Updates to the Endpoint Proposal

Torsten Hoefler for Marc Snir

MPI Forum, Stuttgart, Germany, Sep. 17th 2010

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Endpoints and Shared Memory

- Ron's proposal: adding shared memory interface to MPI
  - Well, it's not quite MPI (POSIX defines a portable shared memory interface and is widely supported)
  - Seems nice in theory (less and clearly defined shared state ☺)
  - Not used in practice – SM was the main IPC mechanism but didn't catch on

Endpoints and Shared Memory

- Current practice is to run more than one MPI process per CC memory domain
  - Various reasons (e.g., scalability limitations in OpenMP regions, more than one NIC per host, …)
  - MPI processes cannot share data
    - Endpoints solve this problem
    - “threaded”/shared memory MPI processes

Changes from last Forum

- Moved MPI_Comm_merge to collectives working group
  - Needed if non-endpoint-aware codes call endpoint-aware libraries
  - Such libraries can just merge their MPI_COMM_PROCESS'
  - MPI Initialization still needs to be changed
    - Common denominator cf. MPI_Init_thread ☹

Proposal “Cleanup”

- “Strengthened” terminology
- Moved non-essential parts to end
  - (rationales, not relevant to standard)
- Better Intro
- Re-worked (non-specific PGAS interface)

New PGAS Interface

- Locale =! UPC thread | CAF image
- Execution Modes
  - Hybrid – mix MPI and PGAS (e.g., enforce locality on NUMA)
    - Different number of endpoints (from one per locate to one per program)
  - Uniform –PGAS program calls MPI library
    - Possible use-case for Comm_merge ☺

Unified PGAS Interface

- Support CAF and UPC
  - And potential new/other models
- Needs support from PGAS environment
  - Propose environment query
    - E.g., `void upc_get_config(int &program, int &process)`
      - program: index of program executed by calling thread
      - process: index of OS process of the calling thread
    - Similar for CAF

New Initialization

- `int MPI_Init_pgas(int *argc, char **argv, int flag)`
  - Create endpoint if flag==1
  - Implementation would most likely need to query PGAS - upc_get_config (or similar)
  - Can be implemented similarly to MPI_INIT_ENDPOINT

Questions/Discussion

Questions?/Discussion