TreadMarks:

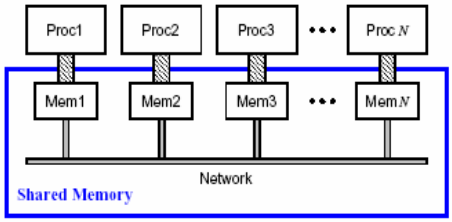
Shared Memory Computing on Networks of Workstations

By: Cristiana Amza, Alan L Cox, Sandhya Dwarkadas, Pete Keleher, Honghui Lu, Ramakrishnan Rajamony, Weimin Yu and Willy Zwaenepoel @ Rice University
Present by: Fung (Monty) Ngai

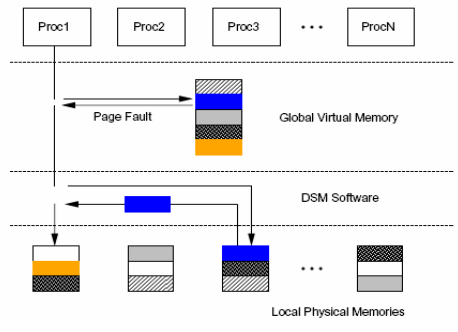
Overview

- Share Memory Programming 101
- Problems & Goals
- Protocols and Implementation
- Alternative Programming Models
- Results Analysis
- References

Share Memory Programming 101



Share Memory Programming 101



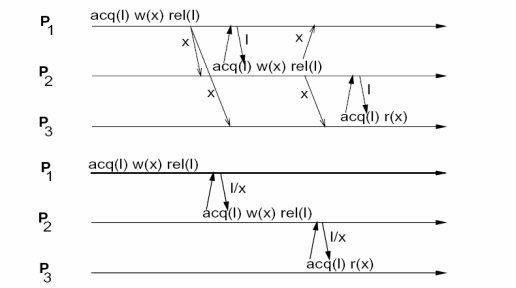
Problems

- Sequential consistency can cause large amount of communications.
- Communications is $$$ on a workstation network (Latency)
- Performance Problem: False Sharing

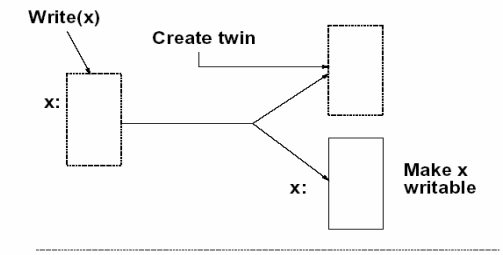
Goals

- Reduce Communication
  - Using Techniques such as Lazy Release Consistency, and Multiple Writer Protocols
- Keep Shared memory model

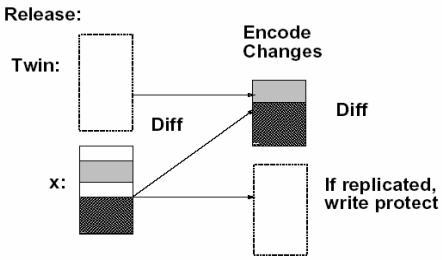
Protocols and Implementation Lazy RC



Protocols and Implementation (Multiple-Writer Protocols)



Protocols and Implementation (cont) (Multiple-Writer Protocols)



Protocols and Implementation

- Init
  1. Create Requested number of processes on remote machines
  2. Set up full duplex sockets between each process
  3. Register SIGIO handler for messaging
  4. Allocate 1 large block for shared memory at the same (VM) address on each machine and mark as non-accessible using mprotect
  5. Choose a processor in round-robin fashion to be the manager for each page of the block and for each lock and barrier
  6. Register SEGV handler for shared memory access

Results Analysis

- With this system, some things that you typically do in the message passing paradigm happen automatically
- This is at a cost (diffs and other overhead), and the message passing can typically be made more efficient
- Sounds similar to an argument about high-level programming vs. assembly programming
- Shared memory does seem to make some things nice

References

1. “TreadMarks: Shared Memory Computing on Networks of Workstations,” C. Amza et. al., Rice University, 1994
2. “Distributed Shared Memory Using Lazy Release Consistency,” P. Keleher, PhD thesis, Rice University, December 1994
3. TreadMarks API documentation of versions 0.9.8 and 0.10.1
4. “The TreadMarks Distributed Shared Memory (DSM) System,” http://www.cs.rice.edu/~willy/TreadMarks/overview.html, website