Treadmarks: Distributed Shared Memory on Standard Workstations and Operating Systems

Pete Keleher, Alan L. Cox, Sandhya Dwarkadas and Willy Zwaenepoel

Proceedings of the Winter Usenix Conference

Hyun Geun Soo

Beatriz Santos

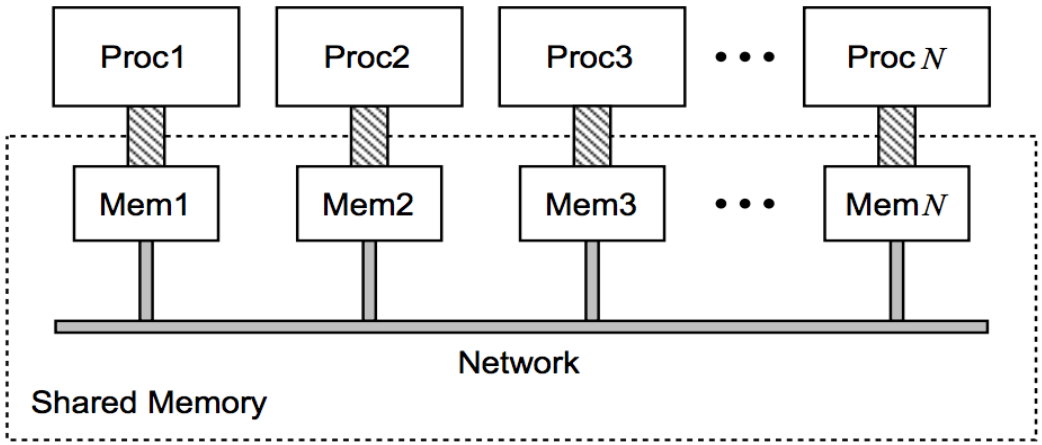
Seoul National University

Contents

- Introduction
- Release consistency
  - Eager Release Consistency (ERC)
  - Lazy Realease Consistency (LRC)
- Multiple-Writer Protocol
- Lazy Diff Creation
- Data Structure and Implementation
- Results
- Conclusion

Distributed Shared Memory

- DMS is a form of memory architecture where the (physically separete) memories can be addressed as one (logically shared) address space
  - Easier to program than MPI
  - High cost of communication



Problems of DMS

- Lack of Portability
  - Hardware
    - In-house research platforms
  - Software
    - Kernell modifications
- Poor Performance
  - Communication overhead
  - False sharing

Problems of DMS

- Lack of Portability
  - Hardware
    - In-house research platforms -> **Standard Unix Systems Platforms**
  - Software
    - Kernell modifications -> **User-level implementation**
- Poor Performance
  - Communication overhead -> **Lazy Release Consistency**
  - False sharing -> **multiple writer protocols**

Release Consistency

- Release Consistency (RC) is a relaxed memory consistency model that permits a processor to delay making its changes to shared data visible to other processors until certain syncronization access occur

- Shared memory access
  - Ordinary
  - Synhonized Accesses
    - Acquire access
    - Release access

Release Consistency

P1:

a1: acq(L)

data access

r1: rel(L)

P2:

a2: acq(L)

data access

r2: rel(L)

