Implementation of neighbor communication algorithm using multi-NICs effectively by extended RDMA interface

[Extended Abstract]

Yoshiyuki Morie
Kyshu university
6-10-1, hakozaki, higashi-ku
fukuoka, Japan 812-8581
morie.yoshiyuki.404@m.kyushu-u.ac.jp

Takeshi Nanri
Kyushu university
6-10-1 hakozaki, higashi-ku
fukuoka, Japan 812-8581
nanri@cc.kyushu-u.ac.jp

ABSTRACT

In this paper, a neighbor communication algorithm for large messages in K computer was proposed. The key idea of this algorithm was to divide messages into fragments according to the number of neighboring processes and the number of NICs available on each node, so that the bandwidth of each NIC was fully used. To show the effectiveness of the proposed algorithm, an implementation of the algorithm for six neighbors with four NICs by MPI functions was examined. Experimental results showed that, for large messages, the performance of the proposed algorithm was two times faster than the existing one in mpich-3.0.4. However, in medium message size, proposed algorithm is lower performance than existing one. Therefore, the proposed algorithm is implemented by extended RDMA interfaces instead of MPI functions. This implementation became faster than existing one in medium message size too. This result showed the proposed neighbor communication was widely effective.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Network communications

General Terms

Performance

Keywords

Neighbor communication, Multi-NICs, Extended RDMA interface

1. INTRODUCTION

Neighbor communications are patterns of communications in which each process exchanges messages among limited number of neighboring processes. They are popularly used in many types of applications, including scientific simulations with domain decompositions and programs with graph-based algorithms. As common interfaces for such communications, some standard functions have been added in the draft of MPI 3.0, recently.

However, studies on the algorithms for implementing these functions are quite limited. For example, in mpich-3.0.4, the latest version of one of the most popular implementation of MPI, the algorithms simply invoke send and receive operations to the neighboring processes. This algorithm is called native algorithm. As the size of the computer systems is becoming larger, multidimensional mesh/torus is chosen instead of crossbar or fat-tree for cost-performance. On some systems with that topology, the number of NICs provided on each node is less than the number of neighboring nodes in the topology. If each node sends neighboring nodes, message can not transmitted data to all neighboring nodes in same time. This degrades the utilization rate of effective bandwidth of multi-NICs.

To achieve sufficient performance, authors proposed the neighbor communication algorithm for K computer[1]. The key idea of this algorithm is full using all NICs in a node. Preliminary experiment shows that performance of neighbor communication improves to existing algorithm in large message size by using four NICs effectively. However, in medium message size, proposed algorithm is lower performance than existing one. Therefore, the proposed algorithm is implemented by extended RDMA interfaces instead of MPI functions.

2. NEIGHBOR COMMUNICATION ALGORITHM EFFECTIVE USING OF NICS

Neighbor communication sends and receives m bytes message to/from the $2r$ neighboring process respectively in process space of r-dimension rectangular parallelepiped. It is assumed that processes allocation on network topology is in agreement in the process allocation in a program. If the number of neighboring processes and the number of NICs is in agreement, the communication bandwidth of NICs can be used fully. However, when the number of neighboring processes is larger than the number of NICs, it is possible that the bandwidth is unable to be used sufficiently. If the number of neighboring processes is $2r$ and the number of NICs is n, the communication bandwidth of all the NIC can be used up by transmitting r/n messages at each NIC. In order to use the communication bandwidth of NIC equally,

a $1/n$ message is communicated in $2r$ step. If it is possible to reduce a fraction $2r/n$ by d, it is able to send d/n messages in a block. In other words, d/n messages can be sent in $2r/d$ steps.

3. IMPLEMENTATION OF NEIGHBOR COMMUNICATION BY EXTENDED RDMA INTERFACE

In previous work of authors, proposed algorithms are implemented by MPI_Isend, MPI_Irecv and MPI_Waitall. Figure 1 shows the bandwidth of the proposed and the native algorithm in 6-neighbor and 4-NICs.

