Teaching Parallel Computing Concepts with OpenMP

Joel Adams (Calvin College)
Dick Brown (St. Olaf College)
Libby Shoop (Macalester College)
Outline

• Welcome and Introductions
• Part I: OpenMP Patternlets
  – Introduction (Joel)
  – Connecting to remote servers (Joel)
  – The Patternlets module (Libby)
  – Self-paced exploration (You!)
• Break
• Part II: OpenMP Exemplars (Dick and Libby)
• Wrap-up: Curricular discussion (Joel)
Hardware: A Diverse Landscape

- Shared-memory systems
- Distributed-memory systems
- Hybrid systems
Shared Memory Multiprocessors
Shared-Memory Software

Lots of options: Communication via...

• Shared-Memory:
  – Shared memory languages like Java, C++11, ...
  – Libraries like OpenMP, POSIX threads, ...

• Message-Passing:
  – Message passing languages like Scala, Erlang, ...
  – Libraries like the message passing interface (MPI)
OpenMP

• An industry standard library for shared-memory parallel computing in C, C++, Fortran
• Consortium includes: AMD, Cray, Fujitsu, HP, IBM, Intel, NEC, Nvidia, Oracle, Redhat, TI, ...
• Implicit multithreading via `#pragma` directives (vs. explicit multithreading in Pthreads)
• Many parallel programming patterns built in
Parallel Patterns

... are strategies that practitioners have found to be frequently useful in parallel problem-solving.

- **Origins: Industry-standard best practices**
  - Cumulative wisdom from decades of experience

- **When solving problems, experts *think* in patterns**
  - The more we can help our students master patterns, the more like experts they will be.
A Few Sample Patterns

• **Algorithmic** Strategies:
  – Data Decomposition, Task Decomposition, ...

• **Implementation** Strategies:
  – SPMD, Fork-Join, Master-Worker, Parallel Loop, ...

• **Concurrent Execution** Strategies:
  – Barrier, Mutual Exclusion, Message Passing, Broadcast, Reduction, Scatter, Gather, ...
Terminology: *Patternlets*

... are minimalist, scalable, complete programs, each illustrating one or more parallel patterns:

- **Minimalist** to help students understand the pattern by eliminating non-essential details
- **Scalable** so that students can vary the number of threads and see the pattern’s behavior change
- **Complete** so that:
  - Instructors can use them in a ‘live coding’ lecture
  - Students can explore them in a hands-on exercise
Terminology: *Exemplars*

... are programs that use one or more patterns to solve a ‘real world’ problem.

Exemplars let students see how the pattern is useful in a more meaningful context.

A *patternlet* is useful for *introducing* students to a pattern; an *exemplar* is useful for helping students see how & why a pattern is *relevant*.
Outline

• Welcome and Introductions
• Part I: OpenMP Patternlets
  – Introduction (Joel) ✓
  – Connecting to remote servers (Joel) ✓
  – The Patternlets module (Libby) ✓
  – Self-paced exploration (You!) ✓
• Break
• Part II: OpenMP Exemplars (Dick & Libby)
• Wrap-up: Curricular discussion (Joel)
Outline

• Welcome and Introductions
• Part I: OpenMP Patternlets ✔
• Break ✔
• Part II: OpenMP Exemplars ✔
  – Area under the curve (Dick)
  – Drug Design (Dick)
  – Pandemic (Libby)
  – Self-paced exploration of Exemplars
• Wrap-up: Curricular discussion (Joel) ✔

Thank you!