

# Chapter 12: Distributed Shared Memory

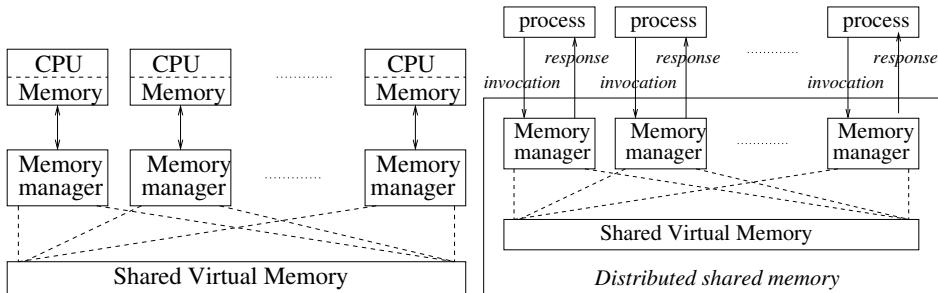
Ajay Kshemkalyani and Mukesh Singhal

Distributed Computing: Principles, Algorithms, and Systems

Cambridge University Press

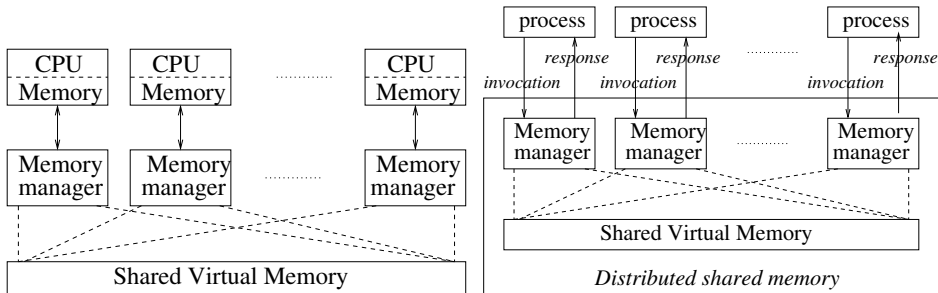
# Distributed Shared Memory Abstractions

- communicate with Read/Write ops in shared virtual space
- No Send and Receive primitives to be used by application
  - ▶ Under covers, Send and Receive used by DSM manager
- Locking is too restrictive; need concurrent access
- With replica management, problem of consistency arises!
- $\implies$  weaker consistency models (weaker than von Neumann) reqd



# Distributed Shared Memory Abstractions

- communicate with Read/Write ops in shared virtual space
- No Send and Receive primitives to be used by application
  - ▶ Under covers, Send and Receive used by DSM manager
- Locking is too restrictive; need concurrent access
- With replica management, problem of consistency arises!
- $\implies$  weaker consistency models (weaker than von Neumann) reqd



# Advantages/Disadvantages of DSM

## Advantages:

- Shields programmer from Send/Receive primitives
- Single address space; simplifies passing-by-reference and passing complex data structures
- Exploit locality-of-reference when a block is moved
- DSM uses simpler software interfaces, and cheaper off-the-shelf hardware. Hence cheaper than dedicated multiprocessor systems
- No memory access bottleneck, as no single bus
- Large virtual memory space
- DSM programs portable as they use common DSM programming interface

## Disadvantages:

- Programmers need to understand consistency models, to write correct programs
- DSM implementations use async message-passing, and hence cannot be more efficient than msg-passing implementations
- By yielding control to DSM manager software, programmers cannot use their own msg-passing solutions.

# Advantages/Disadvantages of DSM

## Advantages:

- Shields programmer from Send/Receive primitives
- Single address space; simplifies passing-by-reference and passing complex data structures
- Exploit locality-of-reference when a block is moved
- DSM uses simpler software interfaces, and cheaper off-the-shelf hardware. Hence cheaper than dedicated multiprocessor systems
- No memory access bottleneck, as no single bus
- Large virtual memory space
- DSM programs portable as they use common DSM programming interface

## Disadvantages:

- Programmers need to understand consistency models, to write correct programs
- DSM implementations use async message-passing, and hence cannot be more efficient than msg-passing implementations
- By yielding control to DSM manager software, programmers cannot use their own msg-passing solutions.

# Issues in Implementing DSM Software

- Semantics for concurrent access must be clearly specified
- Semantics – replication? partial? full? read-only? write-only?
- Locations for replication (for optimization)
- If not full replication, determine location of nearest data for access
- Reduce delays, # msgs to implement the semantics of concurrent access
- Data is replicated or cached
- Remote access by HW or SW
- Caching/replication controlled by HW or SW
- DSM controlled by memory management SW, OS, language run-time system

# Issues in Implementing DSM Software

- Semantics for concurrent access must be clearly specified
- Semantics – replication? partial? full? read-only? write-only?
- Locations for replication (for optimization)
- If not full replication, determine location of nearest data for access
- Reduce delays, # msgs to implement the semantics of concurrent access
- Data is replicated or cached
- Remote access by HW or SW
- Caching/replication controlled by HW or SW
- DSM controlled by memory management SW, OS, language run-time system

# Comparison of Early DSM Systems

Type of DSM	Examples	Management	Caching	Remote access
single-bus multiprocessor	Firefly, Sequent	by MMU	hardware control	by hardware
switched multiprocessor	Alewife, Dash	by MMU	hardware control	by hardware
NUMA system	Butterfly, CM*	by OS	software control	by hardware
Page-based DSM	Ivy, Mirage	by OS	software control	by software
Shared variable DSM	Midway, Munin	by language runtime system	software control	by software
Shared object DSM	Linda, Orca	by language runtime system	software control	by software