UPC Applications

Parry Husbands
14/3/03
Recent focus on Berkeley UPC Compiler
- Performance measurements
- Identify areas for compiler “improvements”

Applications tested
- NAS Benchmarks
  - EP, IS, MG
  - CG working, performs well compared to Aztec implementation, but data set unrealistic
- Delaunay Triangulation (HP compiler so far)
EP on Alpha/Quadrics (GWU)

EP -- class A

Mops/second vs. number of threads

- HP
- Berkeley
IS -- Class B

Mops/seconds

number of threads

HP
Berkeley

IS on Alpha/Quadrics (GWU)
MG Class B

- berkeley
- berkeley2
- berkeley3
- f77+MPI

Processors

MFlop/s

0 1000 2000 3000 4000 5000 6000 7000 8000 9000

0 10 20 30 40

MG on Alpha/Quadrics (B)
MG on T3E (last year)

![Graph showing performance metrics for MG Class B on T3E. The graph plots MFlops/s against the number of processors with two lines: one for UPC and one for MPI. The number of processors ranges from 0 to 300, and MFlops/s from 0 to 14000.]
2-d Delaunay Triangulation

- Mesh generation
  - First step in countless simulations
- Based on a divide-and-conquer algorithm of Blelloch, Miller, and Talmor (1996)
- No MPI version available
- Parallelism managed with UPC and base case solved using Triangle (Shewchuk)
- Interesting features:
  - Uses collective “teams” as algorithm divides both data and processes
  - Implements a simple caching scheme for points
- Demo in UPC booth at SC02
Other Applications Under Consideration

- Splash Benchmarks
  - Barnes Hut
  - FMM
  - Ocean
  - Radiosity
- Sparse Cholesky
  - Will be based on current OpenMP version
- Various Sorting Algorithms
If you build it, they will come…

- Library support missing in UPC
  - Current application writers are forced to re-invent the wheel
  - Collective spec. a step in the right direction
- Non-blocking bulk communications
- Mechanism for handling groups of processes
- UPC++?

- Debugging
- Interoperability with MPI
  - Can always use GASNet MPI conduit…