MPI3:Merging Communicators

Hybrid Programming Working Group

August 28, 2011

0.1 Merge of Communicators

0.1.1 Rationale

MPI does not provide now a convinient way of merging commnunicators with partially overlapping groups. Two communicators with partially overlapping groups can be merged by calling MPI_INTERCOMM_CREATE, followed by a call to MPI_INTERCOMM_MERGE – but additional work is needed to select a leader in each communicator that belongs to the overlap, and to break symmetry when merging. The merge of multiple communicators will require a complex, probabilisitc algorithm, in the general case. The merge can be done in a much simpler manner by the MPI library that always has access to absolute ids for the MPI processes.

The merging of communicator may be required in a variety of situations. One such situation arises when ednpoints are used: A library invoked with one endpoint per OS process may want to use all existing endpoints, by merging the communicator argument with the MPI_PROCESS communicators at each OS process.

0.1.2 Merge Function

The following function is used to merge partially overlapping communicator groups.

MPI_COMM_MERGE(comm1, comm2, newcomm)

IN	comm1	first merged communicator (handle)
IN	comm2	second merged communicator (handle)
OUT	newcomm	new communicator (handle)

```
int MPI_Comm_merge(MPI_Comm comm1, MPI_Comm comm2, MPI_Comm* newcomm)
```

```
MPI_COMM_MERGE(COMM1, COMM2, NEWCOMM, IERROR)
    INTEGER COMM1, COMM2, NEWCOMM, IERROR
```

Either comm1 or comm2 can be null, but not both. If they are both non-null, then the groups of comm1 and comm2 are merged together in the new communicator. As a result the communicator returned in newcomm contains all endpoints that are in the same connected component as the invoking endpoint. The ranks in the returned communicators are arbitrary.

The effect of this function is illustrated in Figure 1.

The call is invoked by all 13 endpoints shown in the figure. 4 of the calls provide two non-null communicator arguments, the other provide one non-null argument. The result is two disjoint communicators, one for each connected component of the input communicators.

MPI_COMM_MERGE is collective over each connected component of the input communicators.

> *Advice to users.* This call is useful in a situation where a library is invoked collectively by one endpoint per OS process (in the old MPI model).The library can invoke MPI_COMM_MERGE with the communicator argument passed to the library and with MPI_COMM_PROCESS in order to create a communicator containing all the endpoints at these OS processes. (*End of advice to users.*)

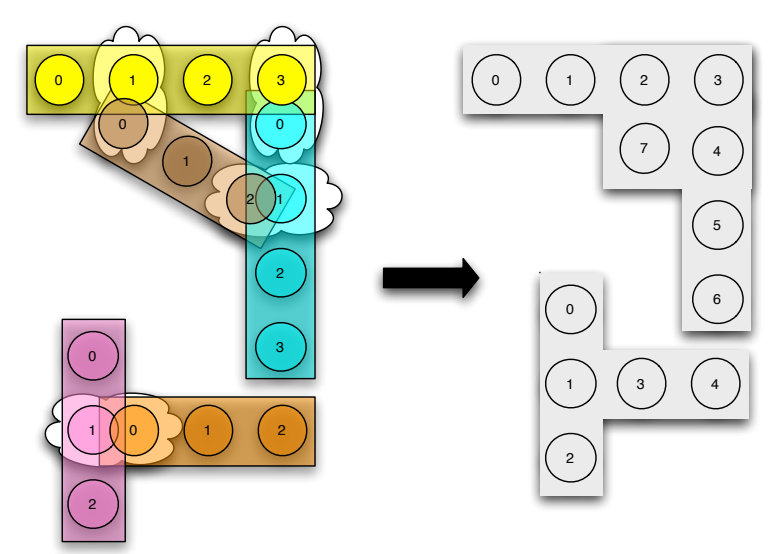

Figure 1: Communicator created by the merge of several communicators