Developing Operating System for High-Performance Computing with a Hardware Approach to the Transactional Memory Support

Rozowski, Wojciech

As we know from the literature, multi-core processors do not achieve as high performance as one could expect. Unfortunately, there is no straightforward linear relationships between processor’s efficiency and the number of its cores. High time costs are caused by using synchronization mechanisms, which prevent losing integrity of shared data. I created a synchronization element (elided_lock() and elided_unlock() functions), which combines the hardware transactional memory instructions –TSX technology with spinlocks, guaranteeing optimal choice of synchronization method, depending on the size of shared data. The functions are designed to work with existing programming model. This paper introduces multicore and multithreaded Unix-based operating system, that uses - in its kernel layer - hardware transactional memory, for collision-free access to shared data. Developed system distribution includes a package of Unix platform’s utility applications, adapted to work with the new synchronization functions, as developed programming libraries support new mechanisms. Empirical evidence collected on simulated architecture shows the proper performance of developed mechanisms. Previous studies proved that using TSX in software layer, compared to spinlocks, guarantees 41% increase in performance of handling critical data, smaller than 256 cache lines. Created element might ensure optimization of handling critical sections on the same or even higher level of performance. Developed solutions could be used in parallel and distributed operating systems, especially in High-Performance Computing systems, consisting of hundreds of processors. In those systems, because of their scale, even a small change in the low-level layer of software could make a significant increase in performance.

Awards Won:

Fourth Award of $500