MPI3 RMA

William Gropp
Rajeev Thakur

MPI-3 RMA

- Presented an overview of some of the issues and constraints at last meeting
- Homework - read Bonachea's "Problems with using MPI 1.1 and 2.0 as compilation targets for parallel language implementations", http://upc.lbl.gov/publications/bonachea-duell-mpi.pdf
- Called for alternative proposals; several volunteers but no activity on mailing lists
  - Note that we're waiting on proposals
- Breakout will be used to sign up members and to identify subgroups to develop proposals
- Possible topics
  - Address issues in Bonachea's paper (e.g., memory model, address-space accessibility)
  - Exploit different memory models (e.g., cache coherence or other consistency models)
  - Minor updates (e.g., need for put-ack)
  - Extending accumulate to some kind of user-defined operations
  - Extending accumulate to support read-modify-write operations
  - Active message (and why not with threads/point-to-point?)

MPI RMA Breakout

Agenda

- Sign up for the MPI3 RMA Mailing list!
- Status of Efforts Launched at last Meeting
- Call for new topics
- Discussion of possible topics

Issues Raised in Bonachea's Paper

- Address-space accessibility
  - Collective creation
  - Potential limitations on passive target window memory
- Memory model
  - Erroneous vs. undefined behavior
  - Local updates vs remote updates
- Passive target allows only one target process
  - cannot easily atomically update memory owned by several processes

- Note that MPI's choices made to reflect hardware realities at the time (some of which still apply, some of which may return)

Exploit different memory models

- Specialization for cache coherence
- Specialization for release consistency, etc.

Minor updates

- Option to turn off the required ack for put

Extending accumulate to some kind of user-defined operations

- Homogenous platforms only?
- Limit to some datatypes (e.g., integers)?

Extending accumulate to support read-modify-write operations

- Fetch and increment
- Compare and swap
- Transactional memory (e.g., update or not, atomically)

Active message

- Why not with threads and point-to-point?