If P1:r1 happened before P2:a2

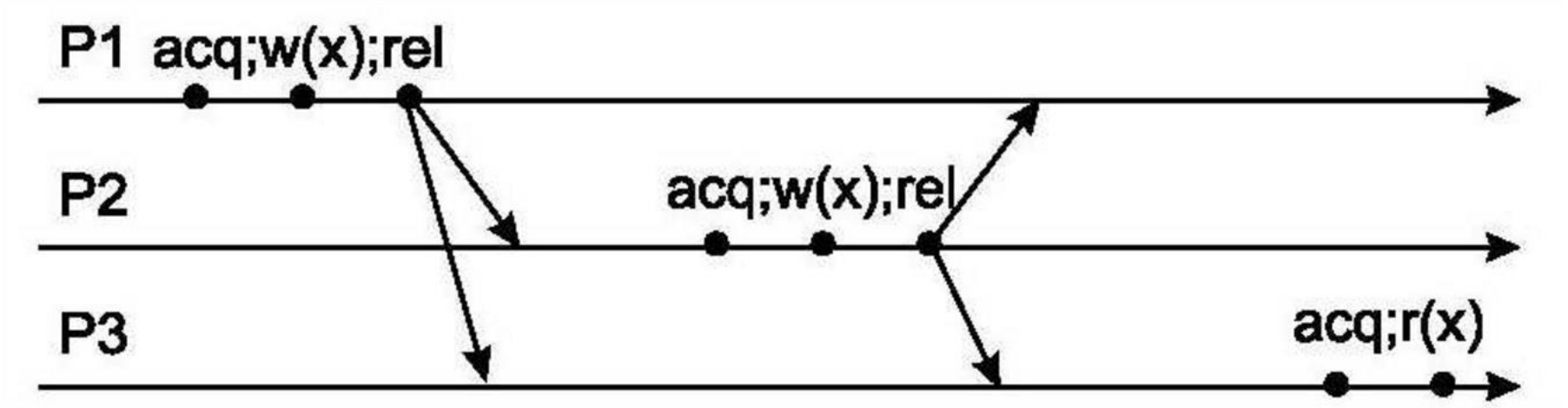
- All coherence actions prior to P1:r1 should be complete before P2:a2

Release Consistency

- Two Types of RC
  - Eager Release Consistency (ERC)
  - Lazy Release Consistency (LRC)

