Treadmarks
Shared Memory Computing
on Networks of Workstations

Hongzhou Zhao
2007/3/21
Distributed shared Memory System
- Consist of N networked workstations, each with its own memory, connected by a network
- A global shared virtual memory rather than a collection of distributed address space
Key design features

- Memory consistency model
  Relaxed consistency model: lazy release consistency
- Virtual Memory Page Protection
  Virtual memory hardware to detect access, use multiple writer protocol
- Application Programming Interface
  Synchronization, critical sections
  Use of barrier, lock
Treadmark C interface

#define TMK_NPROCS
extern unsigned Tmk_nprocs;
extern unsigned Tmk_procid;
#define TMK_NLOCKS
#define TMK_NBARRIERS
void Tmk_startupint ( int argc char **argv);
void Tmk_exit( int status);
void Tmk_barrier(unsigned id);
void Tmk_lock_acquire(unsigned id);
void Tmk_lock_release(unsigned id);
char *Tmk_malloc(unsigned size);
void Tmk_free(char *ptr);
Example – Jacobi

length = M / Tmk_nprocs;
begin = length * Tmk_proc_id;
end = length * (Tmk_proc_id+1);

for( number of iterations ) {

    for( i=begin; i<end; i++ )
        for( j=0; j<N; j++ )
            scratch[i][j] = (grid[i-1][j]+grid[i+1][j]+grid[i][j-1]+grid[i][j+1])/4;

    Tmk_barrier(1);

    for( i=begin; i<end; i++ )
        for( j=0; j<N; j++ )
            grid[i][j] = scratch[i][j];

    Tmk_barrier(2);

}
Problems with sequential consistency model

First DSM system, IVY
- simplicity and intuitive
- unnecessary communication cost
  - send a message involves trap into OS kernel
  - only the processor acquire the lock need to be invalid
- false sharing
  - unrelated data located in the same page
  - due to page size, a serious problem
  - ping-pong effect
● Release Consistency

memory update by p become visible to q only when a subsequent release by p become visible to q

● Lazy Release Consistency

Eager/ lazy release consistency – at release/ acquire
Lazy Release consistency Implementation

- divide execution of each process into interval
- use vector timestamp

q piggy back message with write notice for all intervals in current vector without from requester $p$
multiple-writer protocol
- create a local twin copy when write
- compare word-by-word and create a diff at barrier
- when multi-writer synchronized, modify is informed to processor, when page accessed later, page fault happen, and page is updated applying diff
● Treadmark Data Structure

- Copyset
- Page State

Page Array

Write Notice Records

Interval Records

Diff Pool

Proc Array
Protocol action for Treadmark

**TWIN**

\[ w(x) \]

**Rel**

**DIFF**

\[ x \]

**Update the Page**

\[ \text{mod}(\text{page}(x)) \]

\[ r(x) \]

\[ \text{inv}(\text{page}(x)) \]
TreadMark System
- implemented as a user-level library
- communication using UDP/IP
- use SIGIO signal handler handle request message
- use SIGSEGV signal handler access shared page
Treadmarks:
Shared Memory Computing

Speedup obtained on Treadmark

- Water
- TSP
- ILINK
- Jacobi
- Quicksort
• Execution Time Breakdown

**Figure 5**  TreadMarks Execution Time Breakdown  

**Figure 6**  Unix Overhead Breakdown
● Alternative Approach

- Entry Consistency (Midway)
  each shared data object be associated with a synchronization object, using update protocol

- Structured DSM System (Linda)
  a shared space of objects instead of linear array

- Hardware Shared Memory Implementation (DASH)