RDMA with Double Buffering for Adjacent Communication

Kota Yoshimoto, Akihiro Fujii, Teruo Tanaka
Kogakuin University

Introduction

In large-scale scientific computing programs, parallelization by MPI communication is generally used. MPI is convenient because it can be executed on many computers. However, interprocess communication often becomes a bottleneck in highly parallel computers [1].
In this research, we measured and compared the performance of MPI, RDMA, and RDMA with double buffering for adjacent communication.

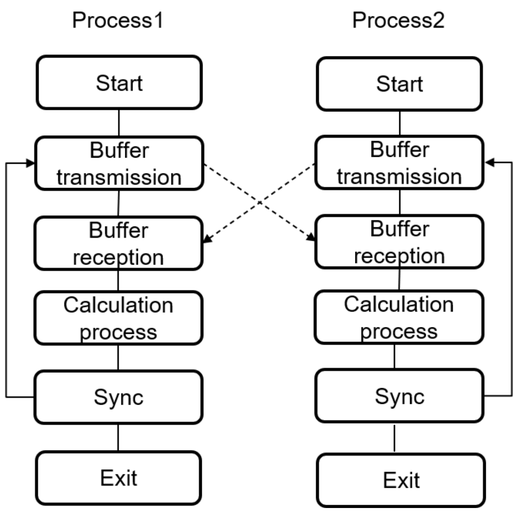
Adjacent Communication

➢ MPI_Isend/Irecv

- It is a communication method generally used for adjacent communication of MPI.
- MPI_Isend / Irecv is two-way communication.

➢ RDMA

- RDMA communication can read and write data without the intervention of the program of the destination node by using a dedicated memory.
- RDMA is one-sided communication.



In RDMA communication, it is necessary to synchronize so that the data will not be overwritten while it is being used.

Fig.1 RDMA

➢ RDMA with Double Buffering

- It does not require synchronization by using two buffers for communication.

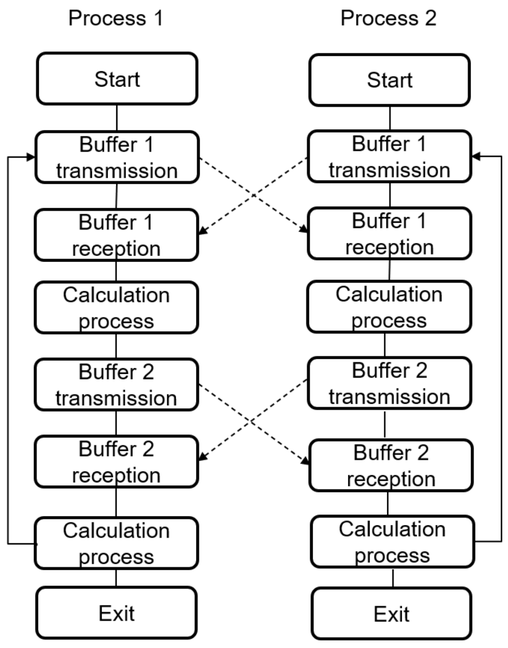
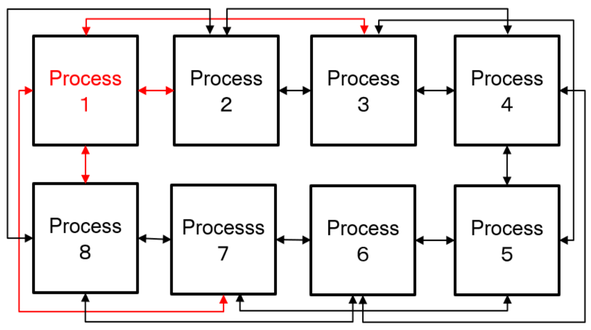


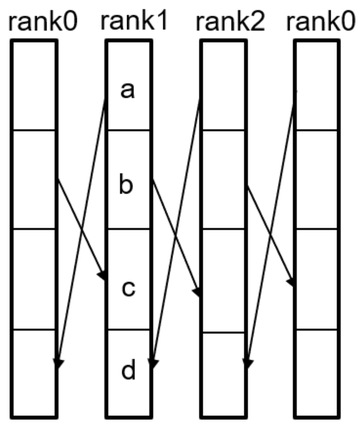
Fig.2 RDMA with Double Buffering:
When Buffer1 reception is not finish, Calculation process cannot start. Therefore, communication timing gap will less than 2.

➢ Communication setting



Process1 communicates with process 7, process 8, process 2, and process 3.

Fig.3 Adjacent Communication with 4 neighboring process case.



Data to be sent is stored in a and b. Received data is stored in c and d.

Fig.4 Vector image on processes from rank0 to rank2

Numerical experiment

In this experiment, we used a supercomputer Flow at Nagoya University. It is equipped with a system called FX1000 [2].

