Virtual Shared Memory (VSM)

Basic Concepts

Virtual shared memory (VSM) was first developed in a Ph.D. thesis by Kai Li (1986). The idea is to implement a coherent shared memory on a network of processors without physically shared memory. The system uses virtual addresses for memory references. The VSM address space is organized in pages which can be accessed by any node in the system. A memory-mapping manager on each node views its local memory as a large cache of pages for its associated processor.

Pages that are marked read-only can have copies in the physical memory of other nodes. A page currently written may reside in only one local memory.

The page fault mechanism of the MMU is used to handle the access and an VSM manager called by the MMU trap can perform the consistency protocol and the copy of pages.

- page based access control
- page fault handler

- Home node
  The node holding the primary page

- primary page
  The page located at the home node, only copies of this page are distributed to other nodes

- copied page
  A Page which is a copy of a primary page

- write permission only for primary pages without copies
Virtual Shared Memory (VSM)

Read access

1. access to variable by a read => page fault
2. find the appropriate primary page (Node #j) and send a read request to this node
3. copy the page to the requester
4. enter new address into page table and restart page access

Page fault handler is software-based. The MMU traps into the OS on a page miss. Instead of transferring the missing page from swap space into memory, the page fault handler sends a message to the home node requesting a copy of the page. Upon reception of the answer, the page is stored at the appropriate location in main memory and the process can resume execution.

Functions of the page fault handler

- finding the source of state information by using a directory; a page has a "home" determined by (a part of) the virtual address
- finding the appropriate copies
- communicating with the copies and copy the page to the requester; setting page access rights
- rescheduling of the process to hide latency
Virtual Shared Memory (VSM)

Write access

1. access to variable by a write => page fault
2. find the appropriate primary page (Node #j) and send a write request to this node
3. messages to all nodes with read copies to enter new single write owner
4. store primary page to write requester
5. invalidate all copies
6. enter new address into page table and restart page access

Write access

The write access in a virtual shared memory (VSM) is much more complicated than a read. The single writer coherency scheme requires the invalidation of all copies handed out for read only and the move of the page to the write requester.
Virtual Shared Memory (VSM)

Example of a Coherency Protocol for VSM

States of the coherence protocol
- invalid: access not possible
- shared: only read-access allowed, copies are present on other nodes
- owned: write access possible, copies are not allowed
- local: state for multiple writer access scheme, see next page
Virtual Shared Memory (VSM)

Method for unification of partly changed copies

Update scheme for partial changed copies