Distributed Shared Memory

EECS 591
Farnam Jahanian
University of Michigan

Reading List

• Supplemental Handout:
  – pp. 312-313, 333-353 from Tanenbaum Dist. OS text (dist. in class)

• What DSM?
• Concept & Design Issues
• Comparison of Shared Memory Systems
• IVY (page-based DSM)
• Munin (shared variable DSM)
What’s DSM?

• First introduced by Li & Hudak in 1986
  – Adv. of multi-processors: easier to program
  – Adv. Of multi-computers: easier to build

• Idea: provide the mechanism for a set of networked workstations to share a single, paged virtual address space.
  – A ref. to local memory is done in hardware → fast
  – Emulate multi-processor caches using MMU and OS: an attempt to ref. an address that is not local causes a page fault, trap to OS, msg. to remote node to fetch the page, and restart faulting instruction
  – Idea is similar to traditional VM systems

• Sounds great! What’s the problem?
  – Performance!!!
  – Simple but potential poor performance (similar to thrashing in uni-processors)

• What to do? Later, but first various forms of shared memory systems.

Comparison of Shared Memory Systems

• Figure 6-10 (supp. handout)

• Spectrum: (tightly-coupled to loosely-coupled architectures)
  – Single-bus multi-processor
  – Switched multi-processor
  – NUMA
  – Paged-based DSM
  – Shared-variable DSM
  – Object-based DSM

• Hardware-controlled caching vs. software-controlled caching
  – Cache (replication) managed by MMU, OS or language run-time system

• Remote access: enabled by hardware vs. software
• Transfer unit: cache block, page, data structure, object
Distributed Shared Memory

• IVY system (page-based DSN) used to illustrate DSM concept and issues:
  – Granularity of transfer
  – Replication
  – Consistency semantics
  – Write-update vs. invalidation
  – Finding the owner
  – Finding the copies
  – Replacement policy

Distributed Shared Memory Systems

a) Pages of address space distributed among four machines

b) Situation after CPU 1 references page 10

c) Situation if page 10 is read only and replication is used
Granularity of Transfer

• How large should the unit of transfer (sharing) be?
  – Word, block, page, segments of pages

• Trade-off:
  – Adv. large blocks: fewer # of page transfers by taking adv. of locality of ref.
  – Disadv. large blocks: false sharing

False Sharing

• False sharing of a page between two independent processes.
Replication and Consistency

- **No replication** → exactly one copy of each page and it moves around → sequential consistency is easy to maintain
- **Replicated read-only copies, single read-write copy** → sequential consistency is easy to maintain
- **Replicated read-write copies** → potential consistency problem
  (similar to a multi-processor case when a processor attempts to modify a word that is present in caches)

**Cache invalidation vs. write-thru** (Figure 6-27 in supp. handout)

DSMs typically use cache invalidation:
write-thru is expensive, very inefficient if many consecutive writes to the same page, difficult to enforce sequential consistency if multiple concurrent updates originating from several processors.

---

Page Ownership

```
  Stationary (fixed)       Migrating (Dynamic)
                      \      /  *
        Centralized Mgr.  Distributed Mgr.
                       \  /  *
                     Fixed Mgr.  Dynamic Mgr.
```

* Considered in IVY systems.
Manager = Directory Server
Finding the Owner

- Broadcast: asking for the owner to respond
  - Optimization: ask for copy of page in r/w mode
  - Overhead and scalability issues

- Designate a centralized manager (i.e. directory server)
  - Figure 6-28 (supp. handout)
  - Potential bottleneck

Finding the Owner (cont.)

- Fixed Distributed Manager
  - Multiple managers in the system
  - Split the work by partitioning the pages among managers
  - How to find the right manager?
    - user lower-order bits of the page as an index into a manager’s table, or use a hash function.
    - The manager uses the incoming requests to keep track of changes in the ownership.

- Dynamic Distributed Manager
  - In this scheme, manager = owner
  - Each process keeps track of the probable owner
  - Request sent to a probable owner is forwarded to the new owner
  - Periodically, location of new owner is broadcast to everyone to ensure updated tables
Finding all Copies to be Invalidated

- Broadcast: requires reliable msg. delivery

- Copyset maintained by owner/manager
  - to invalidate, send a msg. to each processor in the copyset; must wait for acks from everyone

  SEE HANDOUT FOR DETAILS ON READ & WRITE PROTOCOLS (DYNAMIC DISTRIBUTED MANAGER)

What about performance problems of DSMs?

- Don’t share the entire page. Share selected variables and data structures \(\rightarrow\) higher level of abstraction, reduce false sharing

- Exploit data types or semantics of sharing

- Relax consistency

SEE HANDOUT FOR OVERVIEW OF MUNIN SYSTEM