RDMA with Double Buffering for Adjacent Communication

Kota Yoshimoto
Kogakuin University
Japan
k221013@ns.kogakuin.ac.jp

Akihiro Fujii
Kogakuin University
Japan

Teruo Tanaka
Kogakuin University
Japan

1. 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]. There is an interface called RDMA (Remote Direct Memory Access) to reduce the delay caused by communication.

In this research, we measured and compared the performance of MPI, RDMA, and RDMA with double buffering for adjacent communication.

2. RDMA with Double Buffering

RDMA communication can read and write data without the intervention of the program of the destination node by using a dedicated memory, and it is possible to communicate at a higher speed than that by MPI.

In RDMA with double buffering, two buffers for communication are prepared, and when the data in the buffer is being read, the other one communicates. The destination node does not send the next data until the data has been read. In addition, the source node waits for the data sent by the destination node to be received. There is no delay in sending and receiving data more than once, and data is not overwritten during by using double buffers. Thus, it can operate one-sided communication without barrier instruction. Figure 1. shows a flowchart of the operation image.

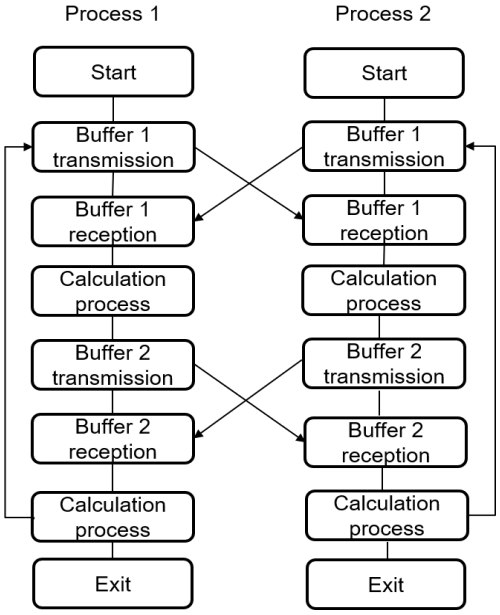

Figure 1. Double buffering

3. Numerical experiment

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

In this experiment, MPI, RDMA, and RDMA with double buffering were evaluated by adjacent communication, respectively. Its condition is the total number of processes was fixed at 128 and the number of adjacent processes was changed from 2 to 32.

Figure 2 shows the case where the number of adjacent processes is changed by adjacent communication. In Figure 2, the vertical axis the communication time ratio, when MPI' s time is fixed to 1.0 and the horizontal axis is the number of adjacent processes. As a result, 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. Also, RDMA with double buffering was confirmed to improve speed by up to 30% compared to MPI.

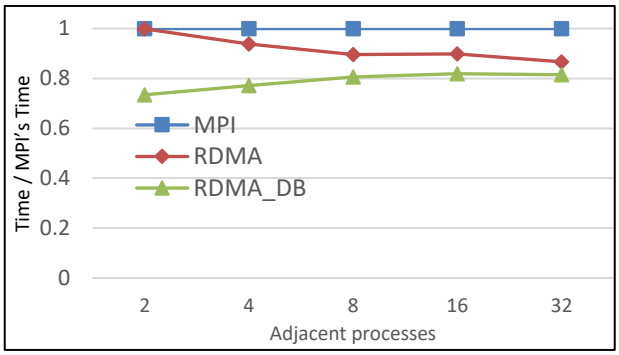

Figure 2. adjacent communication time ratio when the number of adjacent processes is changed.

4. Conclusion

MPI, RDMA and RDMA, with double buffering were evaluated using adjacent communication. 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.

Further research on the way to apply this technique to application programs using adjacent communication.

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).

REFERENCES

[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] Super computer "Flow Type I subsystem", Nagoya Univerisyty. <https://icts.nagoya-u.ac.jp/ja/sc/> (Accessed: 29 November 2021).