| Title | Qualitative Study of Contention-aware Scheduling Algorithm for Asymmetric Multicore Processors |
| --- | --- |
| Author(s) Name | 1. Choudhury and Kamruddin Nur
 |
| Contact Email(s) | kamruddin@aiub.edu |
| Published Conference Name | International Conference on Computing Advancements,ICCA 2020 |
| Type of Publication | Conference |
| Volume |  | Issue |  |
| Publisher | ACM |
| Publication Date | 2020 |
| ISSN | 978-1-4503-7778-2 |
| DOI | <https://doi.org/10.1145/3377049.3377135> |
| URL | <https://dl.acm.org/doi/10.1145/3377049.3377135> |
| Other Related Info. | Page 441 - 445 |
|  |

| Abstract |  |
| --- | --- |
| For last few decades, multitasking is at its highest demand. To achieve multitasking, symmetric & asymmetric multi-core processors system is a popular technology. Asymmetric multi-core processors (AMPs) use the same instruction set architecture (ISA) but different clock frequency. It is shown that AMPs deliver better performance per watt comparing to its symmetric counterpart. The future multi-core system will combine a few fast cores & many slow cores. Fast core means high power consumption with complex pipelines and high clock frequency, where the slow core will have low power consumption with simple pipelines and low clock frequency. To get the best performance from the asymmetric multi-core processors, the best scheduling policy will play an important role. Scheduling co-running applications in the most suitable core types are very vital for AMPs to get its best performance. Various policies like contention-aware, parallelism-aware & asymmetric-aware need to be considered when developing a scheduling algorithm. For AMPs, contention for resource sharing is a key performance-limiting factor. Despite noteworthy research efforts, the contention for resource sharing in the multi-core processor remains unsolved. In this paper, we discuss the latest five contention-aware scheduling algorithms of AMPs. We present a comparative study exploiting the technique, parameter & performance improvement so that the future computer scientist can develop a contention-aware solution more precisely. |