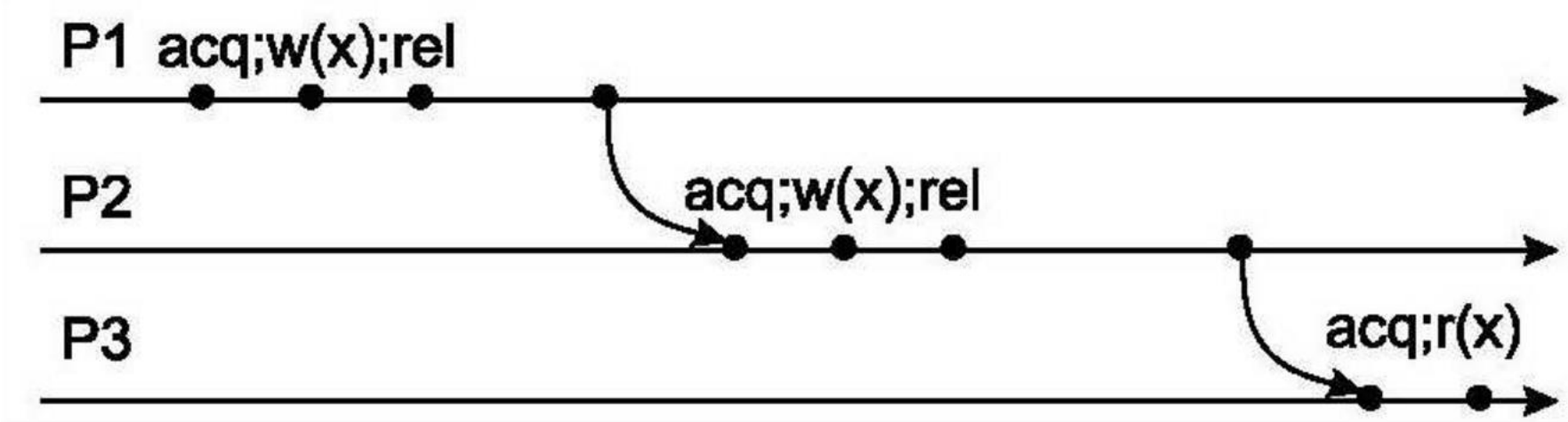
Eager Release Consistency

- Push Model



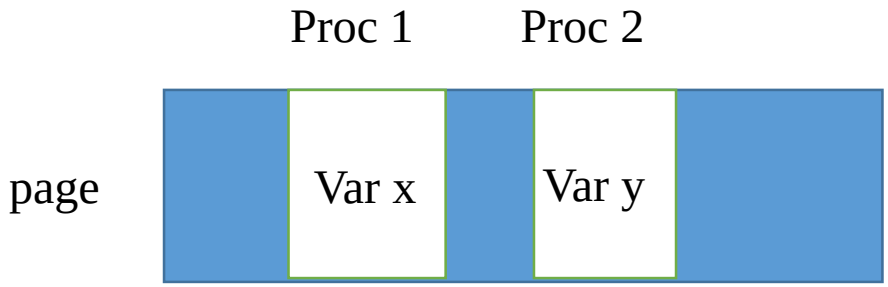
Lazy Release Consistency

- Pull Model



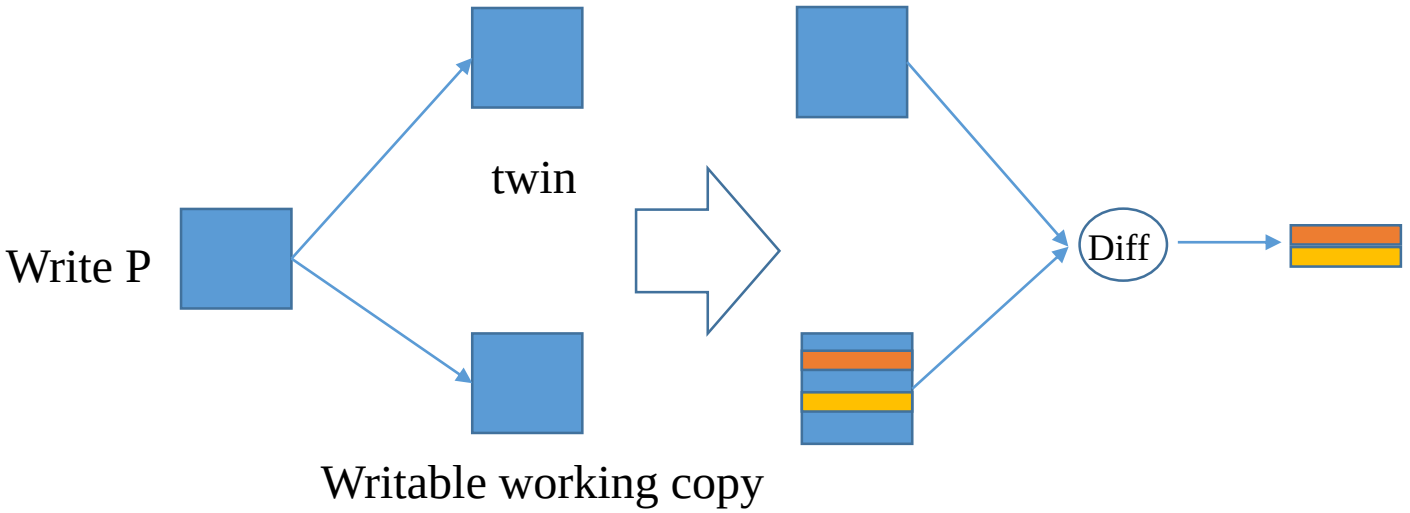
Multiple-Writer Protocol

- To solve False sharing Problem => Multiple writer protocol
  - Several processes make modifications to different variables at the same page



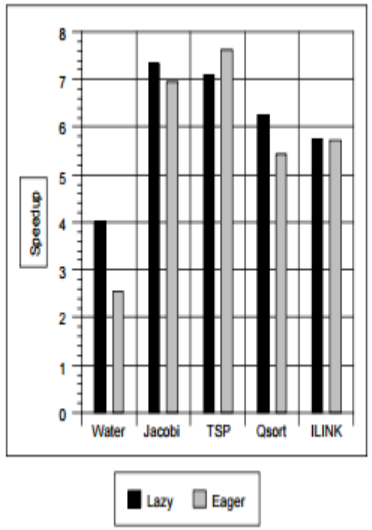
Multiple-Writer Protocol

- Two method
  - Twin
    - 1. Copy original page
    - 2. Compared original page and changed page
  - Diff
    - Difference between twin and copyset

