User-defined Nonblocking Collectives Must Make Progress

Jed Brown

Mathematics and Computer Science Division, Argonne National Laboratory

Exascale Research Conference 2012-04-16
Blocking communication is the enemy

Pipelined GMRES has no blocking communication

In PETSc, Block Jacobi with ILU preconditioning, -ksp_type pgmres
Not all nonblocking algorithms belong “upstream”

- **Tall skinny** \(QR\)
  - Essentially \textit{Allreduce()} with side effects
  - In this case, needed to reconstruct orthogonal \(Q\).

- **Unstructured communication setup**
  - Neighbor discovery from one-sided specification
  - Sparse matrix assembly
  - Many AMR applications

- **Fast multipole method**
  - Coarse levels have little computation
  - Can overlap with local work
Ways to ensure progress

▶ **Just spawn a comm thread**
  ▶ Where should we put it?
  ▶ Comm threads displace computation threads and compete for shared resources.
  ▶ Many libraries with their own comm threads don’t play nicely.

▶ **MPI Generalized Requests**
  ▶ Original MPI-2 had no way to have the request polled.
  ▶ Latham, Gropp, Ross, and Thakur 2007 extended added an extension for polling, but only when *that request* is tested.
  ▶ MPI-3 nonblocking collectives are still “special” in that users cannot provide a nonblocking interface with comparable semantics.

▶ **Common event-driven interface**
  ▶ Could be a simple extension of MPI Generalized Requests.
  ▶ Any new programming models should provide something comparable.