

EE382N (20): Computer Architecture - Parallelism and Locality
Fall 2011

Lecture 13 – Parallelism in Software IV

Mattan Erez



The University of Texas at Austin



Credits

- Most of the slides courtesy Dr. Rodric Rabbah (IBM)
 - Taken from 6.189 IAP taught at MIT in 2007.



Patterns for Parallelizing Programs

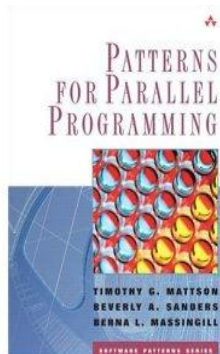
4 Design Spaces

Algorithm Expression

- Finding Concurrency
 - Expose concurrent tasks
- Algorithm Structure
 - Map tasks to processes to exploit parallel architecture

Software Construction

- Supporting Structures
 - Code and data structuring patterns
- Implementation Mechanisms
 - Low level mechanisms used to write parallel programs



Patterns for Parallel Programming.
Mattson, Sanders, and Massingill
(2005).



Quick recap

- Decomposition
 - High-level and fairly abstract
 - Consider machine scale for the most part
 - Task, Data, Pipeline
 - Find dependencies
- Algorithm structure
 - Still abstract, but a bit less so
 - Consider communication, sync, and bookkeeping
 - Task (collection/recursive)
 - Data (geometric/recursive)
 - Dataflow (pipeline/event-based-coordination)
- Supporting structures
 - Loop
 - Master/worker
 - Fork/join
 - SPMD
 - MapReduce



Map/Reduce Pattern

- Two phases in the program
- Map phase applies a single function to all data
 - Each result is a tuple of value and tag
- Reduce phase combines the results
 - The values of elements with the same tag are combined to a single value per tag -- **reduction**
 - Semantics of combining function are associative
 - Can be done in parallel
 - Can be pipelined with map
- Google uses this for *all* their parallel programs



Communication and Synchronization Patterns

- Communication
 - Point-to-point
 - Broadcast
 - Reduction
 - Multicast
- Synchronization
 - Locks (mutual exclusion)
 - Monitors (events)
 - Barriers (wait for all)
 - Split-phase barriers (separate signal and wait)
 - Sometimes called “fuzzy barriers”
 - Named barriers allow waiting on subset



Algorithm Structure and Organization (from the Book)

	Task parallelism	Divide and conquer	Geometric decomposition	Recursive data	Pipeline	Event-based coordination
SPMD	*****	***	*****	**	***	**
Loop Parallelism	*****	**	***			
Master/Worker	*****	**	*	*	*****	*
Fork/Join	**	*****	**		*****	*****

- Patterns can be hierarchically composed so that a program uses more than one pattern



Algorithm Structure and Organization (my view)

	Task parallelism	Divide and conquer	Geometric decomposition	Recursive data	Pipeline	Event-based coordination
SPMD						
Loop Parallelism						
Master/Worker						
Fork/Join						

- Patterns can be hierarchically composed so that a program uses more than one pattern



Algorithm Structure and Organization (my view)

	Task parallelism	Divide and conquer	Geometric decomposition	Recursive data	Pipeline	Event-based coordination
SPMD	****	**	****	**	****	*
Loop Parallelism	**** when no dependencies	*	****	*	**** SWP to hide comm.	
Master/Worker	****	***	***	***	**	****
Fork/Join	****	****	**	****		*

- Patterns can be hierarchically composed so that a program uses more than one pattern

