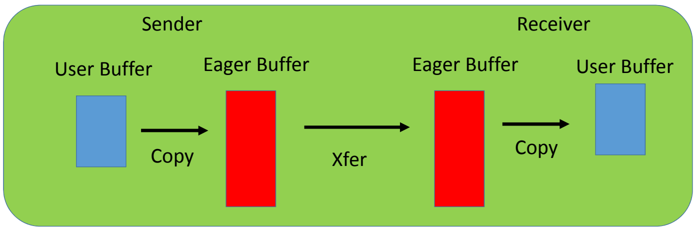
Memory Control Technique of the Point-to-Point Communication

Takeshi Soga[1,3] and Takeshi Nanri[2,3]

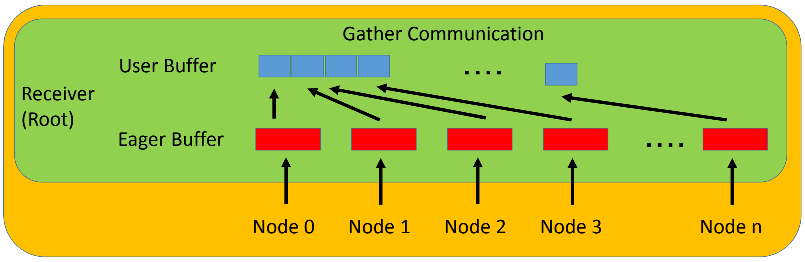
[1] ISIT Kyushu, Japan. E-mail: soga@isit.or.jp, [2] Kyushu University, Japan., [3] JST CREST

Background

・Eager communication is widely used technique on the point-to-point communication of short size (e.g. OpenMPI and MVAPICH). Eager communication transfers data between prepared buffer by communication library.
This technique can hide the synchronization time because the area of the user buffer on the sender becomes available when memory copy is finished, and it can hide the transfer latency because transfer becomes available before the area of the user buffer on the receiver can be overwritten. Moreover, this method reduces the memory registration time in the case when the RDMA facility of the underlying interconnect is used because the fixed memory area is used for transfer data instead of some user buffer areas.



・Usually, the buffer for eager communication is allocated dynamically by communication library in order to reduce initial memory consumption. In general, this does not cause severe problem when the number of processes is small. However, as the number of processes becomes large, and if each of those processes performs communications with all other processes, the total amount of the buffers will be non-negligible. Even if the communications on each pair is performed for small number of times, the buffers once allocated will be kept because the communication library does not know such situation. Therefore, when the number of nodes is large, those memory consumption in the communication library can halt the program because of the shortage of the available memory.



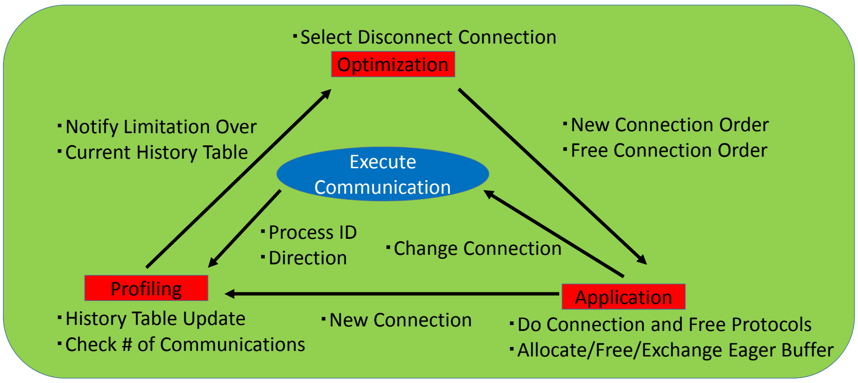
Proposal of Automatic Connection/Disconnection Framework for Controlling Memory Consumption

・We intend to implement dynamic buffer allocation and free mechanism to solve the problem mentioned above problem.
 By reducing the memory size of buffer automatically, user function do not take care of the memory consumption according to the communication library.

```
for(i=0;i<100;i++) {
  user_func();
  gather_communication();
}
```

← Automatic memory allocation and free

・We propose the automatic connection management framework in order to realize this automatic memory control. This mechanism has the online profiler and the asynchronous connection/free protocols. By these techniques, the eager buffer becomes free quickly without too much degrading performance.



Automatic connection management framework

Key Technique

Profiling

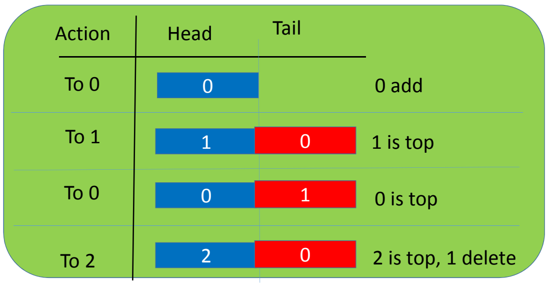
- For each communication, process IDs and direction are monitored.
- Monitored information is managed in the history table.
- Notify that the number of connection is reached to the threshold.

Optimization

- Decide the free connection communication according to the policy.
  Ex) LRU (Least Recently Used) policy: Choose connection that is oldest & satisfies the following conditions:
    Sender : There is not data transfer between eager buffers is started but not finished.
    Receiver : All requested data to sender is already coming.

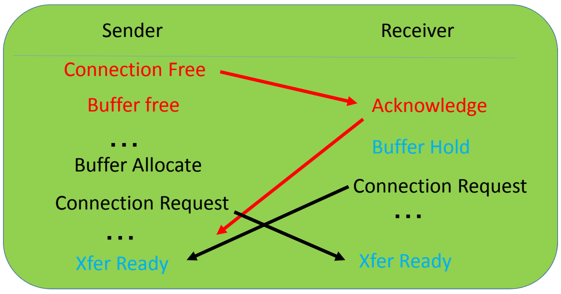
Application

- Chosen connection is asynchronously discarded at both sides.
- Actual free operation on the memory consumed for the connection is delayed to hide the overhead.



Action	Head	Tail	
To 0	0		0 add
To 1	1	0	1 is top
To 0	0	1	0 is top
To 2	2	0	2 is top, 1 delete

LRU basis P2P communication control



Asynchronous connection and free protocol
Red line : Free protocol   Black line : Connection protocol

Future Work

・We plan that this automatic connection management framework advances to the total memory control system inner communication library.
・Also, we should research some applications and advocates which are good to apply this mechanism.