Communication: Alpha Reducing Optimizations

• When you are sending too many tiny messages:
  • Alpha cost is high (a microsecond per msg, for example)
  • How to reduce it?

• Simple combining:
  • Combine messages going to the same destination
  • Cost: delay (less pipelining)

• More complex scenario:
  • AllToAll: everyone wants to send a short message to everyone else
  • Direct method: \( \alpha \cdot (P-1) + \beta \cdot (P-1) \cdot m \)
  • For small m, the \( \alpha \) cost dominates
  • E.g. 1024 nodes, m is 16 bytes, is \( \alpha \) 1 microsecond, \( \beta \) is 1 nanosecond
    • First term is about 1 ms, second term 16 microseconds
An Alternative Algorithm for all-to-all

Organize processors in a 2D (virtual) Mesh

**Phase 1:** Processors send messages to $(\sqrt{P} - 1)$ row neighbors

**Phase 2:** Processors send messages to $(\sqrt{P} - 1)$ column neighbors

$2^* (\sqrt{P} - 1)$ messages instead of $P-1$
Short message all-to-all

• So, 2-d virtual topology is beneficial for short messages

• You can use a 3 (or higher) dimensional virtual topology
  • Or a hypercube

• For very short messages and very larger number of processors, these may be beneficial

• Streaming random communication (a variant of all-to-all pattern)
  • Each process continuously generates a stream of short messages, each meant for some random processor
  • A 2D routing structure like on the previous slide can be useful here too:
  • E.g. the TRAM library in Charm++: allocate fixed size buffers for each destination, and send them when they fill up.
PGAS: Partitioned Global Address Space

• PGAS languages and libraries support global view of data

• Global Arrays
  • Library (no compiler magic)
  • Declare global arrays and partition them across nodes
  • Get and put primitives for tiles (sub-sections) of global array
  • NWChem is a major application developed using it

• UPC and UPC++ (Unified Parallel C)
  • Takes the idea of a pointer with pointer arithmetic to the distributed memory world
  • Berkeley implementation and GWU implementation

• CAF (Co-array Fortran): is now incorporated in Fortran standard
Other programming models

- Task Based Parallel Programming Models
  - ParSEC
  - Legion

- Higher level compiled languages:
  - Chapel, Regent, ..

- C++ based languages
  - STAPL, DARMA, FleCSI
Topics not covered (but are important)

• Many more Parallel Numerical Algorithms (FFT, SOLVERS, ...)
• Parallel Input Output
• Graph Algorithms
• Parallel Machine Learning
• FPGAs
• ... 

• Exciting era of parallel computing coming up, because of architectural innovations spurred by end of Moore’s Law, and new classes of applications