Teaching Shared-Memory Parallel Concepts with OpenMP

Joel Adams (Calvin College)

Libby Shoop (Macalester College)

Dick Brown (St. Olaf College)

MACALESTER COLLEGE

CALVIN MINDS IN THE MAKING

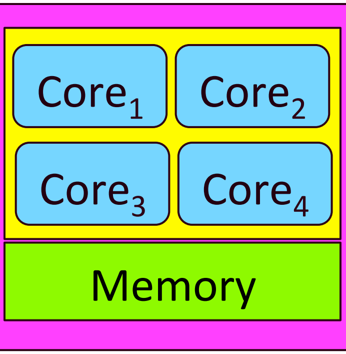
ST · OLAF COLLEGE

Outline

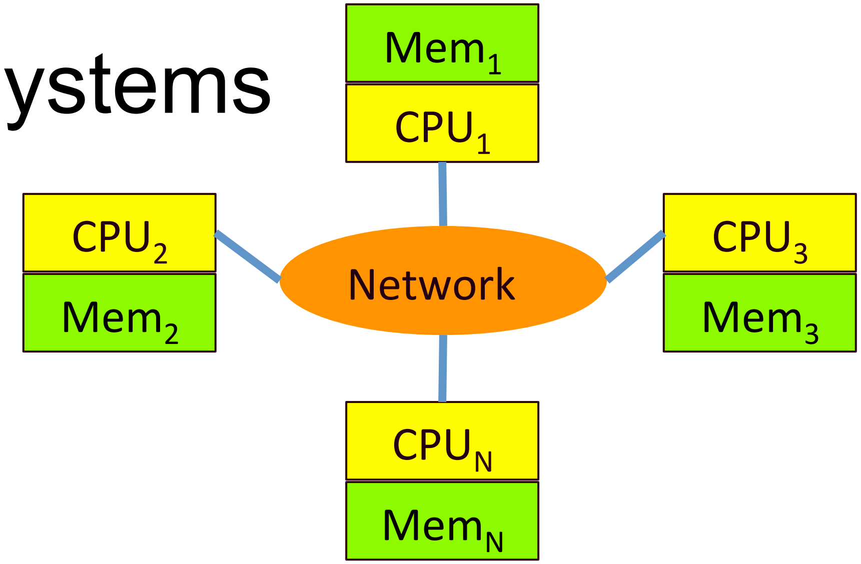
- Welcome and Introductions
- Part I: OpenMP Patternlets
  - Introduction (Joel)
  - The Patternlets module (Libby)
- Part II: Self-paced exploration (You!)
- Part III: OpenMP Exemplars (Libby)
- Wrapup: Discussion & Assessment (Joel, Libby)

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 repeatedly found to be useful in parallel problem-solving.

- Industry-standard best practices
  - Origins in industry, not academia
- Accumulated wisdom of decades of experience

When solving problems, experts *think* in patterns, so the more we can get our students to internalize 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:
  - Mutual Exclusion, Message Passing, Barrier, Broadcast, Reduction, Scatter, Gather, …

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

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 why a pattern *relevant*.

Outline

- Welcome and Introductions ✔
- Part I: OpenMP Patternlets ✔
  - – Introduction (Joel)
  - – The Patternlets module (Libby)
- Part II: Self-paced exploration (You!) ✔
- Part III: OpenMP Exemplars (Libby) ✔
- Wrapup: Discussion & Assessment (Joel, Libby)

Thank you!