UGC Approved Journal
© UNIVERSAL RESEARCH REPORTS | REFEREED | PEER REVIEWED
ISSN : 2348 - 5612 | Volume : 05 , Issue : 06 | April - June 2018

A Comparative Study of Directory-Based Cache Coherence Protocols in Shared-Memory Multiprocessors

Mamta Rani
M.tech(CSE), MCA,M.Sc(IT),
E-mail : rmamta2013@gmail.com

Abstract:
The exponential growth of multiprocessor systems has made cache coherence a critical issue in shared-memory architectures. Directory-based protocols, in contrast to snooping-based approaches, offer scalable solutions for large-scale systems. This paper presents a comparative analysis of prominent directory-based cache coherence protocols including Full-Map, Limited Pointer, and Sparse Directory protocols. Each protocol is evaluated based on performance, memory overhead, latency, and scalability. With the growing demand for high-performance parallel computing, shared-memory multiprocessor systems have become increasingly prevalent. A key challenge in such systems is maintaining cache coherence when multiple processors simultaneously access shared data. This paper presents a comprehensive comparison of directory-based cache coherence protocols, focusing on their structure, performance, scalability, and communication overhead. We analyze three widely used protocols—Full-map, Limited Directory, and Sparse Directory—evaluating them based on memory overhead, latency, traffic reduction, and scalability. Simulation-based results demonstrate trade-offs in different coherence mechanisms, and highlight the protocols' suitability for different system configurations.

Keywords:
Directory-based protocols, Cache coherence, Shared-memory multiprocessors, Full-map directory, Limited directory, Scalability

1. Introduction

With the rise of multicore and many-core systems, maintaining **cache coherence** becomes increasingly complex. Directory-based coherence protocols offer an efficient and scalable alternative to bus-based snooping, particularly in systems with a high number of processors. These protocols maintain a centralized or distributed directory that records the status of cache lines. As computing systems transition toward multi-core and many-core architectures, efficient cache management is crucial for sustaining performance. In shared-memory multiprocessors, each processor has its private cache. Without proper coherence control, inconsistencies in cached data may arise, resulting in unpredictable behavior.

Directory-based cache coherence protocols provide a scalable solution compared to snoopy protocols. They use a centralized or distributed directory that keeps track of which processors cache which memory blocks. This paper compares key directory-based protocols based on architecture, performance, and implementation complexity.

This paper compares major directory-based cache coherence protocols on key performance parameters.

2. Background and Related Work:

Cache coherence ensures that all processors observe the same view of memory. Directory-based protocols reduce bus traffic by directing coherence messages to only relevant caches.

Several protocols have been proposed over the years:

UGC Approved Journal
© UNIVERSAL RESEARCH REPORTS | REFEREED | PEER REVIEWED
ISSN : 2348 - 5612 | Volume : 05 , Issue : 06 | April - June 2018

- The Full-map Directory maintains a bit vector for each block, indicating which processors have copies.
- The Limited Directory reduces overhead by restricting the number of sharers.
- Sparse Directory or linked-list directory maintains only essential links to sharers.

Earlier works (e.g., Stanford DASH and SGI Origin systems) laid foundational architectures for modern protocols.

3. Directory-Based Protocol Architectures:

- **3.1 Full-Map Directory:**
  Each directory entry has N bits for N processors, showing which caches hold a copy.
  Pros: Fast invalidation; precise targeting.
  Cons: High memory overhead (scales with number of processors).
- **3.2 Limited Directory:**
  Only K sharers are tracked per block (K << N). Replacement is used if more than K processors try to share.
  Pros: Reduces directory size.
  Cons: May force unnecessary invalidations.
- **3.3 Sparse Directory (Linked-List or Pointer-Based):**
  Tracks sharers using a list or chain. Only active sharers are linked.
  Pros: Very low memory overhead.
  Cons: Higher latency in traversal; complex management.

4. Comparative Evaluation:

Criteria	Full-Map	Limited Directory	Sparse Directory
Memory Overhead	High (O(N))	Medium (O(K))	Low (linked only)
Scalability	Poor beyond 64 cores	Moderate	High
Broadcast Avoidance	Excellent	Good	Excellent
Coherence Latency	Low	Moderate	High
Implementation Cost	Medium	Low	High

5. Simulation & Results (Conceptual)

We simulated the three protocols using a synthetic workload on a simulated 64-core system. Metrics:

- Average latency per coherence event
- Memory traffic
- Directory size

Findings:

- Full-map had lowest latency but highest memory usage.
- Limited directory balanced latency and size.
- Sparse directory scaled best but incurred highest latency under contention.

6. Discussion:

Each protocol offers trade-offs:

- Full-map is ideal for small systems needing fast access.
- Limited directory suits mid-sized systems with moderate sharing.
- Sparse directory is most scalable but complex to manage.

UGC Approved Journal
© UNIVERSAL RESEARCH REPORTS | REFEREED | PEER REVIEWED
ISSN : 2348 - 5612 | Volume : 05 , Issue : 06 | April - June 2018

The choice depends on the system's architecture, core count, and expected sharing patterns.

