Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 5, No.4, May : 2025

A SURVEY ON CACHE CONSISTENCY MODELS IN MODERN MICROPROCESSORS

Sonali Antad, Professor, Computer Engineering Department, Vishwakarma Institute of Technology, Pune, Maharashtra.
Adnaan Momin, Bachelor of Technology Students, Computer Engineering Department, Vishwakarma Institute of Technology, Pune, Maharashtra.
Tejas Abhang, Bachelor of Technology Students, Computer Engineering Department, Vishwakarma Institute of Technology, Pune, Maharashtra.
Nirwani Adhau, Bachelor of Technology Students, Computer Engineering Department, Vishwakarma Institute of Technology, Pune, Maharashtra.
Akshat Gupta, Bachelor of Technology Students, Computer Engineering Department, Vishwakarma Institute of Technology, Pune, Maharashtra.

ABSTRACT
Cache consistency plays a crucial role in the performance and reliability of modern microprocessors, especially in multi-core and distributed systems Cache consistency is a hot topic in the microprocessor world, particularly with multi-core and distributed systems. As processors continue to evolve to support higher degrees of parallelism and more elaborate workloads, guaranteeing coherence across multiple levels of caches is becoming both more necessary and more difficult. This paper is a broad survey of cache consistency models used in modern microprocessors and their characteristics, technicalities, and trade-offs (for example, MESI, MOESI and relaxed memory consistency models) and their awareness, understanding and perceptions from public and experts. Using a structured survey and targeting both technical and non-technical respondents, we analyze the awareness and perceived importance of cache coherence as well as concerns around its usability. We describe the gap between hardware-level mechanisms and those available and accessible to wider audiences, ultimately offering prescriptive avenue.

Keywords:
Cache consistency, cache coherence, MESI, MOESI, memory models, microprocessors, multi-core systems, memory consistency models, survey, processor architecture

I. Introduction
In the era of multi-core processors and high-speed computing, cache memory plays an important role in enhancing system performance by reducing its time required to access frequently used data. As modern processors execute instructions in parallel and share data among multiple cores, maintaining a consistent view of memory becomes a significant challenge. This has led to the development of various cache consistency models and cache coherence protocols, which ensure correctness in data sharing and memory access across cores.
Cache consistency refers to the ability of a system to ensure that all processors in a multi-core architecture observe memory operations in a coherent and predictable manner. Without effective consistency models, systems risk issues like stale data reads, incorrect instruction execution, or data corruption, especially in concurrent processing environments.
To the date there has been significant research at theoretical and architectural levels on cache memory and coherence protocols (e.g., MESI, MOESI, directory based, snooping based systems, etc.) however, to the authors knowledge there have been no prior studies into the level of understanding of these models by the general technical or non-technical communities. To address this gap, this study conducts a wide-ranging survey across this domain of stakeholders, with particular emphasis on professionals with computer architecture backgrounds versus general users of computing devices. In this paper, we first evaluate the knowledge of the technical aspects of cache memory. We will examine the results of the survey, with an eye toward understanding the current trends in cache management execution in

both hardware and software systems, and areas of possible misunderstanding, as well as directions for improvement.

II. Literature

The work by Kumar and Arora presents an in-depth survey of cache coherence protocols employed in shared memory multiprocessor systems [1]. The authors classify coherence solutions into hardware and software models and discuss their respective merits and drawbacks. Another type of hardware-based protocols like snoopy, Directory-based protocols are highly scalable protocols and support systems with thousands of processors. On the other hand, software-based protocols such as MSI, MESI, MOSI, MOESI, and Dragon present low-cost solutions, but they only partially address coherence issues in real-time systems. It is shown that using an SMP Cache simulator, the Dragon protocol produces higher cache hit rates. Newer optimizations like the CSC protocol appear to offer dramatic reductions in execution time by utilizing a Shared Coherence Cache (SC-cache) as opposed to traditional MESI and Dragon implementations. This is a critical evolution in cache consistency models in that hybrid approaches strive for a balance of performance, cost and scalability for contemporary multiprocessor architectures.

Al-Waisi and Agyeman [2] provide a comprehensive overview of on-chip cache coherence protocols tailored for chip multiprocessor (CMP) systems. The authors classify coherence solutions into hardware and software models and discuss their respective merits and drawbacks. Another type of hardware-based protocols like Snoopy, Directory-based directly also maintains the coherence during the run-time but the cost of implementation is quite high. Directory-based protocols are highly scalable protocols and support systems with thousands of processors. On the other hand, software-based protocols such as MSI, MESI, MOSI, MOESI, and Dragon present low-cost solutions, but they only partially address coherence issues in real-time systems. Additionally, the MECSIF protocol combines snoopy and directory schemes to improve efficiency and reduce coherence traffic. This is a critical evolution in cache consistency models in that hybrid approaches strive for a balance of performance, cost and scalability for the contemporary multiprocessor architectures.

