Cache Coherence Protocol for Hierarchical shared bus multiprocessor

Aditya Giriyalkar & Tejaswini Ananthanarayana
Outline

- Introduction.
- Motivation.
- Hierarchical Shared Bus Multiprocessor.
- BSCP
- ESCP
- SMCP
- Results and analysis.
- Conclusion.
Introduction

- Reducing memory access time in order to improve performance of multiprocessors.
- Memory latency hidden by usage of caches.
- Cache introduce cache coherence problems.

Thus we go ahead with single copy cache coherence protocols for multiprocessors with a hierarchical bus structure.
Motivation.

Topics covered so far:

- Scalable Cache Coherence.
  - Directory-based cache coherence.
  - A combination of hierarchical and directory-based approach.

Topic remaining to be covered:
  Hierarchical shared bus multiprocessor.
Hierarchical Systems

- Hierarchical systems:
  - Effective.
  - Scalable.
- In such a system, adding more processors means adding one or more buses which increase the total bandwidth of the system.
Advantages of hierarchical buses

- It combines the simplicity of a single bus cluster with the scalability of multistage interconnection network;
- Thus avoiding the bottleneck of connecting a large number of processors to a single bus.
Hierarchical shared bus multiprocessor
System Consists of – processors with its own cache.
  - Level – 1 cache or cache’s processors.

At level – 1 – each k processors are attached to a single bus.
  - Forming a first-level cluster.

Every k first-level clusters are attached via k caches to a single bus.
  - Intermediate or shared cache.
  - A second-level cluster, and so on.

The number of processors = \( k^H \)
   where \( H \) is the number of levels in the system.
Three variations of a single copy cache coherence protocol

- BSCP – Basic Single Copy Protocol.
- SMCP – Single Multiple Copies Protocol.
BSCP, ESCP & SMCP.

- **Data blocks** – reside only in the processor’s cache.
- **Intermediate cache** – Information about blocks in the descendant cache.
  - Helps in relocating block quickly
- **Data blocks** – reside anywhere in the system.
- **Intermediate cache** – holds data blocks as well as information about the blocks in descendant cache.
- Multiple copies of the same block within a single level-1 cluster is allowed.
- In the same cluster, a snoopy cache coherence protocol is used to guarantee the consistency of the data
More on BSCP

- Each processor fetches instruction and data from its own cache.
- In case of a cache miss, the processor sends a request to get the missing block.
- The request travels up the hierarchy.
Problems with BSCP

- The delay encountered by the NAK in its way to the processor when the block gets removed while it is waiting for the bus.
Methods to avoid the above scenario.

- LRU replacement policy used.
- The state of the requested block will be the same as in the source cache.
Another way..

Three different priority levels can be assumed:

- The lowest priority – a request from a cache for a block that traveling up in the network.
- The second highest priority – a cache request traveling down the hierarchy.
- The highest priority – a moving data block, a write-back data block, a negative acknowledgment, or adding or removing a directory entry.
Avoids ping-pong effect, where a block is referenced by 2 processors simultaneously.

According to BSCP, this block will be bouncing back and forth between these two processor’s caches.
- Every cache maintains a list with the a Most Recently Used MRU blocks as proposed by the author.
- If processor \( i \) issued a memory request and could not be found in its local cache, the request propagates up in the network till the block containing the requested data is located.
The block could be in one of four locations, the action taken depends on the location of the requested block.

1. The block is found in the main memory
2. The block is found in processor j cache, but not in $\text{MRU}_j$.
3. The block is found in processor j cache and is in $\text{MRU}_j$.
4. The block is found in an intermediate cache.
Detailed SMCP

- In this protocol, data is allowed only in the processor’s cache.
- The intermediate caches hold information about the location of each data block in the tree rooted at this particular intermediate cache except level2 caches.
Level-2 caches contain tables with two fields per entries.

- The first is the block number for all the block in the level-1 cluster.
- The second entry is the number of copies of this block in level-1 cluster.

A snoopy controller monitors the bus and changes the number of copies for each block according to the transaction on the bus.
Each block in any processor cache could be in two states: either SHARED, or EXCLUSIVE.

- A shared block means that there are more copies of this block in the same cluster.
- Exclusive means it is the only copy in the cluster.

Also each block could be:
- clean (not modified)
- dirty (modified).
The protocol can be best described by following the actions of the snoopy controllers in 2 cases:

- Processors read or write
- Requests from another processors.
Actions during processor Read or Write.

- Read hit.
- Write hit.
- Read miss.
- Write miss.
The actions by the snoopy controllers in response to requests from other processors

- Invalidate.
- Read.
- Write.
- Replace: If a block is to be replaced, the actions taken depends on the state of the block.
  - Shared: The block is discarded and the higher level cache decrement the number of copies for this block.
  - Exclusive: the block is written back to the main memory, and level-2 cache invalidates its own entry for this block.
Results and analysis.

- To perform analysis, a process oriented discrete simulation package with C language was used by the author.
- Total number of processors: 27.
- Total Memory: 1MB.
- Cache Size: 128 blocks.
- Block size: 16 Bytes
- MRU list size: 8
- $\beta$ is the probability that reference is to one of the blocks in MRU list.
Figure 4: Average access time vs. incremental probability for $\beta = 0.8$, $p_w = 0.5$

Figure 6: Average access time vs. incremental probability for $\beta = 0.2$, $p_w = 0.5$
Figure 2: Average access time vs. the incremental probability for $\beta = 0.8$, $p_w = 0.2$
Conclusion

- The results have proved that the performance of these three protocols exceeds the performance of directory-based protocol proposed for a wide range of workload.
- It is also observed that the proposed approach is scalable up to a certain level of hierarchy after which the performance degrades and directory-based protocol is preferred.
References


- C. Anderson and J.-L. Baer, A Multi-Level Hierarchical Cache Coherence Protocol for Multiprocessors “Proceedings of the 7th International Symposium on Parallel Processing”.