➢ Fig.5 shows communication time when the total number of processes is changed.
➢ Fig.6 shows communication time and improvement rate when the number of adjacent processes is changed.

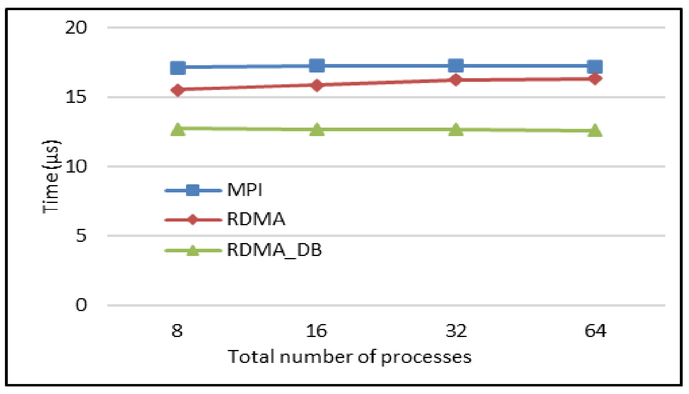


Fig.5 Communication time of adjacent communication when the total number of processes is changed by adjacent communication:
Experiment Condition

- The number of adjacent processes is 2
- Double precision data is 160.

The communication time does not change much when the total number of processes is changed. This is because the amount of communication per process does not change even if the total number of processes is increased.

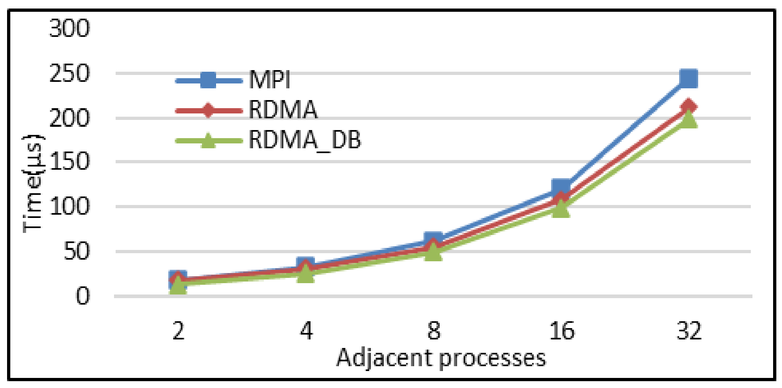


Fig.6.1 Communication time of adjacent communication when the number of adjacent communication adjacent processes is changed.

Experiment Condition

- The number of processes is 128.
- Double precision data is 160.

The communication time increases proportionally when the number of adjacent processes is changed. This is because changing the number of adjacent processes increases the amount of communication per process.

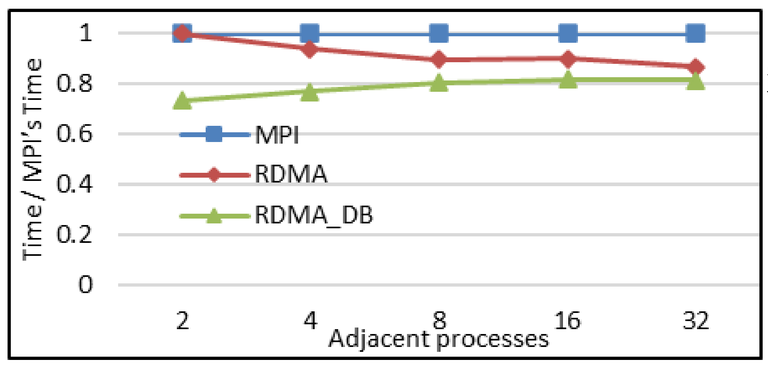


Fig.6.2 Communication time ratio when the MPI time is 1 when the number of adjacent processes is changed in adjacent communication.

RDMA with double buffering was confirmed to improve adjacent communication time by up to 30% compared to MPI.

Conclusion

MPI, RDMA, and RDMA with double buffering were evaluated using simple adjacent communication with various number of neighboring processes and different total process numbers. It showed that RDMA was faster than MPI as the number of adjacent processes increased. In addition, RDMA with double buffering reduced synchronization and was the most efficient communication method of the three by up to 30%.

Acknowledgments

This work is supported by “Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures” and “High Performance Computing Infrastructure” in Japan (Project ID: jh210026-NAH).

Reference

[1] Kanamori Issaku, et al., Acceleration of communication with low latency uTofu interface in LQCD application, IPSJ SIGHPC report, Vol.2020-HPC-177 No.22 (2020).
[2] Supercomputer “Flow Type I subsystem”, Nagoya Univerisyty. <https://icts.nagoya-u.ac.jp/ja/sc/>