

Slide 1

Administrivia

- Homework 3 due today. Okay to leave unresolved for now mathematical issues (choice of seed, how to generate different streams in different processes/threads). Goal is to resolve parallelization issues (dividing up work, communicating/synchronizing properly).

Slide 2

Recap — Algorithm Structure Patterns

- If decomposition/analysis reveals organization in terms of tasks — *Task Parallelism* (probably most common strategy) or *Divide and Conquer*.
- If decomposition/analysis reveals organization in terms of data — *Geometric Decomposition* (second most common strategy) or *Recursive Data*.
- One more possibility — organization is in terms of flow of data (*Pipeline* and *Event-Based Coordination*).

Slide 3

Pipeline

- Problem statement:
Suppose that the overall computation involves performing a calculation on many sets of data, where the calculation can be viewed in terms of data flowing through a sequence of stages. How can the potential concurrency be exploited?
- Key idea in solution — set up “assembly line” (pipeline).
- Canonical example is signal/image processing application, where you have a sequence of incoming images and want to apply same sequence of transformations to each one.

Slide 4

Event-Based Coordination

- Problem statement:
Suppose the application can be decomposed into groups of semi-independent tasks interacting in an irregular fashion. The interaction is determined by the flow of data between them which implies ordering constraints between the tasks. How can these tasks and their interaction be implemented so they can execute concurrently?
- Key idea in solution — structure computation in terms of semi-independent entities, interacting via “events”.
- Canonical example is discrete event simulation — simulating many semi-independent entities that interact in irregular/unpredictable ways.

Minute Essay

- For each of the six patterns, try to give an example (not from class) of an application where it would be useful.

Slide 5