Abstract

Current generation high performance multi-core processors have large shared cache memories. This shared cache memory is accessible by multiple cores. Concurrently running threads under each core do not always demand the entire capacity of the shared cache. Threads running on different cores accessing shared cache concurrently may result in higher cache miss rate and significant performance degradation due to inter-thread cache conflicts and lack of cache space. The cache capacity is the quantity of physical cache memory available with the processor. To achieve certain higher degree of processing performance on multi-core processors, efficient shared cache memory usage plays the defining role. The overall processor performance gets more sensitive to the problem of shortage of cache capacity, as threads sharing the cache compete for their requirement of the cache sizes. In this paper, a cache friendly and capacity conscious thread scheduling strategy is proposed for multi-core processors with multiple shared caches. The proposed scheduling policy ensures that the shared cache is optimally used by the competing threads which minimizes inter-thread resource conflict and hence reduces performance degradation. According to the experimental results the proposed policy reduces
shared cache contention significantly thereby improving the overall performance among threads by up to 5%.

References

8. James H. Anderson, J M Calendrino, and U C Devi, 2006 “Real time scheduling on multicore platforms”, in the proceedings of the 12th IEEE Real Time and Embedded Technology and Applications Symposium (RTAS ’06), San Jose California, USA, April 4-7.


Index Terms

Computer Science
Circuits and Systems

Keywords

CMP, Cache Capacity, Thread Scheduling, Shared Cache.