Yu et al. [3] present Soul, a new synchronization framework that generalizes cache coherence in a spatial and temporal manner to enhance scalability in disaggregated shared memory architectures. The authors claim that conventional cache-coherent substrates lead to unnecessary inter-cache communications if used to implement synchronization primitives such as locks. As a solution for that they suggest the Generalized Cache-coherence Protocol (GCP) where synchronization is integrated directly into the cache coherence layer. GCP adds wait queues for temporal generalization and cache lines of arbitrary sizes for spatial generalization, to efficiently support lock-based synchronization. The work stresses that synchronization primitives are a fundamental extension of cache coherence principles and exploits hardware-software co-design. Soul's support for relaxed memory models such as TSO makes it generalizable to contemporary microarchitectures. This work finds new ways to drive the design of the cache consistency model by adding synchronization primitives directly to the coherence protocols, which is particularly true for new hardware like CXL.

Shukur et al. [4] give a review about cache coherence protocols used for distributed shared memory multi-processor systems, focusing on general and particular impact to the overall system performance. The paper classifies coherence protocols into two basic methodologies namely snoopy-based and directory-based protocols. It looks comprehensively at traditional protocols such as MSI, MESI, MOSI, and MOESI, where these protocols states and efficiency in providing consistency across the caches are outlined. MOESI is particularly commended for its low bus traffic and support of multiple cache participants. The design challenges of these protocols, especially in distributed and multicore systems, where data consistency issues become more involved, are presented by the authors. Their comparative study of the five protocol alternatives demonstrates the implications of protocol selections on scalability, power, latency, and coherence traffic.

Tian et al. [5] presented a new cache coherence protocol for multicore computing systems to overcome problems with caches in bus listening and directory-based coherence methods. A novel architecture using a D-Cache virtual bus to achieve point-to-point consistency transaction transmission is proposed, which effectively relieves the bus idle problem due to broadcast. This design speeds up access by placing D-Cache in proximity to processor cores and then using request gather units for caching transactions. Experimental results on GEMS with SPLASH-2 benchmarks (e.g., LU, Ocean, FFT) show that we have obtained an average 3.84%performance improvement over MESI protocol. The results confirm that the proposed approach can increase bus utilization and decrease latency. Although promising overall, this paper details limitations and suggests more research of hybrid designs towards cache management. This effort makes a major step forward toward efficient cache coherence mechanisms in multi-core architectures.

Since the advent of shared-memory multiprocessors, designers have worked on solving the problem of maintaining cache coherence with a set of private caches. Yousif et al. [6], where both hardware, and software solutions to the cache coherence problem are surveyed. The paper is structured in such a way that it classifies hardware protocols according to the underlying interconnection networks bus based, crossbars, and multistage interconnection networks (MIN)—and it presents the corresponding state diagram models for coherence management. It also underscores the transition of full-map directory protocols to space-efficient schemes like Archibald's two-bit and three-bit protocols. It addresses software-based coherence mechanisms, coherence correctness models and protocol performance analysis, making it an ideal reference for advanced students and researchers in parallel and distributed systems as well as system architects.

III. Survey Methodology

A. Survey Distribution

The final survey was built in Google Forms, a treasure that was chosen for its easy-to-form layout and smooth data collection process. The survey was shared digitally over various online platforms to ensure an expansive and diverse reach. These ranged from college WhatsApp groups to Telegram channels, Instagram stories, LinkedIn posts, computer science and engineering technical forums. This ensured that our multi-channel approach maximized the reach of our technical event to both technical and non-technical audiences.

Voluntary participation was emphasized, and all responses were anonymous. Participants were told that they would not record any personally identifiable information, such as their names, e-mails, This was to encourage genuine feedback and maintain the anonymity of all respondents. The survey was open for 7 weeks and resulted in 135 valid log entries for analysis.

B. Target Audience

The survey was designed to be accessible to a wide and varied audience, technical and non-technical alike, especially in academic environments. The respondents included undergraduates and post graduates, along with faculty from branches of engineering, science, and humanities. The diversity of the study participants made it possible to capture a wide range of views on cache memory, from deep technical insight to a general system performance perspective.

