Critique Week 3

Andres Calderon - SID:861243796

January 27, 2016

Algorithms for Scalable Synchronization on Shared-Memory Multiprocessors (Mellor-Crummey and Scott, 1991)

The paper addresses the issue of synchronization on shared-memory multiprocessors through a detailed explanation and comparison of algorithms for the implementation of spin locks and barriers (however, barriers are not covered in this critique). The paper mentions two approaches to deal with synchronization: blocking and busy-wait algorithms. It clarifies that blocking is not appropriate to implement in the kernel of a O.S., so it focuses on two well-known problems that affect the performance of busy-wait implementations: hot spots and dance halls.

Some hardware support was already available to address those problems, for example: fetch_and_ϕ operations, multi-stage networks or special-purpose caches. Still, the authors stress the assumption that software implementations could be enough to face the problems. The main discussion of the paper focuses in the analysis of 5 spin lock alternatives (in addition with the discussion about barriers). The paper address spin locks in section 2, barriers in section 3 and a deep performance analysis of the alternatives in section 4.

Regarding to spin locks, the authors describe and present pseudo-codes for 5 algorithms. The simple test_and_set lock, where the paper proposes the use of exponential backoff to reduce traffic. The ticket lock, similar to a semaphore but still suffering of high contention (the notion of proportional backoff is introduced as a key solution). Two implementations of array-based queuing, where the main characteristic is the use of atomic fetch_and_ϕ operations. Finally, the authors proposed a new list-based queuing algorithms: the MCS lock. It shares the main advantages of the previous algorithms but adds independence to cache coherence which is unique among them.

Section 4 performs an extend performance comparison of the discussed locks. The analysis was run on both a cache-coherent multiprocessor and a multiprocessor with distributed shared memory. It is interesting that the results do not show an unique superior performance for one of the algorithms, even for that proposed by the authors. The results depend on the context and the performance of the algorithms is quite similar in several cases. The authors reflect this fact in the discussion where recommend different setting according to the available resources in the machine. It is also true that the authors stress that spin locks can be evaluated on the basis of several criteria, so it is difficult to say that one outweighs the others.

In my opinion, although the topic is highly technical the explanations on section 2 help to draws the big picture of the problem and the proposed solutions. The release of pseudo-code for each algorithm is a great asset. It ensures reproducibility and understanding of the ideas (even if it is difficult to follow). The performance comparison section is appropriate and detailed. The results are objective, clear and well discussed. It is also valuable the final recommendations, in particular for future multiprocessor design in order to provide the minimal mechanisms to address the problems discussed in the paper.