Sharing and Hit Based Prioritizing Replacement Algorithm for Multi-Threaded Applications

Volume 90 - Number 12
Year of Publication: 2014

Authors:
Muthukumar S
Jawahar P. K

Abstract

Cache replacement techniques like LRU, MRU etc. that are currently being deployed across multi-core architecture platforms, try to classify elements purely based on the number of hits they receive during their stay in the cache. In multi-threaded applications data can be shared by multiple threads (which might run on the same core or across different cores). Such data needs to be given more priority when compared to private data because miss on those data items may stall the functioning of multiple threads resulting in performance bottleneck. Since the traditional algorithms mentioned above do not possess this additional capability, they might lead to sub-optimal performance for most of the current multi-threaded applications. To address this limitation, our paper proposes a Sharing and Hit Based Prioritizing (SHP) replacement strategy that takes the sharing status of the data elements into consideration while making replacement decisions. Every cache element is associated with a 'Sharing Degree' which indicates the extent to which the element is shared based on the number of threads that try to access that element. There are four degrees of sharing namely – Not shared (or private), lightly shared, heavily shared and very heavily shared. Combining the sharing degree along with the number of hits received by the element, we embark on a priority based on which the replacement decisions are made. Evaluation results obtained using multi-threaded workloads derived from the PARSEC benchmark suite shows an average
improvement of 4% to 5% in the overall hit rate when compared to LRU algorithm.

References

- Christian Bienia, Sanjeev Kumar, Jaswinder Pal Singh, Kai Li, "The PARSEC

Index Terms
Computer Science Distributed Systems

Keywords
Sharing Cache Replacement Hits Threads.