To MPROBE… and beyond! MPI_ARECV

Squyres and Goodell

Madrid, September 2013

Last edit: v0.4 12 Sep 2013

MPROBE

- Typical use:
  - Mprobe to discover unknown message
  - Allocate space for the message
  - Mrecv to actually receive the message
- But:
  - There is no request-based version
  - Can't TEST* or WAIT* for unknown-sized message in conjunction with other known messages

Why not collapse that?

- MPI allocates the receive buffer
  - MPI_Arecv(source, tag, comm, &status)
  - MPI_Iarecv(source, tag, comm, &request)
  - Allows receipt of unknown-sized messages in array TEST/WAIT functions
- When the receive completes, get the message in a contiguous buffer: ← Per Rolf feedback
  - MPI_Status_get_buffer(status, &buffer)
- Later, MPI_Free_mem(buffer)

Assumptions

- Received message is self-describing
  - (this is an application issue)
- Possibly also use MPI_GET_ELEMENTS[_X] and/or MPI_GET_COUNT[_X]

Other common workarounds

1. Post larger receive than necessary
   - Potentially wastes space
   - Not always possible
2. Send 2 messages: a) size, b) actual message
   - Incur latency cost
3. Application based long-message rendezvous
   - Complex application logic

ARECV scenario: pre-posted

- T=0: MPI_Arecv posted
- T=1: matching message arrives
- T=2: matched to pre-posted envelope
- T=3: notices that it's an ARECV
  - malloc() a buffer / get a freelisted buffer / etc.
  - Only happens if the match is an ARECV
  - Bonus:
    - May be able to give network buffer back to caller

ARECV scenario: unexpected

- T=0: message arrives
- T=1: no pre-posted match is found
- T=2: buffers unexpected message, puts on unexpected list
- T=3: matching MPI_Arecv is posted
- T=4: finds match on unexpected queue
  - May be able to give unexpected buffer directly to MPI_ARECV (vs. another malloc+memcpy)

Per Forum feedback

MPI Forum Feedback

- What happens on allocation failure?
  1. Leave the behavior undefined
  2. Define the behavior
- Where's the performance benefit?
- Don't well-written apps not have this problem?
- What if I want to use cudaMalloc()?

Summary: Solving 3 problems

- Cannot MPI_TEST*/WAIT* for known- and unknown-sized messages
- May eliminate extra copy for unexpected (short) messages
- Accelerate rendezvous CTS for unexpected (large) messages

KTHXBYE