

Administrivia

- As mentioned by e-mail, Homework 1 has been slightly extended to include a discussion of results.

Reminder: OpenMP code due Monday.

Slide 1

Minute Essay From Monday's Lecture

- Review answer in online notes. Does this make sense?

Slide 2

Intermezzo — Timing Parallel Programs

Slide 3

- “How long did it take?” often of interest. Can use system tools (e.g., `time` command) to check total elapsed time. Or can time “interesting” parts of program:

`MPI_Wtime` returns elapsed time; call twice and subtract to find out how long something takes (`time_msg.c` on “sample programs” page).

(In OpenMP, use `omp_get_wtime`.)

I like to print time plus enough identifying info (number of processes/threads, problem size) that you can “collect performance data” by capturing program output.

- How meaningful output is depends — e.g., on whether the system is otherwise idle. Probably best to repeat observations a few times, and do some sort of averaging.

Numerical Integration, Revisited

Slide 4

- Recall numerical integration example. How would we write a parallel version using MPI?
- First review the overall parallelization strategy we developed previously (dividing the computation up into the smallest reasonable tasks, figuring out how to distribute that among “units of execution” (processes/threads), figuring out how to coordinate their work).
- Now figure out how to apply it using MPI. Somewhat more complicated than with OpenMP because you don’t have a shared address space.
- Sample program `num-int-par.c`.

Minute Essay

- Any questions about MPI?

Slide 5