

Administrivia

- Homework 2 to be on Web soon. Due about a week from today. Brief discussion today.

Slide 1

Numerical Integration in OpenMP, Revisited/Continued

- Last time we looked at one way to do the numerical integration program in OpenMP, using the same SPMD strategy we used in MPI. There were problems (recall minute essay), and performance was terrible. What was wrong? Look at several versions . . .

Slide 2

Slide 3

Numerical Integration in OpenMP, Revisited/Continued

- Another approach — use worksharing construct to split loop iterations, `private` and `reduce` clauses to compute/combine partial sums.

Slide 4

Homework 2 Background

- In Homework 2, you will make a first pass at writing a set of programs (one using MPI, one using OpenMP, and one using Java) to solve the following problem. (We'll talk more about it in class after you've tried it.)
- We talked about computing π using numerical integration. Another interesting (surprising?) approach uses a "Monte Carlo" method:

Consider a square with sides of length 2 (any unit you like), enclosing a circle of radius 1.

Approximate the area of the circle by "throwing darts" at the square, counting how many fall within the circle, and calculating the ratio of those within the circle to the total number.

Model "throwing darts" by using pseudorandom number generator to generate coordinates of a point.

Minute Essay

- I showed a few results of running a sequential program to estimate π using the Monte Carlo approach. Sometimes more points gave a less accurate result. What do you think causes that?

Slide 5