

Shared-Memory Parallel Computing with OpenMP

Suzanne J. Matthews (West Point)

Joel Adams (Calvin College)

Dick Brown (St. Olaf College)

Libby Shoop (Macalester College)

Shared Memory Multiprocessors



OpenMP

- An industry standard library for shared-mem. 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 Design Patterns

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

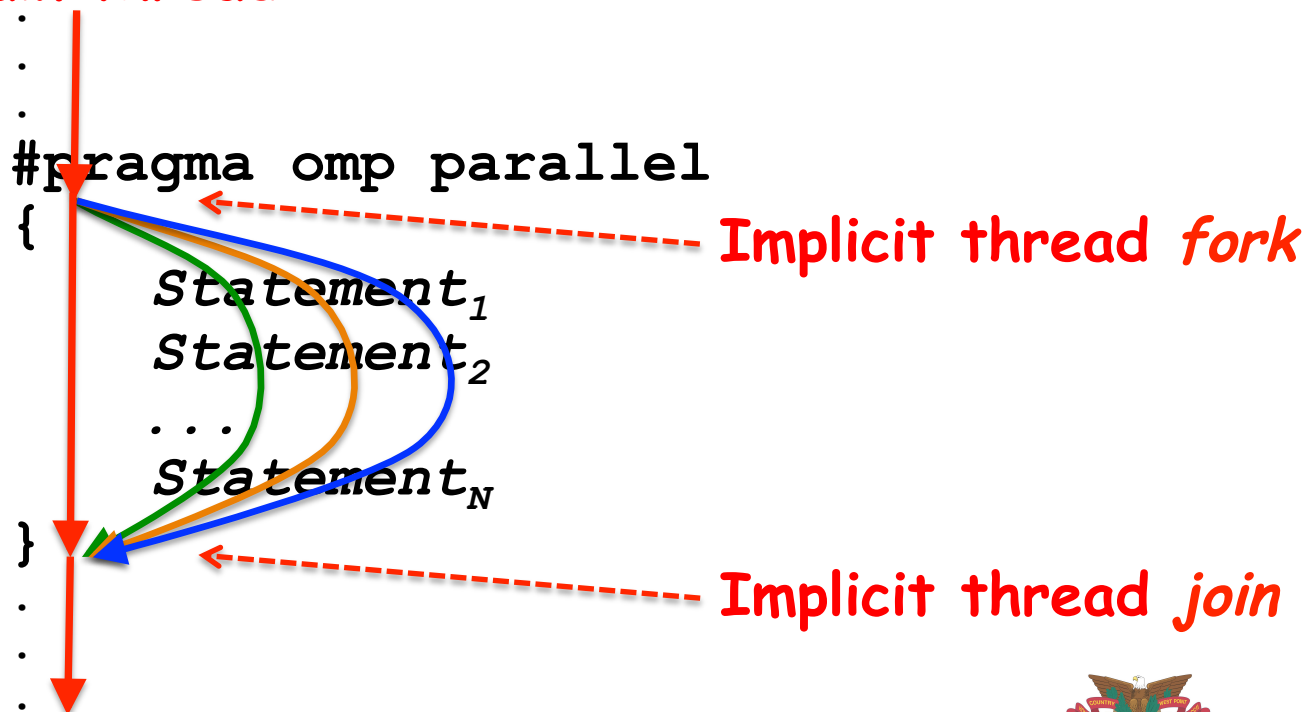
- Industry-standard **best practices**
 - These originated in *industry*, not academia
- Accumulated wisdom of decades of experience

When solving problems, experts *think* in patterns, so the more we can get our students to think in patterns, the more like experts they will be.

The Fork-Join Pattern

OpenMP uses the *fork-join pattern* implicitly:

Main thread



Categorizing Patterns

- *Algorithmic* Strategies:
 - *Data Decomposition, Task Decomposition, ...*
- *Implementation* Strategies:
 - *SPMD, Master-Worker, Parallel Loop, ...*
- *Concurrent Execution* Strategies:
 - *Barrier, Mutual Exclusion, Message Passing, Broadcast, Reduction, Scatter, Gather, ...*

Higher level



Lower level

Most parallel programs use multiple patterns

Pattern: Data Decomposition (1 task)

Task 0



Pattern: Data Decomposition (2 Tasks)

Task
0

Task
1



Pattern: Data Decomposition (4 Tasks)

