Slide 1

Administrivia

- Homework 3 on Web. Due next Wednesday.

Slide 2

Task Parallelism — Recap of Key Ideas

- Defining tasks — want "enough, but not too many".
- Managing any data dependencies (if none, "embarrassingly parallel").
  - "Removable" dependencies — e.g., temporary variable inside loop.
  - "Separable" dependencies — e.g., reductions.
  - Other dependencies — e.g., state of RNG.
- Scheduling tasks and detecting termination.
- (Review examples — molecular dynamics, Mandelbrot computation.)

Slide 3

Geometric Decomposition — Recap of Key Ideas

- Decomposing data — size and shape of "chunks", assigning chunks to PEs / UEs.
- Updating chunks. (If this can be done totally independently, *Task Parallelism* is really a better fit.)
- Synchronization / communication.
- (Review examples — heat diffusion, block-based matrix multiplication (next time).)

Slide 4

Minute Essay

- The simple strategy for parallelizing the heat diffusion program with OpenMP involves a lot of thread creation (twice per time step). Is there a way to do better? (Does the strategy you'd use for MPI provide hints?)

Slide 5

Minute Essay Answer

- Yes! You could essentially duplicate the MPI strategy in OpenMP – make the whole program an OpenMP "parallel section", with each thread doing the time step loop, with barriers at the end of each phase of the calculation. We did something like this with the numerical integration example — various "SPMD" versions in OpenMP.