Endpoints Proposal Update

Pavan Balaji, Jim Dinan, and
MPI Forum Hybrid Working Group
Endpoints Proposal Status

• Endpoints is proposed for MPI 4.0
• Hybrid WG has completed formal proposal
• Formal reading scheduled for December ‘14 meeting
  – Then on to voting!

• Further reading:
Endpoints and Performance Tradeoffs

Threads/proc. are entangled, users must make tradeoff
• Benefits of threads to node-level performance/resources
• Versus benefits of processes to communication throughput
Enable threads to achieve process-like communication performance

- Eliminate negative interference between threads
  - Both semantics (ordering) and mechanics (implementation issues)
- Enable threads to drive independent traffic injection/extraction points

**Goal: MPI Endpoints Relax Tradeoffs**

Communication throughput

- More Processes

Reduce memory pressure

- Improve compute perf.

- More Threads
MPI Endpoints Semantics

MPI_Comm_create_endpoints(MPI_Comm parent_comm, int my_num_ep, MPI_Info info, MPI_Comm out_comm_handles[])

Creates new MPI ranks from existing ranks in parent communicator
- Each process in parent comm. requests a number of endpoints
- Array of output handles, one per local rank (i.e. endpoint) in endpoints communicator
- Endpoints have MPI process semantics (e.g. progress, matching, collectives, ...)

Threads using endpoints behave like MPI processes
- Provide per-thread communication state/resources
- Allows implementation to provide process-like performance for threads
MPI Endpoints
Relax the 1-to-1 mapping of ranks to threads/processes
int main(int argc, char **argv) {
    int world_rank, tl;
    int max_threads = omp_get_max_threads();
    MPI_Comm ep_comm[max_threads];

    MPI_Init_thread(&argc, &argv, MPI_THREAD_MULTIPLE, &tl);
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);

    #pragma omp parallel
    {
        int nt = omp_get_num_threads();
        int tn = omp_get_thread_num();
        int ep_rank;
        #pragma omp master
        {
            MPI_Comm_create_endpoints(MPI_COMM_WORLD, nt, MPI_INFO_NULL, ep_comm);
        }
        #pragma omp barrier
        MPI_Comm_rank(ep_comm[tn], &ep_rank);
        ... // Do work based on ‘ep_rank’
        MPI_Allreduce(..., ep_comm[tn]);

        MPI_Comm_free(&ep_comm[tn]);
    }
    MPI_Finalize();
}
More Info

Endpoints:
- https://svn.mpi-forum.org/trac/mpi-forum-web/ticket/380

Hybrid Working Group:
- https://svn.mpi-forum.org/trac/mpi-forum-web/wiki/MPI3Hybrid