Ch 10

- Shared memory via message passing

Problems
- Explicit user action needed
- Address spaces are distinct
- Small Granularity of Transfer

Distributed Shared memory approach can help with these. Also, unlike tightly coupled multiprocessors
- Cheaper to build using COTS
- Memory pooled together is significant than local workstation memory
- More scalable since data bus is not a bottleneck
- SMM based programs can be easily ported
Implementation Approaches

• Central Server Based
  – A central server maintains all shared data. Provided to processors using request/response model with timeouts.
  – Problem: Scalability
  – Solution: Partition data, allocate each partition to a processor and have it coordinate requests for that partition.
  – Need a “mapping function” to map VM address to corresponding processor.
• Migration Algorithm
  – Instead of sending request to data, send data to request
  – Send a “larger” block of data than needed
    • Locality of reference
  – Access is serialized
  – Can lead to thrashing
    • Avoid by using hold downs
  – Can allow for integration with local VM mechanisms
    • Need to have conforming page sizes in VM and DSM.
  – How do you locate a block
    • Centralized mapping server
    • Hints
    • Broadcast based discovery
• Read Replication
  – Enhance basic migration by allowing multiple read copies and one write copy.
  – Invalidation on read?
  – Useful when read/write >> 1

• Full Replication
  – Allow multiple readers and writers
  – Consistency?
    • Use a sequencer
    • Process in sequence order