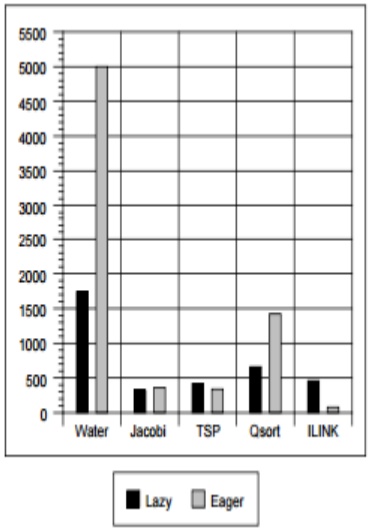


Lazy Diff Creation

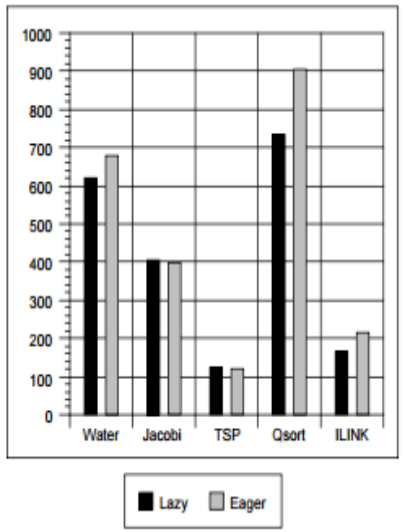
- In TreadMarks, Diff created
  - Modifications to a page
  - Write notice from another process
    - Different from Munin;s implementation
      - Diff creation postponed until the modification are requested
      - Reduce the number of diff created



