Hardware Scheduling Algorithms for Asymmetric Single-ISA CMPs

Nikola Markovic, Daniel Nemirovsky, Osman Unsal, Mateo Valero, Adrian Cristal
Barcelona Supercomputing Center
Universitat Politecnica de Catalunya
{nikola.markovic, daniel.nemorivsky,osman.uunsal,mateo.valero,adrian.cristal}@bsc.es

Abstract

As thread level parallelism in applications has continued to expand, so has research in chip multi-core processors. Since more and more applications become multi-threaded we expect to find a growing number of threads executing on a machine. Consequently, the operating system will require increasingly larger amounts of CPU time to schedule these threads efficiently. Instead of perpetuating the trend of performing more complex thread scheduling in the operating system, we propose a two lightweight hardware thread scheduling mechanisms. First is a Hardware Round-Robin Scheduling (HRRS) policy which is influenced by Fairness Scheduling techniques thereby reducing thread serialization and improving parallel thread performance. Second is a Thread Lock Section-aware Scheduling (TLSS) policy which extends HRRS policy. TLSS policy is influenced by the Fairness-aware Scheduling and bottleneck identification techniques. It complements the HRRS scheduler by identifying multi-threaded application bottlenecks such as thread synchronization sections. We show that HRRS outperforms Fairness scheduler by 17 percent while TLSS outperforms HRRS by 11 percent on an ACMP consisted of one large (out-of-order) core and three small (in-order) cores.

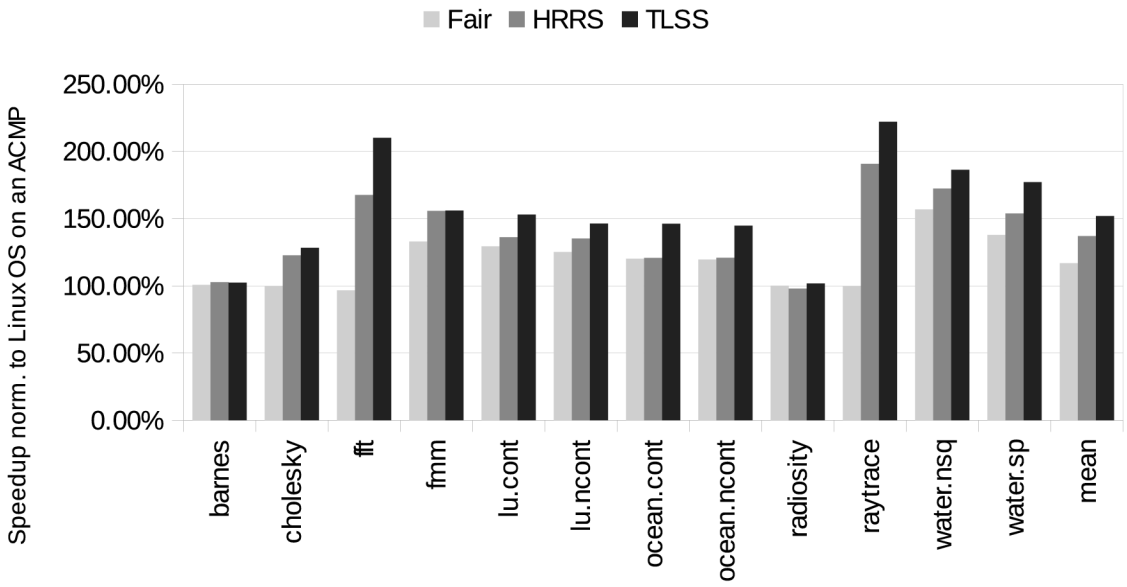

Fig1. Speedup comparison of the TLSS, the HRRS and the Fairness scheduler normalized to Linux OS scheduler for the SPLASH-2 benchmark suite running on four cores (1 OoO+3 InO).