Task
0

Task
1

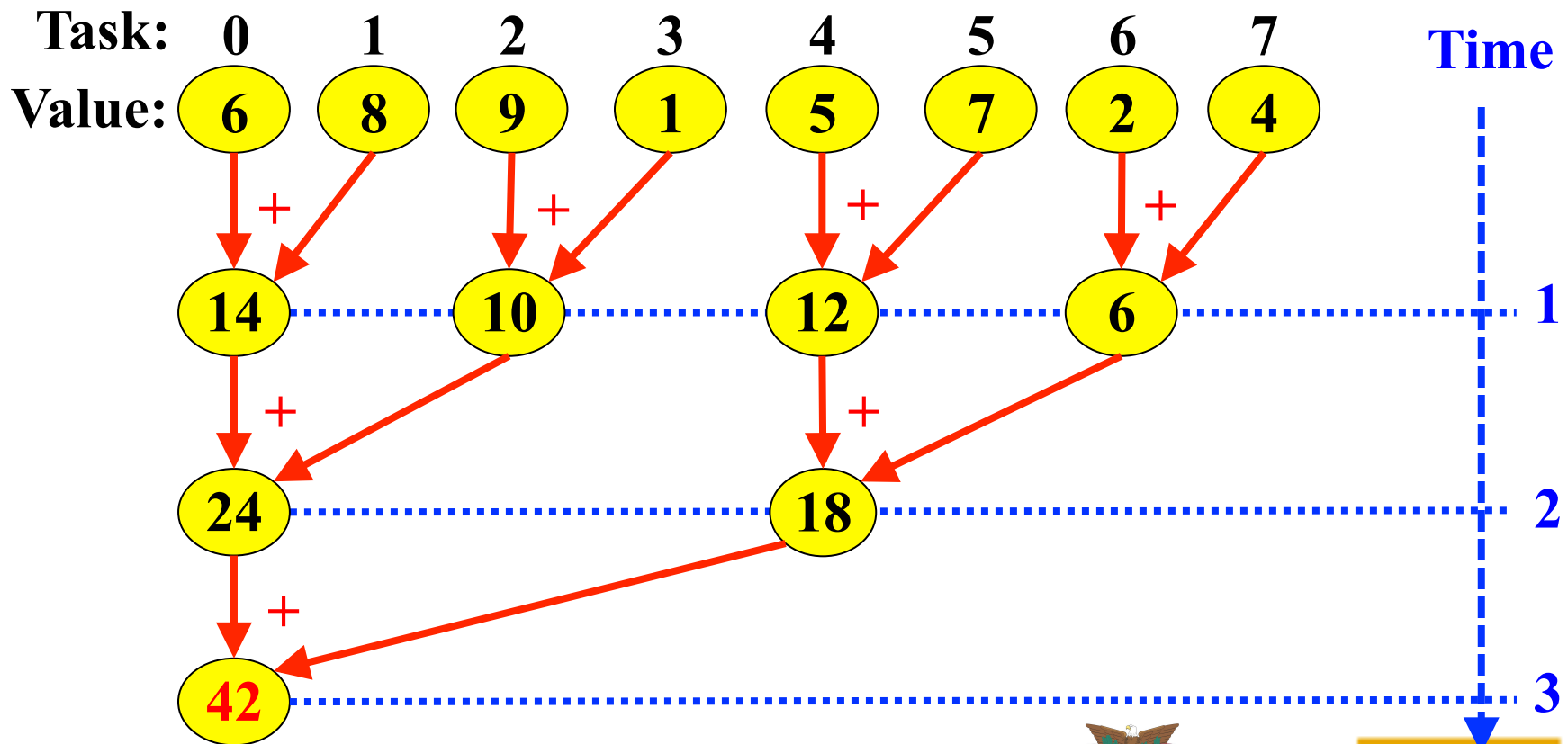
Task
2

Task
3



Pattern: Reduction (8 Tasks)

To sum the local results of N parallel tasks:



Terminology: *Patternlets...*

are minimalist, scalable, and 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 processes and see the pattern's behavior change
- *Complete* for flexible use:
 - Instructors can use them in a 'live coding' lecture
 - Students can explore them in a hands-on exercise, and use them as models for their own programs.

Terminology: *Exemplars...*

are programs that use parallel patterns to solve a 'real world' problem.

Exemplars let students see how a pattern can be useful in a meaningful context.

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

Hands On With OpenMP

- CSinParallel patternlets webpage:

<http://selkie.macalester.edu/csinparallel/modules/Patternlets/build/html/>