The age brackets (Under 18, 18–24, 25–34, 34+) segment of participants and educational backgrounds were framed in such a way that our data can express generational differences and differences based on university-level education. The entry process was intentionally designed to be engaging and was targeted to audiences with no technical background, enabling wider participation and providing insight into real-world knowledge gaps. This study scoped the awareness and understanding of cache consistency models in a variety of users.

C. Data Collection and Management

All responses were automatically recorded and securely stored through the Google Forms platform. Exporting the data into spreadsheets and carrying out an initial cleaning to eliminate duplicates, incomplete responses and standard categories for entry conformity. Data were analyzed using simple

descriptive statistics to classify users' responses and discover trends in their knowledge of cache memory and consistency models in modern microprocessors.
The questions were grouped into key thematic areas for targeted analysis:

- Demographics and Background
- Cache Memory Awareness
- Cache Consistency Understanding
- Impact of Cache on System Performance
- Programming and Cache Optimization
- Future Trends in Cache Technology

In this manner, we could perform systematic study for what users of cache consistence models were acquainted about it in the field and what they didn't and how it associated with microprocessor execution.

D. Ethical Considerations

This survey was ensured the ethical principles to protect the privacy and confidentiality of all respondents. Participation was completely voluntary and did not require the responding individuals to provide personal identifying information, such as a name, contact information, or device-specific identifiers. The name and the e-mail ID was given but they were not mandatory; hence everyone remained anonymous. Demographic data, like education and age, were needed to help analyze changing trends for different user groups.
No data was collected to identify respondents, and the data collected will be used only for academic/research purposes and not for commercial use. Participants were told beforehand that the aim of the survey and how they would help analyze cache consistency models in today's microprocessors. Participants provided implicit consent by agreeing to participate, with confidentiality and security of the responses being guaranteed.
The data was securely stored, and only the research team had access to it for analysis.

IV. Survey Questions Overview

The survey questions have been designed to assess a demographic & Background, Technical skill & Ability, and User experience & perception in order to approach a formalized review. Breaking this down into concepts allows the analysis of user understanding from technical knowledge to physical use of the device. All questions were designed to assess specific obstacles of cache memory awareness and consistency. The survey was also designed into two separate sections according to respondent technical background. The Technical Section is primarily aimed at people with knowledge of, or interest in, computer architecture and related disciplines. It covers various aspects of cache memory and consistency models. It helps in evaluating the level of understanding and practical experience for technically minded participants.
The Non-Technical Section deals with general users and discusses device performance, multitasking behavior, and awareness of cache memory in regular devices, e.g., smartphones, laptops, etc. It captures user perceptions on speed, responsiveness, and the trade-off between performance and accuracy, offering valuable insights into how cache-related behaviors are experienced and interpreted by non-technical users.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 5, No.4, May : 2025

Category	Focus Area	Question Numbers	Purpose
Demographics & Familiarity	Background info, tech comfort, and cache awareness	Section 1: Q1–Q7	To segment users by age, profession, and baseline familiarity with cache memory.
Technical Understanding	Cache memory, consistency models, protocols, system impact	Section 2: Q1–Q12	To assess knowledge of microarchitecture concepts and protocol effectiveness.
Practical Technical Experience	Optimization practices and real-world observations	Section 2: Q13–Q15	To evaluate exposure to cache optimization in programming or development.
Device Usage & Memory Behavior	User experience with multitasking, app loading, and performance	Section 3: Q1–Q4	To understand how everyday users perceive and are affected by caching behavior.
Perception of Cache Importance	Awareness of cache function in devices	Section 3: Q5–Q10	To explore how users connect cache to device speed, responsiveness, and efficiency.
User Preferences & Priorities	Performance vs. accuracy, purchasing influence	Section 3: Q11–Q13	To examine user values regarding speed, data accuracy, and decision factors.

Figure 1. Classification of Survey

V. Results And Analysis

The survey received responses from 135 participants, segmented into 86 technical and 49 non-technical users. The largest age group was 18–24 years (59.3%), primarily composed of students (39.3%), followed by professionals in engineering, IT, and software development. Educationally, 45.2% of participants held a bachelor's degree, suggesting a generally well-informed respondent base with a foundation in computing concepts.

A majority of technical participants demonstrated familiarity with standard cache coherence protocols. Figure 1 illustrates that MESI (Modified, Exclusive, Shared, Invalid) was the most recognized protocol, followed by MOESI and directory-based models. When asked about effectiveness, MESI and directory-based models were rated highest, with many respondents highlighting the scalability and efficiency of directory-based approaches in large-core systems.

