

Module-4
CSEN 3104
Lecture 32
31/10/2019

Dr. Debranjan Sarkar

Cache Coherence

Cache coherence mechanisms

- Two protocol approaches
 - Shared bus: Snoopy protocol
 - Other interconnection schemes: Directory protocol

Cache coherence mechanisms: Snoopy Protocol

- Early multiprocessors used bus-based memory systems
- Bus allows all the processors to observe ongoing memory transactions
- If a bus transaction threatens the consistent state of a locally cached object, the cache controller can invalidate the local copy
- Protocols using this mechanism to ensure coherence is called Snoopy Protocol
- Each processor tracks sharing status of each block

Snoopy Bus Protocol

- Snoopy protocols achieve data consistency between the cache memory and the shared memory through a bus-based memory system.
- Policies used for maintaining cache consistency:
 - **Write-invalidate**
 - **Write-update**
- 'Write Invalidate' policy will invalidate all remote copies when a local cache block is updated
- 'Write Update' policy will broadcast the new data block to all caches containing a copy of the block
- We have three processors P_1 , P_2 , and P_n having a consistent copy of data element 'X' in their local cache memory and in the shared memory (Fig. a)

Snoopy Bus Protocol

- Processor P_1 writes X' in its cache memory using **write-invalidate protocol**
- So, all other copies are invalidated via the bus, denoted by 'I' (Fig. b)
- Invalidated blocks are also known as **dirty**, i.e. they should not be used
- The **write-update protocol** updates all the cache copies via the bus through broadcast mechanism
- The memory copy is also updated, if write-through caches are used (Fig. c)
- The memory copy is updated later at block replacement time, in case of write-back caches

Disadvantages of Snoopy Bus Protocols

- Write-invalidate protocol may lead to heavy bus traffic, caused by read-misses, resulting from the processor updating a variable and other processors trying to read the same variable
- Write-update protocol may update data items in remote caches which will never be used by other processors
- These problems pose additional limitations in using buses to build large multiprocessors

Cache coherence mechanisms: Directory Based Protocols

- To overcome the limitations of using bus in building large (scalable) multiprocessor system, multistage network is used to interconnect processors
- Unlike the bus-based system, the bandwidth of these networks increases as more processors are added to the system
- Broadcasting is very expensive in a multistage network
- Hence, the consistency commands is sent only to those caches that keep a copy of the block
- Such networks do not have a convenient snooping mechanism and do not provide an efficient broadcast capability
- This is the reason for development of directory-based protocols for network-connected multiprocessors to solve the cache coherence problem

Directory based Protocols

- Sharing status of each block kept in one location
- In a directory-based protocol system, data to be shared are placed in a common directory that maintains the coherence among the caches.
- Here, the directory acts as a filter where the processors ask permission to load an entry from the primary memory to its cache memory.
- If an entry is changed the directory either updates it or invalidates the other caches with that entry.

Directory based Protocols

- *Directory based protocols* have a main directory containing information on shared data across processor caches
- The directory works as a look-up table for each processor to identify coherence and consistency of data which is currently being updated
- A directory-based protocol is a smart way of implementing cache consistency on an arbitrary interconnection network
- While the resulting protocol is complex, it is indeed tractable
- Moreover, the hardware needed to implement such a protocol is quite reasonable for the scale of machine in which it is expected to be used

Directory based Protocols

- Various directory-based protocols differ mainly in:
 - how the directory maintains information, and
 - what information it stores
- Central Directory based protocol
 - Uses a central directory which contains duplicates of all cache directories
 - The central directory provides all the information to enforce consistency
 - It is usually very large in size
 - Must be associatively searched
 - Chance of bottleneck
 - Drawbacks for a large multiprocessor system:
 - Contention, and
 - Long search time

Directory based Protocols

- Distributed Directory based protocol
 - Each memory module maintains a separate directory
 - Each directory records the state and presence information for each memory block
 - The state information is local
 - The presence information indicates which caches have a copy of the block
 - Bottleneck is avoided
- Directory based protocols do not use broadcasts
- So, the locations of all cached copies of each block of shared data must have to stored
- The list of cache locations is called a cache directory
- It may be centralized or distributed
- A directory entry for each block of data contains a number of pointers to specify the locations of copies of the block
- Each directory entry also contains a dirty bit to specify whether a particular cache has permissions to write the associated block of data

A few References and Bibliography

- Computer Architecture and Organization *by* John P. Hayes, (WCB/McGraw Hill)
- Advanced Computer Architecture: Parallelism, Scalability, Programmability *by* Kai Hwang & Naresh Jotwani (Tata McGraw Hill Education Pvt. Ltd.)
- Computer Architecture and Parallel Processing *by* Kai Hwang & Faye A. Briggs (McGraw Hill Book Company)

Thank you