ACP: Advanced Communication Primitives for Exa-scale Systems

Shinji Sumimoto[1], Yuichiro Ajima[1], Kazushige Saga[1],
Takafumi Nose[1], Naoyuki Shida[1] and Takeshi Nanri[2]

[1] Fujitsu Ltd. 4-1-1 Kamikodanaka 4-Chome, Nakahara-ku,
Kawasaki-city, Kanagawa, 211-8588, Japan.
E-mail: {sumimoto.shinji,aji,saga.kazushige, nose.takafumi, shidax}@jp.fujitsu.com
[2] Kyushu University, 6-10-1 Hakozaki Higashi-ku, Fukuoka 812-8581 Japan.
E-mail: nanri@cc.kyushu-u.ac.jp

Abstract. Current MPI communication libraries require amount of memories in the proportion of number of processes, and can not be used for exa-scale systems. Therefore, exa-scale communication libraries must reduce memory usage by not having all the other process information in local memory statically. To realize the goal, we propose the Advanced Communication Primitives (ACP). ACP provides global memory access primitives which can operate distributed linked list and distributed data structure easily. ACP also provides low level distributed memory copy and message passing primitives to realize low memory usage communication for MPI and PGAS runtime libraries.

Topics: Parallel and Distributed Computing

1 Motivation

Many of countries have been planning to develop exa-scale systems, and Japanese government also announced to develop exa-scale system by the end of 2020. For the exa-scale system, we are researching high performance communication library which is able to be used for 1-10 million process system. However, current communication libraries, such as Open MPI and MPICH, require amount of memories in the proportion of number of processes. For example, Open MPI requires 22GB of memory at 10 million processes, even if Unreliable Datagram protocol of InfiniBand[1]. This is because current MPI libraries have all the other process information in local memory statically using arrays of structures or linked lists. To realize exa-scale communication, the memory usage must be reduced dramatically.

2 The Advanced Communication Primitives:ACP

To realize the goal, we propose the Advanced Communication Primitives (ACP) with global memory access and management functions. ACP aims to realize low-level communication primitives, and be used to implement MPI libraries and the other PGAS based languages on top of ACP.

Before designing ACP, we analyzed internal memory usage of MPI_Init() function to clarify what kind of memory structure MPI library allocated. The analysis results show that the internal memory structures of the MPI_Init() function has all of the other process information structures, which are implemented as structure arrays and linked lists, and communication buffers statically in each process memory. This is the reason which current MPI communication libraries require amount of memories in the proportion of number of processes.

For exa-scale communication, these information structures must be distributed to the original process memory and accessed remotely when needed. To manipulate distributed structure effectively, ACP provides global address which is able to use remote atomic memory operations on global memory. Current global address handle of ACP is described as 64 bit unsigned integer type data so as to use hardware atomic operation directly. Programs with ACP do not have to recognize whether a global address data exists on local memory or not. They only recognize it when accessing the data on directly. ACP provides the translation function from global address to local logical memory address, and, when the function fails, the data is on the other process memory, and need to be copied from the global data to local memory to access it.

ACP also comprises two main categories of interfaces, the channel interface and the global data structure collection. These interfaces are built on the basic layer that provides a global memory model among processes. Each process can register and deregister its local memory to the global memory individually without inter-process synchronization. The primal data transfer function of the layer is 'copy' on the global memory. The initiator process of copy can be neither the source nor the destination. The 'copy' function is directly implemented by using RDMA when network hardware, such as InfiniBand or Tofu Interconnect, has RDMA.

ACP is different from the other related communication libraries in the point that ACP supports low level global memory access and operation to reduce memory usage. System software developer can develop communication libraries and PGAS languages with less memory usage easily by using ACP.

We are now implementing the basic layer of ACP for UDP, InfiniBand and Tofu which is the interconnect of 10 PFlops K computer. In this poster, we will present our current activities of ACP.

References

1. Shinji Sumimoto, Hideyuki Akimoto, Yuichiro Ajima, Takayuki Okamoto, Tomoya Adachi, and Kenichi Miura. Dynamic Memory Usage Analysis of MPI Libraries Using DMATP-MPI. In *Proc. of the International Conference EuroMPI 2013(poster)*, 2013.