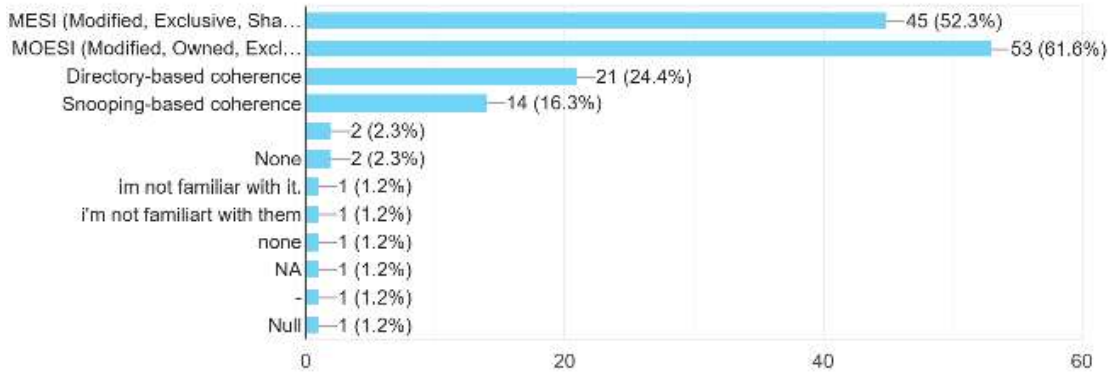

Figure 2. Familiarity with Cache Coherence Protocols among Technical Users

A prominent outcome among technical users was the positive impact of cache consistency mechanisms. A majority stated that cache consistency significantly improves system performance. Others noted a moderate or incremental improvement. Only a negligible portion found no benefit or were unsure.

Some of the technical respondents had worked directly with cache optimization or memory tuning techniques. However, most expressed skepticism about the effectiveness of current compilers in leveraging cache structures efficiently. This perception is captured in Figure 2, where only a minority believed modern compilers are well-optimized for cache usage.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 5, No.4, May : 2025

Figure 3. Perceived Effectiveness of Compilers in Utilizing Cache Memory

This points to a perceived disconnect between software-level optimization and hardware capabilities, emphasizing the need for more integrated system design. When asked where cache optimization has the greatest impact, technical users pointed to AI & Machine Learning, Gaming, and Cloud Computing, as shown in Figure 3.

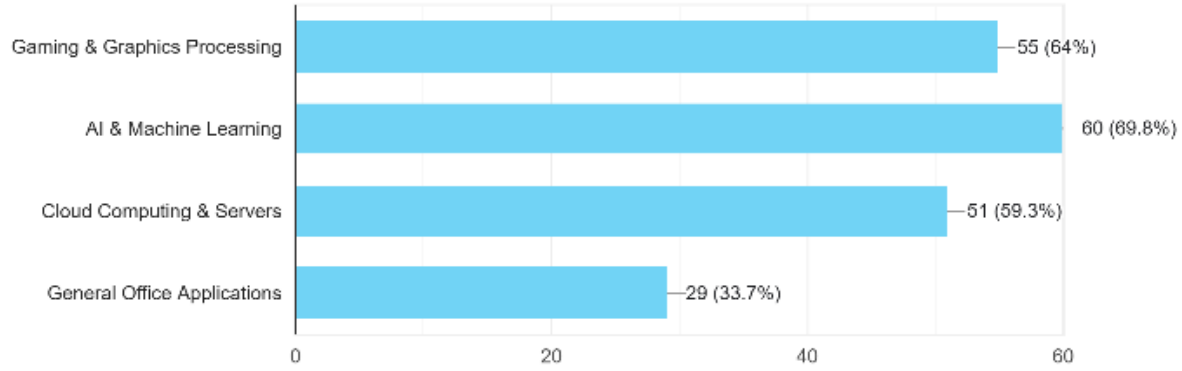

Figure 4. Areas Where Cache Optimization Has the Most Perceived Impact

From the 29 open-ended technical responses, common cache consistency limitations emerged—scalability challenges in many-core environments, latency and performance overhead, synchronization bottlenecks, and complexity of protocol implementation. Participants proposed several forward-looking solutions, such as adaptive hybrid protocols that switch based on workload, AI/ML-based prediction for coherence management, compiler-assisted memory behavior optimization, and NUMA-aware protocols with software-hardware co-design.

Future innovation directions included AI/ML-driven coherence protocols, dynamic workload-aware models, compiler-assisted cache tuning, 3D-stacked memory, and hybrid directory-snooping coherence models.

