Data Structures

Index space

- Box : a rectangular region in index space
- BoxArray : a union of Boxes at a level

Real data at a level

- FAB: FORTRAN-compatible data on a single box
  - Data on a patch
  - These patches are quite large – thousands of points
- MultiFAB: FORTRAN-compatible data on a union of rectangles
  - Data at a level
  - Owner computes rule on FAB data.
- FluxRegister: FORTRAN-compatible data on the border of a union of rectangles
  - Data for synchronization

Data Operations

Index space Operations:

- Create and manage box topology
- Identify neighbors on same level
- Identify which coarse grids underlie a given fine patch

Single-level operations

- Fill boundary data from same-level grids
- Fill data using physical boundary conditions
- Integrate data at a level
  - Patch by patch for explicit algorithms
  - Solve over all patches at a level for implicit algorithms

Multi-level operations

- Interpolate : coarse → fine
- Average : fine → coarse
- Fill boundary data from coarser grids
- Synchronization
  - Local corrections for explicit algorithms
  - Implicit synchronization systems for implicit algorithms

Metadata, communications and solvers

Index space operations are naively $O(n^2)$

- Each box needs to know its neighbors
- Bin BoxArray spatially
- Limit searches to boxes in neighboring bins

Communication

- Every MultiFAB with the same BoxArray has the same distribution
- Each processor caches list of its grids' nearest neighbors and their processors
- Each processor caches list of coarse grids and their processors used to supply boundary conditions
- Messages are ganged: no more than one message is ever exchanged between processors in an operation

Weak vs. Strong Scaling

Strong scaling

- Problem size stays fixed, number of processor increases
- Work per core decreases

Weak scaling

- Problem size increases as number of processor increases
- Work per core stays fixed

Which is more meaningful and/or more relevant?

How might you modify a scaling test to account for hopper's architecture?

Projects for Today

Go to /project/projectdirs/training/HIPACC_2011/almgren

cp -r BoxLibTest into your scratch space

cd BoxLibTest/NewCode in your own scratch space

make

This will build an executable that solves the wave equation on a union of grids.

To run the code, you can use the file “pbs_hopper” to submit a job in the debug queue.

The executable name is main2d.Linux.gcc.gfortran.MPI.ex

First, understand how the code works

Then, possible projects are:

- Convert the code to three dimensions. How many changes to you have to make in the C++?
- Do a scaling test in 2D and 3D. Decide how many steps and how big a problem you should run in order to get reliable timings. Do both a weak scaling test and a strong scaling test.
- Convert one of your own single grid codes to run this way.
- Set plot_int and Nsteps in the inputs file to generate at least 100 plotfiles and make a movie of them.