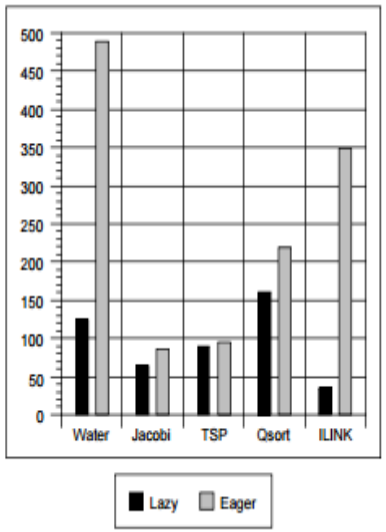
Speedup



Message Rate



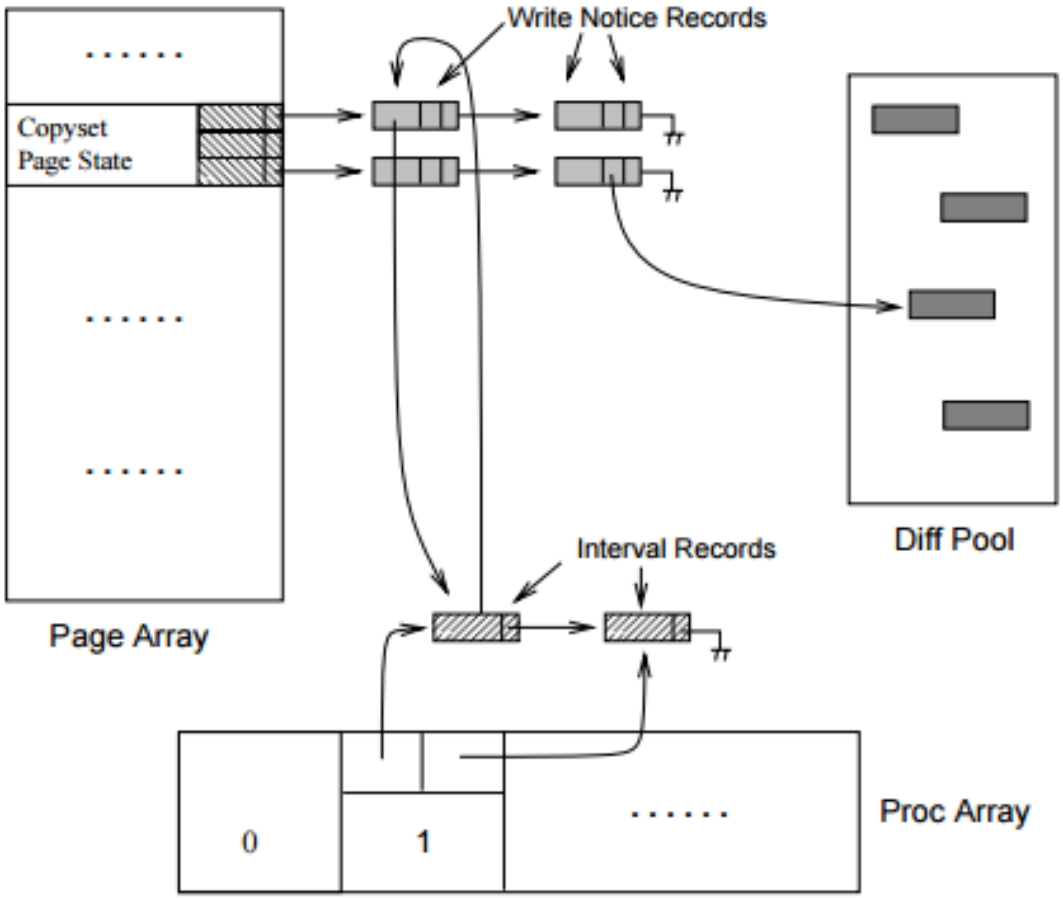
Data Rate



Diff Creation Rate

Data Structure

- Overview of Data Structure



Implementation

- Interval Creation
  - Logically
    - a new interval begins at each release and acquire
  - In practice
    - postponed until we communicate with another process
    - avoiding overhead

- Diff Creation
  - with lazy diff creation
    - Page writable until **diff request** or a **write notice arrives**
    - When **actual diff is created**, page **is read protected**, the twin is **discarded**

Implementation

- Lock and Barriers
  - Lock : Statically assigned manager
  - Assigned in a **round-robin** fashion among the processor
  - All the lock acquire requests are **directed to the manager**
  - When lock is released,
    - The releaser “informs ” the acquirer of all intervals
  - After receiving this messages
    - The acquirer “incorporates” this information into its data structures
    - 1. the acquirer appends an interval record to the interval record list for that processor
    - 2. for each write notice
      - 1) it prepends a write notice record to the page's write notice record list
      - 2) adds pointers from the write notice record to the interval record, and vice versa
  - Barriers : Centralized manager

Implementation

- Access Misses
  - without write notice
    - Initially setup that processor 0 has the page
  - with write notice
    - 1. Get the diffs from the write notice with small timestamp
    - 2. Create an actual diff which is correction of all diff related to the page
    - 3. The twin is discarded and the result is copied to copyset
- Garbage collection
  - Write notice records, Interval records, Diffs
  - It is triggered when the free space drops below a threshold
- Unix Aspects
  - TreadMarks relies on Unix standard libraries
    - Remote process creation, interprocessor communication, and memory management

Performance Evaluation

- Environment
  - 8 DECstation-5000/240
  - Connected to a 100-Mbps ATM LAN and a 10-Mbps Ethernet

- Applications
  - Water – molecular dynamics simulation , 343 molecules for 5 steps
  - Jacobi – Successive Over-Relaxation with a grid of 2000 by 1000 elements
  - TSP – branch & bound algorithm to solve the traveling salesman problem for a 19 cities
  - Quicksort – sorts an array of 256K integers. Using bubblesort to sort subarray of less than 1K element
  - ILINK – genetic linkage analysis