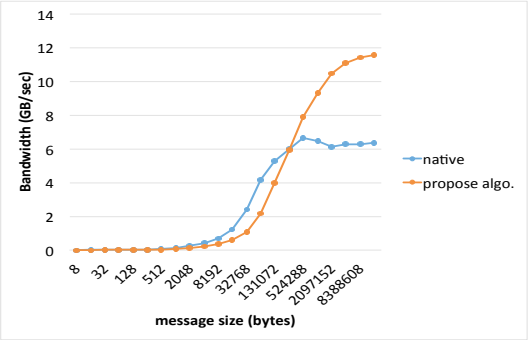

Figure 1: Communicaiton Bandwidth of both of neighbor communication

While the proposed implementation is high-speed in larger message size than 512KB, the native implementation is more high-speed in smaller message size than 256KB. In the proposed algorithm, in order to divide the message, the number of communications becomes double. Therefore, the overheads of the MPI function increase. The overheads of a MPI function are a copy of a buffer, a software overhead of a MPI function, etc. Moreover, when communicating a large message using a MPI function, RDMA is used in it. In RDMA, the exchange of a base address and the registration of a memory are performed for each communication. Therefore, this algorithm is developed by using the extended RDMA interface to reduce these overhead.

4. PRELIMINARY EVALUATION

This section shows the results of experiments that compare the performance of the proposed implementation with the native one. The platform of the experiment is Fujitsu PRIMEHPC FX10 that is family of K computer at University of Tokyo, Japan. On FX10, each node has 16 cores and 32GB of memory, and the clock-rate is 1.848GHz. OS, compiler and MPI library are proprietary, developed by Fujitsu Ltd. The network is the Tofu interconnect and communication bandwidth of each link is 5GB/sec.

Figure 2, 3 shows the communication bandwidth and communication time of each neighbor communication. "native" shows the existing implementation and "3-steps by MPI" shows the proposed algorithm by MPI function. "3-steps by RDMA" shows the proposed algorithm by extended RDMA interface. In figure 2, 3-steps algorithm by RDMA is faster than 3-steps algorithm by MPI function in all messsage size.

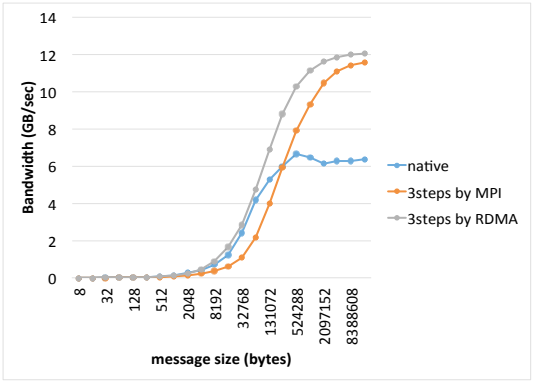

Figure 2: Communicaiton Bandwidth of each neighbor communication

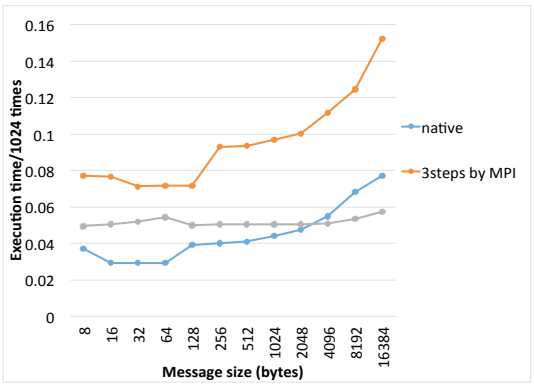

Figure 3: Communication time of each neighbor communication

Moreover, From 4KB in figure 3, 3-steps algorithm by RDMA interface is faster than native implementation.

5. CONCLUSIONS

Neighbor communication implementation using extended RDMA interfaces was proposed for FX10. The performance of this implementation improves for medium and large messages, and it was able to realize twice the performance at a maximum to the existing one.

Future work is consisting of the following. The neighbor communications of other communication pattern are developed. Furthermore, the neighbor communication algorithm when communication contentions occur is developed.

6. ACKNOWLEDGMENTS

This research was conducted using the Fujitsu PRIMEHPC FX10 System (Oakleaf-FX) in the Information Technology Center, The University of Tokyo.

7. REFERENCES

[1] Y. Morie and T. Nanri. A neighbor communication algorithm with making an effective use of nics on multidimensional-mesh/torus. *Internatinal Conference on Simulation Technology*, September 2013.