Slide 1

Administrivia

- Homework 3 on the Web. Due next week. One more regular homework coming, and then we'll use whatever time remains for projects.

Slide 2

Heat-Diffusion Problem — Review/Recap

- Recall example code from last time — sequential code, OpenMP parallel versions, MPI parallel version.
- MPI version seemed to hang when run on multiple machines. In fact it does not, but it does take a *long* time with the parameters I was using in class. Why? well, to get inputs such that the computation converges, and time is long enough to make parallelization attractive, I made the number of steps large. Was that really a good idea?

Slide 3

Heat-Diffusion Problem, Continued

- MPI version has another complication, namely how to write output.
- One way would be to have each process print "its" values to a separate file, and then leave it up to the user to merge them.
- Another way is to do all output from one process, say process 0. Then other processes send "their" values to process 0. (Look at code . . .)

Slide 4

Heat-Diffusion Problem, One More Thing

- For this problem the total amount of data exchanged that needs to be sort-of-shared among UEs is small, but for other problems, particularly involving 2D etc. data structures, maybe not. So a sidebar about MPI . . .

Slide 5

Sidebar: Not-So-Simple Point-to-Point Communication in MPI

- For not-too-long messages and when readability is more important than performance, `MPI_Send` and `MPI_Recv` are probably fine.
- If messages are long, however, buffering can be a problem, and can even lead to deadlock. Also, sometimes it's useful to be able to overlap computation and communication.
- So MPI offers several other kinds of send/receive functions . . .

Slide 6

Not-So-Simple Point-to-Point Communication in MPI, Continued

- Synchronous (`MPI_Ssend`, `MPI_Recv`): locks both sender and receiver until communication can occur.
- Non-blocking send/receive (`MPI_Isend`, `MPI_Irecv`, `MPI_Wait`): doesn't block, program must explicitly test/wait.
- Persistent communication (`MPI_Recv_init`, `MPI_Send_init`, `MPI_Start`, `MPI_Wait`): allows setting up reusable path for commmunication.
- Which is faster/better? probably best to try them and find out. (Sample programs `exchange*`. Also look at one more solution to heat-diffusion problem.)

Slide 7

Heat-Diffusion Problem — OpenCL Version?

- Curiously enough(?), this problem seems less amenable to an OpenCL solution, because different work items need to communicate. “Hm!”?
- (Maybe later.)

Slide 8

Homework 3

- Assignment is to complete and then parallelize a simple text-interface program for Conway’s Game of Life (do all of you know about this?).
- OpenMP should be fairly easy. MPI is harder, but use heat-diffusion example as a model of sorts?

Slide 9

Minute Essay

- How's the reading coming? is it interesting and/or useful? And — did you buy a copy of the book, or — what? and how is that turning out?