Results

Speedups Obtained on TreadMarks

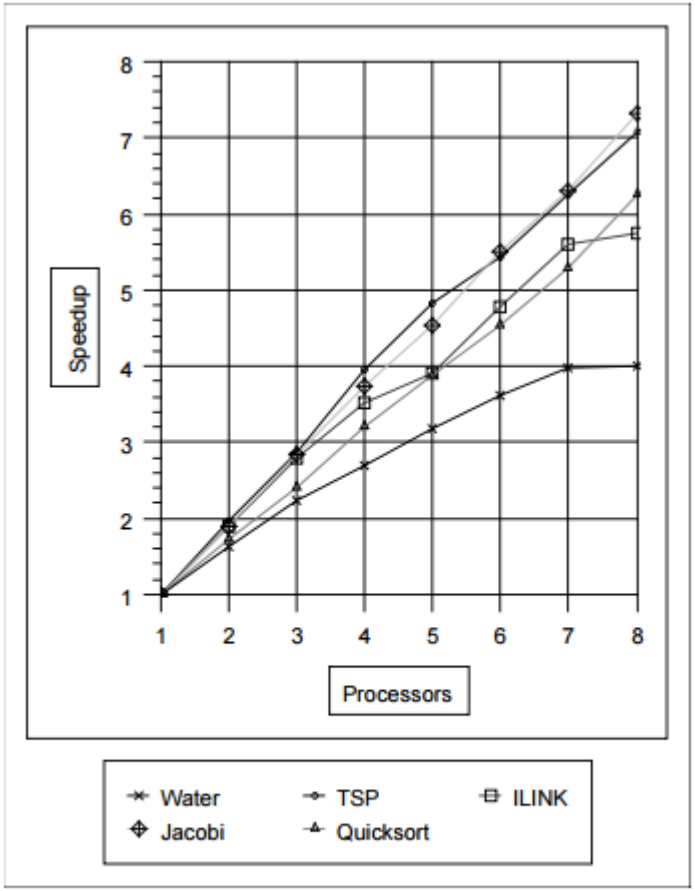


Figure 3 Speedups Obtained on TreadMarks

Execution Statistics
for an 8-processor run on TreadMarks

	Water	Jacobi	TSP	Quicksort	ILINK
Input	343 mols 5 steps	2000x1000 floats	19-city tour	256000 integers	CLP
Time (secs)	15.0	32.0	43.8	13.1	1113
Barriers/sec	2.5	6.3	0	0.4	0.4
Locks/sec	582.4	0	16.1	53.9	0
Msgs/sec	2238	334	404	703	456
Kbytes/sec	798	415	121	788	164