While not all non-technical participants were familiar with the intricacies of cache memory, a substantial portion correctly identified its role in accelerating data access. This basic awareness is essential, especially as device performance becomes increasingly linked to memory efficiency. Many non-technical users cited issues such as slowdowns, app freezes, and frequent restarts, which—although anecdotal—align closely with technical concerns about poor memory management.

The results show a mismatch between user experience and system-level knowledge. Technical users recognized cache consistency as an integral concept in both system architecture and performance, while non-technical users experienced symptoms without understanding the mechanisms that caused them. This makes clear a need for educational tools and for transparency in how these systems behave. Implemented correctly, this balance means we can deliver innovations at the system level while also allowing understanding and access to the hardware innovations and their real-world implications across our user cohorts.

VI. Limitations

Some of the limitations should be acknowledged in this informative study. Since the respondent pool included mostly younger users, particularly the students, it does not necessarily reflect the more general user base. Moreover, self-selection bias may have influenced the findings, as those with an interest in system performance may have been more inclined to participate. The open-ended responses, though insightful, were limited in depth, and the technical section assumed a certain level of prior knowledge, potentially excluding nuanced feedback from intermediate users.

VII. Conclusion

This survey-based study explored user perceptions and understanding of cache consistency models in modern microprocessors, comparing insights across technical and non-technical audiences. The results highlight that while technical users recognize the performance impact and architectural challenges of cache consistency, non-technical users experience its effects more passively, through symptoms like slowdowns and freezes. These findings emphasize the importance of bridging the gap between system-level optimization and everyday user experience. Future directions in cache design—such as AI-driven coherence, adaptive protocols, and hardware-software collaboration—hold promise, but their benefits must also be made transparent and accessible to non-expert users. Ultimately, improving both the efficiency and the interpretability of cache systems will be key to supporting the evolving demands of modern computing.

References

[1] M. Kumar and P. Arora, "A Survey of Cache Coherence Protocols in Multiprocessors with Shared Memory," UACEE International Journal of Computer Science and its Applications, vol. 2, no. 1, pp. 148–152, Feb. 2012.

[2] Z. Alwaisi and M. O. Agyeman, "An Overview of On-Chip Cache Coherence Protocols," in 2017 Intelligent Systems Conference (IntelliSys), London, UK, Sep. 2017, pp. 304–309, doi: 10.1109/IntelliSys.2017.8324309.

[3] Y. Yu, S. Lee, A. Khandelwal, and L. Zhong, “Soul: Generalized Cache Coherence for Efficient Synchronization,” arXiv preprint arXiv:2301.02576, version 3, May 2023. [Online]. Available: https://arxiv.org/abs/2301.02576

[4] H. M. Shukur, S. R. M. Zeebaree, R. R. Zebari, O. M. Ahmed, L. M. Haji, and D. M. Abdulqader, "Cache Coherence Protocols in Distributed Systems," J. Appl. Sci. Technol. Trends, vol. 1, no. 2, pp. 92–97, Jun. 2020, doi: 10.38094/jastt1329.

[5] Q. Tian, J. Li, F. Zheng, and S. Zhao, "A Cache Consistency Protocol with Improved Architecture," in Proc. ADHIP 2017, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 219, pp. 21–24, 2018, doi: 10.1007/978-3-319-73317-3_3.

[6] R. Cleaveland and C. Trippel, "Memory Consistency Model-Aware Cache Coherence for Heterogeneous Hardware," in Proc. Formal Methods in Computer-Aided Design (FMCAD), 2024. [Online]. Available: https://cs.stanford.edu/~trippel/pubs/rcleaveland_FMCAD24.pdf

[7] Y. Lu, W. Wang, and L. Chen, "Cache Coherence Over Disaggregated Memory," arXiv preprint arXiv:2409.02088, 2024. [Online]. Available: https://arxiv.org/html/2409.02088v4

[8] R. Carr, "Automatic Synthesis of Heterogeneous Cache Coherence Protocols," University of Edinburgh, 2021. [Online]. Available: https://www.research.ed.ac.uk/files/332078358/HeteroGen_OSWALD_DOA27102021_AFV.pdf

[9] A. Shpiner, A. Shiloh, and A. Schuster, "A Case Study for Broadcast on Intel Xeon Scalable Processors," in Proc. ACM SIGPLAN Symp. Principles and Practice of Parallel Programming (PPoPP), 2023. [Online]. Available: https://dl.acm.org/doi/fullHtml/10.1145/3605573.3605616