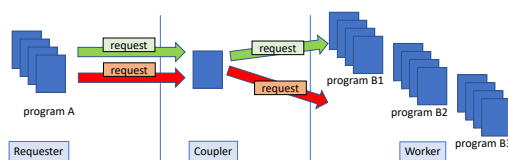


Figure 1: Control flow of CoToCoA

In CoToCoA, data can be transferred from the requester to the coupler or the workers by using following methods:

- Integer parameters within a request.
- An array of integers or real numbers attached to a request.
- Remote read from exposed memory regions of the requester.

Also the exposed memory regions can be used as a place to feedback through remote write from the coupler or the worker.

## 3 USAGE

Programs for the requester and the worker can be prepared by applying minimal modifications to the existing programs, such as:

- Replace `MPI_Init` and `MPI_Finalize` to the corresponding routines of CoToCoA.

- Replace `MPI_COMM_WORLD` to `CTCA_Subcomm`.
- In the requester program, call routines to send requests.
- In the worker programs, construct a main loop that repeats accepting a request and the related data so that they can perform their computation on the data.

The coupler program should be newly created or copied from the previous project. It should repeat accepting requests from the requester and forward each of them according its related data.

The execution of multiple programs of CoToCoA relies on the facility of MPMD (Multiple Program Multiple Data) of MPI libraries. This facility is available on most of the existing MPI libraries such as Open MPI, MVAPICH2, MPICH, Intel MPI and Fujitsu MPI.

## 4 PERFORMANCE EVALUATION

Figure 2 shows the comparison of performance between CoToCoA and MPI on ITO, the supercomputer of Kyushu University, where the number of processes are four for the requester, one for the coupler and six for the worker. Since the worker reads data directly from the requester in "CoToCoA read" it is faster than "CoToCoA attach" in which data is transferred with the request. Also, both performance is sufficiently close to the result with MPI.

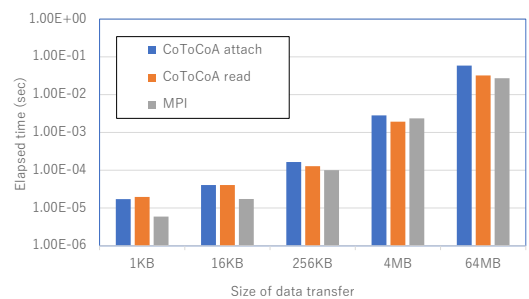


Figure 2: Time for data transfer

## REFERENCES

- [1] Hank Childs. 2012. VisIt: An end-user tool for visualizing and analyzing very large data. (2012).
- [2] Gerasimos Chourdakis. 2020. An introduction to the preCICE coupling library. *ECCOMAS WCCM 2020* (2020).
- [3] Takeshi Nanri, Yuto Kato, Keiichiro Fukazawa, Yohei Miyake, Kazuya Nakazawa, Hingde Zhou, and Youhei Sunada. 2021. *CoToCoA (Code-To-Code Adapter)*. doi://10.5281/zenodo.5655840.