Figure 4 Execution Statistics for an 8-Processor Run on TreadMarks

Execution Time Breakdown

TreadMarks

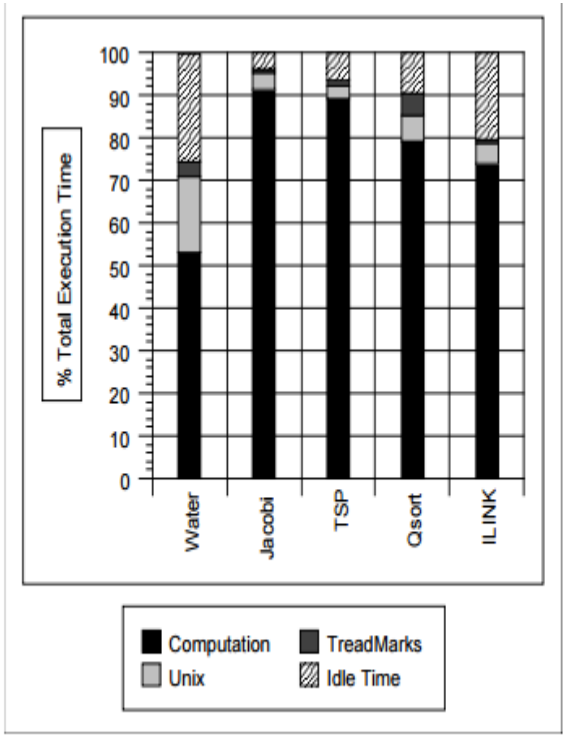


Figure 5 TreadMarks Execution Time Breakdown

Unix Overhead

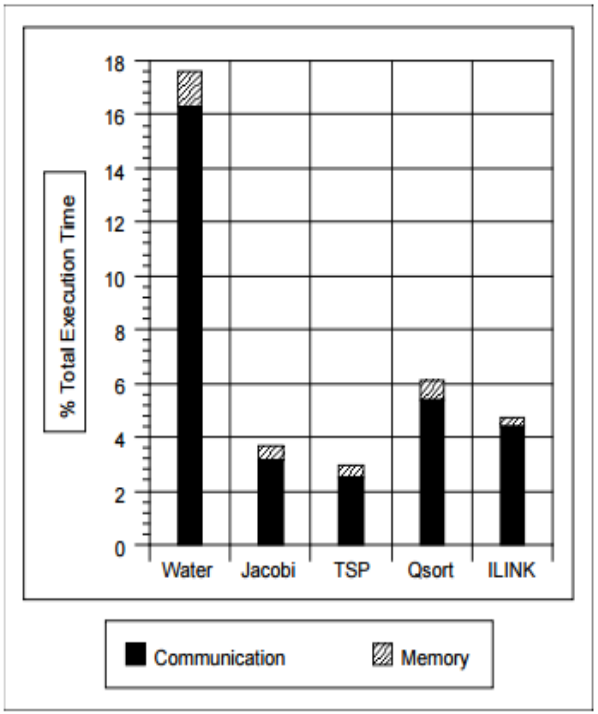


Figure 6 Unix Overhead Breakdown

TreadMarks Overhead

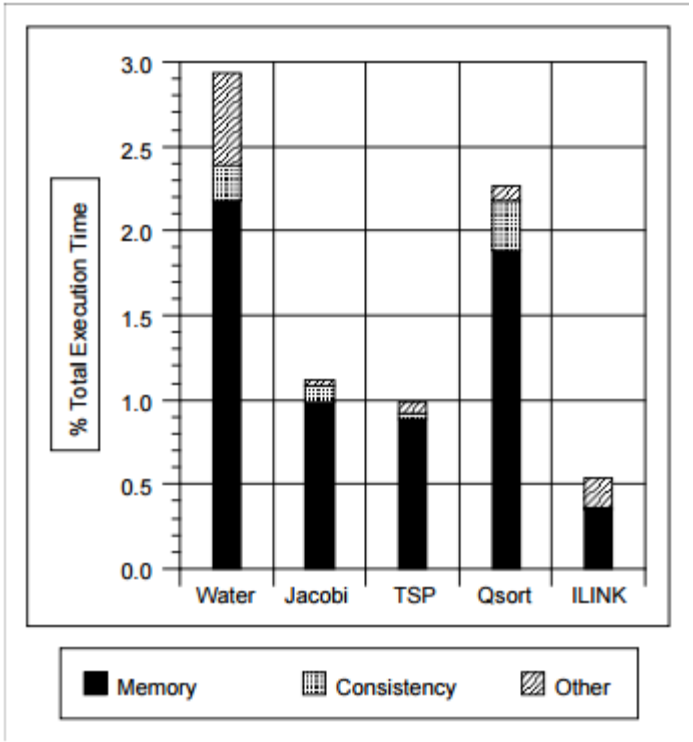


Figure 7 TreadMarks Overhead Breakdown

Conclusion

- Efforts on reducing the cost of communication
  - Lazy release consistency
  - Multiple-writer protocols
  - Lazy diff creation

- User-level DSM is a viable technique for parallel computation on clusters of workstations connected by suitable networking technology