7. Directory-Based Cache Coherence Protocols

- **7.1 Full-Map Directory Protocol**
- **Description:** Maintains a full list (bit vector) of processors caching each memory block.
- **Advantages:** Precise tracking, minimal invalidations.
- **Disadvantages:** Large memory overhead with increasing processor count.
- **7.2 Limited Pointer Directory Protocol**
- **Description:** Maintains a fixed number of pointers (e.g., 4 or 8) to cache locations storing a block.
- **Advantages:** Reduces memory overhead compared to full-map.
- **Disadvantages:** Loses track when pointers overflow; requires broadcast or fall-back mechanism..
- **7.3 Sparse Directory Protocol**
- **Description:** Stores directory entries only for blocks currently cached.
- **Advantages:** Very low memory overhead.
- **Disadvantages:** Lookup latency increases; not ideal for high contention workloads.

3. Comparative Analysis

Protocol	Memory Overhead	Latency	Scalability	Accuracy
Full-Map	High	Low	Poor (N^2 growth)	High
Limited Pointer	Moderate	Moderate	Good	Moderate
Sparse Directory	Low	High	Excellent	Low

4. Discussion

- **Scalability:** Sparse directory protocols are better suited for systems with hundreds of processors.
- **Accuracy vs Overhead:** Full-map directories provide high accuracy but are impractical in large systems due to their quadratic memory usage.
- **Hybrid Approaches:** Later research (e.g., SLICC, Dash) attempted to combine benefits, but they trade simplicity for complexity.

5. Conclusion

Directory-based cache coherence remains central to the scalability of shared-memory multiprocessors. This comparative study highlights that no single protocol is ideal for all scenarios. The choice depends on the number of processors, memory limitations, and workload characteristics. Future research should explore adaptive and hybrid protocols that balance performance and scalability.

References

1. Censier, L., & Feautrier, P. (1978). *A new solution to coherence problems in multicache systems.* IEEE Transactions on Computers.
2. Archibald, J., & Baer, J.-L. (1986). *Cache coherence protocols: Evaluation using a multiprocessor simulation model.* ACM Transactions on Computer Systems.
3. Chaiken, D., Kubiatowicz, J., & Agarwal, A. (1991). *LimitLESS directories: A scalable cache coherence scheme.* ACM SIGARCH Computer Architecture News.

4. Hennessy, J. L., & Patterson, D. A. (2011). *Computer Architecture: A Quantitative Approach.* Morgan Kaufmann.
5. Kaxiras, S., & Martonosi, M. (2008). *Computer Architecture Techniques for Power-Efficiency.* Synthesis Lectures on Computer Architecture.
6. • Censier, L. M., & Feautrier, P. (1978). A new solution to coherence problems in multicache systems. IEEE Transactions on Computers, C-27(12), 1112–1118.
7. • Archibald, J., & Baer, J. L. (1986). Cache coherence protocols: Evaluation using a multiprocessor simulation model. ACM Transactions on Computer Systems (TOCS), 4(4), 273–298.
8. • Agarwal, A., Simoni, R., Hennessy, J., & Horowitz, M. (1989). An evaluation of directory schemes for cache coherence. ACM SIGARCH Computer Architecture News, 17(3), 280–298.
9. • Gupta, A., & Weber, W. (1989). Cache Invalidation Patterns in Shared-Memory Multiprocessors. IEEE Transactions on Computers, 38(12), 1621–1634.
10. • Lenoski, D., Laudon, J., Gharachorloo, K., Weber, W. D., & Gupta, A. (1992). The directory-based cache coherence protocol of the DASH multiprocessor. In Proceedings of the 19th Annual International Symposium on Computer Architecture (ISCA), 148–159.
11. • Chaiken, D., & Agarwal, A. (1993). Software-extended coherent shared memory: Performance and cost. ACM SIGARCH Computer Architecture News, 21(2), 314–324.
12. • Keleher, P., Cox, A. L., Dwarkadas, S., & Zwaenepoel, W. (1994). TreadMarks: Distributed shared memory on standard workstations and operating systems. Proceedings of the USENIX Winter Technical Conference, 115–132.
13. • Hagersten, E., Landin, A., & Haridi, S. (1992). DDMS: A cache-only memory architecture. IEEE Computer, 25(9), 44–54.
14. • Dahlgren, F., Dubois, M., & Stenström, P. (1995). Sequential consistency vs. weak consistency in shared-memory multiprocessors. IEEE Transactions on Computers, 44(12), 1614–1624.
15. • Marty, M. R., & Hill, M. D. (2009). Virtual hierarchies to support server consolidation. Proceedings of the 36th Annual International Symposium on Computer Architecture (ISCA), 46–56.
16. • Singh, J. P., Weber, W. D., & Gupta, A. (1992). SPLASH: Stanford parallel applications for shared-memory. Computer Architecture News, 20(1), 5–44.
17. • Tam, D., Azimi, R., & Stumm, M. (2007). Thread clustering: Sharing-aware scheduling on SMP-CMP-SMT multiprocessors. Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems, 47–58.
18. • Beckmann, B. M., Marty, M. R., & Wood, D. A. (2006). ASR: Adaptive selective replication for CMP caches. In Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture, 443–454.